

**TUGAS AKHIR - EC234801**

**KENDALI AC BERBASIS GESTURE TANGAN  
MENGUNAKAN *LONG SHORT-TERM MEMORY* (LSTM)  
PADA PERANGKAT EDGE**

**Ali Akbar Alhabsyi**

NRP 5024 22 1005

Dosen Pembimbing

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

NIP 19680601 199512 1 009

**Dr. Diah Puspito Wulandari, S.T., M.Sc.**

NIP 19801219 200501 2 001

**Program Studi Strata 1 (S1) Teknik Komputer**

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2026

*[Halaman ini sengaja dikosongkan]*



**TUGAS AKHIR - EC234801**

**KENDALI AC BERBASIS GESTURE TANGAN  
MENGUNAKAN *LONG SHORT-TERM MEMORY*  
(LSTM) PADA PERANGKAT EDGE**

**Ali Akbar Alhabsyi**

NRP 5024 22 1005

Dosen Pembimbing

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

NIP 19680601 199512 1 009

**Dr. Diah Puspito Wulandari, S.T., M.Sc.**

NIP 19801219 200501 2 001

**Program Studi Strata 1 (S1) Teknik Komputer**

Departemen Teknik Komputer

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2026

*[Halaman ini sengaja dikosongkan]*



**FINAL PROJECT - EC234801**

**HAND GESTURE BASED AC CONTROL USING *LONG SHORT-TERM MEMORY* (LSTM) ON EDGE DEVICES**

**Ali Akbar Alhabsyi**

NRP 5024 22 1005

Advisor

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

NIP 19680601 199512 1 009

**Dr. Diah Puspito Wulandari, S.T., M.Sc.**

NIP 19801219 200501 2 001

**Undergraduate Study Program of Computer Engineering**

Department of Computer Engineering

Faculty of Intelligent Electrical And Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya

2026

*[Halaman ini sengaja dikosongkan]*

# LEMBAR PENGESAHAN

## KENDALI AC BERBASIS GESTURE TANGAN MENGGUNAKAN *LONG SHORT-TERM MEMORY (LSTM)* PADA PERANGKAT EDGE

### 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: **Ali Akbar Alhabsyi**  
NRP. 5024 22 1005

Disetujui oleh Tim Penguji Tugas Akhir:

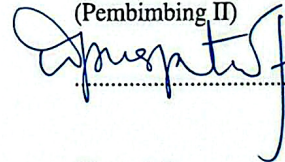
Dr. Eko Mulyanto Yuniarno, S.T., M.T.  
NIP: 19680601 199512 1 009

(Pembimbing I)



Dr. Diah Puspito Wulandari, S.T., M.Sc.  
NIP: 19801219 200501 2 001

(Pembimbing II)



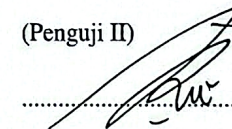
Dr. Arief Kurniawan, S.T., M.T..  
NIP: 19740907 200212 1 001

(Penguji I)



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

(Penguji II)

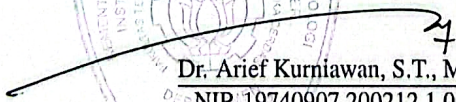


Bambang Widjanarko, S.T., M.T..  
NIP:

(Penguji III)



Mengetahui,  
Kepala Departemen Teknik Komputer FTEIC - ITS

  
Dr. Arief Kurniawan, S.T., M.T..  
NIP: 19740907 200212 1 001

SURABAYA  
Mei, 2026

*[Halaman ini sengaja dikosongkan]*

# APPROVAL SHEET

## HAND GESTURE BASED AC CONTROL USING *LONG SHORT-TERM MEMORY* (LSTM) ON EDGE DEVICES

### 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 Intelligent Electrical And Informatics Technology  
Sepuluh Nopember Institute of Technology

By: **Ali Akbar Alhabsyi**  
NRP. 5024 22 1005

Approved by Final Project Examiner Team:

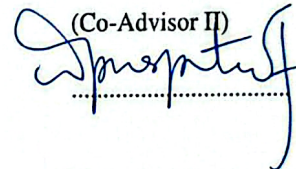
Dr. Eko Mulyanto Yuniarno, S.T., M.T.  
NIP: 19680601 199512 1 009

(Advisor I)



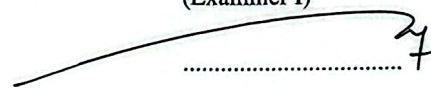
Dr. Diah Puspito Wulandari, S.T., M.Sc.  
NIP: 19801219 200501 2 001

(Co-Advisor II)



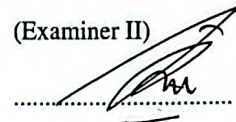
Dr. Arief Kurmiawan, S.T., M.T..  
NIP: 19740907 200212 1 001

(Examiner I)



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

(Examiner II)




Bambang Widjanarko, S.T., M.T..  
NIP:

(Examiner III)



Acknowledged,  
Head of Computer Engineering Department FIEI - ITS

  
Dr. Arief Kurmiawan, S.T., M.T..  
NIP. 19740907 200212 1 001

SURABAYA  
May, 2026

*[Halaman ini sengaja dikosongkan]*

## PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama Mahasiswa / NRP : Ali Akbar Alhabsyi / 5024 22 1005  
Departemen : Teknik Komputer  
Dosen Pembimbing / NIP : Dr. Eko Mulyanto Yuniarno, S.T., M.T. / 19680601 199512 1  
009

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "KENDALI AC BERBASIS GESTURE TANGAN MENGGUNAKAN *LONG SHORT-TERM MEMORY (LSTM)* PADA PERANGKAT EDGE" adalah hasil karya sendiri, bersifat 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, Mei 2026

Mengetahui  
Dosen Pembimbing



Dr. Eko Mulyanto Yuniarno, S.T., M.T.  
NIP. 19680601 199512 1 009

Mahasiswa



Ali Akbar Alhabsyi  
NRP. 5024 22 1005

*[Halaman ini sengaja dikosongkan]*

## STATEMENT OF ORIGINALITY

The undersigned below:

Name of student / NRP : Ali Akbar Alhabsyi / 5024 22 1005  
Department : Computer Engineering  
Advisor / NIP : Dr. Eko Mulyanto Yuniarno, S.T., M.T. / 19680601 199512 1  
009

Hereby declared that the Final Project with the title of "HAND GESTURE BASED AC CONTROL USING *LONG SHORT-TERM MEMORY* (LSTM) ON EDGE DEVICES" 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, May 2026

Acknowledged  
Advisor



Dr. Eko Mulyanto Yuniarno, S.T., M.T.  
NIP. 19680601 199512 1 009

Student



Ali Akbar Alhabsyi  
NRP. 5024 22 1005

*[Halaman ini sengaja dikosongkan]*

## ABSTRAK

Nama Mahasiswa : Ali Akbar Alhabsyi  
Judul Tugas Akhir : KENDALI AC BERBASIS GESTURE TANGAN MENGGUNAKAN  
*LONG SHORT-TERM MEMORY* (LSTM) PADA PERANGKAT  
EDGE  
Pembimbing : 1. Dr. Eko Mulyanto Yuniarno, S.T., M.T.  
2. Dr. Diah Puspito Wulandari, S.T., M.Sc.

Perkembangan teknologi smart home mendorong hadirnya inovasi dalam metode interaksi antara manusia dan perangkat elektronik. Air Conditioner (AC) merupakan salah satu perangkat rumah tangga yang penting, namun hingga kini pengendaliannya masih didominasi oleh remote infrared (IR) konvensional. Metode ini memiliki berbagai keterbatasan, seperti ketergantungan pada *line of sight*, risiko kerusakan atau kehilangan remote, serta kurang fleksibel untuk lingkungan rumah pintar modern. Penelitian ini mengusulkan perancangan sistem kendali AC berbasis gesture tangan dengan memanfaatkan model *Long Short-Term Memory* (LSTM). Kamera digunakan untuk menangkap gerakan tangan, kemudian diproses dengan MediaPipe untuk mengekstraksi koordinat landmark. Data sekuensial hasil ekstraksi diproses menggunakan model LSTM yang dijalankan pada Raspberry Pi sebagai perangkat edge computing, sehingga sistem dapat bekerja secara mandiri tanpa memerlukan server eksternal. Hasil klasifikasi gesture diteruskan ke ESP32 yang dilengkapi dengan modul *infrared transmitter* untuk mengirimkan perintah kendali ke AC. Dengan pendekatan ini, pengguna dapat menyalakan, mematikan, dan mengatur AC hanya dengan menggunakan gesture tangan. Sistem yang dirancang diharapkan mampu bekerja secara *real-time* dengan akurasi tinggi, serta memberikan pengalaman interaksi yang lebih alami, praktis, dan mendukung implementasi smart home.

Kata Kunci: gesture tangan, LSTM, Raspberry Pi, ESP32, infrared, edge computing, smart home, kendali AC.

*[Halaman ini sengaja dikosongkan]*

## ABSTRACT

*Name* : Ali Akbar Alhabsyi  
*Title* : HAND GESTURE BASED AC CONTROL USING *LONG SHORT-TERM MEMORY* (LSTM) ON EDGE DEVICES  
*Advisors* : 1. Dr. Eko Mulyanto Yuniarno, S.T., M.T.  
2. Dr. Diah Puspito Wulandari, S.T., M.Sc.

*The development of smart home technology has driven innovation in human-device interaction methods. Air conditioners (AC) are among the most essential household appliances, yet their control is still dominated by conventional infrared (IR) remotes. This approach has several limitations, such as dependence on line of sight, risk of damage or loss of the remote, and limited flexibility for modern smart home environments. This study proposes the design of an AC control system based on hand gesture recognition using a Long Short-Term Memory (LSTM) model. A camera is used to capture hand movements, which are then processed using MediaPipe to extract landmark coordinates. The resulting sequential data are processed by an LSTM model running on a Raspberry Pi as an edge computing device, enabling the system to operate independently without an external server. The recognized gestures are then forwarded to an ESP32 equipped with an infrared transmitter to send control commands to the AC. With this approach, users can turn the AC on, turn it off, and adjust its settings simply by using hand gestures. The proposed system is expected to operate in real time with high accuracy while providing a more natural, practical, and flexible interaction experience to support smart home implementation.*

*Keywords: hand gesture, LSTM, Raspberry Pi, ESP32, infrared, edge computing, smart home, AC control.*

*[Halaman ini sengaja dikosongkan]*

# KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa, atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penelitian ini yang berjudul **KENDALI AC BERBASIS GESTURE TANGAN MENGGUNAKAN *LONG SHORT-TERM MEMORY* (LSTM) PADA PERANGKAT EDGE**

Penelitian ini disusun dalam rangka pemenuhan Tugas Akhir sebagai syarat kelulusan mahasiswa ITS. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Arief Kurniawan, S.T., M.T., selaku Kepala Departemen Teknik Komputer, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember.
2. Bapak Dr. Eko Mulyanto Yuniarno, S.T., M.T. selaku Dosen Pembimbing I dan Ibu Dr. Diah Puspito Wulandari, S.T., M.Sc. selaku Dosen Pembimbing II yang telah memberikan arahan dan membantu penulis selama pengerjaan tugas akhir ini.
3. Bapak-Ibu dosen pengajar Departemen Teknik Komputer, atas ilmu yang telah diberikan kepada penulis selama menjalani masa perkuliahan.
4. Kedua orang tua, kakak, dan keluarga yang telah memberikan doa serta dukungan selama penulis menempuh pendidikan.
5. Teman-teman Laboratorium Multimedia dan Internet of Things (MIOT) yang membantu penulis dalam pengerjaan tugas akhir ini.
6. Teman-teman Teknik Komputer angkatan 2022 yang memberikan motivasi dan semangat selama masa perkuliahan.

Akhir kata, semoga penelitian ini dapat memberikan manfaat kepada banyak pihak. Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan saran dan kritik yang membangun untuk perbaikan di masa mendatang.

Surabaya, Juli 2026

Ali Akbar Alhabsyi

*[Halaman ini sengaja dikosongkan]*

# DAFTAR ISI

<b>ABSTRAK</b>	<b>i</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>KATA PENGANTAR</b>	<b>v</b>
<b>DAFTAR ISI</b>	<b>vii</b>
<b>DAFTAR GAMBAR</b>	<b>xi</b>
<b>DAFTAR TABEL</b>	<b>xiii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Batasan Masalah . . . . .	2
1.4 Tujuan . . . . .	3
1.5 Manfaat . . . . .	3
<b>2 TINJAUAN PUSTAKA</b>	<b>5</b>
2.1 Hasil Penelitian Terdahulu . . . . .	5
2.1.1 Kendali AC Berbasis Gestur pada ESP32 . . . . .	5
2.1.2 Kursi Roda Pintar Berbasis Pupil Tracking MediaPipe . . . . .	5
2.1.3 Pengenalan Bahasa Isyarat Real-Time Menggunakan LSTM . . . . .	5
2.1.4 Deteksi Gestur Tangan Real-Time Menggunakan MediaPipe dan Bi-LSTM . . . . .	6
2.2 Dasar Teori . . . . .	6
2.2.1 Recurrent Neural Network (RNN) . . . . .	6
2.2.2 Long Short-Term Memory (LSTM) . . . . .	7
2.2.3 Bi-Directional LSTM (BiLSTM) . . . . .	7
2.2.4 MediaPipe (Landmark Gestur Tangan) . . . . .	8
2.2.5 OpenCV (Akuisisi & Pra-Pemrosesan) . . . . .	8
2.2.6 Confusion Matrix & Metrik Evaluasi . . . . .	9

2.2.7	Raspberry Pi sebagai Perangkat Edge . . . . .	9
2.2.8	Rangkaian IR Blaster Berbasis ESP32 . . . . .	10
2.2.9	Pustaka IRremoteESP8266 (Encoder Protokol AC) . . . . .	10
2.2.10	Protokol Komunikasi Jaringan . . . . .	11
<b>3</b>	<b>METODOLOGI</b>	<b>13</b>
3.1	Penjelasan Blok Diagram dan Urutan Pelaksanaan . . . . .	13
3.1.1	Input Data Visual . . . . .	14
3.1.2	Ekstraksi Landmark . . . . .	15
3.1.3	Model LSTM . . . . .	16
3.1.4	Proses Klasifikasi Gestur . . . . .	17
3.1.5	Pengiriman Perintah ke ESP32 . . . . .	18
3.1.6	Pengiriman Sinyal IR dari ESP32 . . . . .	19
3.2	Bahan dan Peralatan . . . . .	19
3.2.1	Perangkat Keras . . . . .	19
3.2.2	Perangkat Lunak . . . . .	20
3.3	Desain Sistem . . . . .	21
3.3.1	Arsitektur Perangkat Keras Sistem . . . . .	21
3.3.2	Arsitektur Perangkat Lunak Sistem . . . . .	23
3.3.3	Perancangan Alur Data dan Logika Kendali . . . . .	24
3.4	Urutan Pelaksanaan Penelitian . . . . .	25
3.4.1	Perancangan Subsistem Perangkat Keras dan Firmware Kendali . . . . .	25
3.4.2	Pengembangan Modul Klasifikasi Gestur Cerdas . . . . .	26
3.4.3	Integrasi Logika Kendali dan Komunikasi Nirkabel . . . . .	28
3.4.4	Evaluasi dan Validasi Sistem . . . . .	29
<b>4</b>	<b>PENGUJIAN DAN ANALISIS</b>	<b>31</b>
4.1	Hasil Pengumpulan dan Pra-pemrosesan Dataset . . . . .	31
4.1.1	Struktur dan Distribusi Dataset . . . . .	31
4.1.2	Visualisasi Perubahan Gestur Dinamis Pada Dataset . . . . .	32
4.1.3	Variasi Pengambilan Data . . . . .	32
4.2	Skenario Pengujian . . . . .	33
4.3	Pengujian Performa Model . . . . .	33
4.3.1	Model Keras . . . . .	33
4.3.2	Model TFLite . . . . .	35

4.4	Pengujian Performa Model Terhadap Jarak . . . . .	36
4.4.1	Pengujian Gestur pada Jarak 1 Meter . . . . .	37
4.4.2	Pengujian Gestur pada Jarak 1,5 Meter . . . . .	38
4.4.3	Pengujian Gestur pada Jarak 2 Meter . . . . .	39
4.4.4	Pengujian Gestur pada Jarak 3 Meter . . . . .	40
4.4.5	Analisis Keseluruhan Pengaruh Jarak . . . . .	41
4.5	Pengujian Performa Model Terhadap Pencahayaan . . . . .	41
4.5.1	Pengujian Gestur pada Kondisi Terang 230 lux . . . . .	42
4.5.2	Pengujian Gestur pada Kondisi Redup 100 lux . . . . .	43
4.5.3	Pengujian Gestur pada Kondisi Gelap 55 lux . . . . .	44
4.5.4	Analisis Keseluruhan Pengaruh Pencahayaan . . . . .	45
4.6	Pengujian Terhadap Beberapa Subjek yang Berbeda . . . . .	45
4.6.1	Pengujian Gestur pada Subjek 1 . . . . .	46
4.6.2	Pengujian Gestur pada Subjek 2 . . . . .	47
4.6.3	Pengujian Gestur pada Subjek 3 . . . . .	48
4.6.4	Analisis Keseluruhan Pengaruh Perbedaan Subjek . . . . .	49
4.7	Pengujian IR Blaster Terhadap Jarak, Arah, dan Desain Casing . . . . .	49
4.7.1	Pengujian IR Blaster dengan Casing Lama Menghadap ke Atas pada Berbagai Jarak . . . . .	49
4.7.2	Pengujian IR Blaster dengan Casing Baru Menghadap ke Atas pada Berbagai Jarak . . . . .	50
4.7.3	Pengujian IR Blaster dengan Casing Lama Menghadap ke Arah AC pada Berbagai Jarak . . . . .	51
4.7.4	Pengujian IR Blaster dengan Casing Baru Menghadap ke Arah AC pada Berbagai Jarak . . . . .	51
4.7.5	Analisis Keseluruhan Pengujian IR Blaster terhadap Jarak, Arah, dan Desain Casing . . . . .	52
4.8	Pengujian IR Blaster Terhadap Pencahayaan . . . . .	52
4.9	Pengujian Kecepatan Pengiriman Perintah . . . . .	53
4.10	Hasil Perancangan Alat . . . . .	54
4.10.1	Hasil Rangkaian IR Blaster . . . . .	55
4.10.2	Hasil Desain PCB . . . . .	55
4.10.3	Hasil Desain Casing . . . . .	57
4.10.4	Hasil Perakitan Alat . . . . .	57
4.11	Pembahasan dan Analisis Hasil . . . . .	58

