

**PENGENALAN ALFABET BAHASA ISYARAT TANGAN  
PADA CITRA DIGITAL MENGGUNAKAN PENDEKATAN  
*CONVEX HULL* DAN *CONVOLUTIONAL NEURAL NETWORK*  
(CNN)**

**SKRIPSI**



**Oleh :**

**WIDIANTO EKA SAPUTRO**  
**170411100074**

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS TRUNOJOYO MADURA  
2022**

**Pengenalan Alfabet Bahasa Isyarat Tangan pada Citra Digital Menggunakan Pendekatan Convex Hull dan Convolutional Neural Network (CNN)**

**SKRIPSI**

**Diajukan untuk Memenuhi Persyaratan Penyelesaian Studi Strata Satu (S1)  
dan Memperoleh Gelar Sarjana Komputer (S.Kom) di Universitas  
Trunojoyo Madura**

**Oleh :**

**Widianto Eka Saputro**

**170411100074**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2022**

## HALAMAN PENGESAHAN

### Pengenalan Alfabet Bahasa Isyarat Tangan pada Citra Digital Menggunakan Pendekatan Convex Hull dan Convolutional Neural Network (CNN)

Skripsi diajukan sebagai salah satu syarat untuk menyelesaikan pendidikan program studi S1 Teknik Informatika Universitas Trunojoyo Madura

Oleh :

Nama : Widiyanto Eka Saputro

NIM : 170411100074

Disetujui oleh Tim Penguji Skripsi :

Tanggal Sidang : 10 Juni 2022

Dr. Budi Dwi Satoto, S.T., M.Kom.  
NIP. 19750909 200212 1 001

(Pembimbing I)

Moch Kautsar Sophan, S.Kom, M.MT.  
NIP. 19770713 200212 1 0047

(Pembimbing II)

Dr. Indah Agustien Siradjuddin,  
S.Kom., M.Kom.  
NIP. 19780820 200212 2 001

(Penguji I)

Dwi Kuswanto, S.Pd., M.T.  
NIP. 19740221 200801 1 006

(Penguji II)

Devie Rosa Anamisa, S.Kom., M.Kom.  
NIP. 19841104 200812 2 003

(Penguji III)

Bangkalan, 10 Juni 2022

Mengetahui,

Ketua Jurusan Teknik Informatika

Dr. Yeni Kustiyahningsih, S.Kom., M.Kom.  
NIP. 19770921 200812 2 002

## HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan dibawah ini :

|                   |   |                                                                                                                                                     |
|-------------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama              | : | Widianto Eka Saputro                                                                                                                                |
| NIM               | : | 170411100074                                                                                                                                        |
| Judul Tugas Akhir | : | Pengenalan Alfabet Bahasa Isyarat Tangan Pada Citra Digital Menggunakan Pendekatan <i>Convex Hull</i> Dan <i>Convolutional Neural Network</i> (CNN) |

Menyatakan dengan sebenarnya bahwa penulisan tugas akhir ini berdasarkan hasil penelitian, pemikiran dan pemaparan saya sendiri, bukan merupakan karya pihak lain serta belum pernah diajukan untuk mendapatkan gelar akademik Sarjana Komputer baik dalam lingkungan Universitas Trunojoyo Madura maupun di Perguruan Tinggi lain. Jika terdapat karya orang lain sebagai referensi pada penelitian ini, maka saya akan mencantumkan sumber yang jelas.

Dengan pernyataan ini saya buat dengan sesungguhnya dan apabila terbukti tugas akhir ini sebagian atau seluruhnya merupakan hasil plagiasi atau terdapat halhal yang tidak sesuai dengan pernyataan diatas, saya sanggup menerima sanksi akademis sesuai peraturan yang berlaku di Universitas Trunojoyo Madura.

Bangkalan, 10 Juni 2022

Pembuat pernyataan,

**Widianto Eka Saputro**  
NIM. 170411100074

# **PENGENALAN ALFABET BAHASA ISYARAT TANGAN PADA CITRA DIGITAL MENGGUNAKAN PENDEKATAN *CONVEX HULL* DAN *CONVOLUTIONAL NEURAL NETWORK (CNN)***

Widianto Eka Saputro

(170411100074)

Pembimbing I : Dr. Budi Dwi Satoto, S.T., M.Kom

Pembimbing II: Moch Kautsar Sophan, S.Kom, M.MT

## **ABSTRAK**

Bahasa isyarat merupakan suatu sarana komunikasi dalam masyarakat, terlebih bagi penyandang tunarungu. Namun masyarakat pada umumnya masih sangat awam perihal bahasa isyarat, terlebih bahasa isyarat dipandang masyarakat umum bukanlah bahasa yang universal. Hal ini menyebabkan meningkatnya kesenjangan komunikasi komunitas tunarungu dengan masyarakat. Sementara itu metode komunikasi tertulis dirasa hanya efisien bila digunakan disaat tertentu. Berdasarkan hal ini, beberapa penelitian telah dilakukan untuk mempermudah masyarakat umum dalam mengenali bahasa isyarat. Salah satunya adalah pengenalan lewat citra digital. Selain itu, akhir-akhir ini tema *machine learning* menjadi topik yang hangat untuk dibahas, Namun pada beberapa penelitian yang sudah ada, penggunaan machine learning pada citra masih menggunakan masukan citra mentah yang diperoleh langsung dari penangkapan citra aslinya. hal ini berpotensi menangkap citra objek lain yang bukan menjadi parameter pengenalan objek dan juga dapat menghambat saat proses pembelajaran melalui machine learning. dalam penelitian ini Peneliti mencoba mengkombinasikan pengolahan citra digital dengan *machine learning* untuk mengenali gesture tangan yang menjadi alphabet bahasa isyarat dengan menggunakan *Convex Hull* pada saat proses preprocessing citra sebelum dimasukan dalam proses pembelajaran dengan metode bernama *Convolutional Neural Network (CNN)*. Penggunaan metode berupa *Convolutional Neural Network* dikarenakan metode ini menggunakan tepat susunan dalam mengenali sebuah citra dua dimensi. Serta penggunaan *Convex Hull* pada tahap pre-processing diharapkan dapat meningkatkan akurasi pengenalan alphabet bahasa isyarat tangan.

**Kata kunci :** Bahasa Isyarat, CNN, *Convex Hull*

## KATA PENGANTAR

Segala puji dan syukur selalu penulis panjatkan atas kehadiran Allah SWT yang telah memberi rahmat, hidayat, serta karunia-Nya, tidak lupa shalawat serta salam penulis curahkan kepada nabi besar Muhammad SAW beserta para sahabat dan keluarganya, berkat dorongan dan bantuan dari berbagai pihak yang telah membantu menyelesaikan tugas akhir ini dengan judul “PENGENALAN ALFABET BAHASA ISYARAT TANGAN PADA CITRA DIGITAL MENGGUNAKAN PENDEKATAN *CONVEX HULL* DAN *CONVOLUTIONAL NEURAL NETWORK* (CNN)” pada kesempatan ini, penulis ingin menyampaikan rasa terimakasih kepada:

1. Keluarga yang selalu mendukung dan selalu mendoakan keberhasilan penulis.
2. Bapak Dr. Budi Dwi Satoto, S.T., M.Kom. dan Bapak Moch Kautsar Sophan, S.Kom, M.MT. selaku dosen pembimbing yang telah memberikan arahan dan motivasi untuk penulis, sehingga penulis bisa menyelesaikan skripsi.
3. Ibu Fika Hastarita Rachman, S.T., M.Eng selaku Koordinator Prodi Teknik Informatika dan Ibu Dr. Yeni Kustiyahningsih, S.Kom., M.Kom. selaku Ketua jurusan Teknik Informatika.
4. Bapak dan Ibu Dosen Teknik Informatika di Universitas Trunojoyo Madura yang telah memberikan ilmu pengetahuan dalam proses perkuliahan.
5. Rusnia Dyah Wardani yang selalu memberikan semangat serta dorongan untuk tidak menyerah.

Dalam penulisan skripsi ini, penulis menyadari bahwa masih banyak kekurangan. Oleh karena itu, kritik dan saran dari semua pihak sangat diharapkan oleh penulis untuk memberikan perubahan positif agar menjadi lebih baik. Demikian, penulis menyusun laporan skripsi ini dengan harapan dapat bermanfaat bagi semua pihak dan juga bagi penulis sendiri.

Bangkalan, 10 Juni 2022

**Widianto Eka Saputro**  
NIM. 170411100074

## DAFTAR ISI

|                                                 |     |
|-------------------------------------------------|-----|
| ABSTRAK .....                                   | V   |
| KATA PENGANTAR .....                            | VI  |
| DAFTAR ISI .....                                | VII |
| DAFTAR TABEL .....                              | X   |
| DAFTAR GAMBAR .....                             | XII |
| BAB I PENDAHULUAN .....                         | 14  |
| 1.1 LATAR BELAKANG .....                        | 14  |
| 1.2 PERUMUSAN MASALAH .....                     | 16  |
| 1.2.1 PERMASALAHAN .....                        | 16  |
| 1.2.2 USULAN SOLUSI .....                       | 16  |
| 1.2.3 PERTANYAAN PENELITIAN .....               | 16  |
| 1.3 TUJUAN DAN MANFAAT .....                    | 17  |
| 1.3.1 TUJUAN .....                              | 17  |
| 1.3.2 MANFAAT .....                             | 17  |
| 1.4 BATASAN .....                               | 17  |
| 1.5 SISTEMATIKA LAPORAN .....                   | 18  |
| BAB II TINJAUAN PUSTAKA .....                   | 19  |
| 2.1 LANDASAN TEORI .....                        | 19  |
| 2.1.1 BAHASA ISYARAT .....                      | 19  |
| 2.1.2 CITRA .....                               | 20  |
| 2.1.2.1 JENIS-JENIS CITRA .....                 | 20  |
| 2.1.2.2 PENGOLAHAN CITRA DIGITAL .....          | 20  |
| 2.1.3 <i>CONVEX HULL</i> .....                  | 21  |
| 2.1.3.1 ALGORITMA PENCARIAN KONTUR TANGAN ..... | 22  |
| 2.1.3.2 ALGORITMA <i>CONVEX HULL</i> .....      | 22  |

|                                   |                                           |    |
|-----------------------------------|-------------------------------------------|----|
| 2.1.4                             | JARINGAN SYARAF TIRUAN.....               | 25 |
| 2.1.5                             | <i>MACHINE LEARNING</i> .....             | 25 |
| 2.1.6                             | <i>SUPERVISED LEARNING</i> .....          | 26 |
| 2.1.7                             | <i>DEEP LEARNING</i> .....                | 26 |
| 2.1.8                             | <i>CONVOLUTIONAL NEURAL NETWORK</i> ..... | 26 |
| 2.1.8.1                           | Layer Konvolusi.....                      | 27 |
| 2.1.8.2                           | Layer <i>Pooling</i> .....                | 28 |
| 2.1.8.3                           | Layer Terkoneksi Penuh .....              | 29 |
| 2.1.8.4                           | Layer <i>Softmax</i> .....                | 29 |
| 2.1.9                             | OPENCV .....                              | 29 |
| 2.1.10                            | TENSOFLOW .....                           | 30 |
| 2.1.11                            | KERAS.....                                | 30 |
| 2.2                               | PENELITIAN TERKAIT .....                  | 30 |
| BAB III METODE PENELITIAN.....    |                                           | 33 |
| 3.1                               | JENIS PENELITIAN .....                    | 33 |
| 3.2                               | LANGKAH-LANGKAH PENELITIAN.....           | 33 |
| 3.3                               | DATASET .....                             | 34 |
| 3.4                               | ARSITEKTUR SISTEM.....                    | 35 |
| 3.5                               | METODE EVALUASI.....                      | 39 |
| 3.6                               | SKENARIO UJICOBA.....                     | 41 |
| BAB IV HASIL DAN PEMBAHASAN ..... |                                           | 43 |
| 4.1                               | LINGKUNGAN UJICOBA .....                  | 43 |
| 4.2                               | SCREENING DATA .....                      | 44 |
| 4.3                               | PEMBUATAN PROGRAM .....                   | 44 |
| 4.3.1                             | <i>PREPARE DATA</i> .....                 | 44 |
| 4.3.2                             | <i>PREPROCESSING</i> .....                | 45 |
| 4.3.3                             | <i>MAKE MODEL &amp; EVALUATE</i> .....    | 49 |
| 4.4                               | STRUKTUR MODEL CNN YANG DIGUNAKAN .....   | 52 |



|         |                                                              |     |
|---------|--------------------------------------------------------------|-----|
| 4.5     | HASIL UJICOBA .....                                          | 54  |
| 4.5.1   | SKENARIO 1 .....                                             | 54  |
| 4.5.1.1 | Hasil Pembelajaran Sistem <i>Convex Hull</i> + CNN.....      | 55  |
| 4.5.1.2 | Hasil Evaluasi Sistem <i>Convex Hull</i> + CNN.....          | 56  |
| 4.5.1.3 | Hasil Pembelajaran Sistem CNN Tanpa <i>Convex Hull</i> ..... | 57  |
| 4.5.1.4 | Hasil Evaluasi Sistem CNN Tanpa <i>Convex Hull</i> .....     | 59  |
| 4.5.2   | SKENARIO 2 .....                                             | 60  |
| 4.5.2.1 | Hasil Pembelajaran Sistem <i>Convex Hull</i> + CNN.....      | 60  |
| 4.5.2.2 | Hasil Evaluasi Sistem <i>Convex Hull</i> + CNN.....          | 62  |
| 4.5.2.3 | Hasil Pembelajaran Sistem CNN Tanpa <i>Convex Hull</i> ..... | 63  |
| 4.5.2.4 | Hasil Evaluasi Sistem CNN Tanpa <i>Convex Hull</i> .....     | 65  |
| 4.5.3   | SKENARIO 3 .....                                             | 66  |
| 4.5.3.1 | Hasil Pembelajaran Sistem <i>Convex Hull</i> + CNN.....      | 66  |
| 4.5.3.2 | Hasil Evaluasi Sistem <i>Convex Hull</i> + CNN.....          | 68  |
| 4.5.3.3 | Hasil Pembelajaran Sistem CNN Tanpa <i>Convex Hull</i> ..... | 69  |
| 4.5.2.4 | Hasil Evaluasi Sistem CNN Tanpa <i>Convex Hull</i> .....     | 70  |
| 4.5.4   | SKENARIO 4 .....                                             | 71  |
| 4.5.4.1 | Hasil Pembelajaran Sistem <i>Convex Hull</i> + CNN.....      | 71  |
| 4.5.4.2 | Hasil Evaluasi Sistem <i>Convex Hull</i> + CNN.....          | 87  |
| 4.5.4.3 | Hasil Pembelajaran Sistem CNN Tanpa <i>Convex Hull</i> ..... | 91  |
| 4.5.4.3 | Hasil Evaluasi Sistem CNN Tanpa <i>Convex Hull</i> .....     | 107 |
|         | BAB V KESIMPULAN DAN SARAN.....                              | 111 |
| 5.1     | KESIMPULAN .....                                             | 111 |
| 5.2     | SARAN .....                                                  | 112 |
|         | DAFTAR PUSTAKA .....                                         | 113 |
|         | LAMPIRAN.....                                                | 116 |

## DAFTAR TABEL

|                                                                                                      |     |
|------------------------------------------------------------------------------------------------------|-----|
| Tabel 3.1. Hasil Prediksi <i>Confusion Matrix</i> .....                                              | 39  |
| Tabel 3.3. Rancangan Tabel Skenario Ujicoba .....                                                    | 42  |
| Tabel 4.1. Perangkat Yang Digunakan .....                                                            | 43  |
| Tabel 3.2. Rancangan Tabel Hasil Prediksi .....                                                      | 121 |
| Tabel 4.2. Tabel Hasil Prediksi Skenario 1 Sistem <i>Convex Hull</i> + CNN .....                     | 123 |
| Tabel 4.3. Tabel Hasil Prediksi Skenario 1 Sistem CNN Tanpa <i>Convex Hull</i> ...                   | 125 |
| Tabel 4.4. Tabel Hasil Prediksi Skenario 2 Sistem <i>Convex Hull</i> + CNN .....                     | 127 |
| Tabel 4.5. Tabel Hasil Prediksi Skenario 2 Sistem CNN Tanpa <i>Convex Hull</i> ...                   | 129 |
| Tabel 4.6. Tabel Hasil Prediksi Skenario 3 Sistem <i>Convex Hull</i> + CNN .....                     | 131 |
| Tabel 4.7. Tabel Hasil Prediksi Skenario 3 Sistem CNN Tanpa <i>Convex Hull</i> ...                   | 133 |
| Tabel 4.8. Tabel Hasil Prediksi Skenario 4 Sistem <i>Convex Hull</i> + CNN Dengan 20 Epoch.....      | 135 |
| Tabel 4.9. Tabel Hasil Prediksi Skenario 4 Sistem <i>Convex Hull</i> + CNN Dengan 40 Epoch.....      | 137 |
| Tabel 4.10. Tabel Hasil Prediksi Skenario 4 Sistem <i>Convex Hull</i> + CNN Dengan 60 Epoch.....     | 139 |
| Tabel 4.11. Tabel Hasil Prediksi Skenario 4 Sistem <i>Convex Hull</i> + CNN Dengan 80 Epoch.....     | 141 |
| Tabel 4.12. Tabel Hasil Prediksi Skenario 4 Sistem <i>Convex Hull</i> + CNN Dengan 100 Epoch .....   | 143 |
| Tabel 4.13. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa <i>Convex Hull</i> Dengan 20 Epoch..... | 145 |
| Tabel 4.14. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa <i>Convex Hull</i> Dengan 40 Epoch..... | 147 |

|                                                                                                |     |
|------------------------------------------------------------------------------------------------|-----|
| Tabel 4.15. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa Convex Hull Dengan 60 Epoch.....  | 149 |
| Tabel 4.16. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa Convex Hull Dengan 80 Epoch.....  | 151 |
| Tabel 4.17. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa Convex Hull Dengan 100 Epoch..... | 153 |

## DAFTAR GAMBAR

|                                                                                                                         |    |
|-------------------------------------------------------------------------------------------------------------------------|----|
| Gambar 2.1 <i>Convex Hull</i> .....                                                                                     | 21 |
| Gambar 2.2 Menghubungkan Titik Pada Algoritma Graham Scan .....                                                         | 23 |
| Gambar 2.3 Algoritma Graham Scan.....                                                                                   | 24 |
| Gambar 2.4 Algoritma Jarvis March.....                                                                                  | 25 |
| Gambar 2.5 Struktur Pemodelan Jaringan Pada <i>Machine Learning</i> .....                                               | 25 |
| Gambar 2.6 Struktur Pemodelan Jaringan Pada <i>Deep Learning</i> .....                                                  | 26 |
| Gambar 2.7 Arsitektur <i>Convolutional Neural Network</i> .....                                                         | 27 |
| Gambar 2.8 Layer Konvolusi.....                                                                                         | 28 |
| Gambar 3.1 Diagram Alir Tahapan Penelitian.....                                                                         | 34 |
| Gambar 3.2 Dataset.....                                                                                                 | 35 |
| Gambar 3.3 Bagan Alir Rancangan Sistem .....                                                                            | 36 |
| Gambar 3.4 Struktur CNN .....                                                                                           | 38 |
| Gambar 4.1 (A) Dan (B) <i>Screening Data</i> .....                                                                      | 44 |
| Gambar 4.2 Struktur Model CNN.....                                                                                      | 53 |
| Gambar 4.3 (A) Dan (B) Hasil <i>Output</i> Pembelajaran Sistem Dengan <i>Convex Hull</i><br>+ CNN Pada Skenario 1 ..... | 56 |
| Gambar 4.4 (A) Dan (B) Hasil <i>Output</i> Pembelajaran Sistem CNN Tanpa <i>Convex Hull</i><br>Pada Skenario 1.....     | 59 |
| Gambar 4.5 (A) Dan (B) Hasil <i>Output</i> Pembelajaran Sistem Dengan <i>Convex Hull</i><br>+ CNN Pada Skenario 2.....  | 62 |
| Gambar 4.6 (A) Dan (B) Hasil <i>Output</i> Pembelajaran Sistem CNN Tanpa <i>Convex Hull</i><br>Pada Skenario 2.....     | 65 |
| Gambar 4.7 Hasil <i>Output</i> Pembelajaran Sistem Dengan <i>Convex Hull</i> + CNN Pada<br>Skenario 3 .....             | 67 |

|                                                                                                                                                                                                                                                                                                                                                                                              |     |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Gambar 4.8 (A) Dan (B) Hasil <i>Output</i> Pembelajaran Sistem CNN Tanpa <i>Convex Hull</i> Pada Skenario 3.....                                                                                                                                                                                                                                                                             | 70  |
| Gambar 4.9 (A) Dan (B) Hasil <i>Output</i> Pembelajaran Sistem Dengan <i>Convex Hull</i> + CNN Pada Skenario 4 Dengan 20 Epoch .....                                                                                                                                                                                                                                                         | 86  |
| Karena Terdapat Masalah Pada Saat Penggambaran Hasil <i>Output</i> Pembelajaran Untuk Epoch 40, 60, 80, Dan 100 Dimana Plot <i>Model Accuracynya</i> Bercampur Dengan Tabel Hasil Prediksi <i>Confusion_Matrix</i> Sehingga Yang Tergambarkan Hanya <i>Model Lossnya</i> Saja, Maka <i>Model Loss</i> Hasil Pembelajaran Untuk Epoch 40, 60, 80, Dan 100 Bisa Dilihat Pada Gambar 4.10. .... | 86  |
| Gambar 4.10 (A), (B), (C), Dan (D) Hasil Output Pembelajaran Sistem Dengan <i>Convex Hull</i> + CNN Pada Skenario 4 Dengan Epoch 40, 60, 80, Dan 100.....                                                                                                                                                                                                                                    | 87  |
| Gambar 4.11 (A) Dan (B) Hasil <i>Output</i> Pembelajaran Sistem Dengan <i>Convex Hull</i> + CNN Pada Skenario 4 Dengan 20 Epoch .....                                                                                                                                                                                                                                                        | 106 |
| <i>Model Loss</i> Hasil Pembelajaran Untuk Epoch 40 Sampai 100 Bisa Dilihat Pada Gambar 4.12. ....                                                                                                                                                                                                                                                                                           | 106 |
| Gambar 4.12 (A), (B), (C), Dan (D) Hasil Output Pembelajaran Sistem Dengan <i>Convex Hull</i> + CNN Pada Skenario 4 Dengan Epoch 40, 60, 80, Dan 100.....                                                                                                                                                                                                                                    | 106 |

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Komunikasi ialah suatu kegiatan yang penting dalam upaya bersosialisasi. Banyak jenis media komunikasi yang dapat berkomunikasi salah satunya adalah dengan menggunakan bahasa isyarat tangan. Bahasa isyarat adalah bahasa yang sangat visual-spasial, secara linguistik lengkap. Bahasa isyarat merupakan bahasa pertama dan alat komunikasi utama untuk penyandang tunarungu [1]. Bahasa isyarat tangan juga digunakan oleh orang-orang yang memiliki kelainan perilaku seperti autisme dan down-syndrome. Bahasa isyarat tangan yang umum digunakan adalah *American Sign Language* yang resmi diakui dan digunakan di Indonesia dengan sebutan SIBI (Sistem Isyarat Bahasa Indonesia) [2]. Pada dasarnya bahasa isyarat memanfaatkan gerakan tubuh, wajah, tangan, dan bibir untuk mempresentasikan sebuah kata namun ada juga bahasa isyarat yang berupa alfabet yang digunakan untuk membantu proses komunikasi untuk kata-kata yang tidak memiliki bahasa tubuh dalam bahasa isyarat.

Bahasa isyarat alfabet merupakan alat bantu dasar yang digunakan para pengajar untuk mengajar para penyandang tunarungu dan tunawicara untuk mengenal huruf alfabet dasar. Bentuk komunikasi berupa bahasa isyarat dilakukan dengan gerakan-gerakan tubuh, tangan, wajah, dan bibir. Namun, gerakan tubuh berupa bibir tidak digunakan secara lisan [3]. Banyak masyarakat yang masih kesulitan dalam berkomunikasi dengan kalangan tunarungu dikarenakan kurangnya wawasan masyarakat tentang bahasa isyarat tangan. Terlebih bahasa isyarat dalam kepercayaan populer bukanlah bahasa yang universal.

Hal ini menyebabkan semakin meningkatnya kesenjangan komunikasi antara komunitas tunarungu dengan masyarakat umum. Sementara itu metode komunikasi tertulis memerlukan waktu dan terkesan lambat saat dilakukan, terlebih hanya dapat dilakukan ketika dalam keadaan duduk diam. Komunikasi tertulis akan terasa canggung saat berjalan atau bergerak. Selain itu, komunitas tunarungu pada umumnya kurang terampil dalam menulis bahasa lisan.

Dengan perkembangan teknologi yang ada saat ini mempelajari bahasa isyarat sangat mudah dilakukan karena banyaknya media yang memfasilitasi hal

tersebut. Salah satunya menggunakan metode berupa jaringan syaraf tiruan. Metode tersebut diimplementasikan ke dalam *machine learning*. Beberapa penelitian mengenai pengenalan bahasa isyarat juga telah mengadopsi penerapan *machine learning* sebagai media pembelajaran untuk mengenali bahasa isyarat tangan dengan berbagai macam struktur jaringan syaraf tiruan.

Namun pada beberapa penelitian yang telah dilakukan sebelumnya, seperti pada penelitian oleh Mochamad Bagus Setiyo Bakti dan Yuliana Melita Pranoto dalam penelitian dengan judul “Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode *Convolutional Neural Network*” pada tahun 2019 [4] dan penelitian yang dilakukan oleh Kumud Alok, Anurag Mehra, Ankita Kaushik, Aman Verma dalam penelitian yang berjudul “*Hand Sign Recognition using Convolutional Neural Network*” [5]. Pada penelitian tersebut pengenalan isyarat tangan dilakukan menggunakan CNN (*Convolutional Neural Network*). Penggunaan *machine learning* pada citra masih menggunakan masukan citra mentah yang diperoleh langsung dari penangkapan citra aslinya. hal ini berpotensi menangkap citra objek lain yang bukan menjadi parameter pengenalan objek, serta juga dapat menghambat proses pembelajaran melalui machine learning berjalan .

Maka dari itu untuk mengefisienkan pembelajaran menggunakan *machine learning*, diperlukan manipulasi pada citra masukan menggunakan algoritma *Convex Hull*, sehingga citra masukan ke machine learning tidak lagi mengandung objek yang tidak sesuai dengan apa yang ingin dipelajari. Penulis memilih menggunakan metode algoritma *Convex Hull* karena algoritma ini dianggap efisien untuk mengurangi objek yang tidak sesuai dengan apa yang ingin dipelajari. Dengan menggunakan metode ini, citra Bahasa Isyarat yang awalnya berupa *image* dengan kualitas buruk dan banyak terdapat *noise* bisa ditingkatkan kualitasnya dengan agoritma *Convex Hull* serta *noise* yang terdapat pada image bisa diminimalkan. Sedangkan untuk proses machine learningnya, penulis memilih menggunakan metode *Convolutional Neural Network* karena dianggap paling populer untuk menganalisis citra visual, selain itu *Convolutional Neural Network* juga dianggap sebagai metode yang memiliki tingkat akurasi yang tinggi dalam kelas Jaringan Syaraf Tiruan. Dalam penelitian ini penulis mencoba

merancang sebuah sistem pengenalan alfabet bahasa isyarat menggunakan pendekatan *Convolutional Neural Network* serta didukung dengan pengolahan citra digital berupa *Convex Hull* dengan harapan dapat membuat proses pembelajaran pada *machine learning* lebih efisien karena meminimalisir objek lain yang bukan parameter pengenalan masuk ke dalam proses machine learning.

## **1.2 Perumusan Masalah**

Berdasarkan latar belakang diatas maka terbentuklah beberapa rumusah masalah yang akan dijelaskan pada sub bab berikut.

### **1.2.1 Permasalahan**

Banyak penelitian mengenai pengenalan bahasa isyarat yang telah mengadopsi penerapan machine learning sebagai media pembelajaran untuk mengenali bahasa isyarat tangan dengan berbagai macam struktur jaringan syaraf tiruan. Namun pada beberapa penelitian yang sudah ada, penggunaan machine learning pada citra masih menggunakan masukan citra mentah yang diperoleh langsung dari penangkapan citra aslinya. hal ini berpotensi menangkap citra objek lain yang bukan menjadi parameter pengenalan objek, serta juga dapat menjadi riak saat proses pembelajaran melalui *machine learning* berjalan

### **1.2.2 Usulan Solusi**

Untuk mengefisienkan pembelajaran menggunakan *Convolutional Neural Network* (CNN), diperlukan manipulasi pada citra masukan menggunakan algoritma *convex hull*, sehingga citra masukan ke *Convolutional Neural Network* (CNN) tidak lagi mengandung objek yang tidak sesuai dengan apa yang ingin dipelajari.

### **1.2.3 Pertanyaan Peneliti**

Berikut pertanyaan – pertanyaan penelitian terkait proposal skripsi yang berjudul “Pengenalan Alfabet Bahasa Isyarat Tangan Pada Citra Digital Menggunakan Pendekatan *Convex Hull* Dan *Convolutional Neural Network* (CNN)” :

1. Berapa akurasi pengenalan alfabet Bahasa Isyarat tangan menggunakan pendekatan *Convex Hull* dan *Convolutional Neural Network* (CNN)?



2. Apakah akurasi pengenalan alfabet Bahasa Isyarat tangan menggunakan pendekatan *Convex Hull* dan *Convolutional Neural Network* (CNN) lebih baik dibandingkan dengan menerapkan metode berupa *Convolutional Neural Network* (CNN) saja ?

### **1.3 Tujuan Dan Manfaat**

Berdasarkan rumusan masalah dan usulan solusi pada sub bab sebelumnya, didapatkan tujuan dan manfaat yang akan dijelaskan pada sub bab berikut.

#### **1.3.1 Tujuan**

Adapun tujuan penelitian meliputi :

1. Untuk mengetahui akurasi pengenalan alfabet bahasa isyarat tangan menggunakan pendekatan *Convex Hull Convolutional Neural Network* (CNN).
2. Untuk mengetahui apakah akurasi pengenalan alfabet bahasa isyarat tangan menggunakan pendekatan *Convex Hull Convolutional Neural Network* (CNN) lebih baik jika dibandingkan dengan menerapkan metode *Convolutional Neural Network* (CNN) saja.

#### **1.3.2 Manfaat**

Beberapa manfaat dalam usulan penelitian yang akan dilakukan meliputi :

1. Dapat menjadi sebuah usaha dalam pengoptimalan pembelajaran model machine learning dengan meminimalisir parameter yang tidak diperlukan dalam citra.
2. Dapat membantu pengenalan alfabet bahasa isyarat dalam komunikasi masyarakat umum dengan komunitas yang menggunakan bahasa isyarat

### **1.4 Batasan**

Beberapa batasan dalam usulan penelitian yang akan dilakukan meliputi :

1. Penelitian yang akan dilakukan tidak mengulas mengenai permasalahan pengenalan bersifat *realtime*.
2. Keluaran yang dihasilkan merupakan model hasil pembelajaran.
3. Pengenalan dilakukan pada 26 alfabet mulai dari A sampai dengan Z.

4. Dataset terdiri dari 26 class dengan jumlah data 1500 pada tiap *class*.
5. Pembagian data *Training* dan *Testing* menyesuaikan dengan pembagian pada Skenario Ujicoba.
6. Convex Hull dilakukan secara manual tanpa bantuan *library* ataupun fungsi yang disediakan oleh OpenCV.
7. Penelitian ini tidak membahas kecepatan pengenalan.

## **1.5 Sistematika Laporan**

Beberapa sistematika di dalam laporan skripsi yang akan dilakukan meliputi:

### **BAB I PENDAHULUAN**

Bab I akan mengulas latar belakang dimana pengenalan Bahasa Isyarat tangan menjadi latar belakang pada penelitian ini, perumusan masalah yang telah dirumuskan oleh penulis untuk melakukan penelitian, tujuan dan manfaat dari penelitian, Batasan-batasan masalah pada penelitian ini, serta sistematika laporan.

### **BAB II KAJIAN PUSTAKA**

Bab II akan membahas beberapa ilmu terkait dengan metode yang digunakan dan beberapa penelitian terkait.

### **BAB III METODE USULAN**

Bab III akan mengulas jenis penelitian, langkah-langkah penelitian, perancangan sistem, dataset yang digunakan, metode evaluasi, skenario ujicoba, dan perkiraan jadwal penelitian.

### **BAB IV HASIL DAN PEMBAHASAN**

Bab IV ini akan mengulas tentang hasil implementasi sesuai arsitektur sistem dan hasil uji coba sesuai dengan skenario uji coba.

### **BAB V KESIMPULAN DAN SARAN**

Bab V ini akan mengulas tentang kesimpulan dan saran, kesimpulan didapatkan dari hasil uji coba yang telah dilakukan dan saran untuk pengembangan penelitian terkait selanjutnya

## **BAB II TINJAUAN PUSTAKA**

### **2.1 Landasan Teori**

Pada sub bab ini akan dilakukan pembahasan teori-teori yang terkait dengan penelitian yang dilakukan.

#### **2.1.1 Bahasa Isyarat**

Bahasa merupakan sarana dalam berkomunikasi. Dengan menggunakan bahasa, masyarakat dapat melakukan komunikasi dengan beberapa orang di sekitar mereka. Namun, penggunaan bahasa sebagai sarana komunikasi tidak menjamin kita bisa saling mengerti. Hal tersebut terjadi karena, agar dapat mengerti kita perlu mengerti bahasa yang digunakan. Bagi orang-orang dengan keterbatasan berbicara maupun keterbatasan mendengar, mereka menggunakan bahasa isyarat untuk berkomunikasi. Bahasa isyarat adalah suatu sarana untuk berkomunikasi dimana penyampaiannya dengan menggunakan bahasa tubuh, bahasa tubuh bisa berupa gerakan tubuh, wajah, tangan, dan bibir untuk mempresentasikan sebuah kata tetapi tanpa secara lisan dalam mengekspresikan sebuah pikiran dari penggunanya [3].

Bahasa isyarat dipelajari oleh siapapun, terutama waktu masih kecil dan belum bisa berbicara secara lisan. Namun bagi orang-orang dengan keterbatasan seperti penyandang Tunarungu atau Tunawicara Bahasa Isyarat digunakan untuk berkomunikasi dalam kehidupan sehari-hari, karena mereka tidak bisa berbicara dengan baik secara lisan, mereka harus menggunakan gerakan tangan, tubuh, maupun gerakan bibir mereka agar orang lain dapat mengerti apa yang mereka ingin sampaikan.

Bahasa isyarat yang digunakan dalam berkomunikasi dibagi berdasarkan daerah penggunaannya seperti pada negara Amerika, mereka memiliki sebutan untuk bahasa isyarat yang mereka gunakan, yaitu *American Sign Language* (ASL), sedangkan Indonesia terdapat sebuah pengaturan berupa Bahasa isyarat yang disebut dengan *Indonesian Sign Language* (ISL), bisa juga disebut sebagai sistem isyarat Bahasa Indonesia (SIBI).

### 2.1.2 Citra

Citra ialah suatu perpaduan milik suatu titik, garis, bidang dan warna yang mewujudkan suatu kemiripan dengan objek tertentu. Citra tersebut dapat berupa hasil dari suatu alat perekam informasi, dapat bersifat optikal seperti foto, dan dapat juga berupa sinyal video sederhana [6]. Citra memiliki atribut atau karakter tersendiri yang tidak dimiliki oleh data yang berupa teks.. Ada peribahasa yang mengatakan "sebuah gambar bernilai lebih dari 1.000 kata" makna pepatah adalah bahwa gambar dapat memberikan lebih banyak informasi daripada data dalam bentuk teks.

Citra bisa juga didefinisikan sebagai suatu fungsi  $f(x,y)$ . fungsi tersebut memiliki ukuran berupa  $M * N$  (baris\*kolom), dimana titik koordinat spasial  $x,y$ , serta amplitudo  $f$  pada titik  $(x,y)$  tersebut menjadi intensitas, biasa disebut tingkat dari keabuan suatu citra yang akan digunakan [7].

#### 2.1.2.1 Jenis-Jenis Citra

Terdapat dua macam jenis citra. Jenis citra tersebut meliputi citra analog serta citra digital.

1. Citra analog tergolong suatu bentuk citra yang bersifat *continue*. Sifat tersebut dicontohkan berupa gambar yang ditampilkan oleh monitor dari televisi, lukisan, gambar yang dicetak dan lain sebagainya. Citra analog tidak bisa direpresentasikan dan diproses melalui komputer secara langsung. Untuk dapat diproses melalui komputer dengan otomatis, citra analog perlu dikonversikan sebagai citra digital.
2. Citra digital merupakan penggambaran dari citra dengan intensitas dari cahaya berbentuk diskrit dengan bidang berupa dua dimensi. Jenis citra digital berupa sebuah matriks dimana titik koordinat dari suatu citra diwakili oleh indeks baris dan kolom dari matriks tersebut, titik-titik ini disebut sebagai suatu elemen gambar(*picture element*) atau piksel(*pixel*) yang membuktikan tingkat dari keabuan milik titik.

#### 2.1.2.2 Pengolahan Citra Digital

Berbeda dengan citra analog, citra digital diolah secara langsung melalui komputer, hal ini disebut dengan pengolahan citra digital. Pengolahan tersebut merupakan sebuah pemrosesan dari citra digital melalui pemanfaatan komputer.

Citra digital diolah dengan susunan berupa beberapa elemen yang terbatas, yang mana setiap citra memiliki lokasi dan nilai tertentu [7].

Terdapat tiga jenis di dalam citra biner, berupa citra biner, citra *grayscale*, dan citra RGB berikut adalah penjelasan dari jenis citra digital:

1. Citra Biner merupakan suatu citra, dimana citra tersebut memiliki 2 warna yaitu hitam dan putih (*White&Black*).
2. Citra *Grayscale* merupakan suatu citra digital, dimana citra tersebut hanya mempunyai 1 nilai kanal pada tiap piksel (*Red=Green=Blue*).
3. Citra RGB merupakan suatu citra, dimana citra tersebut memiliki 3 warna dasar. Beberapa warna dasar tersebut antara lain warna merah, warna hijau, serta warna biru. Setiap piksel mewakili kombinasi warna dari 3 warna tersebut.

Pada pengolahan citra digital terdapat 5 operasi yang sering digunakan yaitu peningkatan kualitas citra, perbaikan citra, pemampatan citra, segmentasi citra, dan analisis citra.

### 2.1.3 *Convex Hull*

*Convex hull* merupakan area yang dilingkupi oleh titik – titik yang terhubung menjadi garis dari suatu himpunan tertentu. Untuk membuat simpul dari kontur tangan, dibutuhkan proses dari *Convex hull* agar titik – titik tersebut dapat dihubungkan dan membentuk *polygon* [8]. *Convex hull* akan menghasilkan jumlah garis yang paling minimal untuk meng-*cover* semua himpunan titik. Untuk contoh hasil penggunaan *Convex Hull* pada citra bahasa isyarat tangan, dapat dilihat pada Gambar 2.1.



**Gambar 2.1** *Convex hull*

Pada Gambar 2.1 dapat didefinisikan bahwa *Convex hull* merupakan himpunan awal di titik luar *polygon* yang terbentuk dari beberapa rangkaian subset titik oleh *polygon* tersebut. Penggunaan *Convex hull* pada penelitian ini

akan didukung dengan menggunakan library Open CV untuk mencari contour dari simpul tangan.

#### **2.1.3.1 Algoritma Pencarian Kontur Tangan**

Untuk membentuk Convex Hull dari sebuah objek, diperlukan pencarian kontur pada objek tersebut terlebih dahulu. Pada kasus ini, akan diperlukan pencarian contour dari simpul tangan. Untuk mencari kontur pada simpul tangan, peneliti menggunakan library Open CV pada python. Pencarian kontur pada library Open CV menggunakan algoritma border following ,algoritma ini pertama kali diperkenalkan oleh Satoshi Suzuki dan Keiichi Abe pada jurnalnya yang berjudul “Topological Structural Analysis of Digitized Binary Images by Border Following”. Untuk mencari kontur dari suatu object dengan menggunakan library Open CV diperlukan 2 parameter penting, yaitu mode pengambilan kontur, dan metode pendekatan yang digunakan untuk mencari kontur.

Untuk mode pengambilan kontur yang digunakan pada penelitian ini adalah RETR\_TREE, cara kerja dari mode ini adalah dengan mengambil semua kontur dan merekonstruksi penuh hierarki kontur bersarang. Sedangkan untuk metode pendekatan yang digunakan adalah dengan menggunakan algoritma CHAIN\_APPROX\_SIMPLE, cara kerja dari algoritma ini adalah dengan memampatkan segmen horizontal, vertikal, dan diagonal dari suatu objek dan hanya menyisakan titik ujungnya. Misalnya, kontur persegi panjang dikodekan dengan 4 titik.

#### **2.1.3.2 Algoritma Convex Hull**

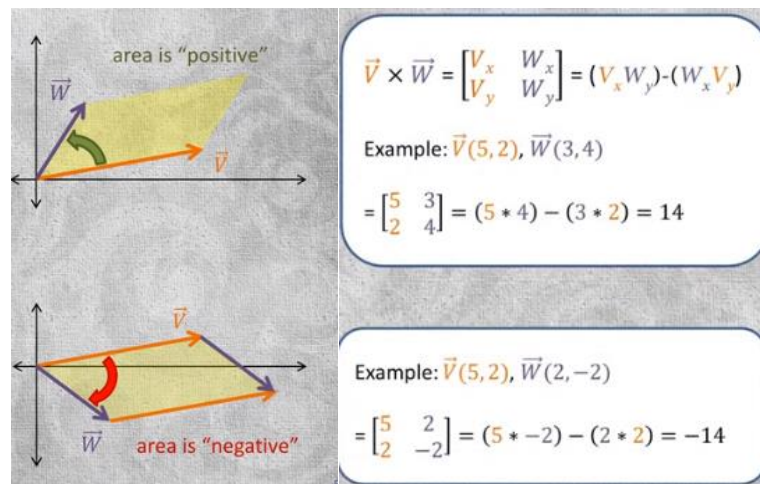
Ada berbagai macam algoritma dalam melakukan proses *Convex Hull*, pencarian algoritma Convex Hull yang efisien sudah dilakukan sejak dulu hingga sekarang. Diantara berbagai macam metode yang ditemukan, ada 2 metode yang terkenal dan dianggap efisien yaitu algoritma Graham Scan dan Jarvis March. Berikut adalah algoritma Graham Scan dan Jarvis March.

##### **1. Graham Scan**

Algoritma Graham Scan adalah algoritma pencarian *Convex Hull* yang ditemukan oleh Ronald Lewis Graham pada tahun 1972. Kelebihan dari algoritma Graham Scan adalah algoritma ini adalah yang paling optimal dalam kasus-kasus terburuk, kecepatan pencarian *Convex Hull* menggunakan metode Graham Scan

adalah yang paling populer dalam bidang Planar. Sedangkan untuk kelemahan dari algoritma Graham Scan adalah algoritma ini tidak bisa digunakan dalam ruang berdimensi diatas 2 [9]. Berikut adalah algoritma Graham Scan :

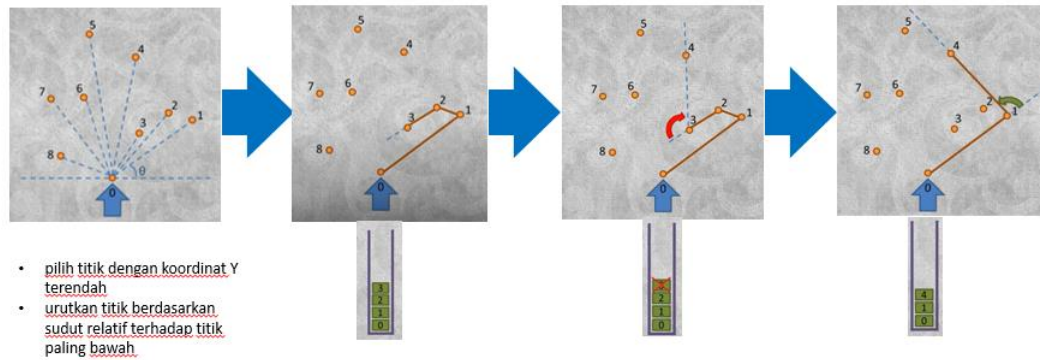
1. Titik awal pengurutan  $p_0$  dipilih dari proses pengurutan data berdasarkan koordinat y terkecil.
2. Cari besar sudut dari titik awal pengurutan  $p_0$  yang arahnya berlawanan dengan jarum jam pada tiap titik.
3. Data sudut dari tiap titik kemudian diurutkan berdasarkan sudut yang terkecil hingga yang terbesar.
4. Pada tiap titik kemudian dilakukan proses scan dan dipilih titik dengan sudut terkecil berlawanan arah jarum jam dan dimasukkan kedalam stack list, kemudian dilakukan pengujian nilai hasil kali silangnya untuk menghubungkan titik terakhir dengan titik target selanjutnya. Apabila pada proses penghubungan pada titik selanjutnya terdapat perubahan arah jarum jam (nilai hasil kali silangnya *negative*), maka titik terakhir pada *stack list* akan dihapus dan diganti dengan titik sebelumnya dan akan dilakukan proses penghubungan ulang dengan titik target. Untuk proses pengujian nilai hasil kali silangnya bisa dilihat pada Gambar 2.2.



**Gambar 2.2** Menghubungkan titik pada algoritma Graham Scan

5. Proses akan dilakukan terus menerus hingga titik terakhir dan menghasilkan kondisi dimana kembali kepada titik awal  $p_0$ .

