



TUGAS AKHIR - TD123456

KONTROL PERGERAKAN KURSI RODA BERBASIS *HEAD GESTURE* MENGGUNAKAN CNN

Batrisyia Zahrani Ananto

NRP 5024 20 1065

Advisor

Dr. Eko Mulyanto Yuniarno,S.T.,M.T.

NIP 19680601199512 1 009

Dion Hayu Fandiantoro, S.T.,M.Eng.

NIP 199420201 1 064

Program Studi Strata 1 (S1) Teknik Komputer

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024



TUGAS AKHIR - TD123456

KONTROL PERGERAKAN KURSI RODA BERBASIS HEAD GESTURE MENGGUNAKAN CNN

Batrisyia Zahrani Ananto

NRP 5024 20 1065

Advisor

Dr. Eko Mulyanto Yuniarno,S.T.,M.T.

NIP 19680601199512 1 009

Dion Hayu Fandiantoro, S.T.,M.Eng.

NIP 199420201 1 064

Program Studi Strata 1 (S1) Teknik Komputer

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - TD123456

***WHEELCHAIR MOVEMENT CONTROL BASED ON
HEAD GESTURE USING CNN***

Batrisyia Zahrani Ananto

NRP 5024 20 1065

Advisor

Dr. Eko Mulyanto Yuniarno, S.T., M.T.

NIP 19680601199512 1 009

Dion Hayu Fandiantoro, S.T., M.Eng.

NIP 199420201 1 064

Undergraduate Study Program of Computer Engineering

Department of Computer Engineering

Faculty of Intelligence Electrical and Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya

2024

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KONTROL PERGERAKAN KURSI RODA BERBASIS *HEAD GESTURE* MENGGUNAKAN CNN

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
memperoleh gelar Sarjana Teknik pada
Program Studi S-1 Teknik Komputer
Departemen Teknik Komputer
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh: **Batrisyia Zahrani Ananto**
NRP. 5024 20 1065

Disetujui oleh Tim Penguji Tugas Akhir:

Dr. Eko Mulyanto Yuniarno, S.T., M.T.
NIP: 19680601199512 1 009

(Pembimbing I)

.....

Dion Hayu Fandiantoro, S.T., M.Eng.
NIP: 1994202011064

(Pembimbing II)

.....

Dr. Diah Puspito Wulandari, S.T., M.Sc..
NIP: 19801219200501 2 001

(Penguji I)

.....

Dr. Arief Kurniawan, S.T., M.T..
NIP: 19740907200212 1 001

(Penguji II)

.....

Arta Kusuma Hernanda, S.T., M.T..
NIP: 1996202311024

(Penguji III)

.....

Mengetahui,
Kepala Departemen Teknik Komputer FTEIC - ITS

Dr. Supeno Mardi Susiki Nugroho,S.T.,M.T..
NIP. 19700313199512 1 001

SURABAYA
Juni, 2024

[Halaman ini sengaja dikosongkan]

APPROVAL SHEET

WHEELCHAIR MOVEMENT CONTROL BASED ON HEAD GESTURE USING CNN

FINAL PROJECT

Submitted to fulfill one of the requirements
for obtaining a degree Bachelor of Engineering at
Undergraduate Study Program of Computer Engineering
Department of Computer Engineering
Faculty of Intelligence Electrical and Informatics Technology
Sepuluh Nopember Institute of Technology

By: **Batrisyia Zahrani Ananto**

NRP. 5024 20 1065

Approved by Final Project Examiner Team:

Dr. Eko Mulyanto Yuniarno, S.T., M.T.
NIP: 19680601199512 1 009

(Advisor I)

.....

Dion Hayu Fandiantoro, S.T., M.Eng.
NIP: 1994202011064

(Co-Advisor II)

.....

Dr. Diah Puspito Wulandari, S.T., M.Sc..
NIP: 19801219200501 2 001

(Examiner I)

.....

Dr. Arief Kurniawan, S.T., M.T..
NIP: 19740907200212 1 001

(Examiner II)

.....

Arta Kusuma Hernanda, S.T., M.T..
NIP: 1996202311024

(Examiner III)

.....

Acknowledged,
Head of Computer Engineering Department ELECTICS - ITS

Dr. Supeno Mardi Susiki Nugroho,S.T.,M.T..
NIP. 19700313199512 1 001

SURABAYA
June, 2024

[Halaman ini sengaja dikosongkan]

PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama Mahasiswa / NRP : Batrisyia Zahrani Ananto / 5024 20 1065
Departemen : Teknik Komputer
Dosen Pembimbing / NIP : Dr. Eko Mulyanto Yuniarno, S.T., M.T. / 19680601199512 1 009

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "KONTROL PERGERAKAN KURSI RODA BERBASIS *HEAD GESTURE* MENGGUNAKAN CNN" adalah hasil karya sendiri, berfsifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, June 2024

Mengetahui
Dosen Pembimbing

Mahasiswa

Dr. Eko Mulyanto Yuniarno, S.T., M.T.
NIP. 19680601199512 1 009

Batrisyia Zahrani Ananto
NRP. 5024 20 1065

[Halaman ini sengaja dikosongkan]

STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Batrisyia Zahrani Ananto / 5024 20 1065
Department : Computer Engineering
Advisor / NIP : Dr. Eko Mulyanto Yuniarno, S.T., M.T. / 19680601199512 1 009

Hereby declared that the Final Project with the title of "*WHEELCHAIR MOVEMENT CONTROL BASED ON HEAD GESTURE USING CNN*" is the result of my own work, is original, and is written by following the rules of scientific writing.

If in future there is a discrepancy with this statement, then I am willing to accept sanctions in accordance with provisions that apply at Sepuluh Nopember Institute of Technology.

Surabaya, June 2024

Acknowledged

Advisor

Student

Dr. Eko Mulyanto Yuniarno, S.T., M.T.
NIP. 19680601199512 1 009

Batrisyia Zahrani Ananto
NRP. 5024 20 1065

[Halaman ini sengaja dikosongkan]

ABSTRAK

Nama Mahasiswa : Batrisyia Zahrani Ananto
Judul Tugas Akhir : KONTROL PERGERAKAN KURSI RODA DENGAN *HEAD GESTURE* MENGGUNAKAN MEDIAPIPE BERBASIS NUC
Pembimbing : 1. Dr. Eko Mulyanto Yuniarso,S.T.,M.T.
2. Dion Hayu Fandiantoro, S.T.,M.Eng.

Kuadriplegia atau tetraplegia merupakan kelumpuhan yang terjadi pada keempat anggota gerak tubuh. Karena limitasi yang mereka miliki, maka mereka memerlukan kursi roda. Mengendalikan pergerakan kursi roda bisa menjadi tantangan, terutama bagi pengguna dengan keterbatasan fisik signifikan. Diperlukan pengembangan sistem kontrol kursi roda yang dapat digunakan penderita tetraplegia. Salah satu pendekatan yang dapat dilakukan untuk mengontrol kursi roda adalah dengan melakukan ekstraksi fitur wajah dan pendekripsi *head gesture* dengan menggunakan mediapipe. Kemudian data *head gesture* yang telah diklasifikasikan akan dikirimkan oleh NUC ke sistem kontrol kursi roda. Sistem kontrol tersebutlah yang akan mengatur arah gerak kursi roda. Dengan menggunakan metodologi yang digunakan, dapat ditarik beberapa kesimpulan dari pengujian yang telah dilakukan. Model yang akan digunakan memiliki arsitektur CNN 7 layer dengan Convolutional 2D 64, 256 dan diakhiri dengan Dense 512. Jarak model yang paling tinggi akurasinya adalah 50 sentimeter. Intensitas cahaya yang paling tinggi akurasinya adalah 110 lux. Kecepatan FPS laptop penulis lebih tinggi daripada NUC yang digunakan. Rata-rata waktu *delay* respons motor adalah 0,3025423729 detik dan *inference time*nya adalah 0,07220 detik. Rata-rata kestabilan gerak motor kursi roda untuk pendekripsi selama 2 detik adalah 8,9364 detik.

Kata Kunci: Gestur Kepala, Kursi Roda, Kontrol , NUC, Tetraplegia.

[Halaman ini sengaja dikosongkan]

ABSTRACT

*Name : Batrisyia Zahrani Ananto
Title : WHEELCHAIR MOVEMENT CONTROL WITH HEAD GESTURE USING
MEDIAPIPE BASED ON NUC
Advisors : 1. Dr. Eko Mulyanto Yuniarno,S.T.,M.T.
2. Dion Hayu Fandiantoro, S.T.,M.Eng.*

Quadriplegia or tetraplegia is paralysis that occurs in all four limbs. Because of their limitations, they need a wheelchair. Controlling the movement of a wheelchair can be a challenge, especially for users with significant physical limitations. It is necessary to develop a wheelchair control system that can be used by tetraplegic sufferers. One approach that can be taken to control a wheelchair is to extract facial features and detect head gesture using mediapipe. Then the classified head gesture data will be sent by the NUC to the wheelchair control system. This control system will regulate the direction of movement of the wheelchair. By using the methodology used, several conclusions can be drawn from the tests that have been carried out. The model that will be used has a 7 layer CNN architecture with Convolutional 2D 64, 256 and ending with Dense 512. The model distance with the highest accuracy is 50 centimeters. The light intensity with the highest accuracy is 110 lux. The FPS speed of the writer's laptop is higher than the NUC used. The average motor response time delay is 0.3025423729 seconds and the inference time is 0.07220 seconds. The average stability of the wheelchair motor for detection for 2 seconds is 8,9364 seconds.

Keywords: Control, Head Gesture, NUC, Tetraplegia, Wheelchair.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji dan syukur kehadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penulisan laporan tugas akhir dengan judul "**Kontrol Pergerakan Kursi Roda Dengan Head Gesture Menggunakan Mediapipe Berbasis NUC**"

Penelitian ini disusun dalam rangka memenuhi salah satu syarat untuk menyelesaikan pendidikan S-1 di Program Studi Teknik Komputer Departemen Teknik Komputer, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dorongan spiritual dan materil dalam penyelesaian penelitian ini.
2. Bapak Dr. Eko Mulyanto Yuniarno, S.T., M.T. dan Bapak Dion Hayu Fandiantoro, S.T., M.Eng. selaku dosen pembimbing yang telah memberikan bimbingan, arahan, serta motivasi kepada penulis dalam menyelesaikan penelitian ini.
3. Bapak-ibu dosen pengajar Departemen Teknik Komputer yang telah memberikan ilmu dan bimbingan kepada penulis selama menempuh perkuliahan.
4. Teman-teman seperjuangan yang telah memberikan dukungan, semangat, dan bantuan dalam menyelesaikan penelitian ini.

Kesempurnaan hanya milik Tuhan Yang Maha Esa, untuk itu penulis memohon segenap kritik dan saran yang membangun. Akhir kata, semoga penelitian ini dapat memberikan manfaat bagi pembaca dan masyarakat luas.

Surabaya, Juni 2024

Batrisyia Zahrani Ananto

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	1
1.3 Tujuan	1
1.4 Batasan Masalah	2
1.5 Manfaat	2
1.6 Sistematika Penulisan	2
2 TINJAUAN PUSTAKA	5
2.1 Hasil penelitian/perancangan terdahulu	5
2.1.1 <i>Wireless Head Gesture Controlled Wheel Chair for Disable Persons</i> . .	5
2.1.2 Kursi Roda Elektrik dengan Kendali Gestur Kepala	5
2.2 Dasar Teori	5
2.2.1 Kursi Roda Elektrik	5
2.2.2 DC Motor	6
2.2.3 Head Gesture	6
2.2.4 Mediapipe	7
2.2.5 <i>Convolutional Neural Network (CNN)</i>	8
2.2.6 <i>Metrics Evaluasi</i>	9
2.2.7 Tensorflow	11
2.2.8 Python	11

2.2.9	Kamera	12
2.2.10	DC-DC Voltage Regulator	13
2.2.11	H-Bridge Motor Driver	13
2.2.12	ESP32 Devkit V1	13
2.2.13	NUC	14
2.2.14	Arduino IDE	14
3	METODOLOGI	17
3.1	<i>Hardware</i>	17
3.1.1	NUC	18
3.1.2	Kontroller Kursi Roda	18
3.2	<i>Software</i>	21
3.2.1	Citra	21
3.2.2	Ekstraksi Fitur Landmark	22
3.2.3	Klasifikasi Landmark	22
4	PENGUJIAN DAN ANALISIS	29
4.1	Skenario Pengujian	29
4.2	Pengujian Performa Model	29
4.2.1	Pengujian Model Pertama	30
4.2.2	Pengujian Model Kedua	32
4.3	Pengujian Performa Model dengan Variasi Jarak Kamera	35
4.3.1	Pengujian jarak 50 sentimeter	36
4.3.2	Pengujian jarak 75 sentimeter	40
4.3.3	Pengujian jarak 100 sentimeter	43
4.4	Pengujian performa model dengan variasi pencahayaan	46
4.4.1	Pengujian intensitas cahaya 15 lux	47
4.4.2	Pengujian intensitas cahaya 46 lux	48
4.4.3	Pengujian intensitas cahaya 110 lux	49
4.5	Pengujian Kecepatan FPS	50
4.5.1	Pengujian Kecepatan FPS Pada Laptop	50
4.5.2	Pengujian Kecepatan FPS Pada NUC	51
4.6	Pengujian Waktu Respons Kontrol Kursi Roda	52
4.7	Pengujian kestabilan Performa Motor Kursi Roda	55
4.7.1	Kestabilan Motor Kelas Kanan	55

4.7.2	Kestabilan Motor Kelas Kiri	56
4.7.3	Kestabilan Motor Kelas Maju	58
4.7.4	Kestabilan Motor Kelas Mundur	60
4.8	Pembahasan Hasil	61
5	PENUTUP	63
5.1	Kesimpulan	63
5.2	Saran	63
DAFTAR PUSTAKA		65
BIOGRAFI PENULIS		67

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

2.1	Kursi Roda Elektrik	6
2.2	Koordinat <i>Landmark</i>	8
2.3	Arsitektur pada <i>Convolutional Neural Network</i>	9
2.4	<i>Aktivasi ReLU</i>	9
2.5	<i>Max Pooling</i>	10
2.6	Proses <i>Flattening</i>	10
2.7	Visualisasi <i>confusion matrix</i>	10
2.8	H-Bridge Motor Driver	13
2.9	ESP32	14
2.10	NUC	14
3.1	Blok diagram <i>hardware</i>	17
3.2	Flowchart Mengirim Data String Melalui WiFi [22]	18
3.3	Flowchart Menerima Data String Melalui Access Point WiFi Pada ESP32 [22] .	19
3.4	Skematik [22]	20
3.5	Flowchart Kontrol Motor Kursi Roda Melalui Access Point WiFi [22]	21
3.6	Blok diagram <i>software</i>	21
3.7	Contoh citra wajah	23
3.8	Arsitektur CNN Model 1	24
3.9	Arsitektur CNN Model 2	25
3.10	Diagram Arsitektur CNN Model 1	26
3.11	Diagram Arsitektur CNN Model 2	26
4.1	Diagram Pembagian Dataset	30
4.2	Accuracy Hasil Pelatihan Model	30
4.3	Model Loss Hasil Pelatihan Model	31
4.4	Confusion Matrix Hasil Pelatihan Model Pertama	31
4.5	Accuracy Hasil Pelatihan Model Kedua	33
4.6	Model Loss Hasil Pelatihan Model Kedua	33
4.7	Confusion Matrix Hasil Pelatihan Model Kedua	34
4.8	Dataset Jarak 50 sentimeter	37

4.9	Accuracy Hasil Pelatihan Jarak 50cm	38
4.10	Loss Hasil Pelatihan Jarak 50cm	38
4.11	Confusion Matrix Hasil Pelatihan Jarak 50cm	39
4.12	Dataset Jarak 75 sentimeter	40
4.13	Accuracy Hasil Pelatihan Jarak 75cm	41
4.14	Loss Hasil Pelatihan Jarak 75cm	41
4.15	Confusion Matrix Hasil Pelatihan Jarak 75cm	42
4.16	Dataset Jarak 100 sentimeter	43
4.17	Accuracy Hasil Pelatihan Jarak 100cm	44
4.18	Loss Hasil Pelatihan Jarak 100cm	44
4.19	Confusion Matrix Hasil Pelatihan Jarak 100cm	45
4.20	Pendeteksian pada 15 lux	47
4.21	Pendeteksian pada 46 lux	48
4.22	Pendeteksian pada 110 lux	49

DAFTAR TABEL

2.1	Spesifikasi <i>Webcam</i>	12
3.1	Kelas yang Digunakan	27
4.1	Hasil Klasifikasi Model Pertama	32
4.2	Hasil Validasi Nilai Model Pertama	32
4.3	Hasil Pengujian Keterdeteksian Model Pertama	32
4.4	Hasil Klasifikasi Model Kedua	34
4.5	Hasil Validasi Nilai Model Kedua	35
4.6	Hasil Pengujian Keterdeteksian Model Kedua	35
4.7	Variasi Jarak Kamera	36
4.8	Hasil Klasifikasi Model Jarak 50cm	39
4.9	Hasil Validasi Nilai Model Jarak 50cm	40
4.10	Hasil Klasifikasi Model Jarak 75cm	42
4.11	Hasil Validasi Nilai Model Jarak 75cm	42
4.12	Hasil Klasifikasi Model Jarak 100cm	45
4.13	Hasil Validasi Nilai Model Jarak 100cm	45
4.14	Variasi Intensitas Cahaya	46
4.15	Hasil Variasi 15 Lux	47
4.16	Hasil Variasi 46 Lux	48
4.17	Hasil Variasi 110 Lux	49
4.18	Spesifikasi Laptop	50
4.19	Hasil Pengujian FPS Laptop	51
4.20	Spesifikasi NUC	52
4.21	Hasil Pengujian FPS NUC	52
4.22	Hasil Pengujian Respons Time	53
4.23	Hasil Pengujian Kestabilan Motor Kelas Kanan	55
4.24	Hasil Pengujian Kestabilan Motor Kelas Kiri	57
4.25	Hasil Pengujian Kestabilan Motor Kelas Maju	58
4.26	Hasil Pengujian Kestabilan Motor Kelas Mundur	60

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kuadriplegia atau tetraplegia merupakan kelumpuhan yang terjadi pada keempat anggota gerak tubuh [1]. Data dari University of Alabama, penderita tetraplegia ini mencapai 40,8 %. Karena limitasi yang mereka miliki, maka mereka memerlukan kursi roda. Kursi roda adalah alat yang digunakan untuk meningkatkan kemampuan mobilitas bagi orang yang memiliki kekurangan, seperti orang yang cacat fisik (khususnya penyandang cacat kaki), pasien rumah sakit yang tidak diperbolehkan untuk melakukan banyak aktivitas fisik, orang tua, lanjut usia, dan orang-orang yang memiliki resiko tinggi untuk terluka bila berjalan sendiri [2]. Penggunaan kursi roda konvensional cenderung berfokus pada penggunaan manual yang masih menguras pengguna dapat menggunakan tangan mereka untuk menggerakan kursi roda secara maksimal [3].

Mengendalikan pergerakan kursi roda bisa menjadi tantangan, terutama bagi pengguna dengan keterbatasan fisik signifikan. Diperlukan pengembangan sistem kontrol kursi roda yang dapat digunakan penderita tetraplegia. Salah satu pendekatan yang dapat dilakukan untuk mengontrol kursi roda adalah dengan melakukan ekstraksi fitur wajah dan mendekripsi head gesture dengan menggunakan mediapipe. Mediapipe adalah sebuah framework yang dirancang dengan cara mengimplementasikan kecerdasan buatan kedalam aplikasi yang akan dibangun[4]. Mediapipe merupakan deep learning sehingga memerlukan kekuatan pemrosesan.

Sebagai *mini computer* yang memadai dan terjangkau, pada penelitian ini digunakanlah NUC sebagai platform penunjang dan pengganti laptop. Dengan begitu, penggabungan teknologi AI dan gestur kepala untuk menciptakan antarmuka kontrol kursi roda yang lebih mudah untuk penderita tetraplegia dapat dilakukan secara real time.

1.2 Permasalahan

Dari latar belakang diatas, terdapat beberapa rumusan masalah yang perlu diidentifikasi dan diselesaikan. Rumusan masalah adalah sebagai berikut:

1. Bagaimana antarmuka kontrol yang dapat digunakan oleh penderita tetraplegia untuk mengendalikan pergerakan kursi roda?
2. Bagaimana performa sistem dalam mendekripsi head gesture untuk menggerakan kursi roda?
3. Bagaimana merancang sistem yang dapat mengenali dan memahami berbagai gestur kepala dengan akurat, termasuk gestur untuk mengubah arah, kecepatan, dan berhenti kursi roda?

1.3 Tujuan

Dari masalah yang telah dirumuskan diatas, dapat dibuat beberapa tujuan sebagai berikut:

1. Membuat antarmuka kontrol yang dapat digunakan oleh penderita tetraplegia untuk mengendalikan pergerakan kursi roda
2. Menguji keandalan, akurasi, dan responsivitas sistem kontrol kursi roda yang menggunakan gestur kepala
3. Merancang sistem yang dapat mengenali dan memahami berbagai gestur kepala dengan akurat, termasuk gestur untuk mengubah arah, kecepatan, dan berhenti kursi roda

1.4 Batasan Masalah

Batasan masalah yang penulis tetapkan pada penelitian ini dapat dibagi menjadi dua, yaitu batasan masalah pada software dan hardware. Untuk batasan masalah pada software, fokus utama dari penelitian ini adalah penerapan teknologi Mediapipe dengan training menggunakan CNN dan yang di deketsi adalah gestur kepala. Hal ini berarti penelitian tidak akan menggali teknologi lain seperti deteksi gerakan tangan atau perintah suara. Selanjutnya, untuk memastikan akurasi dalam deteksi gerakan kepala, penelitian ini akan dilakukan dalam lingkungan indoor dengan pencahayaan yang stabil dan minim gangguan visual lainnya. Lingkungan outdoor dengan variabel pencahayaan yang berfluktuasi tidak akan menjadi ruang lingkup penelitian ini.

Untuk batasan masalah hardware, fokus utama dari penelitian ini adalah penerapan pada NUC dan kursi roda. Dalam konteks kursi roda, asumsi dasar yang diambil adalah penggunaan kursi roda listrik standar, dengan mengabaikan variabel seperti berat pengguna atau kecepatan maksimal kursi. Selain itu, aspek teknis seperti daya tahan baterai NUC atau konsumsi daya oleh Mediapipe tidak akan menjadi fokus penelitian. Pembatasan-pembatasan tersebut didefinisikan untuk memastikan penelitian berjalan efisien dan menghasilkan kesimpulan yang spesifik dan relevan.

1.5 Manfaat

Manfaat yang bisa didapat dari penelitian ini adalah sebuah alat dalam bentuk kursi roda yang dapat digerakkan menggunakan head gesture. Kursi roda ini diharapkan dapat membantu penyandang cacat seluruh anggota gerak untuk dapat bergerak lebih bebas dan lebih mandiri.

1.6 Sistematika Penulisan

Laporan penelitian ini disusun dengan sistematika yang sedemikian rupa agar mudah dipahami oleh pembaca yang awam atau peneliti lain yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan ini adalah sebagai berikut:

1. BAB I Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang dari penelitian ini yang kemudian menimbulkan permasalahan yang akan diselesaikan dan menjadi tujuan dari penelitian ini. Terdapat juga batasan masalah dari permasalahan yang didapat dan manfaat dari dilakukannya penelitian ini.

2. BAB II Tinjauan Pustaka

Bab ini berisi mengenai penelitian terdahulu yang mirip dengan penelitian ini. Selain itu, dipaparkan juga mengenai teori-teori, peralatan dan bahan-bahan yang akan digunakan pada penelitian kali ini.

3. BAB III Desain dan Implementasi Sistem

Bab ini berisi tentang perancangan dan pengimplementasian *hardware* dan *software* yang diperlukan. Dijelaskan juga secara rinci hubungan antara komponen yang digunakan.

4. BAB IV Pengujian dan Analisa

Bab ini berisi skenario-skenario pengujian yang dilakukan beserta data-data yang didapatkan. Data-data tersebut juga akan diolah dan divisualisasikan untuk memudahkan pengambilan kesimpulan.