<b>5 PENUTUP</b>	<b>61</b>
5.1 Kesimpulan . . . . .	61
5.2 Saran . . . . .	61
<b>DAFTAR PUSTAKA</b>	<b>63</b>
<b>BIOGRAFI PENULIS</b>	<b>65</b>

## DAFTAR GAMBAR

2.1	Arsitektur RNN . . . . .	6
2.2	Arsitektur LSTM . . . . .	7
2.3	Arsitektur Bi-LSTM . . . . .	7
2.4	Landmark tangan MediaPipe . . . . .	8
2.5	Confusion matrix . . . . .	9
2.6	Rangkaian IR Blaster . . . . .	10
3.1	Diagram blok sistem . . . . .	13
3.2	Arsitektur perangkat keras sistem kendali AC . . . . .	21
3.3	Diagram alir utama sistem . . . . .	23
3.4	Diagram alir transformasi data dan logika kendali . . . . .	24
3.5	Diagram skematik rangkaian elektronik IR Blaster . . . . .	25
3.6	Contoh visualisasi sampel dataset gestur dinamis . . . . .	27
4.1	Visualisasi urutan temporal gestur dinamis dari frame 1 hingga frame 20 . . . . .	32
4.2	Grafik accuracy dan loss model Keras 20 frame . . . . .	33
4.3	Confusion matrix model Keras 20 frame . . . . .	34
4.4	Confusion matrix model TFLite 20 frame . . . . .	35
4.5	Confusion matrix pengujian gestur pada jarak 1 meter . . . . .	37
4.6	Confusion matrix pengujian gestur pada jarak 1,5 meter . . . . .	38
4.7	Confusion matrix pengujian gestur pada jarak 2 meter . . . . .	39
4.8	Confusion matrix pengujian gestur pada jarak 3 meter . . . . .	40
4.9	Confusion matrix pengujian gestur pada kondisi terang 230 lux . . . . .	42
4.10	Confusion matrix pengujian gestur pada kondisi redup 100 lux . . . . .	43
4.11	Confusion matrix pengujian gestur pada kondisi gelap 55 lux . . . . .	44
4.12	Confusion matrix pengujian gestur pada subjek 1 . . . . .	46
4.13	Confusion matrix pengujian gestur pada subjek 2 . . . . .	47
4.14	Confusion matrix pengujian gestur pada subjek 3 . . . . .	48
4.15	Skematik rangkaian IR Blaster . . . . .	55
4.16	Desain PCB IR Blaster . . . . .	56
4.17	Desain casing lama dan casing baru . . . . .	57

4.18 Hasil perakitan alat IR Blaster . . . . .	58
--	----

## DAFTAR TABEL

4.1	Distribusi jumlah raw sample pada setiap kelas gestur . . . . .	32
4.2	Hasil evaluasi model Keras 20 frame . . . . .	35
4.3	Hasil evaluasi model TFLite 20 frame . . . . .	36
4.4	Rekapitulasi hasil pengujian gestur terhadap jarak . . . . .	36
4.5	Hasil klasifikasi model pada jarak 1 meter . . . . .	37
4.6	Hasil klasifikasi model pada jarak 1,5 meter . . . . .	38
4.7	Hasil klasifikasi model pada jarak 2 meter . . . . .	39
4.8	Hasil klasifikasi model pada jarak 3 meter . . . . .	40
4.9	Rekapitulasi hasil pengujian gestur terhadap pencahayaan . . . . .	41
4.10	Hasil klasifikasi model pada kondisi terang $\pm 230$ lux . . . . .	42
4.11	Hasil klasifikasi model pada kondisi redup $\pm 100$ lux . . . . .	43
4.12	Hasil klasifikasi model pada kondisi gelap $\pm 55$ lux . . . . .	44
4.13	Rekapitulasi hasil pengujian gestur terhadap variasi subjek . . . . .	45
4.14	Hasil klasifikasi model pada Subjek 1 . . . . .	46
4.15	Hasil klasifikasi model pada Subjek 2 . . . . .	47
4.16	Hasil klasifikasi model pada Subjek 3 . . . . .	48
4.17	Pengujian IR Blaster dengan Casing Lama Menghadap ke Atas pada Berbagai Jarak . . . . .	49
4.18	Pengujian IR Blaster dengan Casing Baru Menghadap ke Atas pada Berbagai Jarak . . . . .	50
4.19	Pengujian IR Blaster dengan Casing Lama Menghadap ke Arah AC pada Berbagai Jarak . . . . .	51
4.20	Pengujian IR Blaster dengan Casing Baru Menghadap ke Arah AC pada Berbagai Jarak . . . . .	51
4.21	Hasil pengujian IR Blaster terhadap kondisi pencahayaan . . . . .	53
4.22	Hasil pengujian 15 percobaan waktu pengiriman perintah . . . . .	54

*[Halaman ini sengaja dikosongkan]*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi smart home dan Internet of Things (IoT) telah membawa perubahan besar dalam cara manusia berinteraksi dengan perangkat elektronik sehari-hari. Konsep rumah pintar menekankan pada sistem kendali perangkat yang lebih otomatis, efisien, dan natural. Salah satu perangkat rumah tangga yang paling sering digunakan adalah Air Conditioner (AC). Meskipun perannya vital, pengendalian AC hingga kini masih didominasi oleh remote control berbasis infrared (IR).

Sistem IR memiliki keterbatasan mendasar, seperti ketergantungan pada line of sight, jangkauan terbatas hanya beberapa meter, serta tidak dapat bekerja apabila ada penghalang di antara remote dan sensor. Selain itu, remote fisik juga sering menimbulkan masalah praktis, seperti mudah kotor karena digunakan bersama serta sering hilang atau sulit ditemukan di ruang bersama seperti laboratorium, ruang kelas, atau perkantoran. Hal-hal ini menjadikan remote IR kurang fleksibel dan kurang efisien dalam konteks rumah pintar modern [1].

Keterbatasan remote IR menimbulkan kebutuhan akan metode pengendalian yang lebih fleksibel dan sesuai dengan konsep smart home. Salah satu solusi potensial adalah penggunaan gesture tangan sebagai media interaksi. Gesture merupakan bentuk komunikasi non-verbal yang alami, mudah dipahami, dan dapat dilakukan tanpa menyentuh perangkat. Penelitian terbaru menunjukkan bahwa sistem Gesture Recognition berbasis kamera dapat digunakan untuk otomatisasi rumah dan pengendalian perangkat elektronik dengan tingkat akurasi yang tinggi. Sebagai contoh, penelitian smart home Automation-Based Hand Gesture Recognition membuktikan bahwa Gesture dapat menggantikan fungsi tombol fisik untuk mengontrol perangkat rumah tangga secara real-time [2].

Namun, pengenalan gesture bukanlah hal yang sederhana. Sebagian besar Gesture bersifat dinamis, yang berarti terdiri dari rangkaian gerakan berurutan dalam waktu singkat. Metode pengolahan citra statis seperti CNN (Convolutional Neural Network) cenderung kurang optimal dalam memahami informasi temporal antar frame. Untuk mengatasi keterbatasan ini, dibutuhkan metode yang mampu mengolah data sekuensial.

Salah satu algoritma yang terbukti efektif adalah Long Short-Term Memory (LSTM), yang dirancang khusus untuk mengenali pola pada data berurutan. Dalam konteks edge computing, model LSTM dapat diimplementasikan pada perangkat kecil seperti Raspberry Pi, yang memungkinkan sistem bekerja secara mandiri tanpa harus terhubung ke server eksternal. Penelitian LiReD: A Light-Weight Real-time Fault Detection System for Edge Computing Using LSTM menunjukkan bahwa model LSTM dapat berjalan di Raspberry Pi dan digunakan untuk deteksi real-time dengan hasil yang akurat [3]. Hal ini memperkuat peluang pemanfaatan LSTM dalam aplikasi Gesture Recognition berbasis perangkat edge.

Berdasarkan uraian tersebut, penelitian ini bertujuan merancang sistem kendali AC berbasis gesture tangan menggunakan model LSTM yang dijalankan pada Raspberry Pi sebagai perangkat edge computing. Hasil klasifikasi gesture kemudian dikirim ke ESP32 D1 Mini yang terhubung dengan transistor SS8050 dan LED IR TSAL6400 untuk mengendalikan AC. Dengan pendekatan ini, sistem diharapkan dapat menggantikan remote IR konvensional yang memiliki banyak keterbatasan, sekaligus menghadirkan kendali AC yang lebih natural, praktis, higienis, dan sejalan dengan konsep smart home modern.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka permasalahan yang akan dikaji dalam penelitian ini dapat dirumuskan dalam bentuk pertanyaan berikut:

1. Bagaimana merancang sistem pengenalan gesture tangan berbasis computer vision yang mampu mengenali enam kelas gesture, yaitu ThumbUp, ThumbDown, Mode, Temp\_up, Temp\_down, dan Negative, untuk dijadikan perintah kendali AC?
2. Bagaimana membangun dan mengimplementasikan model Long Short-Term Memory (LSTM) pada Raspberry Pi sehingga dapat memproses gesture tangan secara real-time?
3. Bagaimana cara mengintegrasikan hasil klasifikasi gesture tangan dengan ESP32 D1 Mini, transistor SS8050, dan LED IR TSAL6400 agar mampu menggantikan fungsi remote IR konvensional pada AC?
4. Bagaimana mengevaluasi kinerja sistem, baik dari sisi akurasi pengenalan gesture, kecepatan respons sistem, maupun keandalan transmisi IR dalam mengendalikan AC?

## 1.3 Batasan Masalah

1. Penelitian ini hanya difokuskan pada perangkat Air Conditioner (AC) yang menggunakan remote infrared sebagai media kendali utama.
2. gesture tangan yang digunakan dibatasi pada enam kelas gesture, yaitu ThumbUp, ThumbDown, Mode, Temp\_up, Temp\_down, dan Negative. Gesture kompleks atau kombinasi gerakan di luar keenam kelas tersebut tidak dibahas.
3. Algoritma yang digunakan adalah Long Short-Term Memory (LSTM). Metode lain seperti CNN murni, HMM, atau machine learning klasik tidak diteliti secara mendalam.
4. Sistem dijalankan pada Raspberry Pi sebagai perangkat edge computing untuk pemrosesan gesture, dan ESP32 D1 Mini dengan transistor SS8050 serta LED IR TSAL6400 sebagai pengirim sinyal ke AC.
5. Evaluasi sistem dibatasi pada pengujian akurasi pengenalan gesture, pengaruh jarak kamera, pengaruh pencahayaan, variasi subjek, kecepatan respons sistem, serta keandalan transmisi infrared. Integrasi berbasis cloud dan pengujian pada perangkat elektronik selain AC tidak termasuk dalam lingkup penelitian.

## **1.4 Tujuan**

Tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang dan membangun sistem kendali AC berbasis gesture tangan dengan memanfaatkan computer vision.
2. Mengimplementasikan algoritma Long Short-Term Memory (LSTM) pada Raspberry Pi untuk mengenali gesture tangan secara real-time.
3. Mengintegrasikan hasil klasifikasi gesture dengan ESP32 D1 Mini, transistor SS8050, dan LED IR TSAL6400 agar dapat menggantikan fungsi remote AC konvensional.
4. Mengevaluasi performa sistem berdasarkan parameter akurasi pengenalan gesture, kecepatan respons, dan jarak efektif transmisi IR.

## **1.5 Manfaat**

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Memberikan alternatif kendali AC yang lebih natural, praktis, dan higienis melalui pengenalan gesture tangan tanpa harus menggunakan remote konvensional.
2. Menjadi referensi dalam penerapan algoritma LSTM pada perangkat edge computing (Raspberry Pi) untuk aplikasi real-time Gesture Recognition.
3. Menjadi bahan kajian dan literatur dalam penelitian bidang computer vision, machine learning, dan IoT-based smart home systems.
4. Memberikan gambaran awal tentang potensi integrasi sistem Gesture Recognition dengan perangkat elektronik rumah tangga yang masih menggunakan kendali IR.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Hasil Penelitian Terdahulu**

Pada subbab ini akan dijelaskan mengenai penelitian terdahulu yang menjadi landasan dalam penelitian ini.

##### **2.1.1 Kendali AC Berbasis Gestur pada ESP32**

Hikmal merancang pipeline MediaPipe Hands untuk ekstraksi landmark, diikuti LSTM sebagai pemodel temporal, dan aktuasi perintah AC melalui pemancar IR pada ESP32. Sistem mengklasifikasikan 9 kelas gestur (Suhu Naik, Suhu Turun, Nyala, Mati, Send, NULL, Fan, Speed, Mode) dan pada "model 5" dengan masukan 15 frame pelatihan berhenti pada epoch ke-30 dengan Training/Validation Accuracy masing-masing 1.00 dan Training/Validation Loss 0.0001, serta akurasi uji dan validasi mencapai 100%. Kinerja terbaik tercapai pada jarak pengguna-kamera 30–50 cm dan intensitas cahaya sekitar 78 lux; pada jarak lebih jauh performa menurun. Di sisi aktuasi, modul IR KY-005 pada ESP32 efektif hingga  $\approx 2,5$  m dengan tingkat keberhasilan pengiriman sinyal di atas 90%, meski kelas "NULL" masih menjadi sumber kesalahan dan perlu perbaikan. Temuan ini menegaskan kelayakan arsitektur landmark-sekuens-LSTM untuk kontrol AC real-time di lingkungan rumah pintar. [4]

##### **2.1.2 Kursi Roda Pintar Berbasis Pupil Tracking MediaPipe**

Zhou dkk. membangun kursi roda cerdas berbasis Raspberry Pi 4B yang menggunakan MediaPipe Face (MobileNetV2) untuk pendeteksian wajah dan OpenCV untuk pemrosesan, dengan skema kontrol berbasis pandangan mata. Deteksi kedipan memakai Eye Aspect Ratio, sedangkan arah gerak ditentukan dari posisi pusat pupil (koordinat 468) relatif terhadap sudut mata bagian dalam:  $\downarrow 35\%$  untuk belok kiri,  $\uparrow 65\%$  untuk belok kanan, dan 35–65% untuk lurus. Sistem beroperasi real-time dengan laju pemrosesan  $\approx 30$  fps, dan dilengkapi penghindar rintangan saat jarak ke objek  $\leq 1$  m, kursi berhenti dan mundur  $\approx 10$  cm sebelum menunggu instruksi berikutnya. Hasil uji menunjukkan sistem robust terhadap variasi pencahayaan dan oklusi wajah, sehingga cocok untuk kendali hands-free bagi quadriplegia. [5]

##### **2.1.3 Pengenalan Bahasa Isyarat Real-Time Menggunakan LSTM**

Penulis mengembangkan sistem pengenalan bahasa isyarat dinamis yang mengekstraksi landmark tangan MediaPipe dan fitur sudut gerak, lalu memetakannya menjadi kata dan kalimat bahasa Inggris secara real-time. Mekanisme validasi temporal memberi jeda 3 detik untuk memeriksa kecocokan prediksi tiap kata dinamis. Pada kamus 17 kata, skor deteksi tangan berada di atas 90% untuk seluruh entri, sedangkan akurasi pengenalan gestur dinamis berada pada rentang 64–100% dengan rerata di atas 80%; tantangan utama muncul pada gestur yang mirip secara visual bila hanya mengandalkan landmark tangan dan sudut. [6]

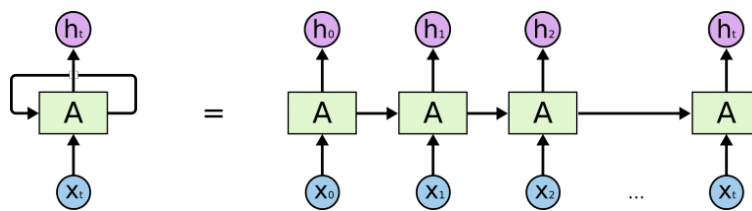
## 2.1.4 Deteksi Gestur Tangan Real-Time Menggunakan MediaPipe dan Bi-LSTM

Dhiran dkk. mengusulkan arsitektur hibrida yang menggabungkan ekstraksi fitur spasial melalui Spiking Convolutional Neural Network (SCNN) dan pemodelan temporal melalui Bi-LSTM, dengan landmark tangan MediaPipe sebagai pendamping fitur. Dataset kustom berisi 10 gestur, masing-masing 100 video berdurasi 2 detik direkam pada 30 fps dengan variasi pencahayaan, latar, dan warna kulit. Dari tiap video diekstrak 16 frame berukuran  $224 \times 224$ , diproses SCNN (dua konvolusi  $3 \times 3$  dengan neuron LIF dan fully-connected 2048-d), serta

digabung dengan 21 landmark tangan (63-d) menjadi vektor fitur 2111-d per frame yang dirangkai sepanjang 32 frame sebelum masuk ke dua lapis Bi-LSTM 128 unit; pelatihan dilakukan 75 epoch dengan batch 16. Sistem mencapai akurasi uji 96,00% dan metrik kuat lain (precision 97,71%, recall 96,00%, F1 96,41%, Cohen's  $\kappa$  0,95), mendekati baseline ResNet50+Bi-LSTM berbasis MediaPipe (97,3%) dengan keunggulan efisiensi komputasi khas SCNN. [7]

## 2.2 Dasar Teori

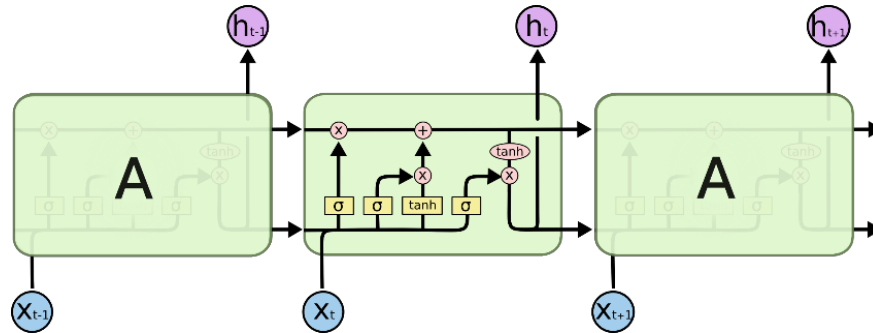
### 2.2.1 Recurrent Neural Network (RNN)



Gambar 2.1: Arsitektur RNN

Recurrent Neural Network (RNN) adalah arsitektur jaringan saraf yang dirancang untuk memproses data berurutan (sequential) dengan memanfaatkan hidden state sebagai memori konteks dari langkah waktu sebelumnya. Berbeda dari jaringan feed-forward yang menganggap setiap input independen, RNN memodelkan ketergantungan temporal sehingga cocok untuk teks, sinyal audio, deret waktu sensor, hingga urutan landmark gestur tangan pada video. Proses pelatihan RNN dilakukan menggunakan Backpropagation Through Time (BPTT) perluasan backpropagation pada dimensi waktu agar gradien dapat disebarkan ke setiap time-step sepanjang urutan [8]. Walau demikian, RNN konvensional rentan vanishing/exploding gradient pada urutan panjang sehingga informasi jangka panjang sulit dipertahankan. Keterbatasan ini memicu adopsi arsitektur ber-gating seperti Long Short-Term Memory (LSTM) yang menstabilkan aliran gradien dan menjaga memori jangka panjang [9].