Ilustrasi dari proses yang dilakukan pada algoritma Graham Scan yang dapat dilihat pada Gambar 2.3:



**Gambar 2.3** Algoritma Graham Scan

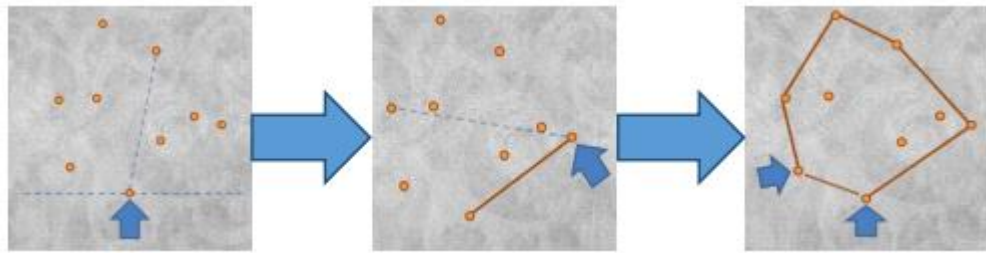
## 2. Jarvis March

Algoritma Jarvis March adalah algoritma pencarian *Convex Hull* yang ditemukan oleh R.A. Jarvis pada 1973. Algoritma ini sering dikenal dengan nama *Convex Hull Gift Wrapping Algorithm* karena cara kerja algoritma ini seperti proses pembungkusan kado [10]. Kelebihan dan kekurangan dari algoritma ini adalah pada waktu, jika jumlah titik yang harus dihubungkan sedikit, maka proses pencarian *Convex Hull* akan lebih cepat, dan sebaliknya. Berikut adalah algoritma Jarvis March.

1. Urutkan data berdasarkan koordinat y terkecil yang nantinya akan dipilih sebagai titik awal pengurutan  $1^{st}$  Vertex.
2. Setelah ditemukan titik awal, lakukan perulangan untuk mengukur sudut semua titik dengan titik awal untuk menemukan titik dengan sudut berlawanan arah jarum jam terkecil.
3. Hubungkan titik dengan titik yang memiliki sudut terkecil berlawanan arah jarum jam.
4. Lakukan berulang kali hingga kembali ke titik awal  $1^{st}$  Vertex.

Ilustrasi proses yang dilakukan pada algoritma Graham Scan dapat dilihat pada Gambar 2.4:





**Gambar 2.4** Algoritma Jarvis March

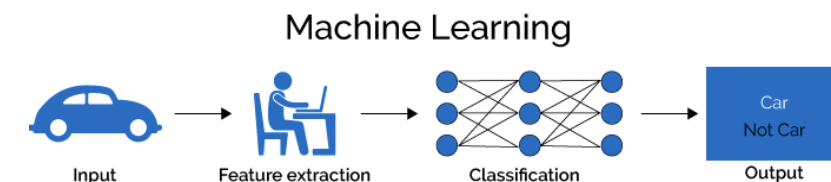
#### 2.1.4 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan yang biasa disebut *Artificial Neural Network* (ANN) ialah pendekatan dalam pengolahan informasi yang diadaptasi dari sistem syaraf [11]. *Artificial Neural Network* (ANN) memiliki salah satu komponen yang penting, yaitu *Neural Network*. Seperti namanya, *Neural Network* diadaptasi dari cara kerja *neuron* pada otak manusia. Diharapkan pada suatu saat agar sistem dapat bekerja seperti *neuron* tersebut [12].

Menurut Simon Haykin, *Artificial Neural Network* (ANN) merupakan prosesor yang berjumlah besar, dimana prosesor tersebut dipasang secara paralel. Unit pemroses dari prosesor tersebut tergolong sederhana. Hasil pembelajaran dari unit pemroses tersebut juga dapat digunakan kembali [13].

#### 2.1.5 Machine Learning

*Machine Learning* merupakan suatu pembelajaran mengenai algoritma dari komputer [14]. Algoritma *machine learning* bekerja dengan membangun model dari data sampel atau data pelatihan yang kemudian menghasilkan prediksi atau keputusan tanpa melakukan pemrograman secara terperinci [15]. Algoritma *Machine Learning* digunakan ketika algoritma konvensional tidak dapat digunakan dalam melakukan tugas – tugas seperti dalam kasus aplikasi yang rumit. Gambaran struktur permodelan jaringan pada *Machine Learning* bisa dilihat pada Gambar 2.5.



**Gambar 2.5** Struktur Pemodelan Jaringan Pada *Machine Learning*

(Sumber : <https://lawtomated.com/>)

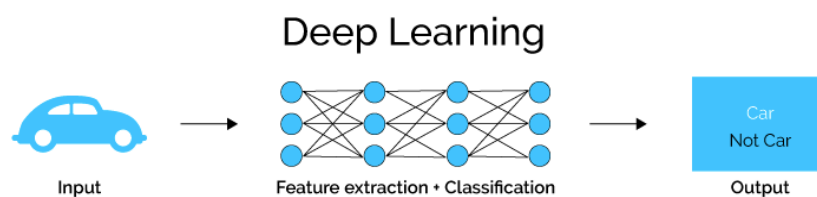
### 2.1.6 Supervised Learning

*Supervised Learning* merupakan sebuah Machine Learning yang berguna untuk pembelajaran dari fungsi pemetaan input ke output. Pemetaan tersebut dibuat dari contoh pemasangan input-output [16]. Fungsi data pelatihan berlabel digunakan dari satu set contoh pelatihan. Dimana tiap contoh yang digunakan merupakan pasangan dari objek masukan dan juga nilai keluaran yang akan diinginkan. Pasangan tersebut dapat disebut dengan isyarat pengawas (*supervisory signal*) [17].

Algoritma *supervised learning* menghasilkan fungsi simpulan untuk memetakan contoh baru. Fungsi simpulan tersebut didapatkan dengan cara menganalisis data pelatihan. Label kelas untuk *instance* yang tidak terlihat dapat ditentukan dengan *scenario* yang maksimal dari algoritma *supervised learning*.

### 2.1.7 Deep Learning

*Deep learning* ialah jaringan syaraf tiruan untuk mempresentasikan jaringan pembelajarannya, dimana masih satu bagian dari *machine learning*. Menurut Goodfellow, dkk. (2016), *deep learning* merupakan sistem pembelajaran *computer* dengan konsep hierarki dengan menggunakan sebuah pendekatan dalam penyelesaian masalahnya. Konsep hierarki bekerja dengan menggabungkan beberapa konsep secara lebih sederhana dan efisien sehingga dapat membuat konsep tersebut dapat dipelajari dengan lebih kompleks oleh komputer. Dapat disebut dengan pembelajaran *deep learning* karena, grafik dari beberapa konsep yang dibangun tersebut memiliki banyak layer [18]. Gambaran dari struktur permodelan jaringan pada *Deep Learning* bisa dilihat pada Gambar 2.6.



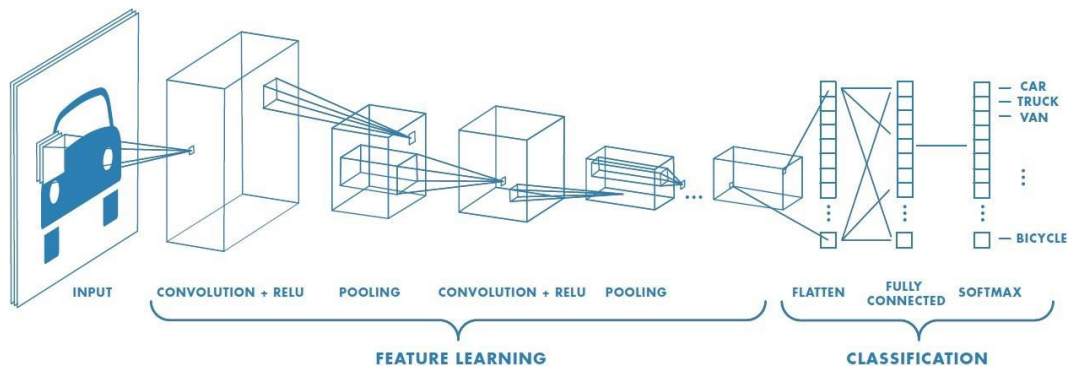
**Gambar 2.6** Struktur Pemodelan Jaringan Pada *Deep Learning*

(Sumber : <https://lawtomated.com/>)

### 2.1.8 Convolutional Neural Network

*Convolutional Neural Network* merupakan pengolahan citra dari data dua dimensi yang dikembangkan berdasarkan *multi layer perceptron* (MLP). CNN

dapat dikatakan sebagai jenis dari *neural network*. Alasan yang dapat mendukung karena, CNN memiliki kedalaman jaringan tinggi, selain itu juga memiliki jumlah yang besar dalam pengaplikasian pada data citra. Metode MLP kurang sesuai dalam klasifikasi citra karena menghasilkan hasil yang kurang baik. Hal tersebut karena, informasi special dari data citra tidak disimpan oleh MLP. Selain itu, dalam MLP, tiap piksel dianggap sebagai fitur yang independen. Arsitektur dalam CNN dapat dilatih dari beberapa tahap. *Feature map* merupakan kumpulan dari beberapa *array* dari tiap tahap masukan (*input*) dan keluaran (*output*). Tiap tahap dari CNN disusun dari 3 *layer*. *Layer* tersebut antara lain adalah *layer* konvolusi, *layer* fungsi aktivasi, dan *layer pooling* [19]. Gambaran arsitektur CNN bisa dilihat pada Gambar 2.7.



**Gambar 2.7** Arsitektur *Convolutional Neural Network*

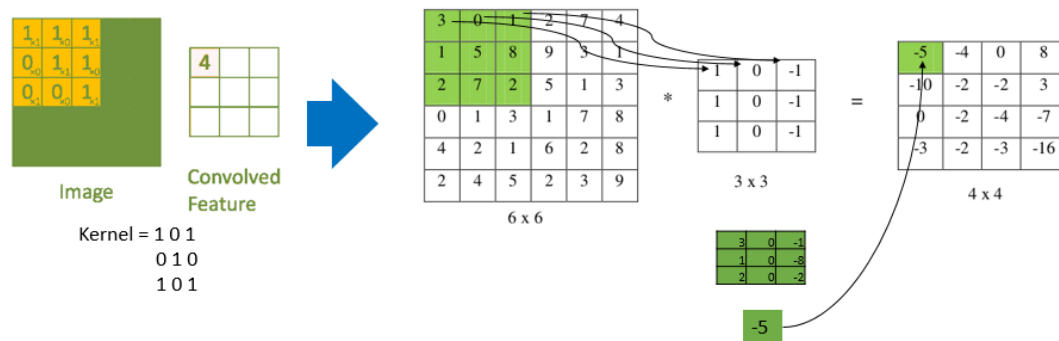
(Sumber : <https://towardsdatascience.com/>)

Konvolusi merupakan tahap pertama dari arsitektur CNN yang bekerja dengan menggunakan kernel, dimana penggunaan kernel tersebut tergantung dari jumlah fitur yang telah dihasilkan. Tahap kedua adalah tahap dari fungsi aktivasi. Setelah melalui tahap dari fungsi aktivasi, kemudian dilanjutkan ke proses *pooling*, dimana proses ini merupakan proses dari tahap ketiga. Proses dilakukan hingga mendapatkan *feature map* yang bernilai cukup agar dapat melanjutkan ke *fully connected neural network*. Setelah serangkaian proses tersebut, *output class* baru dapat dihasilkan.

#### 2.1.8.1 Layer Konvolusi

Proses *convolutional layer* digunakan untuk melakukan ekstraksi ciri dari tiap citra. Proses ini memiliki beberapa komponen, antara lain *stride*, *filter*, serta *zero padding*. *Filter* adalah suatu matriks hasil yang terpotong dan diubah berupa

ukuran yang lebih kecil dari sebelumnya. Dengan ukurannya adalah  $n \times n$  piksel, dimana tiap *filter* akan diberikan bobot yang acak. Dalam *filter* terdapat proses perkalian dari bobot dengan matriks hasil tiap piksel dan kemudian dijumlahkan agar mendapatkan fitur hasil. Rangkaian proses tersebut disebut dengan operasi dot [20]. *Stride* dapat dikatakan sebagai pergeseran jalannya *filter*. Pergeseran tersebut dapat berupa satu buah piksel menuju arah kanan hingga tepi citra, setelah itu bergeser lagi satu buah piksel menuju arah bawah. Kemudian, *zero padding* ialah suatu proses penambahan dari piksel. Dimana penambahan tersebut adalah satu piksel pada seluruh dari tepi milik citra yang menambahkan sebuah angka berupa nol pada tiap tepi tersebut. Gambaran proses yang terjadi pada *layer* konvolusi dapat dilihat pada Gambar 2.8.



**Gambar 2.8** Layer Konvolusi

Setelah melalui *layer* konvolusi dan menghasilkan *matrix Convolved Feature* selanjutnya dilakukan fungsi aktivasi ReLU, dimana merupakan kepanjangan dari *Rectified Linear Unit* (ReLU) yang melakukan penerapan dengan menetapkan nilai minimal = 0. Sehingga jika terdapat nilai *negative* pada *matrix Convolved Feature*, nilai *negative* tersebut akan diubah menjadi 0.

### 2.1.8.2 Layer Pooling

*Pooling* atau *subsampling* digunakan dalam tahapan pengurangan dari ukuran matriks. Pengurangan didapatkan dari operasi *pooling*. *Average pooling* dan *max-pooling* merupakan macam dari *pooling* yang telah sering digunakan. Dalam *average pooling* tersebut menggunakan nilai yang rata-rata, dan *max-pooling* menggunakan nilai yang maksimal [21]. Proses konvolusi serta pooling dilakukan hingga mendapatkan *feature map* yang bernilai cukup agar dapat melanjutkan ke *fully connected neural network*.

#### 2.1.8.3 Layer Terkoneksi Penuh

*Feature map* yang berisi *input* dari *output pooling layer* diambil kedalam *layer* terkoneksi penuh atau dapat disebut dengan *fully connected map*. Dimana bentuk dari *feature map* yang diambil adalah berupa *multidimensional array*. Agar dapat menghasilkan *vector* sebanyak *n*-dimensi, maka *array* tersebut akan dilakukan proses *reshape feature map*. Nilai *n* dari dimensi yang dimaksud adalah banyaknya pilihan jumlah *output class* oleh program. Untuk penggambarannya dapat dicontohkan, terdapat 500 *neuron* yang membentuk sebuah lapisan, *softmax* akan diterapkan untuk mengembalikan daftar probabilitas maksimum pada tiap *label class*. Dimana label tersebut digunakan sebagai klasifikasi akhir dari suatu jaringan [22].

#### 2.1.8.4 Layer Softmax

Dalam hal melakukan metode klasifikasi kelas yang berjumlah banyak, perlu digunakan fungsi *softmax*. Kelas yang termasuk disini meliputi analisis dikriminasi *linear multiclass*, *regresi logistic multinomial*, *Artificial Neural Network* (ANN), dan *Naïve Bayes Classifier*.

Suatu fungsi yang bernama *Softmax* ialah fungsi yang bekerja mengubah *vector* “x” dari sebuah *K*-dimensi menjadi *vector* dalam rentang nilai 0-1 yang berjumlah satu, namun tetap memiliki bentuk yang sama. Ketika terdapat *neural network* dalam sebuah *layer* dan pada *layer* akhir tersebut mendapatkan *output*, maka fungsi *softmax* dapat digunakan. *softmax neuron* bekerja dengan cara menerima *input*. Setelah *input* diterima, kemudian akan melakukan proses pembobotan serta penambahan bias. Setelah proses tersebut, *neuron softmax layer* akan menerapkan fungsi *softmax* dan bukan fungsi aktivasi. Probabilitas terbesar untuk hasil kelas disini dapat ditentukan dengan menggunakan *softmax layer* [23].

#### 2.1.9 OpenCV

OpenCV ialah suatu *library* yang melakukan pengolahan citra yang berfokus pada gambar yang akan diekstrak untuk menemukan suatu informasi dari citra gambar tersebut. Untuk melakukan pengolahan tersebut, perlu menerapkan implementasi dari algoritma beserta sintaks. Didalam *library* OpenCV ini, selain terdapat algoritma untuk penentuan warna, juga terdapat algoritma untuk menentukan lokasi objek.

#### 2.1.10 Tensoflow

Terdapat sebuah *software library* dengan sumber terbuka dan gratis untuk melakukan pemrosesan Machine Learning, yaitu TensorFlow. TensorFlow digunakan untuk berfokus pada pelatihan serta inferensi *Deep Learning*. Namun, tidak menutup kemungkinan, TensorFlow juga dapat digunakan untuk berbagai kasus atau tugas yang lain. TensorFlow memiliki bentuk arsitektur yang fleksibel, dengan maksud agar dapat diterapkan pada komputasi yang mudah untuk berbagai *platform*, seperti CPU, GPU, dan TPU.

#### 2.1.11 Keras

Dalam *High-level-API* terdapat salah satu *framework* bernama Keras. *Framework* ini berjalan dengan menggunakan platform Python yang digunakan untuk membantu dalam proses pembelajaran terhadap *computer* dengan mudah. Python disini mendukung proses komputasi cerdas yang diantaranya adalah jaringan konvolusi. Disini, keras berguna sebagai *interface* dari sebuah library TensorFlow. Dalam suatu kasus dimana perlu adanya pembuatan *prototype* yang cepat, mudah, dan *user friendly*, maka dapat dilakukan oleh Keras. Keras juga mendukung komputasi pada *recurrent neural network*, *convolutional neural network*, dan kombinasi dari keduanya juga. Keras juga dapat bekerja pada CPU dan GPU dengan kompitibel Python adalah versi 2.7 hingga 3.6 [24].

### 2.2 Penelitian Terkait

Penelitian dilakukan dengan menggunakan beberapa referensi yang pernah dilakukan oleh peneliti terdahulu. Dimana, penelitian tersebut bersifat relevan agar dapat menjadi bahan pertimbangan dalam hal yang baik mengenai mengenai kekurangan dan kelebihan, bertujuan dalam memperoleh satu atau lebih informasi, dimana informasi tersebut dengan judul dari penelitian tersebut adalah sesuai, serta mendapatkan landasan dari teori secara ilmiah. Adapun beberapa referensi sebagai berikut:

1. Menurut Mochamad Bagus Setiyo Bakti, Yuliana Melita Pranoto dalam penelitian berjudul “Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode *Convolutional Neural Network*” pada tahun 2019. Pengenalan isyarat angka SIBI dilakukan dalam penelitian tersebut. Penelitian tersebut menggunakan metode bernama *Convolutional Neural*

*Network* (CNN) dengan arsitekturnya menggunakan LeNet. Terdapat tiga tahap dalam proses arsitektur CNN. Tahapan tersebut antara lain, 25 *epoch*, 50 *epoch*, dan 100 *epoch*. Berdasarkan percobaan penelitian yang telah dilakukan, didapatkan bahwa nilai akurasi di tiap tersebut tahap terus meningkat. Nilai akurasi tersebut meningkat mulai dari 67.66% menuju 89.44%, hingga 94.44% sebagai nilai akurasi yang tertinggi dalam proses training. Peningkatan nilai tersebut juga terjadi dalam proses prediksi data tiap tahapnya. Kenaikan tersebut mulai dari 79.23% menuju 90.45%, hingga 98.89% sebagai nilai akurasi yang tertinggi dalam prediksi data tersebut [4].

2. Menurut Devina Yolanda, Kartika Gunadi, Endang Setyati dalam penelitian berjudul “Pengenalan Alfabet Bahasa Isyarat Tangan Secara *Real-Time* dengan Menggunakan Metode *Convolutional Neural Network* dan *Recurrent Neural Network*”. Pengenalan *alphabet* dalam penelitian tersebut menggunakan metode yang bernama *Convolutional Neural Network* (CNN) sebagai *feature extraction* pada *spatial feature* serta *Recurrent Neural Network* (RNN) sebagai pengkoderan antar *frame* yang telah diekstrak oleh CNN pada *temporal feature*. Dimana *input-an* dari penelitian tersebut adalah berupa *video* dan output yang ditampilkan merupakan *text alphabet* dari pengenalan *alphabet* bahasa isyarat yang telah diolah tersebut. Berdasarkan hasil percobaan penelitian yang telah dilakukan tersebut, nilai akurasi rata-rata yang didapatkan untuk seluruh huruf adalah sebesar 60.58%. sedangkan, pada pengujian *real-time* dalam penelitian tersebut, mengalami kegagalan. Kegagalan tersebut disebabkan karena teknologi yang digunakan tidak dapat menunjang arsitektur yang sedang dibuat [25].
3. Menurut Amiraj Dhawan, Vipul Honrao dalam penelitian yang berjudul “*Implementation of Hand Detection based Techniques for Human Computer Interaction*”. Proses deteksi tangan pengguna yang digunakan sebagai *input-an* pada interaksi aplikasi *computer* telah dibahas dalam penelitian ini. Sistem yang digunakan terdiri dari berbagai teknik interaksi dari manusia serta komputer yang efisien. Penelitian ini menggunakan *preprocessing* gambar *input-an*. Penggunaan *preprocessing* berupa gambar inputan ini digunakan

agar sistem bekerja secara akurat serta efisien pada beberapa batasan yang ada [26].

4. Menurut Caroline El Fiorenza, Sandeep Kumar Barik, Ankit Prajapati, Sagar Mahesh dalam penelitian yang berjudul “*Hand Gesture Recognition using Convexity Defect*”. Foto gerakan satu-per-satu dari tangan digunakan dalam penelitian ini. Penelitian dilakukan dengan melakukan pengukuran pusat dengan sudut penjumlahan. Kemudian agar mendapatkan *manual contour*, *convexity defects*, dan *convex shaft* nya perlu melakukan proses pengukuran sudut penjumlahan dengan pusat tersebut [27].
5. Menurut Kumud Alok, Anurag Mehra, Ankita Kaushik, Aman Verma dalam penelitian yang berjudul “*Hand Sign Recognition using Convolutional Neural Network*”. Metode bernama *Convolutional Neural Network* (CNN) dilakukan untuk melakukan penelitian terhadap pengenalan isyarat tangan dengan nilai akurasi validasi sebesar 74.80%. Sedangkan, pada penelitian ini memiliki nilai akurasi pelatihan sebesar 91.90%. berdasarkan nilai akurasi validasi yang rendah, serta nilai akurasi pelatihan yang tinggi, maka generalisasi data yang diberikan dari model tidak berjalan dengan baik. Selain itu pada model yang dihasilkan, juga terdapat adanya beberapa tingkat *overfitting* yang mungkin terjadi [5].



## **BAB III METODE PENELITIAN**

### **3.1 Jenis Penelitian**

Metode sistematis berupa *experimental reaserch* akan dimanfaatkan untuk penelitian. *Experimental reaserch* berguna untuk membuat sebuah relasi yang didalamnya terdapat fenomena sebab dan akibat. Pendekatan kuantitatif digunakan dalam model penelitian ini, dimana inti dari metode tersebut merupakan penelitian eksperimental. Penelitian ini bertujuan untuk merancang sebuah pengenalan alfabet bahasa isyarat menggunakan pendekatan *Convex Hull* dan diimplementasikan ke jaringan syaraf tiruan untuk dilakukan pembelajaran menggunakan bantuan *machine learning* yang menggunakan *convolutional neural network* sebagai struktur *deep learning*-nya.

### **3.2 Langkah-langkah Penelitian**

Penelitian ini memiliki tahapan dengan tujuan untuk mencapai hasil yang diinginkan, berikut tahapan-tahapannya :

#### **1. Studi literatur**

Untuk mempelajari dasar-dasar teori yang berkaitan dengan “Pengenalan Alfabet Bahasa Isyarat Tangan Pada Citra Digital Menggunakan Pendekatan *Convexity Hull* Dan *Convolutional Neural Network (CNN)*”, dilaukan studi literatur dengan cara mempelajari berbagai macam sumber refrensi.

#### **2. Analisis masalah**

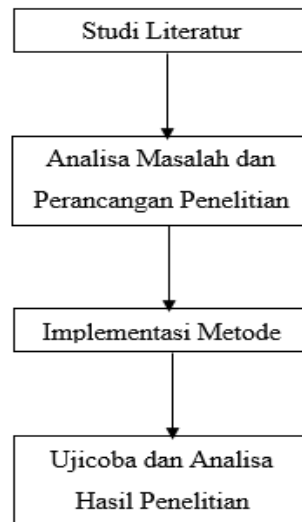
Penelitian ini memiliki masalah yang akan diselesaikan, yaitu “Pengenalan Alfabet Bahasa Isyarat Tangan Pada Citra Digital Menggunakan Pendekatan *Convexi Hull* Dan *Convolutional Neural Network (CNN)*”. Pengolahan dataset citra, proses pre-processing, dan pengukuran tingkat akurasi merupakan metode yang akan dibangun dalam penelitian ini untuk menyelesaikan masalah tersebut.

#### **3. Implementasi metode**

Tahapan selanjutnya adalah implementasi metode, implementasi metode bertujuan untuk menerapkan rancangan metode yang telah dibangun. Pustaka opencv dan tensorflow serta framework keras akan digunakan untuk menerapkan rancangan metode tersebut.

#### **4. Uji coba dan analisis hasil penelitian**

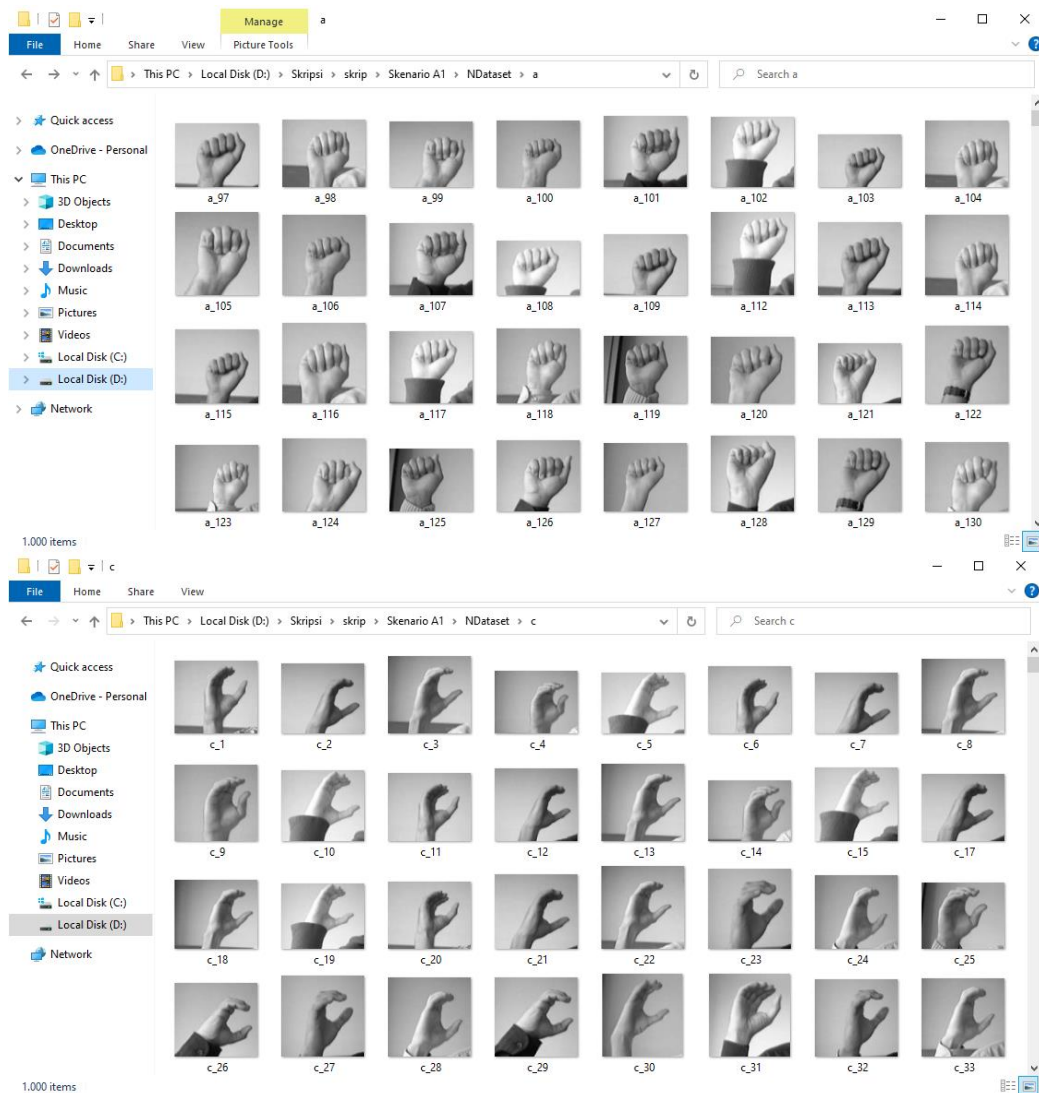
Tahapan selanjutnya adalah uji coba, kemudian dilanjutkan dengan analisis hasil penelitian. Untuk mengetahui kinerja dari metode yang diusulkan, perlu dilakukan ujicoba yang berhubungan dengan sistem. Proses pengujian dikerjakan menggunakan rancangan skenario uji coba. Tahapan dalam metode penelitian dapat lebih mudah dipahami dengan menggunakan diagram alir seperti pada Gambar 3.1.



**Gambar 3.1** Diagram Alir Tahapan Penelitian

### 3.3 Dataset

Penelitian ini memiliki dataset berupa “Sign Language For Alphabets” yang diambil dari Kaggle.com yang dapat diunduh pada <https://www.kaggle.com/muhammadrkhalid/sign-language-for-alphabets>. Dataset terdiri dari 27 *class*, dan 1500 gambar pada tiap *class*. Pada penelitian ini, peneliti hanya menggunakan 26 *class* dan 1000 gambar pada tiap *class*, sehingga total data yang digunakan adalah 26000. Hal ini karena ada beberapa gambar yang memiliki kontras gambar yang terlalu buruk, dan juga terdapat gambar yang memiliki latar belakang yang terlalu ramai, sehingga setelah dilakukan screening data, peneliti memutuskan untuk hanya menggunakan 1000 data pada tiap *class*. Contoh data yang digunakan untuk beberapa *class* dapat dilihat pada Gambar 3.2.

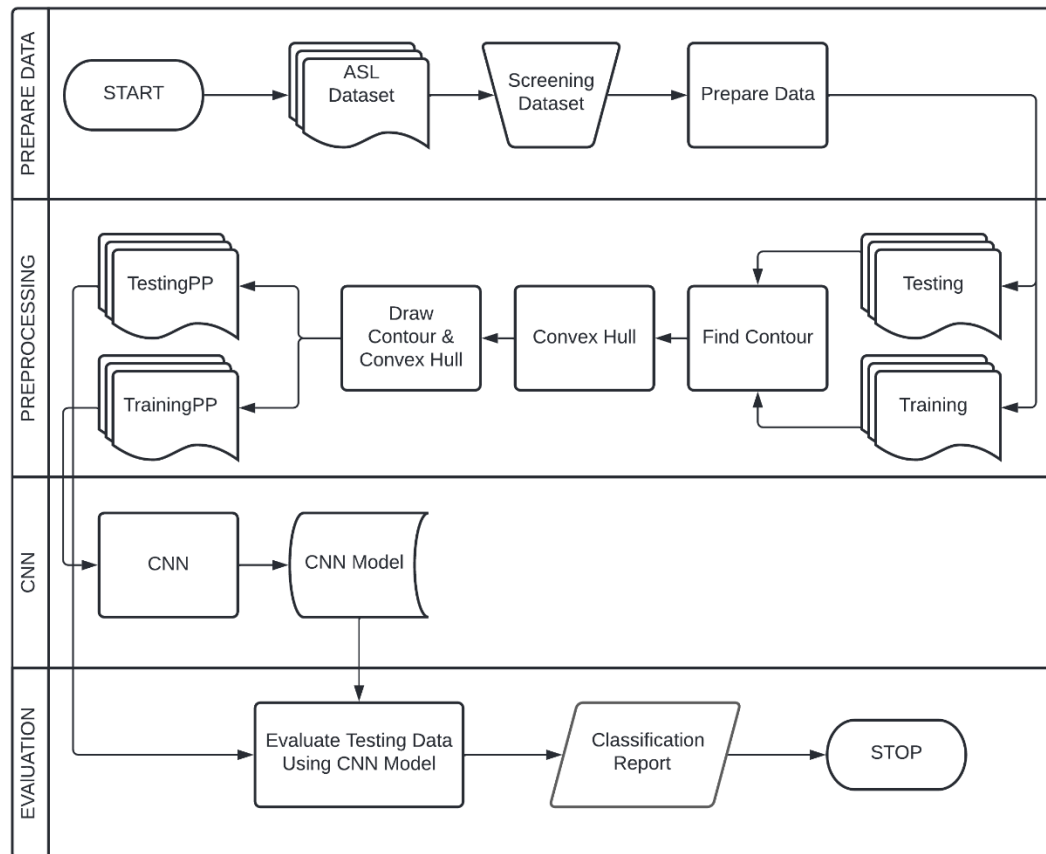


**Gambar 3.2 Dataset**

### 3.4 Arsitektur Sistem

Arsitektur sistem ialah rangkaian langkah yang harus dilakukan saat pembuatan sistem. Agar sistem pengenalan alphabet bahasa isyarat dapat terdesain dengan baik. Dataset yang berisi kumpulan gambar Bahasa isyarat tangan dilakukan *screening* untuk mengurangi data dengan kontras yang buruk serta background yang terlalu ramai akan objek, kemudian data dibagi menjadi data *Training* dan *Testing*. Selanjutnya dilakukan *pre-processing* untuk data *Training* dan *Testing* dengan menggunakan *Convex Hull*, setelah melalui tahap *pre-processing* dengan *Convex Hull* kemudian data *Training* hasil preprocessing akan digunakan sebagai data masukan pada proses CNN (Convolutional Neural Network), pembagian data *Training* dan *Validation* akan dibagi sebesar 80:20 ,

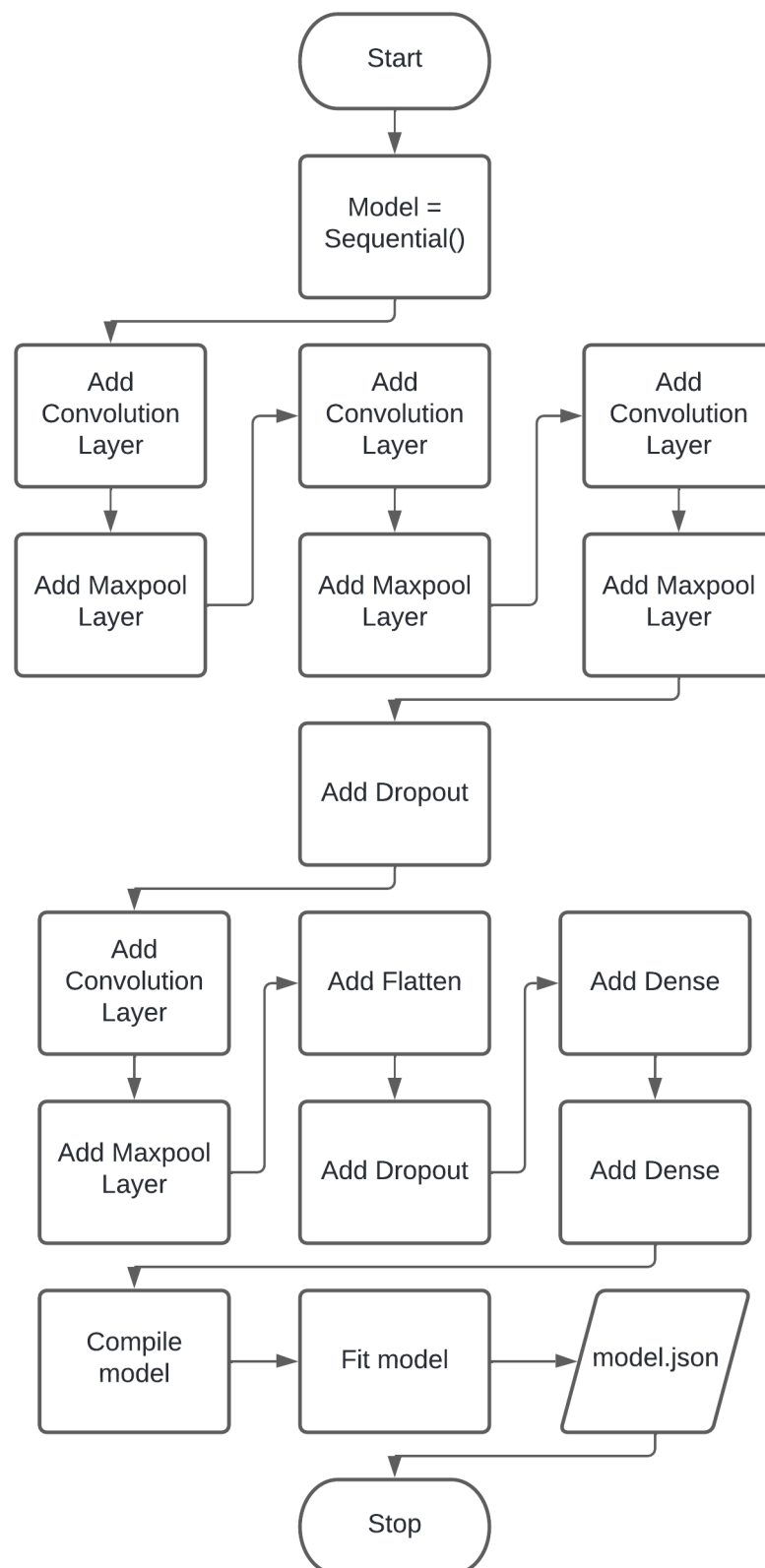
80% berupa data *Training* dan 20% berupa data *Validation*. Setelah melewati proses CNN maka akan menghasilkan file model, dimana file model tersebut digunakan sebagai model bobot untuk mengevaluasi data testing, masukan gambar tersebut kemudian diprediksi menggunakan model yang telah dihasilkan oleh proses CNN. Gambaran bagan alir rancangan sistem dapat dilihat pada Gambar 3.3.



**Gambar 3.3** Bagan Alir Rancangan Sistem

Dalam penelitian ini, struktur CNN akan dibangun oleh framework Keras dengan TensorFlow sebagai backend yang berisi, “input -> conv1 (filter = 32, kernel size = 3, strides = 1, activation function = “ReLU”) -> Maxpool (pool size = 3x3, strides = 2) -> conv2 (filter = 64, kernel size = 3, strides = 1, activation function = “ReLU”) -> Maxpool (pool size = 2x2, strides = 2) -> conv3 (filter = 64, kernel size = 3, strides = 1, activation function = “ReLU”) -> Maxpool (pool size = 2x2, strides = 2) -> dropout -> conv4 (filter = 128, kernel size = 3, strides = 1, activation function = “ReLU”) -> Maxpool (pool size = 2x2, strides = 1) -> flatten -> dropout -> fully connected -> out” , terdapat empat lapisan

konvolusional dan satu lapisan yang fully connected . Fungsi aktivasi *layer Convolution* menggunakan “ReLU” sedangkan fungsi aktivasi lapisan *fully connected* menggunakan “softmax”. *Softmax* ini akan menentukan probabilitas terbesar untuk hasil kelasnya. Gambaran struktur CNN yang digunakan bisa dilihat pada Gambar 3.4.



**Gambar 3.4** Struktur CNN

### 3.5 Metode Evaluasi

Setelah sistem dibuat dan dilakukan uji coba terhadap sistem tersebut, maka selanjutnya yang perlu dilakukan adalah evaluasi. Evaluasi merupakan proses pengujian kinerja algoritma pengenalan yang digunakan untuk menampilkan tingkat akurasi pada setiap kategorinya. Evaluasi ini menggunakan *confusion matrix*. Untuk mencatat atau mengukur hasil kerja klasifikator digunakan metode *confusion matrix* yang merupakan tabel yang umum digunakan untuk melakukan proses tersebut [28]. Pada Tabel 3.1 merupakan tabel *confusion matrix* yang menunjukkan pengukuran hasil kerja pengenalan yang menghasilkan nilai akurasi. Akurasi merupakan sebuah proses untuk mengukur tingkat keakuratan suatu proses pengujian dengan cara mengidentifikasi jumlah dari data yang diklasifikasi secara benar [29].

|           |         | Data Prediksi |         |
|-----------|---------|---------------|---------|
|           |         | Positif       | Negatif |
| Data Asli | Positif | TP            | FN      |
|           | Negatif | FP            | TN      |

**Tabel 3.1. Hasil Prediksi *Confusion Matrix***

Keterangan :

True Positif (TP) = Data positif yang terklasifikasi dengan benar oleh sistem.

True Negatif (TN) = Data negatif yang terklasifikasi dengan benar oleh sistem.

False Positif (FP) = Data positif yang terklasifikasi salah oleh sistem

False Negatif (FN) = Data negatif yang terklasifikasi salah oleh sistem.

Akurasi merupakan sebuah proses untuk mengukur tingkat keakuratan suatu proses pengujian dengan cara mengidentifikasi jumlah dari klasifikasi data berupa benar [29]. Sehingga, nilai akurasi dapat disebut dengan persentase hasil perbandingan data terklasifikasi benar dengan keseluruhan data. Persamaan 3.1 dibawah ini adaah rumus untuk menghitung akurasi.

Keterangan :

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (3.1)$$

Keterangan :

True Positif (TP) = Data positif yang terklasifikasi dengan benar oleh sistem.

True Negatif (TN) = Data negatif yang terklasifikasi dengan benar oleh sistem.

False Positif (FP) = Data positif yang terklasifikasi salah oleh sistem

False Negatif (FN) = Data negatif yang terklasifikasi salah oleh sistem.

Nilai presisi merupakan persentase hasil dari jumlah data positif, dimana data positif tersebut terklasifikasikan secara benar dan kemudian dibagi dengan total data yang juga terklasifikasi positif. Persamaan 3.2 dibawah ini adaah rumus untuk menghitung presisi.

$$Presisi = \frac{TP}{FP+TP} * 100\% \quad (3.2)$$

Keterangan :

True Positif (TP) = Data positif yang terklasifikasi dengan benar oleh sistem.

True Negatif (TN) = Data negatif yang terklasifikasi dengan benar oleh sistem.

False Positif (FP) = Data positif yang terklasifikasi salah oleh sistem

False Negatif (FN) = Data negatif yang terklasifikasi salah oleh sistem.

Sementara itu, recall merupakan persentase hasil dari data berkategori nilai positif yang telah terklasifikasikan dengan nilai benar oleh sistem yang kemudian dibandingkan bersama keseluruhan data dari kategori bernilai positif terklasifikasi dengan nilai benar. Persamaan 3.1 dibawah ini adaah rumus untuk menghitung recall.

$$Recall = \frac{TP}{FN+TP} * 100\% \quad (3.3)$$

Keterangan :

True Positif (TP) = Data positif yang terklasifikasi dengan benar oleh sistem.

True Negatif (TN) = Data negatif yang terklasifikasi dengan benar oleh sistem.

False Positif (FP) = Data positif yang terklasifikasi salah oleh sistem

False Negatif (FN) = Data negatif yang terklasifikasi salah oleh sistem.



Rancangan tabel *confusion matrix* yang akan digunakan untuk menentukan jumlah tingkat kebenaran pada setiap kategori alphabet isyarat tangan dapat dilihat pada Table 3.2.

### 3.6 Skenario Ujicoba

Skenario Ujicoba dari sistem dikerjakan dengan perbandingan tingkat dari akurasi antara sistem, dimana sistem tersebut pernah melalui proses *Convex Hull* lalu kemudian baru melakukan proses pembelajaran dengan CNN dengan sistem yang dari citra aslinya langsung menuju proses pembelajaran dengan CNN. Untuk proses pembandingannya akan dilakukan dengan 3 variasi pembagian data *training* dan *testing* untuk masing-masing sistem, untuk pembagian data *training* dan *validation*nya adalah 80:20. Untuk tiap ujicoba jumlah dan pembagian datanya akan sama untuk tiap sistem, yang berbeda hanya input dan metodenya saja. Tabel skenario ujicoba dapat dilihat pada Tabel 3.3 :

| No | Label Skenario | Deskripsi                                                                                                                                                                                                                 |
|----|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | SKENARIO 1     | Perbandingan sistem yang menggunakan <i>Convex Hull</i> dan CNN dan sistem yang tanpa menggunakan <i>Convex Hull</i> atau CNN saja dengan pembagian data 60:40 dan epoch sebesar 20.                                      |
| 2  | SKENARIO 2     | Perbandingan sistem yang menggunakan <i>Convex Hull</i> dan CNN dan sistem yang tanpa menggunakan <i>Convex Hull</i> atau CNN saja dengan pembagian data <i>Train</i> dan <i>Test</i> sebesar 70:30 dan epoch sebesar 20. |
| 3  | SKENARIO 3     | Perbandingan sistem yang menggunakan <i>Convex Hull</i> dan CNN dan sistem yang tanpa menggunakan <i>Convex Hull</i> atau CNN saja dengan pembagian data <i>Train</i> dan <i>Test</i> sebesar 80:20 dan epoch sebesar 20. |
| 4  | SKENARIO 4     | Setelah dilakuan 3 skenario diatas, kemudian skenario dengan pembagian data yang memberikan akurasi tertinggi akan dibandingkan                                                                                           |

|  |  |                                                                                                                                                                                                                                                                                         |
|--|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |  | <p>lagi sebanyak 5 kali dengan melakukan variasi epoch untuk tiap ujicoba yang dilakukan, untuk variasi epochnya adalah 20, 40, 60, 80, dan 100. Kemudian setelah dilakukan ujicoba sebanyak 5 kali, maka akan dilakukan perhitungan nilai rata-rata akurasi dari 5 kali percobaan.</p> |
|--|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Tabel 3.3. Rancangan Tabel Skenario Ujicoba**

Setelah dilakukan ujicoba dengan skenario seperti pada tabel diatas, bisa disimpulkan bisa atau tidaknya sistem pengenalan bahasa isyarat tangan dengan menggunakan metode *Convex Hull* dan CNN meningkatkan akurasi daripada pengenalan bahasa isyarat tangan yang menggunakan CNN saja.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Lingkungan Ujicoba

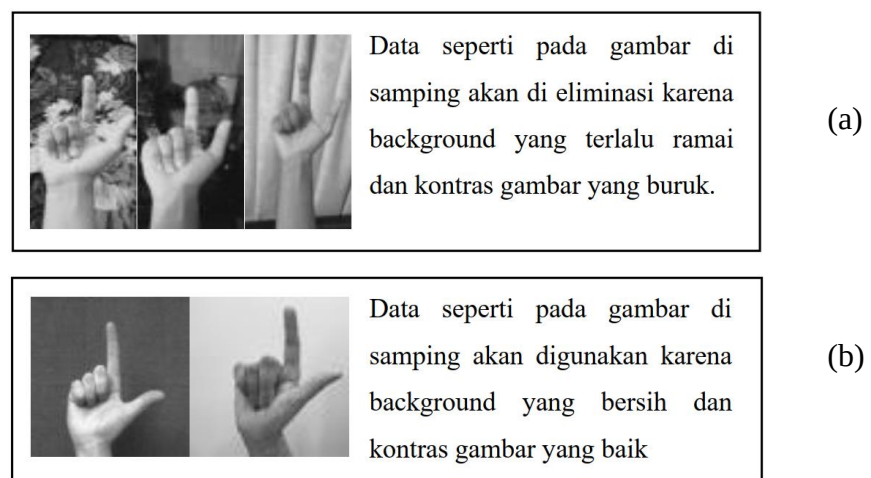
Pada penelitian ini terdapat *Hardware* dan *Software* yang digunakan, dapat dilihat pada Tabel 4.1.

| No | Kebutuhan            | Jenis Perangkat                                                                                                                                                                    |
|----|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Sistem Operasi       | Windows 10                                                                                                                                                                         |
| 2  | Browser              | Google Chrome                                                                                                                                                                      |
| 3  | Platform             | Google Colaboratory PRO                                                                                                                                                            |
| 4  | Runtime              | Python 3 Google Compute Engine Backend (GPU)                                                                                                                                       |
| 5  | RAM                  | 25.46 GB                                                                                                                                                                           |
| 6  | Hardware Accelerator | GPU                                                                                                                                                                                |
| 7  | Runtime Shape        | High RAM                                                                                                                                                                           |
| 8  | Disk                 | 166.83 GB                                                                                                                                                                          |
| 9  | Bahasa Pemrograman   | Python                                                                                                                                                                             |
| 10 | Library              | <i>Tensorflow</i><br><i>Keras</i><br><i>Numpy</i><br><i>Matplotlib</i><br><i>OpenCV</i><br><i>Os</i><br><i>Shutil</i><br><i>Sklearn.metrics</i><br><i>Seaborn</i><br><i>Pandas</i> |
| 11 | Storage              | Google Drive                                                                                                                                                                       |

**Tabel 4.1. Perangkat Yang Digunakan**

## 4.2 Screening Data

Sebelum data digunakan, terlebih dahulu akan dilakukan *screening* data agar mendapatkan hasil yang baik serta untuk mengurangi data-data yang kurang layak, seperti background data yang terlalu ramai dan juga kontras cahaya yang terlalu buruk. Karena jika background dari data terlalu ramai dan kontras cahaya terlalu buruk, akan sulit untuk melakukan pengolahan citra pada data tersebut terutama saat pengambilan kontur. Untuk itu perlu dilakukan *screening* data terlebih dahulu. Contoh *screening* yang dilakukan pada class “L” dapat dilihat pada Gambar 4.1.