5. BAB V Penutup

Pada bab ini terdapat kesimpulan yang berisi mengenai hasil akhir yang didapat setelah dilakukannya penelitian ini. Terdapat juga saran untuk penelitian selanjutnya bagi yang ingin melanjutkan penelitian ini.

[Halaman ini sengaja dikosongkan]

BAB II

TINJAUAN PUSTAKA

2.1 Hasil penelitian/perancangan terdahulu

Pada bagian ini akan dibahas hasil penelitian atau perancangan terdahulu yang relevan dengan penelitian yang akan dilakukan. Penelitian atau perancangan terdahulu tersebut akan dibahas berdasarkan metode yang digunakan, hasil yang diperoleh, dan kesimpulan yang diam-bil.

2.1.1 Wireless Head Gesture Controlled Wheel Chair for Disable Persons

Penelitian yang dilakukan oleh Shayban dan Abdul ini membuat kontrol pergerakan kursi roda dengan menggunakan *accelerometer* yang ditempel diatas topi. *Accelerometer* tersebut dapat mengetahui gestur kepala dari nilai *Axis-X* dan *Axis-Y*. Karena penelitian ini masih mengharuskan penggunanya menggunakan topi yang membuat kontrol pergerakan kursi roda menjadi kurang praktis dan kurang nyaman untuk beberapa pengguna karena ukuran topi yang meskipun bisa di sesuaikan tapi tetap terlalu besar atau terlalu kecil. Jika topinya terlalu besar, maka akan mengganggu akurasi pendektsian dan jika terlalu kecil, maka pengguna tidak dapat memakainya.

2.1.2 Kursi Roda Elektrik dengan Kendali Gestur Kepala

Penelitian yang dilakukan oleh Tiar, Iyus, dan Trie ini membuat kontrol pergerakan kursi roda dengan menggunakan *accelerometer* dan *ultrasonic*. *Accelerometer* digunakan untuk mengendalikan kursi roda dengan gestur kepala dan *ultrasonic* digunakan untuk melakukan pendeteksian jarak. *Accelerometer* yang digunakan ditempel di kepala pengguna agar dapat melakukan pendektsian gestur kepala. Penelitian ini mengharuskan penggunanya mengenakan *accelerometer* sehingga dapat membuat penggunanya kurang nyaman.

2.2 Dasar Teori

Pada bagian ini akan dibahas dasar teori yang digunakan dalam penelitian ini. Dasar teori tersebut meliputi teori-teori yang mendukung penelitian ini, seperti teori tentang kursi roda elektrik, DC motor, head gesture, mediapipe, *convolutional neural network* (CNN), *metrics* evaluasi, TensorFlow, Python, kamera, DC-DC voltage regulator, H-bridge motor driver, ESP32 Devkit V1, dan NUC.

2.2.1 Kursi Roda Elektrik

Kursi roda adalah alat yang digunakan untuk meningkatkan kemampuan mobilitas bagi orang yang memiliki kekurangan, seperti orang yang cacat fisik (khususnya penyandang cacat kaki), pasien rumah sakit yang tidak diperbolehkan untuk melakukan banyak aktivitas fisik, orang tua, lanjut usia, dan orang-orang yang memiliki resiko tinggi untuk terluka bila berjalan sendiri [2]. Penggunaan kursi roda konvensional cenderung berfokus pada penggunaan manual yang masih mengasumsikan pengguna dapat menggunakan tangan mereka untuk menggerakan

kursi roda secara maksimal [3]

Secara umum, kursi roda yang digunakan oleh masyarakat dibagi menjadi dua macam, yaitu kursi roda elektrik dan kursi roda manual. Perbedaan dari kedua kursi roda ini terletak pada tenaga penggeraknya. Kursi roda elektrik bergerak dengan tenaga listrik dari baterai dan memiliki beberapa metode untuk mengendalikannya [5].

Kursi roda elektrik yang akan digunakan pada penelitian kali ini adalah kursi roda elektrik Onehealth *electric wheelchair*KY-123 yang telah diregistrasikan di Kementerian Kesehatan Republik Indonesia dan memiliki surat izin edar secara resmi.



Gambar 2.1: Kursi Roda Elektrik

2.2.2 DC Motor

Pada kursi roda elektrik, terdapat dua buah DC motor untuk menggerakkan rodanya. Satu DC motor dipasang di roda kanan dan satunya lagi dipasang di roda kiri.

DC motor adalah sebuah mesin yang mengubah DC daya menjadi tenaga mekanik dikenal dengan istilah DC motor. Pengoperasiannya didasarkan pada prinsip bahwa ketika ada arus yang mengalir konduktor ditempatkan dalam medan magnet, konduktor mengalami mekanik memaksa. Arah gaya ini diberikan oleh aturan tangan kiri Fleming dan besarnya diberikan oleh:

$$F = BI\ell \text{ newton} \quad (2.1)$$

Pada dasarnya, tidak ada perbedaan konstruksi antara DC motor dan DC generator. DC yang sama. mesin dapat dijalankan sebagai generator atau motor[6]. Pada kursi roda elektrik KY-123, digunakan Motor DC MY1016Z.

2.2.3 Head Gesture

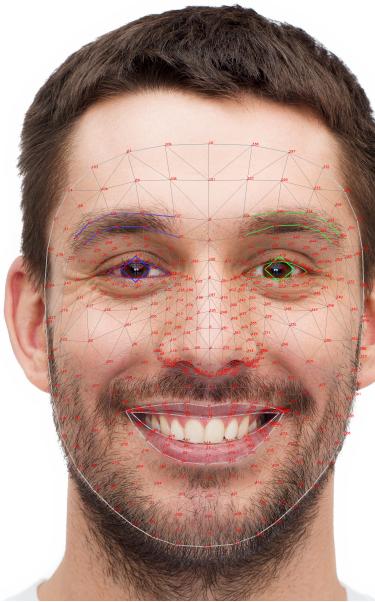
Gestur kepala atau *Head gesture* berarti posisi yang dilakukan oleh pergerakan kepala dengan mempertimbangkan seluruh geometri wajahnya seperti mata, hidung, bibir, dan lain-lain. Kegunaan dari isyarat tersebut adalah untuk mengekspresikan pikiran, emosi, dan lain-lain. Gestur kepala seperti itu sangat berguna dan bermanfaat bagi penyandang cacat atau orang

yang mengalami kelumpuhan dari leher dan seterusnya [7].

2.2.4 Mediapipe

Mediapipe merupakan sebuah framework yang dirancang dengan cara mengimplementasikan kecerdasan buatan kedalam aplikasi yang akan dibangun. Mediapipe tersebut digunakan oleh Google dalam menentukan karakteristik yang ada dalam video. Hal yang dilakukan oleh Google dengan mengumpulkan semua dataset kemudian memberikan label serta mengidentifikasi setiap landmark-nya secara manual. Kecerdasan buatan pada pengaplikasiannya secara garis besar terbagi tujuh cabang, yaitu machine learning, natural language processing, expert system, vision, speech, planning dan robotics. Cabang dari kecerdasan buatan tersebut dimaksudkan untuk ruang lingkup saat pengembangan atau belajar artificial intelligence karena pada dasarnya kecerdasan buatan memiliki ruang lingkup yang sangat luas dan beragam[8].

Pada mediapipe terdapat tiga pendektsian yang kemudian dapat dibuat lebih spesifik lagi, yaitu head gesture, hand gesture, dan pose detection. Head gesture atau pendektsian gestur kepala dapat dibuat lebih spesifik menjadi pendektsian pupil mata dan mulut. Hand gesture atau pendektsian gesture tangan dapat dibuat lebih spesifik menjadi pendektsian jari-jemari. Pose detection adalah pendektsian pose seluruh badan yang dapat dibuat menjadi lebih spesifik seperti pendektsian gerak kaki atau tangan. Hal yang menarik penulis untuk melakukan penelitian terhadap kecerdasan buatan ini adalah bagian head gesture recognition. Head gesture recognition ini berupa face landmarker yang menggunakan serangkaian model untuk memprediksi landmark wajah. Model pertama mendekksi wajah, model kedua menemukan penanda pada wajah yang terdeteksi, dan model ketiga menggunakan penanda tersebut untuk mengidentifikasi fitur dan ekspresi wajah. Model berikut dikemas bersama ke dalam bundel model yang dapat diunduh. Pertama, model deteksi wajah yang mendekksi keberadaan wajah dengan beberapa penanda wajah utama. Kedua, model face mesh yang menambahkan pemetaan wajah secara lengkap. Model tersebut menghasilkan perkiraan 468 landmark wajah 3 dimensi. Ketiga, model prediksi blendshape yang akan menerima keluaran dari model face mesh yang memprediksi 52 skor blendshape, yang merupakan koefisien yang mewakili ekspresi wajah yang berbeda[8].



Gambar 2.2: Koordinat *Landmark*

2.2.4.1 Citra Wajah

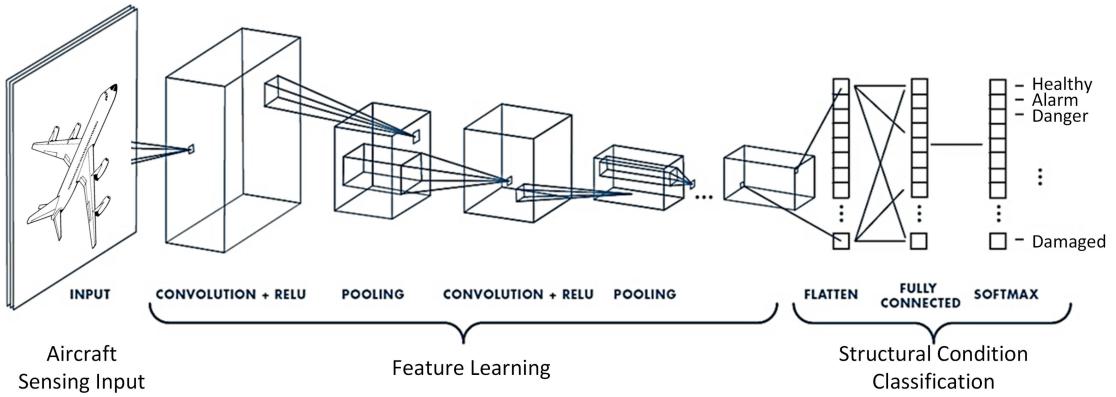
Pada penelitian ini, citra wajah adalah bahan utama untuk melakukan proses eksplorasi dan ekstraksi data. Citra wajah yang digunakan pada penelitian ini diklasifikasikan menjadi lima kondisi, yaitu hadap depan, hadap kanan, hadap kiri, hadap atas, dan hadap bawah. Citra wajah pada penelitian ini ditangkap oleh kamera yang kemudian akan digambarkan garis-garis landmark dan diubah backgroundnya selain landmark-landmark tersebut menjadi hitam.

2.2.5 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) telah mencapai hasil yang revolusioner selama dekade terakhir dalam berbagai bidang yang berkaitan dengan pengenalan pola, mulai dari pemrosesan gambar hingga pengenalan suara. Aspek paling bermanfaat dari CNN adalah mengurangi jumlah parameter dalam *Artificial Neural Network* (ANN). Pencapaian ini telah mendorong para peneliti dan pengembang untuk mengembangkan model yang lebih besar guna menyelesaikan tugas-tugas kompleks, yang tidak mungkin dilakukan dengan ANN klasik. Asumsi penting tentang masalah yang dipecahkan oleh CNN adalah fitur-fitur yang tidak bergantung pada lokasi secara spasial. Sebagai contoh, dalam aplikasi pendekripsi wajah, kita tidak perlu memperhatikan di mana wajah berada dalam gambar. Yang penting adalah mendekripsi mereka tanpa memandang posisi mereka dalam gambar yang diberikan. Aspek penting lain dari CNN adalah untuk memperoleh fitur-fitur abstrak ketika input menyebar menuju lapisan-lapisan yang lebih dalam.[9].

Arsitektur pada CNN terdiri dari tiga bagian, yaitu input, *feature learning*, dan *classification*. *Feature Learning* terdiri dari dua buah *convolution layer* dan dua buah *pooling layer*. Pada *classification* terdiri dari dua *hidden layer* dan satu *output layer*. Arsitektur CNN dapat digambarkan seperti pada Gambar 2.1.

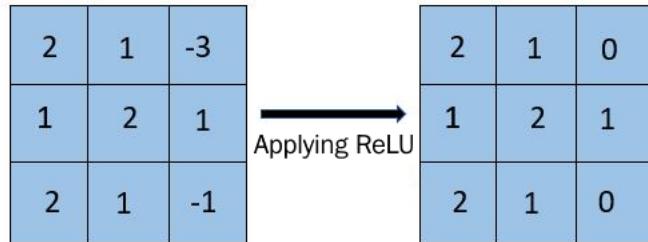
Input CNN merupakan array tiga dimensi dengan ukuran seperti pada Persamaan 2.2. Apabila input merupakan suatu citra maka citra tersebut harus diubah menjadi array dua dimensi.



Gambar 2.3: Arsitektur pada *Convolutional Neural Network*

$$\text{Baris} * \text{Kolom} * \text{Depth} \quad (2.2)$$

Convolution Layer digunakan untuk menyaring (*filter*) matriks dari citra *input*. *Zero Padding* akan diperlukan untuk mempertahankan ukuran matriks dari citra [10]. Ukuran kernel yang digunakan pada layer konvolusi adalah 3×3 dan 5×5 . *Output* dari lapisan konvolusi ini akan digunakan sebagai *input* pada *Pooling Layer* [11]. Apabila output dari *Convolution Layer* bernilai negatif maka akan dilakukan perhitungan tambahan berupa aktifasi ReLU. Fungsi aktivasi ReLU akan mengubah nilai matriks yang bernilai negatif menjadi nol. Contoh penerapan dari aktivasi ReLU dapat dilihat pada Gambar 2.4.



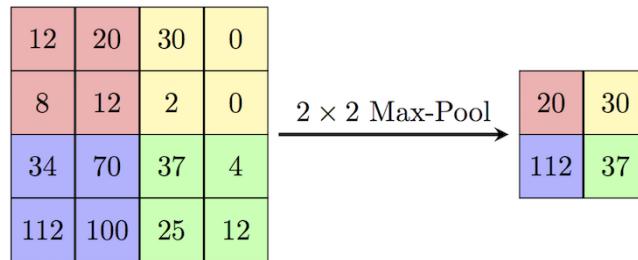
Gambar 2.4: Aktivasi *ReLU*

Pooling Layer digunakan untuk mengurangi jumlah parameter ketika ukuran citra terlalu besar dengan cara mengurangi dimensi setiap fitur. Karena ukuran citra menjadi lebih kecil maka proses *feature map* akan menjadi lebih cepat [11]. *Max Pooling* dilakukan dengan cara mengambil nilai dengan elemen terbesar sesuai dengan ukuran filter. Sebagai contoh pada Gambar 2.5 merupakan *max pooling* dengan filter 2×2 dengan *stride* sebesar 2.

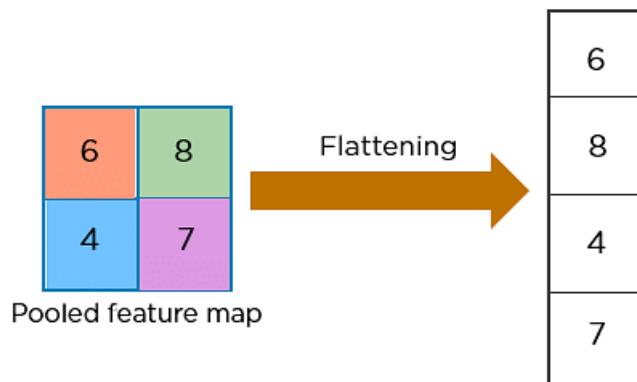
Flatten merupakan suatu proses dimana hasil dari *Feature Learning* diubah menjadi vektor yang selanjutnya akan menjadi input pada proses klasifikasi dengan arsitektur *fully connected layer*. Flatten digunakan untuk mengubah matriks menjadi vektor dengan menyesuaikan sesuai format *input* pada *neural network layer*. Flatten dapat digambarkan seperti pada Gambar 2.10.

2.2.6 Metrics Evaluasi

Dalam melakukan proses pengklasifikasian, dilakukan evaluasi performa model dengan perhitungan efektifitas model yang telah dibuat. *Confusion Matrix* adalah salah satu perhitungan



Gambar 2.5: *Max Pooling*



Gambar 2.6: Proses *Flattening*

yang umum digunakan.

Dalam deep learning, evaluasi performa yang sering digunakan adalah *Metrics* tersebut adalah *accuracy*, *sensitivity (recall)*, *precision* dan *F1-score*. TP, TN, FP, FN adalah parameter yang digunakan dalam pengevaluasian *Metrics* tersebut adalah *accuracy*, *sensitivity (recall)*, *precision* dan *F1-score*. TP atau *True Positive* adalah jumlah gambar wajah yang teridentifikasi. *True Negative* atau TN adalah jumlah gambar bukan wajah yang terdeteksi. *False Positive* atau FP adalah jumlah salahnya pendekripsi diidentifikasi sebagai wajah yang sebenarnya bukan wajah. *False Negative* atau FN adalah salahnya pendekripsi diidentifikasi sebagai bukan wajah yang sebenarnya adalah wajah [**shajihan**]. Visualisasi *confusion matrix* dapat dilihat pada Gambar 2.7.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar 2.7: Visualisasi *confusion matrix*

2.2.6.1 Accuracy

Accuracy didefinisikan sebagai rasio benarnya klasifikasi data dengan total jumlah data yang ada[12]. Nilai *accuracy* dapat diperoleh menggunakan Persamaan 2.3.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.3)$$

2.2.6.2 Precision

Precision (PC) atau nilai prediksi positif, mengukur bagian dari data positif yang diklasifikasikan secara benar[12]. Nilai *precision* dapat diperoleh dengan Persamaan 2.4.

$$Precision = \frac{TP}{TP + FP} \quad (2.4)$$

2.2.6.3 Sensitivity

Sensitivity (SN) yang juga dikenal sebagai tingkat *true positive* atau *recall*, mengukur bagian dari klasifikasi data positif yang benar[12]. Dengan demikian, nilai recall dapat dihitung menggunakan Persamaan 2.5.

$$Recall = \frac{TP}{TP + FN} \quad (2.5)$$

2.2.6.4 F-Score

Metric *F-Score* yang umum adalah dengan menggabungkan perhitungan *precision* dan *recall*. Secara khusus, *F-Score* yang seimbang didapatkan ketika $\beta = 1$, oleh karena itu lebih dikenal sebagai perhitungan *F1* [12]. Nilai *F1-Score* dapat dihitung menggunakan Persamaan 2.6.

$$F1-Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2.6)$$

2.2.7 Tensorflow

TensorFlow merupakan platform untuk machine learning. TensorFlow memiliki fitur untuk menjalankan training model menggunakan Central Processing Unit (CPU) dan training model Graphic Processing Unit (GPU) yang memiliki waktu training yang lebih cepat dibanding training model CPU [13].

TensorFlow awalnya dikembangkan oleh para peneliti dan pengembang yang bekerja sebagai tim di Google Brain untuk melakukan penelitian di bidang machine learning dan neural network. Framework ini dirilis pertama kali pada bulan Februari 2017 dan terus dikembangkan hingga sekarang [14].

2.2.8 Python

Python adalah bahasa pemrograman interpretatif berorientasi objek. Python menawarkan struktur data tingkat tinggi termasuk kamus, modul, kelas, pengecualian, daftar dan array asosiatif, pengetikan dinamis, pengikatan dinamis, dan manajemen memori pintar. Meskipun memi-

liki sintaks yang sangat mudah dimengerti dan bersih, Python adalah bahasa pemrograman yang kuat dan serbaguna. Python diciptakan pada tahun 1990 oleh Guido van Rossum. Python dapat digunakan dengan bebas, bahkan untuk tujuan komersial, seperti banyak bahasa pemrograman lainnya, dan dapat dieksekusi di hampir semua komputer modern. Interpreter secara otomatis mengubah program Python menjadi kode byte yang tidak tergantung pada platform tempat kode tersebut dijalankan [15].

Python merupakan bahasa pemrograman yang sangat fleksibel, serbaguna, dan dapat digunakan untuk berbagai keperluan seperti skrip, pengembangan aplikasi web, dan lain-lain. Python juga bisa diperkaya dengan bahasa C dan C++, memberikan keunggulan kecepatan untuk tugas-tugas yang memerlukan banyak pemrosesan. Python memiliki struktur yang terorganisir dengan baik, termasuk blok kode yang terstruktur, fungsi, kelas, modul, dan paket, serta konsistensi dalam penggunaan objek dan paradigma pemrograman berorientasi objek. Ini memudahkan dalam mengembangkan aplikasi yang efisien dan mudah dipahami, baik untuk tugas kecil maupun besar. Python dilengkapi dengan berbagai tipe data bawaan yang kompleks seperti string, list, dan dictionary, serta kontrol alur standar seperti if, if-else, while, dan for. Bahasa ini secara otomatis mengompilasi kode sumber menjadi byte code, mendukung ekstensi dalam C dan C++, dan menggunakan indentasi untuk menandai blok kode. Indentasi yang standar adalah empat spasi per tingkat tanpa menggunakan tab, yang memudahkan integrasi kode Python dengan kode dari bahasa lain.

2.2.9 Kamera

Kata “kamera” adalah kependekan dari ”kamera obscura”, yang merupakan tenda kedap cahaya, memungkinkan proyeksi gambar melalui lubang kecil dengan lensa. Lensa digunakan untuk mengumpulkan dan menyatukan cahaya ke suatu titik, seperti yang terjadi pada cermin pembesar. Seniman yang duduk di dalam ruangan biasa menjiplak gambar yang dibuat pada permukaan datar. Permukaan datar ini kemudian digantikan oleh film fotografi dengan bahan kimia peka cahaya, yang memungkinkan terjadinya reaksi kimia tertentu, yang dipicu oleh cahaya. Basis bahan ini dimulai dengan lembaran kaca lebar dan berkembang menjadi film kecil berukuran 35 mm, sehingga mengurangi ukurannya dan juga meningkatkan kemudahan penggunaan.

Saat ini, dengan kemajuan teknologi digital, fotografi juga telah berkembang pesat selama tiga dekade terakhir. Kamera digital menggunakan prinsip optik yang sama dengan kamera film, dibantu oleh sensor digital yang menggantikan film. Jadi saat sensor menangkap sinyal cahaya, chip lain, atau kartu memori, menyimpan sinyal tersebut dalam bentuk data digital. Data ini kemudian dapat diedit di komputer, ponsel, dll., dan dibagikan melalui Internet ke media sosial, dan juga dapat dicetak dan ditempel di album foto [16]

Kamera webcam yang digunakan adalah logitech C920. Berikut adalah spesifikasi kamera yang digunakan:

Spesifikasi	
Max Resolution	1080p/30 fps - 720p/30 fps
Camera Mega Pixel	3
Focus Type	Auto Focus

Tabel 2.1: Spesifikasi Webcam

2.2.10 DC-DC Voltage Regulator

Regulator/konverter dc-dc atau nama lain yang dikenal dengan buck atau *boost regulator*, memberikan tegangan keluaran yang diatur secara stabil untuk mensuplai rangkaian listrik dan elektronik.

Konverter daya DC-DC digunakan dalam berbagai aplikasi, termasuk catu daya komputer pribadi, peralatan kantor, sistem tenaga pesawat ruang angkasa, komputer laptop, dan peralatan telekomunikasi, serta penggerak motor DC. Masukan ke konverter dc-dc tidak diatur tegangan DC V_g . Konverter menghasilkan tegangan keluaran yang diatur V , memiliki besarnya (dan mungkin polaritas) itu berbeda dengan V_g [17].

2.2.11 H-Bridge Motor Driver

H-bridge adalah sirkuit sederhana yang terdiri dari empat transistor, yaitu dua N-type dan 2 P-type. Transistor P-type terhubung di sisi *high* motor (bagian yang terhubung ke daya) dan transistor N-type ke sisi *low* motor (bagian yang terhubung ke *ground*). Setiap sisi motor terhubung ke amplifier *push-pull* sederhana yang setiap amplifiernya terhubung ke saklar [18].