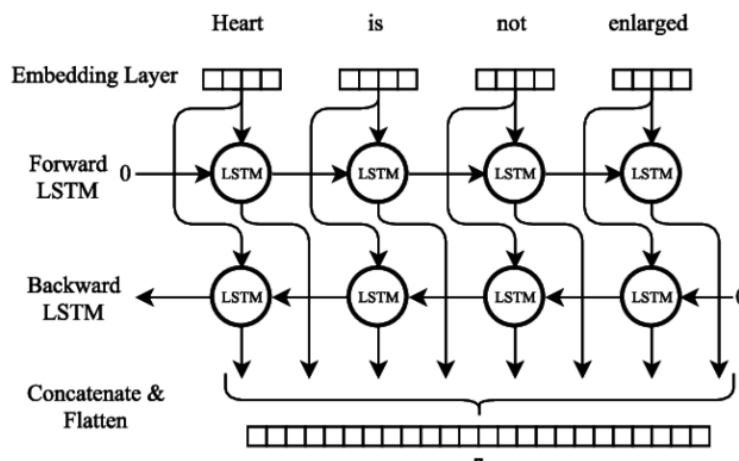
## 2.2.2 Long Short-Term Memory (LSTM)



Gambar 2.2: Arsitektur LSTM

LSTM memperkenalkan cell state sebagai "jalur memori" utama dan tiga gerbang forget, input, output untuk mengatur, menyimpan, atau melupakan informasi secara selektif sepanjang urutan. Mekanisme ini menstabilkan propagasi gradien sehingga ketergantungan jangka panjang dapat dipelajari secara efektif. Pada tugas pengenalan gestur tangan dinamis, masukan model berupa urutan vektor landmark (misalnya 21 titik per frame) yang menggambarkan pola gerak dari waktu ke waktu. LSTM efektif membedakan gestur yang mirip secara spasial tetapi berbeda urutan gerak (mis. "jempol naik" vs "jempol turun") karena memodelkan dinamika antar-frame, bukan hanya bentuk pada satu frame [9]. Kajian sistematis terhadap berbagai varian LSTM menunjukkan komponen krusial yang paling berkontribusi pada kinerja adalah forget gate dan output activation. Implikasi praktisnya, saat deployment di perangkat edge, arsitektur dapat dibuat ringkas tanpa kehilangan performa inti [9].

## 2.2.3 Bi-Directional LSTM (BiLSTM)

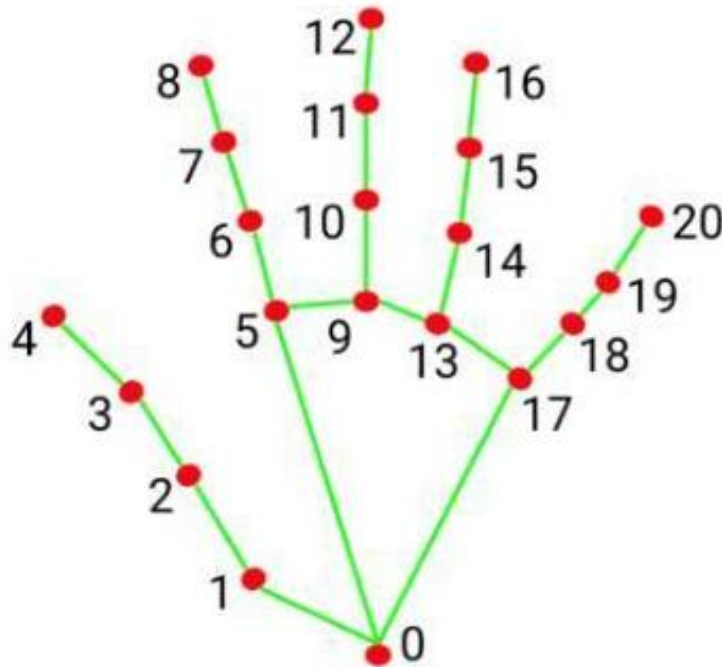


Gambar 2.3: Arsitektur Bi-LSTM

BiLSTM memproses urutan dari dua arah maju (forward) dan mundur (backward) kemudian menggabungkan keduanya menjadi representasi waktu  $-t$  yang lebih kaya konteks. Untuk gestur tangan, pendekatan dua arah membantu ketika segmen gerak bersifat ambigu: model memanfaatkan petunjuk sebelum dan sesudah suatu gerak untuk mempertegas keputusan kelas.

Secara teoretis, bidirectional recurrent networks telah dibuktikan meningkatkan kinerja pada beragam tugas berurutan [10].

## 2.2.4 MediaPipe (Landmark Gestur Tangan)



Gambar 2.4: Landmark tangan MediaPipe

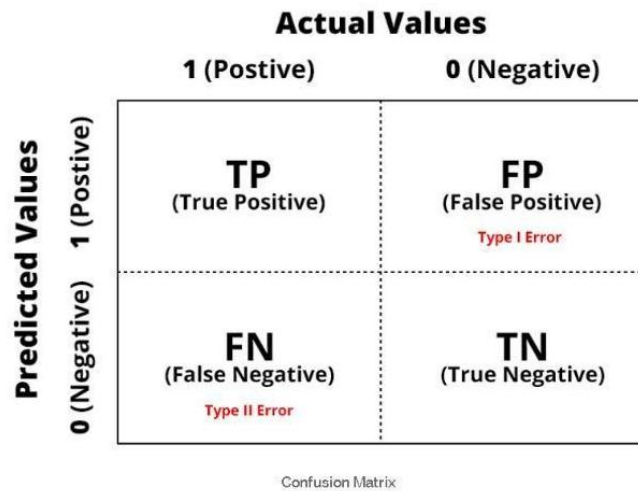
MediaPipe Hands menyediakan pipeline real-time untuk mendeteksi hingga 21 landmark 3D per tangan dari video. Pipeline lazimnya terdiri dari palm detector (pendeteksi telapak) yang cepat dan landmark regressor yang memprediksi koordinat titik jari serta pergelangan. Keluaran per frame ( $21 \times 3$ ) kemudian dirangkai sebagai sekuens 20 frame untuk menjadi masukan LSTM/BiLSTM dalam klasifikasi gestur (mis. jempol naik/turun, melambai). Efisiensi komputasi pipeline ini memudahkan eksekusi real-time di perangkat edge, selaras dengan target kendali AC berbasis gestur [11].

## 2.2.5 OpenCV (Akuisisi & Pra-Pemrosesan)

OpenCV digunakan untuk akuisisi video dari kamera, konversi ruang warna, resize/cropping, denoising, normalization, serta utilitas visualisasi (mis. overlay landmark dan

FPS meter). Tahap pra-pemrosesan ini menstabilkan kualitas input sebelum masuk ke graph MediaPipe sehingga akurasi deteksi landmark dan latensi pipeline tetap terjaga. OpenCV juga memudahkan debugging dan profiling performa sehingga bottleneck pemrosesan dapat diidentifikasi dan dioptimalkan tanpa mengubah arsitektur model.

## 2.2.6 Confusion Matrix & Metrik Evaluasi



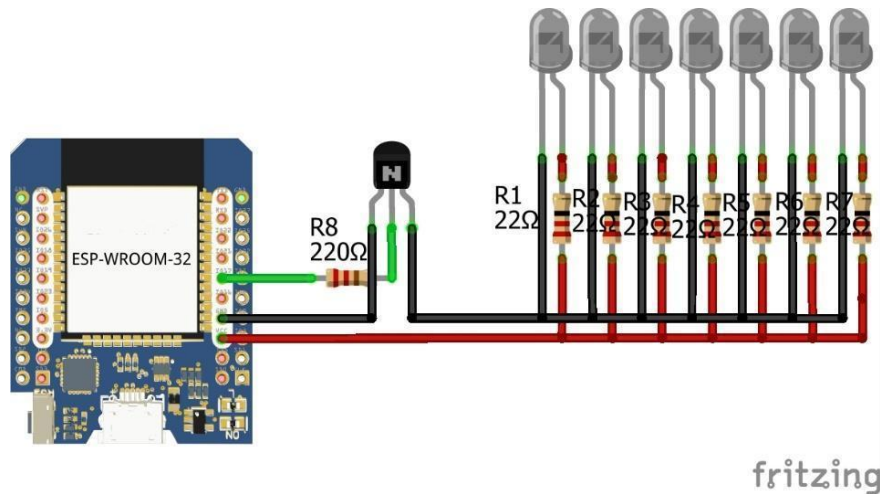
Gambar 2.5: Confusion matrix

Kinerja model dinilai menggunakan confusion matrix (TP, FP, FN, TN) dan metrik turunan: Akurasi, Precision, Recall, serta F1-score. Pada dataset gestur yang sering tidak seimbang (mis. kelas "diam" atau "melambai" dominan), kurva Precision–Recall dan F1 lebih informatif dibanding Akurasi semata. Secara teoretis, terdapat hubungan kuat antara ROC dan Precision– Recall (PR); untuk data skewed, PR memberikan gambaran lebih bermakna terhadap performa pada kelas minor [12].

## 2.2.7 Raspberry Pi sebagai Perangkat Edge

Model LSTM/BiLSTM dijalankan langsung di Raspberry Pi untuk menekan latensi end-to-end dan menjaga keandalan tanpa bergantung koneksi internet. Raspberry Pi menyediakan sumber daya komputasi yang memadai untuk menjalankan pipeline OpenCV, MediaPipe, dan LSTM pada resolusi moderat. Optimisasi praktis meliputi downscaling frame, penentuan ukuran jendela sekuens yang efisien, dan kuantisasi model bila diperlukan.

## 2.2.8 Rangkaian IR Blaster Berbasis ESP32



Gambar 2.6: Rangkaian IR Blaster

Sistem kendali nirkabel yang andal memerlukan integrasi antara unit pemrosesan yang efisien dan antarmuka aktuator yang kuat. Pada penelitian ini, digunakan mikrokontroler ESP32 D1 Mini sebagai otak utama pada sisi penerima atau receiver. Chip ini menggunakan prosesor 32-bit yang memiliki kemampuan komputasi memadai untuk perangkat IoT, serta dilengkapi konektivitas WiFi dan Bluetooth Low Energy. Fitur konektivitas ini sangat krusial untuk implementasi Internet of Things yang kompleks pada lingkungan bangunan pintar [13]. Guna mengatasi tantangan jangkauan sinyal inframerah pada ruangan yang luas, sistem ini dilengkapi dengan rangkaian pengirim sinyal atau IR Blaster khusus yang terdiri dari komponen berikut.

### 2.2.8.1 LED Inframerah 7 Buah

Komponen ini menggunakan tujuh LED IR TSAL6400 yang disusun dalam konfigurasi array untuk memperluas sudut pancaran atau beam angle dan meningkatkan intensitas sinyal total. Konfigurasi ini bertujuan meminimalisir area titik buta sehingga sinyal dapat diterima oleh unit AC dari berbagai posisi tanpa harus mengarahkan alat secara presisi.

### 2.2.8.2 Transistor SS8050

Komponen ini berfungsi sebagai penguat arus dan saklar elektronik. Mengingat pin GPIO pada ESP32 memiliki batas arus keluaran yang rendah (umumnya di bawah 40 mA), pin tersebut tidak mampu menggerakkan tujuh buah LED IR secara langsung. Transistor NPN tipe SS8050 digunakan dalam konfigurasi Common Emitter untuk mengalirkan arus dari sumber daya eksternal ke rangkaian IR LED. Hal ini memastikan LED beroperasi pada intensitas maksimum untuk jangkauan terjauh tanpa membebani mikrokontroler

## 2.2.9 Pustaka IRremoteESP8266 (Encoder Protokol AC)

Keragaman protokol komunikasi antar manufaktur elektronik menjadi hambatan utama dalam menciptakan satu alat pengendali untuk semua perangkat. Kim et al. [14] dalam penelitiannya menggarisbawahi pentingnya konsep Universal Remote Controller yang mampu beradaptasi dengan berbagai perangkat untuk meningkatkan interoperabilitas sistem otomasi rumah tanpa memerlukan modifikasi perangkat keras yang signifikan pada sisi peralatan rumah tangga.

Mengacu pada konsep tersebut, penelitian ini memanfaatkan pustaka open source bernama IR-remoteESP8266, khususnya fitur kelas IRac. Pustaka ini menerapkan lapisan abstraksi perangkat lunak yang menstandarisasi parameter kendali AC (seperti mode operasi, suhu target, dan kecepatan kipas) ke dalam struktur data umum. Saat perintah dieksekusi, sistem secara otomatis menerjemahkan parameter umum tersebut ke dalam format protokol biner spesifik sesuai merek AC yang dikonfigurasi, misalnya Daikin, Panasonic, atau LG.

## **2.2.10 Protokol Komunikasi Jaringan**

Pada sistem kendali AC berbasis gesture tangan ini digunakan dua protokol komunikasi jaringan, yaitu HTTP dan MQTT. HTTP digunakan untuk kebutuhan konfigurasi lokal pada Raspberry Pi, sedangkan MQTT digunakan sebagai jalur komunikasi utama untuk mengirimkan perintah hasil deteksi gesture dari Raspberry Pi menuju ESP32 melalui broker EMQX.

### **2.2.10.1 HTTP sebagai Web Konfigurasi Lokal Raspberry Pi**

HTTP digunakan sebagai protokol untuk menyediakan halaman konfigurasi lokal pada Raspberry Pi. Melalui web konfigurasi ini, pengguna dapat mengakses panel pengaturan menggunakan browser dari perangkat yang berada pada jaringan yang sama. Panel tersebut digunakan untuk mengatur koneksi WiFi, melihat atau memilih daftar device ESP32 yang tersedia, serta menentukan merek AC default yang akan digunakan oleh sistem. Penggunaan HTTP pada bagian ini bertujuan untuk memudahkan proses konfigurasi tanpa perlu mengubah kode program secara manual. Dengan adanya antarmuka web lokal, pengguna dapat menyesuaikan parameter sistem secara lebih praktis, terutama ketika terjadi perubahan jaringan, penambahan perangkat ESP32, atau pergantian merek AC yang dikendalikan.

### **2.2.10.2 MQTT sebagai Jalur Pengiriman Perintah Gesture**

MQTT digunakan sebagai protokol komunikasi utama antara Raspberry Pi, broker EMQX, dan ESP32. Setelah Raspberry Pi melakukan deteksi gesture menggunakan kamera, MediaPipe, dan model LSTM, hasil klasifikasi gesture akan diterjemahkan menjadi perintah kendali AC. Perintah tersebut kemudian dikirim oleh Raspberry Pi ke broker EMQX menggunakan mekanisme publish. ESP32 berperan sebagai subscriber yang menerima pesan dari broker EMQX pada topik tertentu. Setelah pesan diterima, ESP32 memproses isi perintah tersebut dan mengaktifkan rangkaian IR Blaster untuk mengirimkan sinyal infrared ke AC. Dengan mekanisme ini, Raspberry Pi tidak perlu berkomunikasi langsung ke alamat IP ESP32, sehingga sistem menjadi lebih fleksibel dan mudah dikembangkan.

### **2.2.10.3 Integrasi HTTP dan MQTT dalam Sistem**

HTTP dan MQTT digunakan secara bersamaan tetapi memiliki fungsi yang berbeda. HTTP berperan sebagai jalur konfigurasi sistem, sedangkan MQTT berperan sebagai jalur eksekusi perintah. Pengguna terlebih dahulu mengatur WiFi, memilih device ESP32, dan menentukan merek AC default melalui web konfigurasi lokal Raspberry Pi. Setelah konfigurasi tersimpan, sistem dapat berjalan secara otomatis untuk mendeteksi gesture dan mengirimkan perintah kendali AC melalui MQTT. Dengan pembagian fungsi tersebut, sistem menjadi lebih terstruktur. HTTP memudahkan pengguna dalam melakukan pengaturan, sedangkan MQTT memungkinkan pengiriman perintah dari Raspberry Pi ke ESP32 secara ringan dan efisien. Kombinasi keduanya mendukung sistem kendali AC berbasis gesture tangan yang lebih fleksibel, mudah dikonfigurasi, dan sesuai dengan konsep Internet of Things pada lingkungan smart home.