**Gambar 4.1** (a) dan (b) *Screening Data*

Setelah dilakukan *screening* data, jumlah data pada tiap class berkurang, jadi data yang awalnya terdiri dari 1500 data untuk tiap class dan 39000 jumlah total data keseluruhan, berkurang menjadi 1000 data untuk tiap class A-Z dan 26000 jumlah total data keseluruhan.

## 4.3 Pembuatan Program

Setelah sebelumnya data di *screening* maka data sudah siap digunakan untuk penelitian. Berikut adalah tahapan pembuatan program dan ujicoba.

### 4.3.1 Prepare Data

Pada tahap ini, akan dilakukan pembagian data *Training* dan data *Testing*. Pembagian data dilakukan dengan melalui program. Berikut ini adalah Pseudocode dari program *prepare data*.

```
1. SET dirtrain TO "Training"
```

```

2. if(dirtrain not IN CURRENT DIRECTORY):
3.     MAKE DIRECTORY dirtrain
4. SET dirtest TO "Testing"
5. if(dirtest not IN CURRENT DIRECTORY):
6.     MAKE DIRECTORY dirtest
7. SET character TO [A-Z]
8. FOR x IN range(len(character)):
9.     SET cdir TO character[x]
10.    SET traindir TO PATH JOIN(dirtrain,cdir)
11.    if(cdir not IN LIST DIRECTORY INSIDE dirtrain DIRECTORY):
12.        MAKE DIRECTORY traindir
13.    SET testdir TO PATH JOIN(dirtest,cdir)
14.    if(cdir not IN LIST DIRECTORY INSIDE dirtest DIRECTORY):
15.        MAKE DIRECTORY testdir
16. SET totdata TO 1000
17. SET ptrain TO 60
18. SET ptest TO 40
19. trdata=int((totdata*ptrain)/100)
20. tsdata=int((totdata*ptest)/100)
21. SET datadir TO "/content/drive/MyDrive/NDataset"
22. SET datalist TO LIST DIRECTORY INSIDE("/content/drive/MyDrive/NDataset")
23. FOR x IN datalist:
24.     SET fdir TO PATH JOIN(datadir,x)
25.     SET d TO len(LIST DIRECTORY OR FILE INSIDE(fdir))
26.     OUTPUT("jumlah data : ",d,"")
27.     OUTPUT("copying ",x,"from ",fdir," FOR Training data")
28.     FOR z IN range(1,trdata+1):
29.         COPY FILE FROM DATASET FOLDER TO TRAINING FOLDER
30. FOR x IN datalist:
31.     SET fdir TO PATH JOIN(datadir,x)
32.     OUTPUT("copying ",x,"from ",fdir," FOR testing data")
33.     FOR z IN range(trdata+1,totdata+1):
34.         COPY FILE FROM DATASET FOLDER TO TESTING FOLDER

```

#### Pseudocode Program 4.1 Prepare Data

Untuk lebih jelasnya kode program untuk pembagian data beserta penjelasannya dapat dilihat pada Kode Program 4.1. Berikut adalah penjelasan Kode Program 4.1 :

1. Membuat direktori *training* dan *testing*.
2. Membuat list character yang berisi huruf a sampai z.
3. Membuat foldernya a sampai z pada direktori *training* dan *testing*.
4. Inisialisasi variabel jumlah data *training* dan *testing*.
5. Melakukan *looping* untuk menyalin file pada direktori *dataset* untuk tiap *class* ke direktori *training* sesuai dengan *class*nya.
6. Melakukan *looping* untuk menyalin file pada direktori *dataset* untuk tiap *class* ke direktori *testing* sesuai dengan *class*nya.

#### 4.3.2 Preprocessing

Setelah data *training* dan *testing* siap, tahap selanjutnya adalah melakukan *preprocessing* data. Pada tahap *preprocessing* akan dilakukan proses *convex hull*

pada setiap data pada direktori *training* dan *testing*. *Convex hull* dilakukan dengan cara menggunakan algoritma graham scan, algoritma tersebut dilakukan pada titik-titik yang didapatkan dari proses pencarian kontur pada gambar. Berikut ini adalah Pseudocode dari program *preprocessing*.

```

1. DEFINE FUNCTION cvh(pts):
2.   DEFINE FUNCTION get_slope(p1, p2):
3.     IF p1[0] EQUALS p2[0]:
4.       RETURN float('inf')
5.     ELSE:
6.       RETURN 1.0*(p1[1]-p2[1])/(p1[0]-p2[0])
7.   DEFINE FUNCTION cmp(a, b):
8.     RETURN int(a > b) - int(a < b)
9.   DEFINE FUNCTION turn(p1, p2, p3):
10.    RETURN cmp((p2[0] - p1[0])*(p3[1] - p1[1]) - (p3[0] - p1[0])*(p2[1]
    - p1[1]), 0)
11.  SET hull TO []
12.  SET pts TO sorted(pts)
13.  SET start TO pts.pop(0)
14.  SORT pts(key=lambda p: (get_slope(p,start), -p[1],p[0]))
15.  hull.append(start)
16.  FOR x IN range(len(pts)):
17.    hull.append(pts[x])
18.    WHILE len(hull) > 2 and turn(hull[-3],hull[-2],hull[-1]) != 1:
19.      hull.pop(-2)
20.  RETURN hull
21. DEFINE FUNCTION pp(diri,diro):
22.  SET directory TO diri
23.  SET outdirectory TO diro
24.  SET dirlist TO os.listdir(directory)
25.  FOR x IN dirlist:
26.    SET y TO PATH JOIN(directory,x)
27.    SET dirlist2 TO os.listdir(y)
28.    FOR z IN dirlist2:
29.      SET path TO PATH JOIN(y,z)
30.      SET outpath TO PATH JOIN(outdirectory,x)
31.      SET outpath2 TO PATH JOIN(outpath,z)
32.      SET img TO READ IMAGE(path)
33.      SET img_gray TO CONVERT IMAGE TO
    GRAYSCALE(img,cv2.COLOR_BGR2GRAY)
34.      SET blur TO GAUSSIAN BLUR(img_gray,(1,1),0)
35.      SET canny_output TO EDGE DETECTION USING CANNY(blur, 40, 255)
36.      SET contours, hierarchy TO FIND
    CONTOURS(canny_output,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
37.      SET arr TO []
38.      FOR i IN range(len(contours)):
39.        SET ca TO []
40.        FOR j IN range(len(contours[i])):
41.          ca.append([contours[i][j][0][0], contours[i][j][0][1]])
42.        arr.append(ca)
43.      SET hull TO []
44.      FOR i IN range(len(arr)):
45.        hull.append(np.array(cvh(arr[i])))
46.      SET drawing TO np.zeros((canny_output.shape[0],
    canny_output.shape[1], 3), dtype=np.uint8)
47.      DRAW CONTOURS(drawing, hull, -1, (255,0,0), 2, 8)
48.      DRAW CONTOURS(drawing, contours, -1, (0,255,0), -1)
49.      IF(outdirectory not IN CURRENT DIRECTORY):
50.        MAKE DIRECTORY(outdirectory)

```

```

51.         if(x not IN LIST DIRECTORY INSIDE(outdirectory)):
52.             MAKE DIRECTORY(outpath)
53.             WRITE IMAGE FILE(outpath2, drawing)
54. pp("Training", "TrainingPP")
55. pp("Testing", "TestingPP")

```

### Pseudocode Program 4.2 Preprocessing

Untuk lebih jelasnya kode program untuk tahap *preprocessing* beserta penjelasannya dapat dilihat pada Kode Program 4.2. Berikut adalah penjelasan Kode Program 4.2 :

1. Membuat fungsi *convex hull* dengan parameter *pts* yang nantinya akan berisi titik-titik yang didapatkan dari pencarian kontur. Algoritma *convex hull* yang digunakan adalah Graham Scan .
2. Membuat fungsi untuk menghitung lereng (*slope*).
3. Membuat fungsi untuk membandingkan (*compare*) *truth value* antara point dengan cara mengurangi hasil *truth value* dari titik tersebut kemudian mereturn hasil perbandingan *truth value* sebagai integer.
4. Membuat fungsi untuk melakukan pengecekan rotasi antara 3 titik pada stack *hull* yang mereturn nilai *truth value* dari hasil operasi dan perbandingan titik. Jika return adalah 1 maka menunjukkan bahwa rotasi antara titik tersebut adalah kekiri, dan jika hasilnya 0 atau  $< 0$  maka rotasi antara titik tersebut adalah kekanan.
5. Membuat list kosong dengan nama *hull* , dan mengurutkan titik-titik yang didapatkan dari pencarian kontur yang telah disimpan pada parameter *pts*.
6. Setelah titik diurutkan, simpan titik awal/*index* ke 0 pada variabel *start* dan keluarkan dari list *pts*.
7. Mengurutkan kembali sisa titik-titik pada variabel *pts* sesuai dengan urutan nilai hasil perhitungan lereng (*slope*) antara tiap titik pada list *pts* dengan titik awal pada variabel *start*.
8. Menambahkan titik awal / *start* pada list *hull*.
9. Melakukan *looping* sebanyak jumlah titik pada variabel *pts* yang telah diurutkan lalu menambahkannya ke list *hull*.
10. Melakukan *looping* lagi jika titik pada list *hull* sudah lebih dari 3, ketika titik pada list *hull* sudah lebih dari 2, maka akan dilakukan pengecekan rotasi antara tiga titik terakhir pada list *hull* (termasuk titik awal / *start* yang

telah ditambahkan sebelum dimulainya perulangan). Jika hasil pengecekan rotasi hasilnya adalah 0 atau  $< 0$ , artinya rotasi dari tiap titik menuju ke kanan dan 2 titik terakhir harus dikeluarkan dari list *hull*. Namun jika hasil pengecekan rotasi adalah 1, artinya rotasi dari tiap titik menuju ke arah kiri jadi tidak akan dilakukan Tindakan apapun pada titik tersebut yang telah ditambahkan ke list *hull*.

11. Mereturn list *hull*.
12. Membuat fungsi *preprocessing* / *pp* dengan parameter berupa nama direktori input dan direktori output. Fungsi ini akan digunakan untuk melakukan *preprocessing* pada direktori training dan testing.
13. Melakukan *looping* untuk membaca direktori pada direktori *training* ataupun *testing*.
14. Melakukan *looping* untuk membaca file pada tiap class yang ada di direktori *training* ataupun *testing*.
15. Mengubah gambar menjadi *grayscale*.
16. Menjalankan fungsi *gaussian blur* pada gambar yang telah dirubah ke *grayscale* untuk mengurangi *noise* pada gambar.
17. Melakukan deteksi tepi dengan fungsi *canny* untuk mendapatkan tepi dari tiap objek yang terbaca pada gambar.
18. Menjalankan fungsi pencarian kontur dengan sumber *canny\_output* dengan algoritma RETR\_TREE dan metode CHAIN\_APPROX\_SIMPLE.
19. Membuat list kosong dengan nama *arr*. Kemudian akan dilakukan *looping* untuk mengambil titik-titik pada hasil deteksi kontur dan menambahkannya pada list *arr*. Hal ini bertujuan agar titik-titik hasil pencarian kontur tersebut lebih mudah dibaca, karena nantinya akan digunakan untuk pencarian *convex hull*.
20. Membuat list *hull* kosong, kemudian melakukan *looping* *i* sebanyak panjang data pada list *arr* dan menjalankan fungsi *cvh* dengan parameter titik pada list *arr* index ke *i* yang nantinya akan mereturn titik-titik yang telah melalui algoritma *convex hull* dan menambahkannya ke list *hull*.
21. Membuat lembar hitam kosong dengan nama variabel *drawing* yang kemudian akan digambari dengan fungsi *drawcontour* dengan titik-titik



dari list *hull* dengan warna biru dan titik-titik dari list *contour* dengan warna hijau.

22. Melakukan pengecekan apakah direktori output sudah ada, jika belum ada maka buat direktori output sesuai dengan nama pada parameter. Selanjutnya melakukan pengecekan apakah direktori *x* sudah ada, jika belum maka buat direktori *x* yang berisi class abjad a-z. Setelah semuanya dibuat, selanjutnya adalah tulis file yang berisi gambar pada variabel *drawing* dengan nama file sesuai dengan nilai variabel *outpath2*.
23. Menjalankan fungsi *pp / preprocessing* dengan nilai parameter berupa direktori *input* dan *output* untuk tiap data *training* dan *testing*. Nantinya hasil dari fungsi tersebut akan menghasilkan direktori yang berisi file hasil *preprocessing* dengan nama sesuai dengan parameter direktori *output*.

#### 4.3.3 Make Model & Evaluate

Setelah data *training* dan *testing* melalui tahap *preprocessing*. Tahap selanjutnya adalah pembuatan model pembelajaran dengan menggunakan data *training* dan evaluasi model hasil pembelajaran dengan menggunakan data *testing*. Berikut ini adalah Pseudocode dari program make model & evaluate.

```
1. SET data_dir TO TRAINING DATA PATH AFTER PREPROCESSING
2. SET test_dir TO TESTING DATA PATH AFTER PREPROCESSING
3. SET target_size TO (100,100)
4. SET classes TO 26
5. SET train_datagen TO ImageDataGenerator(rescale=1./255, zoom_range=0.2, validation_split=0.2)
6. SET train_gen TO train_datagen.flow_from_directory(data_dir, target_size=target_size, shuffle=True, batch_size=32, color_mode='rgb', class_mode='categorical', subset='training')
7. SET test_generator TO ImageDataGenerator(rescale=1./255)
8. SET test_data_generator TO test_generator.flow_from_directory(test_dir, target_size=target_size, batch_size=32, shuffle=False)
9. SET val_gen TO train_datagen.flow_from_directory(data_dir, target_size=target_size, batch_size=32, color_mode='rgb', class_mode='categorical', subset='validation')
10. DEFINE FUNCTION make_model(ep):
11. SET model TO Sequential()
12. ADD CONVOLUTIONAL LAYER AND POOLING LAYER
13. COMPILE model(optimizer='Adam', loss='categorical_crossentropy', metrics=['accuracy'])
14. FIT model(train_gen, epochs=ep, validation_data=val_gen)
15. SAVE MODEL
16. DEFINE FUNCTION evaluate_test():
17. LOAD MODEL
18. SET prediksi TO load_model.predict(test_data_generator)
19. SET y_pred TO np.argmax(prediksi, axis=1)
```

```

20. SET class_labels TO list(test_data_generator.class_indices.keys())
21. SET cm TO confusion_matrix(test_data_generator.classes,y_pred)
22. OUTPUT (cm)
23. cr=classification_report(test_data_generator.classes,y_pred,target_names=
    class_labels,zero_division=1)
24. OUTPUT(cr)
25. make_model(20)
26. evaluate_test()

```

### Pseudocode Program 4.3 Make Model & Evaluate

Untuk lebih jelasnya kode program dan penjelasan untuk tahap pembuatan model dan evaluasi yang dapat dilihat pada Kode Program 4.3. Berikut adalah penjelasan kode program 4.3 :

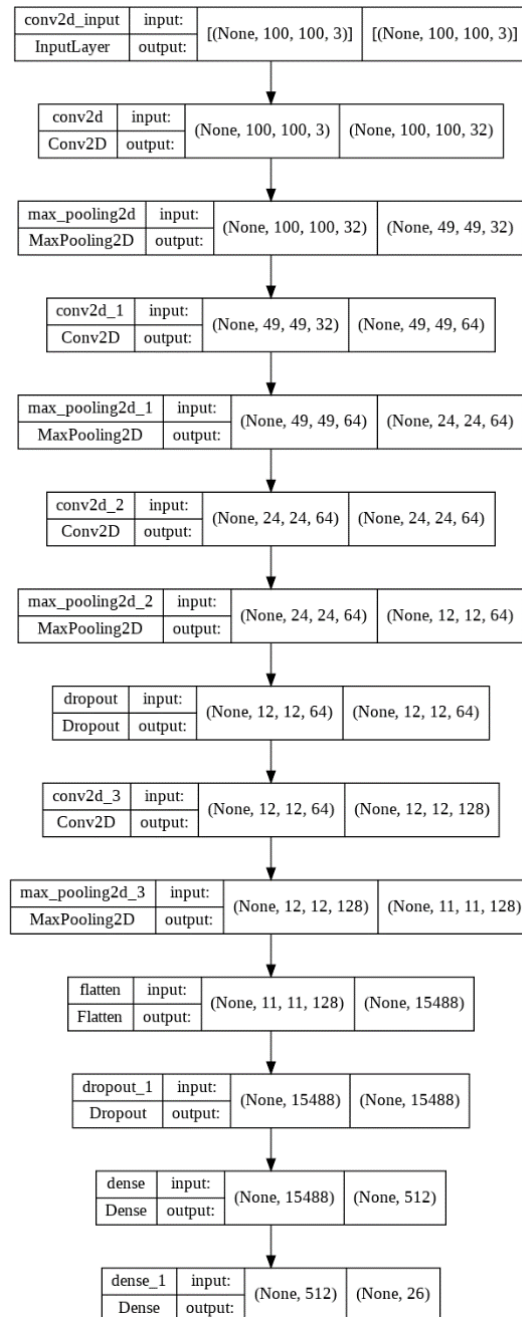
1. Membuat variabel *train\_datagen* menggunakan fungsi *ImageDataGenerator* dengan sedikit data augmentasi berupa *zoom\_range* dan *rescale* yang berfungsi untuk normalisasi nilai pixel. Serta melakukan pembagian/*split* untuk *training* dan *validation* sebesar 90:10. Kemudian memuat data *training* kedalam variabel *train\_gen* melalui variabel *train\_datagen* dan mengubah ukuran untuk tiap gambar menjadi (100x100), adapun parameter lainnya seperti *shuffle*, *batch\_size*, *color\_mode*, *class\_mode*, serta *subset* yang nilainya bisa dilihat pada kode diatas.
2. Membuat variabel *test\_generator* menggunakan fungsi *ImageDataGenerator* dengan sedikit data augmentasi berupa *rescale* yang berfungsi untuk normalisasi nilai pixel. Kemudian memuat data *testing* kedalam variabel *test\_data\_generator* melalui variabel *test\_generator* dan mengubah ukuran untuk tiap gambar menjadi (100x100), adapun parameter lainnya seperti *batch\_size*, dan *shuffle* yang nilainya bisa dilihat pada kode diatas.
3. Membuat variabel *val\_gen* dan memuat data *validation* kedalam variabel *val\_gen* melalui variabel *train\_datagen* yang telah dilakukan pembagian / *split* data tadi dengan pembagian 90:10 adapun parameter lainnya seperti *batch\_size*, *color\_mode*, *class\_mode*, serta *subset* yang nilainya bisa dilihat pada kode diatas
4. Membuat fungsi *make\_model* dengan parameter berupa jumlah *epoch*.
5. Mendefinisikan model yang digunakan, disini peneliti menggunakan model *sequential()*.

6. Menambahkan layer konvolusi pada model dengan jumlah *filter* seperti pada kode, dengan ukuran kernelnya adalah 3, pegeseran/*stride* sebesar 1, fungsi aktivasi yang digunakan adalah *RELU*, padding yang sama, dan *input\_shape* dengan ukuran (100x100) dengan kedalaman/*depth* sebesar 3.
7. Menambahkan layer pooling dengan fungsi *MaxPool* dengan ukuran *pool/pool\_size* dan pergeseran/*stride* yang bervariasi seperti yang dapat dilihat pada Kode Program 4.3.
8. Menambahkan dropout dengan nilai sebesar 0.5, artinya jika nilai yang didapat adalah dibawah 0.5 maka parameter tidak akan dilatih. Hal ini juga bertujuan untuk menghindari *Overfitting*.
9. Mengkonversikan matriks multidimensi menjadi matriks 2 dimensi dengan memanggil fungsi *Flatten()*.
10. Menambahkan dense layer dengan 512 *neuron* dengan fungsi aktivasi *relu* dan 26 *neuron* / sesuai dengan jumlah class dengan fungsi aktivasi *softmax*.
11. Mengkompil / *compile* model dengan menggunakan *optimizer* “Adam” , *loss* “*categorical\_crossentropy*”, dan metrik *accuracy*.
12. Menggambar struktur model CNN / plot model yang menghasilkan file bernama “model\_plot.png”. Hal ini bertujuan agar struktur model yang dibuat bisa dilihat.
13. Menampilkan ringkasan model yang akan dibuat
14. Membuat variabel *history* dan menjalankan fungsi *model.fit* untuk memulai proses pelatihan. Nantinya hasil pelatihannya akan tersimpan pada variabel *history*. Dan hasil trainingnya akan digambarkan dengan menggunakan *library matplotlib*.
15. Menyimpan model yang telah dibuat dengan mengkonversinya menjadi file ber ekstensi *.json* dengan nama “model.json” dan juga menyimpan model bobot dengan nama “model.h5”.
16. Membuat metode *evaluate\_test()* untuk melakukan evaluasi terhadap data *testing* dengan menggunakan model hasil pembelajaran CNN.
17. Memuat / *load model.json* dan model bobot *model.h5* untuk digunakan pada proses evaluasi data testing.

18. Membuat variabel *prediksi* yang didalamnya memanggil variabel *load\_model* untuk memuat model dan melakukan prediksi data testing dari variabel *test\_data\_generator*. Dan membuat variabel *y\_pred* yang didalamnya memanggil fungsi *numpy argmax* dengan parameter *prediksi* dan *axis=1*. Selanjutnya adalah membuat variabel yang akan memuat label dari class *test\_data\_generator* yang nantinya akan ditampilkan saat pembuatan *classification\_report* dan *confusion\_matrix*.
19. Membuat variabel *cm* yang didalamnya menjalankan fungsi untuk membuat *confusion\_matrix* dengan parameter berupa label sebenarnya yang berasal dari class label data *testing* dan label prediksi yang berasal dari variabel *predict*.
20. Setelah didapatkan hasil *confusion\_matrix*nya, buat variabel untuk menampilkan gambar tabel *confusion\_matrix* dan tampilkan tabelnya.
21. Membuat variabel *cr* dan *crv* yang didalamnya akan menjalankan fungsi *classification\_report* untuk membuat laporan hasil klasifikasi data testing yang telah diprediksi. Perbedaan *cr* dan *crv* adalah *crv* menampilkan laporan hasil klasifikasi dalam bentuk *dictionary* sehingga tiap variabel dalam hasil klasifikasinya nantinya bisa di akses, karena nantinya nilai dari akurasi dalam laporan hasil klasifikasi akan diambil dan ditambahkan kedalam list *acc\_avg*.
22. Membuat list kosong *acc\_avg* yang nantinya akan digunakan untuk menampung hasil dari akurasi. Kemudian jalankan fungsi pembuatan model CNN dengan parameter berupa jumlah epoch. Setelah model dibuat lalu lanjutkan dengan memanggil fungsi evaluasi.

#### **4.4 Struktur Model CNN yang digunakan**

Setelah kita membuat program untuk membuat model CNN yang akan digunakan untuk proses pelatihan, didalamnya akan terdapat struktur CNN yang akan digunakan, dimana pada struktur CNN tersebut akan terdapat keterangan tentang *layer*, *output shape*, dan jumlah parameter. Plot struktur CNN yang dihasilkan dan akan digunakan dapat dilihat pada Gambar 4.2.



**Gambar 4.2** Struktur Model CNN

Gambar 4.2 adalah struktur model yang dihasilkan jika ditampilkan dalam bentuk gambar. Namun, jika ditampilkan dalam bentuk gambar jumlah parameter tidak akan ditampilkan. Jika ingin melihat jumlah parameter, perlu ditampilkan ringkasan dari struktur CNN yang digunakan. Untuk menampilkannya bisa menggunakan perintah `model.summary()`. Berikut adalah *output* ringkasan model yang dibuat jika ditampilkan dengan perintah `model.summary()`.

```
Model: "sequential"
```

| Layer (type)                    | Output Shape         | Param # |
|---------------------------------|----------------------|---------|
| conv2d (Conv2D)                 | (None, 100, 100, 32) | 896     |
| max_pooling2d (MaxPooling 2D)   | (None, 49, 49, 32)   | 0       |
| conv2d_1 (Conv2D)               | (None, 49, 49, 64)   | 18496   |
| max_pooling2d_1 (MaxPooling 2D) | (None, 24, 24, 64)   | 0       |
| conv2d_2 (Conv2D)               | (None, 24, 24, 64)   | 36928   |
| max_pooling2d_2 (MaxPooling 2D) | (None, 12, 12, 64)   | 0       |
| dropout (Dropout)               | (None, 12, 12, 64)   | 0       |
| conv2d_3 (Conv2D)               | (None, 12, 12, 128)  | 73856   |
| max_pooling2d_3 (MaxPooling 2D) | (None, 11, 11, 128)  | 0       |
| flatten (Flatten)               | (None, 15488)        | 0       |
| dropout_1 (Dropout)             | (None, 15488)        | 0       |
| dense (Dense)                   | (None, 512)          | 7930368 |
| dense_1 (Dense)                 | (None, 26)           | 13338   |

Total params: 8,073,882

Trainable params: 8,073,882

Non-trainable params: 0

Struktur CNN yang digunakan untuk membandingkan sistem dengan *Convex Hull* + CNN dan sistem dengan CNN tanpa *Convex Hull* adalah sama. Struktur diatas akan digunakan untuk semua skenario pengujian.

## 4.5 Hasil Ujicoba

Berdasarkan penelitian yang dilakukan terdapat 4 skenario yang telah ditentukan, berikut adalah hasil ujicoba dari tiap scenario tersebut.

### 4.5.1 Skenario 1

Pada skenario 1, akan dilakukan ujicoba dengan presenase pembagian data *training* dan *testing* 60:40, pembagian data *training* dan *validation* 80:20, dengan epoch 20 untuk pembelajaran dalam pembuatan model CNNnya. Berikut adalah hasil pengujian dari skenario 1.

#### 4.5.1.1 Hasil Pembelajaran Sistem *Convex Hull* + CNN

Setelah struktur model dibuat, akan dilakukan proses kompilasi dan memulai proses pembelajaran terhadap data *TrainingPP* dengan menggunakan struktur model tersebut. Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch.

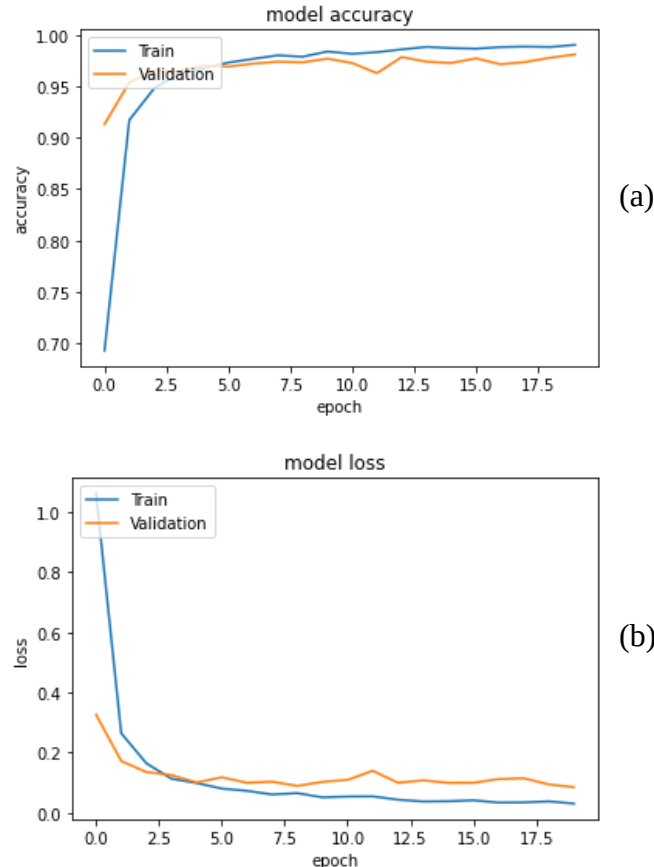
```
Epoch 1/20
390/390 [=====] - 3377s 9s/step - loss: 1.0615 -
accuracy: 0.6928 - val_loss: 0.3260 - val_accuracy: 0.9131
Epoch 2/20
390/390 [=====] - 51s 131ms/step - loss: 0.2644
- accuracy: 0.9171 - val_loss: 0.1720 - val_accuracy: 0.9538
Epoch 3/20
390/390 [=====] - 51s 130ms/step - loss: 0.1642
- accuracy: 0.9479 - val_loss: 0.1350 - val_accuracy: 0.9654
Epoch 4/20
390/390 [=====] - 51s 131ms/step - loss: 0.1128
- accuracy: 0.9622 - val_loss: 0.1252 - val_accuracy: 0.9651
Epoch 5/20
390/390 [=====] - 51s 130ms/step - loss: 0.0987
- accuracy: 0.9675 - val_loss: 0.1005 - val_accuracy: 0.9702
Epoch 6/20
390/390 [=====] - 51s 130ms/step - loss: 0.0801
- accuracy: 0.9732 - val_loss: 0.1179 - val_accuracy: 0.9692
Epoch 7/20
390/390 [=====] - 51s 131ms/step - loss: 0.0727
- accuracy: 0.9768 - val_loss: 0.0994 - val_accuracy: 0.9721
Epoch 8/20
390/390 [=====] - 51s 132ms/step - loss: 0.0606
- accuracy: 0.9802 - val_loss: 0.1032 - val_accuracy: 0.9740
Epoch 9/20
390/390 [=====] - 51s 132ms/step - loss: 0.0651
- accuracy: 0.9788 - val_loss: 0.0891 - val_accuracy: 0.9734
Epoch 10/20
390/390 [=====] - 51s 130ms/step - loss: 0.0511
- accuracy: 0.9839 - val_loss: 0.1026 - val_accuracy: 0.9769
Epoch 11/20
390/390 [=====] - 51s 130ms/step - loss: 0.0536
- accuracy: 0.9817 - val_loss: 0.1091 - val_accuracy: 0.9728
Epoch 12/20
390/390 [=====] - 51s 131ms/step - loss: 0.0542
- accuracy: 0.9833 - val_loss: 0.1395 - val_accuracy: 0.9628
Epoch 13/20
390/390 [=====] - 51s 130ms/step - loss: 0.0432
- accuracy: 0.9860 - val_loss: 0.0999 - val_accuracy: 0.9785
Epoch 14/20
390/390 [=====] - 51s 132ms/step - loss: 0.0373
- accuracy: 0.9884 - val_loss: 0.1074 - val_accuracy: 0.9740
Epoch 15/20
390/390 [=====] - 51s 131ms/step - loss: 0.0381
- accuracy: 0.9873 - val_loss: 0.0995 - val_accuracy: 0.9728
Epoch 16/20
390/390 [=====] - 51s 130ms/step - loss: 0.0412
- accuracy: 0.9867 - val_loss: 0.0997 - val_accuracy: 0.9772
Epoch 17/20
390/390 [=====] - 51s 131ms/step - loss: 0.0337
- accuracy: 0.9882 - val_loss: 0.1117 - val_accuracy: 0.9715
Epoch 18/20
390/390 [=====] - 50s 128ms/step - loss: 0.0341
- accuracy: 0.9887 - val_loss: 0.1146 - val_accuracy: 0.9737
Epoch 19/20
```

```

390/390 [=====] - 51s 130ms/step - loss: 0.0377
- accuracy: 0.9883 - val_loss: 0.0942 - val_accuracy: 0.9779
Epoch 20/20
390/390 [=====] - 51s 132ms/step - loss: 0.0303
- accuracy: 0.9904 - val_loss: 0.0849 - val_accuracy: 0.9811

```

Jika hasil *output* proses pembelajaran tiap epoch diatas digambarkan dalam bentuk grafik hasilnya akan seperti pada Gambar 4.3.



**Gambar 4.3** (a) dan (b) Hasil *Output* Pembelajaran Sistem Dengan *Convex Hull* + CNN Pada Skenario 1

Setelah proses pembelajaran / *learning* dilakukan, maka akan menghasilkan file model hasil pembelajaran, yaitu model\_json.json dan model.h5 yang nantinya akan digunakan untuk proses evaluasi.

#### 4.5.1.2 Hasil Evaluasi Sistem *Convex Hull* + CNN

Setelah file model hasil pembelajaran dihasilkan, selanjutnya akan dilakukan prediksi dan evaluasi pada data *testingPP* dengan menggunakan model hasil hasil pembelajaran tersebut. Tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* dapat dilihat pada Tabel 4.2. Untuk mendapatkan nilai akurasi, presisi, dan recall, maka akan dilakukan evaluasi menggunakan fungsi



*classification\_report* . Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasi.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| a            | 0.97      | 0.97   | 0.97     | 400     |
| b            | 0.98      | 0.95   | 0.96     | 400     |
| c            | 0.98      | 1.00   | 0.99     | 400     |
| d            | 0.99      | 0.91   | 0.95     | 400     |
| e            | 0.99      | 0.99   | 0.99     | 400     |
| f            | 0.98      | 0.99   | 0.99     | 400     |
| g            | 0.99      | 0.99   | 0.99     | 400     |
| h            | 1.00      | 1.00   | 1.00     | 400     |
| i            | 0.99      | 0.95   | 0.97     | 400     |
| j            | 1.00      | 1.00   | 1.00     | 400     |
| k            | 0.98      | 0.98   | 0.98     | 400     |
| l            | 0.96      | 0.99   | 0.98     | 400     |
| m            | 1.00      | 1.00   | 1.00     | 400     |
| n            | 1.00      | 1.00   | 1.00     | 400     |
| o            | 0.99      | 0.98   | 0.98     | 400     |
| p            | 1.00      | 1.00   | 1.00     | 400     |
| q            | 1.00      | 0.99   | 0.99     | 400     |
| r            | 0.95      | 0.99   | 0.97     | 400     |
| s            | 0.99      | 1.00   | 1.00     | 400     |
| t            | 1.00      | 1.00   | 1.00     | 400     |
| u            | 1.00      | 0.99   | 1.00     | 400     |
| v            | 0.94      | 0.95   | 0.94     | 400     |
| w            | 0.98      | 0.98   | 0.98     | 400     |
| x            | 1.00      | 1.00   | 1.00     | 400     |
| y            | 1.00      | 1.00   | 1.00     | 400     |
| z            | 0.98      | 1.00   | 0.99     | 400     |
| accuracy     |           |        | 0.99     | 10400   |
| macro avg    | 0.99      | 0.99   | 0.99     | 10400   |
| weighted avg | 0.99      | 0.99   | 0.99     | 10400   |

Accuracy : [0.9854807692307692]

Nilai akurasi yang didapatkan dari fungsi *classification\_report* adalah 0.99 atau 99%, namun nilai tersebut adalah nilai akurasi yang dibulatkan. Maka dari itu perlu dilakukan pengambilan nilai akurasi yang sebenarnya dari *dictionary* hasil *classification\_report*. Setelah dilakukan pengambilan nilai, akurasi yang didapatkan adalah 0.9854807692307692 atau 98.85%.

#### 4.5.1.3 Hasil Pembelajaran Sistem CNN Tanpa *Convex Hull*

Setelah sebelumnya proses pembelajaran untuk sistem *Convex Hull* + CNN, selanjutnya akan dilakukan pembelajaran untuk sistem CNN tanpa *Convex Hull* dengan menggunakan struktur model CNN terhadap data *training*. Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch.

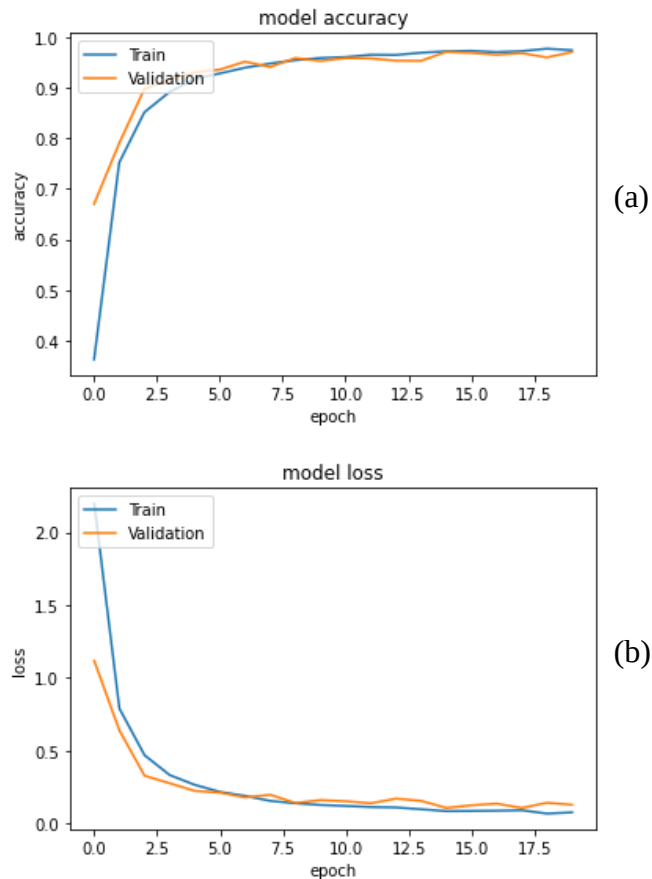
```
Epoch 1/20
390/390 [=====] - 4359s 11s/step - loss: 2.1979
- accuracy: 0.3631 - val_loss: 1.1184 - val_accuracy: 0.6702
Epoch 2/20
390/390 [=====] - 51s 131ms/step - loss: 0.7910
- accuracy: 0.7518 - val_loss: 0.6409 - val_accuracy: 0.7901
Epoch 3/20
390/390 [=====] - 51s 130ms/step - loss: 0.4703
- accuracy: 0.8513 - val_loss: 0.3292 - val_accuracy: 0.8974
```

```

Epoch 4/20
390/390 [=====] - 51s 130ms/step - loss: 0.3337
- accuracy: 0.8913 - val_loss: 0.2776 - val_accuracy: 0.9157
Epoch 5/20
390/390 [=====] - 51s 130ms/step - loss: 0.2666
- accuracy: 0.9172 - val_loss: 0.2251 - val_accuracy: 0.9298
Epoch 6/20
390/390 [=====] - 50s 129ms/step - loss: 0.2184
- accuracy: 0.9280 - val_loss: 0.2137 - val_accuracy: 0.9356
Epoch 7/20
390/390 [=====] - 50s 129ms/step - loss: 0.1909
- accuracy: 0.9393 - val_loss: 0.1803 - val_accuracy: 0.9513
Epoch 8/20
390/390 [=====] - 51s 131ms/step - loss: 0.1565
- accuracy: 0.9476 - val_loss: 0.1986 - val_accuracy: 0.9407
Epoch 9/20
390/390 [=====] - 51s 130ms/step - loss: 0.1401
- accuracy: 0.9541 - val_loss: 0.1426 - val_accuracy: 0.9583
Epoch 10/20
390/390 [=====] - 50s 129ms/step - loss: 0.1286
- accuracy: 0.9584 - val_loss: 0.1623 - val_accuracy: 0.9526
Epoch 11/20
390/390 [=====] - 51s 130ms/step - loss: 0.1220
- accuracy: 0.9603 - val_loss: 0.1542 - val_accuracy: 0.9583
Epoch 12/20
390/390 [=====] - 51s 131ms/step - loss: 0.1144
- accuracy: 0.9648 - val_loss: 0.1403 - val_accuracy: 0.9577
Epoch 13/20
390/390 [=====] - 51s 130ms/step - loss: 0.1115
- accuracy: 0.9644 - val_loss: 0.1724 - val_accuracy: 0.9532
Epoch 14/20
390/390 [=====] - 51s 132ms/step - loss: 0.0991
- accuracy: 0.9688 - val_loss: 0.1568 - val_accuracy: 0.9529
Epoch 15/20
390/390 [=====] - 51s 130ms/step - loss: 0.0863
- accuracy: 0.9716 - val_loss: 0.1085 - val_accuracy: 0.9705
Epoch 16/20
390/390 [=====] - 51s 130ms/step - loss: 0.0874
- accuracy: 0.9725 - val_loss: 0.1266 - val_accuracy: 0.9686
Epoch 17/20
390/390 [=====] - 51s 130ms/step - loss: 0.0892
- accuracy: 0.9700 - val_loss: 0.1377 - val_accuracy: 0.9647
Epoch 18/20
390/390 [=====] - 50s 129ms/step - loss: 0.0923
- accuracy: 0.9720 - val_loss: 0.1083 - val_accuracy: 0.9679
Epoch 19/20
390/390 [=====] - 50s 129ms/step - loss: 0.0695
- accuracy: 0.9768 - val_loss: 0.1437 - val_accuracy: 0.9596
Epoch 20/20
390/390 [=====] - 51s 131ms/step - loss: 0.0778
- accuracy: 0.9739 - val_loss: 0.1300 - val_accuracy: 0.9702

```

Jika proses hasil pembelajaran untuk tiap epoch diatas digambarkan dalam bentuk grafik hasilnya akan seperti pada Gambar 4.4.



**Gambar 4.4** (a) dan (b) Hasil *Output* Pembelajaran Sistem CNN Tanpa *Convex Hull* Pada Skenario 1

Setelah proses pembelajaran dilakukan, maka akan menghasilkan file model hasil pembelajaran, yaitu `modeln_json.json` dan `modeln.h5` yang nantinya akan digunakan untuk proses evaluasi.

#### 4.5.1.4 Hasil Evaluasi Sistem CNN Tanpa *Convex Hull*

Setelah file model hasil pembelajaran dihasilkan, selanjutnya akan dilakukan prediksi dan evaluasi pada data *testing* dengan menggunakan model hasil hasil pembelajaran tersebut. Tabel hasil prediksi dengan menggunakan fungsi `confusion_matrix` dapat dilihat pada Tabel 4.3. Berikut adalah nilai akurasi, presisi, dan recall untuk tiap kelas dengan menggunakan fungsi `classification_report`.

|   | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| a | 0.93      | 0.93   | 0.93     | 400     |
| b | 0.90      | 0.56   | 0.69     | 400     |
| c | 1.00      | 0.90   | 0.94     | 400     |
| d | 0.95      | 0.53   | 0.68     | 400     |
| e | 0.99      | 1.00   | 0.99     | 400     |
| f | 0.87      | 0.97   | 0.92     | 400     |
| g | 0.97      | 0.98   | 0.97     | 400     |

|              |      |      |      |       |
|--------------|------|------|------|-------|
| h            | 0.99 | 0.99 | 0.99 | 400   |
| i            | 0.99 | 0.89 | 0.93 | 400   |
| j            | 1.00 | 1.00 | 1.00 | 400   |
| k            | 0.95 | 0.96 | 0.95 | 400   |
| l            | 0.97 | 0.93 | 0.95 | 400   |
| m            | 1.00 | 0.99 | 1.00 | 400   |
| n            | 1.00 | 1.00 | 1.00 | 400   |
| o            | 0.86 | 0.99 | 0.92 | 400   |
| p            | 1.00 | 1.00 | 1.00 | 400   |
| q            | 0.99 | 1.00 | 1.00 | 400   |
| r            | 0.90 | 0.95 | 0.92 | 400   |
| s            | 0.97 | 1.00 | 0.99 | 400   |
| t            | 1.00 | 0.99 | 1.00 | 400   |
| u            | 1.00 | 0.99 | 1.00 | 400   |
| v            | 0.68 | 0.93 | 0.78 | 400   |
| w            | 0.75 | 0.97 | 0.85 | 400   |
| x            | 0.99 | 1.00 | 1.00 | 400   |
| y            | 1.00 | 1.00 | 1.00 | 400   |
| z            | 0.99 | 1.00 | 0.99 | 400   |
| accuracy     |      |      | 0.94 | 10400 |
| macro avg    | 0.95 | 0.94 | 0.94 | 10400 |
| weighted avg | 0.95 | 0.94 | 0.94 | 10400 |

Accuracy : [0.9406730769230769]

Nilai akurasi yang didapatkan dari fungsi *classification\_report* adalah 0.99 atau 99%, dan setelah dilakukan pengambilan nilai akurasi yang sebenarnya dari *dictionary* hasil *classification\_report*, akurasi yang didapatkan adalah 0.9406730769230769 atau 94.06%.