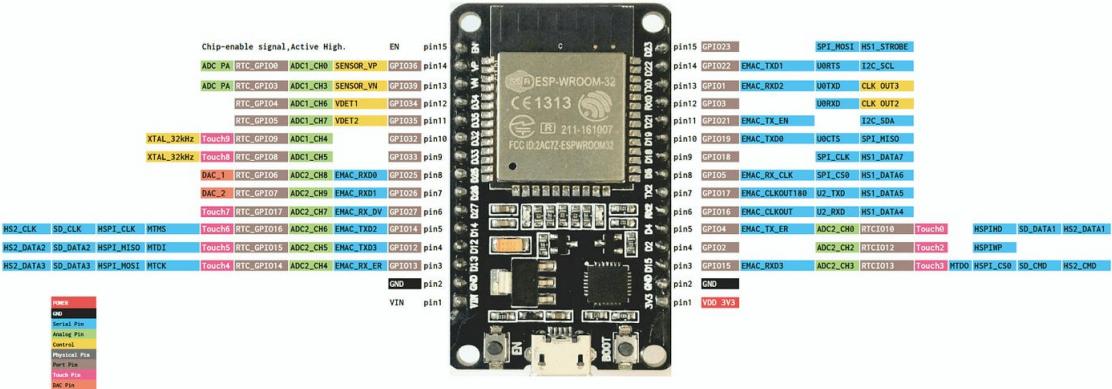


Gambar 2.8: H-Bridge Motor Driver

2.2.12 ESP32 Devkit V1

ESP32 Devkit V1 adalah mikrokontroller yang dikembangkan oleh Espressif Systems. ESP32 dirancang berdasarkan arsitektur Xtensa dual-core 32-bit. Hal ini berarti ESP32 memiliki dua buah inti prosesor yang dapat beroperasi secara independen. Salah satu fitur ESP32 yang paling penting adalah WiFi dan Bluetooth yang sudah terintegrasi sehingga memungkinkan ESP32 untuk berkomunikasi dengan perangkat lain secara mudah [19].

DOIT ESP32 DEVKIT V1 PINOUT



Gambar 2.9: ESP32

2.2.13 NUC

NUC, yang merupakan singkatan dari “Next Unit of Computing”, adalah komputer berbentuk kotak kecil yang dikembangkan oleh Intel. Meskipun ukurannya hanya beberapa inci atau lebih dalam, NUC berisi seluruh sistem yang dijejalkan ke dalam chassis kecilnya. NUC juga bisa dijual sebagai kit barebone yang perlu dirakit oleh pengguna untuk membuatnya bekerja[20].

Intel NUC memiliki motherboard yang hanya berukuran 4×4 inci (10.16×10.16 cm). Ada banyak pilihan tipe prosesor untuk NUC ini seperti Celeron, Pentium, Core i3, Core i7, hingga Core i9. Intel NUC juga menawarkan solusi Modular Computing dimana komponennya dapat dikostumisasi sesuai dengan kebutuhan. NUC yang digunakan pada penelitian ini adalah NUC dengan seri NUC11PAHi7.



Gambar 2.10: NUC

2.2.14 Arduino IDE

Arduino IDE adalah perangkat lunak sumber terbuka yang dirancang oleh Arduino.cc dan digunakan terutama untuk menulis, mengkompilasi, dan mengunggah kode ke hampir semua Modul Arduino. Ini adalah perangkat lunak Arduino resmi, yang membuat kompilasi kode sangat mudah sehingga bahkan orang biasa tanpa pengetahuan teknis sebelumnya dapat memulai proses pembelajaran. Arduino IDE berjalan pada Platform Java dan dilengkapi dengan fungsi

dan perintah bawaan yang memainkan peran penting dalam debugging, pengeditan, dan kompilasi kode [21].

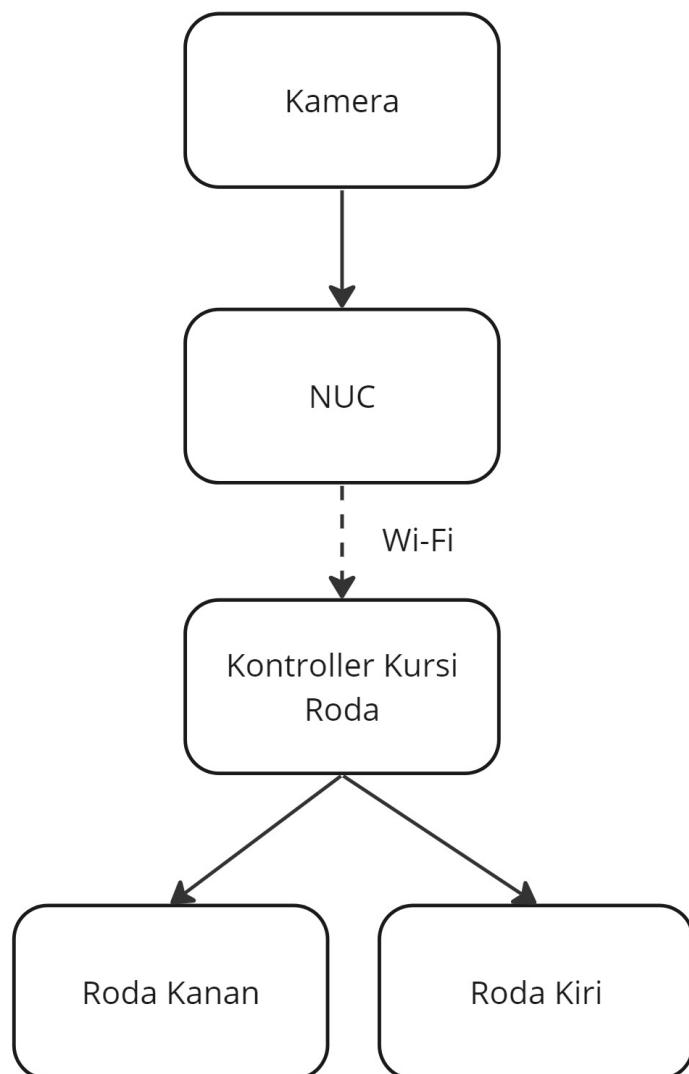
[Halaman ini sengaja dikosongkan]

BAB III

METODOLOGI

Metodologi berisi tentang langkah-langkah yang akan dilakukan pada penelitian kali ini. Langkah-langkah tersebut akan digambarkan dengan menggunakan blok diagram. Pada penelitian ini terdapat dua buah blok diagram yang masing-masing blok diagram akan merepresentasikan langkah-langkah untuk *hardware* dan *software*.

3.1 *Hardware*

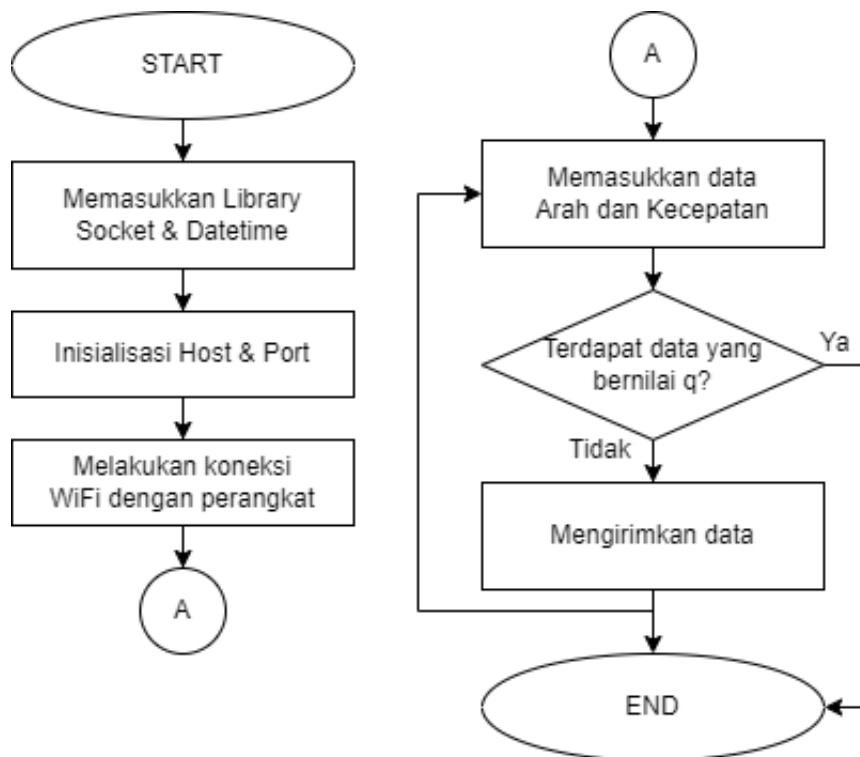


Gambar 3.1: Blok diagram *hardware*

3.1.1 NUC

Pada tahapan ini, dilakukanlah akuisisi data dengan menggunakan kamera yang telah dipasangkan pada bracket yang memungkinkan kamera sejajar dengan wajah pengguna kursi roda. Kemudian, dilakukanlah serangkaian pengolahan data citra wajah oleh mediapipe dan model.h5 yang sudah di-*training* dan berada pada NUC. Setelah itu, hasil klasifikasi yang didapatkan akan dikirimkan oleh NUC ke ESP32 dengan melalui WiFi agar output yang didapatkan bisa desainimplementasikan pada motor kursi roda. Untuk dapat melakukan tahapan tersebut, NUC harus terhubung dengan *access point* ESP32.

Untuk proses pengiriman data hasil pengklasifikasian ke ESP32, telah direpresentasikan pada flowchart Gambar 3.2 sesuai dengan penelitian yang telah dilakukan [22] dengan judul Perancangan Sistem Kontrol Motor Kursi Roda Secara Nirkabel Berbasis ESP32. Pengiriman ini menggunakan bahasa pemrograman Python dan dirancang untuk mengirim dalam bentuk string. Agar dapat membuat koneksi untuk mengirim dan menerima data melalui jarigan, digunakan beberapa *library* seperti socket, time, dan datetime. Selain itu, IP Address yang dimiliki oleh *Access Point* ESP32 dijadikan sebagai variabel *host*. Dijadikan juga port 80 sebagai variabel portnya. Selanjutnya, akan dilakukan pengkoneksi ke server yang telah ditentukan sesuai dengan IP Address dan port yang telah diinputkan. Program akan berjalan secara berulang-ulang tanpa batas dan akan terus-menerus menerima inputan dari pengguna.

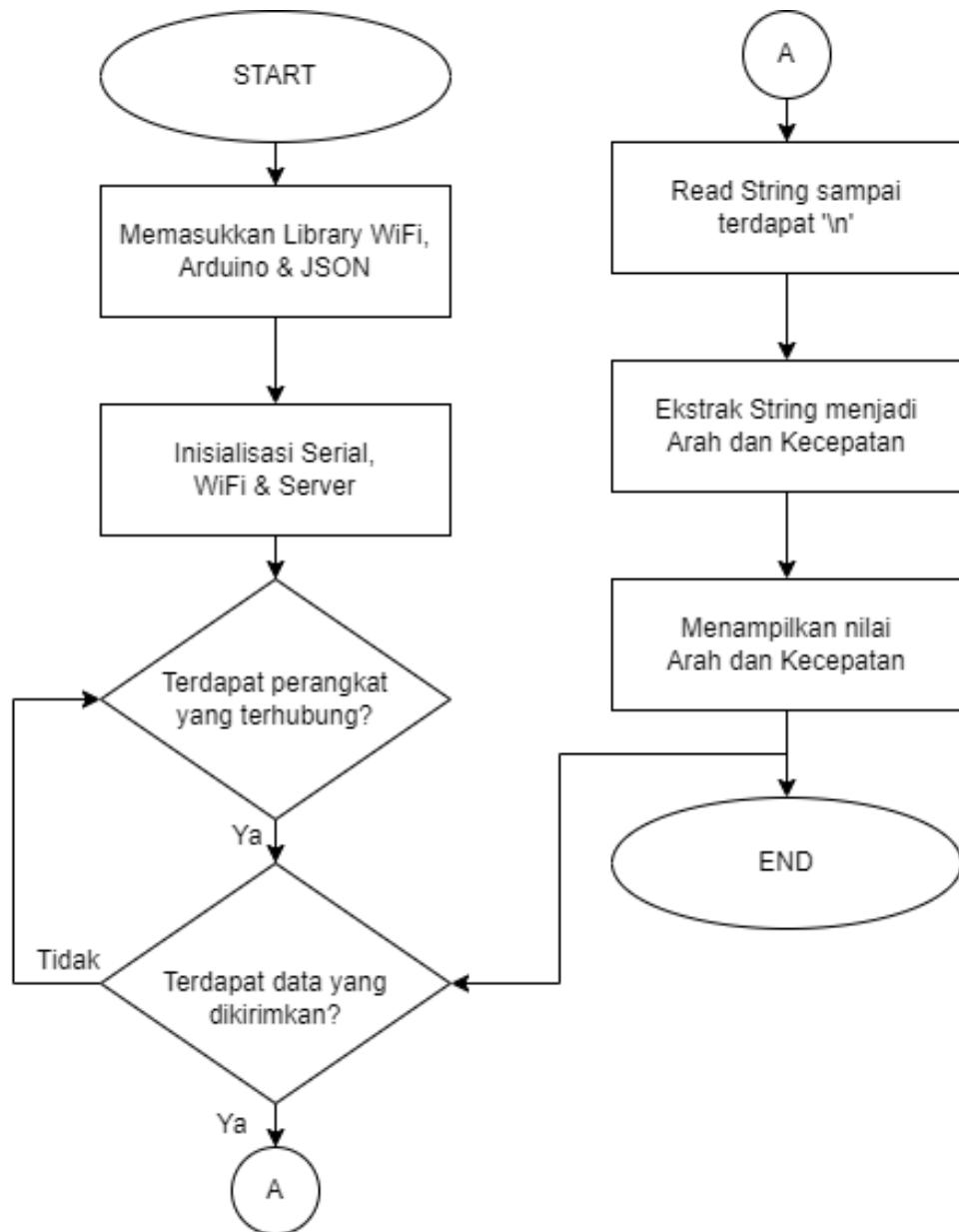


Gambar 3.2: Flowchart Mengirim Data String Melalui WiFi [22]

3.1.2 Kontroller Kursi Roda

Pada tahap ini, ESP32 akan mendapatkan arah hasil klasifikasi pose wajah dalam bentuk huruf A, B, C, D, atau E. Menurut penelitian yang telah dilakukan sebelumnya [22], proses ESP32 menerima data string dari NUC telah divisualisasikan pada flowchart Gambar 3.3. Program ini dirancang sebagai server WiFi yang akan menerima data dari perangkat yang terhubung

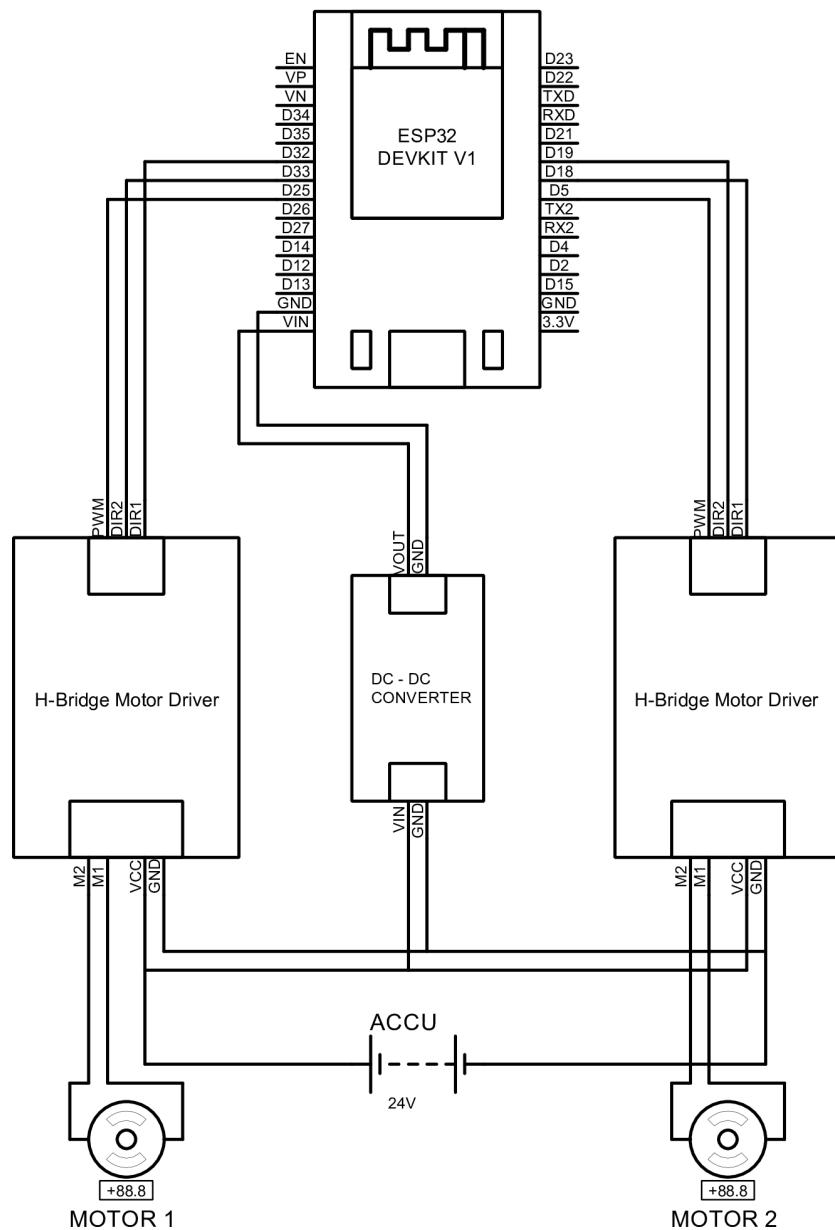
yang dalam pengujian kali ini adalah NUC. Jika server terhubung dengan *client*, maka program akan memasuki fungsi `if` yang akan mengulang-ulang terus dan menunggu adanya data yang dikirimkan oleh *client*. Data yang diterima akan dibaca secara terus-menerus dan akan dimasukkan kedalam variabel `receivedData`. Setelah itu, data yang diterima akan di-*decode* menjadi bentuk string. Program akan terus berjalan dan menunggu adanya data yang dikirimkan oleh *client*.



Gambar 3.3: Flowchart Menerima Data String Melalui Access Point WiFi Pada ESP32 [22]

Berdasarkan penelitian yang telah dilakukan [22], telah dirancang sebuah kontroller motor kursi roda dengan skematik alatnya seperti yang tertera pada Gambar 3.4. ESP32 akan terhubung dengan dua buah H-Bridge Motor Driver dan sebuah DC-DC Converter. Masing-masing H-Bridge Motor Driver akan terhubung ke motor roda kiri dan motor roda kanan agar dapat menggerakan roda kursi roda. DC-DC Converter akan terhubung ke baterai sebagai sumber daya. Kontroller ini akan mengontrol pergerakan motor kursi roda berdasarkan arah yang

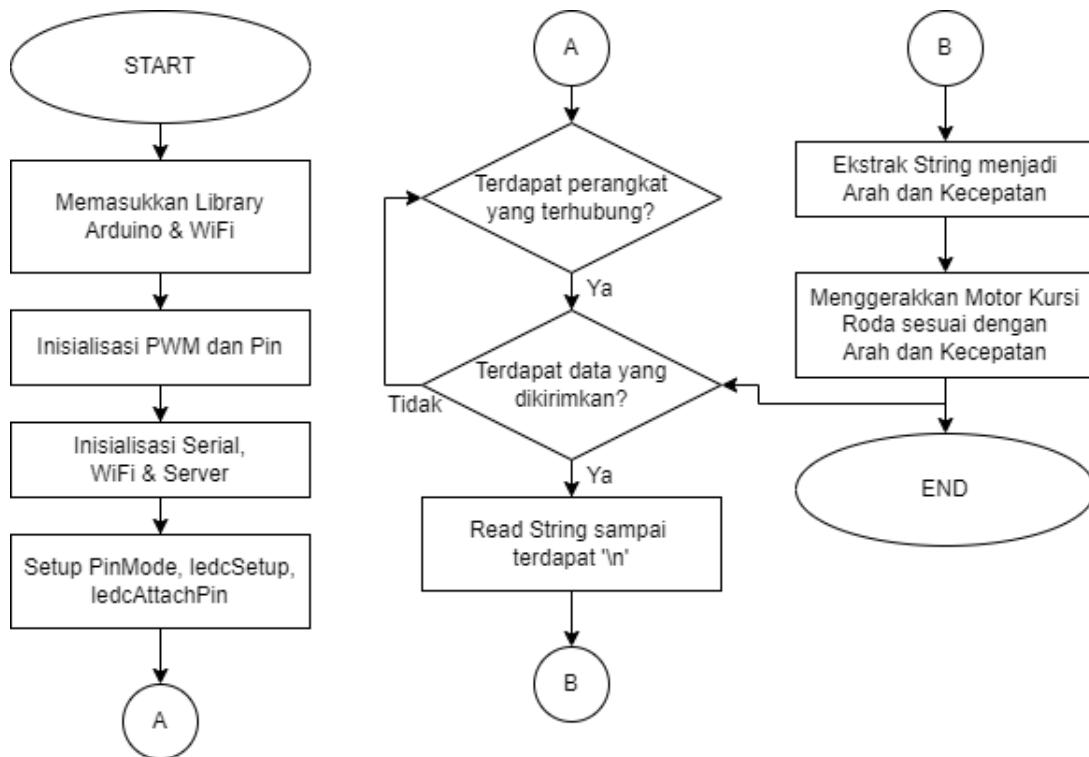
didapatkan dari hasil klasifikasi pose wajah.



Gambar 3.4: Skematik [22]

Untuk proses kontrol motor kursi roda, telah direpresentasikan pada flowchart Gambar 3.5 sesuai dengan penelitian yang telah dilakukan [22]. Program ini dirancang untuk mengendalikan pergerakan motor DC dengan cara memproses perintah yang telah diterima melalui koneksi jaringan wifi. Untuk mengontrol pergerakan kursi roda, terdapat dua buah metode. Metode pertama adalah differential drive yang mana ketika kursi roda akan berbelok ke kanan, maka roda kanan akan bergerak mundur dan berbelok ke kiri sedangkan roda kirinya akan bergerak maju dan berbelok ke kanan. Ketika akan berbelok ke kiri, maka akan terjadi hal sebaliknya. Ketika akan bergerak maju atau mundur makan kedua roda akan bergerak secara bersamaan kearah yang diinginkan. Metode kedua adalah pergerakan biasa yang mana ketika kursi roda akan berbelok ke kanan, maka roda kanan akan diam sedangkan roda kirinya akan

bergerak maju dan berbelok ke kanan. Ketika akan berbelok ke kiri, maka akan terjadi hal sebaliknya. Ketika akan bergerak maju atau mundur makan kedua roda akan bergerak secara bersamaan kearah yang diinginkan. Untuk penelitian kali ini akan digunakan metode kedua untuk menggerakan roda pada kursi roda.



Gambar 3.5: Flowchart Kontrol Motor Kursi Roda Melalui Access Point WiFi [22]

3.2 Software



Gambar 3.6: Blok diagram *software*

3.2.1 Citra

Data yang digunakan sebagai dataset adalah citra pose wajah yang diambil secara langsung dari seseorang. Data citra pose wajah diambil dengan cara menggunakan kamera webcam yang terhubung pada laptop dan kemudian setiap frame dari video tersebut akan dijadikan satu buah citra pose wajah. Kemudian, citra wajah yang telah didapat akan diproses apakah terdapat citra wajah atau tidak. Kemudian, akan dilakukan pendekripsi secara *realtime* untuk mendekripsi ada tidaknya citra wajah.