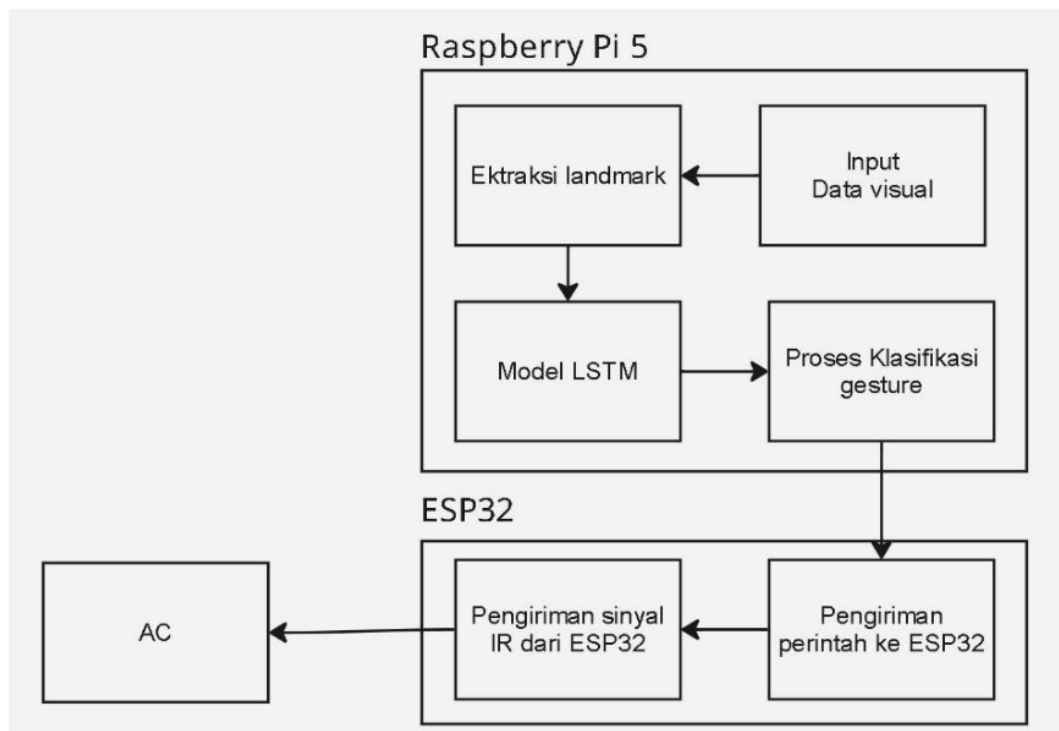
*[Halaman ini sengaja dikosongkan]*

## BAB III

### METODOLOGI

Bab ini menjelaskan tahapan penelitian, rancangan sistem, kebutuhan perangkat, serta skenario evaluasi yang digunakan untuk mengembangkan sistem kendali AC berbasis gestur tangan.

#### 3.1 Penjelasan Blok Diagram dan Urutan Pelaksanaan



Gambar 3.1: Diagram blok sistem

Pengembangan sistem otomasi lingkungan cerdas (smart home) berbasis antarmuka nirsentuh (touchless interface) memerlukan orkestrasi yang terstruktur antara pemrosesan visi komputer tingkat tinggi dan aktuasi perangkat keras tingkat rendah. Berdasarkan topologi yang dirancang pada diagram blok, keseluruhan siklus pemrosesan direpresentasikan ke dalam sebuah alur sekuensial yang terbagi ke dalam dua ranah komputasi fisik utama: subsistem kecerdasan buatan dan pemrosesan sensorik di dalam lingkungan Single Board Computer (SBC) Raspberry Pi (mencakup Input Data Visual, Ekstraksi Landmark, Model LSTM, dan Proses Klasifikasi Gestur), serta eksekusi jaringan aktuator di dalam ekosistem ESP32 (mencakup Pengiriman Perintah dan Pengiriman Sinyal IR). Pendekatan edge computing ini dipilih untuk meminimalkan latensi jaringan, menjaga privasi data visual pengguna agar tidak perlu diunggah ke cloud, serta memastikan keandalan sistem secara waktu nyata. Berikut adalah penjabaran

operasional, mekanis, serta urutan pelaksanaan dari setiap tahapan fungsional secara rinci dan komprehensif.

### **3.1.1 Input Data Visual**

Siklus perambatan data bermula pada tahap Input Data Visual, yang berfungsi sebagai gerbang transduser utama di mana realitas fisik berupa gerakan kinetik pengguna dikonversi menjadi representasi sinyal digital dua dimensi secara terus-menerus. Tahapan ini dioperasikan menggunakan antarmuka Universal Serial Bus (USB) yang mengendalikan sebuah perangkat kamera webcam beresolusi tinggi. Sistem diprogram untuk menjaga laju pengambilan sampel visual (sampling rate) secara konstan pada angka 30 bingkai per detik (FPS). Konsistensi laju bingkai ini merupakan metrik yang kritis; fluktuasi FPS dapat merusak integritas pengukuran delta waktu pergerakan tangan, yang akan berakibat pada kegagalan model deret waktu dalam mengenali kecepatan lambaian. Data citra mentah dari perangkat keras ditransformasikan ke dalam format matriks ruang warna Red-Green-Blue (RGB) menggunakan pustaka OpenCV. Selain itu, diterapkan operasi penyusutan dimensi matriks (downscaling) resolusi secara proporsional. Strategi ini krusial untuk menekan hambatan aliran data (bottleneck) pada bus memori akses acak dan mencegah terjadinya kelebihan beban termal (thermal throttling) pada prosesor ARM arsitektur Raspberry Pi.

#### **3.1.1.1 Persiapan Alat dan Lingkungan**

Sebelum proses perekaman data secara massal dilaksanakan untuk pembuatan dataset latih maupun saat inferensi langsung, dilakukan serangkaian persiapan kalibrasi pada lingkungan kerja. Persiapan ini esensial guna menjamin konsistensi kualitas matriks piksel yang dihasilkan. Pengaturan tata letak pencahayaan ruangan diukur secara empiris, mengingat sensor Complementary Metal-Oxide-Semiconductor (CMOS) pada kamera sangat sensitif terhadap rasio sinyal terhadap derau (Signal-to-Noise Ratio/SNR). Pencahayaan diatur sedemikian rupa agar menyorot objek tangan dengan tingkat lux yang memadai tanpa menimbulkan gangguan saturasi cahaya berlebih (overexposure) atau silau (glare) dari lampu latar. Posisi kamera dipasang tegak lurus dengan sudut pandang frontal terhadap posisi pengguna berdiri atau duduk. Selain itu, alokasi memori pada sistem operasi Raspberry Pi disesuaikan agar tidak ada proses latar belakang berat yang berjalan secara paralel, sehingga siklus pengambilan data dari antarmuka USB berjalan stabil tanpa kehilangan bingkai.

#### **3.1.1.2 Pengkategorian Data yang Diambil**

Sistem ini dirancang untuk mengenali dan membedakan enam kelas gestur tangan secara spesifik. Setiap gestur secara logis dipetakan menjadi instruksi perintah Air Conditioner (AC). Definisi operasional dari keenam kelas tersebut dirancang sedemikian rupa agar tidak memiliki irisan kemiripan kinematika yang membingungkan bagi algoritma:

1. ThumbUp: Gestur ini didefinisikan sebagai pergerakan ibu jari yang diacungkan tegak lurus ke atas dengan sisa jari menggenggam. Fungsi utamanya diekstraksi sebagai perintah inisiasi Power ON untuk menghidupkan AC.
2. ThumbDown: Gestur ini adalah inversi spasial dari gerakan sebelumnya, menuntut rotasi pergelangan tangan hingga ibu jari menunjuk ke bawah. Gestur ini diklasifikasikan sebagai perintah Power OFF untuk mematikan perangkat.
3. Mode: Gestur ini dirancang sebagai gerakan ritmik menyapu (swiping) secara horizontal,

merepresentasikan perintah pertukaran mode operasi. Setiap gestur tervalidasi akan mengirimkan komando iterasi sirkuler untuk mengubah mode sirkulasi dari Cool (pendinginan penuh) ke Auto (otomatis) dan seterusnya.

4. Temp\_up: Direpresentasikan oleh kinetika spasial berupa jemari telunjuk yang didorong secara vertikal ke atas (seperti menunjuk langit). Ini memicu skrip peningkatan batas atas suhu termostat kompresor AC sebesar satu derajat Celsius secara inkremental.
5. Temp\_down: Berbanding terbalik, gestur ini merupakan depresi spasial menunjuk ke arah bawah. Algoritma akan menerjemahkannya sebagai instruksi penurunan suhu target (dekremen) sebesar satu derajat Celsius per validasi lambaian.
6. Negative: Entitas kelas pengecoh atau null class yang mengandung himpunan pergerakan acak, seperti menggaruk wajah, melipat tangan, atau diam tidak bergerak. Eksistensi kelas Negative memberikan model kemampuan untuk menolak probabilitas acak, sehingga sistem tidak memicu false trigger saat tidak ada perintah yang sengaja diberikan.

### **3.1.1.3 Variasi Kondisi Pengambilan Data**

Proses akuisisi data dilakukan pada kondisi yang merepresentasikan penggunaan nyata sistem kendali AC berbasis gestur tangan. Variasi pengambilan data mencakup perbedaan jarak pengguna terhadap kamera hingga sekitar 1 meter, kondisi pencahayaan ruangan, serta variasi posisi dan orientasi tangan di dalam area tangkapan kamera. Variasi ini diperlukan agar dataset tidak hanya merepresentasikan satu kondisi ideal, tetapi juga mencakup perubahan ukuran tangan pada citra, perbedaan kejelasan landmark, dan perbedaan arah hadap tangan saat gestur dilakukan.

Selain variasi jarak dan pencahayaan, data juga diambil dari beberapa posisi tangan, seperti tangan agak miring, mengarah ke depan, mengarah agak ke belakang, dan berada pada lokasi berbeda di dalam bingkai. Pendekatan ini bertujuan membantu model LSTM mempelajari pola gestur dari berbagai kemungkinan bentuk, arah, dan posisi gerakan, sehingga model memiliki kemampuan generalisasi yang lebih baik ketika digunakan secara langsung.

### **3.1.2 Ekstraksi Landmark**

Matriks piksel beresolusi teroptimasi dari tahap sebelumnya kemudian dialirkan ke proses reduksi komputasional pada tahap Ekstraksi Landmark. Sistem mendelegasikan tugas peringkasan citra menjadi struktur anatomis kepada pipeline visi komputer Google MediaPipe Hands. Keluaran akhir dari fase ini tidak lagi berupa citra penuh, melainkan larik berisi representasi topologis 21 titik spasial utama dari struktur tangan manusia. Transformasi ini membebaskan sistem dari keharusan menghitung jutaan nilai piksel yang mahal secara komputasi dan menyusutkannya menjadi barisan vektor data yang siap diproses secara berurutan.

#### **3.1.2.1 Ekstraksi Titik Landmark**

Implementasi pra-pemrosesan ekstraksi fitur ini dijalankan melalui dua sub-layer di dalam MediaPipe. Pertama, pendeteksi telapak ringan (BlazePalm) melakukan lokalisasi wilayah observasi pada keseluruhan gambar. Setelah telapak tangan terlokalisasi dalam bentuk kotak pembatas (bounding box), lapisan regresi (Landmark Regressor) memprediksi 21 titik koordinat tiga dimensi (koordinat x, y, dan estimasi kedalaman relatif z). Titik-titik ini mencakup persendian krusial: pangkal pergelangan tangan, ruas-ruas proksimal, intermedial, hingga titik ujung jari

untuk masing-masing jari kelingking, manis, tengah, telunjuk, dan jempol. Format data yang telah direduksi dari gangguan visual seperti corak pakaian, kondisi pencahayaan, dan bayangan menjadikan abstraksi data lebih sesuai untuk pembelajaran mesin.

### 3.1.2.2 Normalisasi Koordinat

Nilai koordinat landmark yang diperoleh dari pipeline MediaPipe memiliki variasi alami karena titik-titik tersebut bergantung pada ukuran bingkai kamera 640x480 serta posisi tangan terhadap sudut pandang lensa. Untuk menjamin algoritma mengenali pola gestur secara konsisten, serangkaian operasi normalisasi diaplikasikan. Pertama, sistem mereposisi matriks koordinat persendian pangkal (wrist) sebagai titik pusat atau origin Cartesian lokal pada koordinat (0, 0, 0). Sisa 20 titik persendian dihitung secara diferensial terhadap pangkal ini. Kedua, sistem menerapkan transformasi scaling atau penskalaan uniform. Jarak titik pangkal menuju buku sendi pangkal jari tengah dihitung menggunakan metrik Euclidean, lalu digunakan sebagai konstanta pembagi. Proses ini mengeliminasi pengaruh tangan besar, tangan kecil, tangan dekat, maupun tangan jauh menjadi satu dimensi relatif. Terakhir, jika sistem mendeteksi handedness berupa tangan kiri, algoritma pencerminan skalar (scalar inversion) akan membalikkan polaritas sumbu-X secara simetris, sehingga model melihat dinamika tangan dalam orientasi yang seragam.

### 3.1.2.3 Pelabelan Data dan Penyimpanan Akhir

Pasca normalisasi dan ekstraksi fitur, seluruh titik data dibundel menjadi blok komputasi siap latih. Proses ini mengimplementasikan metode pengambilan data berjendela (sliding window). Vektor posisi 21 koordinat di setiap bingkai dirangkai secara paralel bersama perhitungan metrik arah gerakannya. Optimisasi ini menyempitkan ekstraksi fitur menjadi matriks berisi 67 kolom fitur per bingkai. Ke-67 metrik ini merupakan kombinasi langsung dari 63 skalar koordinat persendian tangan ditambah 4 metrik spesifik tambahan untuk mewakili vektor deteksi arah gerak. Data fitur per bingkai disusun menjadi sekuens sepanjang 20 bingkai sesuai konfigurasi input model LSTM. Setiap potongan blok temporal matriks 3 dimensi ini disematkan label identifikasi kategori gesturnya (one-hot encoded target label). Koleksi sampel matriks ini kemudian diekspor dan dikompres menggunakan utilitas NumPy berformat `.npz`, sehingga data dapat dimuat secara efisien ke dalam memori saat proses pelatihan model berlangsung.

## 3.1.3 Model LSTM

Setelah data teroptimasi dalam bentuk tensor numerik berdimensi ( $20 \times 67$ ), alur sistem membawa matriks fitur tersebut ke model Long Short-Term Memory (LSTM). Model ini digunakan karena pengenalan gestur dinamis tidak cukup dianalisis dari satu citra statis, melainkan memerlukan urutan perubahan posisi tangan dari waktu ke waktu. Arsitektur LSTM sebagai salah satu varian Recurrent Neural Network bertugas mempelajari hubungan temporal antar-bingkai dan mengubahnya menjadi probabilitas kelas perintah.

### 3.1.3.1 Long Short-Term Memory

Rancangan arsitektur model dibangun menggunakan TensorFlow Keras dengan mempertimbangkan keseimbangan antara kemampuan deteksi temporal dan efisiensi komputasi pada Raspberry Pi. LSTM dipilih karena mekanisme gerbangnya (forget gate, input gate, dan output gate) mampu mempertahankan informasi penting pada urutan data serta mengurangi risiko vanishing gradient. Secara struktural, model terdiri dari beberapa lapisan berikut:

1. Input Layer: Menerima tensor berbentuk (batch\_size, 20 bingkai, 67 fitur). Fitur tersebut terdiri dari 63 komponen koordinat landmark tangan tiga dimensi dan 4 komponen tambahan yang merepresentasikan arah serta kecepatan gerakan.
2. LSTM Layer 1: Menggunakan 64 hidden unit dengan pengaturan return\_sequences=True agar informasi temporal dari seluruh bingkai tetap diteruskan ke lapisan berikutnya.
3. Dropout Layer: Menggunakan dropout sebesar 20% (0.2) untuk mengurangi risiko overfitting dan membantu model memiliki generalisasi yang lebih baik terhadap data uji.
4. LSTM Layer 2: Menggunakan 32 unit neuron untuk memadatkan informasi temporal dari lapisan sebelumnya menjadi representasi akhir dari satu sekuens gestur.
5. Dense Layer: Menggunakan fungsi aktivasi Softmax untuk menghasilkan distribusi probabilitas pada enam kelas, yaitu ThumbUp, ThumbDown, Mode, Temp\_up, Temp\_down, dan Negative.

### 3.1.3.2 Pelatihan Model

Periode pelatihan dijalankan menggunakan kerangka kerja Keras. Seluruh parameter model (hyperparameters) disetel agar proses pencarian nilai minimum kurva kesalahan berjalan efisien dan tidak mudah terjebak pada kesimpulan lokal yang kurang optimal.

1. Optimizer yang digunakan adalah Adam Optimizer karena mekanisme pembaruan bobot adaptifnya dapat membantu proses konvergensi lebih stabil dibandingkan penurunan gradien stokastik (SGD) klasik.
2. Fungsi kehilangan yang digunakan adalah Categorical Crossentropy karena target klasifikasi disajikan dalam bentuk one-hot encoded label.
3. Pelatihan dikonfigurasi hingga batas maksimum 60 epoch dengan ukuran mini-batch 32 sampel agar proses komputasi tetap sesuai dengan kapasitas memori perangkat.
4. Mekanisme callback ReduceLRonPlateau dan Early Stopping digunakan untuk menjaga kualitas pelatihan. ReduceLRonPlateau menurunkan learning rate ketika performa validasi stagnan, sedangkan Early Stopping menghentikan pelatihan lebih awal apabila performa validasi tidak lagi membaik dalam sejumlah iterasi tertentu.

### 3.1.4 Proses Klasifikasi Gestur

Jalur evaluasi model pasca pelatihan berlanjut pada integrasi waktu nyata di tahap Proses Klasifikasi Gestur. Keluaran representasi temporal dari LSTM diteruskan ke lapisan Dense dengan fungsi aktivasi Softmax untuk menghasilkan enam nilai probabilitas kelas pada rentang 0,00 hingga 1,00. Proses ini menjadi gerbang keputusan terakhir sebelum label gestur diterjemahkan menjadi perintah aktuator.

### 3.1.4.1 Proses Klasifikasi Gestur (Implementasi Waktu Nyata)

Pada operasi waktu nyata, sistem dapat menghadapi prediksi parsial akibat latar yang bising, pencahayaan yang berubah, atau pergerakan jari yang ambigu. Untuk mengurangi risiko perintah salah, program Python menerapkan teknik penyimpanan sementara berbasis Fixed-Length FIFO Queue Buffer.