## 4.5.2 Skenario 2

Pada skenario 2, akan dilakukan ujicoba dengan presenase pembagian data *training* dan *testing* 70:30, pembagian data *training* dan *validation* 80:20, dengan epoch 20 untuk pembelajaran dalam pembuatan model CNNnya. Berikut adalah hasil pengujian dari skenario 2.

### 4.5.2.1 Hasil Pembelajaran Sistem *Convex Hull* + CNN

Setelah struktur model dibuat, akan dilakukan proses kompilasi dan memulai proses pembelajaran terhadap data *TrainingPP* dengan menggunakan struktur model yang telah dibuat. Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch.

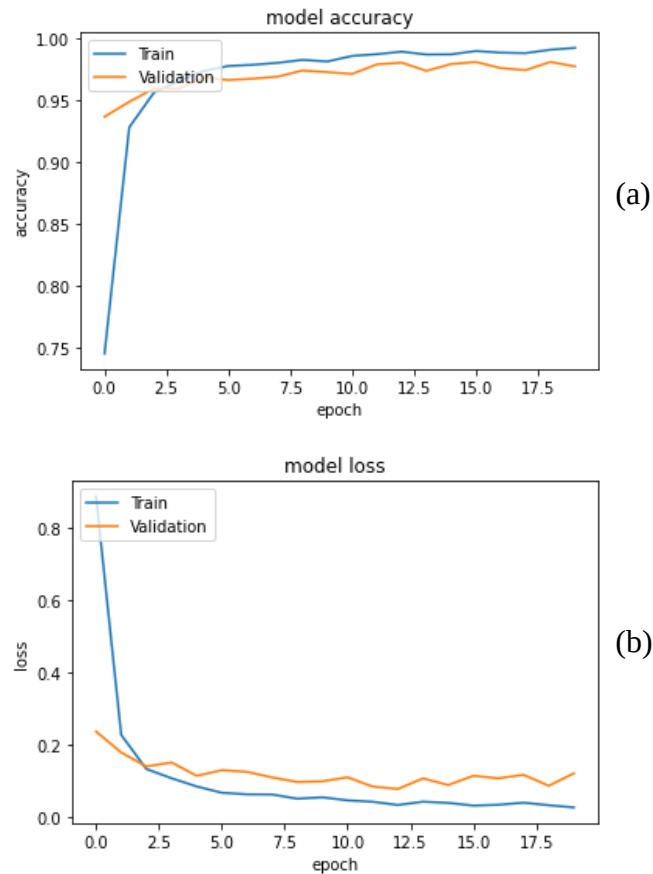
```
Epoch 1/20
455/455 [=====] - 4000s 9s/step - loss: 0.8866 -
accuracy: 0.7455 - val_loss: 0.2363 - val_accuracy: 0.9368
Epoch 2/20
455/455 [=====] - 57s 126ms/step - loss: 0.2269
- accuracy: 0.9283 - val_loss: 0.1782 - val_accuracy: 0.9489
Epoch 3/20
455/455 [=====] - 57s 126ms/step - loss: 0.1324
- accuracy: 0.9565 - val_loss: 0.1396 - val_accuracy: 0.9596
Epoch 4/20
455/455 [=====] - 57s 126ms/step - loss: 0.1070
- accuracy: 0.9655 - val_loss: 0.1506 - val_accuracy: 0.9596
```

```

Epoch 5/20
455/455 [=====] - 57s 125ms/step - loss: 0.0842
- accuracy: 0.9737 - val_loss: 0.1137 - val_accuracy: 0.9684
Epoch 6/20
455/455 [=====] - 57s 126ms/step - loss: 0.0668
- accuracy: 0.9778 - val_loss: 0.1292 - val_accuracy: 0.9665
Epoch 7/20
455/455 [=====] - 57s 126ms/step - loss: 0.0624
- accuracy: 0.9788 - val_loss: 0.1246 - val_accuracy: 0.9676
Epoch 8/20
455/455 [=====] - 58s 126ms/step - loss: 0.0619
- accuracy: 0.9804 - val_loss: 0.1093 - val_accuracy: 0.9692
Epoch 9/20
455/455 [=====] - 57s 126ms/step - loss: 0.0503
- accuracy: 0.9828 - val_loss: 0.0966 - val_accuracy: 0.9742
Epoch 10/20
455/455 [=====] - 57s 126ms/step - loss: 0.0541
- accuracy: 0.9815 - val_loss: 0.0985 - val_accuracy: 0.9728
Epoch 11/20
455/455 [=====] - 57s 126ms/step - loss: 0.0458
- accuracy: 0.9859 - val_loss: 0.1095 - val_accuracy: 0.9714
Epoch 12/20
455/455 [=====] - 57s 126ms/step - loss: 0.0418
- accuracy: 0.9874 - val_loss: 0.0838 - val_accuracy: 0.9791
Epoch 13/20
455/455 [=====] - 57s 125ms/step - loss: 0.0328
- accuracy: 0.9894 - val_loss: 0.0772 - val_accuracy: 0.9805
Epoch 14/20
455/455 [=====] - 58s 127ms/step - loss: 0.0420
- accuracy: 0.9871 - val_loss: 0.1066 - val_accuracy: 0.9739
Epoch 15/20
455/455 [=====] - 57s 125ms/step - loss: 0.0387
- accuracy: 0.9872 - val_loss: 0.0881 - val_accuracy: 0.9794
Epoch 16/20
455/455 [=====] - 57s 126ms/step - loss: 0.0312
- accuracy: 0.9899 - val_loss: 0.1138 - val_accuracy: 0.9810
Epoch 17/20
455/455 [=====] - 57s 126ms/step - loss: 0.0335
- accuracy: 0.9886 - val_loss: 0.1068 - val_accuracy: 0.9761
Epoch 18/20
455/455 [=====] - 57s 126ms/step - loss: 0.0392
- accuracy: 0.9882 - val_loss: 0.1163 - val_accuracy: 0.9745
Epoch 19/20
455/455 [=====] - 58s 127ms/step - loss: 0.0322
- accuracy: 0.9909 - val_loss: 0.0861 - val_accuracy: 0.9810
Epoch 20/20
455/455 [=====] - 57s 126ms/step - loss: 0.0262
- accuracy: 0.9924 - val_loss: 0.1205 - val_accuracy: 0.9775

```

Jika proses hasil pembelajaran untuk tiap epoch diatas digambarkan dalam bentuk grafik hasilnya akan seperti pada Gambar 4.5.



**Gambar 4.5** (a) dan (b) Hasil *Output* Pembelajaran Sistem Dengan *Convex Hull* + CNN Pada Skenario 2

Setelah proses pembelajaran / *learning* dilakukan, maka akan menghasilkan file model hasil pembelajaran, yaitu `model_json.json` dan `model.h5` yang nantinya akan digunakan untuk proses evaluasi.

#### 4.5.2.2 Hasil Evaluasi Sistem *Convex Hull* + CNN

Setelah file model hasil pembelajaran dihasilkan, selanjutnya akan dilakukan prediksi dan evaluasi pada data *testingPP* dengan menggunakan model hasil hasil pembelajaran tersebut. Tabel hasil prediksi dengan menggunakan fungsi `confusion_matrix` dapat dilihat pada Tabel 4.4. Untuk mendapatkan nilai akurasi, presisi, dan recall, maka akan dilakukan evaluasi menggunakan fungsi `classification_report`. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya.

|   | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| a | 0.94      | 1.00   | 0.97     | 300     |
| b | 0.96      | 0.94   | 0.95     | 300     |
| c | 0.99      | 0.99   | 0.99     | 300     |
| d | 0.99      | 0.83   | 0.91     | 300     |
| e | 0.99      | 1.00   | 1.00     | 300     |

|              |      |      |      |      |
|--------------|------|------|------|------|
| f            | 0.99 | 0.98 | 0.99 | 300  |
| g            | 0.98 | 0.93 | 0.96 | 300  |
| h            | 1.00 | 1.00 | 1.00 | 300  |
| i            | 0.99 | 0.93 | 0.96 | 300  |
| j            | 0.99 | 1.00 | 1.00 | 300  |
| k            | 0.97 | 0.98 | 0.98 | 300  |
| l            | 0.97 | 0.97 | 0.97 | 300  |
| m            | 1.00 | 1.00 | 1.00 | 300  |
| n            | 1.00 | 1.00 | 1.00 | 300  |
| o            | 0.98 | 0.96 | 0.97 | 300  |
| p            | 0.95 | 1.00 | 0.98 | 300  |
| q            | 1.00 | 1.00 | 1.00 | 300  |
| r            | 0.94 | 0.99 | 0.97 | 300  |
| s            | 0.98 | 1.00 | 0.99 | 300  |
| t            | 1.00 | 1.00 | 1.00 | 300  |
| u            | 0.99 | 0.99 | 0.99 | 300  |
| v            | 0.85 | 0.97 | 0.91 | 300  |
| w            | 0.97 | 0.96 | 0.97 | 300  |
| x            | 1.00 | 1.00 | 1.00 | 300  |
| y            | 1.00 | 0.99 | 1.00 | 300  |
| z            | 1.00 | 1.00 | 1.00 | 300  |
| accuracy     |      |      | 0.98 | 7800 |
| macro avg    | 0.98 | 0.98 | 0.98 | 7800 |
| weighted avg | 0.98 | 0.98 | 0.98 | 7800 |

Accuracy : [0.9774358974358974]

Nilai akurasi yang didapatkan dari fungsi *classification\_report* adalah 0.98 atau 98% namun nilai akurasi ini adalah nilai yang dibulatkan, dan setelah dilakukan pengambilan nilai akurasi yang sebenarnya dari *dictionary* hasil *classification\_report*, akurasi yang didapatkan adalah 0.9774358974358974 atau 97.74%.

#### 4.5.2.3 Hasil Pembelajaran Sistem CNN Tanpa *Convex Hull*

Setelah sebelumnya proses pembelajaran untuk sistem *Convex Hull* + CNN, selanjutnya akan dilakukan pembelajaran untuk sistem CNN tanpa *Convex Hull* dengan menggunakan struktur model CNN terhadap data *training*. Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch.

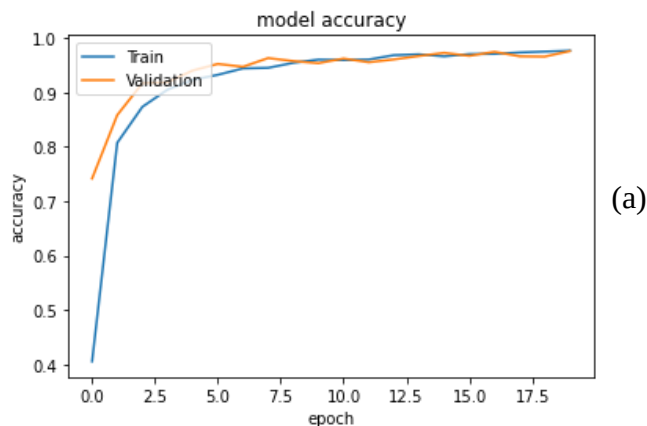
```
Epoch 1/20
455/455 [=====] - 57s 124ms/step - loss: 1.9905
- accuracy: 0.4058 - val_loss: 0.8862 - val_accuracy: 0.7415
Epoch 2/20
455/455 [=====] - 56s 123ms/step - loss: 0.6173
- accuracy: 0.8071 - val_loss: 0.4792 - val_accuracy: 0.8580
Epoch 3/20
455/455 [=====] - 56s 123ms/step - loss: 0.3993
- accuracy: 0.8731 - val_loss: 0.3204 - val_accuracy: 0.9165
Epoch 4/20
455/455 [=====] - 56s 123ms/step - loss: 0.2986
- accuracy: 0.9047 - val_loss: 0.2692 - val_accuracy: 0.9181
Epoch 5/20
455/455 [=====] - 56s 123ms/step - loss: 0.2412
- accuracy: 0.9227 - val_loss: 0.2036 - val_accuracy: 0.9398
Epoch 6/20
455/455 [=====] - 56s 124ms/step - loss: 0.2105
- accuracy: 0.9319 - val_loss: 0.1642 - val_accuracy: 0.9519
Epoch 7/20
```

```

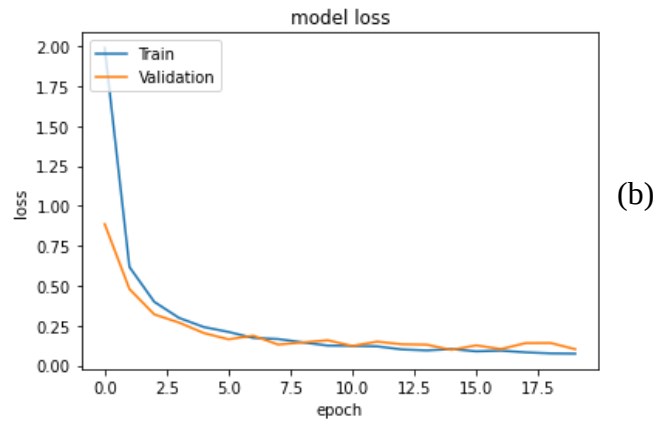
455/455 [=====] - 56s 124ms/step - loss: 0.1728
- accuracy: 0.9437 - val_loss: 0.1870 - val_accuracy: 0.9467
Epoch 8/20
455/455 [=====] - 56s 124ms/step - loss: 0.1660
- accuracy: 0.9447 - val_loss: 0.1313 - val_accuracy: 0.9626
Epoch 9/20
455/455 [=====] - 56s 123ms/step - loss: 0.1452
- accuracy: 0.9539 - val_loss: 0.1448 - val_accuracy: 0.9571
Epoch 10/20
455/455 [=====] - 56s 124ms/step - loss: 0.1250
- accuracy: 0.9599 - val_loss: 0.1583 - val_accuracy: 0.9533
Epoch 11/20
455/455 [=====] - 56s 123ms/step - loss: 0.1230
- accuracy: 0.9591 - val_loss: 0.1228 - val_accuracy: 0.9618
Epoch 12/20
455/455 [=====] - 57s 124ms/step - loss: 0.1200
- accuracy: 0.9600 - val_loss: 0.1498 - val_accuracy: 0.9549
Epoch 13/20
455/455 [=====] - 56s 124ms/step - loss: 0.1009
- accuracy: 0.9680 - val_loss: 0.1331 - val_accuracy: 0.9602
Epoch 14/20
455/455 [=====] - 56s 123ms/step - loss: 0.0946
- accuracy: 0.9696 - val_loss: 0.1307 - val_accuracy: 0.9665
Epoch 15/20
455/455 [=====] - 56s 123ms/step - loss: 0.1030
- accuracy: 0.9659 - val_loss: 0.0980 - val_accuracy: 0.9723
Epoch 16/20
455/455 [=====] - 56s 123ms/step - loss: 0.0889
- accuracy: 0.9701 - val_loss: 0.1264 - val_accuracy: 0.9668
Epoch 17/20
455/455 [=====] - 56s 124ms/step - loss: 0.0931
- accuracy: 0.9704 - val_loss: 0.1033 - val_accuracy: 0.9742
Epoch 18/20
455/455 [=====] - 57s 125ms/step - loss: 0.0827
- accuracy: 0.9729 - val_loss: 0.1405 - val_accuracy: 0.9659
Epoch 19/20
455/455 [=====] - 56s 124ms/step - loss: 0.0760
- accuracy: 0.9746 - val_loss: 0.1412 - val_accuracy: 0.9654
Epoch 20/20
455/455 [=====] - 56s 124ms/step - loss: 0.0746
- accuracy: 0.9762 - val_loss: 0.1024 - val_accuracy: 0.9755

```

Jika proses hasil pembelajaran untuk tiap epoch diatas digambarkan dalam bentuk grafik hasilnya akan seperti pada Gambar 4.6 berikut.







**Gambar 4.6** (a) dan (b) Hasil Output Pembelajaran Sistem CNN Tanpa *Convex Hull* Pada Skenario 2

#### 4.5.2.4 Hasil Evaluasi Sistem CNN Tanpa *Convex Hull*

Setelah file model hasil pembelajaran dihasilkan, selanjutnya akan dilakukan prediksi dan evaluasi pada data *testing* dengan menggunakan model hasil hasil pembelajaran tersebut. Tabel hasil prediksi dengan menggunakan fungsi `confusion_matrix` dapat dilihat pada Tabel 4.5. Berikut adalah nilai akurasi, presisi, dan recall untuk tiap kelas dengan menggunakan fungsi `classification_report`.

|           | precision | recall | f1-score | support |
|-----------|-----------|--------|----------|---------|
| a         | 0.98      | 1.00   | 0.99     | 300     |
| b         | 0.91      | 0.93   | 0.92     | 300     |
| c         | 1.00      | 1.00   | 1.00     | 300     |
| d         | 0.96      | 0.73   | 0.83     | 300     |
| e         | 0.99      | 1.00   | 1.00     | 300     |
| f         | 0.88      | 0.99   | 0.93     | 300     |
| g         | 1.00      | 0.96   | 0.98     | 300     |
| h         | 0.98      | 1.00   | 0.99     | 300     |
| i         | 0.98      | 0.94   | 0.96     | 300     |
| j         | 1.00      | 1.00   | 1.00     | 300     |
| k         | 0.95      | 0.97   | 0.96     | 300     |
| l         | 0.97      | 0.94   | 0.96     | 300     |
| m         | 1.00      | 1.00   | 1.00     | 300     |
| n         | 1.00      | 1.00   | 1.00     | 300     |
| o         | 0.98      | 1.00   | 0.99     | 300     |
| p         | 1.00      | 1.00   | 1.00     | 300     |
| q         | 1.00      | 1.00   | 1.00     | 300     |
| r         | 0.97      | 0.95   | 0.96     | 300     |
| s         | 1.00      | 1.00   | 1.00     | 300     |
| t         | 1.00      | 1.00   | 1.00     | 300     |
| u         | 1.00      | 1.00   | 1.00     | 300     |
| v         | 0.86      | 0.95   | 0.90     | 300     |
| w         | 0.94      | 0.96   | 0.95     | 300     |
| x         | 1.00      | 1.00   | 1.00     | 300     |
| y         | 1.00      | 1.00   | 1.00     | 300     |
| z         | 1.00      | 1.00   | 1.00     | 300     |
| accuracy  |           |        | 0.97     | 7800    |
| macro avg | 0.97      | 0.97   | 0.97     | 7800    |

```
weighted avg      0.97      0.97      0.97      7800
Accuracy : [0.9737179487179487]
```

Nilai akurasi yang didapatkan dari fungsi *classification\_report* adalah 0.97 atau 97%, dan setelah dilakukan pengambilan nilai akurasi yang sebenarnya dari *dictionary* hasil *classification\_report*, akurasi yang didapatkan adalah 0.9737179487179487 atau 97.37%.

### 4.5.3 Skenario 3

Pada skenario 3, akan dilakukan ujicoba dengan presenase pembagian data *training* dan *testing* 80:20, pembagian data *training* dan *validation* 80:20, dengan epoch 20 untuk pembelajaran dalam pembuatan model CNNnya. Berikut adalah hasil pengujian dari skenario 3.

#### 4.5.3.1 Hasil Pembelajaran Sistem *Convex Hull* + CNN

Setelah struktur model dibuat, akan dilakukan proses kompilasi dan memulai proses pembelajaran terhadap data *TrainingPP* dengan menggunakan struktur model yang telah dibuat. Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch.

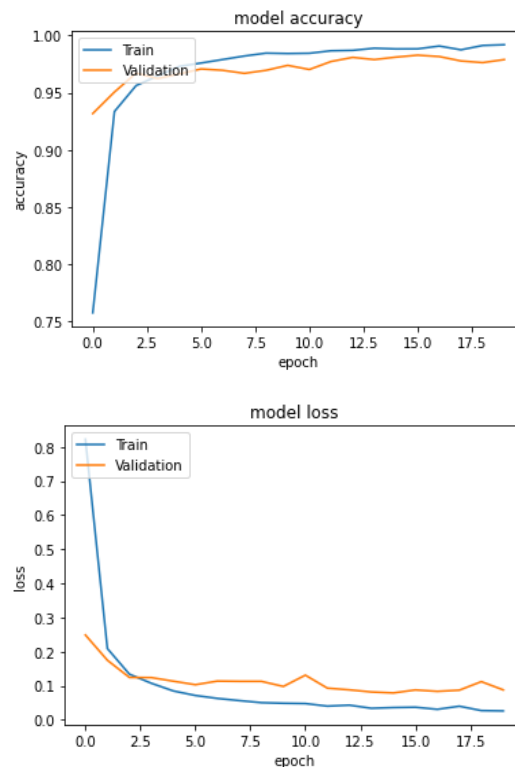
```
Epoch 1/20
520/520 [=====] - 97s 168ms/step - loss: 0.8225
- accuracy: 0.7577 - val_loss: 0.2487 - val_accuracy: 0.9317
Epoch 2/20
520/520 [=====] - 86s 165ms/step - loss: 0.2089
- accuracy: 0.9335 - val_loss: 0.1747 - val_accuracy: 0.9505
Epoch 3/20
520/520 [=====] - 86s 166ms/step - loss: 0.1339
- accuracy: 0.9561 - val_loss: 0.1243 - val_accuracy: 0.9671
Epoch 4/20
520/520 [=====] - 85s 164ms/step - loss: 0.1072
- accuracy: 0.9646 - val_loss: 0.1238 - val_accuracy: 0.9623
Epoch 5/20
520/520 [=====] - 85s 164ms/step - loss: 0.0847
- accuracy: 0.9730 - val_loss: 0.1132 - val_accuracy: 0.9663
Epoch 6/20
520/520 [=====] - 85s 164ms/step - loss: 0.0714
- accuracy: 0.9758 - val_loss: 0.1030 - val_accuracy: 0.9707
Epoch 7/20
520/520 [=====] - 86s 165ms/step - loss: 0.0627
- accuracy: 0.9789 - val_loss: 0.1136 - val_accuracy: 0.9695
Epoch 8/20
520/520 [=====] - 86s 166ms/step - loss: 0.0563
- accuracy: 0.9820 - val_loss: 0.1129 - val_accuracy: 0.9668
Epoch 9/20
520/520 [=====] - 85s 164ms/step - loss: 0.0503
- accuracy: 0.9845 - val_loss: 0.1128 - val_accuracy: 0.9695
Epoch 10/20
520/520 [=====] - 86s 165ms/step - loss: 0.0486
- accuracy: 0.9841 - val_loss: 0.0979 - val_accuracy: 0.9738
Epoch 11/20
520/520 [=====] - 85s 164ms/step - loss: 0.0477
- accuracy: 0.9844 - val_loss: 0.1310 - val_accuracy: 0.9702
Epoch 12/20
```

```

520/520 [=====] - 85s 164ms/step - loss: 0.0402
- accuracy: 0.9865 - val_loss: 0.0929 - val_accuracy: 0.9772
Epoch 13/20
520/520 [=====] - 86s 166ms/step - loss: 0.0430
- accuracy: 0.9869 - val_loss: 0.0879 - val_accuracy: 0.9808
Epoch 14/20
520/520 [=====] - 86s 166ms/step - loss: 0.0338
- accuracy: 0.9888 - val_loss: 0.0817 - val_accuracy: 0.9788
Epoch 15/20
520/520 [=====] - 86s 166ms/step - loss: 0.0360
- accuracy: 0.9882 - val_loss: 0.0788 - val_accuracy: 0.9810
Epoch 16/20
520/520 [=====] - 85s 164ms/step - loss: 0.0370
- accuracy: 0.9882 - val_loss: 0.0876 - val_accuracy: 0.9827
Epoch 17/20
520/520 [=====] - 86s 165ms/step - loss: 0.0308
- accuracy: 0.9907 - val_loss: 0.0835 - val_accuracy: 0.9815
Epoch 18/20
520/520 [=====] - 85s 164ms/step - loss: 0.0399
- accuracy: 0.9874 - val_loss: 0.0869 - val_accuracy: 0.9776
Epoch 19/20
520/520 [=====] - 85s 163ms/step - loss: 0.0271
- accuracy: 0.9912 - val_loss: 0.1120 - val_accuracy: 0.9762
Epoch 20/20
520/520 [=====] - 84s 161ms/step - loss: 0.0261
- accuracy: 0.9918 - val_loss: 0.0877 - val_accuracy: 0.9788

```

Jika proses hasil pembelajaran untuk tiap epoch diatas digambarkan dalam bentuk grafik hasilnya akan seperti pada Gambar 4.7.



**Gambar 4.7** Hasil *Output* Pembelajaran Sistem Dengan *Convex Hull* + CNN  
Pada Skenario 3

Setelah proses pembelajaran dilakukan, maka akan menghasilkan file model hasil pembelajaran, yaitu `modeln_json.json` dan `modeln.h5` yang nantinya akan digunakan untuk proses evaluasi.

#### 4.5.3.2 Hasil Evaluasi Sistem *Convex Hull* + CNN

Setelah file model hasil pembelajaran dihasilkan, selanjutnya akan dilakukan prediksi dan evaluasi pada data *testingPP* dengan menggunakan model hasil hasil pembelajaran tersebut. Tabel hasil prediksi dengan menggunakan fungsi `confusion_matrix` dapat dilihat pada Tabel 4.6. Untuk mendapatkan nilai akurasi, presisi, dan recall, maka akan dilakukan evaluasi menggunakan fungsi `classification_report`. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya.

|              | precision | recall | f1-score | support   |
|--------------|-----------|--------|----------|-----------|
| a            | 0.97      | 0.99   | 0.98     | 200       |
| b            | 0.98      | 0.96   | 0.97     | 200       |
| c            | 1.00      | 1.00   | 1.00     | 200       |
| d            | 0.98      | 0.96   | 0.97     | 200       |
| e            | 1.00      | 1.00   | 1.00     | 200       |
| f            | 0.97      | 0.99   | 0.98     | 200       |
| g            | 1.00      | 0.95   | 0.98     | 200       |
| h            | 1.00      | 1.00   | 1.00     | 200       |
| i            | 0.96      | 0.97   | 0.97     | 200       |
| j            | 1.00      | 1.00   | 1.00     | 200       |
| k            | 0.99      | 0.97   | 0.98     | 200       |
| l            | 0.97      | 0.99   | 0.98     | 200       |
| m            | 1.00      | 1.00   | 1.00     | 200       |
| n            | 1.00      | 1.00   | 1.00     | 200       |
| o            | 0.99      | 0.97   | 0.98     | 200       |
| p            | 1.00      | 1.00   | 1.00     | 200       |
| q            | 1.00      | 1.00   | 1.00     | 200       |
| r            | 0.99      | 0.98   | 0.98     | 200       |
| s            | 1.00      | 1.00   | 1.00     | 200       |
| t            | 1.00      | 1.00   | 1.00     | 200       |
| u            | 1.00      | 1.00   | 1.00     | 200       |
| v            | 0.96      | 0.97   | 0.97     | 200       |
| w            | 0.98      | 0.98   | 0.98     | 200       |
| x            | 1.00      | 1.00   | 1.00     | 200       |
| y            | 1.00      | 1.00   | 1.00     | 200       |
| z            | 1.00      | 1.00   | 1.00     | 200       |
| accuracy     |           |        |          | 0.99 5200 |
| macro avg    |           |        |          | 0.99 5200 |
| weighted avg |           |        |          | 0.99 5200 |

Accuracy : [0.9890384615384615]

Nilai akurasi yang didapatkan dari fungsi `classification_report` adalah 0.99 atau 99% namun nilai akurasi ini adalah nilai yang dibulatkan, dan setelah dilakukan pengambilan nilai akurasi yang sebenarnya dari `dictionary` hasil `classification_report`, akurasi yang didapatkan adalah 0.9890384615384615 atau 98.90%.

#### 4.5.3.3 Hasil Pembelajaran Sistem CNN Tanpa *Convex Hull*

Setelah sebelumnya proses pembelajaran untuk sistem *Convex Hull* + CNN, selanjutnya akan dilakukan pembelajaran untuk sistem CNN tanpa *Convex Hull* dengan menggunakan struktur model CNN terhadap data *training*. Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch.

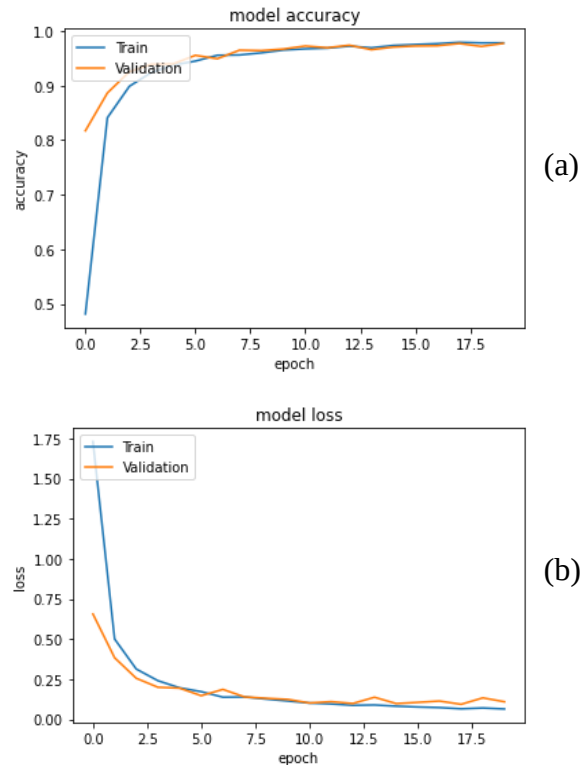
```
Epoch 1/20
520/520 [=====] - 86s 164ms/step - loss: 1.7302
- accuracy: 0.4816 - val_loss: 0.6562 - val_accuracy: 0.8173
Epoch 2/20
520/520 [=====] - 82s 158ms/step - loss: 0.5008
- accuracy: 0.8412 - val_loss: 0.3841 - val_accuracy: 0.8865
Epoch 3/20
520/520 [=====] - 82s 158ms/step - loss: 0.3138
- accuracy: 0.8988 - val_loss: 0.2562 - val_accuracy: 0.9255
Epoch 4/20
520/520 [=====] - 83s 159ms/step - loss: 0.2411
- accuracy: 0.9241 - val_loss: 0.2003 - val_accuracy: 0.9392
Epoch 5/20
520/520 [=====] - 82s 158ms/step - loss: 0.1969
- accuracy: 0.9377 - val_loss: 0.1947 - val_accuracy: 0.9399
Epoch 6/20
520/520 [=====] - 82s 159ms/step - loss: 0.1729
- accuracy: 0.9450 - val_loss: 0.1477 - val_accuracy: 0.9555
Epoch 7/20
520/520 [=====] - 89s 172ms/step - loss: 0.1385
- accuracy: 0.9556 - val_loss: 0.1869 - val_accuracy: 0.9493
Epoch 8/20
520/520 [=====] - 82s 158ms/step - loss: 0.1397
- accuracy: 0.9562 - val_loss: 0.1402 - val_accuracy: 0.9649
Epoch 9/20
520/520 [=====] - 82s 158ms/step - loss: 0.1266
- accuracy: 0.9600 - val_loss: 0.1315 - val_accuracy: 0.9642
Epoch 10/20
520/520 [=====] - 82s 159ms/step - loss: 0.1144
- accuracy: 0.9652 - val_loss: 0.1247 - val_accuracy: 0.9671
Epoch 11/20
520/520 [=====] - 87s 168ms/step - loss: 0.1020
- accuracy: 0.9674 - val_loss: 0.1025 - val_accuracy: 0.9726
Epoch 12/20
520/520 [=====] - 82s 158ms/step - loss: 0.0965
- accuracy: 0.9687 - val_loss: 0.1106 - val_accuracy: 0.9695
Epoch 13/20
520/520 [=====] - 83s 159ms/step - loss: 0.0878
- accuracy: 0.9724 - val_loss: 0.0993 - val_accuracy: 0.9740
Epoch 14/20
520/520 [=====] - 87s 167ms/step - loss: 0.0900
- accuracy: 0.9694 - val_loss: 0.1380 - val_accuracy: 0.9659
Epoch 15/20
520/520 [=====] - 86s 166ms/step - loss: 0.0825
- accuracy: 0.9736 - val_loss: 0.0987 - val_accuracy: 0.9707
Epoch 16/20
520/520 [=====] - 83s 160ms/step - loss: 0.0774
- accuracy: 0.9754 - val_loss: 0.1062 - val_accuracy: 0.9726
Epoch 17/20
520/520 [=====] - 85s 164ms/step - loss: 0.0737
- accuracy: 0.9769 - val_loss: 0.1143 - val_accuracy: 0.9731
Epoch 18/20
520/520 [=====] - 85s 164ms/step - loss: 0.0661
- accuracy: 0.9794 - val_loss: 0.0941 - val_accuracy: 0.9772
Epoch 19/20
```

```

520/520 [=====] - 84s 162ms/step - loss: 0.0712
- accuracy: 0.9782 - val_loss: 0.1342 - val_accuracy: 0.9719
Epoch 20/20
520/520 [=====] - 84s 161ms/step - loss: 0.0651
- accuracy: 0.9784 - val_loss: 0.1099 - val_accuracy: 0.9774

```

Jika proses hasil pembelajaran untuk tiap epoch diatas digambarkan dalam bentuk grafik hasilnya akan seperti pada Gambar 4.8.



**Gambar 4.8** (a) dan (b) Hasil *Output* Pembelajaran Sistem CNN Tanpa *Convex Hull* Pada Skenario 3

Setelah proses pembelajaran dilakukan, maka akan menghasilkan file model hasil pembelajaran, yaitu `modeln_json.json` dan `modeln.h5` yang nantinya akan digunakan untuk proses evaluasi.

#### 4.5.2.4 Hasil Evaluasi Sistem CNN Tanpa *Convex Hull*

Setelah file model hasil pembelajaran dihasilkan, selanjutnya akan dilakukan prediksi dan evaluasi pada data *testing* dengan menggunakan model hasil hasil pembelajaran tersebut. Untuk tabel hasil prediksi dengan menggunakan fungsi `confusion_matrix` bisa dilihat pada Tabel 4.7. Berikut adalah nilai akurasi, presisi, dan recall untuk tiap kelas dengan menggunakan fungsi `classification_report`.

|   | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| a | 0.96      | 1.00   | 0.98     | 200     |

|              |      |      |      |      |
|--------------|------|------|------|------|
| b            | 0.96 | 0.81 | 0.88 | 200  |
| c            | 1.00 | 1.00 | 1.00 | 200  |
| d            | 0.95 | 0.95 | 0.95 | 200  |
| e            | 0.99 | 0.99 | 0.99 | 200  |
| f            | 0.96 | 0.98 | 0.97 | 200  |
| g            | 1.00 | 0.92 | 0.96 | 200  |
| h            | 0.98 | 1.00 | 0.99 | 200  |
| i            | 0.99 | 0.94 | 0.96 | 200  |
| j            | 1.00 | 1.00 | 1.00 | 200  |
| k            | 0.98 | 0.95 | 0.97 | 200  |
| l            | 0.97 | 0.96 | 0.97 | 200  |
| m            | 1.00 | 1.00 | 1.00 | 200  |
| n            | 1.00 | 1.00 | 1.00 | 200  |
| o            | 0.99 | 0.99 | 0.99 | 200  |
| p            | 1.00 | 1.00 | 1.00 | 200  |
| q            | 1.00 | 1.00 | 1.00 | 200  |
| r            | 0.95 | 0.97 | 0.96 | 200  |
| s            | 1.00 | 1.00 | 1.00 | 200  |
| t            | 1.00 | 1.00 | 1.00 | 200  |
| u            | 1.00 | 1.00 | 1.00 | 200  |
| v            | 0.93 | 1.00 | 0.96 | 200  |
| w            | 0.88 | 0.97 | 0.93 | 200  |
| x            | 0.97 | 1.00 | 0.98 | 200  |
| y            | 1.00 | 1.00 | 1.00 | 200  |
| z            | 1.00 | 1.00 | 1.00 | 200  |
| accuracy     |      |      | 0.98 | 5200 |
| macro avg    | 0.98 | 0.98 | 0.98 | 5200 |
| weighted avg | 0.98 | 0.98 | 0.98 | 5200 |

Accuracy : [0.9788461538461538]

Nilai akurasi yang didapatkan dari fungsi *classification\_report* adalah 0.98 atau 98%, dan setelah dilakukan pengambilan nilai akurasi yang sebenarnya dari *dictionary* hasil *classification\_report*, akurasi yang didapatkan adalah 0.9788461538461538 atau 97.88%.

#### 4.5.4 Skenario 4

Setelah dilakukan ujicoba pada 3 skenario diatas, didapatkan skenario dengan tingkat akurasi tertinggi adalah skenario ke 3 dengan pembagian data 80:20 untuk data *training* dan testing. Pada skenario 4, akan dilakukan 5 kali ujicoba dengan presenase pembagian data *training* dan *testing* 80:20, pembagian data *training* dan *validation* 80:20, dengan variasi epoch 20, 40, 60, 80, dan 100 untuk pembelajaran dalam pembuatan model CNNnya. Berikut adalah hasil pengujian dari skenario 4.

##### 4.5.4.1 Hasil Pembelajaran Sistem *Convex Hull* + CNN

Setelah struktur model dibuat, akan dilakukan proses kompilasi dan memulai proses pembelajaran terhadap data *TrainingPP* menggunakan struktur model yang telah dibuat dengan menggunakan variasi 20, 40, 60, 80, dan 100

epoch. Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 20 epoch.

```
Epoch 1/20
520/520 [=====] - 1501s 3s/step - loss: 0.8380 -
accuracy: 0.7556 - val_loss: 0.3377 - val_accuracy: 0.8947
Epoch 2/20
520/520 [=====] - 66s 126ms/step - loss: 0.2121
- accuracy: 0.9334 - val_loss: 0.2084 - val_accuracy: 0.9351
Epoch 3/20
520/520 [=====] - 65s 126ms/step - loss: 0.1264
- accuracy: 0.9582 - val_loss: 0.1163 - val_accuracy: 0.9695
Epoch 4/20
520/520 [=====] - 65s 126ms/step - loss: 0.0978
- accuracy: 0.9691 - val_loss: 0.1219 - val_accuracy: 0.9637
Epoch 5/20
520/520 [=====] - 65s 125ms/step - loss: 0.0824
- accuracy: 0.9732 - val_loss: 0.1111 - val_accuracy: 0.9704
Epoch 6/20
520/520 [=====] - 65s 126ms/step - loss: 0.0705
- accuracy: 0.9772 - val_loss: 0.0894 - val_accuracy: 0.9755
Epoch 7/20
520/520 [=====] - 65s 125ms/step - loss: 0.0598
- accuracy: 0.9798 - val_loss: 0.0957 - val_accuracy: 0.9752
Epoch 8/20
520/520 [=====] - 66s 126ms/step - loss: 0.0571
- accuracy: 0.9817 - val_loss: 0.1012 - val_accuracy: 0.9719
Epoch 9/20
520/520 [=====] - 65s 125ms/step - loss: 0.0565
- accuracy: 0.9828 - val_loss: 0.0956 - val_accuracy: 0.9774
Epoch 10/20
520/520 [=====] - 65s 125ms/step - loss: 0.0463
- accuracy: 0.9855 - val_loss: 0.0999 - val_accuracy: 0.9745
Epoch 11/20
520/520 [=====] - 65s 125ms/step - loss: 0.0474
- accuracy: 0.9852 - val_loss: 0.0834 - val_accuracy: 0.9774
Epoch 12/20
520/520 [=====] - 65s 125ms/step - loss: 0.0382
- accuracy: 0.9861 - val_loss: 0.0916 - val_accuracy: 0.9817
Epoch 13/20
520/520 [=====] - 65s 125ms/step - loss: 0.0393
- accuracy: 0.9877 - val_loss: 0.1046 - val_accuracy: 0.9776
Epoch 14/20
520/520 [=====] - 65s 125ms/step - loss: 0.0355
- accuracy: 0.9889 - val_loss: 0.1006 - val_accuracy: 0.9762
Epoch 15/20
520/520 [=====] - 65s 125ms/step - loss: 0.0424
- accuracy: 0.9871 - val_loss: 0.0824 - val_accuracy: 0.9781
Epoch 16/20
520/520 [=====] - 65s 125ms/step - loss: 0.0305
- accuracy: 0.9905 - val_loss: 0.0909 - val_accuracy: 0.9803
Epoch 17/20
520/520 [=====] - 65s 125ms/step - loss: 0.0325
- accuracy: 0.9895 - val_loss: 0.0848 - val_accuracy: 0.9832
Epoch 18/20
520/520 [=====] - 65s 126ms/step - loss: 0.0333
- accuracy: 0.9889 - val_loss: 0.0834 - val_accuracy: 0.9808
Epoch 19/20
520/520 [=====] - 65s 126ms/step - loss: 0.0352
- accuracy: 0.9889 - val_loss: 0.0720 - val_accuracy: 0.9817
Epoch 20/20
520/520 [=====] - 65s 125ms/step - loss: 0.0266
- accuracy: 0.9919 - val_loss: 0.1030 - val_accuracy: 0.9784
```



Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 40 epoch.

```
Epoch 1/40
520/520 [=====] - 67s 128ms/step - loss: 0.8718
- accuracy: 0.7491 - val_loss: 0.2849 - val_accuracy: 0.9219
Epoch 2/40
520/520 [=====] - 66s 127ms/step - loss: 0.2376
- accuracy: 0.9245 - val_loss: 0.1837 - val_accuracy: 0.9459
Epoch 3/40
520/520 [=====] - 66s 128ms/step - loss: 0.1527
- accuracy: 0.9506 - val_loss: 0.1328 - val_accuracy: 0.9615
Epoch 4/40
520/520 [=====] - 67s 128ms/step - loss: 0.1118
- accuracy: 0.9630 - val_loss: 0.1109 - val_accuracy: 0.9714
Epoch 5/40
520/520 [=====] - 66s 128ms/step - loss: 0.0933
- accuracy: 0.9692 - val_loss: 0.1294 - val_accuracy: 0.9651
Epoch 6/40
520/520 [=====] - 66s 127ms/step - loss: 0.0792
- accuracy: 0.9749 - val_loss: 0.0982 - val_accuracy: 0.9745
Epoch 7/40
520/520 [=====] - 66s 127ms/step - loss: 0.0680
- accuracy: 0.9771 - val_loss: 0.1014 - val_accuracy: 0.9776
Epoch 8/40
520/520 [=====] - 66s 127ms/step - loss: 0.0615
- accuracy: 0.9810 - val_loss: 0.1111 - val_accuracy: 0.9726
Epoch 9/40
520/520 [=====] - 66s 126ms/step - loss: 0.0560
- accuracy: 0.9816 - val_loss: 0.0874 - val_accuracy: 0.9755
Epoch 10/40
520/520 [=====] - 66s 126ms/step - loss: 0.0536
- accuracy: 0.9822 - val_loss: 0.0902 - val_accuracy: 0.9755
Epoch 11/40
520/520 [=====] - 66s 126ms/step - loss: 0.0484
- accuracy: 0.9839 - val_loss: 0.1000 - val_accuracy: 0.9767
Epoch 12/40
520/520 [=====] - 66s 127ms/step - loss: 0.0420
- accuracy: 0.9861 - val_loss: 0.1201 - val_accuracy: 0.9750
Epoch 13/40
520/520 [=====] - 66s 127ms/step - loss: 0.0429
- accuracy: 0.9870 - val_loss: 0.1134 - val_accuracy: 0.9719
Epoch 14/40
520/520 [=====] - 66s 127ms/step - loss: 0.0425
- accuracy: 0.9868 - val_loss: 0.1049 - val_accuracy: 0.9784
Epoch 15/40
520/520 [=====] - 66s 127ms/step - loss: 0.0406
- accuracy: 0.9873 - val_loss: 0.1140 - val_accuracy: 0.9702
Epoch 16/40
520/520 [=====] - 66s 126ms/step - loss: 0.0371
- accuracy: 0.9873 - val_loss: 0.0889 - val_accuracy: 0.9808
Epoch 17/40
520/520 [=====] - 65s 126ms/step - loss: 0.0319
- accuracy: 0.9892 - val_loss: 0.1141 - val_accuracy: 0.9772
Epoch 18/40
520/520 [=====] - 65s 126ms/step - loss: 0.0352
- accuracy: 0.9881 - val_loss: 0.1115 - val_accuracy: 0.9774
Epoch 19/40
520/520 [=====] - 66s 127ms/step - loss: 0.0336
- accuracy: 0.9889 - val_loss: 0.1144 - val_accuracy: 0.9784
Epoch 20/40
520/520 [=====] - 66s 127ms/step - loss: 0.0340
- accuracy: 0.9890 - val_loss: 0.0936 - val_accuracy: 0.9791
Epoch 21/40
```

```

520/520 [=====] - 66s 126ms/step - loss: 0.0271
- accuracy: 0.9911 - val_loss: 0.1225 - val_accuracy: 0.9736
Epoch 22/40
520/520 [=====] - 65s 126ms/step - loss: 0.0316
- accuracy: 0.9905 - val_loss: 0.0851 - val_accuracy: 0.9810
Epoch 23/40
520/520 [=====] - 65s 126ms/step - loss: 0.0321
- accuracy: 0.9895 - val_loss: 0.0851 - val_accuracy: 0.9786
Epoch 24/40
520/520 [=====] - 66s 127ms/step - loss: 0.0235
- accuracy: 0.9930 - val_loss: 0.0929 - val_accuracy: 0.9832
Epoch 25/40
520/520 [=====] - 66s 127ms/step - loss: 0.0313
- accuracy: 0.9903 - val_loss: 0.1226 - val_accuracy: 0.9772
Epoch 26/40
520/520 [=====] - 66s 126ms/step - loss: 0.0322
- accuracy: 0.9907 - val_loss: 0.1117 - val_accuracy: 0.9784
Epoch 27/40
520/520 [=====] - 65s 126ms/step - loss: 0.0252
- accuracy: 0.9926 - val_loss: 0.0920 - val_accuracy: 0.9825
Epoch 28/40
520/520 [=====] - 66s 126ms/step - loss: 0.0206
- accuracy: 0.9933 - val_loss: 0.1172 - val_accuracy: 0.9784
Epoch 29/40
520/520 [=====] - 66s 126ms/step - loss: 0.0263
- accuracy: 0.9916 - val_loss: 0.1192 - val_accuracy: 0.9810
Epoch 30/40
520/520 [=====] - 65s 126ms/step - loss: 0.0274
- accuracy: 0.9917 - val_loss: 0.1114 - val_accuracy: 0.9798
Epoch 31/40
520/520 [=====] - 65s 126ms/step - loss: 0.0258
- accuracy: 0.9922 - val_loss: 0.0894 - val_accuracy: 0.9834
Epoch 32/40
520/520 [=====] - 65s 126ms/step - loss: 0.0210
- accuracy: 0.9938 - val_loss: 0.1015 - val_accuracy: 0.9834
Epoch 33/40
520/520 [=====] - 66s 126ms/step - loss: 0.0317
- accuracy: 0.9907 - val_loss: 0.1199 - val_accuracy: 0.9772
Epoch 34/40
520/520 [=====] - 66s 127ms/step - loss: 0.0260
- accuracy: 0.9918 - val_loss: 0.0870 - val_accuracy: 0.9832
Epoch 35/40
520/520 [=====] - 66s 126ms/step - loss: 0.0246
- accuracy: 0.9922 - val_loss: 0.1063 - val_accuracy: 0.9781
Epoch 36/40
520/520 [=====] - 66s 127ms/step - loss: 0.0211
- accuracy: 0.9936 - val_loss: 0.1319 - val_accuracy: 0.9803
Epoch 37/40
520/520 [=====] - 65s 126ms/step - loss: 0.0237
- accuracy: 0.9936 - val_loss: 0.1124 - val_accuracy: 0.9815
Epoch 38/40
520/520 [=====] - 66s 126ms/step - loss: 0.0196
- accuracy: 0.9941 - val_loss: 0.1122 - val_accuracy: 0.9844
Epoch 39/40
520/520 [=====] - 65s 125ms/step - loss: 0.0293
- accuracy: 0.9922 - val_loss: 0.0927 - val_accuracy: 0.9829
Epoch 40/40
520/520 [=====] - 65s 126ms/step - loss: 0.0261
- accuracy: 0.9931 - val_loss: 0.0993 - val_accuracy: 0.9812

```

Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 60 epoch.