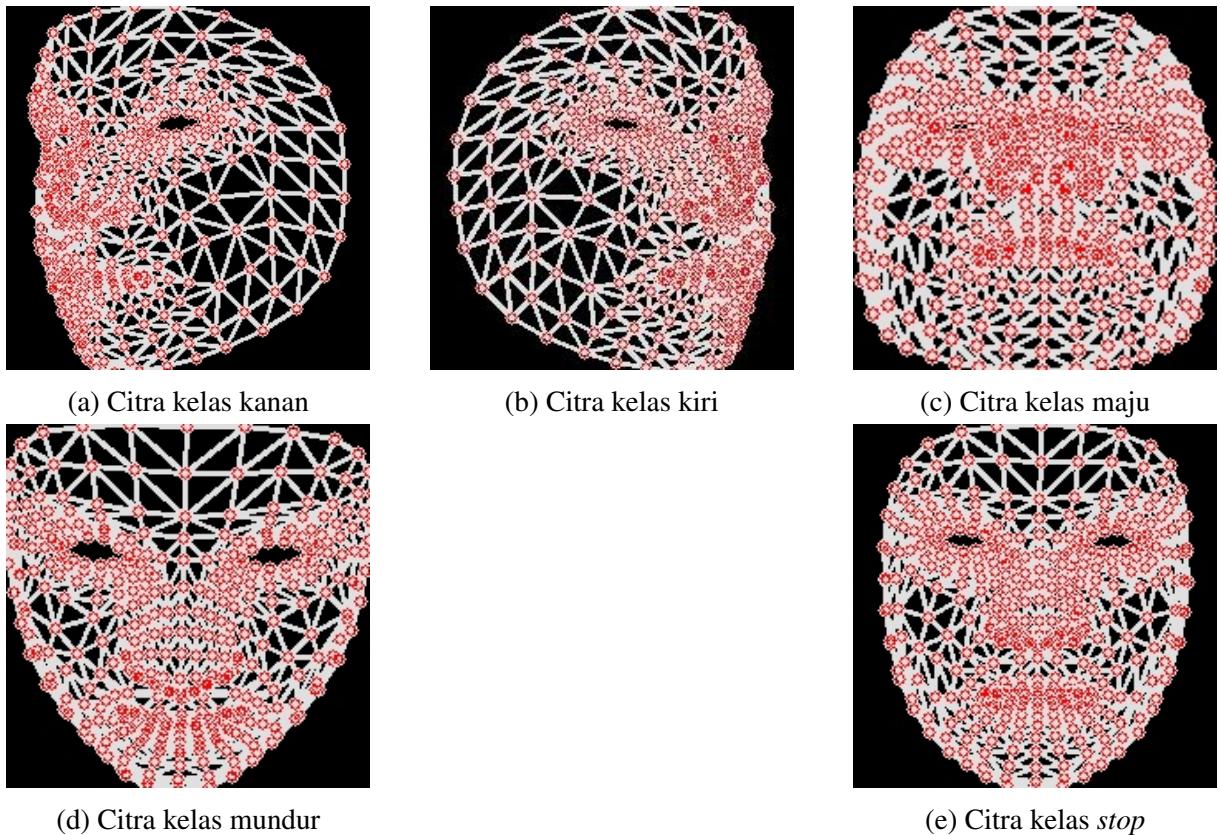
3.2.2 Ekstraksi Fitur Landmark

Setelah terdeteksi adanya wajah, mediapipe akan melakukan tracking dengan menggunakan *pre-trained* model yang dimiliki dari mediapipe itu sendiri. Kemudian untuk setiap *frame* yang terdeteksi akan diproses data gambarnya. Mediapipe akan mengidentifikasi *Region of Interest* (ROI) dimana sekiranya terdapat fitur wajah. *Pre-trained* model kemudian akan memprediksi koordinat titik-titik tertentu (landmark) pada ROI. Landmark yang didapat adalah dalam bentuk koordinat *pixel*. Dilakukanlah normalisasi yang mengubah koordinat *pixel* menjadi skala relatif berdasarkan dimensi frame yang telah dideteksi. Normalisasi ini dilakukan agar landmark tetap konsisten meskipun dalam ukuran frame dan resolusi yang berbeda. Selanjutnya, untuk meningkatkan akurasi dan independensi dataset, dibuatlah gambar yang hitam sepenuhnya. Hal ini akan mengisolasi landmark yang akan digambar dan membuatnya menjadi lebih mudah terdeteksi ketika melakukan training nantinya.

Kemudian, mediapipe menggambarkan fitur-fitur landmark wajah pada background hitam. Terdapat tiga jenis style landmark untuk wajah yang dapat dibuat oleh mediapipe, yaitu FACE-CEMESH_TESSELATION, FACEMESH_CONTOURS, dan FACEMESH_IRISES. Pada FACE-CEMESH_TESSELATION, landmark digambarkan dengan sejumlah segitiga yang membentuk permukaan wajah. Setiap segitiga yang digambar menghubungkan tiga buah titik landmark pada wajah. Sementara itu, FACEMESH_CONTOURS, landmark yang digambarkan hanya merepresentasikan kontur atau garis luar dari wajah tanpa menggambarkan permukaan wajah secara detail. Di sisi lain, FACEMESH_IRISES, landmark yang digambar hanya pada iris mata saja. Pada penelitian kali ini akan digunakan FACE-CEMESH_TESSELATION karena landmark yang digambar lebih detail dan kompleks sehingga akan meningkatkan akurasi pendekripsi pose wajah. Pada dokumentasi Mediapipe mengenai face landmark, telah disebutkan bahwa pada landmark wajah terdapat titik-titik yang berjumlah 468 yang diperkirakan dalam ruang 3D secara real-time. Akan tetapi, pada penelitian kali ini hanya menggunakan landmark 2D yang memungkinkan representasi wajah menjadi invariant terhadap skala.

3.2.3 Klasifikasi Landmark

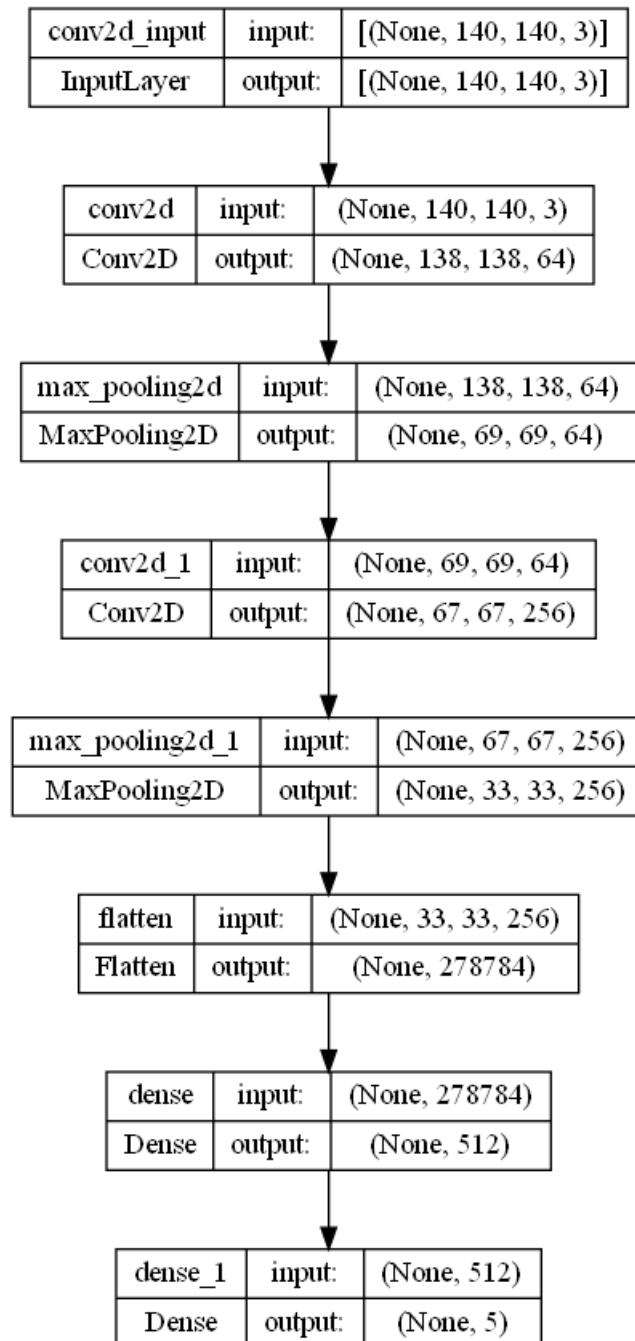
Untuk dapat melakukan pengklasifikasian, diperlukan sebuah model yang merupakan hasil *training* dari *Convolutional Neural Network* (CNN). Model yang didapat dari hasil *training* akan melakukan pengklasifikasian secara *real-time* setiap kali wajah terdeteksi. Model CNN yang digunakan pada penelitian kali ini memiliki lima kelas yang akan merepresentasikan arah gerak kursi roda. Kelima kelas tersebut adalah kanan yang direpresentasikan dengan wajah menghadap kanan, kiri yang direpresentasikan dengan wajah menghadap kiri, maju yang direpresentasikan dengan wajah menghadap atas, mundur yang direpresentasikan dengan wajah menghadap bawah, dan *stop* yang direpresentasikan dengan wajah menghadap depan. Pada Gambar 3.7 dapat dilihat contoh data citra wajah yang digunakan untuk tiap-tiap kelasnya.



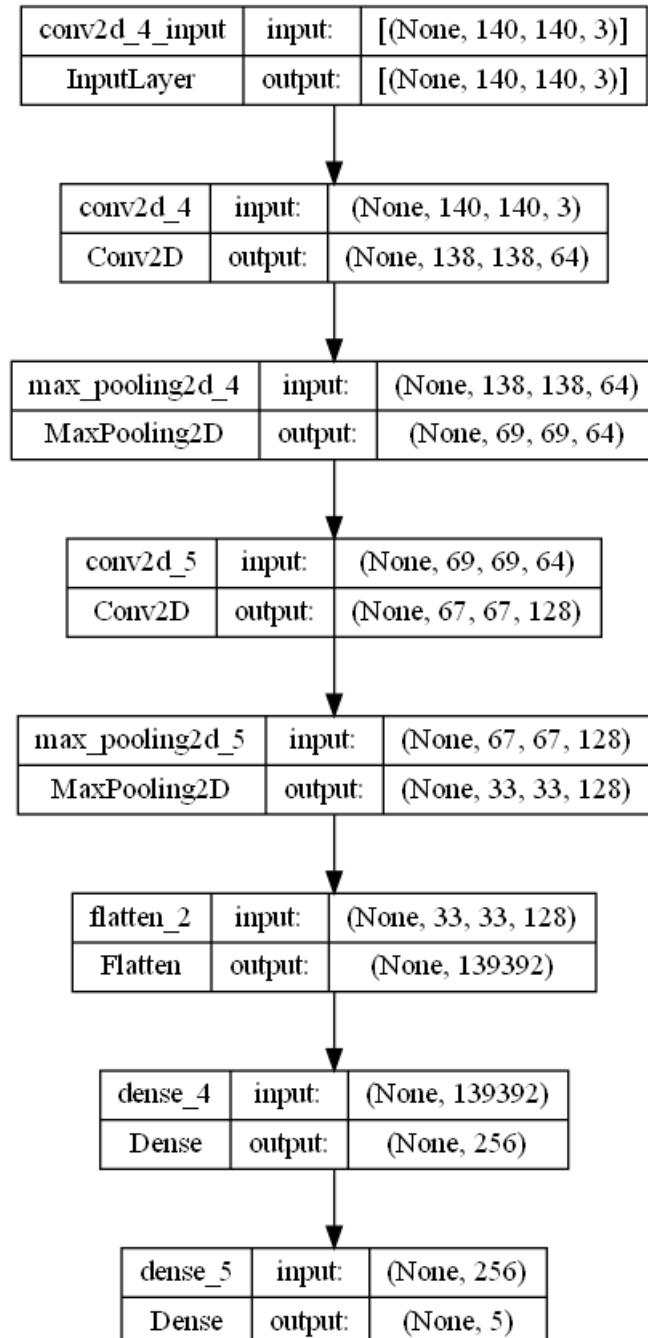
Gambar 3.7: Contoh citra wajah

Terdapat 2 model dengan arsitektur berbeda pada penelitian kali ini. Arsitektur CNN yang digunakan untuk mendapatkan pertama model pada penelitian kali ini memiliki 7 *layer*. Layer pertama adalah Convolutional 2D layer yang melakukan 64 filter dengan ukuran kernel (3,3) pada input dengan dimensi 140 x 140 pixels dan 3 /emphcolor channels. Fungsi utama layer ini adalah untuk mengekstraksi fitur-fitur dari gambar yang diinputkan dengan cara menggeser filter melintasi gambar yang mana akan membantu dalam melakukan pendekripsi tepi, warna, gradien, dan lain-lain. Setiap filter akan menghasilkan map fitur yang merepresentasikan fitur-fitur yang ada pada gambar input. Layer ini juga merupakan layer inputnya. Selanjutnya layer kedua adalah *hidden layer* yang pertama yang berupa Max Pooling 2D dengan *pool size* (2,2). Layer ini akan mengurangi dimensi spasial pada map fitur dengan cara mengambil nilai maksimum yang ada pada *pool* 2 x 2. Hal ini akan mengurangi kompleksitas komputasi dengan mengecilkan map fitur dan membantu membuat pendekripsi menjadi invariant terhadap skala dan orientasi. Layer ketiga adalah *hidden layer* kedua yang merupakan Convolutional 2D dengan 256 filter, *kernel size* (3,3), dan ReLU sebagai fungsi aktivasinya. Layer tersebut akan melakukan pemrosesan lebih lanjut terhadap map fitur yang telah didapatkan dari layer sebelumnya, sehingga dapat mengenali pola yang lebih rumit seperti tekstur dan bentuk. Layer keempat adalah Max Pooling 2D dengan *pool size* (2,2) yang akan terus mengecilkan ukuran map fitur, sehingga dapat meringkas semua fitur-fitur yang telah didapatkan dengan data yang lebih sedikit. Layer kelima adalah Flatten yang menjadikan output dari layer sebelumnya menjadi array satu dimensi atau menjadi satu vektor yang panjang. Layer ini berfungsi sebagai jembatan antara convolutional layer dengan dense layer karena dense layer memerlukan data 1D. Layer selanjutnya atau layer keenam adalah Dense atau *fully connected* layer pertama yang memiliki 512 unit dan menggunakan ReLU sebagai fungsi aktivasinya. Layer ini akan mengambil fitur-

fitur yang sudah di-*flatten* dan kemudian mempelajari kombinasi fitur yang non-linear. Layer terakhir atau ketujuh yang digunakan adalah dense atau *fully connected* layer kedua yang memiliki jumlah unit yang sama dengan jumlah kelas yang akan diprediksi, yaitu 5. Layer ini juga merupakan layer output. Arsitektur CNN untuk model pertama divisualisasikan pada Gambar 3.8. Untuk model kedua, memiliki arsitektur yang mirip dengan model pertama akan tetapi terdapat sedikit perbedaan pada jumlah filter dan jumlah unit pada dense layer. Model kedua memiliki 64 filter pada layer pertama, 128 unit pada layer ketiga, dan 256 unit pada dense layer pertama. Arsitektur CNN untuk model kedua divisualisasikan pada Gambar 3.9.

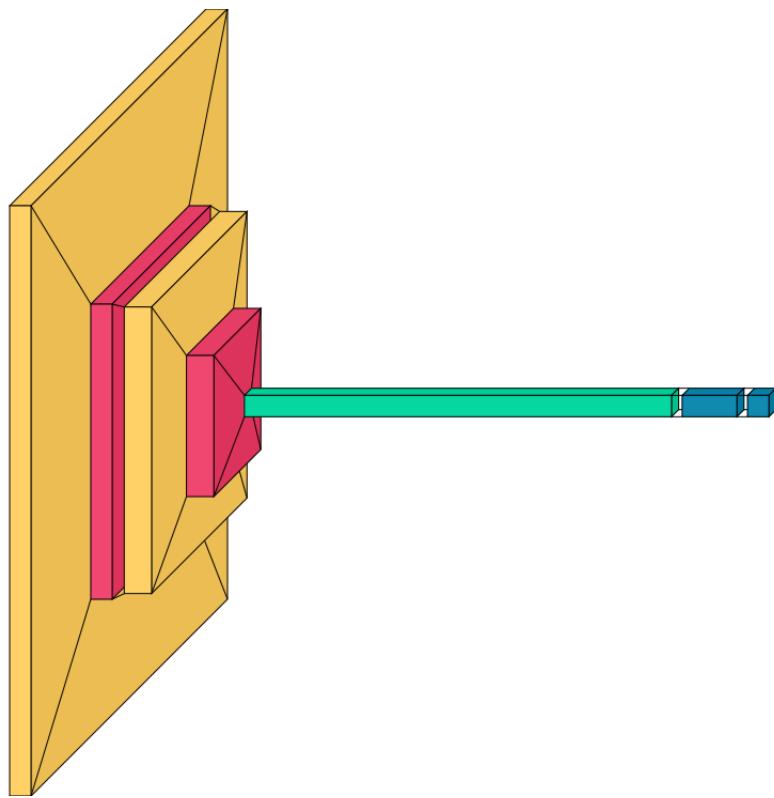


Gambar 3.8: Arsitektur CNN Model 1

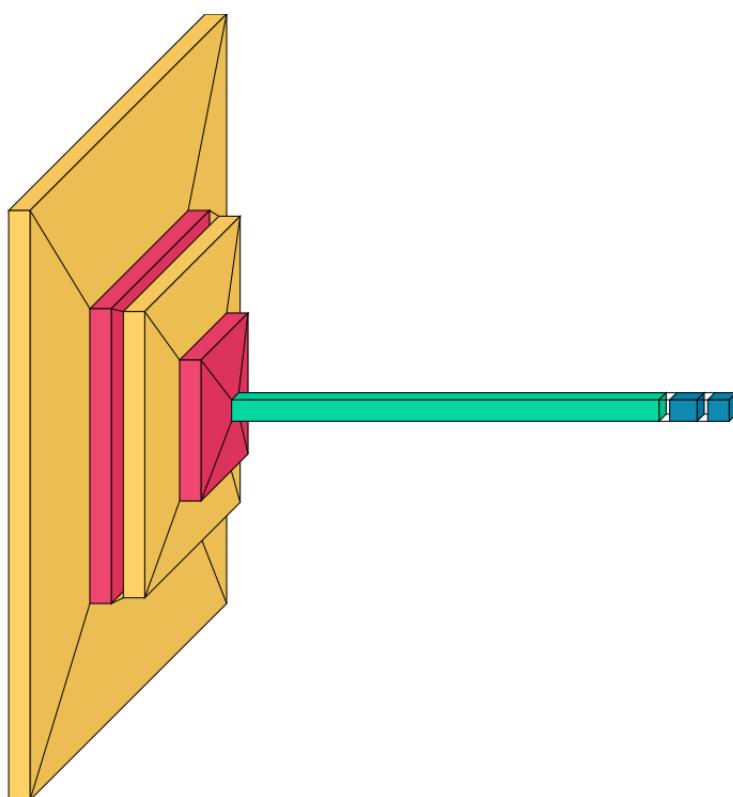


Gambar 3.9: Arsitektur CNN Model 2

Pada gambar 3.11, setiap jenis layer direpresentasikan dengan warna-warna yang berbeda. Warna kuning pada layer merepresentasikan Convolutional Layer, warna merah merepresentasikan layer Max Pooling, warna hijau merepresentasikan layer *flatten*, dan warna biru merepresentasikan layer *dense*.



Gambar 3.10: Diagram Arsitektur CNN Model 1



Gambar 3.11: Diagram Arsitektur CNN Model 2

Sistem kontrol yang digunakan untuk menggerakkan kursi roda terbagi menjadi lima kelas arah yang memiliki indeks dan direpresentasikan dengan menggunakan huruf. Hal ini dilakukan karena perbedaan antara cara kerja Python dan Arduino IDE. Indeks ini digunakan pada saat penghitungan pada proses klasifikasi. Ketika citra wajah diklasifikasikan, citra wajah tersebut akan dicocokkan ke indeks. Jika tidak cocok, maka indeks akan dilanjutkan dari 0 ke 1, 1 ke 2 sampai data citranya cocok barulah diklasifikasikan kepada kelas yang direpresentasikan oleh indeks tersebut.

Tabel 3.1: Kelas yang Digunakan

Index Kelas	Index Arah	Hasil Klasifikasi
0	E	Kanan
1	A	Kiri
2	B	Maju
3	D	Mundur
4	C	Stop

[Halaman ini sengaja dikosongkan]

BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini akan dipaparkan mengenai hasil pengujian dan pembahasan dari penelitian yang telah dipaparkan pada metodologi. Dipaparkan juga mengenai skenario pengujian yang dilakukan.

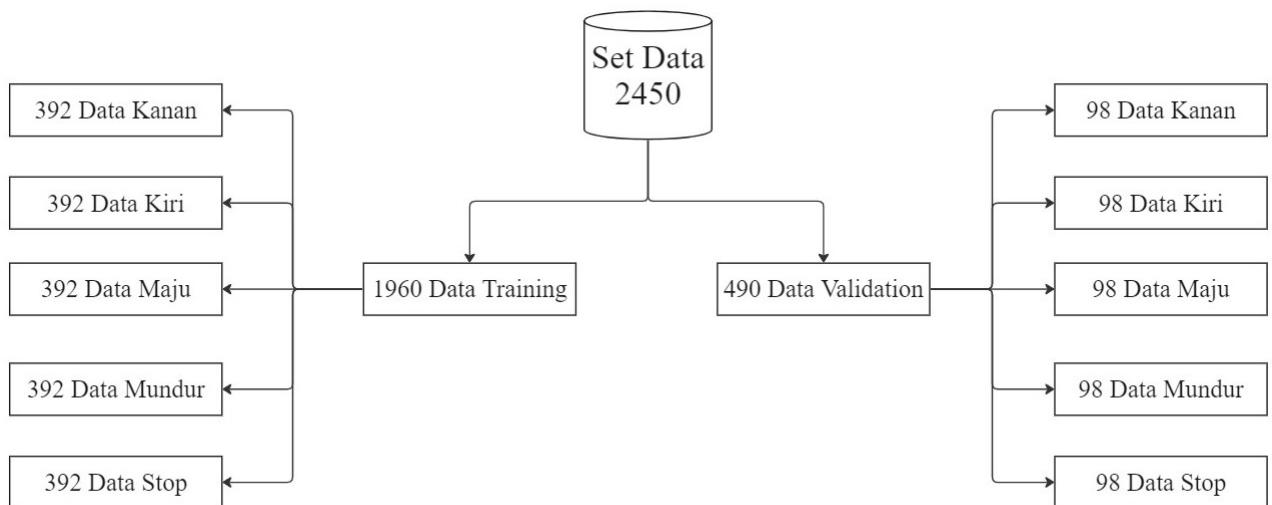
4.1 Skenario Pengujian

Pengujian dilakukan untuk mengetahui performa model dalam melakukan klasifikasi. Skenario pengujian yang akan dilakukan adalah sebagai berikut:

1. Pengujian Performa Model
2. Pengujian Performa Model dengan Variasi Jarak Kamera
3. Pengujian Performa Model dengan Variasi Pencahayaan
4. Pengujian Kecepatan FPS
5. Pengujian Waktu Respons Kontrol Kursi Roda
6. Pengujian kestabilan Performa Kursi Roda

4.2 Pengujian Performa Model

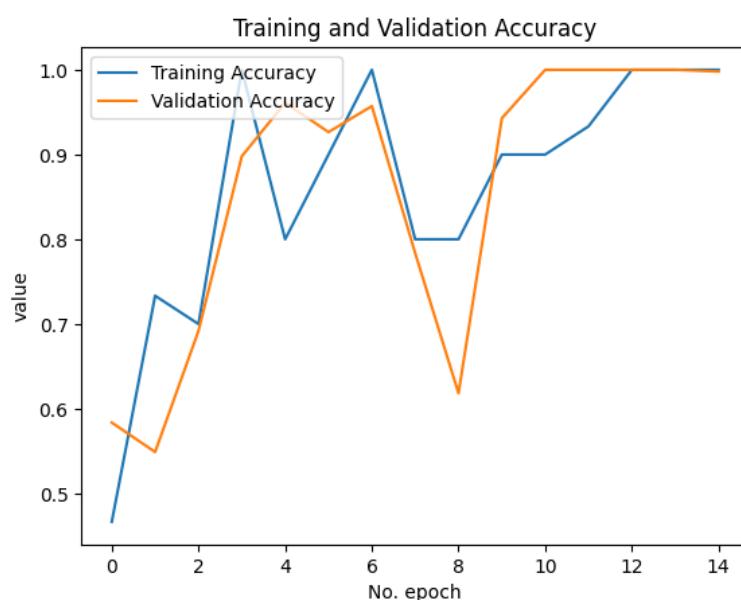
Dalam pembuatan model *Convolutional Neural Network* (CNN) yang akan digunakan, dilakukan pengelompokan data citra kedalam lima kelas yang kemudian akan *ditraining*. Lima kelas tersebut adalah Kanan, Kiri, Maju, Mundur, dan *Stop*. Total data citra yang dijadikan sebagai dataset berjumlah 2450. Data tersebut kemudian dibagi menjadi dua buah dataset, yaitu dataset *training* dan dataset *validation* dengan rasio pembagian 80:20, sehingga data yang digunakan berjumlah 392 data *training* dan 98 data *validation* untuk tiap kelasnya. Pembagian dataset yang dilakukan telah divisualisasikan pada diagram Gambar 4.1. Pada skenario pengujian ini terdapat dua buah model yang memiliki perbedaan pada jumlah filet di layer yang digunakan. Pengujian ini dilakukan untuk mendapatkan model yang memiliki performa terbaik dalam melakukan klasifikasi. Model yang memiliki performa terbaik akan digunakan untuk pengujian selanjutnya.