1. Setiap kali vektor fitur diperoleh, antrean buffer menerima satu baris berisi 67 fitur ke bagian akhir antrean. Jika jumlah data belum mencapai 20 bingkai, proses inferensi belum dijalankan dan sistem tetap berada pada kondisi idle.
2. Setelah antrean mencapai 20 bingkai, bingkai tertua dikeluarkan ketika bingkai baru masuk. Seluruh isi antrean kemudian dikirim ke TFLite Interpreter dengan model berformat `.tflite`.
3. Hasil keluaran Softmax diproses kembali menggunakan Moving Average Window Smoothing untuk meredam fluktuasi prediksi. Sebuah perintah hanya dianggap valid apabila probabilitas kelas dominan secara konsisten melampaui ambang batas yang ditentukan, misalnya lebih dari 80% atau nilai confidence lebih dari 0,8. Mekanisme ini membantu mencegah ghost triggering pada kondisi operasional nyata.

### 3.1.4.2 Evaluasi dan Optimasi Model

Evaluasi model dilakukan menggunakan Confusion Matrix untuk memetakan sebaran prediksi benar dan salah pada setiap kelas. Selain itu, metrik seperti precision, recall, dan F1-score digunakan untuk menilai performa klasifikasi, terutama pada kelas yang berpotensi memiliki kemiripan pola gerak seperti Temp\_up dan Temp\_down. Untuk mengurangi beban komputasi pada Raspberry Pi, model hasil pelatihan dikonversi ke format TensorFlow Lite dengan teknik kuantisasi pasca pelatihan. Optimasi ini bertujuan memperkecil ukuran model dan mempercepat inferensi, sambil tetap menjaga penurunan akurasi agar tidak signifikan.

### 3.1.5 Pengiriman Perintah ke ESP32

Panah transisi dari Raspberry Pi menuju mikrokontroler menandakan perpindahan proses dari interpretasi kecerdasan algoritma ke eksekusi aktuator fisik. Pada tahap ini, Raspberry Pi bertugas membentuk perintah kendali dan mengirimkannya ke ESP32 D1 Mini melalui jaringan lokal.

#### 3.1.5.1 Konversi Klasifikasi Menjadi Perintah

Hasil akhir dari proses penyaringan prediksi berupa string perintah, seperti POWER, MODE\_TOGGLE, TEMP\_UP, atau TEMP\_DOWN. String tersebut tidak dikirim secara langsung, tetapi dikemas ke dalam muatan pesan JavaScript Object Notation (JSON). Muatan pesan JSON berisi informasi kendali seperti ID perangkat, merek AC, jenis perintah, dan parameter tambahan apabila diperlukan. Paket tersebut selanjutnya ditransmisikan melalui protokol Message Queuing Telemetry Transport (MQTT) pada topik lokal ESP32-AC01/control. Penggunaan muatan pesan JSON melalui MQTT membuat format komunikasi lebih terstruktur dan mudah divalidasi di sisi penerima.

### **3.1.5.2 Mikrokontroler Menerima Perintah**

Pada sisi penerima, mikrokontroler ESP32 D1 Mini diprogram untuk memantau (subscribe) aliran data pada topik MQTT ESP32-AC01/control. Setelah muatan pesan JSON diterima, sistem memanggil fungsi callback MQTT untuk melakukan penguraian dan mengurai isi pesan. Program kemudian melakukan validasi serta pembatasan nilai (clamping), misalnya membatasi target suhu agar tetap berada pada rentang 17°C hingga 30°C. Selanjutnya, algoritma normalisasi status memastikan perubahan mode operasi tetap selaras dengan status kecepatan kipas. Sebagai contoh, saat komando Mode Toggle mengubah status menuju Auto, mikrokontroler dapat menyesuaikan kecepatan kipas ke level Auto dan mengatur target suhu bawaan pada 25°C.

### **3.1.6 Pengiriman Sinyal IR dari ESP32**

Fase akhir dalam rantai otomasi adalah pengiriman sinyal IR dari ESP32. Pada tahap ini, perintah digital berformat JSON yang telah diproses diubah menjadi sinyal inframerah yang dapat diterima oleh unit pendingin ruangan.

#### **3.1.6.1 Penerjemahan Perintah ke Protokol Inframerah**

Penerjemahan perintah dilakukan menggunakan pustaka open-source IRremoteESP8266, khususnya objek IRac. Pustaka ini menyediakan basis data protokol dari berbagai produsen AC, seperti COOLIX (Midea), LG, Sharp, Daikin, dan Panasonic. Program mendefinisikan Sharp sebagai protokol bawaan, tetapi tetap memungkinkan pergantian protokol vendor secara dinamis. Mikrokontroler membandingkan status AC terbaru (current state) dengan status sebelumnya (previous state), lalu membentuk rentetan pulsa (pulse timing) yang merepresentasikan logika 0 dan 1 sesuai merek AC yang dipilih.

#### **3.1.6.2 Transmisi Sinyal Melalui Rangkaian LED**

Sinyal keluaran protokol AC dipancarkan melalui rangkaian LED inframerah. Proses transmisi dirancang melalui pin GPIO 17 pada mikrokontroler ESP32 D1 Mini. Karena pin GPIO tidak dirancang untuk menggerakkan beberapa LED inframerah berintensitas tinggi secara langsung, rangkaian menggunakan transistor NPN SS8050 sebagai penguat arus. Denyut tegangan keluaran dari pin GPIO 17 diberikan ke basis transistor, sehingga arus eksternal dapat mengalir menuju susunan tujuh unit LED IR TSAL6400. Sinyal inframerah kemudian dipancarkan menuju sensor AC sasaran agar perangkat dapat merespons perintah dengan baik.

## **3.2 Bahan dan Peralatan**

Pelaksanaan penelitian dan pengembangan sistem didukung oleh perangkat keras dan perangkat lunak dengan rincian sebagai berikut.

### **3.2.1 Perangkat Keras**

Perangkat keras utama yang digunakan untuk membangun sistem ini terdiri dari lima komponen inti.

1. Raspberry Pi: Perangkat ini berfungsi sebagai unit pemrosesan utama atau edge computing device. Pemilihan Raspberry Pi didasarkan pada kebutuhan komputasi model kecerdasan buatan yang responsif. Sebagaimana dijelaskan dalam studi analisis kinerja oleh

Lee et al. [15], purwarupa komputasi berbasis Raspberry Pi modern memiliki kemampuan yang memadai untuk menjalankan pemrosesan data kompleks di sisi edge dengan latensi yang rendah, sehingga sangat sesuai untuk aplikasi kendali waktu nyata.

2. Mikrokontroler ESP32: Komponen ini berperan sebagai node aktuator yang menghubungkan perintah digital dengan perangkat fisik. Varian D1 Mini dipilih karena ukurannya yang ringkas dan efisiensi daya yang tinggi dalam menangani konektivitas WiFi serta modulasi sinyal inframerah.
3. Kamera Webcam: Perangkat ini digunakan sebagai sensor visual untuk menangkap citra gestur tangan pengguna yang kemudian dikirimkan ke unit pemrosesan utama melalui antarmuka USB.
4. Rangkaian IR Blaster Kustom: Rangkaian ini dibuat khusus untuk memancarkan sinyal kendali ke unit AC. Komponen penyusunnya meliputi 7 buah LED IR TSAL6400 yang disusun secara larik untuk memperluas jangkauan pancaran, serta satu buah transistor SS8050 yang berfungsi sebagai penguat arus agar LED dapat bekerja pada intensitas maksimal tanpa membebani mikrokontroler.
5. Unit Pendingin Udara atau AC: Perangkat ini berfungsi sebagai objek kendali dalam pengujian sistem. Penelitian ini menggunakan minimal dua unit AC dari pabrikan yang berbeda guna memvalidasi kemampuan fitur kendali multi-merek.

### 3.2.2 Perangkat Lunak

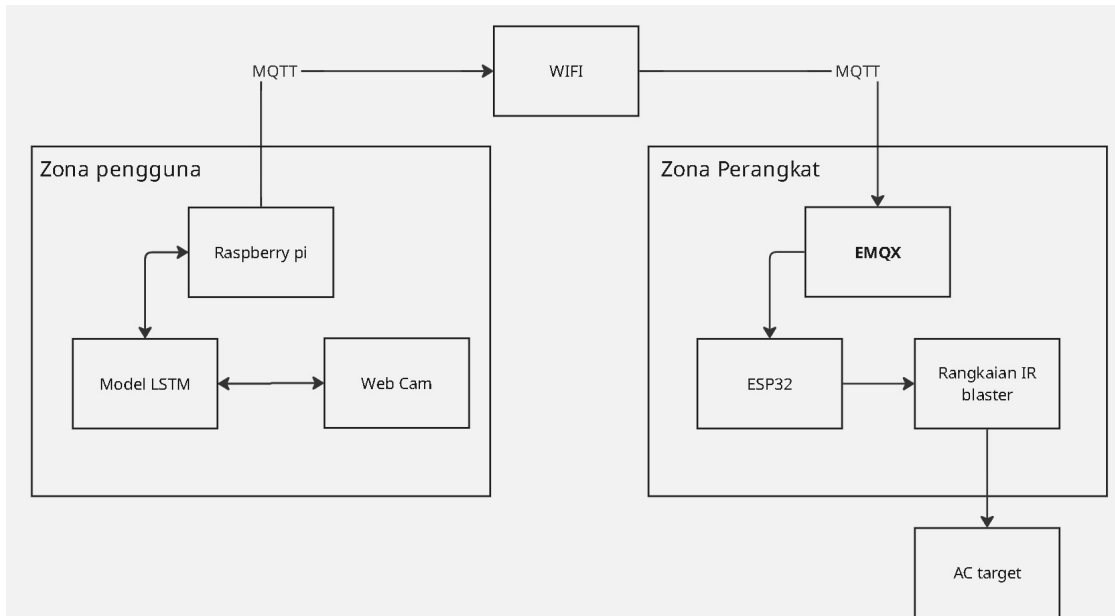
Pengembangan logika dan antarmuka sistem didukung oleh perangkat lunak serta pustaka pemrograman berikut.

1. Python 3: Bahasa pemrograman ini digunakan sebagai fondasi utama di lingkungan Raspberry Pi untuk mengembangkan algoritma pemrosesan citra, logika model deep learning, dan protokol komunikasi jaringan.
2. MediaPipe Hands: Kerangka kerja persepsi mesin ini digunakan untuk mendeteksi keberadaan tangan dan mengekstraksi 21 titik koordinat kerangka tangan. Menurut Zhang et al. [16], arsitektur MediaPipe Hands dirancang khusus untuk memberikan performa pelacakan tangan yang presisi tinggi secara waktu nyata pada perangkat seluler dan tertanam, menjadikannya solusi yang efisien untuk sistem kendali berbasis gestur.
3. TensorFlow dan Keras: Pustaka ini digunakan untuk merancang arsitektur model Long Short-Term Memory atau LSTM. Selain itu, pustaka ini juga digunakan untuk melatih model dan mengonversinya ke format TensorFlow Lite agar dapat dijalankan dengan ringan pada Raspberry Pi.
4. Arduino IDE: Lingkungan pengembangan terintegrasi ini digunakan untuk menulis dan mengunggah kode program atau firmware ke dalam mikrokontroler ESP32, termasuk di dalamnya pustaka untuk protokol inframerah.
5. EMQX MQTT Broker: Perangkat lunak ini digunakan sebagai broker MQTT yang menjadi perantara komunikasi antara Raspberry Pi dan ESP32. Raspberry Pi bertindak sebagai penerbit yang mengirimkan perintah hasil deteksi gestur, sedangkan ESP32 bertindak sebagai pelanggan yang menerima perintah tersebut untuk menjalankan aksi kendali AC melalui IR Blaster.

### 3.3 Desain Sistem

Desain sistem pada penelitian ini dirancang menggunakan arsitektur Edge-IoT terdistribusi, yaitu pemrosesan gestur dilakukan pada perangkat edge Raspberry Pi, sedangkan eksekusi kendali AC dilakukan oleh ESP32 melalui rangkaian IR Blaster. Raspberry Pi bertugas menangkap citra dari kamera, mengekstraksi fitur tangan, menjalankan inferensi model LSTM, serta mengirimkan hasil perintah melalui protokol MQTT. Sementara itu, ESP32 bertugas menerima perintah dari broker EMQX, menerjemahkan perintah tersebut menjadi sinyal inframerah, dan mengirimkannya ke AC target. Secara umum, sistem terdiri dari dua zona utama, yaitu zona pengguna dan zona perangkat. Zona pengguna berisi Raspberry Pi, webcam, dan model LSTM yang berfungsi sebagai pusat pemrosesan gestur. Zona perangkat berisi broker EMQX, ESP32, rangkaian IR Blaster, dan AC target. Kedua zona tersebut dihubungkan melalui jaringan WiFi menggunakan protokol MQTT sebagai jalur komunikasi utama untuk pengiriman perintah kendali.

#### 3.3.1 Arsitektur Perangkat Keras Sistem



Gambar 3.2: Arsitektur perangkat keras sistem kendali AC

Arsitektur perangkat keras sistem terdiri dari beberapa komponen utama, yaitu webcam, Raspberry Pi, jaringan WiFi, broker EMQX, ESP32, rangkaian IR Blaster, dan AC target. Webcam digunakan sebagai sensor visual untuk menangkap gerakan tangan pengguna. Data video dari webcam kemudian diproses oleh Raspberry Pi untuk mendeteksi tangan, mengekstraksi fitur, dan melakukan klasifikasi gestur menggunakan model LSTM. Raspberry Pi berperan sebagai pusat pemrosesan utama pada sisi pengguna. Perangkat ini menjalankan program akuisisi video, ekstraksi fitur, inferensi model LSTM, serta komunikasi MQTT. Setelah gestur berhasil dikenali dengan tingkat kepercayaan tertentu, Raspberry Pi akan memetakan hasil klasifikasi menjadi perintah kendali AC, seperti menyalakan AC, mematikan AC, menaikkan suhu, menurunkan suhu, atau mengganti mode. Perintah tersebut kemudian dikirim melalui jaringan WiFi menuju broker EMQX. Pada sisi perangkat, ESP32 berfungsi sebagai aktuator nirkabel yang menerima perintah dari EMQX. ESP32 kemudian melakukan penguraian perintah, menentukan

format sinyal sesuai merek AC, melakukan proses encoding inframerah, lalu mengirimkan sinyal melalui rangkaian IR Blaster. Rangkaian IR Blaster diarahkan ke AC target agar sinyal inframerah dapat diterima oleh sensor AC seperti halnya perintah dari remote konvensional.

### **3.3.1.1 Zona Pengguna (Unit Pemrosesan Visual)**

Zona pengguna merupakan bagian sistem yang berinteraksi langsung dengan pengguna melalui gestur tangan. Komponen utama pada zona ini adalah webcam, Raspberry Pi, dan model LSTM. Webcam menangkap gerakan tangan pengguna secara waktu nyata, kemudian Raspberry Pi memproses citra tersebut untuk mendeteksi keberadaan tangan. Jika tangan terdeteksi, sistem akan mengekstraksi fitur gerakan tangan dan menyusunnya menjadi data sekuensial. Data sekuensial tersebut menjadi masukan bagi model LSTM. Model LSTM digunakan karena gestur yang dikenali bersifat dinamis dan bergantung pada urutan gerakan antar-bingkai. Hasil keluaran model berupa kelas gestur beserta nilai confidence. Nilai confidence digunakan untuk memastikan bahwa perintah hanya dikirim apabila prediksi model cukup yakin, sehingga dapat mengurangi kemungkinan kesalahan perintah akibat gerakan yang tidak disengaja.

### **3.3.1.2 Zona Perangkat**

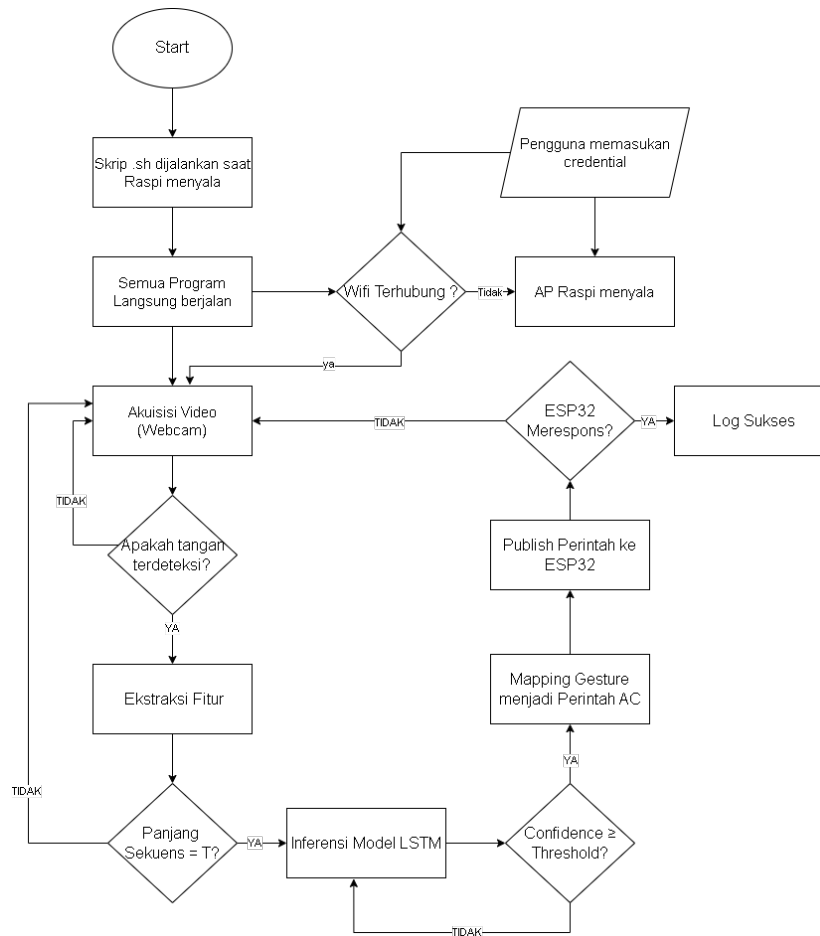
Zona perangkat merupakan bagian sistem yang bertugas menjalankan perintah kendali fisik terhadap AC. Komponen pada zona ini terdiri dari broker EMQX, ESP32, rangkaian IR Blaster, dan AC target. EMQX berperan sebagai broker MQTT yang menjadi perantara komunikasi antara Raspberry Pi dan ESP32. Dengan adanya broker, Raspberry Pi tidak perlu mengirimkan perintah langsung ke alamat IP ESP32, melainkan cukup memublikasikan pesan ke topik MQTT tertentu. ESP32 berperan sebagai pelanggan yang menerima perintah dari broker EMQX. Setelah pesan diterima, ESP32 membaca isi perintah dan menerjemahkannya menjadi sinyal inframerah. Proses ini dilakukan dengan bantuan pustaka inframerah yang mendukung pengkodean protokol berbagai merek AC. Setelah sinyal berhasil dibentuk, ESP32 mengaktifkan rangkaian IR Blaster untuk memancarkan perintah ke AC target.