Epoch 1/60

```

520/520 [=====] - 67s 127ms/step - loss: 0.8971
- accuracy: 0.7342 - val_loss: 0.3166 - val_accuracy: 0.9137
Epoch 2/60
520/520 [=====] - 66s 127ms/step - loss: 0.2168
- accuracy: 0.9298 - val_loss: 0.1712 - val_accuracy: 0.9507
Epoch 3/60
520/520 [=====] - 66s 126ms/step - loss: 0.1334
- accuracy: 0.9566 - val_loss: 0.1371 - val_accuracy: 0.9596
Epoch 4/60
520/520 [=====] - 66s 126ms/step - loss: 0.1005
- accuracy: 0.9669 - val_loss: 0.1137 - val_accuracy: 0.9663
Epoch 5/60
520/520 [=====] - 66s 127ms/step - loss: 0.0815
- accuracy: 0.9736 - val_loss: 0.1090 - val_accuracy: 0.9707
Epoch 6/60
520/520 [=====] - 65s 126ms/step - loss: 0.0710
- accuracy: 0.9774 - val_loss: 0.1103 - val_accuracy: 0.9719
Epoch 7/60
520/520 [=====] - 66s 127ms/step - loss: 0.0612
- accuracy: 0.9796 - val_loss: 0.1242 - val_accuracy: 0.9647
Epoch 8/60
520/520 [=====] - 66s 127ms/step - loss: 0.0571
- accuracy: 0.9815 - val_loss: 0.1054 - val_accuracy: 0.9719
Epoch 9/60
520/520 [=====] - 66s 126ms/step - loss: 0.0537
- accuracy: 0.9814 - val_loss: 0.0945 - val_accuracy: 0.9738
Epoch 10/60
520/520 [=====] - 66s 126ms/step - loss: 0.0503
- accuracy: 0.9826 - val_loss: 0.1098 - val_accuracy: 0.9728
Epoch 11/60
520/520 [=====] - 66s 127ms/step - loss: 0.0421
- accuracy: 0.9862 - val_loss: 0.0876 - val_accuracy: 0.9793
Epoch 12/60
520/520 [=====] - 66s 127ms/step - loss: 0.0409
- accuracy: 0.9873 - val_loss: 0.1128 - val_accuracy: 0.9716
Epoch 13/60
520/520 [=====] - 66s 127ms/step - loss: 0.0413
- accuracy: 0.9864 - val_loss: 0.0861 - val_accuracy: 0.9796
Epoch 14/60
520/520 [=====] - 66s 127ms/step - loss: 0.0370
- accuracy: 0.9886 - val_loss: 0.0697 - val_accuracy: 0.9808
Epoch 15/60
520/520 [=====] - 66s 126ms/step - loss: 0.0293
- accuracy: 0.9898 - val_loss: 0.0815 - val_accuracy: 0.9832
Epoch 16/60
520/520 [=====] - 66s 127ms/step - loss: 0.0371
- accuracy: 0.9882 - val_loss: 0.0935 - val_accuracy: 0.9772
Epoch 17/60
520/520 [=====] - 66s 127ms/step - loss: 0.0324
- accuracy: 0.9900 - val_loss: 0.0913 - val_accuracy: 0.9800
Epoch 18/60
520/520 [=====] - 66s 127ms/step - loss: 0.0354
- accuracy: 0.9883 - val_loss: 0.0868 - val_accuracy: 0.9779
Epoch 19/60
520/520 [=====] - 65s 126ms/step - loss: 0.0260
- accuracy: 0.9922 - val_loss: 0.0699 - val_accuracy: 0.9839
Epoch 20/60
520/520 [=====] - 66s 126ms/step - loss: 0.0307
- accuracy: 0.9903 - val_loss: 0.0832 - val_accuracy: 0.9817
Epoch 21/60
520/520 [=====] - 65s 126ms/step - loss: 0.0268
- accuracy: 0.9918 - val_loss: 0.0808 - val_accuracy: 0.9827
Epoch 22/60
520/520 [=====] - 66s 127ms/step - loss: 0.0298
- accuracy: 0.9909 - val_loss: 0.0856 - val_accuracy: 0.9786

```

```

Epoch 23/60
520/520 [=====] - 66s 127ms/step - loss: 0.0280
- accuracy: 0.9915 - val_loss: 0.0909 - val_accuracy: 0.9808
Epoch 24/60
520/520 [=====] - 66s 126ms/step - loss: 0.0281
- accuracy: 0.9917 - val_loss: 0.0949 - val_accuracy: 0.9808
Epoch 25/60
520/520 [=====] - 66s 127ms/step - loss: 0.0244
- accuracy: 0.9925 - val_loss: 0.0957 - val_accuracy: 0.9812
Epoch 26/60
520/520 [=====] - 66s 127ms/step - loss: 0.0235
- accuracy: 0.9928 - val_loss: 0.1072 - val_accuracy: 0.9776
Epoch 27/60
520/520 [=====] - 66s 127ms/step - loss: 0.0221
- accuracy: 0.9931 - val_loss: 0.0918 - val_accuracy: 0.9776
Epoch 28/60
520/520 [=====] - 66s 127ms/step - loss: 0.0205
- accuracy: 0.9936 - val_loss: 0.0989 - val_accuracy: 0.9808
Epoch 29/60
520/520 [=====] - 67s 128ms/step - loss: 0.0292
- accuracy: 0.9910 - val_loss: 0.0916 - val_accuracy: 0.9815
Epoch 30/60
520/520 [=====] - 66s 127ms/step - loss: 0.0217
- accuracy: 0.9928 - val_loss: 0.1040 - val_accuracy: 0.9793
Epoch 31/60
520/520 [=====] - 66s 128ms/step - loss: 0.0234
- accuracy: 0.9934 - val_loss: 0.1078 - val_accuracy: 0.9805
Epoch 32/60
520/520 [=====] - 66s 127ms/step - loss: 0.0224
- accuracy: 0.9929 - val_loss: 0.0972 - val_accuracy: 0.9825
Epoch 33/60
520/520 [=====] - 65s 126ms/step - loss: 0.0211
- accuracy: 0.9940 - val_loss: 0.0997 - val_accuracy: 0.9803
Epoch 34/60
520/520 [=====] - 67s 128ms/step - loss: 0.0256
- accuracy: 0.9932 - val_loss: 0.1000 - val_accuracy: 0.9788
Epoch 35/60
520/520 [=====] - 66s 127ms/step - loss: 0.0235
- accuracy: 0.9933 - val_loss: 0.0732 - val_accuracy: 0.9837
Epoch 36/60
520/520 [=====] - 68s 130ms/step - loss: 0.0200
- accuracy: 0.9946 - val_loss: 0.0729 - val_accuracy: 0.9841
Epoch 37/60
520/520 [=====] - 66s 127ms/step - loss: 0.0234
- accuracy: 0.9931 - val_loss: 0.0943 - val_accuracy: 0.9817
Epoch 38/60
520/520 [=====] - 67s 129ms/step - loss: 0.0203
- accuracy: 0.9941 - val_loss: 0.0960 - val_accuracy: 0.9834
Epoch 39/60
520/520 [=====] - 67s 128ms/step - loss: 0.0204
- accuracy: 0.9933 - val_loss: 0.1163 - val_accuracy: 0.9837
Epoch 40/60
520/520 [=====] - 67s 129ms/step - loss: 0.0256
- accuracy: 0.9931 - val_loss: 0.1574 - val_accuracy: 0.9738
Epoch 41/60
520/520 [=====] - 66s 128ms/step - loss: 0.0261
- accuracy: 0.9930 - val_loss: 0.0839 - val_accuracy: 0.9839
Epoch 42/60
520/520 [=====] - 66s 128ms/step - loss: 0.0134
- accuracy: 0.9959 - val_loss: 0.1425 - val_accuracy: 0.9779
Epoch 43/60
520/520 [=====] - 66s 128ms/step - loss: 0.0201
- accuracy: 0.9933 - val_loss: 0.0936 - val_accuracy: 0.9837
Epoch 44/60

```

```

520/520 [=====] - 67s 128ms/step - loss: 0.0211
- accuracy: 0.9937 - val_loss: 0.0962 - val_accuracy: 0.9856
Epoch 45/60
520/520 [=====] - 66s 128ms/step - loss: 0.0210
- accuracy: 0.9940 - val_loss: 0.1065 - val_accuracy: 0.9837
Epoch 46/60
520/520 [=====] - 66s 127ms/step - loss: 0.0186
- accuracy: 0.9950 - val_loss: 0.1183 - val_accuracy: 0.9829
Epoch 47/60
520/520 [=====] - 66s 127ms/step - loss: 0.0152
- accuracy: 0.9955 - val_loss: 0.0991 - val_accuracy: 0.9820
Epoch 48/60
520/520 [=====] - 66s 127ms/step - loss: 0.0207
- accuracy: 0.9947 - val_loss: 0.0866 - val_accuracy: 0.9849
Epoch 49/60
520/520 [=====] - 66s 127ms/step - loss: 0.0167
- accuracy: 0.9952 - val_loss: 0.1248 - val_accuracy: 0.9793
Epoch 50/60
520/520 [=====] - 67s 129ms/step - loss: 0.0243
- accuracy: 0.9944 - val_loss: 0.1156 - val_accuracy: 0.9800
Epoch 51/60
520/520 [=====] - 66s 128ms/step - loss: 0.0222
- accuracy: 0.9945 - val_loss: 0.1107 - val_accuracy: 0.9817
Epoch 52/60
520/520 [=====] - 66s 127ms/step - loss: 0.0215
- accuracy: 0.9951 - val_loss: 0.0921 - val_accuracy: 0.9820
Epoch 53/60
520/520 [=====] - 66s 127ms/step - loss: 0.0248
- accuracy: 0.9936 - val_loss: 0.1129 - val_accuracy: 0.9817
Epoch 54/60
520/520 [=====] - 66s 128ms/step - loss: 0.0150
- accuracy: 0.9956 - val_loss: 0.1120 - val_accuracy: 0.9853
Epoch 55/60
520/520 [=====] - 67s 128ms/step - loss: 0.0167
- accuracy: 0.9951 - val_loss: 0.0966 - val_accuracy: 0.9865
Epoch 56/60
520/520 [=====] - 66s 126ms/step - loss: 0.0186
- accuracy: 0.9940 - val_loss: 0.1065 - val_accuracy: 0.9844
Epoch 57/60
520/520 [=====] - 66s 128ms/step - loss: 0.0182
- accuracy: 0.9951 - val_loss: 0.0911 - val_accuracy: 0.9853
Epoch 58/60
520/520 [=====] - 66s 127ms/step - loss: 0.0206
- accuracy: 0.9951 - val_loss: 0.0949 - val_accuracy: 0.9856
Epoch 59/60
520/520 [=====] - 66s 127ms/step - loss: 0.0162
- accuracy: 0.9958 - val_loss: 0.0943 - val_accuracy: 0.9853
Epoch 60/60
520/520 [=====] - 66s 127ms/step - loss: 0.0237
- accuracy: 0.9948 - val_loss: 0.0899 - val_accuracy: 0.9868

```

Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 80 epoch.

```

Epoch 1/80
520/520 [=====] - 67s 127ms/step - loss: 0.9070
- accuracy: 0.7371 - val_loss: 0.3275 - val_accuracy: 0.9000
Epoch 2/80
520/520 [=====] - 66s 127ms/step - loss: 0.2157
- accuracy: 0.9323 - val_loss: 0.1579 - val_accuracy: 0.9529
Epoch 3/80
520/520 [=====] - 66s 127ms/step - loss: 0.1368
- accuracy: 0.9572 - val_loss: 0.1545 - val_accuracy: 0.9575
Epoch 4/80

```

```

520/520 [=====] - 67s 129ms/step - loss: 0.1082
- accuracy: 0.9638 - val_loss: 0.1127 - val_accuracy: 0.9685
Epoch 5/80
520/520 [=====] - 66s 127ms/step - loss: 0.0825
- accuracy: 0.9733 - val_loss: 0.1171 - val_accuracy: 0.9709
Epoch 6/80
520/520 [=====] - 66s 126ms/step - loss: 0.0744
- accuracy: 0.9750 - val_loss: 0.1230 - val_accuracy: 0.9675
Epoch 7/80
520/520 [=====] - 66s 127ms/step - loss: 0.0662
- accuracy: 0.9775 - val_loss: 0.1034 - val_accuracy: 0.9733
Epoch 8/80
520/520 [=====] - 67s 128ms/step - loss: 0.0593
- accuracy: 0.9801 - val_loss: 0.1256 - val_accuracy: 0.9714
Epoch 9/80
520/520 [=====] - 67s 128ms/step - loss: 0.0481
- accuracy: 0.9856 - val_loss: 0.1218 - val_accuracy: 0.9733
Epoch 10/80
520/520 [=====] - 66s 127ms/step - loss: 0.0481
- accuracy: 0.9839 - val_loss: 0.0986 - val_accuracy: 0.9776
Epoch 11/80
520/520 [=====] - 66s 127ms/step - loss: 0.0442
- accuracy: 0.9858 - val_loss: 0.0995 - val_accuracy: 0.9750
Epoch 12/80
520/520 [=====] - 66s 127ms/step - loss: 0.0434
- accuracy: 0.9849 - val_loss: 0.1004 - val_accuracy: 0.9772
Epoch 13/80
520/520 [=====] - 66s 128ms/step - loss: 0.0420
- accuracy: 0.9873 - val_loss: 0.1031 - val_accuracy: 0.9743
Epoch 14/80
520/520 [=====] - 66s 127ms/step - loss: 0.0385
- accuracy: 0.9877 - val_loss: 0.0974 - val_accuracy: 0.9769
Epoch 15/80
520/520 [=====] - 66s 127ms/step - loss: 0.0385
- accuracy: 0.9878 - val_loss: 0.1033 - val_accuracy: 0.9769
Epoch 16/80
520/520 [=====] - 66s 128ms/step - loss: 0.0318
- accuracy: 0.9892 - val_loss: 0.0984 - val_accuracy: 0.9743
Epoch 17/80
520/520 [=====] - 66s 127ms/step - loss: 0.0285
- accuracy: 0.9904 - val_loss: 0.1013 - val_accuracy: 0.9796
Epoch 18/80
520/520 [=====] - 67s 128ms/step - loss: 0.0288
- accuracy: 0.9904 - val_loss: 0.0916 - val_accuracy: 0.9798
Epoch 19/80
520/520 [=====] - 66s 127ms/step - loss: 0.0316
- accuracy: 0.9907 - val_loss: 0.1223 - val_accuracy: 0.9748
Epoch 20/80
520/520 [=====] - 65s 126ms/step - loss: 0.0323
- accuracy: 0.9891 - val_loss: 0.1276 - val_accuracy: 0.9743
Epoch 21/80
520/520 [=====] - 66s 127ms/step - loss: 0.0329
- accuracy: 0.9896 - val_loss: 0.1113 - val_accuracy: 0.9762
Epoch 22/80
520/520 [=====] - 66s 127ms/step - loss: 0.0298
- accuracy: 0.9908 - val_loss: 0.1352 - val_accuracy: 0.9764
Epoch 23/80
520/520 [=====] - 65s 126ms/step - loss: 0.0260
- accuracy: 0.9922 - val_loss: 0.0956 - val_accuracy: 0.9825
Epoch 24/80
520/520 [=====] - 65s 125ms/step - loss: 0.0232
- accuracy: 0.9929 - val_loss: 0.1160 - val_accuracy: 0.9820
Epoch 25/80
520/520 [=====] - 66s 126ms/step - loss: 0.0328
- accuracy: 0.9896 - val_loss: 0.1130 - val_accuracy: 0.9776

```

```

Epoch 26/80
520/520 [=====] - 66s 126ms/step - loss: 0.0285
- accuracy: 0.9918 - val_loss: 0.0884 - val_accuracy: 0.9817
Epoch 27/80
520/520 [=====] - 66s 126ms/step - loss: 0.0271
- accuracy: 0.9916 - val_loss: 0.0909 - val_accuracy: 0.9820
Epoch 28/80
520/520 [=====] - 65s 126ms/step - loss: 0.0264
- accuracy: 0.9919 - val_loss: 0.0899 - val_accuracy: 0.9815
Epoch 29/80
520/520 [=====] - 65s 126ms/step - loss: 0.0216
- accuracy: 0.9937 - val_loss: 0.1004 - val_accuracy: 0.9784
Epoch 30/80
520/520 [=====] - 66s 127ms/step - loss: 0.0243
- accuracy: 0.9922 - val_loss: 0.0862 - val_accuracy: 0.9796
Epoch 31/80
520/520 [=====] - 65s 126ms/step - loss: 0.0257
- accuracy: 0.9928 - val_loss: 0.1205 - val_accuracy: 0.9774
Epoch 32/80
520/520 [=====] - 66s 127ms/step - loss: 0.0178
- accuracy: 0.9941 - val_loss: 0.1086 - val_accuracy: 0.9825
Epoch 33/80
520/520 [=====] - 65s 126ms/step - loss: 0.0250
- accuracy: 0.9916 - val_loss: 0.1099 - val_accuracy: 0.9858
Epoch 34/80
520/520 [=====] - 66s 126ms/step - loss: 0.0270
- accuracy: 0.9931 - val_loss: 0.1008 - val_accuracy: 0.9788
Epoch 35/80
520/520 [=====] - 65s 125ms/step - loss: 0.0182
- accuracy: 0.9941 - val_loss: 0.1032 - val_accuracy: 0.9822
Epoch 36/80
520/520 [=====] - 66s 127ms/step - loss: 0.0235
- accuracy: 0.9934 - val_loss: 0.1079 - val_accuracy: 0.9791
Epoch 37/80
520/520 [=====] - 66s 126ms/step - loss: 0.0208
- accuracy: 0.9933 - val_loss: 0.1111 - val_accuracy: 0.9817
Epoch 38/80
520/520 [=====] - 65s 126ms/step - loss: 0.0247
- accuracy: 0.9936 - val_loss: 0.0953 - val_accuracy: 0.9829
Epoch 39/80
520/520 [=====] - 66s 126ms/step - loss: 0.0259
- accuracy: 0.9929 - val_loss: 0.1072 - val_accuracy: 0.9837
Epoch 40/80
520/520 [=====] - 66s 126ms/step - loss: 0.0191
- accuracy: 0.9941 - val_loss: 0.0951 - val_accuracy: 0.9837
Epoch 41/80
520/520 [=====] - 66s 128ms/step - loss: 0.0179
- accuracy: 0.9950 - val_loss: 0.1193 - val_accuracy: 0.9825
Epoch 42/80
520/520 [=====] - 65s 126ms/step - loss: 0.0199
- accuracy: 0.9944 - val_loss: 0.0989 - val_accuracy: 0.9822
Epoch 43/80
520/520 [=====] - 66s 126ms/step - loss: 0.0204
- accuracy: 0.9944 - val_loss: 0.0787 - val_accuracy: 0.9851
Epoch 44/80
520/520 [=====] - 66s 127ms/step - loss: 0.0236
- accuracy: 0.9931 - val_loss: 0.1173 - val_accuracy: 0.9827
Epoch 45/80
520/520 [=====] - 66s 126ms/step - loss: 0.0208
- accuracy: 0.9934 - val_loss: 0.0958 - val_accuracy: 0.9834
Epoch 46/80
520/520 [=====] - 66s 127ms/step - loss: 0.0205
- accuracy: 0.9948 - val_loss: 0.1191 - val_accuracy: 0.9817
Epoch 47/80

```

```

520/520 [=====] - 66s 127ms/step - loss: 0.0243
- accuracy: 0.9928 - val_loss: 0.0921 - val_accuracy: 0.9846
Epoch 48/80
520/520 [=====] - 65s 126ms/step - loss: 0.0188
- accuracy: 0.9944 - val_loss: 0.1523 - val_accuracy: 0.9793
Epoch 49/80
520/520 [=====] - 66s 126ms/step - loss: 0.0164
- accuracy: 0.9954 - val_loss: 0.1207 - val_accuracy: 0.9798
Epoch 50/80
520/520 [=====] - 65s 125ms/step - loss: 0.0227
- accuracy: 0.9942 - val_loss: 0.1198 - val_accuracy: 0.9827
Epoch 51/80
520/520 [=====] - 66s 127ms/step - loss: 0.0184
- accuracy: 0.9947 - val_loss: 0.1307 - val_accuracy: 0.9800
Epoch 52/80
520/520 [=====] - 66s 127ms/step - loss: 0.0237
- accuracy: 0.9938 - val_loss: 0.1166 - val_accuracy: 0.9820
Epoch 53/80
520/520 [=====] - 65s 126ms/step - loss: 0.0214
- accuracy: 0.9938 - val_loss: 0.1322 - val_accuracy: 0.9815
Epoch 54/80
520/520 [=====] - 66s 127ms/step - loss: 0.0253
- accuracy: 0.9945 - val_loss: 0.1037 - val_accuracy: 0.9822
Epoch 55/80
520/520 [=====] - 66s 128ms/step - loss: 0.0200
- accuracy: 0.9947 - val_loss: 0.1211 - val_accuracy: 0.9822
Epoch 56/80
520/520 [=====] - 65s 126ms/step - loss: 0.0191
- accuracy: 0.9951 - val_loss: 0.1154 - val_accuracy: 0.9827
Epoch 57/80
520/520 [=====] - 66s 127ms/step - loss: 0.0181
- accuracy: 0.9950 - val_loss: 0.0937 - val_accuracy: 0.9851
Epoch 58/80
520/520 [=====] - 66s 127ms/step - loss: 0.0235
- accuracy: 0.9944 - val_loss: 0.1197 - val_accuracy: 0.9844
Epoch 59/80
520/520 [=====] - 66s 126ms/step - loss: 0.0168
- accuracy: 0.9946 - val_loss: 0.0836 - val_accuracy: 0.9834
Epoch 60/80
520/520 [=====] - 66s 128ms/step - loss: 0.0180
- accuracy: 0.9952 - val_loss: 0.1097 - val_accuracy: 0.9839
Epoch 61/80
520/520 [=====] - 65s 126ms/step - loss: 0.0121
- accuracy: 0.9974 - val_loss: 0.1120 - val_accuracy: 0.9815
Epoch 62/80
520/520 [=====] - 65s 126ms/step - loss: 0.0219
- accuracy: 0.9944 - val_loss: 0.1255 - val_accuracy: 0.9834
Epoch 63/80
520/520 [=====] - 66s 127ms/step - loss: 0.0251
- accuracy: 0.9943 - val_loss: 0.1011 - val_accuracy: 0.9822
Epoch 64/80
520/520 [=====] - 66s 126ms/step - loss: 0.0173
- accuracy: 0.9959 - val_loss: 0.1096 - val_accuracy: 0.9839
Epoch 65/80
520/520 [=====] - 66s 127ms/step - loss: 0.0165
- accuracy: 0.9956 - val_loss: 0.1280 - val_accuracy: 0.9817
Epoch 66/80
520/520 [=====] - 66s 126ms/step - loss: 0.0276
- accuracy: 0.9939 - val_loss: 0.1209 - val_accuracy: 0.9841
Epoch 67/80
520/520 [=====] - 66s 127ms/step - loss: 0.0180
- accuracy: 0.9951 - val_loss: 0.1402 - val_accuracy: 0.9846
Epoch 68/80
520/520 [=====] - 66s 128ms/step - loss: 0.0245
- accuracy: 0.9945 - val_loss: 0.1070 - val_accuracy: 0.9837

```



```

Epoch 69/80
520/520 [=====] - 66s 127ms/step - loss: 0.0175
- accuracy: 0.9963 - val_loss: 0.1555 - val_accuracy: 0.9796
Epoch 70/80
520/520 [=====] - 66s 127ms/step - loss: 0.0177
- accuracy: 0.9960 - val_loss: 0.1495 - val_accuracy: 0.9812
Epoch 71/80
520/520 [=====] - 66s 128ms/step - loss: 0.0178
- accuracy: 0.9963 - val_loss: 0.1588 - val_accuracy: 0.9808
Epoch 72/80
520/520 [=====] - 67s 128ms/step - loss: 0.0281
- accuracy: 0.9937 - val_loss: 0.1350 - val_accuracy: 0.9810
Epoch 73/80
520/520 [=====] - 66s 127ms/step - loss: 0.0239
- accuracy: 0.9948 - val_loss: 0.1367 - val_accuracy: 0.9844
Epoch 74/80
520/520 [=====] - 67s 129ms/step - loss: 0.0150
- accuracy: 0.9966 - val_loss: 0.0945 - val_accuracy: 0.9877
Epoch 75/80
520/520 [=====] - 66s 127ms/step - loss: 0.0194
- accuracy: 0.9956 - val_loss: 0.1316 - val_accuracy: 0.9841
Epoch 76/80
520/520 [=====] - 67s 128ms/step - loss: 0.0229
- accuracy: 0.9950 - val_loss: 0.1493 - val_accuracy: 0.9858
Epoch 77/80
520/520 [=====] - 66s 127ms/step - loss: 0.0172
- accuracy: 0.9956 - val_loss: 0.1322 - val_accuracy: 0.9861
Epoch 78/80
520/520 [=====] - 66s 127ms/step - loss: 0.0217
- accuracy: 0.9953 - val_loss: 0.0927 - val_accuracy: 0.9834
Epoch 79/80
520/520 [=====] - 66s 127ms/step - loss: 0.0218
- accuracy: 0.9946 - val_loss: 0.1150 - val_accuracy: 0.9853
Epoch 80/80
520/520 [=====] - 66s 126ms/step - loss: 0.0209
- accuracy: 0.9959 - val_loss: 0.1122 - val_accuracy: 0.9839

```

Dan berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 100 epoch.

```

Epoch 1/100
520/520 [=====] - 66s 126ms/step - loss: 0.8566
- accuracy: 0.7498 - val_loss: 0.2727 - val_accuracy: 0.9168
Epoch 2/100
520/520 [=====] - 66s 126ms/step - loss: 0.2163
- accuracy: 0.9340 - val_loss: 0.2026 - val_accuracy: 0.9363
Epoch 3/100
520/520 [=====] - 65s 126ms/step - loss: 0.1291
- accuracy: 0.9590 - val_loss: 0.1221 - val_accuracy: 0.9666
Epoch 4/100
520/520 [=====] - 66s 126ms/step - loss: 0.1005
- accuracy: 0.9671 - val_loss: 0.1238 - val_accuracy: 0.9661
Epoch 5/100
520/520 [=====] - 66s 127ms/step - loss: 0.0850
- accuracy: 0.9709 - val_loss: 0.1137 - val_accuracy: 0.9692
Epoch 6/100
520/520 [=====] - 66s 127ms/step - loss: 0.0739
- accuracy: 0.9756 - val_loss: 0.1199 - val_accuracy: 0.9690
Epoch 7/100
520/520 [=====] - 66s 127ms/step - loss: 0.0593
- accuracy: 0.9808 - val_loss: 0.0935 - val_accuracy: 0.9750
Epoch 8/100
520/520 [=====] - 66s 127ms/step - loss: 0.0605
- accuracy: 0.9796 - val_loss: 0.0933 - val_accuracy: 0.9757

```

```

Epoch 9/100
520/520 [=====] - 66s 127ms/step - loss: 0.0565
- accuracy: 0.9819 - val_loss: 0.1291 - val_accuracy: 0.9632
Epoch 10/100
520/520 [=====] - 66s 126ms/step - loss: 0.0486
- accuracy: 0.9840 - val_loss: 0.0934 - val_accuracy: 0.9757
Epoch 11/100
520/520 [=====] - 66s 127ms/step - loss: 0.0459
- accuracy: 0.9845 - val_loss: 0.0891 - val_accuracy: 0.9776
Epoch 12/100
520/520 [=====] - 66s 127ms/step - loss: 0.0416
- accuracy: 0.9867 - val_loss: 0.0881 - val_accuracy: 0.9774
Epoch 13/100
520/520 [=====] - 66s 128ms/step - loss: 0.0390
- accuracy: 0.9870 - val_loss: 0.1373 - val_accuracy: 0.9688
Epoch 14/100
520/520 [=====] - 66s 127ms/step - loss: 0.0399
- accuracy: 0.9876 - val_loss: 0.0999 - val_accuracy: 0.9784
Epoch 15/100
520/520 [=====] - 66s 127ms/step - loss: 0.0322
- accuracy: 0.9894 - val_loss: 0.0803 - val_accuracy: 0.9791
Epoch 16/100
520/520 [=====] - 66s 126ms/step - loss: 0.0347
- accuracy: 0.9880 - val_loss: 0.0973 - val_accuracy: 0.9776
Epoch 17/100
520/520 [=====] - 67s 128ms/step - loss: 0.0340
- accuracy: 0.9905 - val_loss: 0.0827 - val_accuracy: 0.9781
Epoch 18/100
520/520 [=====] - 67s 128ms/step - loss: 0.0354
- accuracy: 0.9880 - val_loss: 0.0867 - val_accuracy: 0.9803
Epoch 19/100
520/520 [=====] - 66s 126ms/step - loss: 0.0259
- accuracy: 0.9919 - val_loss: 0.0843 - val_accuracy: 0.9825
Epoch 20/100
520/520 [=====] - 66s 127ms/step - loss: 0.0234
- accuracy: 0.9930 - val_loss: 0.1133 - val_accuracy: 0.9772
Epoch 21/100
520/520 [=====] - 66s 126ms/step - loss: 0.0273
- accuracy: 0.9919 - val_loss: 0.1079 - val_accuracy: 0.9767
Epoch 22/100
520/520 [=====] - 66s 127ms/step - loss: 0.0309
- accuracy: 0.9907 - val_loss: 0.0950 - val_accuracy: 0.9798
Epoch 23/100
520/520 [=====] - 66s 126ms/step - loss: 0.0266
- accuracy: 0.9917 - val_loss: 0.0884 - val_accuracy: 0.9827
Epoch 24/100
520/520 [=====] - 66s 126ms/step - loss: 0.0314
- accuracy: 0.9904 - val_loss: 0.0995 - val_accuracy: 0.9810
Epoch 25/100
520/520 [=====] - 65s 126ms/step - loss: 0.0213
- accuracy: 0.9934 - val_loss: 0.0936 - val_accuracy: 0.9815
Epoch 26/100
520/520 [=====] - 66s 127ms/step - loss: 0.0301
- accuracy: 0.9910 - val_loss: 0.0962 - val_accuracy: 0.9781
Epoch 27/100
520/520 [=====] - 67s 129ms/step - loss: 0.0220
- accuracy: 0.9936 - val_loss: 0.0900 - val_accuracy: 0.9815
Epoch 28/100
520/520 [=====] - 65s 126ms/step - loss: 0.0248
- accuracy: 0.9931 - val_loss: 0.0865 - val_accuracy: 0.9834
Epoch 29/100
520/520 [=====] - 66s 127ms/step - loss: 0.0256
- accuracy: 0.9925 - val_loss: 0.0895 - val_accuracy: 0.9815
Epoch 30/100

```

```

520/520 [=====] - 66s 127ms/step - loss: 0.0270
- accuracy: 0.9918 - val_loss: 0.0936 - val_accuracy: 0.9832
Epoch 31/100
520/520 [=====] - 66s 127ms/step - loss: 0.0251
- accuracy: 0.9925 - val_loss: 0.0933 - val_accuracy: 0.9798
Epoch 32/100
520/520 [=====] - 66s 128ms/step - loss: 0.0246
- accuracy: 0.9935 - val_loss: 0.1232 - val_accuracy: 0.9784
Epoch 33/100
520/520 [=====] - 66s 127ms/step - loss: 0.0252
- accuracy: 0.9927 - val_loss: 0.1072 - val_accuracy: 0.9839
Epoch 34/100
520/520 [=====] - 66s 126ms/step - loss: 0.0204
- accuracy: 0.9943 - val_loss: 0.0935 - val_accuracy: 0.9868
Epoch 35/100
520/520 [=====] - 66s 126ms/step - loss: 0.0237
- accuracy: 0.9932 - val_loss: 0.1099 - val_accuracy: 0.9791
Epoch 36/100
520/520 [=====] - 66s 127ms/step - loss: 0.0255
- accuracy: 0.9928 - val_loss: 0.0845 - val_accuracy: 0.9829
Epoch 37/100
520/520 [=====] - 65s 125ms/step - loss: 0.0181
- accuracy: 0.9951 - val_loss: 0.1111 - val_accuracy: 0.9815
Epoch 38/100
520/520 [=====] - 66s 127ms/step - loss: 0.0209
- accuracy: 0.9937 - val_loss: 0.1253 - val_accuracy: 0.9784
Epoch 39/100
520/520 [=====] - 66s 127ms/step - loss: 0.0254
- accuracy: 0.9922 - val_loss: 0.0925 - val_accuracy: 0.9844
Epoch 40/100
520/520 [=====] - 66s 126ms/step - loss: 0.0212
- accuracy: 0.9938 - val_loss: 0.1100 - val_accuracy: 0.9784
Epoch 41/100
520/520 [=====] - 66s 127ms/step - loss: 0.0207
- accuracy: 0.9939 - val_loss: 0.1093 - val_accuracy: 0.9810
Epoch 42/100
520/520 [=====] - 65s 126ms/step - loss: 0.0208
- accuracy: 0.9939 - val_loss: 0.0997 - val_accuracy: 0.9837
Epoch 43/100
520/520 [=====] - 66s 127ms/step - loss: 0.0270
- accuracy: 0.9930 - val_loss: 0.1153 - val_accuracy: 0.9786
Epoch 44/100
520/520 [=====] - 66s 127ms/step - loss: 0.0237
- accuracy: 0.9939 - val_loss: 0.1023 - val_accuracy: 0.9837
Epoch 45/100
520/520 [=====] - 65s 126ms/step - loss: 0.0239
- accuracy: 0.9928 - val_loss: 0.1353 - val_accuracy: 0.9808
Epoch 46/100
520/520 [=====] - 67s 128ms/step - loss: 0.0213
- accuracy: 0.9935 - val_loss: 0.1223 - val_accuracy: 0.9808
Epoch 47/100
520/520 [=====] - 66s 126ms/step - loss: 0.0188
- accuracy: 0.9948 - val_loss: 0.0945 - val_accuracy: 0.9839
Epoch 48/100
520/520 [=====] - 66s 127ms/step - loss: 0.0215
- accuracy: 0.9940 - val_loss: 0.1140 - val_accuracy: 0.9810
Epoch 49/100
520/520 [=====] - 66s 127ms/step - loss: 0.0215
- accuracy: 0.9946 - val_loss: 0.1022 - val_accuracy: 0.9822
Epoch 50/100
520/520 [=====] - 66s 127ms/step - loss: 0.0161
- accuracy: 0.9956 - val_loss: 0.1733 - val_accuracy: 0.9803
Epoch 51/100
520/520 [=====] - 66s 127ms/step - loss: 0.0247
- accuracy: 0.9937 - val_loss: 0.0873 - val_accuracy: 0.9834

```

```

Epoch 52/100
520/520 [=====] - 66s 127ms/step - loss: 0.0193
- accuracy: 0.9937 - val_loss: 0.0981 - val_accuracy: 0.9825
Epoch 53/100
520/520 [=====] - 66s 127ms/step - loss: 0.0184
- accuracy: 0.9941 - val_loss: 0.1234 - val_accuracy: 0.9832
Epoch 54/100
520/520 [=====] - 67s 129ms/step - loss: 0.0209
- accuracy: 0.9950 - val_loss: 0.1123 - val_accuracy: 0.9825
Epoch 55/100
520/520 [=====] - 67s 128ms/step - loss: 0.0233
- accuracy: 0.9945 - val_loss: 0.0823 - val_accuracy: 0.9853
Epoch 56/100
520/520 [=====] - 67s 128ms/step - loss: 0.0157
- accuracy: 0.9957 - val_loss: 0.1194 - val_accuracy: 0.9812
Epoch 57/100
520/520 [=====] - 67s 128ms/step - loss: 0.0274
- accuracy: 0.9934 - val_loss: 0.1316 - val_accuracy: 0.9832
Epoch 58/100
520/520 [=====] - 66s 127ms/step - loss: 0.0229
- accuracy: 0.9944 - val_loss: 0.1598 - val_accuracy: 0.9798
Epoch 59/100
520/520 [=====] - 66s 128ms/step - loss: 0.0255
- accuracy: 0.9942 - val_loss: 0.1223 - val_accuracy: 0.9815
Epoch 60/100
520/520 [=====] - 67s 128ms/step - loss: 0.0190
- accuracy: 0.9954 - val_loss: 0.1248 - val_accuracy: 0.9805
Epoch 61/100
520/520 [=====] - 66s 127ms/step - loss: 0.0193
- accuracy: 0.9948 - val_loss: 0.1029 - val_accuracy: 0.9858
Epoch 62/100
520/520 [=====] - 66s 127ms/step - loss: 0.0260
- accuracy: 0.9939 - val_loss: 0.1317 - val_accuracy: 0.9820
Epoch 63/100
520/520 [=====] - 66s 128ms/step - loss: 0.0161
- accuracy: 0.9957 - val_loss: 0.1154 - val_accuracy: 0.9844
Epoch 64/100
520/520 [=====] - 67s 128ms/step - loss: 0.0270
- accuracy: 0.9940 - val_loss: 0.1360 - val_accuracy: 0.9841
Epoch 65/100
520/520 [=====] - 66s 128ms/step - loss: 0.0193
- accuracy: 0.9951 - val_loss: 0.1537 - val_accuracy: 0.9827
Epoch 66/100
520/520 [=====] - 66s 127ms/step - loss: 0.0213
- accuracy: 0.9939 - val_loss: 0.1330 - val_accuracy: 0.9815
Epoch 67/100
520/520 [=====] - 66s 127ms/step - loss: 0.0207
- accuracy: 0.9954 - val_loss: 0.1732 - val_accuracy: 0.9829
Epoch 68/100
520/520 [=====] - 66s 126ms/step - loss: 0.0303
- accuracy: 0.9927 - val_loss: 0.1207 - val_accuracy: 0.9832
Epoch 69/100
520/520 [=====] - 66s 127ms/step - loss: 0.0217
- accuracy: 0.9949 - val_loss: 0.1505 - val_accuracy: 0.9832
Epoch 70/100
520/520 [=====] - 66s 127ms/step - loss: 0.0156
- accuracy: 0.9962 - val_loss: 0.1383 - val_accuracy: 0.9817
Epoch 71/100
520/520 [=====] - 66s 127ms/step - loss: 0.0216
- accuracy: 0.9947 - val_loss: 0.1312 - val_accuracy: 0.9849
Epoch 72/100
520/520 [=====] - 65s 126ms/step - loss: 0.0211
- accuracy: 0.9951 - val_loss: 0.1634 - val_accuracy: 0.9846
Epoch 73/100

```

```

520/520 [=====] - 66s 126ms/step - loss: 0.0292
- accuracy: 0.9937 - val_loss: 0.1497 - val_accuracy: 0.9827
Epoch 74/100
520/520 [=====] - 66s 127ms/step - loss: 0.0201
- accuracy: 0.9953 - val_loss: 0.1542 - val_accuracy: 0.9817
Epoch 75/100
520/520 [=====] - 65s 126ms/step - loss: 0.0207
- accuracy: 0.9960 - val_loss: 0.1310 - val_accuracy: 0.9829
Epoch 76/100
520/520 [=====] - 66s 126ms/step - loss: 0.0222
- accuracy: 0.9948 - val_loss: 0.1798 - val_accuracy: 0.9825
Epoch 77/100
520/520 [=====] - 66s 127ms/step - loss: 0.0209
- accuracy: 0.9953 - val_loss: 0.1980 - val_accuracy: 0.9796
Epoch 78/100
520/520 [=====] - 66s 127ms/step - loss: 0.0254
- accuracy: 0.9949 - val_loss: 0.2110 - val_accuracy: 0.9793
Epoch 79/100
520/520 [=====] - 66s 127ms/step - loss: 0.0314
- accuracy: 0.9937 - val_loss: 0.1528 - val_accuracy: 0.9817
Epoch 80/100
520/520 [=====] - 65s 126ms/step - loss: 0.0222
- accuracy: 0.9956 - val_loss: 0.1407 - val_accuracy: 0.9839
Epoch 81/100
520/520 [=====] - 66s 127ms/step - loss: 0.0178
- accuracy: 0.9956 - val_loss: 0.1964 - val_accuracy: 0.9837
Epoch 82/100
520/520 [=====] - 67s 128ms/step - loss: 0.0220
- accuracy: 0.9955 - val_loss: 0.2035 - val_accuracy: 0.9822
Epoch 83/100
520/520 [=====] - 66s 128ms/step - loss: 0.0238
- accuracy: 0.9951 - val_loss: 0.1866 - val_accuracy: 0.9781
Epoch 84/100
520/520 [=====] - 66s 127ms/step - loss: 0.0187
- accuracy: 0.9953 - val_loss: 0.1744 - val_accuracy: 0.9851
Epoch 85/100
520/520 [=====] - 65s 126ms/step - loss: 0.0273
- accuracy: 0.9950 - val_loss: 0.1438 - val_accuracy: 0.9861
Epoch 86/100
520/520 [=====] - 66s 126ms/step - loss: 0.0151
- accuracy: 0.9959 - val_loss: 0.1488 - val_accuracy: 0.9834
Epoch 87/100
520/520 [=====] - 65s 126ms/step - loss: 0.0192
- accuracy: 0.9961 - val_loss: 0.1613 - val_accuracy: 0.9827
Epoch 88/100
520/520 [=====] - 66s 127ms/step - loss: 0.0240
- accuracy: 0.9949 - val_loss: 0.1252 - val_accuracy: 0.9863
Epoch 89/100
520/520 [=====] - 65s 126ms/step - loss: 0.0175
- accuracy: 0.9962 - val_loss: 0.1790 - val_accuracy: 0.9832
Epoch 90/100
520/520 [=====] - 65s 126ms/step - loss: 0.0225
- accuracy: 0.9954 - val_loss: 0.1346 - val_accuracy: 0.9873
Epoch 91/100
520/520 [=====] - 66s 127ms/step - loss: 0.0247
- accuracy: 0.9948 - val_loss: 0.1913 - val_accuracy: 0.9841
Epoch 92/100
520/520 [=====] - 66s 126ms/step - loss: 0.0210
- accuracy: 0.9958 - val_loss: 0.1957 - val_accuracy: 0.9837
Epoch 93/100
520/520 [=====] - 67s 128ms/step - loss: 0.0271
- accuracy: 0.9945 - val_loss: 0.1732 - val_accuracy: 0.9825
Epoch 94/100
520/520 [=====] - 66s 127ms/step - loss: 0.0202
- accuracy: 0.9959 - val_loss: 0.1189 - val_accuracy: 0.9841

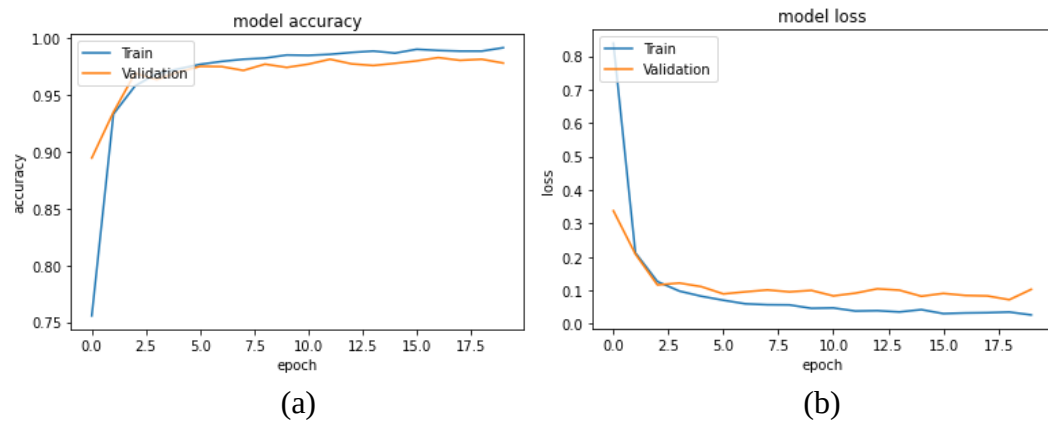
```

```

Epoch 95/100
520/520 [=====] - 65s 126ms/step - loss: 0.0167
- accuracy: 0.9962 - val_loss: 0.1671 - val_accuracy: 0.9844
Epoch 96/100
520/520 [=====] - 66s 127ms/step - loss: 0.0246
- accuracy: 0.9959 - val_loss: 0.1350 - val_accuracy: 0.9868
Epoch 97/100
520/520 [=====] - 66s 128ms/step - loss: 0.0163
- accuracy: 0.9965 - val_loss: 0.1834 - val_accuracy: 0.9837
Epoch 98/100
520/520 [=====] - 66s 127ms/step - loss: 0.0212
- accuracy: 0.9960 - val_loss: 0.1482 - val_accuracy: 0.9870
Epoch 99/100
520/520 [=====] - 66s 127ms/step - loss: 0.0183
- accuracy: 0.9963 - val_loss: 0.1613 - val_accuracy: 0.9829
Epoch 100/100
520/520 [=====] - 66s 127ms/step - loss:
0.0237 - accuracy: 0.9959 - val_loss: 0.1799 - val_accuracy: 0.9825

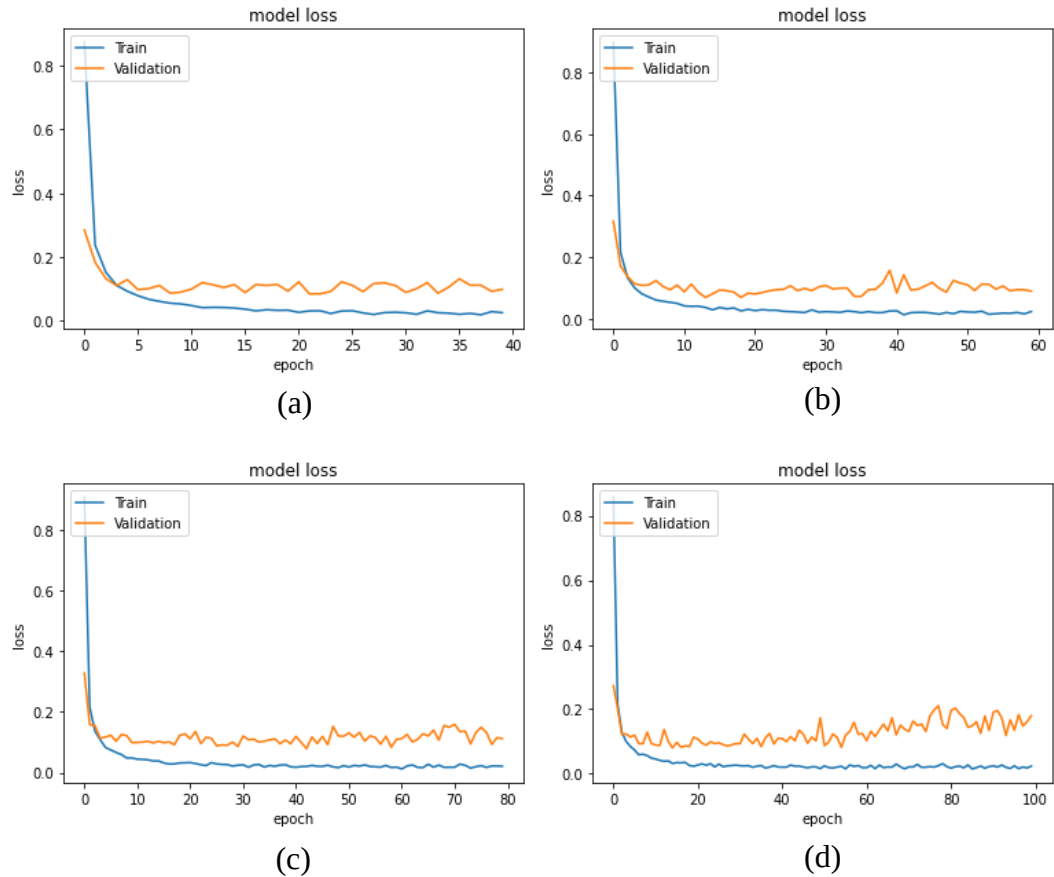
```

Jika proses hasil pembelajaran untuk tiap epoch pada tiap variasi epoch diatas digambarkan dalam bentuk grafik hasilnya akan seperti pada Gambar 4.9.