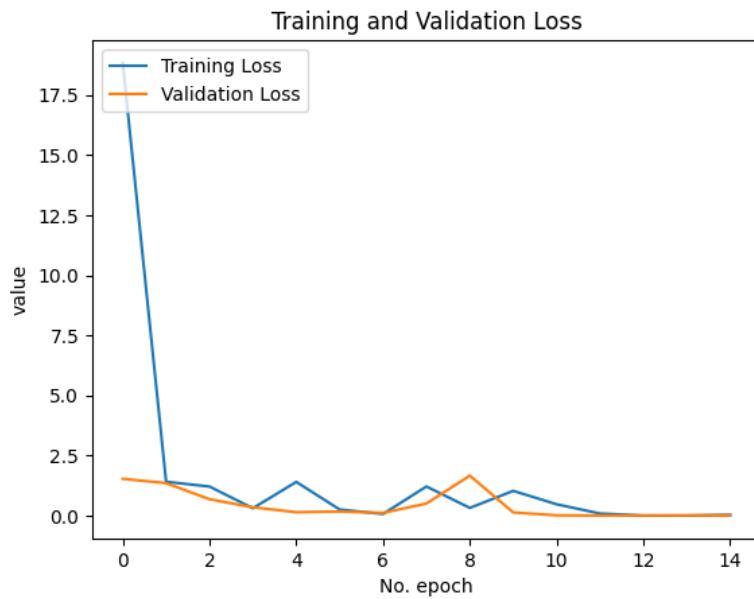
Gambar 4.1: Diagram Pembagian Dataset

4.2.1 Pengujian Model Pertama

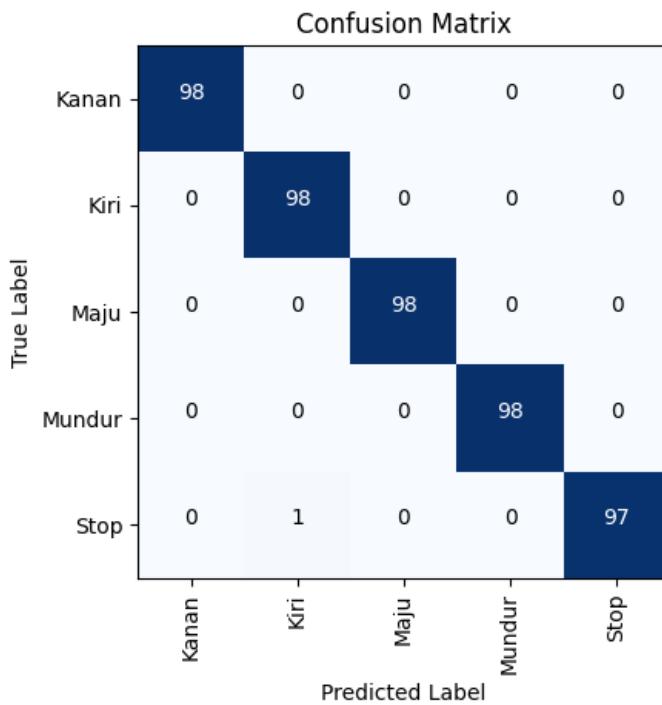
Model pertama memiliki 64 filter pada layer pertama, 256 unit pada layer ketiga, dan 512 unit pada dense layer pertama. Berdasarkan grafik yang didapat setelah dilakukan *training*, model ini memiliki *training accuracy* sebesar 100% dan *validation accuracy* sebesar 99.80% seperti pada Gambar 4.2. Model ini juga memiliki *training loss* sebesar 0.0307 dan *validation loss* sebesar 0.0104 yang dapat dilihat pada Gambar 4.3. *Training loss* dan *validation loss* yang didapat sangatlah kecil yang berarti error yang didapat sangatlah kecil yang berarti pendeksi model sangatlah efisien.



Gambar 4.2: Accuracy Hasil Pelatihan Model



Gambar 4.3: Model Loss Hasil Pelatihan Model



Gambar 4.4: Confusion Matrix Hasil Pelatihan Model Pertama

Dalam pengujian model CNN yang telah dibuat terhadap data validasi, dengan menggunakan perhitungan *confusion matrix*, didapatkan hasil klasifikasi seperti yang telah divisualisasikan pada Gambar 4.4. Dari Gambar tersebut, dapat dilihat bahwa dari 490 data pada model yang diujikan terhadap data citra validasi, terdapat 1 data citra yang hasil prediksinya tidak sesuai dengan data aktual yang diujikan. Satu data citra tersebut merupakan data citra dari kelas *stop* yang 1 data terdeteksi sebagai data citra kelas *kiri*. Persentase keberhasilan pada kelas *stop*

adalah sekitar 98,98% dengan jumlah 97 data dan persentase ketidakberhasilannya adalah sekitar 1,02% dengan jumlah 1 data. Pada Tabel 4.1 dapat dilihat terdapat 1 data kelas *stop* yang terdeteksi sebagai *false negative* dan 97 data yang terdeteksi sebagai *true positive*. Selain itu, dapat dilihat juga terdapat 1 data kelas kiri yang terdeteksi sebagai *false positive* dan 391 data yang terdeteksi sebagai *true negative*.

Tabel 4.1: Hasil Klasifikasi Model Pertama

	TP	TN	FP	FN
Kanan	98	392	0	0
Kiri	98	391	1	0
Maju	98	392	0	0
Mundur	98	392	0	0
Stop	97	392	0	1

Pada Tabel 4.2 dapat dilihat bahwa nilai *accuracy* pada kiri adalah sebesar 0.998 dan nilai *precisionnya* 0.9899. Hal ini menyebabkan kelas kiri memiliki *F1-Score* sebesar 0.9949. Sedangkan pada kelas *stop*, nilai *accuracynya* adalah 0.998 dan nilai *recallnya* 0.9898. Hal ini menyebabkan kelas *stop* memiliki *F1-Score* sebesar 0.9949. Pada kelas kanan, maju, dan mundur memiliki nilai *accuracy*, *precision*, *recall*, dan *F1-Score* sebesar 1.0000.

Tabel 4.2: Hasil Validasi Nilai Model Pertama

	Accuracy	Precision	Recall	F1-Score
Kanan	1.0000	1.0000	1.0000	1.0000
Kiri	0.9980	0.9899	1.0000	0.9949
Maju	1.0000	1.0000	1.0000	1.0000
Mundur	1.0000	1.0000	1.0000	1.0000
Stop	0.9980	1.0000	0.9898	0.9949

Pada pengujian ini, dilakukan 480 kali pendekripsi yang mana kemudian akan dihitung perbandingan terdeteksi dan tidaknya pada tiap model yang diujikan. Hasil yang didapatkan dari pengujian ini dapat dilihat pada Tabel 4.3. Dari hasil pengujian ini, dapat dilihat bahwa model pertama memiliki persentase keberhasilan sebesar 98,125% dan persentase ketidakberhasilan sebesar 1,875%. Hal ini menunjukkan bahwa model pertama memiliki performa yang sangat baik dalam melakukan klasifikasi secara *real-time*.

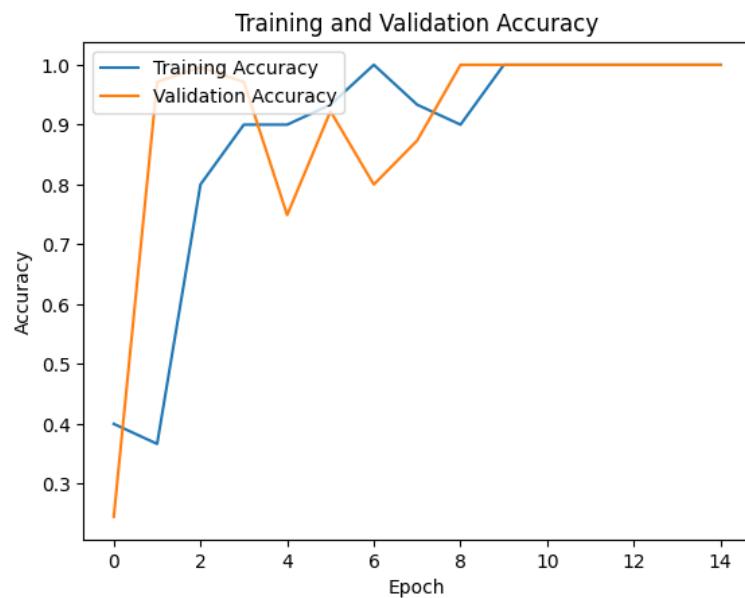
Tabel 4.3: Hasil Pengujian Keterdeteksian Model Pertama

Jumlah Keberhasilan:	471
Jumlah Ketidakberhasilan:	9
Persentasi Keberhasilan:	98,125%
Persentasi Ketidakberhasilan:	1,875%

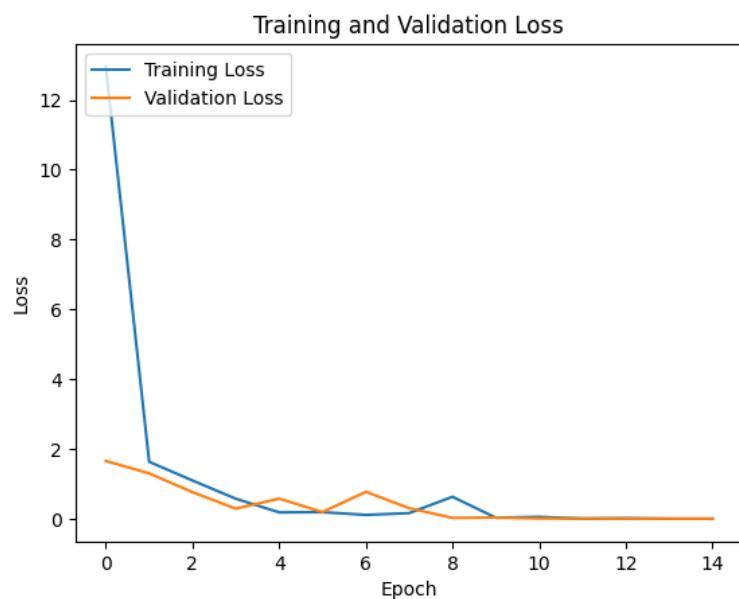
4.2.2 Pengujian Model Kedua

Model kedua memiliki 64 filter pada layer pertama, 128 unit pada layer ketiga, dan 256 unit pada dense layer pertama. Berdasarkan grafik yang didapat setelah dilakukan *training*,

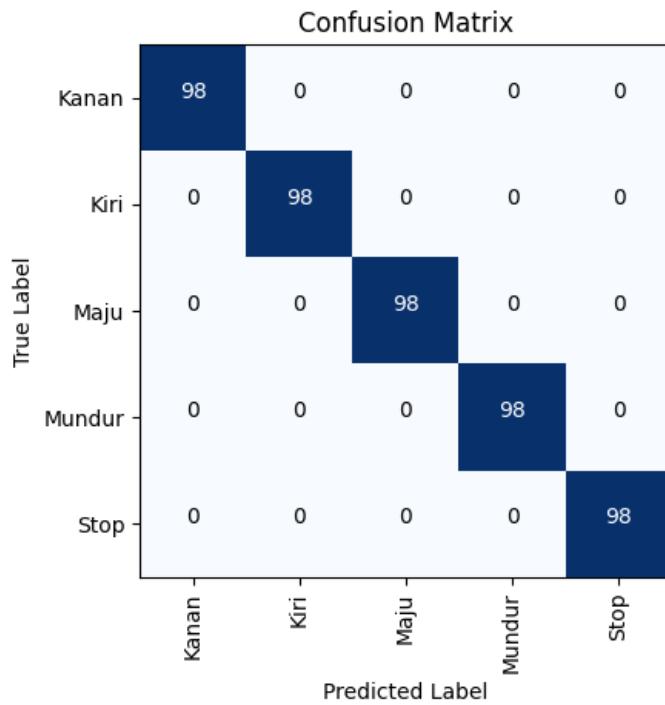
model ini memiliki *training accuracy* sebesar 100% dan *validation accuracy* sebesar 100.00% seperti pada Gambar 4.5. Model ini juga memiliki *training loss* sebesar 0.0005 dan *validation loss* sebesar 0.0020 yang dapat dilihat pada Gambar 4.6. *Training loss* dan *validation loss* yang didapat sangatlah kecil yang berarti error yang didapat sangatlah kecil yang berarti pendekstrian model sangatlah efisien.



Gambar 4.5: Accuracy Hasil Pelatihan Model Kedua



Gambar 4.6: Model Loss Hasil Pelatihan Model Kedua



Gambar 4.7: Confusion Matrix Hasil Pelatihan Model Kedua

Dalam pengujian model CNN yang telah dibuat terhadap data validasi, dengan menggunakan perhitungan *confusion matrix*, didapatkan hasil klasifikasi seperti yang telah divisualisasikan pada Gambar 4.7. Dari Gambar tersebut, dapat dilihat bahwa dari 490 data pada model yang diujikan terhadap data citra validasi, tidak terdapat data citra yang hasil prediksinya tidak sesuai dengan data aktual yang diujikan. Hal ini membuat persentase keberhasilan pada kelas kiri, kanan, maju, dan mundur adalah sebesar 100% dan persentase ketidakberhasilannya adalah sebesar 0%. Pada Tabel 4.4 dapat dilihat tidak terdapat data yang terdeteksi sebagai *false negative* dan *false positive* pada setiap kelasnya.

Tabel 4.4: Hasil Klasifikasi Model Kedua

	TP	TN	FP	FN
Kanan	98	392	0	0
Kiri	98	392	0	0
Maju	98	392	0	0
Mundur	98	392	0	0
Stop	98	392	0	0

Pada Tabel 4.5 pada semua kelas, nilai *accuracy*, *precision*, dan *recall* bernilai 1.0000. Hal ini membuat semua kelas memiliki nilai *F1-Score* sebesar 1.0000. Hal ini menunjukkan bahwa model kedua memiliki performa yang sangat baik dalam melakukan klasifikasi.

Tabel 4.5: Hasil Validasi Nilai Model Kedua

	Accuracy	Precision	Recall	F1-Score
Kanan	1.0000	1.0000	1.0000	1.0000
Kiri	1.0000	1.0000	1.0000	1.0000
Maju	1.0000	1.0000	1.0000	1.0000
Mundur	1.0000	1.0000	1.0000	1.0000
Stop	1.0000	1.0000	1.0000	1.0000

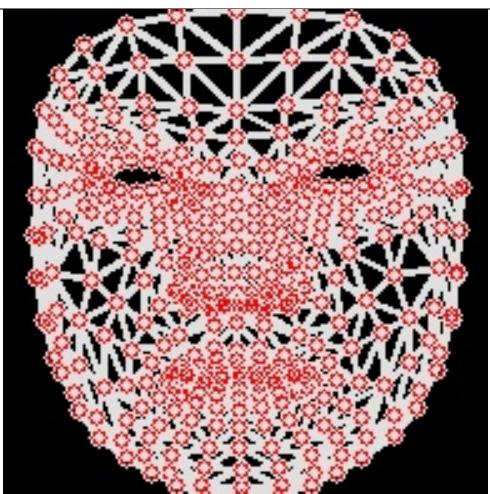
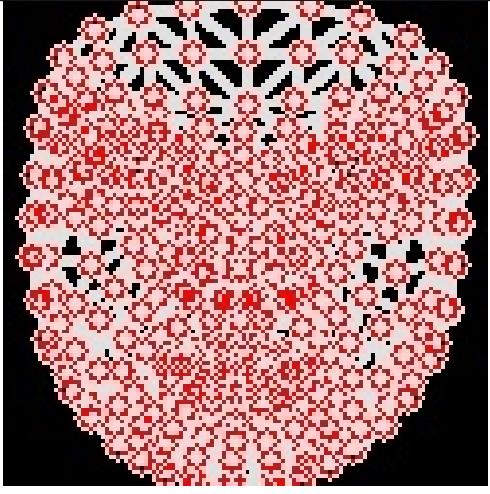
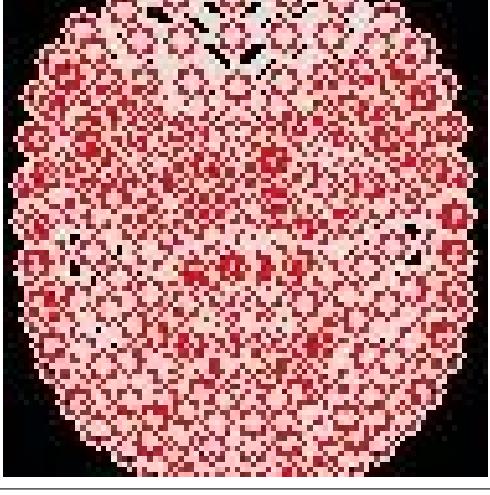
Pada pengujian ini, dilakukan 480 kali pendekripsi yang mana kemudian akan dihitung perbandingan terdeteksi dan tidaknya pada tiap model yang diujikan. Hasil yang didapatkan dari pengujian ini dapat dilihat pada Tabel 4.6. Dari hasil pengujian ini, dapat dilihat bahwa model kedua memiliki persentase keberhasilan sebesar 69,375% dan persentase ketidakberhasilan sebesar 30,625%. Hal ini menunjukkan bahwa performa model kedua dalam melakukan klasifikasi secara *real-time* tidak sebaik model pertama. Karena model pertama memiliki performa yang lebih baik dibandingkan model kedua, maka model pertama yang akan digunakan untuk pengujian selanjutnya.

Tabel 4.6: Hasil Pengujian Keterdeteksian Model Kedua

Jumlah Keberhasilan:	333
Jumlah Ketidakberhasilan:	147
Persentasi Keberhasilan:	69,375%
Persentasi Ketidakberhasilan:	30,625%

4.3 Pengujian Performa Model dengan Variasi Jarak Kamera

Pada skenario pengujian ini, digunakan variasi jarak dari citra wajah ke *webcam* yang digunakan sebagai variabel bebasnya. Digunakan jarak 50 sentimeter, 75 sentimeter, dan 100 sentimeter untuk tiap-tiap pengujian. Perbandingan gambar dataset dapat dilihat pada Tabel 4.7. Variabel terikat yang akan didapatkan dari pengujian ini adalah berupa nilai dari *accuracy*, *loss*, dan *confusion matrix*. Pada setiap dataset akan dilakukan *training* sebanyak 15 *epoch* dengan ukuran citra wajah yang telah disamaratakan sebesar 140 x 140. Hasil yang didapatkan dari pengujian performa model dengan variasi jarak kamera yang telah dilakukan menunjukkan bahwa model yang memiliki akurasi pendekripsi yang paling baik adalah model yang diambil dengan jarak antara wajah dan kamera sejauh 50 sentimeter. Untuk pembahasan lebih detailnya mengenai penelitian ini dapat dilihat dari subbab masing-masing jarak dibawah ini.

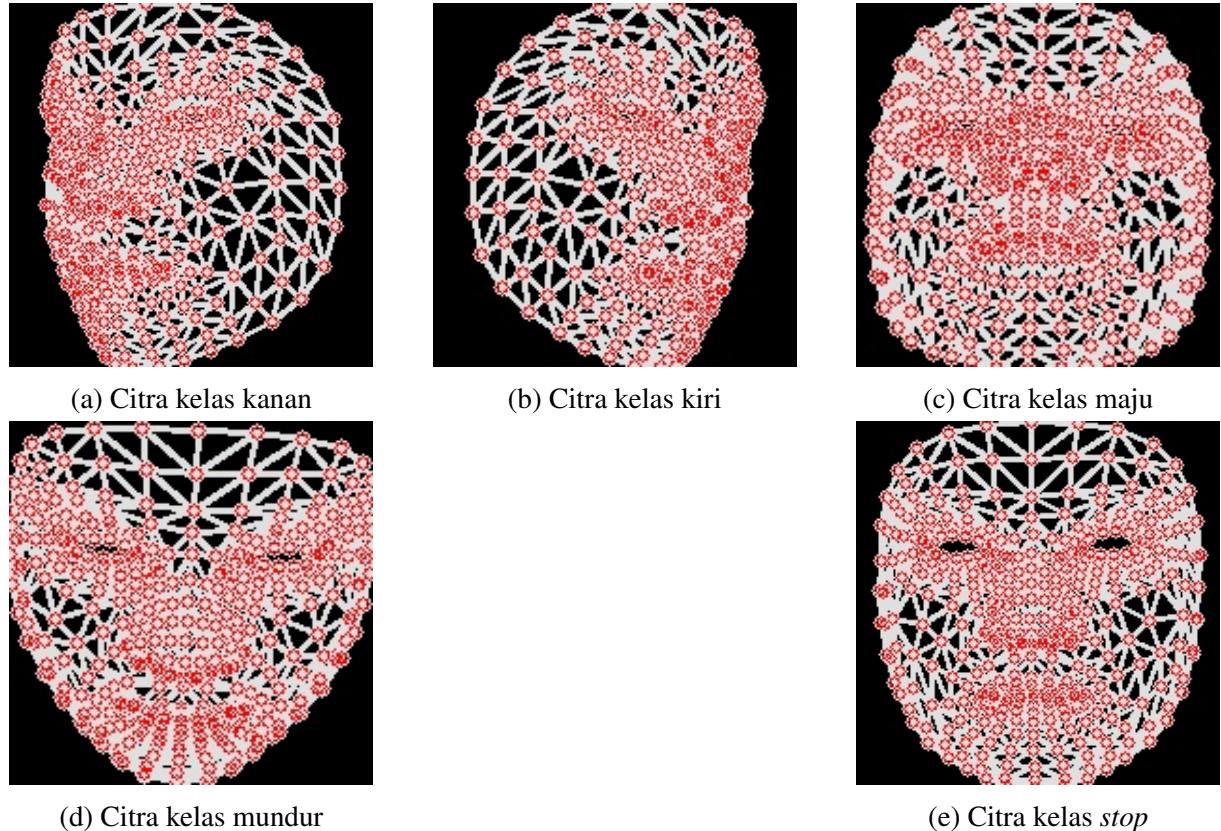
Jarak Kamera	Contoh Dataset
50	
75	
100	

Tabel 4.7: Variasi Jarak Kamera

4.3.1 Pengujian jarak 50 sentimeter

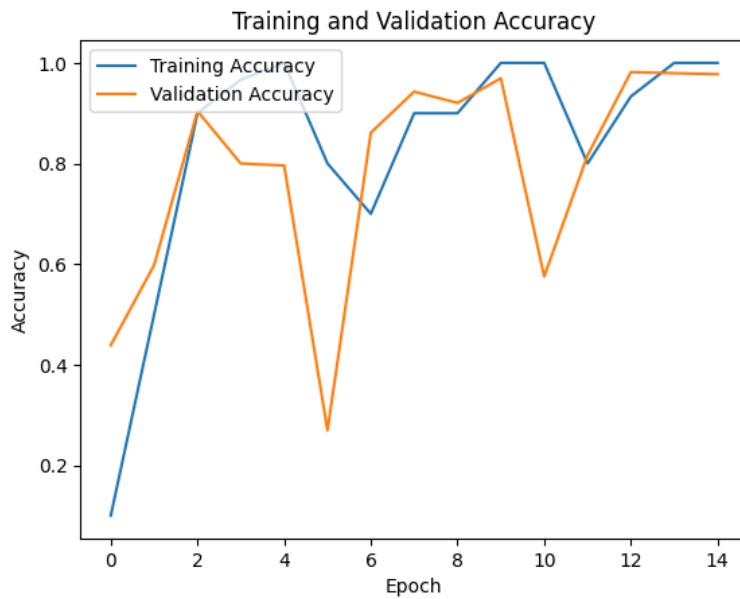
Pada bagian pengujian ini, citra wajah yang diambil sebagai dataset diambil dari jarak 50 sentimeter dari *webcam* yang digunakan. Diambil sebanyak 490 data citra wajah untuk setiap kelasnya yang berarti terdapat total 2450 data citra dari kelima kelas yang ada. Kemudian,

dataset tersebut dibagi menjadi dataset *training* dan dataset *validation* dengan rasio perbandingan 80 : 20 sehingga didapatkan 392 data citra wajah yang dijadikan sebagai dataset *training* dan 98 data citra wajah sebagai dataset *validation* untuk setiap kelasnya.