### **3.3.1.3 Mekanisme Komunikasi Antar Zona**

Komunikasi antar zona dilakukan melalui jaringan WiFi menggunakan protokol MQTT. Raspberry Pi bertindak sebagai penerbit yang mengirimkan perintah hasil deteksi gestur, sedangkan ESP32 bertindak sebagai pelanggan yang menerima dan menjalankan perintah tersebut. Broker EMQX berada di antara keduanya sebagai penghubung komunikasi.

Alur komunikasi dimulai ketika Raspberry Pi memperoleh hasil klasifikasi gestur yang valid. Gestur tersebut dipetakan menjadi perintah AC, kemudian dikirim ke broker EMQX melalui topik MQTT. ESP32 yang telah berlangganan pada topik tersebut menerima pesan, memproses isi muatan pesan, lalu mengaktifkan IR Blaster. Dengan mekanisme ini, sistem menjadi lebih fleksibel karena perangkat pemroses gestur dan perangkat pengirim inframerah dapat ditempatkan secara terpisah sesuai kebutuhan ruangan.

### 3.3.2 Arsitektur Perangkat Lunak Sistem

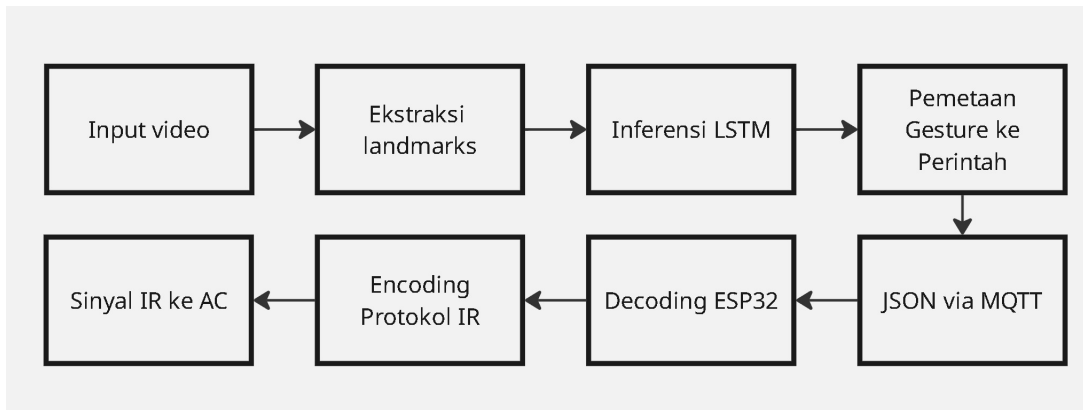


Gambar 3.3: Diagram alir utama sistem

Arsitektur perangkat lunak sistem dibagi menjadi dua bagian, yaitu perangkat lunak pada Raspberry Pi dan firmware pada ESP32. Pada Raspberry Pi, program utama dijalankan secara otomatis setelah perangkat menyala. Program tersebut meliputi pengecekan koneksi WiFi, layanan konfigurasi lokal, akuisisi video dari webcam, deteksi tangan, ekstraksi fitur, inferensi model LSTM, pemetaan gestur ke perintah AC, dan publikasi perintah melalui MQTT. Pada tahap awal, Raspberry Pi melakukan pengecekan koneksi WiFi. Jika Raspberry Pi belum terhubung ke jaringan, sistem akan mengaktifkan mode access point dan menyediakan halaman konfigurasi lokal agar pengguna dapat memasukkan kredensial WiFi. Setelah koneksi berhasil, program utama akan mulai melakukan akuisisi video dari webcam. Setiap bingkai diperiksa untuk mengetahui apakah tangan pengguna terdeteksi. Jika tangan tidak terdeteksi, sistem kembali mengambil bingkai berikutnya. Jika tangan terdeteksi, sistem melakukan ekstraksi fitur dan menyusun data ke dalam sekuens sepanjang 20 bingkai. Ketika panjang sekuens sudah terpenuhi, data dimasukkan ke model LSTM untuk proses inferensi. Hasil inferensi kemudian diperiksa berdasarkan nilai confidence. Jika confidence belum memenuhi threshold, sistem tidak mengirimkan perintah dan kembali melakukan proses deteksi. Jika confidence memenuhi threshold, hasil gestur dipetakan menjadi perintah AC dan dikirim ke ESP32 melalui broker EMQX. Pada sisi ESP32, firmware diawali dengan proses inialisasi serial, GPIO, dan pustaka yang diperlukan. Setelah itu, ESP32 memeriksa apakah kredensial WiFi telah tersimpan.

Jika belum tersedia, ESP32 akan mengaktifkan access point agar pengguna dapat memasukkan kredensial jaringan. Jika kredensial sudah tersedia, ESP32 akan mencoba terhubung ke WiFi. Setelah berhasil terhubung, ESP32 melakukan koneksi ke server EMQX dan mulai menunggu perintah dari broker. Diagram alir lengkap dari sistem ini diilustrasikan pada Gambar 3.3.

### 3.3.3 Perancangan Alur Data dan Logika Kendali



Gambar 3.4: Diagram alir transformasi data dan logika kendali

Alur kerja sistem dimulai dari proses penyalaan Raspberry Pi. Setelah perangkat aktif, seluruh program utama dijalankan secara otomatis. Sistem terlebih dahulu memeriksa koneksi WiFi. Jika koneksi belum tersedia, Raspberry Pi mengaktifkan access point dan pengguna memasukkan kredensial melalui halaman konfigurasi lokal. Setelah WiFi terhubung, sistem masuk ke proses akuisisi video menggunakan webcam. Bingkai video yang diperoleh akan diproses untuk mendeteksi keberadaan tangan. Jika tangan tidak terdeteksi, sistem kembali melakukan akuisisi video secara terus-menerus. Jika tangan terdeteksi, sistem mengekstraksi fitur landmark tangan dan menyusun data ke dalam sekuens. Apabila panjang sekuens belum mencapai 20 bingkai, sistem akan terus menambahkan data dari bingkai berikutnya. Setelah panjang sekuens terpenuhi, data dikirim ke model LSTM untuk proses klasifikasi gestur. Hasil klasifikasi dari model LSTM berupa kelas gestur dan nilai confidence. Sistem hanya akan melanjutkan proses apabila nilai confidence lebih besar atau sama dengan threshold yang telah ditentukan. Jika nilai confidence belum memenuhi threshold, sistem tidak melakukan eksekusi perintah dan kembali ke proses pengambilan data. Hal ini bertujuan untuk mencegah perintah AC terkirim akibat gerakan tangan yang tidak jelas atau prediksi model yang belum cukup kuat. Jika hasil klasifikasi dinyatakan valid, gestur akan dipetakan menjadi perintah kendali AC. Perintah tersebut kemudian dipublikasikan oleh Raspberry Pi ke broker EMQX melalui MQTT.

ESP32 yang telah terhubung ke broker akan menerima pesan tersebut. Jika ESP32 merespons, sistem mencatat proses sebagai log sukses. Jika ESP32 tidak merespons, sistem kembali ke proses akuisisi video dan dapat melakukan pengiriman ulang pada deteksi berikutnya. Pada ESP32, alur kendali dimulai dari inialisasi perangkat, pengecekan kredensial WiFi, koneksi ke jaringan, dan koneksi ke server EMQX. Setelah berhasil terhubung ke EMQX, ESP32 berada pada kondisi menunggu perintah. Ketika terdapat perintah masuk, ESP32 melakukan penguraian muatan pesan untuk membaca jenis perintah, merek AC, dan parameter kendali yang diperlukan. Setelah itu, perintah dikonversi melalui proses IR encoding menjadi sinyal inframerah

yang sesuai dengan protokol AC. Sinyal tersebut kemudian dikirim melalui IR Blaster ke AC target. Setelah pengiriman selesai, ESP32 kembali ke kondisi menunggu perintah berikutnya. Dengan alur tersebut, sistem dapat bekerja secara otomatis mulai dari deteksi gestur hingga eksekusi perintah pada AC. Raspberry Pi bertanggung jawab pada proses pengenalan gestur dan pengiriman perintah, sedangkan ESP32 bertanggung jawab pada penerimaan perintah dan pengiriman sinyal inframerah. Pemisahan tugas ini membuat sistem lebih modular, mudah dikembangkan, dan sesuai untuk implementasi kendali AC berbasis gestur tangan pada lingkungan smart home.

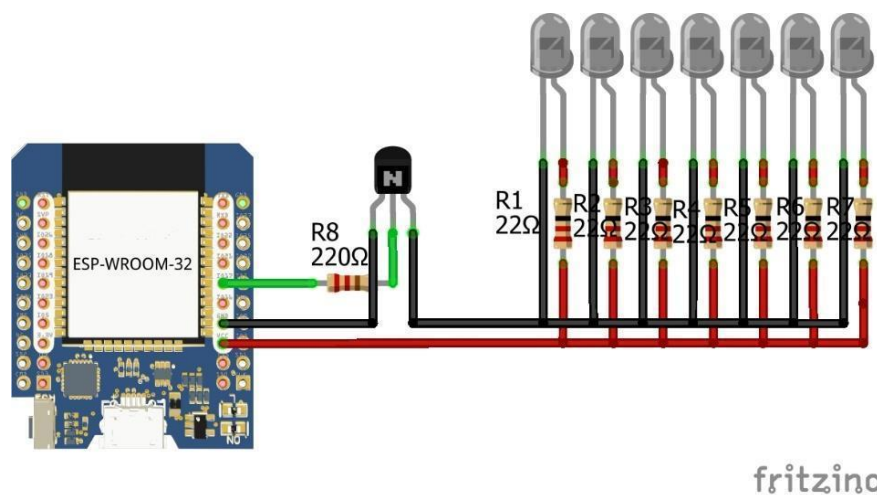
### 3.4 Urutan Pelaksanaan Penelitian

Pelaksanaan penelitian disusun secara sistematis menjadi tahapan-tahapan yang saling terintegrasi, dimulai dari perancangan perangkat keras hingga pengembangan model kecerdasan buatan. Pendekatan bertahap ini diterapkan untuk memastikan setiap subsistem mampu beroperasi secara optimal dan stabil sebelum disatukan menjadi sistem kendali yang utuh dan kompleks.

#### 3.4.1 Perancangan Subsistem Perangkat Keras dan Firmware Kendali

Tahap awal penelitian difokuskan pada pengembangan antarmuka fisik yang bertugas menerjemahkan perintah digital dari sistem komputasi menjadi sinyal fisik yang dapat dipahami oleh unit pendingin udara. Pengembangan ini mencakup dua aspek utama, yaitu perancangan sirkuit elektronik pemancar dan pemrograman logika kendali pada mikrokontroler.

##### 3.4.1.1 Pengembangan Rangkaian Pemancar Sinyal (IR Blaster)



Gambar 3.5: Diagram skematik rangkaian elektronik IR Blaster

Proses perancangan perangkat keras dimulai dengan pengembangan rangkaian pemancar sinyal inframerah atau IR Blaster yang dirancang mampu menjangkau unit AC dari berbagai sudut ruangan. Mengingat keterbatasan arus pada pin keluaran mikrokontroler ESP32 yang tidak memadai untuk menggerakkan LED inframerah berintensitas tinggi secara langsung, diperlukan sebuah rangkaian penguat arus. Solusi teknis yang diterapkan adalah penggunaan transistor tipe NPN SS8050 dalam konfigurasi sakelar elektronik, yang berfungsi mengalirkan daya langsung dari sumber tegangan eksternal ke rangkaian LED. Untuk memperluas sudut

pancaran sinyal (beam angle), sebanyak tujuh buah LED IR TSAL6400 disusun dalam konfigurasi larik paralel dan seri. Konfigurasi ini bertujuan meminimalkan area titik buta (blind spot), sehingga sinyal kendali dapat diterima oleh sensor penerima pada AC meskipun alat tidak diarahkan secara presisi.

### **3.4.1.2 Implementasi Protokol Kendali pada Firmware**

Setelah perangkat keras tersusun, fokus beralih pada pengembangan firmware untuk mikrokontroler ESP32 D1 Mini. Berbeda dengan pendekatan konvensional yang mengharuskan perekaman sinyal mentah dari remote asli, sistem ini mengadopsi pustaka IRremoteESP8266, khususnya kelas IRac. Pustaka ini menyediakan pangkalan data protokol dari berbagai produsen AC terkemuka seperti Daikin, Panasonic, LG, Sharp, dan Midea. Keunggulan utama pendekatan ini adalah penerapan lapisan abstraksi perangkat lunak yang menstandarisasi parameter kendali, seperti mode operasi, suhu target, dan kecepatan kipas, ke dalam struktur data umum bernama `stdAc::state_t`. Mekanisme ini memungkinkan mikrokontroler membangkitkan sinyal protokol yang sesuai dengan merek AC yang dipilih secara langsung, tanpa perlu melalui proses decoding sinyal manual yang rentan terhadap gangguan noise.

### **3.4.1.3 Logika Konektivitas dan Integrasi IoT**

Logika konektivitas pada ESP32 dirancang agar perangkat dapat terhubung ke jaringan WiFi dan broker MQTT secara mandiri. Saat ESP32 dinyalakan, firmware melakukan inisialisasi serial, GPIO, serta pustaka yang diperlukan. Setelah itu, ESP32 memeriksa apakah kredensial WiFi telah tersimpan. Jika belum tersedia, ESP32 mengaktifkan access point agar pengguna dapat memasukkan kredensial WiFi. Setelah berhasil terhubung ke jaringan WiFi, ESP32 melakukan koneksi ke broker EMQX. ESP32 kemudian berperan sebagai pelanggan yang menunggu perintah dari Raspberry Pi melalui topik MQTT tertentu. Ketika perintah diterima, ESP32 melakukan penguraian muatan pesan, menentukan jenis perintah kendali AC, melakukan proses IR encoding sesuai merek AC, lalu mengaktifkan IR Blaster untuk mengirimkan sinyal inframerah ke AC target.

## **3.4.2 Pengembangan Modul Klasifikasi Gestur Cerdas**

Tahapan ini merupakan inti dari kecerdasan sistem yang bertujuan mengolah data visual mentah menjadi prediksi perintah yang akurat. Alur pengembangan perangkat lunak ini dirancang secara khusus untuk mengatasi tantangan variabilitas gerakan manusia serta keterbatasan sumber daya komputasi pada perangkat edge.

### **3.4.2.1 Definisi Kelas Gestur dan Pemetaan Logika Kontrol**

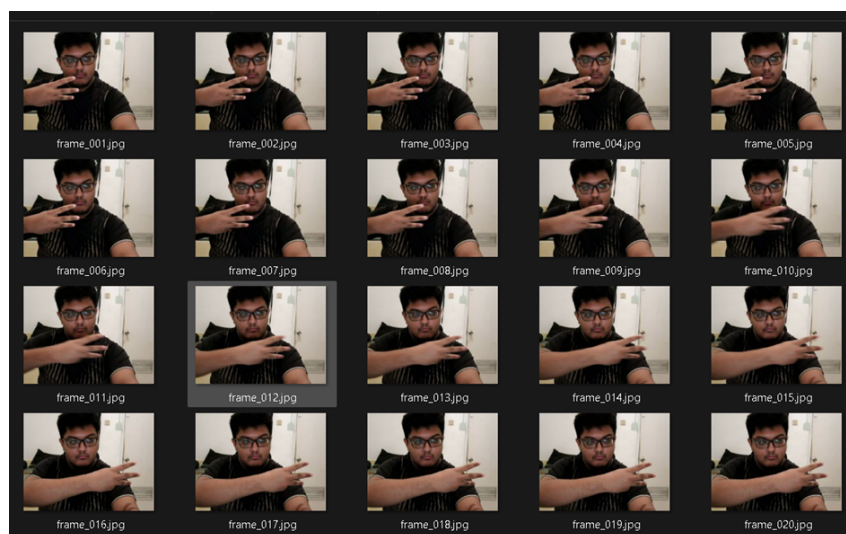
Sistem dirancang untuk mengenali enam kelas gestur spesifik. Setiap gestur dipetakan ke dalam logika kontrol yang diproses oleh mikrokontroler untuk mengubah status operasional AC. Berikut adalah rincian definisi gerakan dan logika kendalinya:

1. Kelas ThumbUp (Power ON): Pengguna melakukan gerakan mengangkat jempol ke arah atas (thumbs up) sebanyak satu kali. Model mendeteksi pola naik-turun pada vektor kecepatan vertikal tangan. Setelah hasil klasifikasi sampai di ESP32, sistem mengirimkan sinyal inframerah "Power ON". Jika AC dalam keadaan mati, unit akan menyala dan mengembalikan pengaturan terakhir (last state memory).
2. Kelas ThumbDown (Power OFF): Pengguna melakukan gerakan menurunkan jempol ke

arah bawah (thumbs down) sebanyak satu kali. Gerakan ini merupakan kebalikan vektor dari gestur ThumbUp. Setelah hasil klasifikasi sampai di ESP32, sistem mengirimkan sinyal "Power OFF" untuk memutus operasional unit AC sepenuhnya.