**Gambar 4.9** (a) dan (b) Hasil *Output* Pembelajaran Sistem Dengan *Convex Hull* + CNN Pada Skenario 4 Dengan 20 Epoch

Karena terdapat masalah pada saat penggambaran hasil *output* pembelajaran untuk epoch 40, 60, 80, dan 100 dimana plot *model accuracy*nya bercampur dengan tabel hasil prediksi *confusion\_matrix* sehingga yang tergambarakan hanya *model loss*nya saja, maka *model loss* hasil pembelajaran untuk epoch 40, 60, 80, dan 100 bisa dilihat pada Gambar 4.10.



**Gambar 4.10** (a), (b), (c), dan (d) Hasil Output Pembelajaran Sistem Dengan *Convex Hull* + CNN Pada Skenario 4 Dengan Epoch 40, 60, 80, dan 100

Setelah model dibuat, selanjutnya akan dilakukan prediksi dan evaluasi terhadap model pembelajaran CNN untuk tiap epoch.

#### 4.5.4.2 Hasil Evaluasi Sistem *Convex Hull* + CNN

Setelah file model hasil pembelajaran dihasilkan, selanjutnya akan dilakukan prediksi dan evaluasi pada data *testingPP* dengan menggunakan model hasil hasil pembelajaran tersebut untuk masing-masing epoch. Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 20 epoch bisa dilihat pada Tabel 4.8. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasi pada evaluasi model dengan 20 epoch dengan menggunakan fungsi *classification\_report*.

|   | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| a | 0.98      | 1.00   | 0.99     | 200     |
| b | 0.99      | 0.99   | 0.99     | 200     |
| c | 0.98      | 1.00   | 0.99     | 200     |
| d | 0.99      | 0.93   | 0.96     | 200     |
| e | 1.00      | 1.00   | 1.00     | 200     |
| f | 1.00      | 1.00   | 1.00     | 200     |

|              |      |      |      |      |
|--------------|------|------|------|------|
| g            | 0.99 | 0.99 | 0.99 | 200  |
| h            | 1.00 | 1.00 | 1.00 | 200  |
| i            | 0.98 | 0.96 | 0.97 | 200  |
| j            | 1.00 | 1.00 | 1.00 | 200  |
| k            | 0.97 | 0.98 | 0.98 | 200  |
| l            | 0.97 | 0.97 | 0.97 | 200  |
| m            | 1.00 | 1.00 | 1.00 | 200  |
| n            | 1.00 | 1.00 | 1.00 | 200  |
| o            | 0.99 | 0.98 | 0.99 | 200  |
| p            | 1.00 | 1.00 | 1.00 | 200  |
| q            | 0.99 | 0.98 | 0.99 | 200  |
| r            | 0.99 | 0.98 | 0.99 | 200  |
| s            | 1.00 | 1.00 | 1.00 | 200  |
| t            | 1.00 | 1.00 | 1.00 | 200  |
| u            | 1.00 | 1.00 | 1.00 | 200  |
| v            | 0.95 | 0.99 | 0.97 | 200  |
| w            | 0.98 | 0.98 | 0.98 | 200  |
| x            | 1.00 | 1.00 | 1.00 | 200  |
| y            | 0.99 | 0.99 | 0.99 | 200  |
| z            | 1.00 | 1.00 | 1.00 | 200  |
| accuracy     |      |      | 0.99 | 5200 |
| macro avg    | 0.99 | 0.99 | 0.99 | 5200 |
| weighted avg | 0.99 | 0.99 | 0.99 | 5200 |

Accuracty : 0.990576923076923.

Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 40 epoch bisa dilihat pada Tabel 4.9. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya pada evaluasi model dengan 40 epoch dengan menggunakan fungsi *classification\_report*.

|          | precision | recall | f1-score | support |
|----------|-----------|--------|----------|---------|
| a        | 1.00      | 1.00   | 1.00     | 200     |
| b        | 0.99      | 0.94   | 0.96     | 200     |
| c        | 0.99      | 1.00   | 0.99     | 200     |
| d        | 0.99      | 0.97   | 0.98     | 200     |
| e        | 1.00      | 1.00   | 1.00     | 200     |
| f        | 0.97      | 1.00   | 0.99     | 200     |
| g        | 0.99      | 0.90   | 0.94     | 200     |
| h        | 1.00      | 0.99   | 1.00     | 200     |
| i        | 0.98      | 0.97   | 0.97     | 200     |
| j        | 1.00      | 1.00   | 1.00     | 200     |
| k        | 0.98      | 0.98   | 0.98     | 200     |
| l        | 0.98      | 0.98   | 0.98     | 200     |
| m        | 1.00      | 1.00   | 1.00     | 200     |
| n        | 1.00      | 1.00   | 1.00     | 200     |
| o        | 0.99      | 0.99   | 0.99     | 200     |
| p        | 1.00      | 1.00   | 1.00     | 200     |
| q        | 1.00      | 0.98   | 0.99     | 200     |
| r        | 0.98      | 0.99   | 0.99     | 200     |
| s        | 1.00      | 1.00   | 1.00     | 200     |
| t        | 1.00      | 1.00   | 1.00     | 200     |
| u        | 1.00      | 1.00   | 1.00     | 200     |
| v        | 0.95      | 0.99   | 0.97     | 200     |
| w        | 0.97      | 0.98   | 0.98     | 200     |
| x        | 0.94      | 1.00   | 0.97     | 200     |
| y        | 0.99      | 1.00   | 1.00     | 200     |
| z        | 1.00      | 1.00   | 1.00     | 200     |
| accuracy |           |        | 0.99     | 5200    |



```

macro avg      0.99      0.99      0.99      5200
weighted avg   0.99      0.99      0.99      5200
Accuracy :    0.9876923076923076

```

Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 60 epoch bisa dilihat pada Tabel 4.10. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya pada evaluasi model dengan 60 epoch dengan menggunakan fungsi *classification\_report*.

|   | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| a | 0.99      | 1.00   | 0.99     | 200     |
| b | 0.97      | 0.96   | 0.97     | 200     |
| c | 1.00      | 1.00   | 1.00     | 200     |
| d | 0.95      | 0.98   | 0.97     | 200     |
| e | 1.00      | 1.00   | 1.00     | 200     |
| f | 0.99      | 0.99   | 0.99     | 200     |
| g | 1.00      | 0.97   | 0.98     | 200     |
| h | 1.00      | 1.00   | 1.00     | 200     |
| i | 0.99      | 0.96   | 0.97     | 200     |
| j | 1.00      | 1.00   | 1.00     | 200     |
| k | 0.98      | 0.98   | 0.98     | 200     |
| l | 0.98      | 0.97   | 0.98     | 200     |
| m | 1.00      | 1.00   | 1.00     | 200     |
| n | 1.00      | 1.00   | 1.00     | 200     |
| o | 0.99      | 0.98   | 0.99     | 200     |
| p | 1.00      | 1.00   | 1.00     | 200     |
| q | 1.00      | 1.00   | 1.00     | 200     |
| r | 0.97      | 0.99   | 0.98     | 200     |
| s | 1.00      | 1.00   | 1.00     | 200     |
| t | 1.00      | 1.00   | 1.00     | 200     |
| u | 1.00      | 1.00   | 1.00     | 200     |
| v | 0.98      | 0.97   | 0.98     | 200     |
| w | 0.99      | 0.99   | 0.99     | 200     |
| x | 1.00      | 1.00   | 1.00     | 200     |
| y | 1.00      | 0.99   | 1.00     | 200     |
| z | 1.00      | 1.00   | 1.00     | 200     |

```

accuracy      0.99      5200
macro avg     0.99      0.99      0.99      5200
weighted avg  0.99      0.99      0.99      5200
Accuracy :    0.9911538461538462

```

Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 80 epoch bisa dilihat pada table 4.11. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya pada evaluasi model dengan 80 epoch dengan menggunakan fungsi *classification\_report*.

|   | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| a | 0.98      | 1.00   | 0.99     | 200     |
| b | 0.98      | 0.93   | 0.95     | 200     |
| c | 0.99      | 1.00   | 1.00     | 200     |
| d | 0.98      | 0.97   | 0.98     | 200     |
| e | 1.00      | 1.00   | 1.00     | 200     |
| f | 0.99      | 0.99   | 0.99     | 200     |
| g | 1.00      | 0.98   | 0.99     | 200     |
| h | 1.00      | 1.00   | 1.00     | 200     |
| i | 0.97      | 0.96   | 0.97     | 200     |
| j | 1.00      | 1.00   | 1.00     | 200     |

|              |      |      |      |      |
|--------------|------|------|------|------|
| k            | 0.98 | 0.98 | 0.98 | 200  |
| l            | 0.98 | 0.98 | 0.98 | 200  |
| m            | 1.00 | 0.99 | 1.00 | 200  |
| n            | 1.00 | 1.00 | 1.00 | 200  |
| o            | 1.00 | 0.98 | 0.99 | 200  |
| p            | 1.00 | 1.00 | 1.00 | 200  |
| q            | 1.00 | 0.99 | 1.00 | 200  |
| r            | 0.94 | 0.99 | 0.97 | 200  |
| s            | 1.00 | 1.00 | 1.00 | 200  |
| t            | 1.00 | 1.00 | 1.00 | 200  |
| u            | 1.00 | 1.00 | 1.00 | 200  |
| v            | 0.97 | 0.97 | 0.97 | 200  |
| w            | 0.98 | 0.98 | 0.98 | 200  |
| x            | 1.00 | 1.00 | 1.00 | 200  |
| y            | 0.99 | 0.99 | 0.99 | 200  |
| z            | 1.00 | 1.00 | 1.00 | 200  |
| accuracy     |      |      | 0.99 | 5200 |
| macro avg    | 0.99 | 0.99 | 0.99 | 5200 |
| weighted avg | 0.99 | 0.99 | 0.99 | 5200 |

Accuracty : 0.989423076923077

Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 100 epoch bisa dilihat pada Tabel 4.12. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya pada evaluasi model dengan 100 epoch dengan menggunakan fungsi *classification\_report*.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| a            | 0.97      | 1.00   | 0.98     | 200     |
| b            | 0.99      | 0.97   | 0.98     | 200     |
| c            | 1.00      | 1.00   | 1.00     | 200     |
| d            | 0.98      | 0.94   | 0.96     | 200     |
| e            | 1.00      | 1.00   | 1.00     | 200     |
| f            | 0.99      | 0.99   | 0.99     | 200     |
| g            | 0.99      | 0.96   | 0.98     | 200     |
| h            | 1.00      | 1.00   | 1.00     | 200     |
| i            | 0.99      | 0.95   | 0.97     | 200     |
| j            | 1.00      | 1.00   | 1.00     | 200     |
| k            | 0.99      | 0.98   | 0.99     | 200     |
| l            | 0.97      | 0.99   | 0.98     | 200     |
| m            | 1.00      | 1.00   | 1.00     | 200     |
| n            | 1.00      | 1.00   | 1.00     | 200     |
| o            | 0.99      | 0.98   | 0.99     | 200     |
| p            | 1.00      | 1.00   | 1.00     | 200     |
| q            | 1.00      | 0.98   | 0.99     | 200     |
| r            | 0.99      | 0.99   | 0.99     | 200     |
| s            | 0.99      | 1.00   | 0.99     | 200     |
| t            | 0.99      | 1.00   | 1.00     | 200     |
| u            | 1.00      | 1.00   | 1.00     | 200     |
| v            | 0.94      | 0.99   | 0.97     | 200     |
| w            | 0.99      | 0.98   | 0.99     | 200     |
| x            | 1.00      | 1.00   | 1.00     | 200     |
| y            | 1.00      | 1.00   | 1.00     | 200     |
| z            | 1.00      | 1.00   | 1.00     | 200     |
| accuracy     |           |        | 0.99     | 5200    |
| macro avg    | 0.99      | 0.99   | 0.99     | 5200    |
| weighted avg | 0.99      | 0.99   | 0.99     | 5200    |

Accuracty : 0.9896153846153846

Setelah dilakukan evaluasi dengan menggunakan 20, 40, 60, 80, dan 100 epoch, didapatkan hasil akurasi dari 5x ujicoba sebagai berikut.

```
===== Hasil Rata - Rata =====  
Accuracy from epouch = 20 : 0.990576923076923  
Accuracy from epouch = 40 : 0.9876923076923076  
Accuracy from epouch = 60 : 0.9911538461538462  
Accuracy from epouch = 80 : 0.989423076923077  
Accuracy from epouch = 100 : 0.9896153846153846  
Average accuracy : 0.9896923076923076
```

Nilai akurasi rata-rata yang didapatkan adalah 0.9896923076923076 atau 98.96%.

#### 4.5.4.3 Hasil Pembelajaran Sistem CNN Tanpa *Convex Hull*

Setelah struktur model dibuat, akan dilakukan proses kompilasi dan memulai proses pembelajaran terhadap data *Training* menggunakan struktur model yang telah dibuat dengan menggunakan variasi 20, 40, 60, 80, dan 100 epoch. Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 20 epoch.

```
Epoch 1/20  
520/520 [=====] - 6810s 13s/step - loss: 1.7685  
- accuracy: 0.4834 - val_loss: 0.6771 - val_accuracy: 0.8087  
Epoch 2/20  
520/520 [=====] - 66s 126ms/step - loss: 0.4691  
- accuracy: 0.8568 - val_loss: 0.3205 - val_accuracy: 0.9060  
Epoch 3/20  
520/520 [=====] - 65s 126ms/step - loss: 0.3055  
- accuracy: 0.9053 - val_loss: 0.2398 - val_accuracy: 0.9300  
Epoch 4/20  
520/520 [=====] - 66s 126ms/step - loss: 0.2308  
- accuracy: 0.9272 - val_loss: 0.2168 - val_accuracy: 0.9337  
Epoch 5/20  
520/520 [=====] - 66s 127ms/step - loss: 0.1950  
- accuracy: 0.9397 - val_loss: 0.1821 - val_accuracy: 0.9512  
Epoch 6/20  
520/520 [=====] - 65s 126ms/step - loss: 0.1644  
- accuracy: 0.9478 - val_loss: 0.1530 - val_accuracy: 0.9575  
Epoch 7/20  
520/520 [=====] - 65s 126ms/step - loss: 0.1440  
- accuracy: 0.9550 - val_loss: 0.1498 - val_accuracy: 0.9553  
Epoch 8/20  
520/520 [=====] - 66s 126ms/step - loss: 0.1291  
- accuracy: 0.9570 - val_loss: 0.1220 - val_accuracy: 0.9647  
Epoch 9/20  
520/520 [=====] - 66s 126ms/step - loss: 0.1223  
- accuracy: 0.9579 - val_loss: 0.1328 - val_accuracy: 0.9644  
Epoch 10/20  
520/520 [=====] - 65s 126ms/step - loss: 0.1108  
- accuracy: 0.9626 - val_loss: 0.1560 - val_accuracy: 0.9582  
Epoch 11/20  
520/520 [=====] - 66s 126ms/step - loss: 0.1001  
- accuracy: 0.9689 - val_loss: 0.1129 - val_accuracy: 0.9702  
Epoch 12/20  
520/520 [=====] - 65s 125ms/step - loss: 0.0908  
- accuracy: 0.9698 - val_loss: 0.1056 - val_accuracy: 0.9697  
Epoch 13/20
```

```

520/520 [=====] - 66s 126ms/step - loss: 0.0837
- accuracy: 0.9739 - val_loss: 0.1230 - val_accuracy: 0.9675
Epoch 14/20
520/520 [=====] - 66s 126ms/step - loss: 0.0844
- accuracy: 0.9727 - val_loss: 0.1052 - val_accuracy: 0.9692
Epoch 15/20
520/520 [=====] - 65s 126ms/step - loss: 0.0901
- accuracy: 0.9706 - val_loss: 0.1035 - val_accuracy: 0.9755
Epoch 16/20
520/520 [=====] - 65s 125ms/step - loss: 0.0735
- accuracy: 0.9767 - val_loss: 0.1161 - val_accuracy: 0.9697
Epoch 17/20
520/520 [=====] - 66s 126ms/step - loss: 0.0717
- accuracy: 0.9772 - val_loss: 0.1117 - val_accuracy: 0.9714
Epoch 18/20
520/520 [=====] - 66s 127ms/step - loss: 0.0780
- accuracy: 0.9737 - val_loss: 0.0871 - val_accuracy: 0.9803
Epoch 19/20
520/520 [=====] - 66s 127ms/step - loss: 0.0681
- accuracy: 0.9782 - val_loss: 0.1066 - val_accuracy: 0.9750
Epoch 20/20
520/520 [=====] - 65s 126ms/step - loss: 0.0652
- accuracy: 0.9786 - val_loss: 0.1080 - val_accuracy: 0.9726

```

Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 40 epoch.

```

Epoch 1/40
520/520 [=====] - 66s 126ms/step - loss: 1.6867
- accuracy: 0.4971 - val_loss: 0.6730 - val_accuracy: 0.8070
Epoch 2/40
520/520 [=====] - 65s 126ms/step - loss: 0.5277
- accuracy: 0.8366 - val_loss: 0.3878 - val_accuracy: 0.8875
Epoch 3/40
520/520 [=====] - 66s 126ms/step - loss: 0.3337
- accuracy: 0.8897 - val_loss: 0.2875 - val_accuracy: 0.9118
Epoch 4/40
520/520 [=====] - 66s 126ms/step - loss: 0.2518
- accuracy: 0.9172 - val_loss: 0.2207 - val_accuracy: 0.9368
Epoch 5/40
520/520 [=====] - 66s 126ms/step - loss: 0.2104
- accuracy: 0.9329 - val_loss: 0.2362 - val_accuracy: 0.9291
Epoch 6/40
520/520 [=====] - 65s 125ms/step - loss: 0.1789
- accuracy: 0.9428 - val_loss: 0.1726 - val_accuracy: 0.9514
Epoch 7/40
520/520 [=====] - 65s 125ms/step - loss: 0.1576
- accuracy: 0.9496 - val_loss: 0.1578 - val_accuracy: 0.9531
Epoch 8/40
520/520 [=====] - 65s 125ms/step - loss: 0.1382
- accuracy: 0.9543 - val_loss: 0.1209 - val_accuracy: 0.9673
Epoch 9/40
520/520 [=====] - 66s 126ms/step - loss: 0.1295
- accuracy: 0.9581 - val_loss: 0.1385 - val_accuracy: 0.9579
Epoch 10/40
520/520 [=====] - 65s 125ms/step - loss: 0.1116
- accuracy: 0.9632 - val_loss: 0.1148 - val_accuracy: 0.9704
Epoch 11/40
520/520 [=====] - 66s 126ms/step - loss: 0.1130
- accuracy: 0.9639 - val_loss: 0.1435 - val_accuracy: 0.9587
Epoch 12/40
520/520 [=====] - 66s 126ms/step - loss: 0.1013
- accuracy: 0.9672 - val_loss: 0.1253 - val_accuracy: 0.9651
Epoch 13/40

```

```

520/520 [=====] - 65s 126ms/step - loss: 0.0927
- accuracy: 0.9697 - val_loss: 0.1261 - val_accuracy: 0.9635
Epoch 14/40
520/520 [=====] - 66s 127ms/step - loss: 0.0865
- accuracy: 0.9724 - val_loss: 0.1234 - val_accuracy: 0.9671
Epoch 15/40
520/520 [=====] - 65s 125ms/step - loss: 0.0825
- accuracy: 0.9746 - val_loss: 0.0957 - val_accuracy: 0.9769
Epoch 16/40
520/520 [=====] - 65s 125ms/step - loss: 0.0743
- accuracy: 0.9762 - val_loss: 0.1070 - val_accuracy: 0.9767
Epoch 17/40
520/520 [=====] - 65s 126ms/step - loss: 0.0787
- accuracy: 0.9750 - val_loss: 0.1146 - val_accuracy: 0.9707
Epoch 18/40
520/520 [=====] - 66s 126ms/step - loss: 0.0741
- accuracy: 0.9763 - val_loss: 0.0923 - val_accuracy: 0.9764
Epoch 19/40
520/520 [=====] - 66s 127ms/step - loss: 0.0721
- accuracy: 0.9773 - val_loss: 0.0880 - val_accuracy: 0.9743
Epoch 20/40
520/520 [=====] - 65s 125ms/step - loss: 0.0652
- accuracy: 0.9793 - val_loss: 0.1155 - val_accuracy: 0.9731
Epoch 21/40
520/520 [=====] - 65s 126ms/step - loss: 0.0714
- accuracy: 0.9775 - val_loss: 0.1008 - val_accuracy: 0.9760
Epoch 22/40
520/520 [=====] - 65s 126ms/step - loss: 0.0624
- accuracy: 0.9808 - val_loss: 0.1040 - val_accuracy: 0.9745
Epoch 23/40
520/520 [=====] - 65s 126ms/step - loss: 0.0590
- accuracy: 0.9806 - val_loss: 0.0975 - val_accuracy: 0.9748
Epoch 24/40
520/520 [=====] - 66s 127ms/step - loss: 0.0672
- accuracy: 0.9803 - val_loss: 0.1077 - val_accuracy: 0.9752
Epoch 25/40
520/520 [=====] - 66s 126ms/step - loss: 0.0661
- accuracy: 0.9794 - val_loss: 0.1008 - val_accuracy: 0.9745
Epoch 26/40
520/520 [=====] - 65s 126ms/step - loss: 0.0657
- accuracy: 0.9806 - val_loss: 0.1502 - val_accuracy: 0.9613
Epoch 27/40
520/520 [=====] - 66s 126ms/step - loss: 0.0572
- accuracy: 0.9823 - val_loss: 0.0975 - val_accuracy: 0.9764
Epoch 28/40
520/520 [=====] - 66s 127ms/step - loss: 0.0523
- accuracy: 0.9834 - val_loss: 0.1042 - val_accuracy: 0.9786
Epoch 29/40
520/520 [=====] - 66s 127ms/step - loss: 0.0469
- accuracy: 0.9861 - val_loss: 0.0935 - val_accuracy: 0.9784
Epoch 30/40
520/520 [=====] - 65s 126ms/step - loss: 0.0450
- accuracy: 0.9840 - val_loss: 0.0985 - val_accuracy: 0.9767
Epoch 31/40
520/520 [=====] - 65s 125ms/step - loss: 0.0559
- accuracy: 0.9838 - val_loss: 0.0999 - val_accuracy: 0.9800
Epoch 32/40
520/520 [=====] - 66s 126ms/step - loss: 0.0566
- accuracy: 0.9822 - val_loss: 0.0973 - val_accuracy: 0.9788
Epoch 33/40
520/520 [=====] - 66s 127ms/step - loss: 0.0493
- accuracy: 0.9852 - val_loss: 0.1109 - val_accuracy: 0.9745
Epoch 34/40
520/520 [=====] - 66s 126ms/step - loss: 0.0510
- accuracy: 0.9841 - val_loss: 0.0975 - val_accuracy: 0.9793

```

```

Epoch 35/40
520/520 [=====] - 65s 126ms/step - loss: 0.0459
- accuracy: 0.9859 - val_loss: 0.0996 - val_accuracy: 0.9798
Epoch 36/40
520/520 [=====] - 65s 126ms/step - loss: 0.0486
- accuracy: 0.9844 - val_loss: 0.1128 - val_accuracy: 0.9779
Epoch 37/40
520/520 [=====] - 66s 126ms/step - loss: 0.0544
- accuracy: 0.9831 - val_loss: 0.0932 - val_accuracy: 0.9784
Epoch 38/40
520/520 [=====] - 65s 126ms/step - loss: 0.0451
- accuracy: 0.9863 - val_loss: 0.1027 - val_accuracy: 0.9762
Epoch 39/40
520/520 [=====] - 66s 126ms/step - loss: 0.0427
- accuracy: 0.9859 - val_loss: 0.1037 - val_accuracy: 0.9767
Epoch 40/40
520/520 [=====] - 66s 126ms/step - loss: 0.0454
- accuracy: 0.9866 - val_loss: 0.1002 - val_accuracy: 0.9793

```

Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 60 epoch.

```

Epoch 1/60
520/520 [=====] - 66s 127ms/step - loss: 1.5818
- accuracy: 0.5312 - val_loss: 0.6508 - val_accuracy: 0.8195
Epoch 2/60
520/520 [=====] - 66s 127ms/step - loss: 0.5031
- accuracy: 0.8437 - val_loss: 0.3250 - val_accuracy: 0.9034
Epoch 3/60
520/520 [=====] - 66s 127ms/step - loss: 0.3154
- accuracy: 0.8992 - val_loss: 0.2437 - val_accuracy: 0.9276
Epoch 4/60
520/520 [=====] - 65s 126ms/step - loss: 0.2249
- accuracy: 0.9266 - val_loss: 0.1974 - val_accuracy: 0.9440
Epoch 5/60
520/520 [=====] - 66s 127ms/step - loss: 0.1846
- accuracy: 0.9392 - val_loss: 0.1637 - val_accuracy: 0.9536
Epoch 6/60
520/520 [=====] - 66s 128ms/step - loss: 0.1518
- accuracy: 0.9525 - val_loss: 0.1607 - val_accuracy: 0.9548
Epoch 7/60
520/520 [=====] - 66s 127ms/step - loss: 0.1322
- accuracy: 0.9568 - val_loss: 0.1386 - val_accuracy: 0.9587
Epoch 8/60
520/520 [=====] - 66s 127ms/step - loss: 0.1212
- accuracy: 0.9617 - val_loss: 0.1344 - val_accuracy: 0.9625
Epoch 9/60
520/520 [=====] - 66s 127ms/step - loss: 0.1128
- accuracy: 0.9640 - val_loss: 0.1194 - val_accuracy: 0.9671
Epoch 10/60
520/520 [=====] - 66s 127ms/step - loss: 0.1098
- accuracy: 0.9659 - val_loss: 0.1381 - val_accuracy: 0.9606
Epoch 11/60
520/520 [=====] - 66s 127ms/step - loss: 0.0934
- accuracy: 0.9700 - val_loss: 0.1361 - val_accuracy: 0.9654
Epoch 12/60
520/520 [=====] - 66s 127ms/step - loss: 0.0886
- accuracy: 0.9716 - val_loss: 0.1090 - val_accuracy: 0.9704
Epoch 13/60
520/520 [=====] - 66s 127ms/step - loss: 0.0883
- accuracy: 0.9713 - val_loss: 0.1106 - val_accuracy: 0.9685
Epoch 14/60
520/520 [=====] - 66s 127ms/step - loss: 0.0775
- accuracy: 0.9744 - val_loss: 0.1058 - val_accuracy: 0.9721

```

```

Epoch 15/60
520/520 [=====] - 66s 126ms/step - loss: 0.0779
- accuracy: 0.9755 - val_loss: 0.1203 - val_accuracy: 0.9675
Epoch 16/60
520/520 [=====] - 66s 128ms/step - loss: 0.0781
- accuracy: 0.9750 - val_loss: 0.1093 - val_accuracy: 0.9712
Epoch 17/60
520/520 [=====] - 65s 125ms/step - loss: 0.0692
- accuracy: 0.9776 - val_loss: 0.0988 - val_accuracy: 0.9731
Epoch 18/60
520/520 [=====] - 64s 124ms/step - loss: 0.0680
- accuracy: 0.9779 - val_loss: 0.1170 - val_accuracy: 0.9724
Epoch 19/60
520/520 [=====] - 64s 123ms/step - loss: 0.0654
- accuracy: 0.9796 - val_loss: 0.0930 - val_accuracy: 0.9767
Epoch 20/60
520/520 [=====] - 65s 125ms/step - loss: 0.0697
- accuracy: 0.9784 - val_loss: 0.0962 - val_accuracy: 0.9774
Epoch 21/60
520/520 [=====] - 65s 125ms/step - loss: 0.0633
- accuracy: 0.9801 - val_loss: 0.0917 - val_accuracy: 0.9726
Epoch 22/60
520/520 [=====] - 64s 124ms/step - loss: 0.0586
- accuracy: 0.9807 - val_loss: 0.0954 - val_accuracy: 0.9786
Epoch 23/60
520/520 [=====] - 64s 124ms/step - loss: 0.0613
- accuracy: 0.9812 - val_loss: 0.1038 - val_accuracy: 0.9755
Epoch 24/60
520/520 [=====] - 64s 123ms/step - loss: 0.0593
- accuracy: 0.9809 - val_loss: 0.1164 - val_accuracy: 0.9736
Epoch 25/60
520/520 [=====] - 64s 124ms/step - loss: 0.0607
- accuracy: 0.9820 - val_loss: 0.1075 - val_accuracy: 0.9702
Epoch 26/60
520/520 [=====] - 64s 124ms/step - loss: 0.0509
- accuracy: 0.9844 - val_loss: 0.1028 - val_accuracy: 0.9733
Epoch 27/60
520/520 [=====] - 64s 124ms/step - loss: 0.0593
- accuracy: 0.9823 - val_loss: 0.1410 - val_accuracy: 0.9675
Epoch 28/60
520/520 [=====] - 64s 124ms/step - loss: 0.0609
- accuracy: 0.9817 - val_loss: 0.0959 - val_accuracy: 0.9786
Epoch 29/60
520/520 [=====] - 64s 123ms/step - loss: 0.0458
- accuracy: 0.9862 - val_loss: 0.0967 - val_accuracy: 0.9772
Epoch 30/60
520/520 [=====] - 64s 123ms/step - loss: 0.0499
- accuracy: 0.9838 - val_loss: 0.1103 - val_accuracy: 0.9757
Epoch 31/60
520/520 [=====] - 65s 124ms/step - loss: 0.0503
- accuracy: 0.9846 - val_loss: 0.1018 - val_accuracy: 0.9815
Epoch 32/60
520/520 [=====] - 65s 124ms/step - loss: 0.0568
- accuracy: 0.9819 - val_loss: 0.1247 - val_accuracy: 0.9767
Epoch 33/60
520/520 [=====] - 64s 123ms/step - loss: 0.0537
- accuracy: 0.9837 - val_loss: 0.0941 - val_accuracy: 0.9776
Epoch 34/60
520/520 [=====] - 64s 123ms/step - loss: 0.0477
- accuracy: 0.9862 - val_loss: 0.1170 - val_accuracy: 0.9750
Epoch 35/60
520/520 [=====] - 64s 123ms/step - loss: 0.0528
- accuracy: 0.9836 - val_loss: 0.1028 - val_accuracy: 0.9805
Epoch 36/60

```

```

520/520 [=====] - 65s 124ms/step - loss: 0.0515
- accuracy: 0.9841 - val_loss: 0.1093 - val_accuracy: 0.9791
Epoch 37/60
520/520 [=====] - 64s 123ms/step - loss: 0.0451
- accuracy: 0.9867 - val_loss: 0.1401 - val_accuracy: 0.9671
Epoch 38/60
520/520 [=====] - 64s 124ms/step - loss: 0.0441
- accuracy: 0.9875 - val_loss: 0.1148 - val_accuracy: 0.9731
Epoch 39/60
520/520 [=====] - 64s 123ms/step - loss: 0.0539
- accuracy: 0.9839 - val_loss: 0.1170 - val_accuracy: 0.9772
Epoch 40/60
520/520 [=====] - 64s 123ms/step - loss: 0.0432
- accuracy: 0.9868 - val_loss: 0.1421 - val_accuracy: 0.9724
Epoch 41/60
520/520 [=====] - 65s 125ms/step - loss: 0.0503
- accuracy: 0.9840 - val_loss: 0.1026 - val_accuracy: 0.9776
Epoch 42/60
520/520 [=====] - 64s 124ms/step - loss: 0.0378
- accuracy: 0.9880 - val_loss: 0.1012 - val_accuracy: 0.9786
Epoch 43/60
520/520 [=====] - 64s 124ms/step - loss: 0.0438
- accuracy: 0.9879 - val_loss: 0.1401 - val_accuracy: 0.9702
Epoch 44/60
520/520 [=====] - 64s 124ms/step - loss: 0.0480
- accuracy: 0.9859 - val_loss: 0.1107 - val_accuracy: 0.9764
Epoch 45/60
520/520 [=====] - 65s 124ms/step - loss: 0.0426
- accuracy: 0.9878 - val_loss: 0.1083 - val_accuracy: 0.9791
Epoch 46/60
520/520 [=====] - 64s 124ms/step - loss: 0.0437
- accuracy: 0.9873 - val_loss: 0.1084 - val_accuracy: 0.9776
Epoch 47/60
520/520 [=====] - 64s 123ms/step - loss: 0.0407
- accuracy: 0.9880 - val_loss: 0.1294 - val_accuracy: 0.9750
Epoch 48/60
520/520 [=====] - 64s 123ms/step - loss: 0.0394
- accuracy: 0.9878 - val_loss: 0.1181 - val_accuracy: 0.9810
Epoch 49/60
520/520 [=====] - 64s 123ms/step - loss: 0.0496
- accuracy: 0.9862 - val_loss: 0.1363 - val_accuracy: 0.9757
Epoch 50/60
520/520 [=====] - 65s 125ms/step - loss: 0.0434
- accuracy: 0.9863 - val_loss: 0.1257 - val_accuracy: 0.9769
Epoch 51/60
520/520 [=====] - 64s 124ms/step - loss: 0.0429
- accuracy: 0.9868 - val_loss: 0.1087 - val_accuracy: 0.9815
Epoch 52/60
520/520 [=====] - 64s 123ms/step - loss: 0.0421
- accuracy: 0.9883 - val_loss: 0.1414 - val_accuracy: 0.9760
Epoch 53/60
520/520 [=====] - 64s 123ms/step - loss: 0.0359
- accuracy: 0.9901 - val_loss: 0.1081 - val_accuracy: 0.9803
Epoch 54/60
520/520 [=====] - 64s 124ms/step - loss: 0.0489
- accuracy: 0.9855 - val_loss: 0.1176 - val_accuracy: 0.9798
Epoch 55/60
520/520 [=====] - 65s 125ms/step - loss: 0.0443
- accuracy: 0.9885 - val_loss: 0.1280 - val_accuracy: 0.9772
Epoch 56/60
520/520 [=====] - 64s 123ms/step - loss: 0.0381
- accuracy: 0.9894 - val_loss: 0.1237 - val_accuracy: 0.9791
Epoch 57/60
520/520 [=====] - 64s 123ms/step - loss: 0.0433
- accuracy: 0.9865 - val_loss: 0.1136 - val_accuracy: 0.9803

```



```

Epoch 58/60
520/520 [=====] - 64s 123ms/step - loss: 0.0443
- accuracy: 0.9871 - val_loss: 0.1186 - val_accuracy: 0.9808
Epoch 59/60
520/520 [=====] - 64s 124ms/step - loss: 0.0342
- accuracy: 0.9903 - val_loss: 0.1245 - val_accuracy: 0.9757
Epoch 60/60
520/520 [=====] - 64s 124ms/step - loss: 0.0442
- accuracy: 0.9882 - val_loss: 0.1090 - val_accuracy: 0.9815

```

Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 80 epoch.

```

Epoch 1/80
520/520 [=====] - 65s 124ms/step - loss: 1.9034
- accuracy: 0.4314 - val_loss: 0.8976 - val_accuracy: 0.7428
Epoch 2/80
520/520 [=====] - 65s 125ms/step - loss: 0.6432
- accuracy: 0.7992 - val_loss: 0.5108 - val_accuracy: 0.8466
Epoch 3/80
520/520 [=====] - 65s 124ms/step - loss: 0.3908
- accuracy: 0.8764 - val_loss: 0.3227 - val_accuracy: 0.9065
Epoch 4/80
520/520 [=====] - 65s 124ms/step - loss: 0.3003
- accuracy: 0.9032 - val_loss: 0.2562 - val_accuracy: 0.9286
Epoch 5/80
520/520 [=====] - 64s 124ms/step - loss: 0.2439
- accuracy: 0.9188 - val_loss: 0.2284 - val_accuracy: 0.9382
Epoch 6/80
520/520 [=====] - 64s 124ms/step - loss: 0.2074
- accuracy: 0.9328 - val_loss: 0.2078 - val_accuracy: 0.9430
Epoch 7/80
520/520 [=====] - 64s 124ms/step - loss: 0.1813
- accuracy: 0.9417 - val_loss: 0.1782 - val_accuracy: 0.9481
Epoch 8/80
520/520 [=====] - 64s 124ms/step - loss: 0.1613
- accuracy: 0.9471 - val_loss: 0.1667 - val_accuracy: 0.9524
Epoch 9/80
520/520 [=====] - 65s 124ms/step - loss: 0.1513
- accuracy: 0.9501 - val_loss: 0.1676 - val_accuracy: 0.9500
Epoch 10/80
520/520 [=====] - 64s 124ms/step - loss: 0.1298
- accuracy: 0.9581 - val_loss: 0.1664 - val_accuracy: 0.9570
Epoch 11/80
520/520 [=====] - 64s 124ms/step - loss: 0.1265
- accuracy: 0.9587 - val_loss: 0.1365 - val_accuracy: 0.9651
Epoch 12/80
520/520 [=====] - 65s 125ms/step - loss: 0.1232
- accuracy: 0.9593 - val_loss: 0.1567 - val_accuracy: 0.9510
Epoch 13/80
520/520 [=====] - 64s 124ms/step - loss: 0.1137
- accuracy: 0.9632 - val_loss: 0.1426 - val_accuracy: 0.9594
Epoch 14/80
520/520 [=====] - 65s 125ms/step - loss: 0.0968
- accuracy: 0.9672 - val_loss: 0.1390 - val_accuracy: 0.9579
Epoch 15/80
520/520 [=====] - 64s 124ms/step - loss: 0.1005
- accuracy: 0.9664 - val_loss: 0.1238 - val_accuracy: 0.9666
Epoch 16/80
520/520 [=====] - 65s 124ms/step - loss: 0.0878
- accuracy: 0.9734 - val_loss: 0.1218 - val_accuracy: 0.9668
Epoch 17/80
520/520 [=====] - 64s 124ms/step - loss: 0.0873
- accuracy: 0.9728 - val_loss: 0.1267 - val_accuracy: 0.9673
Epoch 18/80

```

```

520/520 [=====] - 65s 125ms/step - loss: 0.0844
- accuracy: 0.9722 - val_loss: 0.1160 - val_accuracy: 0.9702
Epoch 19/80
520/520 [=====] - 66s 127ms/step - loss: 0.0767
- accuracy: 0.9758 - val_loss: 0.1270 - val_accuracy: 0.9656
Epoch 20/80
520/520 [=====] - 65s 124ms/step - loss: 0.0836
- accuracy: 0.9731 - val_loss: 0.1320 - val_accuracy: 0.9635
Epoch 21/80
520/520 [=====] - 65s 124ms/step - loss: 0.0872
- accuracy: 0.9724 - val_loss: 0.1195 - val_accuracy: 0.9712
Epoch 22/80
520/520 [=====] - 65s 125ms/step - loss: 0.0698
- accuracy: 0.9778 - val_loss: 0.1508 - val_accuracy: 0.9611
Epoch 23/80
520/520 [=====] - 64s 124ms/step - loss: 0.0692
- accuracy: 0.9768 - val_loss: 0.1221 - val_accuracy: 0.9721
Epoch 24/80
520/520 [=====] - 64s 124ms/step - loss: 0.0718
- accuracy: 0.9775 - val_loss: 0.1022 - val_accuracy: 0.9769
Epoch 25/80
520/520 [=====] - 64s 124ms/step - loss: 0.0640
- accuracy: 0.9780 - val_loss: 0.1215 - val_accuracy: 0.9724
Epoch 26/80
520/520 [=====] - 65s 126ms/step - loss: 0.0625
- accuracy: 0.9797 - val_loss: 0.1119 - val_accuracy: 0.9740
Epoch 27/80
520/520 [=====] - 65s 125ms/step - loss: 0.0632
- accuracy: 0.9794 - val_loss: 0.1192 - val_accuracy: 0.9721
Epoch 28/80
520/520 [=====] - 64s 124ms/step - loss: 0.0675
- accuracy: 0.9787 - val_loss: 0.1103 - val_accuracy: 0.9738
Epoch 29/80
520/520 [=====] - 65s 124ms/step - loss: 0.0620
- accuracy: 0.9802 - val_loss: 0.1235 - val_accuracy: 0.9752
Epoch 30/80
520/520 [=====] - 65s 125ms/step - loss: 0.0675
- accuracy: 0.9791 - val_loss: 0.1133 - val_accuracy: 0.9757
Epoch 31/80
520/520 [=====] - 65s 125ms/step - loss: 0.0534
- accuracy: 0.9834 - val_loss: 0.1082 - val_accuracy: 0.9714
Epoch 32/80
520/520 [=====] - 65s 124ms/step - loss: 0.0587
- accuracy: 0.9820 - val_loss: 0.1000 - val_accuracy: 0.9781
Epoch 33/80
520/520 [=====] - 65s 125ms/step - loss: 0.0576
- accuracy: 0.9819 - val_loss: 0.1143 - val_accuracy: 0.9731
Epoch 34/80
520/520 [=====] - 65s 125ms/step - loss: 0.0477
- accuracy: 0.9852 - val_loss: 0.1190 - val_accuracy: 0.9755
Epoch 35/80
520/520 [=====] - 65s 124ms/step - loss: 0.0555
- accuracy: 0.9837 - val_loss: 0.1030 - val_accuracy: 0.9796
Epoch 36/80
520/520 [=====] - 64s 124ms/step - loss: 0.0510
- accuracy: 0.9836 - val_loss: 0.1049 - val_accuracy: 0.9743
Epoch 37/80
520/520 [=====] - 65s 124ms/step - loss: 0.0544
- accuracy: 0.9829 - val_loss: 0.1022 - val_accuracy: 0.9798
Epoch 38/80
520/520 [=====] - 65s 125ms/step - loss: 0.0456
- accuracy: 0.9857 - val_loss: 0.1265 - val_accuracy: 0.9724
Epoch 39/80
520/520 [=====] - 64s 124ms/step - loss: 0.0640
- accuracy: 0.9800 - val_loss: 0.1152 - val_accuracy: 0.9764