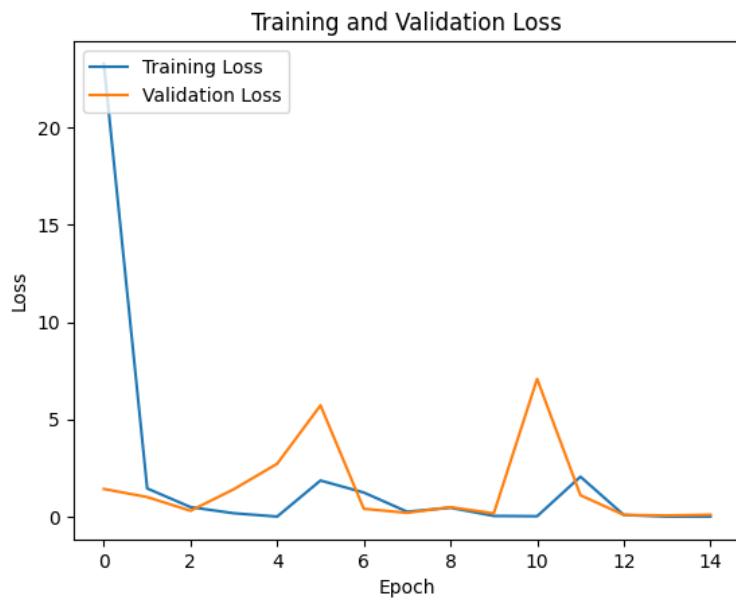


Gambar 4.8: Dataset Jarak 50 sentimeter

Berdasarkan grafik yang didapat setelah dilakukan *training*, model yang diambil dari jarak 50 sentimeter ini memiliki *training accuracy* sebesar 100% dan *validation accuracy* sebesar 97.76% seperti pada Gambar 4.9. Hal ini menunjukkan bahwa prediksi terhadap model pada dataset training semuanya akurat dan pada dataset validation juga memiliki akurasi yang tinggi meskipun tidak semuanya akurat. Model ini juga memiliki *training loss* sebesar 0.0001 dan *validation loss* sebesar 0.0960 yang dapat dilihat pada Gambar 4.10. *Training loss* dan *validation loss* yang didapat sangatlah kecil yang berarti error yang didapat sangatlah kecil yang berarti pendekstrian model sangatlah efisien. Selain itu, dapat dilihat juga pada grafik bahwa *training* yang dilakukan belum selesai di 15 *epoch*.



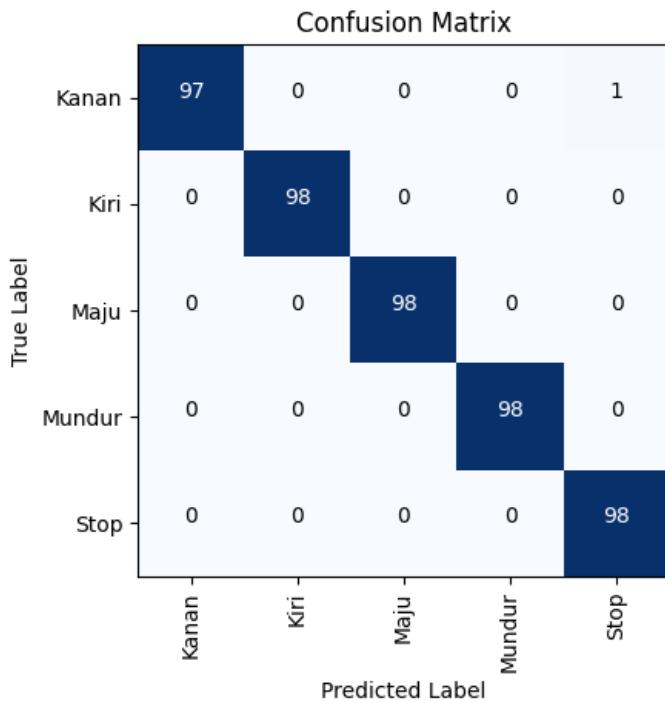
Gambar 4.9: Accuracy Hasil Pelatihan Jarak 50cm



Gambar 4.10: Loss Hasil Pelatihan Jarak 50cm

Dalam pengujian model CNN yang telah dibuat terhadap data validasi dan diambil dengan jarak 50 sentimeter, dengan menggunakan perhitungan *confusion matrix*, didapatkan hasil klasifikasi seperti yang telah divisualisasikan pada Gambar 4.11. Dari Gambar 4.11, dapat dilihat bahwa dari 490 data pada model yang diujikan terhadap data citra validasi, terdapat 11 data citra yang hasil prediksinya tidak sesuai dengan data aktual yang diujikan. Sebelas data citra tersebut merupakan data citra dari kelas kiri yang 4 data terdeteksi sebagai data citra kelas kanan dan 7 data citra terdeteksi sebagai kelas *stop*. Persentase keberhasilan pada kelas kiri adalah sekitar 88,78% dengan jumlah 87 data dan persentase ketidakberhasilannya adalah sekitar 11,22% dengan jumlah 11 data. Dengan detail persentase ketidakberhasilan 7,14% terdeteksi sebagai

kelas *stop* dan 4,08% terdeteksi sebagai citra kelas kanan. Pada Tabel 4.8 dapat dilihat terdapat 1 data kelas kanan yang terdeteksi sebagai *false negative* dan 97 data yang terdeteksi sebagai *true positive*. Selain itu, dapat dilihat juga terdapat 1 data kelas *stop* yang terdeteksi sebagai *false positive* dan 391 data yang terdeteksi sebagai *true negative*.



Gambar 4.11: Confusion Matrix Hasil Pelatihan Jarak 50cm

Tabel 4.8: Hasil Klasifikasi Model Jarak 50cm

	TP	TN	FP	FN
Kanan	97	392	0	1
Kiri	98	392	0	0
Maju	98	392	0	0
Mundur	98	392	0	0
Stop	98	391	1	0

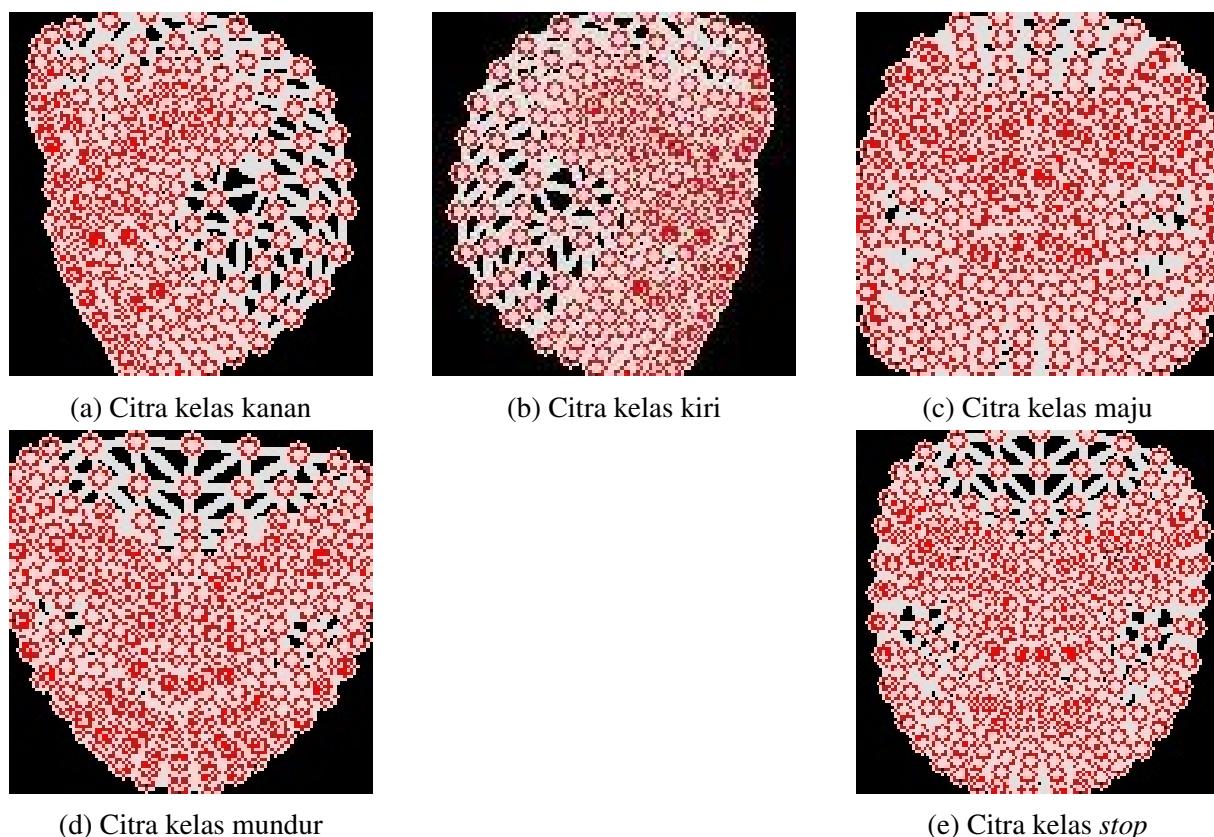
Pada Tabel 4.9 dapat dilihat bahwa nilai *accuracy* pada kelas kanan adalah sebesar 0.9898 dan nilai *recallnya* 0.9898. Hal ini menyebabkan kelas kanan memiliki *F1-Score* sebesar 0.9949. Sedangkan pada kelas *stop*, nilai *accuracynya* adalah 0.998 dan nilai *precisionnya* 0.9899. Hal ini menyebabkan kelas kiri memiliki *F1-Score* sebesar 0.9949. Pada kelas kiri, maju, dan mundur memiliki nilai *accuracy*, *precision*, *recall*, dan *F1-Score* sebesar 1.0000.

Tabel 4.9: Hasil Validasi Nilai Model Jarak 50cm

	Accuracy	Precision	Recall	F1-Score
Kanan	0.998	1.0000	0.9898	0.9949
Kiri	1.000	1.0000	1.0000	1.0000
Maju	1.000	1.0000	1.0000	1.0000
Mundur	1.000	1.0000	1.0000	1.0000
Stop	0.998	0.9899	1.0000	0.9949

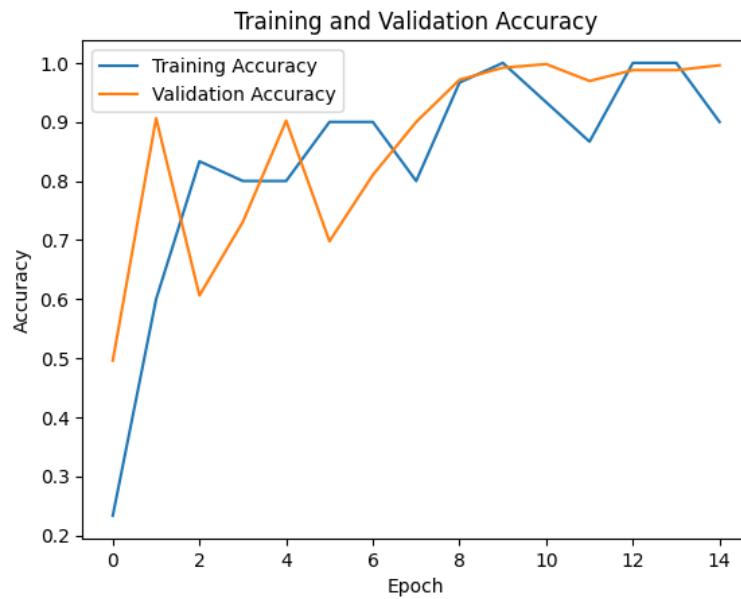
4.3.2 Pengujian jarak 75 sentimeter

Pada bagian pengujian ini, citra wajah yang diambil sebagai dataset diambil dari jarak 75 sentimeter dari *webcam* yang digunakan. Diambil sebanyak 490 data citra wajah untuk setiap kelasnya yang berarti terdapat total 2450 data citra dari kelima kelas yang ada. Kemudian, dataset tersebut dibagi menjadi dataset *training* dan dataset *validation* dengan rasio perbandingan 80 : 20 sehingga didapatkan 392 data citra wajah yang dijadikan sebagai dataset *training* dan 98 data citra wajah sebagai dataset *validation* untuk setiap kelasnya.

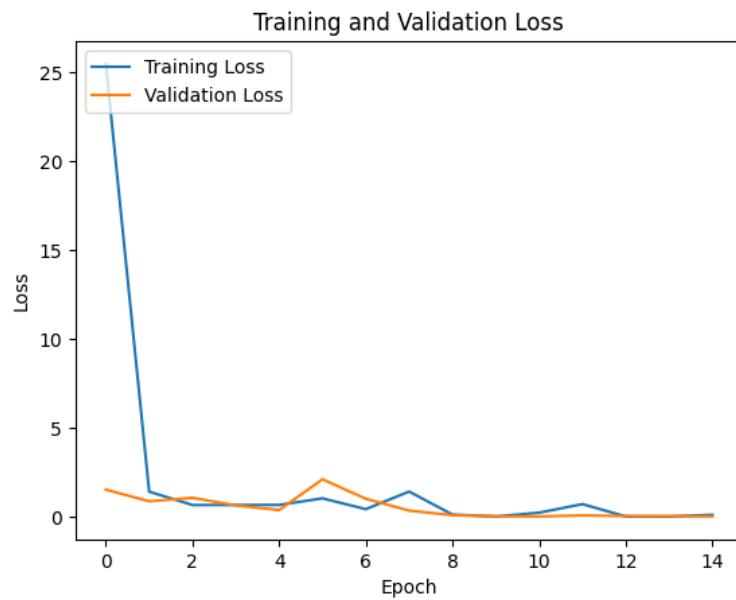


Gambar 4.12: Dataset Jarak 75 sentimeter

Berdasarkan grafik yang didapat setelah dilakukan *training*, model yang diambil dari jarak 75 sentimeter ini memiliki *training accuracy* sebesar 90% dan *validation accuracy* sebesar 99.59% seperti pada Gambar 4.13. Model ini juga memiliki *training loss* sebesar 0.0994 dan *validation loss* sebesar 0.0103 yang dapat dilihat pada Gambar 4.14. Selain itu, dapat dilihat juga pada grafik bahwa *training* yang dilakukan belum selesai di 15 epoch.



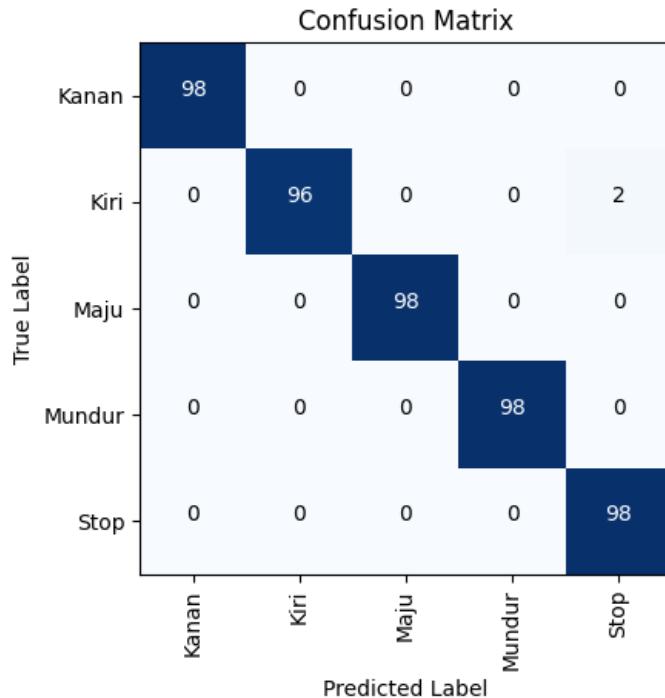
Gambar 4.13: Accuracy Hasil Pelatihan Jarak 75cm



Gambar 4.14: Loss Hasil Pelatihan Jarak 75cm

Dalam pengujian model CNN yang telah dibuat terhadap data validasi dan diambil dengan jarak 75 sentimeter, dengan menggunakan perhitungan *confusion matrix*, didapatkan hasil klasifikasi seperti yang telah divisualisasikan pada Gambar 4.15. Dari Gambar 4.15, dapat dilihat bahwa dari 490 data pada model yang diujikan terhadap data citra validasi, terdapat 2 data citra yang hasil prediksinya tidak sesuai dengan data aktual yang diujikan. Dua data citra tersebut merupakan data citra dari kelas kiri yang terdeteksi sebagai data citra dari kelas *stop*. Persentase keberhasilan pada kelas kiri adalah sekitar 97,96% dengan jumlah 96 data dan persentase ketidakberhasilannya adalah sekitar 2,04% dengan jumlah 2 data. Pada Tabel 4.10 dapat dilihat terdapat 2 data kelas kiri yang terdeteksi sebagai *false negative* dan 96 data yang terdeteksi

sebagai *true positive*. Selain itu, dapat dilihat juga terdapat 2 data kelas *stop* yang terdeteksi sebagai *false positive* dan 390 data yang terdeteksi sebagai *true negative*.



Gambar 4.15: Confusion Matrix Hasil Pelatihan Jarak 75cm

Tabel 4.10: Hasil Klasifikasi Model Jarak 75cm

	TP	TN	FP	FN
Kanan	98	392	0	0
Kiri	96	392	0	2
Maju	98	392	0	0
Mundur	98	392	0	0
Stop	98	390	2	0

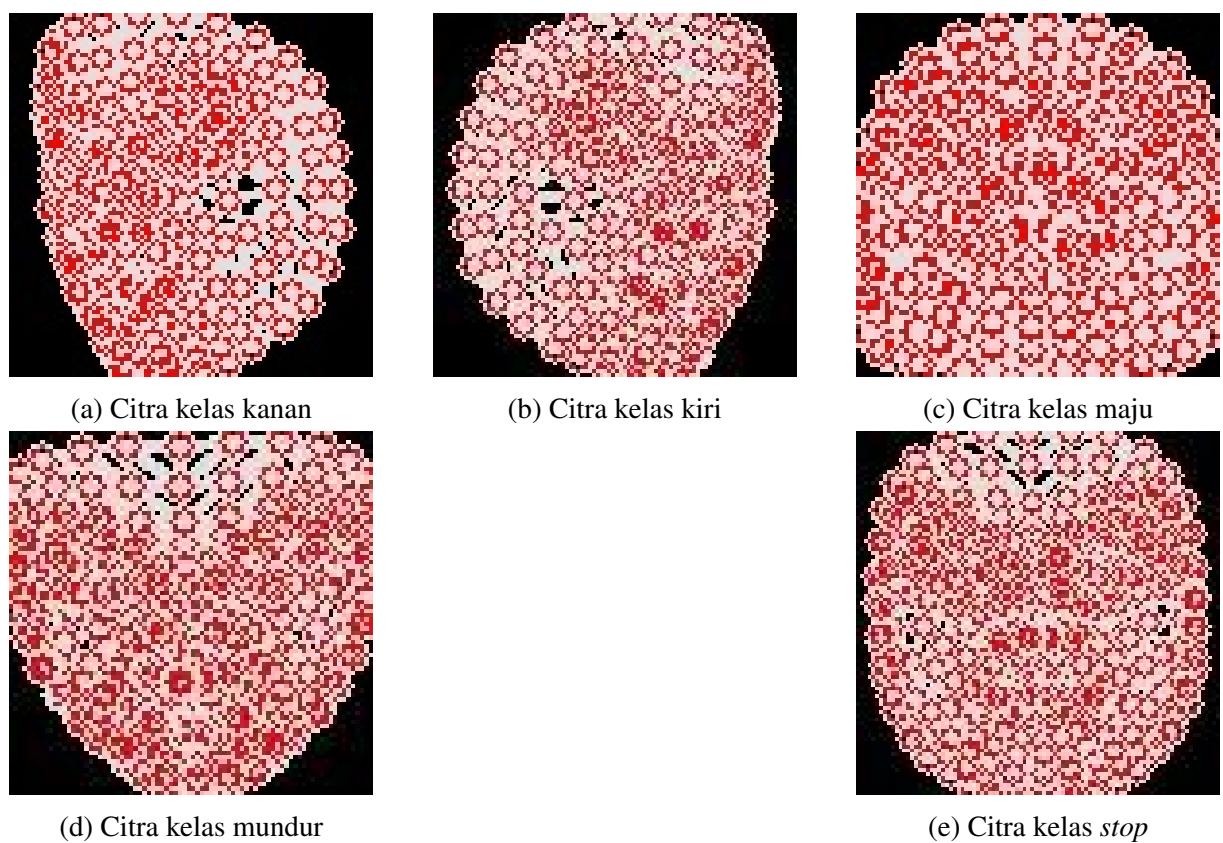
Pada Tabel 4.11 dapat dilihat bahwa nilai *accuracy* pada kelas kiri adalah sebesar 0.9959 dan nilai *recallnya* 0.9796. Hal ini menyebabkan kelas kiri memiliki *F1-Score* sebesar 0.9897. Sedangkan pada kelas *stop*, nilai *accuracynya* adalah 0.9959 dan nilai *precisionnya* 0.9800. Hal ini menyebabkan kelas *stop* memiliki *F1-Score* sebesar 0.9899. Pada kelas kanan, maju, dan mundur memiliki nilai *accuracy*, *precision*, *recall*, dan *F1-Score* sebesar 1.0000.

Tabel 4.11: Hasil Validasi Nilai Model Jarak 75cm

	Accuracy	Precision	Recall	F1-Score
Kanan	1.0000	1.0000	1.0000	1.0000
Kiri	0.9959	1.0000	0.9796	0.9897
Maju	1.0000	1.0000	1.0000	1.0000
Mundur	1.0000	1.0000	1.0000	1.0000
Stop	0.9959	0.9800	1.0000	0.9899

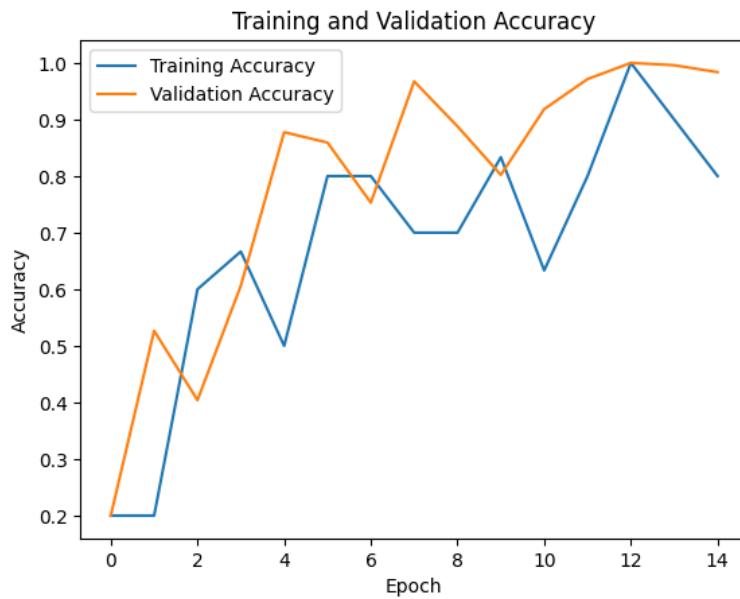
4.3.3 Pengujian jarak 100 sentimeter

Pada bagian pengujian ini, citra wajah yang diambil sebagai dataset diambil dari jarak 100 sentimeter dari *webcam* yang digunakan. Diambil sebanyak 490 data citra wajah untuk setiap kelasnya yang berarti terdapat total 2450 data citra dari kelima kelas yang ada. Kemudian, dataset tersebut dibagi menjadi dataset *training* dan dataset *validation* dengan rasio perbandingan 80 : 20 sehingga didapatkan 392 data citra wajah yang dijadikan sebagai dataset *training* dan 98 data citra wajah sebagai dataset *validation* untuk setiap kelasnya.