3. Kelas Mode (Switch Operation Mode): Pengguna melakukan gerakan melambai (waving) atau membuka telapak tangan secara statis/dinamis yang berbeda dari bentuk kepala jempol. Setelah hasil klasifikasi sampai di ESP32, mikrokontroler menjalankan fungsi logika `normalizeTwoModes`. Sistem menukar mode operasi secara bergantian (toggle). Jika mode saat ini Cool, sistem mengubah mode ke Auto dan mereset suhu ke  $25^{\circ}\text{C}$ . Jika mode saat ini Auto, sistem mengubah mode ke Cool dan mereset suhu ke  $20^{\circ}\text{C}$ .
4. Kelas Temp\_up (Increase Temperature): Pengguna menggerakkan seluruh telapak tangan dengan jempol terbuka dan jari telunjuk mengarah ke atas secara vertikal. Sistem membaca vektor perpindahan positif pada sumbu Y. Setelah hasil klasifikasi sampai di ESP32, sistem menaikkan suhu target sebesar  $+1^{\circ}\text{C}$  dari suhu saat ini. Logika firmware menerapkan batas atas (clamping), di mana perintah akan diabaikan jika suhu sudah mencapai batas maksimum  $30^{\circ}\text{C}$ .
5. Kelas Temp\_down (Decrease Temperature): Pengguna menggerakkan seluruh telapak tangan dengan jempol terbuka dan jari telunjuk mengarah ke bawah. Sistem membaca vektor perpindahan negatif pada sumbu Y. Setelah hasil klasifikasi sampai di ESP32, sistem menurunkan suhu target sebesar  $-1^{\circ}\text{C}$  dari suhu saat ini. Logika firmware menerapkan batas bawah (clamping), di mana perintah akan diabaikan jika suhu sudah mencapai batas minimum  $17^{\circ}\text{C}$ .
6. Kelas Negative (Idle State): Mencakup kondisi saat tangan diam, gerakan acak yang tidak terdefinisi, atau saat tidak ada tangan di dalam bingkai kamera. Sistem berada dalam status Idle dan tidak mengirimkan sinyal apa pun ke IR Blaster. Kelas ini sangat krusial untuk mencegah pengiriman perintah palsu (false positive) saat pengguna tidak berniat memberikan instruksi.

### 3.4.2.2 Akuisisi dan Penyiapan Dataset Gestur



Gambar 3.6: Contoh visualisasi sampel dataset gestur dinamis

Proses pengembangan model dimulai dengan akuisisi dataset gestur dinamis yang merepresentasikan enam kelas perintah fungsional. Kelas-kelas tersebut meliputi menyalakan perangkat, mematikan perangkat, menukar mode operasi, menaikkan suhu, menurunkan suhu, serta kelas negatif untuk gerakan diam atau acak. Data direkam dalam bentuk urutan video dengan panjang tetap sebanyak 20 bingkai per sampel guna menangkap konteks waktu dari setiap gerakan secara utuh. Resolusi pengambilan gambar diatur pada dimensi optimal untuk menjaga keseimbangan antara detail visual dan beban pemrosesan. Guna memperkaya keragaman data latih dan meningkatkan generalisasi model, variasi kondisi pengambilan data diterapkan secara sengaja, meliputi perbedaan pencahayaan, jarak pengguna terhadap kamera, serta variasi latar belakang.

### **3.4.2.3 Ekstraksi Fitur dan Normalisasi Data**

Data citra mentah yang telah dikumpulkan selanjutnya masuk ke tahap ekstraksi fitur atau feature engineering guna mereduksi dimensi data. Penelitian ini memanfaatkan kerangka kerja MediaPipe untuk mengekstraksi 21 titik koordinat sendi tangan (landmark) dalam ruang tiga dimensi, alih-alih memproses piksel gambar secara langsung. Sistem menghitung total 67 fitur numerik untuk setiap bingkai waktu, yang terdiri dari 63 nilai posisi koordinat relatif yang dinormalisasi terhadap pergelangan tangan agar invarian terhadap posisi tangan di layar. Selain itu, terdapat 4 fitur vektor tambahan yang merepresentasikan kecepatan dan arah pergerakan pergelangan tangan serta jari tengah. Penambahan fitur vektor gerak ini krusial untuk mempertegas sinyal dinamika gerakan, sehingga model dapat lebih mudah membedakan gestur yang memiliki kemiripan bentuk statis namun berbeda arah eksekusi.

### **3.4.2.4 Arsitektur Model dan Optimasi**

Berdasarkan data fitur yang telah diproses dan dinormalisasi, model kecerdasan buatan dibangun menggunakan arsitektur Jaringan Saraf Tiruan Berulang jenis Long Short-Term Memory (LSTM) bertumpuk. Arsitektur ini terdiri dari beberapa lapisan LSTM yang disusun secara sekuensial dengan mekanisme regularisasi Dropout pada setiap lapisan untuk mengurangi risiko overfitting selama pelatihan. Lapisan akhir menggunakan fungsi aktivasi Softmax untuk menghasilkan probabilitas kelas gestur. Setelah pelatihan selesai, dilakukan konversi model ke format TensorFlow Lite dengan teknik kuantisasi rentang dinamis, yang bertujuan mengurangi ukuran model dan mempercepat waktu inferensi pada Raspberry Pi.

## **3.4.3 Integrasi Logika Kendali dan Komunikasi Nirkabel**

Tahap integrasi bertujuan untuk menggabungkan modul klasifikasi gestur yang berjalan di Raspberry Pi dengan modul aktuator inframerah pada ESP32 agar dapat bekerja sebagai satu kesatuan sistem kendali yang utuh. Logika integrasi ini dikembangkan dalam sebuah perangkat lunak utama menggunakan bahasa Python yang mengelola alur data dari input kamera hingga eksekusi perintah secara otonom.

### **3.4.3.1 Logika Pasca-Pemrosesan dan Mesin Status**

Keluaran model LSTM berupa kelas gestur dan nilai confidence tidak langsung dieksekusi sebagai perintah. Sistem terlebih dahulu melakukan proses validasi menggunakan threshold confidence untuk memastikan bahwa prediksi model memiliki tingkat keyakinan yang cukup. Jika nilai confidence lebih kecil dari threshold, hasil prediksi diabaikan dan sistem kembali ke proses akuisisi video. Jika confidence memenuhi threshold, gestur dianggap valid dan

diteruskan ke tahap pemetaan perintah. Setiap kelas gestur dipetakan menjadi perintah kendali AC tertentu. Setelah perintah terbentuk, Raspberry Pi memublikasikan pesan MQTT ke broker EMQX. Proses ini membantu mencegah false trigger akibat gerakan tangan yang tidak jelas atau tidak disengaja.

### **3.4.3.2 Mekanisme Transmisi Perintah Terdistribusi**

Transmisi perintah antara Raspberry Pi dan ESP32 dilakukan menggunakan protokol MQTT melalui broker EMQX. Raspberry Pi berperan sebagai penerbit yang mengirimkan perintah hasil deteksi gestur, sedangkan ESP32 berperan sebagai pelanggan yang menerima perintah tersebut pada topik MQTT yang telah ditentukan. Setelah gestur tervalidasi, Raspberry Pi membentuk muatan pesan perintah yang berisi informasi seperti ID perangkat, merek AC, jenis perintah, dan nilai parameter apabila diperlukan. Muatan pesan tersebut dipublikasikan ke broker EMQX, kemudian broker meneruskan pesan ke ESP32 yang sesuai. Setelah menerima pesan, ESP32 melakukan penguraian perintah, mengubahnya menjadi sinyal inframerah melalui proses IR encoding, lalu mengirimkan sinyal tersebut ke AC menggunakan IR Blaster. Dengan mekanisme ini, jalur eksekusi perintah tetap konsisten menggunakan MQTT, sedangkan HTTP hanya digunakan untuk kebutuhan konfigurasi lokal sistem.

### **3.4.4 Evaluasi dan Validasi Sistem**

Tahap akhir dari metodologi penelitian adalah evaluasi menyeluruh untuk mengukur kinerja sistem. Pengujian dirancang dalam beberapa skenario percobaan untuk memastikan bahwa setiap bagian, baik pengolahan gambar maupun pengiriman sinyal nirkabel, memenuhi standar kecepatan dan keandalan yang diharapkan dalam kondisi nyata.

#### **3.4.4.1 Skenario Pengujian Variabel Lingkungan**

Untuk memastikan keandalan sistem, pengujian dilakukan dengan mengubah kondisi lingkungan pada dua bagian utama, yaitu bagian masukan kamera dan keluaran sinyal ke AC.

1. Pengujian Subsistem Visi: pengujian pada bagian penglihatan bertujuan untuk mengetahui batas kemampuan kamera dan sistem dalam mengenali gerakan tangan. Variabel pertama yang diuji adalah jarak antara tangan dan kamera untuk menentukan area efektif di mana gestur masih dapat dikenali dengan tepat. Jarak pengujian dilakukan pada 1 meter, 1,5 meter, 2 meter, dan 3 meter. Variabel kedua yang diuji adalah kondisi cahaya ruangan untuk menilai kemampuan sistem menyesuaikan diri terhadap perubahan intensitas cahaya. Pengujian dilakukan pada kondisi terang  $\pm 230$  lux, redup  $\pm 100$  lux, dan gelap  $\pm 55$  lux untuk memastikan sistem tetap dapat mendeteksi tangan pada berbagai kondisi pencahayaan.
2. Pengujian Subsistem Aktuasi: pengujian pada bagian penggerak bertujuan mengukur keandalan alat pemancar dalam mengirimkan sinyal perintah ke AC. Variabel pertama yang diuji adalah keberhasilan pancaran sinyal dari alat ke sensor penerima pada AC pada beberapa posisi, yaitu di bawah AC, 2 meter, 4,3 meter, dan 6,9 meter. Pengujian juga dilakukan pada variasi arah casing dan desain casing untuk mengetahui pengaruh orientasi alat terhadap penerimaan sinyal. Variabel kedua yang diuji adalah pengaruh intensitas cahaya ruangan terhadap penerimaan sinyal pada AC. Pengujian dilakukan pada kondisi terang  $\pm 230$  lux, redup  $\pm 100$  lux, dan gelap  $\pm 55$  lux untuk memastikan sinyal perintah tetap dapat diterima dengan baik pada berbagai kondisi ruangan.

### **3.4.4.2 Pengukuran Metrik Kinerja**

Kuantifikasi kinerja sistem dilakukan menggunakan dua metrik utama yang saling berkaitan. Pertama, akurasi deteksi diukur menggunakan Confusion Matrix untuk memetakan tingkat keberhasilan klasifikasi pada setiap kelas gestur serta mengidentifikasi potensi kebingungan model antar-kelas. Kedua, latensi sistem diukur secara end-to-end, mulai dari saat gestur selesai dilakukan hingga sinyal inframerah dipancarkan oleh ESP32. Pengukuran latensi ini mencakup waktu inferensi model, waktu pemrosesan logika, dan waktu transmisi jaringan WiFi, guna memastikan respons sistem terasa instan dan natural bagi pengguna.

## **BAB IV**

### **PENGUJIAN DAN ANALISIS**

Pada bab pengujian ini dibahas hasil pengujian dan analisis dari penelitian yang telah dilakukan. Pembahasan disusun berdasarkan metodologi yang telah dijelaskan pada bab sebelumnya. Bab ini berfokus pada skenario pengujian, hasil yang diperoleh dari setiap skenario, serta analisis menyeluruh terhadap kinerja sistem.

#### **4.1 Hasil Pengumpulan dan Pra-pemrosesan Dataset**

Tahap pengumpulan dan pra-pemrosesan dataset dilakukan untuk menghasilkan data gestur tangan yang siap digunakan dalam proses pelatihan model LSTM. Dataset yang digunakan pada penelitian ini merupakan dataset gestur dinamis, sehingga setiap sampel tidak hanya merepresentasikan bentuk tangan pada satu frame, tetapi juga perubahan posisi tangan dalam beberapa frame secara berurutan. Setiap data gestur direkam menggunakan kamera, kemudian diproses menggunakan MediaPipe untuk memperoleh titik landmark tangan. Hasil ekstraksi landmark tersebut selanjutnya diubah menjadi fitur numerik yang merepresentasikan posisi dan pergerakan tangan. Data hasil pra-pemrosesan kemudian disusun menjadi sekuens frame agar dapat digunakan sebagai input model LSTM.

##### **4.1.1 Struktur dan Distribusi Dataset**

Data awal yang digunakan pada penelitian ini berupa kumpulan raw sample untuk setiap kelas gestur. Setiap frame terlebih dahulu melalui proses pra-pemrosesan menggunakan MediaPipe Hands untuk mengekstraksi titik landmark tangan. Hasil ekstraksi tersebut kemudian diubah menjadi data numerik yang merepresentasikan posisi serta arah pergerakan tangan. Setiap satu sampel gestur dinamis direpresentasikan sebagai urutan data temporal atau sequence. Pada penelitian ini, satu sampel terdiri dari 20 frame berturut-turut. Setiap frame memiliki 67 fitur, yang terdiri dari 63 fitur landmark tangan dan 4 fitur arah gerak. Fitur landmark diperoleh dari 21 titik landmark tangan dengan masing-masing titik memiliki koordinat tiga dimensi, yaitu  $x$ ,  $y$ , dan  $z$ . Sementara itu, 4 fitur arah gerak terdiri dari  $d\_wrist\_x$ ,  $d\_wrist\_y$ ,  $d\_mid\_x$ , dan  $d\_mid\_y$ , yang digunakan untuk merepresentasikan perubahan posisi pergelangan tangan dan titik tengah tangan antar-frame. Dengan struktur tersebut, bentuk input untuk satu sampel data adalah matriks berukuran  $20 \times 67$ .

Dataset hasil pra-pemrosesan disimpan dalam format `.npz` agar lebih mudah digunakan pada proses pelatihan model. Format ini memungkinkan data fitur dan label disimpan secara terstruktur dalam satu berkas, sehingga proses pemuatan dataset menjadi lebih efisien. Dataset kemudian dibagi menjadi dua bagian, yaitu data latih dan data uji. Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk mengevaluasi kemampuan model dalam mengenali gestur yang belum pernah dilihat selama proses pelatihan. Total raw sample yang berhasil dikumpulkan berjumlah 2.740 sampel, yang terbagi ke dalam enam kelas gestur, yaitu Mode, Negative, Temp\_down, Temp\_up, ThumbDown, dan ThumbUp. Distribusi jumlah raw sample pada setiap kelas ditunjukkan pada Tabel 4.1. Pada tahap evaluasi model, data uji yang

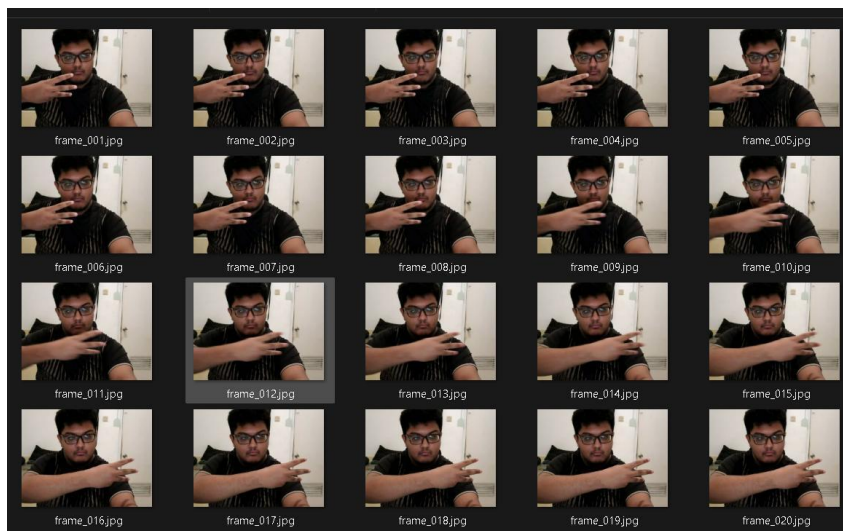
digunakan pada classification report berjumlah 501 sekuens.

Tabel 4.1: Distribusi jumlah raw sample pada setiap kelas gestur

Kelas Gestur	Raw Sample
Mode	416
Negative	468
Temp_down	432
Temp_up	424
ThumbDown	533
ThumbUp	467
Total	2740

Struktur sequence tersebut membuat model mampu mempelajari pola perubahan gestur secara temporal dan menghasilkan klasifikasi yang lebih akurat.

#### 4.1.2 Visualisasi Perubahan Gestur Dinamis Pada Dataset



Gambar 4.1: Visualisasi urutan temporal gestur dinamis dari frame 1 hingga frame 20

Berbeda dengan gestur statis yang hanya mengandalkan satu citra tunggal, dataset dalam penelitian ini bersifat dinamis. Artinya, informasi utama tidak hanya berasal dari bentuk tangan pada satu frame, tetapi juga dari perubahan posisi tangan terhadap waktu. Oleh karena itu, setiap sampel gestur direpresentasikan sebagai urutan temporal sepanjang 20 frame seperti ditunjukkan pada Gambar 4.1. Representasi temporal ini penting karena beberapa kelas gestur memiliki bentuk tangan yang mirip, tetapi dibedakan oleh arah dan pola gerakannya.

#### 4.1.3 Variasi Pengambilan Data

Pengambilan data dilakukan dengan melibatkan 6 orang subjek, yang terdiri dari 3 laki-laki dan 3 perempuan. Variasi subjek diperlukan agar dataset tidak hanya merepresentasikan satu bentuk tangan atau satu gaya gerakan tertentu. Selain variasi subjek, proses pengambilan data juga memperhatikan variasi posisi tangan, orientasi telapak tangan, jarak terhadap kamera,

dan kondisi pencahayaan. Variasi tersebut bertujuan meningkatkan kemampuan generalisasi model ketika sistem digunakan oleh pengguna yang berbeda.

Secara umum, variasi data yang dikumpulkan mencakup perbedaan bentuk tangan, ukuran tangan, kecepatan gerakan, posisi tangan di dalam bingkai kamera, serta perubahan arah hadap tangan. Dengan adanya variasi tersebut, model diharapkan tidak hanya menghafal pola dari satu kondisi ideal, tetapi mampu mengenali gestur pada kondisi penggunaan yang lebih realistis.