```

Epoch 40/80  
520/520 [=====] - 64s 124ms/step - loss: 0.0579  
- accuracy: 0.9828 - val\_loss: 0.1169 - val\_accuracy: 0.9772  
Epoch 41/80  
520/520 [=====] - 65s 125ms/step - loss: 0.0453  
- accuracy: 0.9857 - val\_loss: 0.1166 - val\_accuracy: 0.9769  
Epoch 42/80  
520/520 [=====] - 65s 126ms/step - loss: 0.0409  
- accuracy: 0.9866 - val\_loss: 0.0945 - val\_accuracy: 0.9800  
Epoch 43/80  
520/520 [=====] - 66s 126ms/step - loss: 0.0480  
- accuracy: 0.9846 - val\_loss: 0.1376 - val\_accuracy: 0.9728  
Epoch 44/80  
520/520 [=====] - 65s 125ms/step - loss: 0.0538  
- accuracy: 0.9837 - val\_loss: 0.1149 - val\_accuracy: 0.9772  
Epoch 45/80  
520/520 [=====] - 64s 124ms/step - loss: 0.0517  
- accuracy: 0.9839 - val\_loss: 0.1134 - val\_accuracy: 0.9776  
Epoch 46/80  
520/520 [=====] - 65s 125ms/step - loss: 0.0554  
- accuracy: 0.9823 - val\_loss: 0.1112 - val\_accuracy: 0.9798  
Epoch 47/80  
520/520 [=====] - 65s 125ms/step - loss: 0.0445  
- accuracy: 0.9868 - val\_loss: 0.1313 - val\_accuracy: 0.9762  
Epoch 48/80  
520/520 [=====] - 66s 126ms/step - loss: 0.0458  
- accuracy: 0.9856 - val\_loss: 0.1019 - val\_accuracy: 0.9820  
Epoch 49/80  
520/520 [=====] - 64s 124ms/step - loss: 0.0483  
- accuracy: 0.9857 - val\_loss: 0.1184 - val\_accuracy: 0.9788  
Epoch 50/80  
520/520 [=====] - 65s 124ms/step - loss: 0.0447  
- accuracy: 0.9862 - val\_loss: 0.1180 - val\_accuracy: 0.9803  
Epoch 51/80  
520/520 [=====] - 65s 124ms/step - loss: 0.0617  
- accuracy: 0.9817 - val\_loss: 0.1025 - val\_accuracy: 0.9829  
Epoch 52/80  
520/520 [=====] - 66s 126ms/step - loss: 0.0402  
- accuracy: 0.9884 - val\_loss: 0.1100 - val\_accuracy: 0.9786  
Epoch 53/80  
520/520 [=====] - 65s 125ms/step - loss: 0.0413  
- accuracy: 0.9871 - val\_loss: 0.1354 - val\_accuracy: 0.9760  
Epoch 54/80  
520/520 [=====] - 65s 125ms/step - loss: 0.0432  
- accuracy: 0.9872 - val\_loss: 0.1696 - val\_accuracy: 0.9728  
Epoch 55/80  
520/520 [=====] - 66s 126ms/step - loss: 0.0505  
- accuracy: 0.9861 - val\_loss: 0.0931 - val\_accuracy: 0.9798  
Epoch 56/80  
520/520 [=====] - 65s 124ms/step - loss: 0.0482  
- accuracy: 0.9868 - val\_loss: 0.1126 - val\_accuracy: 0.9805  
Epoch 57/80  
520/520 [=====] - 65s 125ms/step - loss: 0.0448  
- accuracy: 0.9869 - val\_loss: 0.1158 - val\_accuracy: 0.9781  
Epoch 58/80  
520/520 [=====] - 65s 125ms/step - loss: 0.0393  
- accuracy: 0.9879 - val\_loss: 0.1182 - val\_accuracy: 0.9776  
Epoch 59/80  
520/520 [=====] - 65s 125ms/step - loss: 0.0454  
- accuracy: 0.9868 - val\_loss: 0.1186 - val\_accuracy: 0.9764  
Epoch 60/80  
520/520 [=====] - 65s 124ms/step - loss: 0.0479  
- accuracy: 0.9846 - val\_loss: 0.1409 - val\_accuracy: 0.9772  
Epoch 61/80

```

520/520 [=====] - 65s 124ms/step - loss: 0.0493
- accuracy: 0.9854 - val_loss: 0.1349 - val_accuracy: 0.9796
Epoch 62/80
520/520 [=====] - 65s 125ms/step - loss: 0.0505
- accuracy: 0.9852 - val_loss: 0.1006 - val_accuracy: 0.9788
Epoch 63/80
520/520 [=====] - 64s 123ms/step - loss: 0.0407
- accuracy: 0.9889 - val_loss: 0.1103 - val_accuracy: 0.9829
Epoch 64/80
520/520 [=====] - 65s 125ms/step - loss: 0.0420
- accuracy: 0.9880 - val_loss: 0.1119 - val_accuracy: 0.9796
Epoch 65/80
520/520 [=====] - 64s 124ms/step - loss: 0.0447
- accuracy: 0.9878 - val_loss: 0.1254 - val_accuracy: 0.9803
Epoch 66/80
520/520 [=====] - 64s 124ms/step - loss: 0.0487
- accuracy: 0.9850 - val_loss: 0.1230 - val_accuracy: 0.9776
Epoch 67/80
520/520 [=====] - 65s 125ms/step - loss: 0.0389
- accuracy: 0.9892 - val_loss: 0.1193 - val_accuracy: 0.9793
Epoch 68/80
520/520 [=====] - 64s 123ms/step - loss: 0.0450
- accuracy: 0.9880 - val_loss: 0.1053 - val_accuracy: 0.9817
Epoch 69/80
520/520 [=====] - 64s 124ms/step - loss: 0.0436
- accuracy: 0.9877 - val_loss: 0.1146 - val_accuracy: 0.9781
Epoch 70/80
520/520 [=====] - 64s 124ms/step - loss: 0.0444
- accuracy: 0.9878 - val_loss: 0.1198 - val_accuracy: 0.9796
Epoch 71/80
520/520 [=====] - 64s 124ms/step - loss: 0.0447
- accuracy: 0.9869 - val_loss: 0.1110 - val_accuracy: 0.9851
Epoch 72/80
520/520 [=====] - 65s 125ms/step - loss: 0.0418
- accuracy: 0.9885 - val_loss: 0.1163 - val_accuracy: 0.9774
Epoch 73/80
520/520 [=====] - 64s 124ms/step - loss: 0.0360
- accuracy: 0.9894 - val_loss: 0.1301 - val_accuracy: 0.9788
Epoch 74/80
520/520 [=====] - 65s 124ms/step - loss: 0.0388
- accuracy: 0.9895 - val_loss: 0.1655 - val_accuracy: 0.9748
Epoch 75/80
520/520 [=====] - 64s 124ms/step - loss: 0.0436
- accuracy: 0.9878 - val_loss: 0.1136 - val_accuracy: 0.9817
Epoch 76/80
520/520 [=====] - 65s 125ms/step - loss: 0.0432
- accuracy: 0.9886 - val_loss: 0.1389 - val_accuracy: 0.9769
Epoch 77/80
520/520 [=====] - 64s 124ms/step - loss: 0.0392
- accuracy: 0.9889 - val_loss: 0.1570 - val_accuracy: 0.9762
Epoch 78/80
520/520 [=====] - 64s 124ms/step - loss: 0.0457
- accuracy: 0.9876 - val_loss: 0.1249 - val_accuracy: 0.9800
Epoch 79/80
520/520 [=====] - 65s 124ms/step - loss: 0.0383
- accuracy: 0.9891 - val_loss: 0.1271 - val_accuracy: 0.9832
Epoch 80/80
520/520 [=====] - 65s 124ms/step - loss: 0.0506
- accuracy: 0.9876 - val_loss: 0.1213 - val_accuracy: 0.9815

```

Berikut adalah hasil *output* proses pembelajarannya untuk tiap epoch jika menggunakan 100 epoch.

Epoch 1/100

```

520/520 [=====] - 66s 125ms/step - loss: 1.7581
- accuracy: 0.4687 - val_loss: 0.6520 - val_accuracy: 0.8072
Epoch 2/100
520/520 [=====] - 65s 124ms/step - loss: 0.5042
- accuracy: 0.8395 - val_loss: 0.3739 - val_accuracy: 0.8829
Epoch 3/100
520/520 [=====] - 64s 124ms/step - loss: 0.3116
- accuracy: 0.8993 - val_loss: 0.2311 - val_accuracy: 0.9332
Epoch 4/100
520/520 [=====] - 64s 124ms/step - loss: 0.2339
- accuracy: 0.9237 - val_loss: 0.2124 - val_accuracy: 0.9416
Epoch 5/100
520/520 [=====] - 64s 124ms/step - loss: 0.1945
- accuracy: 0.9376 - val_loss: 0.1718 - val_accuracy: 0.9505
Epoch 6/100
520/520 [=====] - 65s 124ms/step - loss: 0.1618
- accuracy: 0.9474 - val_loss: 0.1446 - val_accuracy: 0.9591
Epoch 7/100
520/520 [=====] - 64s 123ms/step - loss: 0.1417
- accuracy: 0.9550 - val_loss: 0.1356 - val_accuracy: 0.9577
Epoch 8/100
520/520 [=====] - 65s 124ms/step - loss: 0.1262
- accuracy: 0.9590 - val_loss: 0.1247 - val_accuracy: 0.9673
Epoch 9/100
520/520 [=====] - 65s 125ms/step - loss: 0.1112
- accuracy: 0.9642 - val_loss: 0.1296 - val_accuracy: 0.9678
Epoch 10/100
520/520 [=====] - 65s 124ms/step - loss: 0.1150
- accuracy: 0.9610 - val_loss: 0.1306 - val_accuracy: 0.9623
Epoch 11/100
520/520 [=====] - 65s 125ms/step - loss: 0.1008
- accuracy: 0.9681 - val_loss: 0.1168 - val_accuracy: 0.9685
Epoch 12/100
520/520 [=====] - 65s 124ms/step - loss: 0.0989
- accuracy: 0.9677 - val_loss: 0.1472 - val_accuracy: 0.9594
Epoch 13/100
520/520 [=====] - 65s 125ms/step - loss: 0.0906
- accuracy: 0.9701 - val_loss: 0.1074 - val_accuracy: 0.9721
Epoch 14/100
520/520 [=====] - 65s 125ms/step - loss: 0.0776
- accuracy: 0.9745 - val_loss: 0.0900 - val_accuracy: 0.9767
Epoch 15/100
520/520 [=====] - 65s 126ms/step - loss: 0.0795
- accuracy: 0.9748 - val_loss: 0.1038 - val_accuracy: 0.9716
Epoch 16/100
520/520 [=====] - 65s 125ms/step - loss: 0.0784
- accuracy: 0.9764 - val_loss: 0.0897 - val_accuracy: 0.9755
Epoch 17/100
520/520 [=====] - 64s 124ms/step - loss: 0.0682
- accuracy: 0.9787 - val_loss: 0.1279 - val_accuracy: 0.9678
Epoch 18/100
520/520 [=====] - 64s 124ms/step - loss: 0.0713
- accuracy: 0.9767 - val_loss: 0.1001 - val_accuracy: 0.9743
Epoch 19/100
520/520 [=====] - 64s 124ms/step - loss: 0.0731
- accuracy: 0.9788 - val_loss: 0.1001 - val_accuracy: 0.9757
Epoch 20/100
520/520 [=====] - 65s 125ms/step - loss: 0.0676
- accuracy: 0.9794 - val_loss: 0.0987 - val_accuracy: 0.9719
Epoch 21/100
520/520 [=====] - 65s 124ms/step - loss: 0.0631
- accuracy: 0.9800 - val_loss: 0.1155 - val_accuracy: 0.9750
Epoch 22/100
520/520 [=====] - 64s 124ms/step - loss: 0.0608
- accuracy: 0.9808 - val_loss: 0.1104 - val_accuracy: 0.9738

```

```

Epoch 23/100
520/520 [=====] - 65s 124ms/step - loss: 0.0637
- accuracy: 0.9800 - val_loss: 0.1322 - val_accuracy: 0.9712
Epoch 24/100
520/520 [=====] - 64s 123ms/step - loss: 0.0716
- accuracy: 0.9797 - val_loss: 0.0868 - val_accuracy: 0.9776
Epoch 25/100
520/520 [=====] - 65s 125ms/step - loss: 0.0602
- accuracy: 0.9798 - val_loss: 0.0878 - val_accuracy: 0.9774
Epoch 26/100
520/520 [=====] - 65s 124ms/step - loss: 0.0475
- accuracy: 0.9848 - val_loss: 0.0832 - val_accuracy: 0.9788
Epoch 27/100
520/520 [=====] - 65s 124ms/step - loss: 0.0672
- accuracy: 0.9794 - val_loss: 0.0962 - val_accuracy: 0.9796
Epoch 28/100
520/520 [=====] - 65s 125ms/step - loss: 0.0566
- accuracy: 0.9823 - val_loss: 0.0954 - val_accuracy: 0.9755
Epoch 29/100
520/520 [=====] - 64s 124ms/step - loss: 0.0562
- accuracy: 0.9812 - val_loss: 0.0971 - val_accuracy: 0.9786
Epoch 30/100
520/520 [=====] - 66s 126ms/step - loss: 0.0446
- accuracy: 0.9864 - val_loss: 0.1082 - val_accuracy: 0.9774
Epoch 31/100
520/520 [=====] - 65s 125ms/step - loss: 0.0505
- accuracy: 0.9840 - val_loss: 0.1218 - val_accuracy: 0.9757
Epoch 32/100
520/520 [=====] - 65s 125ms/step - loss: 0.0455
- accuracy: 0.9858 - val_loss: 0.1027 - val_accuracy: 0.9812
Epoch 33/100
520/520 [=====] - 65s 125ms/step - loss: 0.0475
- accuracy: 0.9850 - val_loss: 0.1190 - val_accuracy: 0.9772
Epoch 34/100
520/520 [=====] - 65s 124ms/step - loss: 0.0524
- accuracy: 0.9837 - val_loss: 0.0955 - val_accuracy: 0.9767
Epoch 35/100
520/520 [=====] - 65s 125ms/step - loss: 0.0465
- accuracy: 0.9865 - val_loss: 0.1080 - val_accuracy: 0.9793
Epoch 36/100
520/520 [=====] - 65s 125ms/step - loss: 0.0424
- accuracy: 0.9876 - val_loss: 0.1100 - val_accuracy: 0.9745
Epoch 37/100
520/520 [=====] - 65s 125ms/step - loss: 0.0550
- accuracy: 0.9834 - val_loss: 0.1005 - val_accuracy: 0.9800
Epoch 38/100
520/520 [=====] - 65s 125ms/step - loss: 0.0517
- accuracy: 0.9842 - val_loss: 0.0852 - val_accuracy: 0.9798
Epoch 39/100
520/520 [=====] - 65s 125ms/step - loss: 0.0444
- accuracy: 0.9862 - val_loss: 0.1078 - val_accuracy: 0.9827
Epoch 40/100
520/520 [=====] - 65s 125ms/step - loss: 0.0443
- accuracy: 0.9867 - val_loss: 0.0958 - val_accuracy: 0.9793
Epoch 41/100
520/520 [=====] - 65s 124ms/step - loss: 0.0427
- accuracy: 0.9867 - val_loss: 0.1064 - val_accuracy: 0.9750
Epoch 42/100
520/520 [=====] - 65s 125ms/step - loss: 0.0453
- accuracy: 0.9863 - val_loss: 0.0964 - val_accuracy: 0.9781
Epoch 43/100
520/520 [=====] - 65s 124ms/step - loss: 0.0509
- accuracy: 0.9858 - val_loss: 0.1247 - val_accuracy: 0.9764
Epoch 44/100

```

```

520/520 [=====] - 65s 126ms/step - loss: 0.0394
- accuracy: 0.9882 - val_loss: 0.1134 - val_accuracy: 0.9805
Epoch 45/100
520/520 [=====] - 65s 125ms/step - loss: 0.0330
- accuracy: 0.9898 - val_loss: 0.1250 - val_accuracy: 0.9839
Epoch 46/100
520/520 [=====] - 64s 124ms/step - loss: 0.0430
- accuracy: 0.9884 - val_loss: 0.1043 - val_accuracy: 0.9817
Epoch 47/100
520/520 [=====] - 65s 125ms/step - loss: 0.0376
- accuracy: 0.9886 - val_loss: 0.1052 - val_accuracy: 0.9796
Epoch 48/100
520/520 [=====] - 65s 125ms/step - loss: 0.0492
- accuracy: 0.9861 - val_loss: 0.0933 - val_accuracy: 0.9788
Epoch 49/100
520/520 [=====] - 65s 125ms/step - loss: 0.0408
- accuracy: 0.9884 - val_loss: 0.1233 - val_accuracy: 0.9767
Epoch 50/100
520/520 [=====] - 65s 124ms/step - loss: 0.0371
- accuracy: 0.9885 - val_loss: 0.0888 - val_accuracy: 0.9832
Epoch 51/100
520/520 [=====] - 65s 125ms/step - loss: 0.0364
- accuracy: 0.9894 - val_loss: 0.1051 - val_accuracy: 0.9839
Epoch 52/100
520/520 [=====] - 65s 125ms/step - loss: 0.0385
- accuracy: 0.9891 - val_loss: 0.1206 - val_accuracy: 0.9757
Epoch 53/100
520/520 [=====] - 65s 125ms/step - loss: 0.0429
- accuracy: 0.9878 - val_loss: 0.1091 - val_accuracy: 0.9788
Epoch 54/100
520/520 [=====] - 65s 126ms/step - loss: 0.0364
- accuracy: 0.9888 - val_loss: 0.1238 - val_accuracy: 0.9791
Epoch 55/100
520/520 [=====] - 65s 125ms/step - loss: 0.0401
- accuracy: 0.9880 - val_loss: 0.1261 - val_accuracy: 0.9815
Epoch 56/100
520/520 [=====] - 64s 124ms/step - loss: 0.0442
- accuracy: 0.9883 - val_loss: 0.1115 - val_accuracy: 0.9817
Epoch 57/100
520/520 [=====] - 65s 125ms/step - loss: 0.0404
- accuracy: 0.9889 - val_loss: 0.1126 - val_accuracy: 0.9820
Epoch 58/100
520/520 [=====] - 65s 124ms/step - loss: 0.0368
- accuracy: 0.9904 - val_loss: 0.1080 - val_accuracy: 0.9800
Epoch 59/100
520/520 [=====] - 66s 126ms/step - loss: 0.0384
- accuracy: 0.9884 - val_loss: 0.1092 - val_accuracy: 0.9810
Epoch 60/100
520/520 [=====] - 64s 124ms/step - loss: 0.0365
- accuracy: 0.9897 - val_loss: 0.1072 - val_accuracy: 0.9815
Epoch 61/100
520/520 [=====] - 65s 125ms/step - loss: 0.0381
- accuracy: 0.9897 - val_loss: 0.1336 - val_accuracy: 0.9791
Epoch 62/100
520/520 [=====] - 65s 124ms/step - loss: 0.0343
- accuracy: 0.9906 - val_loss: 0.1114 - val_accuracy: 0.9849
Epoch 63/100
520/520 [=====] - 65s 125ms/step - loss: 0.0398
- accuracy: 0.9889 - val_loss: 0.1039 - val_accuracy: 0.9810
Epoch 64/100
520/520 [=====] - 65s 125ms/step - loss: 0.0410
- accuracy: 0.9890 - val_loss: 0.1038 - val_accuracy: 0.9815
Epoch 65/100
520/520 [=====] - 65s 124ms/step - loss: 0.0302
- accuracy: 0.9906 - val_loss: 0.1240 - val_accuracy: 0.9805

```

```

Epoch 66/100
520/520 [=====] - 65s 124ms/step - loss: 0.0393
- accuracy: 0.9901 - val_loss: 0.1233 - val_accuracy: 0.9827
Epoch 67/100
520/520 [=====] - 64s 124ms/step - loss: 0.0426
- accuracy: 0.9895 - val_loss: 0.1534 - val_accuracy: 0.9779
Epoch 68/100
520/520 [=====] - 65s 125ms/step - loss: 0.0379
- accuracy: 0.9891 - val_loss: 0.1133 - val_accuracy: 0.9827
Epoch 69/100
520/520 [=====] - 65s 125ms/step - loss: 0.0454
- accuracy: 0.9882 - val_loss: 0.1243 - val_accuracy: 0.9825
Epoch 70/100
520/520 [=====] - 65s 124ms/step - loss: 0.0491
- accuracy: 0.9874 - val_loss: 0.1514 - val_accuracy: 0.9791
Epoch 71/100
520/520 [=====] - 65s 125ms/step - loss: 0.0376
- accuracy: 0.9902 - val_loss: 0.0933 - val_accuracy: 0.9853
Epoch 72/100
520/520 [=====] - 65s 124ms/step - loss: 0.0375
- accuracy: 0.9898 - val_loss: 0.0979 - val_accuracy: 0.9820
Epoch 73/100
520/520 [=====] - 65s 126ms/step - loss: 0.0343
- accuracy: 0.9904 - val_loss: 0.1263 - val_accuracy: 0.9815
Epoch 74/100
520/520 [=====] - 65s 125ms/step - loss: 0.0361
- accuracy: 0.9902 - val_loss: 0.1472 - val_accuracy: 0.9812
Epoch 75/100
520/520 [=====] - 65s 124ms/step - loss: 0.0354
- accuracy: 0.9904 - val_loss: 0.1094 - val_accuracy: 0.9849
Epoch 76/100
520/520 [=====] - 65s 124ms/step - loss: 0.0383
- accuracy: 0.9904 - val_loss: 0.1520 - val_accuracy: 0.9791
Epoch 77/100
520/520 [=====] - 65s 125ms/step - loss: 0.0436
- accuracy: 0.9896 - val_loss: 0.1215 - val_accuracy: 0.9812
Epoch 78/100
520/520 [=====] - 65s 126ms/step - loss: 0.0313
- accuracy: 0.9915 - val_loss: 0.1210 - val_accuracy: 0.9844
Epoch 79/100
520/520 [=====] - 64s 124ms/step - loss: 0.0414
- accuracy: 0.9891 - val_loss: 0.1305 - val_accuracy: 0.9796
Epoch 80/100
520/520 [=====] - 65s 125ms/step - loss: 0.0441
- accuracy: 0.9890 - val_loss: 0.1214 - val_accuracy: 0.9791
Epoch 81/100
520/520 [=====] - 65s 125ms/step - loss: 0.0357
- accuracy: 0.9906 - val_loss: 0.1152 - val_accuracy: 0.9827
Epoch 82/100
520/520 [=====] - 65s 125ms/step - loss: 0.0450
- accuracy: 0.9890 - val_loss: 0.0948 - val_accuracy: 0.9834
Epoch 83/100
520/520 [=====] - 65s 125ms/step - loss: 0.0337
- accuracy: 0.9910 - val_loss: 0.1264 - val_accuracy: 0.9815
Epoch 84/100
520/520 [=====] - 65s 125ms/step - loss: 0.0316
- accuracy: 0.9910 - val_loss: 0.1450 - val_accuracy: 0.9808
Epoch 85/100
520/520 [=====] - 65s 124ms/step - loss: 0.0415
- accuracy: 0.9904 - val_loss: 0.1179 - val_accuracy: 0.9841
Epoch 86/100
520/520 [=====] - 65s 125ms/step - loss: 0.0449
- accuracy: 0.9884 - val_loss: 0.1739 - val_accuracy: 0.9772
Epoch 87/100

```

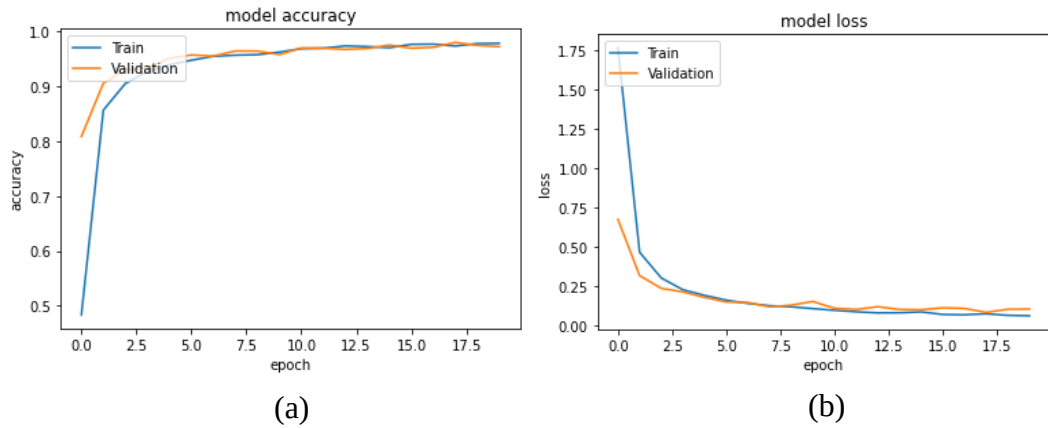


```

520/520 [=====] - 65s 125ms/step - loss: 0.0461
- accuracy: 0.9893 - val_loss: 0.1303 - val_accuracy: 0.9803
Epoch 88/100
520/520 [=====] - 65s 125ms/step - loss: 0.0358
- accuracy: 0.9907 - val_loss: 0.1197 - val_accuracy: 0.9837
Epoch 89/100
520/520 [=====] - 65s 124ms/step - loss: 0.0415
- accuracy: 0.9893 - val_loss: 0.1703 - val_accuracy: 0.9704
Epoch 90/100
520/520 [=====] - 65s 124ms/step - loss: 0.0380
- accuracy: 0.9906 - val_loss: 0.1199 - val_accuracy: 0.9808
Epoch 91/100
520/520 [=====] - 65s 125ms/step - loss: 0.0363
- accuracy: 0.9910 - val_loss: 0.1691 - val_accuracy: 0.9815
Epoch 92/100
520/520 [=====] - 65s 125ms/step - loss: 0.0496
- accuracy: 0.9892 - val_loss: 0.1502 - val_accuracy: 0.9808
Epoch 93/100
520/520 [=====] - 65s 125ms/step - loss: 0.0486
- accuracy: 0.9892 - val_loss: 0.1250 - val_accuracy: 0.9827
Epoch 94/100
520/520 [=====] - 65s 124ms/step - loss: 0.0273
- accuracy: 0.9930 - val_loss: 0.1522 - val_accuracy: 0.9798
Epoch 95/100
520/520 [=====] - 65s 125ms/step - loss: 0.0288
- accuracy: 0.9921 - val_loss: 0.1679 - val_accuracy: 0.9791
Epoch 96/100
520/520 [=====] - 65s 125ms/step - loss: 0.0530
- accuracy: 0.9893 - val_loss: 0.1323 - val_accuracy: 0.9829
Epoch 97/100
520/520 [=====] - 65s 125ms/step - loss: 0.0401
- accuracy: 0.9899 - val_loss: 0.1437 - val_accuracy: 0.9834
Epoch 98/100
520/520 [=====] - 65s 124ms/step - loss: 0.0280
- accuracy: 0.9925 - val_loss: 0.1256 - val_accuracy: 0.9829
Epoch 99/100
520/520 [=====] - 65s 125ms/step - loss: 0.0509
- accuracy: 0.9885 - val_loss: 0.1868 - val_accuracy: 0.9796
Epoch 100/100
520/520 [=====] - 65s 125ms/step - loss: 0.0419
- accuracy: 0.9907 - val_loss: 0.1776 - val_accuracy: 0.9820

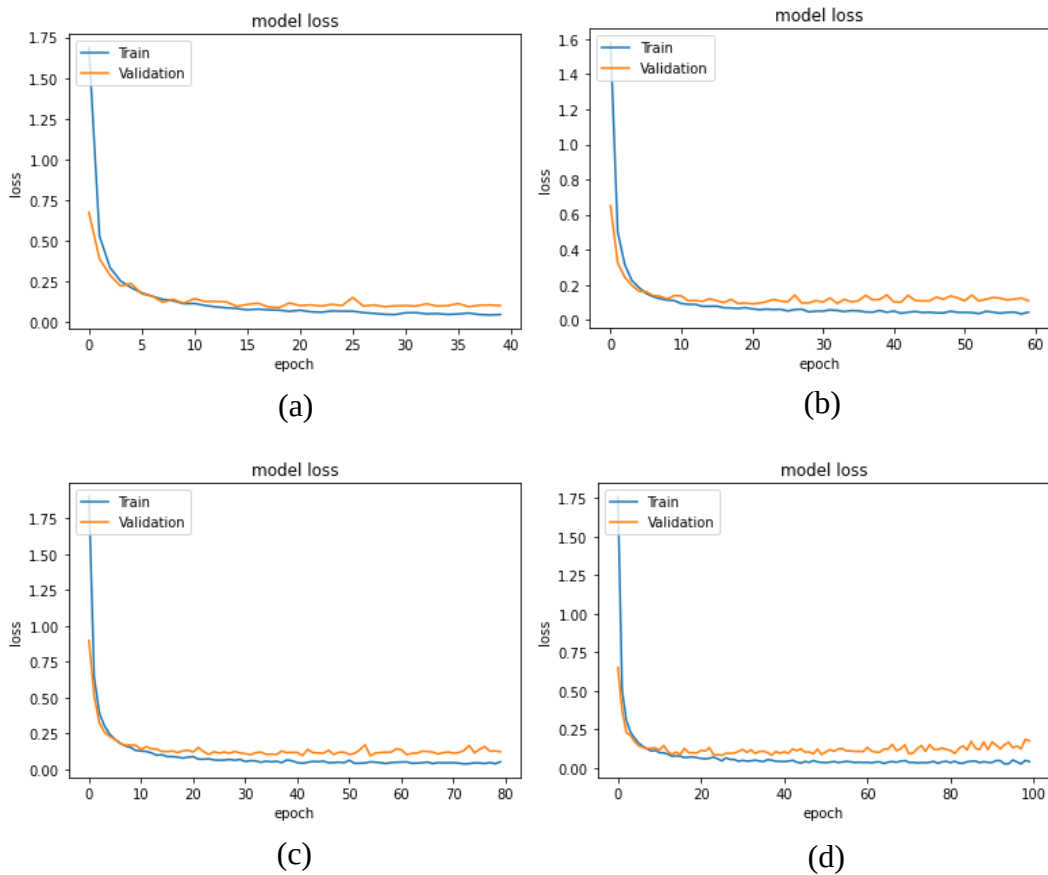
```

Jika proses hasil pembelajaran untuk 20, 40, 60, 80, 100 epoch digambarkan dalam bentuk grafik akan diperoleh grafik yang hasilnya dapat dilihat pada Gambar 4.11.



**Gambar 4.11** (a) dan (b) Hasil *Output* Pembelajaran Sistem Dengan *Convex Hull* + CNN Pada Skenario 4 Dengan 20 Epoch

*Model loss* hasil pembelajaran untuk epoch 40 sampai 100 bisa dilihat pada Gambar 4.12.



**Gambar 4.12** (a), (b), (c), dan (d) Hasil *Output* Pembelajaran Sistem Dengan *Convex Hull* + CNN Pada Skenario 4 Dengan Epoch 40, 60, 80, dan 100

Setelah model dibuat, selanjutnya akan dilakukan prediksi dan evaluasi terhadap model pembelajaran CNN untuk tiap epoch.

#### 4.5.4.3 Hasil Evaluasi Sistem CNN Tanpa Convex Hull

Setelah file model hasil pembelajaran dihasilkan, selanjutnya akan dilakukan prediksi dan evaluasi pada data *testing* dengan menggunakan model hasil hasil pembelajaran tersebut untuk masing-masing epoch. Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 20 epoch bisa dilihat pada Tabel 4.13. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya pada evaluasi model dengan 20 epoch dengan menggunakan fungsi *classification\_report*.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| a            | 0.97      | 1.00   | 0.99     | 200     |
| b            | 0.95      | 0.93   | 0.94     | 200     |
| c            | 1.00      | 1.00   | 1.00     | 200     |
| d            | 0.95      | 0.96   | 0.96     | 200     |
| e            | 1.00      | 1.00   | 1.00     | 200     |
| f            | 0.98      | 0.99   | 0.99     | 200     |
| g            | 0.99      | 0.97   | 0.98     | 200     |
| h            | 1.00      | 1.00   | 1.00     | 200     |
| i            | 0.96      | 0.94   | 0.95     | 200     |
| j            | 1.00      | 1.00   | 1.00     | 200     |
| k            | 0.98      | 0.96   | 0.97     | 200     |
| l            | 0.97      | 0.97   | 0.97     | 200     |
| m            | 1.00      | 0.99   | 0.99     | 200     |
| n            | 1.00      | 1.00   | 1.00     | 200     |
| o            | 0.99      | 1.00   | 0.99     | 200     |
| p            | 0.99      | 1.00   | 1.00     | 200     |
| q            | 1.00      | 1.00   | 1.00     | 200     |
| r            | 0.99      | 0.97   | 0.98     | 200     |
| s            | 1.00      | 1.00   | 1.00     | 200     |
| t            | 1.00      | 1.00   | 1.00     | 200     |
| u            | 0.99      | 1.00   | 0.99     | 200     |
| v            | 0.95      | 0.95   | 0.95     | 200     |
| w            | 0.95      | 0.95   | 0.95     | 200     |
| x            | 1.00      | 1.00   | 1.00     | 200     |
| y            | 1.00      | 0.98   | 0.99     | 200     |
| z            | 1.00      | 1.00   | 1.00     | 200     |
| accuracy     |           |        | 0.98     | 5200    |
| macro avg    | 0.98      | 0.98   | 0.98     | 5200    |
| weighted avg | 0.98      | 0.98   | 0.98     | 5200    |

Accuracty : 0.9838461538461538

Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 40 epoch bisa dilihat pada Tabel 4.14. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya pada evaluasi model dengan 40 epoch dengan menggunakan fungsi *classification\_report*.

|  | precision | recall | f1-score | support |
|--|-----------|--------|----------|---------|
|--|-----------|--------|----------|---------|

|              |      |      |      |      |
|--------------|------|------|------|------|
| a            | 0.93 | 0.99 | 0.96 | 200  |
| b            | 0.95 | 0.92 | 0.93 | 200  |
| c            | 0.99 | 0.99 | 0.99 | 200  |
| d            | 0.95 | 0.97 | 0.96 | 200  |
| e            | 1.00 | 0.99 | 1.00 | 200  |
| f            | 0.93 | 0.99 | 0.96 | 200  |
| g            | 0.99 | 0.94 | 0.97 | 200  |
| h            | 1.00 | 1.00 | 1.00 | 200  |
| i            | 0.96 | 0.93 | 0.94 | 200  |
| j            | 1.00 | 1.00 | 1.00 | 200  |
| k            | 0.98 | 0.92 | 0.95 | 200  |
| l            | 0.97 | 0.97 | 0.97 | 200  |
| m            | 1.00 | 0.99 | 1.00 | 200  |
| n            | 1.00 | 1.00 | 1.00 | 200  |
| o            | 0.99 | 0.99 | 0.99 | 200  |
| p            | 1.00 | 1.00 | 1.00 | 200  |
| q            | 1.00 | 0.99 | 1.00 | 200  |
| r            | 0.94 | 0.98 | 0.96 | 200  |
| s            | 0.99 | 1.00 | 1.00 | 200  |
| t            | 1.00 | 1.00 | 1.00 | 200  |
| u            | 0.99 | 1.00 | 1.00 | 200  |
| v            | 0.97 | 0.94 | 0.96 | 200  |
| w            | 0.95 | 0.96 | 0.96 | 200  |
| x            | 1.00 | 1.00 | 1.00 | 200  |
| y            | 1.00 | 0.99 | 1.00 | 200  |
| z            | 1.00 | 1.00 | 1.00 | 200  |
| accuracy     |      |      | 0.98 | 5200 |
| macro avg    | 0.98 | 0.98 | 0.98 | 5200 |
| weighted avg | 0.98 | 0.98 | 0.98 | 5200 |

Accuracty : 0.9807692307692307

Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 60 epoch bisa dilihat pada Tabel 4.15. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya pada evaluasi model dengan 60 epoch dengan menggunakan fungsi *classification\_report*.

|   | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| a | 0.99      | 1.00   | 1.00     | 200     |
| b | 0.93      | 0.86   | 0.90     | 200     |
| c | 1.00      | 1.00   | 1.00     | 200     |
| d | 0.99      | 0.96   | 0.98     | 200     |
| e | 1.00      | 1.00   | 1.00     | 200     |
| f | 0.97      | 0.99   | 0.98     | 200     |
| g | 0.99      | 0.94   | 0.97     | 200     |
| h | 1.00      | 1.00   | 1.00     | 200     |
| i | 0.98      | 0.95   | 0.97     | 200     |
| j | 1.00      | 1.00   | 1.00     | 200     |
| k | 0.98      | 0.98   | 0.98     | 200     |
| l | 0.97      | 0.97   | 0.97     | 200     |
| m | 1.00      | 1.00   | 1.00     | 200     |
| n | 1.00      | 1.00   | 1.00     | 200     |
| o | 0.98      | 1.00   | 0.99     | 200     |
| p | 1.00      | 1.00   | 1.00     | 200     |

|              |      |      |      |      |
|--------------|------|------|------|------|
| q            | 1.00 | 1.00 | 1.00 | 200  |
| r            | 0.98 | 0.98 | 0.98 | 200  |
| s            | 1.00 | 1.00 | 1.00 | 200  |
| t            | 1.00 | 1.00 | 1.00 | 200  |
| u            | 1.00 | 1.00 | 1.00 | 200  |
| v            | 0.97 | 0.99 | 0.98 | 200  |
| w            | 0.91 | 0.97 | 0.94 | 200  |
| x            | 0.97 | 1.00 | 0.98 | 200  |
| y            | 1.00 | 1.00 | 1.00 | 200  |
| z            | 1.00 | 1.00 | 1.00 | 200  |
| accuracy     |      |      | 0.98 | 5200 |
| macro avg    | 0.99 | 0.98 | 0.98 | 5200 |
| weighted avg | 0.99 | 0.98 | 0.98 | 5200 |

Accuracty : 0.985

Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 80 epoch bisa dilihat pada Tabel 4.16. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya pada evaluasi model dengan 80 epoch dengan menggunakan fungsi *classification\_report*.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| a            | 0.99      | 1.00   | 0.99     | 200     |
| b            | 0.93      | 0.95   | 0.94     | 200     |
| c            | 1.00      | 1.00   | 1.00     | 200     |
| d            | 0.95      | 0.96   | 0.96     | 200     |
| e            | 1.00      | 1.00   | 1.00     | 200     |
| f            | 0.98      | 0.98   | 0.98     | 200     |
| g            | 1.00      | 0.94   | 0.97     | 200     |
| h            | 1.00      | 0.99   | 1.00     | 200     |
| i            | 0.99      | 0.95   | 0.97     | 200     |
| j            | 1.00      | 1.00   | 1.00     | 200     |
| k            | 0.98      | 0.96   | 0.97     | 200     |
| l            | 0.98      | 0.98   | 0.98     | 200     |
| m            | 1.00      | 1.00   | 1.00     | 200     |
| n            | 1.00      | 1.00   | 1.00     | 200     |
| o            | 0.99      | 0.99   | 0.99     | 200     |
| p            | 1.00      | 1.00   | 1.00     | 200     |
| q            | 1.00      | 1.00   | 1.00     | 200     |
| r            | 0.99      | 0.98   | 0.99     | 200     |
| s            | 1.00      | 1.00   | 1.00     | 200     |
| t            | 1.00      | 0.99   | 0.99     | 200     |
| u            | 0.99      | 1.00   | 1.00     | 200     |
| v            | 0.95      | 0.99   | 0.97     | 200     |
| w            | 0.96      | 0.96   | 0.96     | 200     |
| x            | 0.98      | 1.00   | 0.99     | 200     |
| y            | 1.00      | 1.00   | 1.00     | 200     |
| z            | 1.00      | 1.00   | 1.00     | 200     |
| accuracy     |           |        | 0.99     | 5200    |
| macro avg    | 0.99      | 0.99   | 0.99     | 5200    |
| weighted avg | 0.99      | 0.99   | 0.99     | 5200    |

Accuracty : 0.9869230769230769

Untuk tabel hasil prediksi dengan menggunakan fungsi *confusion\_matrix* jika menggunakan 100 epoch bisa dilihat pada Tabel 4.17. Berikut adalah nilai presisi, dan recall untuk tiap kelas dan juga nilai akurasinya pada evaluasi model dengan 80 epoch dengan menggunakan fungsi *classification\_report*.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| a            | 0.98      | 0.98   | 0.98     | 200     |
| b            | 0.95      | 0.91   | 0.93     | 200     |
| c            | 0.99      | 1.00   | 1.00     | 200     |
| d            | 0.99      | 0.94   | 0.96     | 200     |
| e            | 1.00      | 1.00   | 1.00     | 200     |
| f            | 0.99      | 0.99   | 0.99     | 200     |
| g            | 1.00      | 0.98   | 0.99     | 200     |
| h            | 1.00      | 1.00   | 1.00     | 200     |
| i            | 0.96      | 0.96   | 0.96     | 200     |
| j            | 1.00      | 1.00   | 1.00     | 200     |
| k            | 0.98      | 0.97   | 0.98     | 200     |
| l            | 0.96      | 0.98   | 0.97     | 200     |
| m            | 1.00      | 1.00   | 1.00     | 200     |
| n            | 1.00      | 1.00   | 1.00     | 200     |
| o            | 0.99      | 0.99   | 0.99     | 200     |
| p            | 1.00      | 1.00   | 1.00     | 200     |
| q            | 1.00      | 0.99   | 0.99     | 200     |
| r            | 0.99      | 0.99   | 0.99     | 200     |
| s            | 0.99      | 1.00   | 0.99     | 200     |
| t            | 1.00      | 1.00   | 1.00     | 200     |
| u            | 1.00      | 1.00   | 1.00     | 200     |
| v            | 0.95      | 0.98   | 0.96     | 200     |
| w            | 0.93      | 0.97   | 0.95     | 200     |
| x            | 1.00      | 1.00   | 1.00     | 200     |
| y            | 1.00      | 1.00   | 1.00     | 200     |
| z            | 1.00      | 1.00   | 1.00     | 200     |
| accuracy     |           |        | 0.99     | 5200    |
| macro avg    | 0.99      | 0.99   | 0.99     | 5200    |
| weighted avg | 0.99      | 0.99   | 0.99     | 5200    |

Accuracty : 0.9863461538461539

Setelah dilakukan evaluasi dengan menggunakan 20, 40, 60, 80, dan 100 epoch, didapatkan hasil akurasi dari 5x ujicoba sebagai berikut.

```

===== Hasil Rata - Rata =====
Accuracy from epouch = 20 : 0.9838461538461538
Accuracy from epouch = 40 : 0.9807692307692307
Accuracy from epouch = 60 : 0.985
Accuracy from epouch = 80 : 0.9869230769230769
Accuracy from epouch = 100 : 0.9863461538461539
Average accuracy : 0.984576923076923

```

Nilai akurasi rata-rata yang didapatkan adalah 0.984576923076923 atau 98.45%.

## BAB V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Hasil dari penelitian setelah dilakukan pengujian adalah sebagai berikut :

1. Pada pembagian data 60:40, 70:30, dan 80:20 sistem pengenalan bahasa isyarat tangan dengan menggunakan metode *Convex Hull* dan *Convolutional Neural Network* (CNN) mendapatkan akurasi sebesar 0.9854807692307692 atau 98.85%, 0.9774358974358974 atau 97.74%, dan 0.9890384615384615 atau 98.90%. Pada pembagian data 80:20 dengan variasi epoch 20, 40, 60, 80, dan 100 sistem pengenalan bahasa isyarat tangan dengan menggunakan metode *Convex Hull* dan *Convolutional Neural Network* (CNN) mendapatkan akurasi rata-rata sebesar 0.9896923076923076 atau 98.96%
2. Setelah dibandingkan antara sistem pengenalan bahasa isyarat tangan dengan menggunakan metode *Convex Hull* dan *Convolutional Neural Network* (CNN) dengan sistem pengenalan bahasa isyarat tangan yang menggunakan *Convolutional Neural Network* (CNN) saja, didapatkan hasil rata-rata setelah melakukan 5 kali ujicoba dengan variasi epoch 20, 40, 60, 80, dan 100 yaitu sistem pengenalan bahasa isyarat tangan dengan menggunakan metode *Convex Hull* dan *Convolutional Neural Network* (CNN) mendapatkan akurasi rata-rata sebesar 0.9896923076923076 atau 98.96%, sedangkan sistem pengenalan bahasa isyarat tangan yang menggunakan *Convolutional Neural Network* (CNN) saja mendapatkan akurasi rata-rata sebesar 0.984576923076923 atau 98.45%. Hasil tersebut membuktikan bahwa, pengenalan bahasa isyarat tangan dengan menggunakan metode *Convex Hull* dan *Convolutional Neural Network* (CNN) memiliki akurasi yang lebih tinggi sebesar 0.51% jika dibandingkan dengan sistem pengenalan bahasa isyarat tangan yang menggunakan *Convolutional Neural Network* (CNN) saja

## 5.2 Saran

Hasil dari penelitian tentang pengenalan bahasa isyarat tangan dengan menggunakan metode *Convex Hull* dan *Convolutional Neural Network* (CNN) sudah menunjukkan hasil yang lebih baik dibandingkan dengan pengenalan bahasa isyarat tangan dengan menggunakan *Convolutional Neural Network* (CNN) saja. Untuk pengembangan penelitian selanjutnya dapat menambahkan metode *Convexity Defect* pada tahap preprocessing setelah *Convex Hull* yang diharapkan dapat meningkatkan akurasi dengan data yang lebih bervariasi ataupun pengembangan juga dapat dilakukan dengan merubah struktur CNN yang digunakan untuk proses pelatihan data, karena perubahan struktur CNN dapat menghasilkan akurasi yang berbeda.



## DAFTAR PUSTAKA

- [1] Sulfayanti, Dewiani and A. Lawi, A real time alphabets sign language recognition system using hands tracking, 2016.
- [2] Andrian; Muhammad Yunus, Penerjemah Isyarat Indonesia Menggunakan Kamera Telepon Genggam Android, 2016.
- [3] M. B. S. Bakti and Y. M. Pranoto, Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network, 2019.
- [4] K. Alok, A. Mehra, A. Kaushik, A. Verma, K. Alok, A. Mehra, A. Kaushik and A. Verma, Hand Sign Recognition using Convolutional Neural Network, 2020.
- [5] I. Hestningsih, Pengolahan Citra Digital, 2008.
- [6] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2002.
- [7] H. Khaled, S. Gaber, E. S. Mostafa and H. Alli, "Hand gesture recognition using average background and logical heuristic equations," *International Journal of Computers and Technology*, vol. 11, pp. 2635 - 2640, 2013.
- [8] Awodele and Oludele, Neural Networks and its Application in Engineering, 2009.
- [9] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Information Processing Letters*, 1972.
- [10] R. A. Jarvis, " On the identification of the convex hull of a finite set of points in the plane," *Information Processing Letters*, 1973.
- [11] H. S, Neural Network: A Comprehensive Foundation, 1998.
- [12] Gurney K., An Introduction to Neural Networks, UCL Press, 1977.
- [13] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, New jersey: Prentice Hall, 2016.
- [14] M. Mohri, A. Rostamizadeh and A. Talwalkar, Foundations of Machine Learning, MIT Press, 2018.