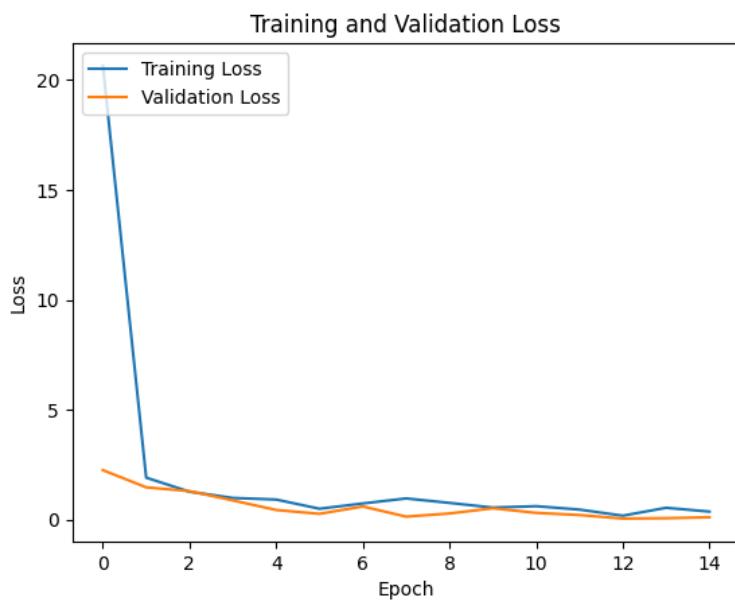


Gambar 4.16: Dataset Jarak 100 sentimeter

Berdasarkan grafik yang didapat setelah dilakukan *training*, model yang diambil dari jarak 100 sentimeter ini memiliki *training accuracy* sebesar 80% dan *validation accuracy* sebesar 98.37% seperti pada Gambar 4.17. Model ini juga memiliki *training loss* sebesar 0.3637 dan *validation loss* sebesar 0.1045 yang dapat dilihat pada Gambar 4.18. Selain itu, dapat dilihat juga pada grafik bahwa *training* yang dilakukan belum selesai di 15 *epoch*.



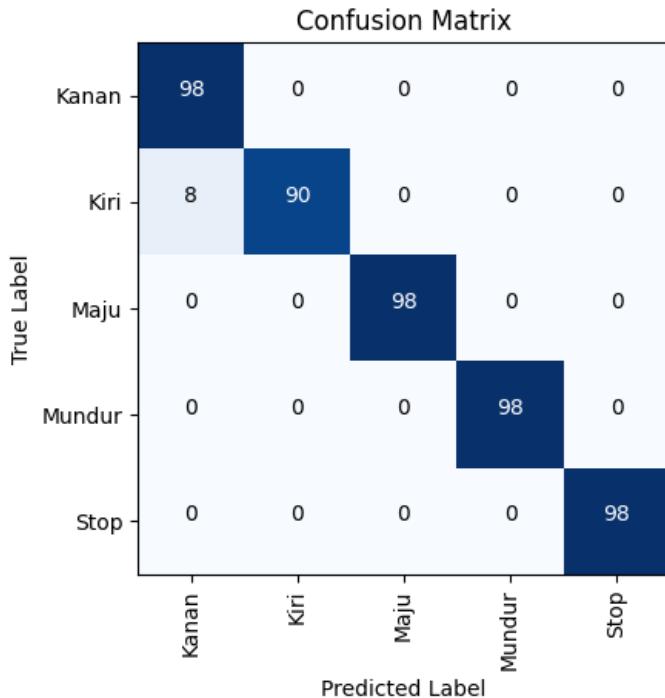
Gambar 4.17: Accuracy Hasil Pelatihan Jarak 100cm



Gambar 4.18: Loss Hasil Pelatihan Jarak 100cm

Dalam pengujian model CNN yang telah dibuat terhadap data validasi dan diambil dengan jarak 100 sentimeter, dengan menggunakan perhitungan *confusion matrix*, didapatkan hasil klasifikasi seperti yang telah divisualisasikan pada Gambar 4.19. Dari Gambar 4.19, dapat dilihat bahwa dari 490 data pada model yang diujikan terhadap data citra validasi, terdapat 8 data citra yang hasil prediksinya tidak sesuai dengan data aktual yang diujikan. Delapan data citra tersebut merupakan data citra dari kelas kiri yang terdeteksi sebagai data citra dari kelas kanan. Persentase keberhasilan pada kelas kiri adalah sekitar 91,84% dengan jumlah 90 data dan persentase ketidakberhasilannya adalah sekitar 8,16% dengan jumlah 8 data. Pada Tabel 4.13 dapat dilihat terdapat 8 data kelas kanan yang terdeteksi sebagai *false positive* dan 384

data yang terdeteksi sebagai *true negative*. Selain itu, dapat dilihat juga terdapat 8 data kelas kiri yang terdeteksi sebagai *false negative* dan 90 data yang terdeteksi sebagai *true positive*.



Gambar 4.19: Confusion Matrix Hasil Pelatihan Jarak 100cm

Tabel 4.12: Hasil Klasifikasi Model Jarak 100cm

	TP	TN	FP	FN
Kanan	98	384	8	0
Kiri	90	392	0	8
Maju	98	392	0	0
Mundur	98	392	0	0
Stop	98	392	0	0

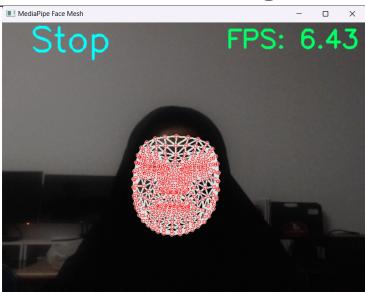
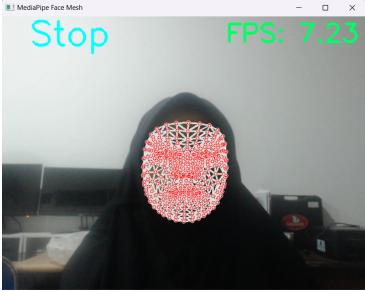
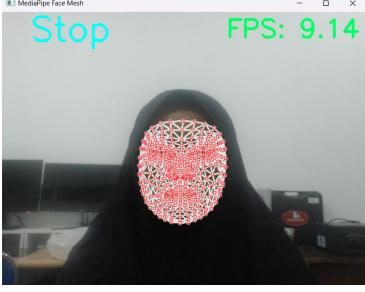
Pada Tabel 4.13 dapat dilihat bahwa nilai *accuracy* pada kelas kanan adalah sebesar 0.9837 dan nilai *precisionnya* 0.9245. Hal ini menyebabkan kelas kanan memiliki *F1-Score* sebesar 0.9608. Sedangkan pada kelas kiri, nilai *accuracynya* adalah 0.9837 dan nilai *recallnya* 0.9184. Hal ini menyebabkan kelas kiri memiliki *F1-Score* sebesar 0.9574. Pada kelas maju, mundur, dan stop memiliki nilai *accuracy*, *precision*, *recall*, dan *F1-Score* sebesar 1.0000.

Tabel 4.13: Hasil Validasi Nilai Model Jarak 100cm

	Accuracy	Precision	Recall	F1-Score
Kanan	0.9837	0.9245	1.0000	0.9608
Kiri	0.9837	1.0000	0.9184	0.9574
Maju	1.0000	1.0000	1.0000	1.0000
Mundur	1.0000	1.0000	1.0000	1.0000
Stop	1.0000	1.0000	1.0000	1.0000

4.4 Pengujian performa model dengan variasi pencahayaan

Pada bagian pengujian ini, penulis telah melakukan serangkaian uji coba untuk menggali pengaruh tingkat intensitas cahaya terhadap proses pendekripsi citra wajah secara real-time. Dalam hal ini, digunakan tiga level intensitas cahaya yang akan menjadi variabel bebas, yaitu 15 lux, 46 lux, dan 110 lux. Perbandingan keadaan ruangan secara *real-time* dapat dilihat pada Tabel 4.14. Pengukuran nilai lux yang didapatkan dilakukan dengan menggunakan bantuan dari lightmeter yang terdapat di *handphone* penulis untuk meningkatkan ketelitian pengukuran. Ketiga level intensitas Cahaya ini merepresentasikan tiga kondisi yang terdapat pada suatu ruangan. Dari dilakukannya pengujian dengan variasi intensitas cahaya ini diharapkan didapatkannya data mengenai adaptabilitas model pada kondisi pencahayaan yang berbeda-beda. Pada setiap pengujian akan dilakukan 460 kali pendekripsi yang mana kemudian akan dihitung perbandingan terdeteksi dan tidaknya pada intensitas cahaya yang sedang diujikan. Hasil yang didapatkan dari pengujian performa model dengan variasi pencahayaan yang telah dilakukan menunjukkan bahwa akurasi pendekripsi model yang paling tinggi adalah dalam keadaan ruangan dengan intensitas cahaya sebesar 110 lux yaitu keadaan pada saat malam hari dan semua lampu pada ruangan dinyalakan. Untuk pembahasan lebih detailnya mengenai penelitian ini dapat dilihat dari subbab masing-masing intensitas pencahayaan dibawah ini.

Intensitas Cahaya	Keadaan Ruangan
15	
46	
110	

Tabel 4.14: Variasi Intensitas Cahaya

4.4.1 Pengujian intensitas cahaya 15 lux

Pengujian pertama dilakukan dengan menguji intensitas cahaya sebesar 15 lux yang merepresentasikan kondisi dalam ruangan dengan lampu mati pada malam hari kemudian dinyalakan sebuah senter yang dihadapkan kearah langit-langit ruangan. Gambar 4.20 menunjukkan keadaan pada saat cahaya dengan intensitas 15 lux ketika dilakukannya pengujian ini.



Gambar 4.20: Pendektsian pada 15 lux

Hasil yang didapatkan dari proses pengujian ini menunjukkan tingkat akurasi yang cukup baik, yaitu sekitar 95,87% dari 460 kali pendektsian yang dilakukan. Hal ini menunjukkan bahwa dari 460 kali pendektsian yang dilakukan, 441 kali diantaranya berhasil terdeteksi dan 19 lainnya tidak berhasil terdeteksi. Informasi mengenai performa ini telah di representasikan pada Tabel 4.15. Hasil yang diperoleh dari pengujian ini menunjukkan bahwa meskipun kondisi cahaya dan penerangannya sangat minim di sekitar 15 lux, kamera masih dapat mendekksi citra wajah dan model masih mampu memproses landmark-landmark pada citra wajah dan kemudian melakukan klasifikasi untuk pose wajah yang sedang didekksi dengan akurasi yang cukup baik. Hal ini memperlihatkan kemampuan model untuk beradaptasi terhadap intensitas pencahayaan suatu ruangan sehingga model tidak bergantung pada tingkat pencahayaan tertentu.

Tabel 4.15: Hasil Variasi 15 Lux

Jumlah Keberhasilan:	441
Jumlah Ketidakberhasilan:	19
Persentasi Keberhasilan:	95,87%
Persentasi Ketidakberhasilan:	4,13%

4.4.2 Pengujian intensitas cahaya 46 lux

Pengujian kedua dilakukan dengan menguji tingkat intensitas cahaya sebesar 46 lux yang merepresentasikan kondisi dalam ruangan pada malam hari yang setengah lampunya dinyalakan dan setengah lainnya dimatikan. Gambar 4.21 menunjukkan keadaan pada saat cahaya dengan intensitas 46 lux ketika dilakukannya pengujian ini.



Gambar 4.21: Pendekripsi pada 46 lux

Hasil yang didapatkan dari proses pengujian ini menunjukkan tingkat akurasi yang sangat baik, yaitu sekitar 95,87% dari 460 kali pendekripsi yang dilakukan. Hal ini menunjukkan bahwa dari 460 kali pendekripsi yang dilakukan, 458 kali diantaranya berhasil terdeteksi dan 2 lainnya tidak berhasil terdeteksi. Informasi mengenai performa ini telah di representasikan pada Tabel 4.16. Hasil yang diperoleh dari pengujian ini menunjukkan bahwa ketika kondisi cahaya dan penerangannya remang-remang atau sekitar 46 lux, kamera dapat mendekripsi citra wajah dan model mampu memproses landmark-landmark pada citra wajah dan kemudian melakukan klasifikasi untuk pose wajah yang sedang dideteksi dengan akurasi yang sangat baik. Hal ini memperlihatkan kemampuan model untuk beradaptasi terhadap intensitas pencahayaan suatu ruangan sehingga model tidak terikat pada tingkat pencahayaan tertentu.

Tabel 4.16: Hasil Variasi 46 Lux

Jumlah Keberhasilan:	458
Jumlah Ketidakberhasilan:	2
Persentasi Keberhasilan:	99,57%
Persentasi Ketidakberhasilan:	0,43%

4.4.3 Pengujian intensitas cahaya 110 lux

Pengujian ketiga dilakukan dengan menguji tingkat intensitas cahaya sebesar 110 lux, merepresentasikan kondisi dalam ruangan pada malam hari yang semua lampunya dinyalakan. Gambar 4.22 menunjukkan keadaan pada saat cahaya dengan intensitas 46 lux ketika dilakukannya pengujian ini.



Gambar 4.22: Pendekripsi pada 110 lux

Hasil yang didapatkan dari proses pengujian ini menunjukkan tingkat akurasi yang sangat baik, yaitu sekitar 99,78% dari 460 kali pendekripsi yang dilakukan. Hal ini menunjukkan bahwa dari 460 kali pendekripsi yang dilakukan, 459 kali diantaranya berhasil terdeteksi dan 1 lainnya tidak berhasil terdeteksi. Informasi mengenai performa ini telah di representasikan pada Tabel ???. Hasil yang diperoleh dari pengujian ini menunjukkan bahwa ketika kondisi cahaya dan penerangannya terang atau sekitar 110 lux, kamera dapat dengan mudah mendekripsi citra wajah sehingga model mampu memproses landmark-landmark pada citra wajah dengan sangat baik. Proses klasifikasi untuk pose wajah yang sedang didekripsi juga memiliki tingkat akurasi yang sangat tinggi. Hal ini menunjukkan bahwa model dapat bekerja dengan sangat baik pada ruangan yang terang.

Tabel 4.17: Hasil Variasi 110 Lux

Jumlah Keberhasilan:	459
Jumlah Ketidakberhasilan:	1
Persentasi Keberhasilan:	99,78%
Persentasi Ketidakberhasilan:	0,22%

4.5 Pengujian Kecepatan FPS

Dalam skenario pengujian ini, akan dilakukan dua kali pengujian. Pengujian pertama akan dilakukan pada laptop penulis dan pengujian kedua akan dilakukan pada NUC yang mana hal ini menjadikan laptop dan NUC menjadi variabel bebas dari skenario pengujian kali ini. Pada pengujian ini kecepatan FPS yang akan dijadikan variabel terikatnya. Skenario pengujian ini akan dilakukan selama 1 menit untuk setiap pengujinya.

4.5.1 Pengujian Kecepatan FPS Pada Laptop

Pada pengujian ini, dilakukan pengambilan data FPS selama 60 detik dengan menggunakan laptop penulis. Didapatkan nilai rata-rata FPS pada laptop sebesar 7.64566667. Pengambilan data pengujian dilakukan dengan menggunakan perangkat dengan spesifikasi yang telah dijabarkan pada Tabel 4.18.

Selama dilakukannya pengujian pada FPS laptop, didapatkan FPS paling tinggi sebesar 12.13 dan FPS paling rendahnya sebesar 3.25. Data lengkap yang didapatkan dari pengujian ini dapat dilihat pada Tabel 4.19.

Tabel 4.18: Spesifikasi Laptop

Komponen	Tipe
CPU	Intel(R) Core(TM) Ultra 7 155H
RAM	16 GB DDR5-5600
OS	Windows 11 Home Single Language

Tabel 4.19: Hasil Pengujian FPS Laptop

No	FPS	No	FPS
1	9.38	31	7.23
2	7.18	32	3.53
3	6.13	33	8.1
4	8.72	34	8.1
5	9.4	35	5.84
6	9.36	36	5.6
7	7.64	37	7.23
8	6.81	38	9.4
9	7.07	39	7.26
10	9.4	40	9.37
11	8.66	41	5.61
12	3.25	42	8.18
13	8.13	43	5.56
14	9.4	44	9.4
15	10.69	45	6.41
16	6.82	46	5.77
17	8.55	47	6.15
18	7.97	48	4.94
19	9.92	49	7.62
20	9.34	50	8.08
21	8.14	51	9.26
22	6.8	52	6.67
23	7.91	53	10.03
24	3.95	54	8.72
25	4.74	55	7.55
26	8.53	56	8.72
27	6.39	57	5.89
28	8.74	58	9.4
29	8.17	59	7.67
30	12.13	60	6.13

4.5.2 Pengujian Kecepatan FPS Pada NUC

Pada pengujian ini, dilakukan pengambilan data FPS selama 60 detik dengan menggunakan NUC. Didapatkan nilai rata-rata FPS pada NUC sebesar 6.314. Pengambilan data pengujian dilakukan dengan menggunakan perangkat dengan spesifikasi yang telah dijabarkan pada Tabel 4.20. Hal ini menunjukkan bahwa laptop penulis memiliki kekuatan pemrosesan yang lebih baik daripada NUC yang digunakan pada penelitian kali ini. Meskipun perbedaan rata-rata FPS antara laptop penulis dan NUC yang digunakan tidak jauh, hanya sekitar 1.331667.

Selama dilakukannya pengujian pada FPS NUC, didapatkan FPS paling tinggi sebesar 7.79 dan FPS paling rendahnya sebesar 2.24. Data lengkap yang didapatkan dari pengujian ini dapat dilihat pada Tabel 4.21.

Tabel 4.20: Spesifikasi NUC

Komponen	Tipe
CPU	Intel® Core™ i7-1165G7
RAM	32 GB LPDDR4
OS	Windows 11 Home Single Language

Tabel 4.21: Hasil Pengujian FPS NUC

No	FPS	No	FPS
1	6.67	31	5.9
2	5.56	32	5.18
3	4.99	33	6.07
4	7.79	34	5.93
5	6.06	35	7.31
6	5.41	36	7.41
7	6.49	37	6.53
8	6.6	38	7.28
9	6.42	39	2.24
10	7.23	40	6.96
11	5.45	41	6.65
12	6.75	42	7.29
13	6.6	43	6.57
14	6.6	44	7.32
15	5.64	45	6.92
16	5.84	46	4.93
17	5.26	47	7.57
18	6.59	48	6.46
19	5.88	49	5.55
20	7.05	50	7.65
21	7.35	51	5.53
22	6.91	52	6.57
23	7.43	53	2.76
24	3.31	54	6.68
25	7.81	55	6.86
26	6.66	56	6.72
27	5.35	57	7.27
28	6.77	58	4.73
29	7.21	59	6.78
30	6.83	60	6.71

4.6 Pengujian Waktu Respons Kontrol Kursi Roda

Pada pengujian ini, dilakukan perhitungan lama waktu yang dibutuhkan untuk melakukan pendekripsi dengan model yang kemudian diklasifikasikan dan dikirim ke ESP32 hingga motor kursi roda mulai bergerak. Pengujian ini dilakukan secara real-time. Perhitungan waktu *delay* adalah dari timestamp mulai dikirimnya data hingga timestamp berhentinya motor kursi

roda, sedangkan perhitungan *inference time* dimulai dari dimulainya proses prediksi hingga didapatkan hasil dari proses klasifikasi. Rata-rata waktu *delay* yang didapatkan adalah 0,3025423729. Dari data hasil yang telah didapatkan dan dijabarkan pada Tabel 4.22, dapat dilihat bahwa waktu delay terlama adalah 0,784 dan waktu delay tersingkatnya adalah 0,008. Nilai *inference time* rata-rata yang didapatkan dari pengujian kali ini adalah sekitar 0,07220. Dapat dilihat juga dari Tabel 4.22 bahwa waktu terlama yang diperlukan untuk melakukan inferensi adalah 0.17112 dan waktu tersingkat yang diperlukan adalah 0.06040.

Dari Tabel 4.22 didapatkan juga rata-rata delay untuk tiap-tiap kelasnya. Rata-rata waktu delay yang didapatkan untuk kelas kanan adalah 0,1664285714 detik dari 7 kelas kanan yang diujikan. Rata-rata waktu delay yang didapatkan untuk kelas kiri adalah 0,2398888889 detik dari 9 kelas kiri yang diujikan. Rata-rata waktu delay yang didapatkan untuk kelas maju adalah 0,4099 detik dari 10 kelas maju yang diujikan. Rata-rata waktu delay yang didapatkan untuk kelas mundur adalah 0,195 detik dari 3 kelas mundur yang diujikan. Rata-rata waktu delay yang didapatkan untuk kelas *stop* adalah 0,3280666667 detik dari 30 kelas *stop* yang diujikan. Terdapat 30 data kelas *stop* dari 60 data yang ada karena setiap kali akan mengganti arah harus melakukan arah *stop* terlebih dahulu.

Selain itu, dari Tabel 4.22 didapatkan juga rata-rata *inference time* untuk tiap-tiap kelasnya. Rata-rata waktu *inference time* yang didapatkan untuk kelas kanan adalah 0,07392142857 detik dari 7 kelas kanan yang diujikan. Rata-rata waktu *inference time* yang didapatkan untuk kelas kiri adalah 0,071874444444 detik dari 9 kelas kiri yang diujikan. Rata-rata waktu *inference time* yang didapatkan untuk kelas maju adalah 0,080469 detik dari 10 kelas maju yang diujikan. Rata-rata waktu *inference time* yang didapatkan untuk kelas mundur adalah 0,07338333333 detik dari 3 kelas mundur yang diujikan. Rata-rata waktu *inference time* yang didapatkan untuk kelas *stop* adalah 0,069029 detik dari 30 kelas *stop* yang diujikan.