- [15] Mitchell, Tom, Machine Learning, 1997.
- [16] J. R. Koza, F. H. Bennett, D. Andre and M. A. Keane, "Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design," p. 96, 1996.
- [17] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, The MIT Press, 2016.
- [18] D. Hubel and T. Wiesel, Receptive Fields and Functional Architecture Of Monkey Stiate Cortex, 1968.
- [19] M Zufar and B Setiyono, Convolutional Neural Network untuk Pengenalan Wajah Secara Real Time, 2016.
- [20] V Dumoulin and F Visin, A Guide to Convolution Arithmetic for Deep Learning, 2018.
- [21] A. Dutt and A. Dutt, "Handwritten digit recognition using deep learning," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. VI, no. 7, p. 994, 2017.
- [22] K. P. Danukusumo, Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Citra Candi Berbasis GPU, 2017.
- [23] Chollet, F. and others, Keras: The Python Deep Learning library, Astrophysics Source Code Library, 2018.
- [24] D. Yolanda, K. Gunadi and E. Setyati, Pengenalan Alfabet Bahasa Isyarat Tangan Secara Real-Time dengan Menggunakan Metode Convolutional Neural Network dan Recurrent Neural Network, Universitas Kristen Petra, 2020.
- [25] Amiraj Dhawan and Vipul Honrao, "Implementation of Hand Detection based Techniques for Human Computer Interaction," *International Journal of Computer Applications*, vol. 72, no. 17, 2013.
- [26] S. K. B. A. P. S. M. Caroline El Fiorenza, "Hand Gesture Recognition using Convexity Defect," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 1, 2019.

- [27] E. Prasetyo, Data Mining - Mengolah Data Informasi Menggunakan MATLAB, ANDI Cetakan, 2014.
- [28] J. H. a. M. Kamber, Data mining; Concepts and Techniques Second Edition, 2006.
- [29] F Gorunescu, Data Mining Concepts, Model and Techniques, 2011.

## LAMPIRAN

### Kode Program 4.1 Prepare Data

```
1. import os
2. import shutil
3.
4. os.chdir("/content/drive/MyDrive/Skenario")
5. dirtrain = "Training" . . . 1
6. if(dirtrain not in os.listdir(os.getcwd())):
7.     os.mkdir(dirtrain)
8. dirtest = "Testing"
9. if(dirtest not in os.listdir(os.getcwd())):
10.     os.mkdir(dirtest)
11. character = [chr(i) for i in range(ord('a'), ord('z') + 1)] . . . 2
12. for x in range(len(character)): . . . 3
13.     cdir = character[x]
14.     traindir = os.path.join(dirtrain,cdir)
15.     if(cdir not in os.listdir(dirtrain)):
16.         os.mkdir(traindir)
17.     testdir = os.path.join(dirtest,cdir)
18.     if(cdir not in os.listdir(dirtest)):
19.         os.mkdir(testdir)
20.
21. totdata = 1000 . . . 4
22. ptrain = 80
23. ptest = 20
24. trdata=int((totdata*ptrain)/100)
25. tsdata=int((totdata*ptest)/100)
26. print("Jumlah data train : ",trdata)
27. print("Jumlah data test : ",tsdata)
28.
29. datadir = "/content/drive/MyDrive/Dataset"
30. datalist = os.listdir("/content/drive/MyDrive/Dataset")
31.
32. for x in datalist: . . . 5
33.     fdir = os.path.join(datadir,x)
34.     d = len(os.listdir(fdir))
35.     print("copying ",x,"from ",fdir," for Training data")
36.     for z in range(1,trdata+1):
37.         fnamedir = os.path.join(fdir,x)
38.         fname=fnamedir+str(z)+".jpg"
39.         tardir=os.path.join(dirtrain,x)
40.         tardirname=os.path.join(tardir,x)
41.         tname=tardirname+str(z)+".jpg"
42.         shutil.copyfile(fname,tname)
43.
44. for x in datalist: . . . 6
45.     fdir = os.path.join(datadir,x)
46.     print("copying ",x,"from ",fdir," for Testing data")
47.     for z in range(trdata+1,totdata+1):
48.         fnamedir = os.path.join(fdir,x)
49.         fname=fnamedir+str(z)+".jpg"
50.         tardir=os.path.join(dirtest,x)
51.         tardirname=os.path.join(tardir,x)
52.         tname=tardirname+str(z)+".jpg"
53.         shutil.copyfile(fname,tname)
54.
55. print("complete")
```

## Kode Program 4.2 Preprocessing Data

```
1. import cv2
2. import os
3. import numpy as np
4.
5. def cvh(pts): . . . 1
6.     def get_slope(p1, p2): . . . 2
7.         if p1[0] == p2[0]:
8.             return float('inf')
9.         else:
10.            return 1.0*(p1[1]-p2[1])/(p1[0]-p2[0])
11.     def cmp(a, b): . . . 3
12.         return int(a > b) - int(a < b)
13.     def turn(p1, p2, p3): . . . 4
14.         return cmp((p2[0] - p1[0])*(p3[1] - p1[1]) - (p3[0] -
p1[0])*(p2[1] - p1[1]), 0)
15.
16.     hull=[] . . . 5
17.     pts = sorted(pts)
18.     start = pts.pop(0) . . . 6
19.     pts.sort(key=lambda p: (get_slope(p,start), -p[1],p[0])) 7
20.     hull.append(start) . . . 8
21.
22.     for x in range(len(pts)): . . . 9
23.         hull.append(pts[x])
24.         while len(hull) > 2 and turn(hull[-3],hull[-2],hull[-1]) != 1: . . . 10
25.             hull.pop(-2)
26.
27.     return hull . . . 11
28.
29. def pp(diri,diro): . . . 12
30.     os.chdir("/content/drive/MyDrive/Skenario")
31.     directory = diri
32.     outdirectory = diro
33.     dirlist = os.listdir(directory)
34.     for x in dirlist: . . . 13
35.         y = os.path.join(directory,x)
36.         dirlist2 = os.listdir(y)
37.         for z in dirlist2: . . . 14
38.             path = os.path.join(y,z)
39.             outpath = os.path.join(outdirectory,x)
40.             outpath2 = os.path.join(outpath,z)
41.             img = cv2.imread(path)
42.             img_gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY) . . . 15
43.             blur = cv2.GaussianBlur(img_gray,(1,1),0) . . . 16
44.             canny_output = cv2.Canny(blur, 40, 255) . . . 17
45.             . . . 18
46.             contours, hierarchy =
cv2.findContours(canny_output,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
47.
48.             arr=[] . . . 19
49.             for i in range(len(contours)):
50.                 ca = []
51.                 for j in range(len(contours[i])):
52.                     ca.append([contours[i][j][0][0], contours[i][j][0][1]])
53.                 arr.append(ca)
54.
55.             hull=[] . . . 20
56.             for i in range(len(arr)):
57.                 hull.append(np.array(cvh(arr[i])))
```

```

58.         drawing = np.zeros((canny_output.shape[0],
canny_output.shape[1], 3), dtype=np.uint8) . . . 21
59.         cv2.drawContours(drawing, hull, -1, (255,0,0), 2, 8)
60.         cv2.drawContours(drawing, contours, -1, (0,255,0), -1)
61.
62.         if(outdirectory not in os.listdir()): . . . 22
63.             os.mkdir(outdirectory)
64.         if(x not in os.listdir(outdirectory)):
65.             os.mkdir(outpath)
66.         cv2.imwrite(outpath2, drawing)
67.
68. print("->Preprocessing Training") . . . 23
69. pp("Training", "TrainingPP")
70. print("->Preprocessing Testing ")
71. pp("Testing", "TestingPP")
72. print("->Preprocessing Complete ")

```

### Kode Program 4.3 Make Model & Evaluate

```
1. from tensorflow.keras.layers import Dense, Dropout, Conv2D, MaxPool2D,
   Flatten
2. from tensorflow.keras.models import Sequential
3. from tensorflow.keras.preprocessing.image import ImageDataGenerator
4. import os
5. from keras.utils.vis_utils import plot_model
6. import matplotlib.pyplot as plt
7. from tensorflow.keras.models import model_from_json
8. import numpy as np
9. from sklearn.metrics import confusion_matrix, classification_report
10. import seaborn as sns
11. import pandas as pd
12. import matplotlib
13. import matplotlib.pyplot as plt
14. import os
15. os.chdir("/content/drive/MyDrive/Skenario")
16. #prepare data
17. data_dir = '/content/drive/MyDrive/Skenario/TrainingPP'
18. test_dir = '/content/drive/MyDrive/Skenario/TestingPP'
19. target_size = (100,100)
20. classes = 26
21. train_datagen = ImageDataGenerator(
22.     rescale=1./255,
23.     zoom_range=0.2,
24.     validation_split=0.1) . . . 1
25. train_gen = train_datagen.flow_from_directory(
26.     data_dir,
27.     target_size=target_size,
28.     shuffle=True,
29.     batch_size=32,
30.     color_mode='rgb',
31.     class_mode='categorical',
32.     subset='training')
33. test_generator = ImageDataGenerator(
34.     rescale=1./255) . . . 2
35. test_data_generator = test_generator.flow_from_directory(
36.     test_dir,
37.     target_size=target_size,
38.     batch_size=32,
39.     shuffle=False)
40. val_gen = train_datagen.flow_from_directory(
41.     data_dir,
42.     target_size=target_size,
43.     batch_size=32 ,
44.     color_mode='rgb',
45.     class_mode='categorical',
46.     subset='validation' ) . . . 3
47. def make_model(ep): . . . 4
48.     model = Sequential() . . . 5
49.     model.add(Conv2D(32 , kernel_size=3, strides=1, activation='relu',
   padding='same', input_shape=(100,100,3))) . . . 6
50.     model.add(MaxPool2D(pool_size=(3,3),strides=2)) . . . 7
51.     model.add(Conv2D(64, kernel_size=3, strides=1, activation='relu',
   padding='same'))
52.     model.add(MaxPool2D(pool_size=(2,2),strides=2))
53.     model.add(Conv2D(64, kernel_size=3, strides=1, activation='relu',
   padding='same'))
54.     model.add(MaxPool2D(pool_size=(2,2),strides=2))
55.     model.add(Dropout(0.5)) . . . 8
```

```

56. model.add(Conv2D(128, kernel_size=3, strides=1, activation='relu',
padding='same'))
57. model.add(MaxPool2D(pool_size=(2,2),strides=1))
58. model.add(Flatten()) . . . 9
59. #Fullyconnected
60. model.add(Dropout(0.5))
61. model.add(Dense(512,activation='relu')) . . . 10
62. model.add(Dense(classes,activation='softmax'))
63. model.compile(optimizer='Adam',loss='categorical_crossentropy',metrics=[
'accuracy']) . . . 11
64. plot_model(model, to_file='model_plot.png', show_shapes=True,
show_layer_names=True) . . . 12
65. model.summary() . . . 13
66. #compile
67. history = model.fit(train_gen,epochs=ep,validation_data=val_gen). . . 14
68. # summarize history for accuracy and loss
69. plt.plot(history.history['accuracy'])
70. plt.plot(history.history['val_accuracy'])
71. plt.title('model accuracy')
72. plt.ylabel('accuracy')
73. plt.xlabel('epoch')
74. plt.legend(['Train', 'Validation'], loc='upper left')
75. plt.show()
76. plt.plot(history.history['loss'])
77. plt.plot(history.history['val_loss'])
78. plt.title('model loss')
79. plt.ylabel('loss')
80. plt.xlabel('epoch')
81. plt.legend(['Train', 'Validation'], loc='upper left')
82. plt.show()
83. model_json=model.to_json() . . . 15
84. with open("modeln_json","w") as file:
85.     file.write(model_json)
86. model.save_weights("modeln.h5")
87. print("Model Saved Succesfully")
88. def evaluate_test(): . . . 16
89.     json_file = open('modeln_json','r') . . . 17
90.     loaded_model_json = json_file.read()
91.     json_file.close()
92.     load_model = model_from_json(loaded_model_json)
93.     load_model.load_weights("modeln.h5")
94.     print("Model Loaded")
95.     prediksi = load_model.predict(test_data_generator) . . . 18
96.     y_pred = np.argmax(prediksi,axis=1)
97.     class_labels = list(test_data_generator.class_indices.keys())
98.     print("===== Tabel hasil Prediksi =====")
99.     cm = confusion_matrix(test_data_generator.classes,y_pred) . . . 19
100.    df_cm = pd.DataFrame(cm, index = class_labels,columns = class_labels)
101.    plt.figure(figsize=(15,15)) . . . 20
102.    sns.heatmap(df_cm, annot=True, annot_kws={"size":
9},cmap='viridis',fmt='g')
103.    print("===== Hasil =====")
104.    cr=classification_report(test_data_generator.classes,y_pred,target_names
=class_labels,zero_division=1) . . . 21
105.    print(cr)
106.    crv=classification_report(test_data_generator.classes,y_pred,target_name
s=class_labels,zero_division=1,output_dict=True)
107.    acc = crv['accuracy']
108.    acc_avg.append(acc)
109. acc_avg=[] . . . 22
110. make_model(20)
111. evaluate_test()
112. print("Accuracy : ",acc_avg)

```



Tabel 3.2. Rancangan Tabel Hasil Prediksi

|                    |   | Dikenali |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--------------------|---|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|                    |   | a        | b | c | D | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | V | w | x | y | Z |
| Isyarat Sebenarnya | a | T        | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | b | F        | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | c | F        | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | d | F        | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | e | F        | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | f | F        | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | g | F        | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | h | F        | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | i | F        | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | j | F        | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | k | F        | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | l | F        | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | m | F        | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | n | F        | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F | F |
|                    | o | F        | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F | F |

|  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | p | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F | F |
|  | q | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F | F |
|  | r | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F | F |
|  | s | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F |
|  | t | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F | F |
|  | u | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F | F |
|  | v | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F | F |
|  | w | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F | F |
|  | x | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F | F |
|  | y | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F | F |
|  | z | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | F | T | F |

Tabel 4.2. Tabel Hasil Prediksi Skenario 1 Sistem *Convex Hull* + CNN

|                    |   | Dikenali Oleh CNN Dengan <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |    |   |   |   |   |
|--------------------|---|---------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|----|---|---|---|---|
|                    |   | a                                           | b   | c   | d   | e   | f   | g   | H   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v  | w | x | y | z |
| Isyarat Sebenarnya | a | 390                                         | 0   | 3   | 1   | 0   | 0   | 0   | 0   | 1   | 0   | 2   | 0   | 0   | 0   | 0   | 0 | 0 | 2 | 1 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | b | 0                                           | 381 | 0   | 0   | 0   | 3   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 2   | 0 | 0 | 9 | 0 | 0 | 0 | 0  | 4 | 0 | 0 | 0 |
|                    | c | 0                                           | 0   | 400 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | d | 2                                           | 1   | 0   | 364 | 0   | 0   | 2   | 0   | 0   | 0   | 0   | 7   | 0   | 0   | 2   | 0 | 0 | 1 | 0 | 0 | 0 | 20 | 1 | 0 | 0 | 0 |
|                    | e | 0                                           | 0   | 0   | 0   | 398 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 2 |
|                    | f | 0                                           | 0   | 0   | 0   | 0   | 398 | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 1 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | g | 0                                           | 0   | 0   | 0   | 0   | 0   | 395 | 0   | 1   | 0   | 0   | 1   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 2  | 0 | 0 | 0 | 1 |
|                    | h | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | i | 1                                           | 1   | 0   | 0   | 0   | 3   | 0   | 0   | 381 | 0   | 1   | 6   | 0   | 0   | 1   | 0 | 0 | 1 | 2 | 0 | 0 | 0  | 1 | 0 | 0 | 2 |
|                    | j | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | k | 2                                           | 2   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 393 | 0   | 0   | 0   | 0   | 0 | 0 | 1 | 0 | 0 | 0 | 1  | 0 | 0 | 0 | 0 |
|                    | l | 2                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 397 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | m | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | n | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | o | 3                                           | 0   | 1   | 0   | 0   | 1   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 392 | 0 | 0 | 0 | 0 | 0 | 0 | 1  | 0 | 1 | 0 | 0 |

|  |   | Dikenali Oleh CNN Dengan <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|--|---|---------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|  |   | a                                           | b | c | d | e | f | g | H | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
|  | p | 0                                           | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 399 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|  | q | 0                                           | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 395 | 0   | 0   | 0   | 1   | 1   | 0   | 0   | 0   | 0   |
|  | r | 0                                           | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 395 | 0   | 0   | 0   | 1   | 1   | 0   | 0   | 0   |
|  | s | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 400 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|  | t | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 400 | 0   | 0   | 0   | 0   | 0   | 0   |
|  | u | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 398 | 0   | 0   | 0   | 0   | 0   |
|  | v | 0                                           | 0 | 0 | 3 | 1 | 0 | 2 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0   | 0   | 3   | 0   | 0   | 1   | 380 | 2   | 0   | 0   | 4   |
|  | w | 0                                           | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0   | 0   | 2   | 0   | 0   | 0   | 1   | 394 | 0   | 0   | 0   |
|  | x | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   | 0   |
|  | y | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   |
|  | z | 0                                           | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 399 |

Tabel 4.3. Tabel Hasil Prediksi Skenario 1 Sistem CNN Tanpa *Convex Hull*

|                    |   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |    |    |   |   |     |     |   |   |   |
|--------------------|---|--------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|----|----|---|---|-----|-----|---|---|---|
|                    |   | a                                          | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r  | s  | t | u | v   | w   | x | y | z |
| Isyarat Sebenarnya | a | 372                                        | 1   | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 4   | 0   | 0   | 10  | 0 | 0 | 0  | 11 | 0 | 0 | 0   | 0   | 0 | 0 | 0 |
|                    | b | 2                                          | 225 | 0   | 0   | 0   | 23  | 0   | 0   | 0   | 0   | 4   | 0   | 0   | 0   | 0   | 0 | 0 | 38 | 0  | 0 | 0 | 0   | 107 | 0 | 0 | 1 |
|                    | c | 0                                          | 0   | 359 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 39  | 0 | 0 | 0  | 0  | 0 | 0 | 0   | 0   | 1 | 0 | 0 |
|                    | d | 0                                          | 2   | 0   | 210 | 0   | 20  | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0  | 0  | 0 | 0 | 152 | 2   | 0 | 0 | 0 |
|                    | e | 0                                          | 0   | 0   | 0   | 400 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0  | 0  | 0 | 0 | 0   | 0   | 0 | 0 | 0 |
|                    | f | 0                                          | 5   | 0   | 0   | 0   | 390 | 0   | 0   | 0   | 0   | 1   | 1   | 0   | 0   | 0   | 0 | 0 | 1  | 0  | 0 | 0 | 2   | 0   | 0 | 0 | 0 |
|                    | g | 0                                          | 0   | 0   | 0   | 2   | 0   | 391 | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0  | 0  | 0 | 0 | 2   | 0   | 0 | 0 | 3 |
|                    | h | 0                                          | 0   | 0   | 0   | 2   | 0   | 0   | 398 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0  | 0  | 0 | 0 | 0   | 0   | 0 | 0 | 0 |
|                    | i | 21                                         | 6   | 0   | 5   | 0   | 3   | 1   | 0   | 354 | 0   | 0   | 3   | 0   | 0   | 1   | 0 | 0 | 3  | 0  | 0 | 0 | 3   | 0   | 0 | 0 | 0 |
|                    | j | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0  | 0  | 0 | 0 | 0   | 0   | 0 | 0 | 0 |
|                    | k | 0                                          | 3   | 0   | 0   | 0   | 3   | 0   | 0   | 1   | 0   | 384 | 1   | 0   | 0   | 0   | 0 | 0 | 0  | 0  | 0 | 0 | 7   | 1   | 0 | 0 | 0 |
|                    | l | 5                                          | 2   | 0   | 5   | 0   | 0   | 1   | 0   | 2   | 0   | 9   | 371 | 0   | 0   | 0   | 0 | 2 | 0  | 0  | 0 | 1 | 2   | 0   | 0 | 0 | 0 |
|                    | m | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 398 | 0   | 0   | 0 | 0 | 0  | 0  | 0 | 0 | 0   | 0   | 2 | 0 | 0 |
|                    | n | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   | 0 | 0 | 0  | 0  | 0 | 0 | 0   | 0   | 0 | 0 | 0 |
|                    | o | 0                                          | 0   | 0   | 0   | 0   | 2   | 0   | 1   | 1   | 0   | 1   | 0   | 0   | 0   | 395 | 0 | 0 | 0  | 0  | 0 | 0 | 0   | 0   | 0 | 0 | 0 |

|   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|---|--------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   | a                                          | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| p | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 399 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 400 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r | 1                                          | 2 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0   | 0   | 382 | 0   | 0   | 0   | 2   | 4   | 0   | 0   | 0   |
| s | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 400 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 2   | 0   | 0   | 398 | 0   | 0   | 0   | 0   | 0   | 0   |
| u | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0   | 0   | 0   | 0   | 0   | 398 | 0   | 0   | 0   | 0   | 0   |
| v | 1                                          | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0   | 0   | 1   | 0   | 0   | 0   | 372 | 16  | 0   | 0   | 0   |
| w | 0                                          | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 0   | 0   | 1   | 0   | 0   | 0   | 3   | 388 | 0   | 0   | 0   |
| x | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   | 0   |
| y | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 400 | 0   |
| z | 0                                          | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 399 |

Tabel 4.4. Tabel Hasil Prediksi Skenario 2 Sistem *Convex Hull* + CNN

|                    |   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |    |   |   |   |   |    |   |   |   |   |
|--------------------|---|---------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|----|---|---|---|---|----|---|---|---|---|
|                    |   | a                                           | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q  | r | s | t | u | v  | w | x | y | z |
| Isyarat Sebenarnya | a | 299                                         | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0  | 0 | 1 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | b | 1                                           | 283 | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0  | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | c | 0                                           | 1   | 298 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0  | 0 | 1 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | d | 2                                           | 0   | 0   | 250 | 0   | 0   | 0   | 0   | 0   | 2   | 1   | 5   | 0   | 0   | 1   | 0 | 0  | 1 | 0 | 0 | 0 | 37 | 1 | 0 | 0 | 0 |
|                    | e | 0                                           | 0   | 0   | 0   | 300 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0  | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | f | 0                                           | 2   | 0   | 0   | 0   | 294 | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0  | 0 | 3 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | g | 0                                           | 1   | 0   | 1   | 0   | 0   | 279 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 15 | 0 | 0 | 0 | 0 | 3  | 0 | 0 | 0 | 1 |
|                    | h | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0  | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | i | 7                                           | 1   | 0   | 0   | 1   | 0   | 3   | 0   | 278 | 0   | 0   | 1   | 0   | 0   | 6   | 0 | 0  | 1 | 0 | 1 | 0 | 1  | 0 | 0 | 0 | 0 |
|                    | j | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0   | 0   | 0   | 0   | 0 | 0  | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | k | 0                                           | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 293 | 0   | 0   | 0   | 0   | 0 | 0  | 1 | 0 | 0 | 0 | 3  | 1 | 0 | 0 | 0 |
|                    | l | 2                                           | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 5   | 291 | 0   | 0   | 0   | 0 | 0  | 0 | 1 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | m | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0   | 0 | 0  | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | n | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0 | 0  | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |
|                    | o | 5                                           | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 1   | 0   | 0   | 287 | 0 | 0  | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 | 0 |

|   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|---|---------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   | a                                           | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| p | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 300 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q | 0                                           | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 299 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 298 | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 0   |
| s | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 300 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 300 | 0   | 0   | 0   | 0   | 0   | 0   |
| u | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 298 | 0   | 0   | 0   | 0   | 0   |
| v | 1                                           | 1 | 2 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0   | 0   | 1   | 0   | 0   | 0   | 292 | 0   | 0   | 0   | 0   |
| w | 0                                           | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 3   | 0   | 0   | 0   | 5   | 288 | 0   | 0   | 0   |
| x | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0   |
| y | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 298 | 0   |
| z | 0                                           | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 299 |



Tabel 4.5. Tabel Hasil Prediksi Skenario 2 Sistem CNN Tanpa *Convex Hull*

|                    |   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |    |    |   |   |   |
|--------------------|---|--------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|----|----|---|---|---|
|                    |   | a                                          | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v  | w  | x | y | z |
| Isyarat Sebenarnya | a | 300                                        | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | b | 1                                          | 279 | 0   | 0   | 0   | 4   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 1 | 0 | 0 | 0 | 0  | 15 | 0 | 0 | 0 |
|                    | c | 0                                          | 0   | 299 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | d | 0                                          | 0   | 0   | 218 | 0   | 28  | 0   | 0   | 1   | 0   | 4   | 1   | 0   | 0   | 4   | 0 | 1 | 1 | 0 | 0 | 0 | 42 | 0  | 0 | 0 | 0 |
|                    | e | 0                                          | 0   | 0   | 0   | 300 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | f | 0                                          | 3   | 0   | 0   | 0   | 296 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 1 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | g | 0                                          | 1   | 0   | 1   | 2   | 0   | 287 | 5   | 1   | 0   | 0   | 2   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 1  | 0  | 0 | 0 | 0 |
|                    | h | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | i | 5                                          | 1   | 0   | 5   | 0   | 1   | 1   | 0   | 283 | 0   | 2   | 2   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | j | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | k | 0                                          | 3   | 1   | 0   | 0   | 2   | 0   | 0   | 0   | 0   | 291 | 0   | 0   | 0   | 0   | 0 | 1 | 0 | 0 | 2 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | l | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | m | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | n | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |
|                    | o | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 300 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0 | 0 | 0 |

|   |  | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |    |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |   |     |     |     |     |
|---|--|--------------------------------------------|----|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|---|-----|-----|-----|-----|
|   |  | a                                          | b  | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v | w   | x   | y   | z   |
| p |  | 0                                          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 300 | 0   | 0   | 0   | 0   | 0   | 0 | 0   | 0   | 0   | 0   |
| q |  | 0                                          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 300 | 0   | 0   | 0   | 0   | 0 | 0   | 0   | 0   | 0   |
| r |  | 0                                          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 286 | 0   | 0   | 0   | 0 | 0   | 0   | 0   | 0   |
| s |  | 0                                          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 300 | 0   | 0   | 0 | 0   | 0   | 0   | 0   |
| t |  | 0                                          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 300 | 0   | 0 | 0   | 0   | 0   | 0   |
| u |  | 0                                          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 300 | 0 | 0   | 0   | 0   | 0   |
| v |  | 1                                          | 2  | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 3 | 0 | 0 | 1 | 0   | 0   | 3   | 0   | 0   | 0   | 0 | 286 | 1   | 0   | 0   |
| w |  | 0                                          | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 1   | 0   | 0   | 0   | 1 | 287 | 0   | 0   | 0   |
| x |  | 0                                          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0   | 300 | 0   | 0   |
| y |  | 0                                          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0   | 0   | 300 | 0   |
| z |  | 0                                          | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0   | 0   | 0   | 300 |

Tabel 4.6. Tabel Hasil Prediksi Skenario 3 Sistem *Convex Hull* + CNN

|                    |   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |   |   |   |   |
|--------------------|---|---------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|---|
|                    |   | a                                           | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v | w | x | y | z |
| Isyarat Sebenarnya | a | 198                                         | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | b | 0                                           | 193 | 0   | 0   | 0   | 4   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 |
|                    | c | 0                                           | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | d | 1                                           | 0   | 0   | 193 | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 3   | 0   | 0   | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|                    | e | 0                                           | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | f | 0                                           | 1   | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | g | 0                                           | 0   | 0   | 0   | 0   | 0   | 191 | 0   | 4   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 1 |
|                    | h | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | i | 1                                           | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 195 | 0   | 0   | 2   | 0   | 0   | 0   | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | j | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | k | 0                                           | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 194 | 2   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
|                    | l | 1                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 198 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | m | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | n | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | o | 3                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 3   | 0   | 0   | 0   | 0   | 0   | 194 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|---|---------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   | a                                           | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| p | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r | 0                                           | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 196 | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 0   |
| s | 0                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t | 0                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
| u | 0                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v | 0                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 195 | 1   | 0   | 0   | 0   |
| w | 0                                           | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 197 | 0   | 0   | 0   |
| x | 0                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y | 0                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   |
| z | 0                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |

Tabel 4.7. Tabel Hasil Prediksi Skenario 3 Sistem CNN Tanpa *Convex Hull*

|                    |   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |    |   |   |   |
|--------------------|---|--------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|----|---|---|---|
|                    |   | a                                          | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v | w  | x | y | z |
| Isyarat Sebenarnya | a | 200                                        | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | b | 0                                          | 162 | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 2   | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 26 | 0 | 0 | 0 |
|                    | c | 0                                          | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | d | 0                                          | 0   | 0   | 191 | 0   | 3   | 0   | 2   | 1   | 1   | 0   | 1   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1  | 0 | 0 | 0 |
|                    | e | 0                                          | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | f | 0                                          | 1   | 0   | 0   | 0   | 196 | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0  | 0 | 0 | 0 |
|                    | g | 2                                          | 0   | 0   | 0   | 1   | 0   | 183 | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 0  | 7 | 0 | 0 |
|                    | h | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | i | 2                                          | 0   | 0   | 7   | 0   | 0   | 0   | 0   | 187 | 0   | 0   | 3   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | j | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | k | 0                                          | 2   | 0   | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 191 | 1   | 0   | 0   | 0   | 0 | 0 | 1 | 0 | 0 | 0 | 3 | 0  | 0 | 0 | 0 |
|                    | l | 2                                          | 0   | 0   | 2   | 0   | 0   | 0   | 0   | 1   | 0   | 2   | 193 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | m | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | n | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | o | 2                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 198 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |

|   |  | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|---|--|--------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   |  | a                                          | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| p |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r |  | 0                                          | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0   | 0   | 195 | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 0   |
| s |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
| u |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   |
| w |  | 0                                          | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 3   | 195 | 0   | 0   | 0   |
| x |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   |
| z |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |

Tabel 4.8. Tabel Hasil Prediksi Skenario 4 Sistem *Convex Hull* + CNN Dengan 20 Epoch

|                    |   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |   |   |   |   |
|--------------------|---|---------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|---|
|                    |   | a                                           | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v | w | x | y | z |
| Isyarat Sebenarnya | a | 200                                         | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | b | 1                                           | 198 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | c | 0                                           | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | D | 1                                           | 0   | 0   | 186 | 0   | 1   | 1   | 0   | 1   | 1   | 0   | 5   | 0   | 0   | 0   | 0 | 0 | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
|                    | e | 0                                           | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | f | 0                                           | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | g | 0                                           | 0   | 0   | 1   | 0   | 0   | 198 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|                    | h | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | i | 1                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 192 | 0   | 2   | 0   | 1   | 0   | 2   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
|                    | j | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | k | 1                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 197 | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
|                    | l | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 4   | 195 | 0   | 0   | 0   | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | m | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | n | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | o | 0                                           | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 0   | 0   | 197 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |   |     |
|---|---------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|
|   | a                                           | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z |     |
| p | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0   |
| q | 0                                           | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 197 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0   |
| r | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 197 | 0   | 0   | 0   | 1   | 2   | 0   | 0   | 0 | 0   |
| s | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0   |
| t | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0   |
| u | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0 | 0   |
| v | 1                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 198 | 0   | 0   | 0   | 0 | 0   |
| w | 0                                           | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 2   | 197 | 0   | 0   | 0 | 0   |
| x | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0   |
| y | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 199 | 0 | 0   |
| z | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 100 |



Tabel 4.9. Tabel Hasil Prediksi Skenario 4 Sistem *Convex Hull* + CNN Dengan 40 Epoch

|                    |   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |   |   |    |   |
|--------------------|---|---------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|---|---|----|---|
|                    |   | a                                           | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v | w | x | y  | z |
| Isyarat Sebenarnya | a | 200                                         | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |
|                    | b | 0                                           | 188 | 0   | 0   | 0   | 3   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 5 | 0 | 0  | 0 |
|                    | c | 0                                           | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |
|                    | d | 0                                           | 0   | 0   | 194 | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 3   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0  | 0 |
|                    | e | 0                                           | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |
|                    | f | 0                                           | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |
|                    | G | 0                                           | 0   | 0   | 1   | 0   | 0   | 180 | 0   | 3   | 0   | 0   | 0   | 0   | 0   | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 13 | 0 |
|                    | h | 0                                           | 0   | 0   | 0   | 0   | 0   | 1   | 199 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |
|                    | i | 0                                           | 0   | 0   | 0   | 0   | 2   | 0   | 0   | 194 | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0  | 1 |
|                    | j | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |
|                    | k | 0                                           | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 196 | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0  | 0 |
|                    | l | 0                                           | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 3   | 196 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |
|                    | m | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |
|                    | n | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |
|                    | o | 1                                           | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 198 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 |

|   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|---|---------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   | a                                           | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| p | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q | 0                                           | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 197 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   |
| r | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 198 | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 0   |
| s | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
| u | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 0   |
| w | 0                                           | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 1   | 0   | 0   | 0   | 1   | 197 | 0   | 0   | 0   |
| x | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   |
| z | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |

Tabel 4.10. Tabel Hasil Prediksi Skenario 4 Sistem *Convex Hull* + CNN Dengan 60 Epoch

|                    |   | dikenali oleh CNN dengan <i>convex hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |   |   |   |   |
|--------------------|---|---------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|---|
|                    |   | a                                           | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v | w | x | y | z |
| isyarat sebenarnya | a | 200                                         | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | b | 0                                           | 193 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 3 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 |
|                    | c | 0                                           | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | d | 0                                           | 0   | 0   | 197 | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 1   | 0   | 0   | 0   | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | e | 0                                           | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | f | 0                                           | 1   | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | g | 0                                           | 0   | 0   | 3   | 0   | 0   | 194 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
|                    | h | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | i | 1                                           | 0   | 0   | 1   | 0   | 1   | 0   | 0   | 192 | 0   | 0   | 3   | 0   | 0   | 0   | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|                    | j | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | k | 0                                           | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 196 | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
|                    | l | 1                                           | 0   | 0   | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 2   | 195 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | m | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | n | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | o | 1                                           | 1   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 197 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   | dikenali oleh CNN dengan <i>convex hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|---|---------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   | a                                           | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| p | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r | 0                                           | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 198 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| s | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
| u | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v | 0                                           | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0   | 0   | 0   | 0   | 0   | 0   | 195 | 0   | 0   | 0   | 0   |
| w | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 199 | 0   | 0   | 0   |
| x | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 199 | 0   |
| z | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |

Tabel 4.11. Tabel Hasil Prediksi Skenario 4 Sistem *Convex Hull* + CNN Dengan 80 Epoch

|                    |   | dikenali oleh CNN dengan <i>convex hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |   |   |   |   |   |
|--------------------|---|---------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|---|---|
|                    |   | a                                           | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o | p | q | r | s | t | u | v | w | x | y | z |
| isyarat sebenarnya | a | 200                                         | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | b | 0                                           | 186 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 4 | 1 | 0 | 0 |
|                    | c | 0                                           | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | d | 1                                           | 0   | 0   | 194 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 3   | 0   | 0   | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|                    | e | 0                                           | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | f | 0                                           | 1   | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | g | 0                                           | 0   | 0   | 1   | 0   | 0   | 197 | 0   | 1   | 1   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | h | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | i | 1                                           | 1   | 0   | 0   | 0   | 1   | 0   | 0   | 193 | 0   | 0   | 2   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
|                    | j | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | k | 0                                           | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 197 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
|                    | l | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 3   | 197 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | m | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 199 | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | n | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   | dikenali oleh CNN dengan <i>convex hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |     |
|---|---------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   | a                                           | b | c | d | e | f | g | h | i | j | k | l | m | n | o   | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| o | 1                                           | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 196 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| p | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q | 0                                           | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 199 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r | 0                                           | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 198 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| s | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
| u | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v | 1                                           | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 194 | 0   | 0   | 0   | 0   |
| w | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 2   | 0   | 0   | 0   | 1   | 197 | 0   | 0   | 0   |
| x | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 199 | 0   |
| z | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |

Tabel 4.12. Tabel Hasil Prediksi Skenario 4 Sistem *Convex Hull* + CNN Dengan 100 Epoch

|                    |   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |   |   |   |   |   |
|--------------------|---|---------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|---|---|
|                    |   | a                                           | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o | p | q | r | s | t | u | v | w | x | y | z |
| Isyarat Sebenarnya | a | 200                                         | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | b | 1                                           | 194 | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
|                    | c | 0                                           | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | D | 1                                           | 0   | 0   | 189 | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 5   | 0   | 0   | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 |
|                    | e | 0                                           | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | f | 0                                           | 1   | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | g | 0                                           | 0   | 0   | 1   | 1   | 0   | 192 | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 1 |
|                    | h | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | i | 1                                           | 1   | 0   | 0   | 0   | 1   | 1   | 0   | 191 | 0   | 0   | 1   | 1   | 0   | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 |
|                    | j | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | k | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 197 | 1   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
|                    | l | 1                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 198 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | m | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | n | 0                                           | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   | Dikenali oleh CNN Dengan <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |     |
|---|---------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   | a                                           | b | c | d | e | f | g | h | i | j | k | l | m | n | o   | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| o | 2                                           | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 197 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| p | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q | 1                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 197 | 0   | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r | 0                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 198 | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   |
| s | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
| u | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v | 0                                           | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 198 | 0   | 0   | 0   | 0   |
| w | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 3   | 196 | 0   | 0   | 0   |
| x | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   |
| z | 0                                           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |



Tabel 4.13. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa Convex Hull Dengan 20 Epoch

|                    |   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |   |    |   |   |   |
|--------------------|---|--------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|---|----|---|---|---|
|                    |   | a                                          | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o | p | q | r | s | t | u | v | w  | x | y | z |
| Isyarat Sebenarnya | a | 200                                        | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | b | 0                                          | 185 | 0   | 0   | 0   | 4   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 |
|                    | c | 0                                          | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | D | 0                                          | 1   | 0   | 193 | 0   | 0   | 0   | 0   | 3   | 0   | 0   | 1   | 0   | 0   | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1  | 0 | 0 | 0 |
|                    | e | 0                                          | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | f | 0                                          | 0   | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | g | 0                                          | 0   | 0   | 1   | 0   | 0   | 195 | 0   | 0   | 0   | 0   | 2   | 0   | 0   | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | h | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | i | 4                                          | 0   | 0   | 3   | 0   | 1   | 0   | 0   | 187 | 0   | 0   | 2   | 0   | 0   | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0  | 0 | 0 | 0 |
|                    | j | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | k | 0                                          | 2   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 192 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4  | 0 | 0 | 0 |
|                    | l | 2                                          | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 1   | 0   | 1   | 195 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | m | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 198 | 1   | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | n | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |

|   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |     |
|---|--------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   | a                                          | b | c | d | e | f | g | h | i | j | k | l | m | n | o   | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| o | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| p | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r | 0                                          | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 195 | 0   | 0   | 1   | 1   | 0   | 0   | 0   | 0   |
| s | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
| u | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v | 0                                          | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 190 | 1   | 0   | 0   | 0   |
| w | 0                                          | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 4   | 190 | 0   | 0   | 0   |
| x | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 197 | 0   |
| z | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |

Tabel 4.14. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa Convex Hull Dengan 40 Epoch

|                    |   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |   |   |   |   |
|--------------------|---|--------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|---|
|                    |   | a                                          | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v | w | x | y | z |
| Isyarat Sebenarnya | a | 199                                        | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | b | 0                                          | 183 | 0   | 0   | 0   | 7   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 |
|                    | c | 0                                          | 1   | 199 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | D | 0                                          | 0   | 0   | 195 | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|                    | e | 0                                          | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|                    | f | 1                                          | 0   | 0   | 0   | 0   | 198 | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | g | 3                                          | 0   | 0   | 3   | 0   | 1   | 189 | 0   | 3   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | h | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | i | 4                                          | 1   | 0   | 5   | 0   | 2   | 0   | 0   | 186 | 0   | 0   | 1   | 0   | 0   | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | j | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | k | 1                                          | 3   | 0   | 1   | 0   | 1   | 0   | 0   | 0   | 0   | 184 | 2   | 0   | 0   | 0   | 0 | 0 | 6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
|                    | l | 3                                          | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 1   | 0   | 1   | 194 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | m | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 199 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | n | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | o | 0                                          | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 199 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   |  | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|---|--|--------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   |  | a                                          | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| p |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 199 | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0   | 0   | 197 | 1   | 0   | 0   | 1   | 0   | 0   | 0   | 0   |
| s |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
| u |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v |  | 2                                          | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 188 | 4   | 0   | 0   | 0   |
| w |  | 0                                          | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 2   | 0   | 0   | 0   | 1   | 193 | 0   | 0   | 0   |
| x |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 199 | 0   |
| z |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |

Tabel 4.15. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa Convex Hull Dengan 60 Epoch

|                    |   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |    |   |   |   |
|--------------------|---|--------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|----|---|---|---|
|                    |   | a                                          | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v | w  | x | y | z |
| Isyarat Sebenarnya | a | 200                                        | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | b | 0                                          | 172 | 0   | 0   | 0   | 7   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 |
|                    | c | 0                                          | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | D | 0                                          | 1   | 0   | 192 | 0   | 0   | 1   | 0   | 2   | 0   | 0   | 3   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0  | 0 | 0 | 0 |
|                    | e | 0                                          | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | f | 0                                          | 0   | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | g | 0                                          | 0   | 0   | 0   | 0   | 0   | 189 | 0   | 1   | 0   | 0   | 1   | 0   | 0   | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0  | 7 | 0 | 0 |
|                    | h | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | i | 1                                          | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 191 | 0   | 0   | 2   | 0   | 0   | 3   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1  | 0 | 0 | 0 |
|                    | j | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | k | 0                                          | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 196 | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0  | 0 | 0 | 0 |
|                    | l | 1                                          | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 2   | 195 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | m | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | n | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | o | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |

|   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|---|--------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   | a                                          | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| p | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r | 0                                          | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0   | 0   | 196 | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   |
| s | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
| u | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v | 0                                          | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 198 | 1   | 0   | 0   | 0   |
| w | 0                                          | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 1   | 0   | 0   | 0   | 1   | 194 | 0   | 0   | 0   |
| x | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   |
| z | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |

Tabel 4.16. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa Convex Hull Dengan 80 Epoch

|                    |   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |   |   |   |   |
|--------------------|---|--------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|---|---|---|---|
|                    |   | a                                          | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v | w | x | y | z |
| Isyarat Sebenarnya | a | 200                                        | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | b | 0                                          | 190 | 0   | 0   | 0   | 3   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | c | 0                                          | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | D | 1                                          | 0   | 0   | 192 | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 2   | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
|                    | e | 0                                          | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | f | 0                                          | 2   | 0   | 0   | 0   | 197 | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | g | 1                                          | 0   | 0   | 2   | 0   | 0   | 189 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 4 | 0 | 0 |
|                    | h | 0                                          | 0   | 0   | 1   | 0   | 0   | 0   | 199 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | i | 0                                          | 1   | 0   | 6   | 0   | 0   | 0   | 0   | 190 | 0   | 0   | 2   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
|                    | j | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | k | 0                                          | 3   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 193 | 1   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 |
|                    | l | 1                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 1   | 197 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | m | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | n | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|                    | o | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 0   | 199 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|   |  | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|---|--|--------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|   |  | a                                          | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
| p |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| q |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| r |  | 0                                          | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0   | 0   | 197 | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   |
| s |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| t |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 198 | 2   | 0   | 0   | 0   | 0   | 0   |
| u |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
| v |  | 0                                          | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 198 | 0   | 0   | 0   | 0   |
| w |  | 0                                          | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 193 | 0   | 0   | 0   |
| x |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
| y |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   |
| z |  | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |



Tabel 4.17. Tabel Hasil Prediksi Skenario 4 Sistem CNN Tanpa Convex Hull Dengan 100 Epoch

|                    |   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |     |     |     |     |     |     |     |     |     |     |     |     |     |     |   |   |   |   |   |   |   |    |   |   |   |
|--------------------|---|--------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|---|---|---|---|---|----|---|---|---|
|                    |   | a                                          | b   | c   | d   | e   | f   | g   | h   | i   | j   | k   | l   | m   | n   | o   | p | q | r | s | t | u | v | w  | x | y | z |
| Isyarat Sebenarnya | a | 197                                        | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 3   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | b | 0                                          | 181 | 0   | 0   | 0   | 3   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 0 |
|                    | c | 0                                          | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | D | 4                                          | 0   | 0   | 187 | 0   | 0   | 0   | 0   | 3   | 0   | 0   | 1   | 0   | 0   | 2   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3  | 0 | 0 | 0 |
|                    | e | 0                                          | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | f | 0                                          | 1   | 0   | 0   | 0   | 198 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0  | 0 | 0 | 0 |
|                    | g | 0                                          | 0   | 0   | 0   | 0   | 0   | 196 | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0  | 0 | 0 | 0 |
|                    | h | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | i | 1                                          | 1   | 0   | 1   | 0   | 0   | 0   | 0   | 193 | 0   | 0   | 2   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0  | 0 | 0 | 0 |
|                    | j | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | k | 0                                          | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 195 | 2   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0  | 0 | 0 | 0 |
|                    | l | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 197 | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | m | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | n | 0                                          | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |
|                    | o | 0                                          | 0   | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 198 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0 | 0 | 0 |

|  |   | Dikenali oleh CNN Tanpa <i>Convex Hull</i> |   |   |   |   |   |   |   |   |   |   |   |   |   |   |     |     |     |     |     |     |     |     |     |     |     |
|--|---|--------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|  |   | a                                          | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p   | q   | r   | s   | t   | u   | v   | w   | x   | y   | z   |
|  | p | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|  | q | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 198 | 0   | 2   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|  | r | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 199 | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|  | s | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
|  | t | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   | 0   |
|  | u | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   | 0   | 0   | 0   |
|  | V | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 196 | 1   | 0   | 0   | 0   |
|  | W | 0                                          | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 194 | 0   | 0   | 0   |
|  | X | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   | 0   |
|  | y | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 | 0   |
|  | z | 0                                          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 200 |