Tabel 4.22: Hasil Pengujian Respons Time

Inference Time	Timestamp Sent	Timestamp Received	Timestamp Motor	Delay	Arah
0.17112	08:11:00.908	08:11:00.966	08:11:01.335	0,427	Maju
0.06747	08:11:09.112	08:11:09.156	08:11:09.527	0,415	Stop
0.06656	08:11:11.341	08:11:11.394	08:11:11.576	0,235	Kiri
0.06704	08:11:14.759	08:11:14.819	08:11:15.005	0,246	Stop
0.06689	08:11:16.922	08:11:16.953	08:11:16.953	0,031	Kanan
0.06804	08:11:21.697	08:11:21.731	08:11:21.917	0,22	Stop
0.06740	08:11:26.121	08:11:26.150	08:11:26.336	0,215	Stop
0.06642	08:11:28.358	08:11:28.385	08:11:28.746	0,388	Maju
0.06187	08:11:31.240	08:11:31.260	08:11:31.634	0,394	Stop
0.06122	08:11:34.243	08:11:34.286	08:11:34.472	0,229	Kiri
0.06358	08:11:40.575	08:11:40.604	08:11:40.791	0,216	Stop
0.06837	08:11:43.433	08:11:43.476	08:11:43.649	0,216	Kanan
0.06090	08:11:48.368	08:11:48.466	08:11:48.653	0,285	Stop
0.06040	08:11:55.2	08:11:55.046	08:11:55.232	0,032	Stop
0.06618	08:11:56.595	08:11:56.618	08:11:56.989	0,394	Maju
0.05936	08:11:59.914	08:11:59.956	08:12:00.330	0,416	Stop

Dilanjutkan di halaman selanjutnya

Table 4.22 dilanjutkan dari halaman sebelumnya

Inference Time	Timestamp Sent	Timestamp Received	Timestamp Motor	Delay	Arah
0.06686	08:12:02.125	08:12:02.216	08:12:02.402	0,277	Kiri
0.06181	08:12:02.547	08:12:02.577	08:12:02.763	0,216	Stop
0.07173	08:12:02.725	08:12:02.763	08:12:02.948	0,223	Kiri
0.07078	08:12:08.189	08:12:08.230	08:12:08.415	0,226	Stop
0.06469	08:12:10.22	08:12:10.042	08:12:10.228	0,008	Kanan
0.06698	08:12:14.704	08:12:14.718	08:12:15.125	0,421	Stop
0.06677	08:12:20.406	08:12:20.445	08:12:20.819	0,413	Stop
0.06346	08:12:21.837	08:12:21.979	08:12:22.150	0,313	Mundur
0.06106	08:12:26.679	08:12:26.793	08:12:26.979	0,3	Stop
0.06709	08:12:28.366	08:12:28.376	08:12:28.607	0,241	Kiri
0.06255	08:12:32.613	08:12:32.655	08:12:32.887	0,274	Stop
0.06194	08:12:33.543	08:12:33.582	08:12:33.769	0,226	Kanan
0.06357	08:12:38.367	08:12:38.407	08:12:38.777	0,41	Stop
0.06157	08:12:41.19	08:12:41.044	08:12:41.414	0,224	Maju
0.06589	08:12:45.467	08:12:45.505	08:12:45.877	0,41	Stop
0.06070	08:12:47.102	08:12:47.124	08:12:47.496	0,394	Maju
0.06334	08:12:48.502	08:12:48.566	08:12:48.937	0,435	Stop
0.07318	08:12:48.707	08:12:49.124	08:12:49.491	0,784	Maju
0.07005	08:12:49.277	08:12:49.677	08:12:50.050	0,773	Stop
0.07054	08:12:52.961	08:12:53.001	08:12:53.370	0,409	Maju
0.06099	08:12:54.344	08:12:54.394	08:12:54.766	0,422	Stop
0.06225	08:12:55.906	08:12:55.972	08:12:56.340	0,434	Maju
0.07139	08:12:58.193	08:12:58.238	08:12:58.612	0,419	Stop
0.06290	08:12:59.794	08:12:59.817	08:13:00.003	0,209	Kanan
0.06111	08:12:59.980	08:13:00.283	08:13:00.470	0,49	Stop
0.07855	18:18:48.22	18:18:48.055	18:18:48.437	0,217	Maju
0.09415	18:18:50.515	18:18:50.564	18:18:50.911	0,396	Stop
0.11154	18:18:52.281	18:18:52.329	18:18:52.518	0,237	Kanan
0.07871	18:18:54.843	18:18:54.892	18:18:55.082	0,239	Stop
0.07861	18:18:56.227	18:18:56.242	18:18:56.467	0,24	Kiri
0.08311	18:18:59.190	18:18:59.270	18:18:59.429	0,239	Stop
0.07813	18:19:02.784	18:19:02.848	18:19:03.007	0,223	Kiri
0.07856	18:19:03.150	18:19:03.198	18:19:03.198	0,048	Mundur
0.07867	18:19:06.234	18:19:06.298	18:19:06.488	0,254	Stop
0.07813	18:19:07.282	18:19:07.331	18:19:07.506	0,224	Mundur
0.07813	18:19:12.992	18:19:13.042	18:19:13.217	0,225	Stop
0.09418	18:19:14.554	18:19:14.618	18:19:14.982	0,428	Maju
0.07862	18:19:17.455	18:19:17.503	18:19:17.852	0,397	Stop
0.08112	18:19:18.569	18:19:18.583	18:19:18.807	0,238	Kanan
0.07868	18:19:24.491	18:19:24.540	18:19:24.729	0,238	Stop
0.07854	18:19:25.452	18:19:25.500	18:19:25.690	0,238	Kiri
0.07845	18:19:26.249	18:19:26.281	18:19:26.455	0,206	Stop

Dilanjutkan di halaman selanjutnya

Table 4.22 dilanjutkan dari halaman sebelumnya

Inference Time	Timestamp Sent	Timestamp Received	Timestamp Motor	Delay	Arah
0.07813	18:19:27.823	18:19:27.871	18:19:28.076	0,253	Kiri

4.7 Pengujian kestabilan Performa Motor Kursi Roda

Pada skenario pengujian ini dilakukan dengan menggunakan video wajah yang menunjukkan tiap arah selama 2 detik. Hal ini dilakukan dengan tujuan mengecek kestabilan motor pada kursi roda. Setiap arah akan dilakukan pengulangan sebanyak 30 kali. Pengukuran yang dihitung adalah ketika motor pada kursi roda mulai bergerak hingga motor tersebut tepat berhenti. Hasil data yang diharapkan berupa rata-rata lama motor berjalan untuk keempat kelas yang akan diujikan.

4.7.1 Kestabilan Motor Kelas Kanan

Pengujian ketiga dilakukan dengan menguji kestabilan lama gerak motor untuk pose wajah menghadap keatas atau kelas maju. Hasil yang dari proses pengujian yang telah dilakukan berupa rata-rata lama gerak motor. Rata-rata yang didapatkan adalah selama 8.945 detik untuk setiap kali pose wajah menghadap ke kanan selama 2 detik. Untuk seluruh hasil data yang didapat melalui pengujian ini dapat dilihat pada Tabel 4.23. Pada tabel tersebut telah didokumentasikan dalam bentuk timestamp ketika motor pada kursi roda mulai bergerak ke kanan dan timestamp ketika motor kursi roda berhenti bergerak sepenuhnya.

Tabel 4.23: Hasil Pengujian Kestabilan Motor Kelas Kanan

Timestamp	Arah	Lama Motor Berjalan
12:24:01.612	Kanan	9,938
12:24:11.550	Stop	
12:24:12.998	Kanan	8,736
12:24:21.734	Stop	
12:24:23.050	Kanan	8,867
12:24:31.917	Stop	
12:24:33.334	Kanan	8,665
12:24:41.999	Stop	
12:24:43.333	Kanan	8,515
12:24:51.848	Stop	
12:26:18.634	Kanan	9,549
12:26:28.183	Stop	
12:26:29.583	Kanan	8,518
12:26:38.101	Stop	
12:26:39.500	Kanan	8,766
12:26:48.266	Stop	
12:26:49.734	Kanan	8,916
12:26:58.650	Stop	
12:26:59.949	Kanan	9
12:27:08.949	Stop	

Dilanjutkan di halaman selanjutnya

Table 4.23 dilanjutkan dari halaman sebelumnya

Timestamp	Arah	Lama Motor Berjalan
12:28:56.502	Kanan	9,367
12:29:05.869	Stop	
12:29:07.419	Kanan	8,665
12:29:16.084	Stop	
12:29:17.469	Kanan	8,698
12:29:26.167	Stop	
12:29:27.683	Kanan	8,717
12:29:36.400	Stop	
12:29:37.767	Kanan	8,667
12:29:46.434	Stop	
13:38:29.238	Kanan	9,635
13:38:38.873	Stop	
13:38:40.256	Kanan	9,084
13:38:49.340	Stop	
13:38:50.723	Kanan	9,785
13:39:00.508	Stop	
13:39:02.290	Kanan	9,998
13:39:12.288	Stop	
13:39:13.689	Kanan	8,952
13:39:22.641	Stop	
13:39:53.369	Kanan	10,604
13:40:03.973	Stop	
13:40:05.758	Kanan	8,999
13:40:14.757	Stop	
13:40:16.175	Kanan	11,133
13:40:27.308	Stop	
13:40:29.256	Kanan	9,918
13:40:39.174	Stop	
13:40:40.624	Kanan	10,215
13:40:50.839	Stop	
13:41:33.167	Kanan	9,726
13:41:42.893	Stop	
13:41:44.458	Kanan	8,884
13:41:53.342	Stop	
13:41:54.840	Kanan	8,886
13:42:03.726	Stop	
13:42:05.256	Kanan	9,252
13:42:14.508	Stop	
13:42:15.773	Kanan	8,683
13:42:24.456	Stop	

4.7.2 Kestabilan Motor Kelas Kiri

Pengujian ketiga dilakukan dengan menguji kestabilan lama gerak motor untuk pose wajah menghadap ke kiri atau kelas kiri. Hasil yang dari proses pengujian yang telah dilakukan berupa

rata-rata lama gerak motor. Rata-rata yang didapatkan adalah selama 8.934 detik untuk setiap kali pose wajah menghadap ke kiri selama 2 detik. Untuk seluruh hasil data yang didapat melalui pengujian ini dapat dilihat pada Tabel 4.24. Pada tabel tersebut telah didokumentasikan dalam bentuk timestamp ketika motor pada kursi roda mulai bergerak ke kiri dan timestamp ketika motor kursi roda berhenti bergerak sepenuhnya.

Tabel 4.24: Hasil Pengujian Kestabilan Motor Kelas Kiri

Timestamp	Arah	Lama Motor Berjalan
13:25:33.320	Kiri	8.716
13:25:42.036	Stop	
13:25:43.503	Kiri	9.165
13:25:52.668	Stop	
13:26:05.437	Kiri	9.100
13:26:14.537	Stop	
13:26:16.036	Kiri	9.166
13:26:25.202	Stop	
13:26:26.653	Kiri	9.233
13:26:35.886	Stop	
13:26:37.302	Kiri	8.967
13:26:46.269	Stop	
13:26:47.737	Kiri	8.700
13:26:56.437	Stop	
13:28:26.402	Kiri	8.906
13:28:35.308	Stop	
13:28:36.903	Kiri	9.100
13:28:46.003	Stop	
13:28:47.288	Kiri	8.913
13:28:56.201	Stop	
13:29:08.071	Kiri	8.796
13:29:16.867	Stop	
13:29:18.337	Kiri	9.166
13:29:27.503	Stop	
13:29:28.987	Kiri	8.865
13:29:37.852	Stop	
13:29:39.536	Kiri	9.252
13:29:48.788	Stop	
13:29:50.302	Kiri	9.085
13:29:59.387	Stop	
13:31:07.802	Kiri	8.450
13:31:16.252	Stop	
13:31:17.737	Kiri	8.967
13:31:26.704	Stop	
13:31:27.936	Kiri	8.917
13:31:36.853	Stop	
13:31:38.519	Kiri	8.802

Dilanjutkan di halaman selanjutnya

Table 4.24 dilanjutkan dari halaman sebelumnya

Timestamp	Arah	Lama Motor Berjalan
13:31:47.321	Stop	
13:31:48.785	Kiri	8.534
13:31:57.319	Stop	
13:32:09.069	Kiri	8.768
13:32:17.837	Stop	
13:32:19.238	Kiri	8.948
13:32:28.186	Stop	
13:33:02.398	Kiri	9.706
13:33:12.104	Stop	
13:33:13.520	Kiri	9.151
13:33:22.671	Stop	
13:33:24.187	Kiri	8.833
13:33:33.020	Stop	
13:33:34.369	Kiri	9.003
13:33:43.372	Stop	
13:33:45.020	Kiri	8.935
13:33:53.955	Stop	
13:34:05.653	Kiri	8.834
13:34:14.487	Stop	
13:34:15.903	Kiri	8.600
13:34:24.503	Stop	
13:34:25.838	Kiri	8.450
13:34:34.288	Stop	

4.7.3 Kestabilan Motor Kelas Maju

Pengujian ketiga dilakukan dengan menguji kestabilan lama gerak motor untuk pose wajah menghadap keatas atau kelas maju. Hasil yang dari proses pengujian yang telah dilakukan berupa rata-rata lama gerak motor. Rata-rata yang didapatkan adalah selama 8,916166667 detik untuk setiap kali pose wajah menghadap ke atas selama 2 detik. Untuk seluruh hasil data yang didapat melalui pengujian ini dapat dilihat pada Tabel 4.25. Pada tabel tersebut telah didokumentasikan dalam bentuk timestamp ketika motor pada kursi roda mulai bergerak maju dan timestamp ketika motor kursi roda berhenti bergerak sepenuhnya.

Tabel 4.25: Hasil Pengujian Kestabilan Motor Kelas Maju

Timestamp	Arah	Lama Motor Berjalan
15:16:25.804	Maju	9,202
15:16:35.006	Stop	
15:16:37.808	Maju	8,616
15:16:46.424	Stop	
15:16:48.930	Maju	8,629
15:16:57.559	Stop	
15:17:00.141	Maju	8,935

Dilanjutkan di halaman selanjutnya

Table 4.25 dilanjutkan dari halaman sebelumnya

Timestamp	Arah	Lama Motor Berjalan
15:17:09.076	Stop	
15:17:11.673	Maju	
15:17:20.489	Stop	8,816
15:17:23.155	Maju	
15:17:32.075	Stop	8,92
15:17:34.673	Maju	
15:17:43.474	Stop	8,801
15:17:46.290	Maju	
15:17:54.993	Stop	8,703
15:17:57.575	Maju	
15:18:06.375	Stop	8,8
15:18:08.922	Maju	
15:18:17.758	Stop	8,836
15:19:58.907	Maju	
15:20:08.429	Stop	9,522
15:20:11.275	Maju	
15:20:19.909	Stop	8,634
15:20:22.460	Maju	
15:20:31.581	Stop	9,121
15:20:34.142	Maju	
15:20:43.136	Stop	8,994
15:20:45.792	Maju	
15:20:54.759	Stop	8,967
15:20:57.513	Maju	
15:21:06.509	Stop	8,996
15:21:09.216	Maju	
15:21:17.931	Stop	8,715
15:21:20.693	Maju	
15:21:29.476	Stop	8,783
15:21:32.268	Maju	
15:21:41.043	Stop	8,775
15:21:43.655	Maju	
15:21:52.459	Stop	8,804
15:30:13.681	Maju	
15:30:23.361	Stop	9,68
15:30:26.278	Maju	
15:30:35.195	Stop	8,917
15:30:37.762	Maju	
15:30:46.511	Stop	8,749
15:30:49.163	Maju	
15:30:58.313	Stop	9,15
15:31:00.962	Maju	
15:31:09.813	Stop	8,851
15:31:12.428	Maju	9,05

Dilanjutkan di halaman selanjutnya

Table 4.25 dilanjutkan dari halaman sebelumnya

Timestamp	Arah	Lama Motor Berjalan
15:31:21.478	Stop	
15:31:24.145	Maju	8,885
15:31:33.030	Stop	
15:31:35.847	Maju	8,749
15:31:44.596	Stop	
15:31:47.196	Maju	9,1
15:31:56.296	Stop	
15:31:58.894	Maju	8,785
15:32:07.679	Stop	

4.7.4 Kestabilan Motor Kelas Mundur

Pengujian ketiga dilakukan dengan menguji kestabilan lama gerak motor untuk pose wajah menghadap kebawah atau kelas mundur. Hasil yang dari proses pengujian yang telah dilakukan berupa rata-rata lama gerak motor. Rata-rata yang didapatkan adalah selama 8,950433333 detik untuk setiap kali pose wajah menghadap ke bawah selama 2 detik. Untuk seluruh hasil data yang didapat melalui pengujian ini dapat dilihat pada Tabel 4.26. Pada tabel tersebut telah di-dokumentasikan dalam bentuk timestamp ketika motor pada kursi roda mulai bergerak mundur dan timestamp ketika motor kursi roda berhenti berpenuhnya.

Tabel 4.26: Hasil Pengujian Kestabilan Motor Kelas Mundur

Timestamp	Arah	Lama Motor Berjalan
12:32:14.148	Mundur	9,438
12:32:23.586	Stop	
12:32:27.101	Mundur	8,683
12:32:35.784	Stop	
12:32:39.268	Mundur	9,101
12:32:48.369	Stop	
12:32:51.802	Mundur	9,065
12:33:00.867	Stop	
12:33:04.100	Mundur	9,086
12:33:13.186	Stop	
12:33:16.617	Mundur	8,968
12:33:25.585	Stop	
12:33:28.971	Mundur	8,981
12:33:37.952	Stop	
12:33:41.034	Mundur	9,63
12:33:50.664	Stop	
12:33:54.434	Mundur	9,119
12:34:03.553	Stop	
12:34:07.371	Mundur	9,164
12:34:16.535	Stop	
12:35:50.911	Mundur	9,52

Dilanjutkan di halaman selanjutnya

Table 4.26 dilanjutkan dari halaman sebelumnya

Timestamp	Arah	Lama Motor Berjalan
12:36:00.431	Stop	
12:36:03.752	Mundur	8,766
12:36:12.518	Stop	
12:36:16.020	Mundur	8,817
12:36:24.837	Stop	
12:36:28.138	Mundur	8,865
12:36:37.003	Stop	
12:36:40.238	Mundur	8,899
12:36:49.137	Stop	
12:36:52.404	Mundur	8,782
12:37:01.186	Stop	
12:37:04.487	Mundur	8,851
12:37:13.338	Stop	
12:37:16.519	Mundur	8,71
12:37:25.229	Stop	
12:37:28.637	Mundur	8,782
12:37:37.419	Stop	
12:37:40.820	Mundur	8,768
12:37:49.588	Stop	
13:55:23.563	Mundur	9,03
13:55:32.593	Stop	
13:55:35.928	Mundur	8,799
13:55:44.727	Stop	
13:55:48.078	Mundur	8,467
13:55:56.545	Stop	
13:55:59.844	Mundur	8,719
13:56:08.563	Stop	
13:56:11.828	Mundur	8,932
13:56:20.760	Stop	
13:56:24.111	Mundur	8,815
13:56:32.926	Stop	
13:56:36.177	Mundur	9,401
13:56:45.578	Stop	
13:56:48.596	Mundur	8,725
13:56:57.321	Stop	
13:57:00.779	Mundur	8,631
13:57:09.410	Stop	
13:57:12.694	Mundur	8,999
13:57:21.693	Stop	

4.8 Pembahasan Hasil

Pada penelitian ini, akan dilakukan prediksi dengan model yang sudah *ditraining* yang kemudian hasil prediksi tersebut akan diklasifikasikan kedalam empat kelas. Output yang didapatkan ini selanjutnya akan dikirimkan ke ESP32 untuk menggerakan motor pada kursi roda.

Model yang digunakan memiliki akurasi pendektsian yang paling tinggi ketika jarak antara wajah dengan kamera sekitar 50 sentimeter. Model tersebut juga memiliki akurasi pendektsian yang paling baik pada keadaan ruangan dengan intensitas cahaya 110 lux. Kecepatan pemrosesan NUC yang digunakan lebih rendah dari laptop pengguna, hal ini dapat dilihat dari kecepatan FPS NUC yang lebih rendah 1.331667 daripada FPS laptop. Pada respons kontrol kursi roda terdapat *delay* dengan rata-rata 0,3025423729 detik dan rata-rata *inference time* sekitar 0,07220 detik. Untuk pergerakan motor kursi roda, rata-rata kestabilan lama gerak motor dari keempat kelas adalah 9.017241667 dengan detail rata-rata kelas kanan 8.945, kelas kiri 9.257366667, kelas maju 8.916166667, dan kelas mundur 8.950433333.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Pendekslsian dengan model memiliki akurasi yang paling tinggi ketika jarak antara wajah dengan kamera sejauh 50 sentimeter.
2. Pendekslsian dengan model memiliki akurasi yang paling baik ketika pendekslsian dilakukan pada keadaan intensitas cahaya 110 lux.
3. Laptop yang digunakan penulis memiliki kecepatan FPS yang lebih tinggi daripada NUC dengan selisih sekitar 1.331667
4. Rata-rata delay secara *real-time* adalah sekitar 0,3025423729 detik dengan rata-rata *inference time* yang juga diambil secara *real-time* adalah 0,07220 detik.
5. Rata-rata kestabilan lama gerak motor dari keempat kelas adalah 8,9364 dengan detail rata-rata kelas kanan 8.945, kelas kiri 8.934, kelas maju 8.916166667, dan kelas mundur 8.950433333.

5.2 Saran

Untuk pengembangan lebih lanjut pada penelitian selanjutnya, dapat dilakukan beberapa hal berikut:

1. Menggunakan *Single Board Computer* yang memiliki kemampuan pemrosesan yang lebih kuat.
2. Menggunakan kamera yang dapat melakukan pendekslsian yang baik dalam keadaan intensitas cahaya seperti apapun, seperti kamera yang memiliki *built-in LED*.
3. Melakukan pengintegrasian alat yang lebih baik dengan merancang kursi roda secara khusus untuk digerakan dengan menggunakan *head gesture*
4. Membuat bracket kamera yang memang didesain secara khusus untuk disematkan pada kursi roda.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] A. H. Ropper, M. A. Samuels, and J. P. Klein, *Adams and Victor's: Principles of Neurology*, 10th. US: McGraw-Hill Education, 2014.
- [2] W. A. G. Ady, "Pengembangan desain kursi roda khususnya pada lansia berdasarkan citra (image) produk dengan metode kansei engineering," 2011.
- [3] S. Desai, S. Mantha, and V. Phalle, "Advances in smart wheelchair technology," in *2017 International Conference on Nascent Technologies in Engineering (ICNTE)*, 2017.
- [4] S. Nur Budiman, S. Lestanti, S. Marselius Evvandri, and R. Kartika Putri, "Finger gesture recognition to control volume on computer using opencv and mediapipe library," *Antivirus : Jurnal Ilmiah Teknik Informatika*, 2022.
- [5] M. H. Fatoni, E. A. Suprayitno, A. Arifin, N. F. Hikmah, T. A. Sardjono, and M. Nuh, "Pemanfaatan kursi roda elektrik dengan kendali joystick guna meningkatkan kemandirian siswa berkebutuhan khusus di sekolah luar biasa d yayasan pembinaan anak cacat surabaya," *Sewagati*, vol. 7, no. 2, pp. 167–175, Jan. 2023. doi: [10.12962/j26139960.v7i2.446](https://doi.org/10.12962/j26139960.v7i2.446). [Online]. Available: <https://journal.its.ac.id/index.php/sewagati/article/view/446>.
- [6] R. M. V. K. Mehta, *Principles of Electrical Machines*. S Chand and Company Limited, 2002.
- [7] R. T. Bankar and S. S. Salankar, "Head gesture recognition system using gesture cam," in *2015 Fifth International Conference on Communication Systems and Network Technologies*, 2015, pp. 535–538. doi: [10.1109/CSNT.2015.81](https://doi.org/10.1109/CSNT.2015.81).
- [8] MediaPipe, *Face landmark detection guide*, [Online; accessed 13-Dec-2023], 2023. [Online]. Available: https://developers.google.com/Mediapipe/solutions/vision/face_landmarker.
- [9] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. doi: [10.1109/ICEngTechnol.2017.8308186](https://doi.org/10.1109/ICEngTechnol.2017.8308186).
- [10] A. P. J. Dwitama, "Klasifikasi tingkat retakan pada bangunan berbasis citra menggunakan metode convolution neural network," Ph.D. dissertation, Universitas Mataram, 2019.
- [11] R. F. Hakim *et al.*, "Penerapan deep learning menggunakan convolutional neural network untuk klasifikasi citra wayang punakawan," 2018.
- [12] E. Ovalle-Magallanes, J. G. Avina-Cervantes, I. Cruz-Aceves, and J. Ruiz-Pinales, "Transfer learning for stenosis detection in x-ray coronary angiography," *Mathematics*, vol. 8, no. 9, 2020, ISSN: 2227-7390. [Online]. Available: <https://www.mdpi.com/2227-7390/8/9/1510>.
- [13] R. D. Nurfita and S. Gunawan Ariyanto, "Implementasi deep learning berbasis tensorflow untuk pengenalan sidik jari," Ph.D. dissertation, Universitas Muhammadiyah Surakarta, 2018.

- [14] T. Developers, *Tensorflow*, May 2021. [Online]. Available: <https://zenodo.org/record/4758419>.
- [15] M. F. Sanner *et al.*, “Python: A programming language for software integration and development,” *J Mol Graph Model*, vol. 17, no. 1, pp. 57–61, 1999.
- [16] S. Srivastav, “Camera - types, structure and functions,” in Jul. 2021, pp. 83–98.
- [17] F. Samman, D. Hamkah, M. Diatmika, and I. Sahali, “Voltage regulator using a dc-dc converter controlled by interpolated pi gain scheduler for solar charge applications,” *ICIC Express Letters*, vol. 12, pp. 1099–1106, Nov. 2018. doi: [10.24507/icicel.12.11.1099](https://doi.org/10.24507/icicel.12.11.1099).
- [18] M. Staff, 2019. [Online]. Available: <https://www.digikey.com/en/maker/tutorials/2019/controlling-h-bridges>.
- [19] Electins, 2023. [Online]. Available: <https://electins.id/esp32-devkit-v1/>.
- [20] I. S. Hidayat, 2018. [Online]. Available: <https://www.murdockcruz.com/2018/07/03/apa-sih-next-unit-of-computing-atau-nuc-itu-begini-penjelasannya/>.
- [21] D. Watson, 2022. [Online]. Available: <https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html>.
- [22] I. Ekatama, “Perancangan sistem kontrol motor kursi roda secara nirkabel berbasis esp32,” Ph.D. dissertation, Institut Teknologi Sepuluh Nopember, 2024.

BIOGRAFI PENULIS



Batrisyia Zahrani Ananto, lahir pada 01 April 2002 di Jakarta. Penulis merupakan anak pertama dari dua bersaudara. Setelah lulus dari SMA Negeri 8 Jakarta, penulis kemudian melanjutkan pendidikan ke jenjang strata satu di Departemen Teknik Komputer, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember mulai tahun 2020.

[Halaman ini sengaja dikosongkan]