## 4.2 Skenario Pengujian

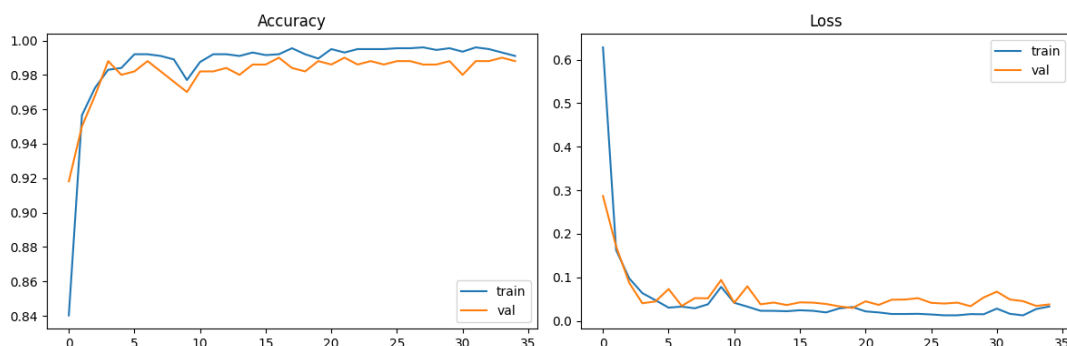
Pengujian dilakukan untuk mengevaluasi kinerja model klasifikasi gestur dan kinerja sistem kendali secara keseluruhan. Skenario pengujian disusun menjadi beberapa kelompok. Pertama, pengujian performa model dilakukan pada model Keras dan model TensorFlow Lite (TFLite) menggunakan data uji sebanyak 501 sekuens. Kedua, pengujian jarak dilakukan pada variasi 1 meter, 1,5 meter, 2 meter, dan 3 meter dengan total 240 data uji. Ketiga, pengujian pencahayaan dilakukan pada kondisi terang  $\pm 230$  lux, redup  $\pm 100$  lux, dan gelap  $\pm 55$  lux dengan total 180 data uji. Keempat, pengujian variasi subjek dilakukan untuk melihat kemampuan generalisasi model terhadap pengguna yang berbeda; penelitian ini melibatkan 6 orang subjek yang terdiri dari 3 laki-laki dan 3 perempuan. Kelima, pengujian aktuasi dilakukan pada IR Blaster untuk melihat keberhasilan pengiriman sinyal ke AC pada variasi jarak, arah perangkat, desain casing, dan pencahayaan.

Metrik utama yang digunakan dalam evaluasi model adalah accuracy, precision, recall, F1-score, dan confusion matrix. Accuracy digunakan untuk melihat persentase prediksi benar secara keseluruhan. Precision menunjukkan ketepatan prediksi pada suatu kelas, recall menunjukkan kemampuan model menangkap seluruh data pada kelas tersebut, sedangkan F1-score menunjukkan keseimbangan antara precision dan recall. Confusion matrix digunakan untuk melihat pola kesalahan antar-kelas secara lebih rinci.

## 4.3 Pengujian Performa Model

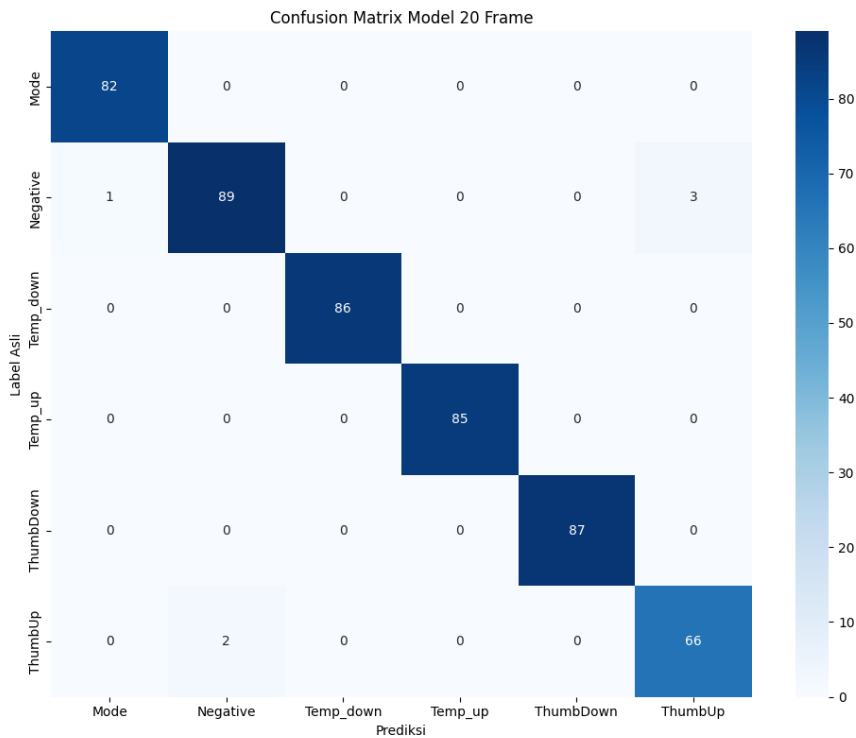
Pengujian performa model dilakukan untuk mengetahui kemampuan model LSTM dalam mengenali enam kelas gestur tangan. Evaluasi dilakukan pada model Keras hasil pelatihan dan model TensorFlow Lite hasil konversi. Kedua model diuji menggunakan data uji yang sama, yaitu 501 sekuens, sehingga hasilnya dapat dibandingkan secara langsung.

### 4.3.1 Model Keras



Gambar 4.2: Grafik accuracy dan loss model Keras 20 frame

Gambar 4.2 menunjukkan bahwa proses pelatihan berlangsung stabil. Accuracy data latih meningkat tajam pada epoch awal, kemudian berada pada rentang tinggi mendekati 0,99. Accuracy validasi juga mengikuti pola yang sama dan stabil di sekitar 0,98 hingga 0,99. Pada grafik loss, nilai loss data latih dan validasi turun signifikan pada epoch awal, kemudian berada pada nilai rendah. Pola tersebut menunjukkan bahwa model mampu mempelajari pola gestur tanpa indikasi overfitting yang besar, karena jarak antara performa training dan validation relatif kecil.



Gambar 4.3: Confusion matrix model Keras 20 frame

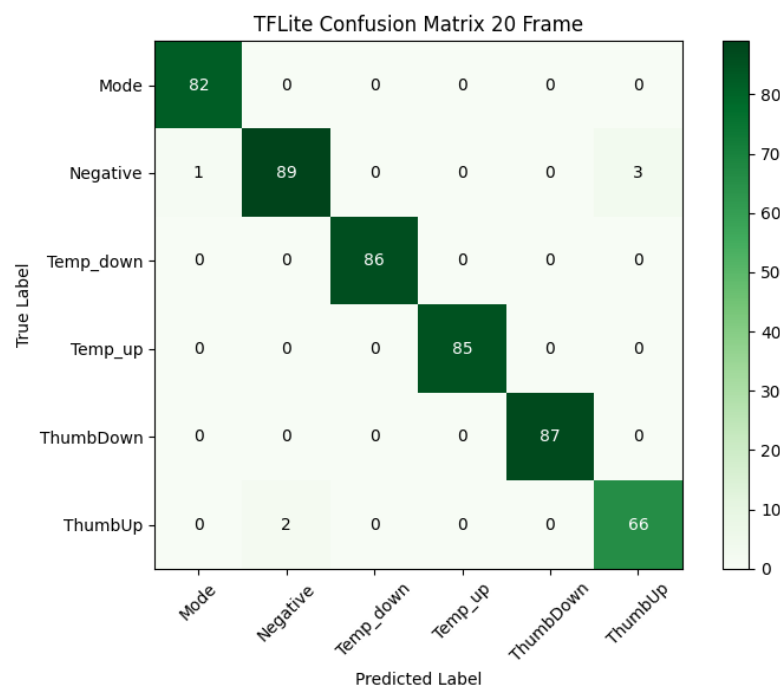
Model Keras memperoleh accuracy sebesar 98,80%, dengan precision, recall, dan F1-score keseluruhan sebesar 98,80%. Berdasarkan confusion matrix pada Gambar 4.3, kelas Temp\_down, Temp\_up, dan ThumbDown dapat dikenali sempurna pada data uji. Kesalahan utama terjadi pada kelas Negative dan ThumbUp. Pada kelas Negative, terdapat 1 data yang diprediksi sebagai Mode dan 3 data yang diprediksi sebagai ThumbUp. Pada kelas ThumbUp, terdapat 2 data yang diprediksi sebagai Negative. Hal ini menunjukkan bahwa bentuk atau dinamika ThumbUp masih menjadi sumber kebingungan utama, terutama ketika gerakan memiliki kemiripan dengan kondisi non-perintah.

Tabel 4.2: Hasil evaluasi model Keras 20 frame

Kelas	Precision	Recall	F1-score	Support
Mode	0.99	1.00	0.99	82
Negative	0.98	0.96	0.97	93
Temp_down	1.00	1.00	1.00	86
Temp_up	1.00	1.00	1.00	85
ThumbDown	1.00	1.00	1.00	87
ThumbUp	0.96	0.97	0.96	68
Accuracy			0.99	501
Macro Avg	0.99	0.99	0.99	501
Weighted Avg	0.99	0.99	0.99	501

Secara keseluruhan, model Keras menunjukkan performa yang sangat baik untuk klasifikasi gestur tangan dinamis. Nilai precision, recall, dan F1-score yang tinggi menunjukkan bahwa model mampu menjaga keseimbangan performa pada hampir seluruh kelas. Kelas yang masih perlu diperkuat adalah Negative dan ThumbUp karena keduanya menjadi sumber kesalahan terbesar pada data uji.

### 4.3.2 Model TFLite



Gambar 4.4: Confusion matrix model TFLite 20 frame

Model TensorFlow Lite atau TFLite merupakan hasil konversi dari model Keras. Konversi ini dilakukan agar model dapat dijalankan secara lebih ringan dan efisien pada perangkat edge seperti Raspberry Pi. Pengujian model TFLite dilakukan menggunakan data uji yang sama dengan model Keras. Berdasarkan hasil pengujian, model TFLite memperoleh accuracy sebesar

98,80%, sama seperti model Keras. Hasil ini menunjukkan bahwa proses konversi ke format TFLite tidak menurunkan performa klasifikasi.

Berdasarkan confusion matrix pada Gambar 4.4, pola klasifikasi model TFLite sama dengan model Keras. Kelas Temp\_down, Temp\_up, dan ThumbDown tetap diklasifikasikan dengan sempurna, sedangkan kesalahan masih berpusat pada kelas Negative dan ThumbUp. Tidak terdapat tambahan pola kesalahan baru setelah model dikonversi ke TFLite.

Tabel 4.3: Hasil evaluasi model TFLite 20 frame

Kelas	Precision	Recall	F1-score	Support
Mode	0.99	1.00	0.99	82
Negative	0.98	0.96	0.97	93
Temp_down	1.00	1.00	1.00	86
Temp_up	1.00	1.00	1.00	85
ThumbDown	1.00	1.00	1.00	87
ThumbUp	0.96	0.97	0.96	68
Accuracy			0.99	501
Macro Avg	0.99	0.99	0.99	501
Weighted Avg	0.99	0.99	0.99	501

Hasil ini menunjukkan bahwa model TFLite tetap mempertahankan performa model Keras. Dengan akurasi yang sama dan pola kesalahan yang tidak berubah, model TFLite layak digunakan sebagai model inferensi pada Raspberry Pi. Perbaikan yang dapat dilakukan pada tahap selanjutnya adalah menambah variasi data untuk kelas ThumbUp dan Negative, terutama pada variasi sudut tangan, posisi tangan, kecepatan gerakan, dan kondisi pencahayaan.

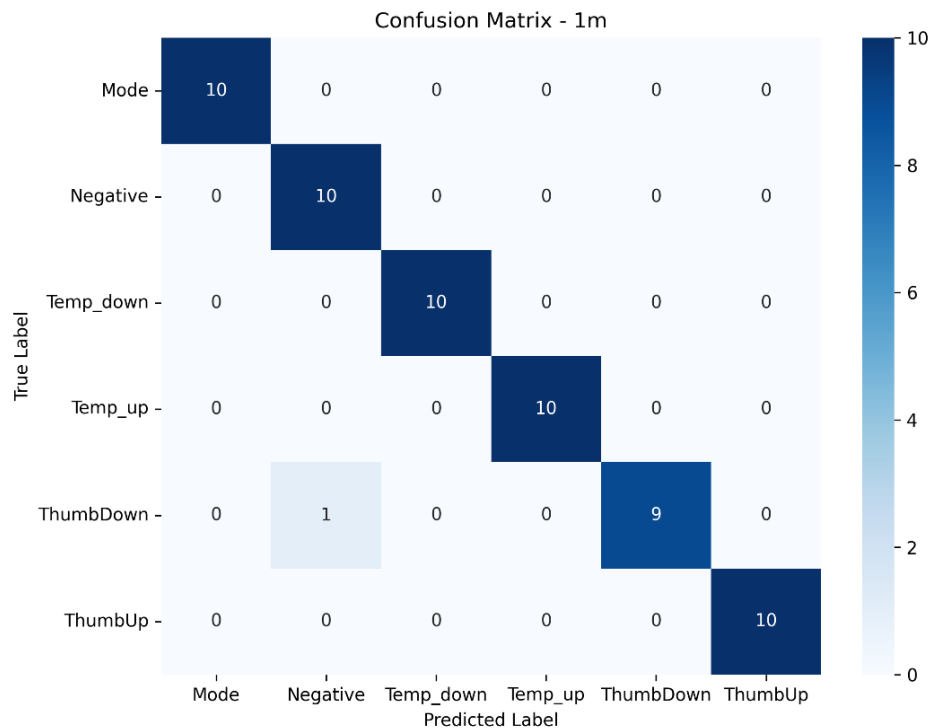
#### 4.4 Pengujian Performa Model Terhadap Jarak

Pengujian performa model terhadap jarak dilakukan untuk mengetahui pengaruh jarak antara pengguna dan kamera terhadap kemampuan model mengenali gestur tangan. Pengujian dilakukan pada empat variasi jarak, yaitu 1 meter, 1,5 meter, 2 meter, dan 3 meter. Pada setiap jarak digunakan 60 data uji, yang terdiri dari enam kelas gestur dengan 10 data untuk setiap kelas. Dengan demikian, total data uji pada skenario jarak adalah 240 data. Rekapitulasi hasil pengujian ditunjukkan pada Tabel 4.4.

Tabel 4.4: Rekapitulasi hasil pengujian gestur terhadap jarak

Jarak	Akurasi	Precision	Recall	F1-score
1 meter	0.97	0.97	0.97	0.97
1,5 meter	0.95	0.96	0.95	0.95
2 meter	0.95	0.96	0.95	0.95
3 meter	0.93	0.94	0.93	0.93

#### 4.4.1 Pengujian Gestur pada Jarak 1 Meter



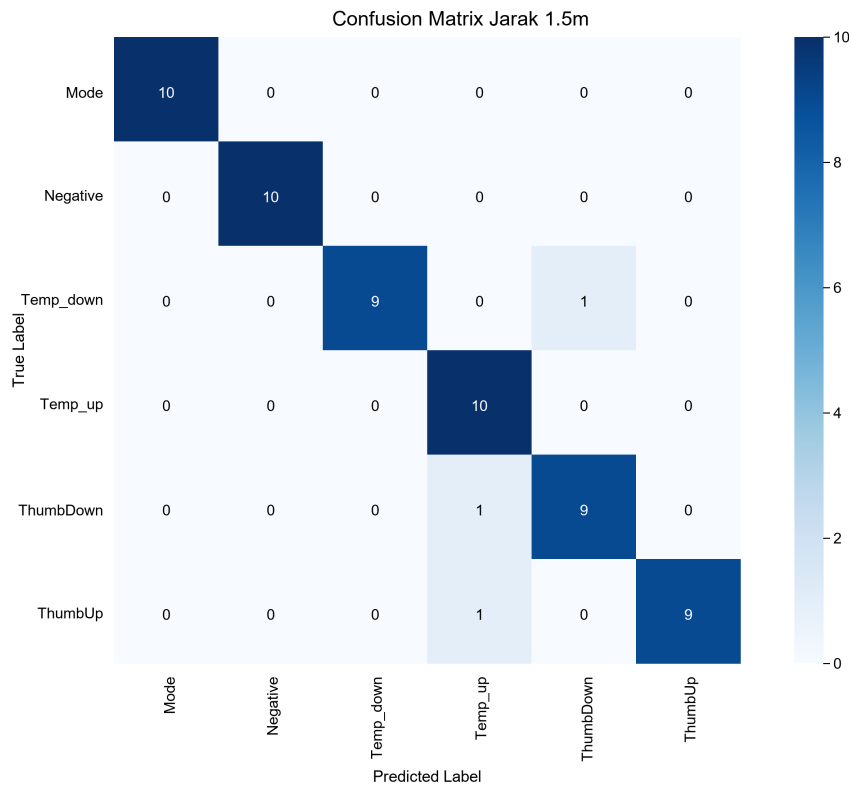
Gambar 4.5: Confusion matrix pengujian gestur pada jarak 1 meter

Tabel 4.5: Hasil klasifikasi model pada jarak 1 meter

Kelas	Precision	Recall	F1-score	Support
Mode	1.00	0.90	0.95	10
Negative	0.91	1.00	0.95	10
Temp_down	0.91	1.00	0.95	10
Temp_up	1.00	0.90	0.95	10
ThumbDown	1.00	1.00	1.00	10
ThumbUp	1.00	1.00	1.00	10
Accuracy			0.97	60
Macro Avg	0.97	0.97	0.97	60
Weighted Avg	0.97	0.97	0.97	60

Berdasarkan Gambar 4.5 dan Tabel 4.5, model memperoleh akurasi sebesar 96,67% pada jarak 1 meter. Kesalahan terjadi pada kelas Mode yang diprediksi sebagai Negative dan kelas Temp\_up yang diprediksi sebagai Temp\_down. Kelas Negative, Temp\_down, ThumbDown, dan ThumbUp masih memiliki recall 1,00. Hasil ini menunjukkan bahwa pada jarak dekat, landmark tangan masih dapat ditangkap dengan jelas, tetapi arah gerak vertikal tetap perlu dibedakan secara stabil agar Temp\_up tidak tertukar dengan Temp\_down.

#### 4.4.2 Pengujian Gestur pada Jarak 1,5 Meter



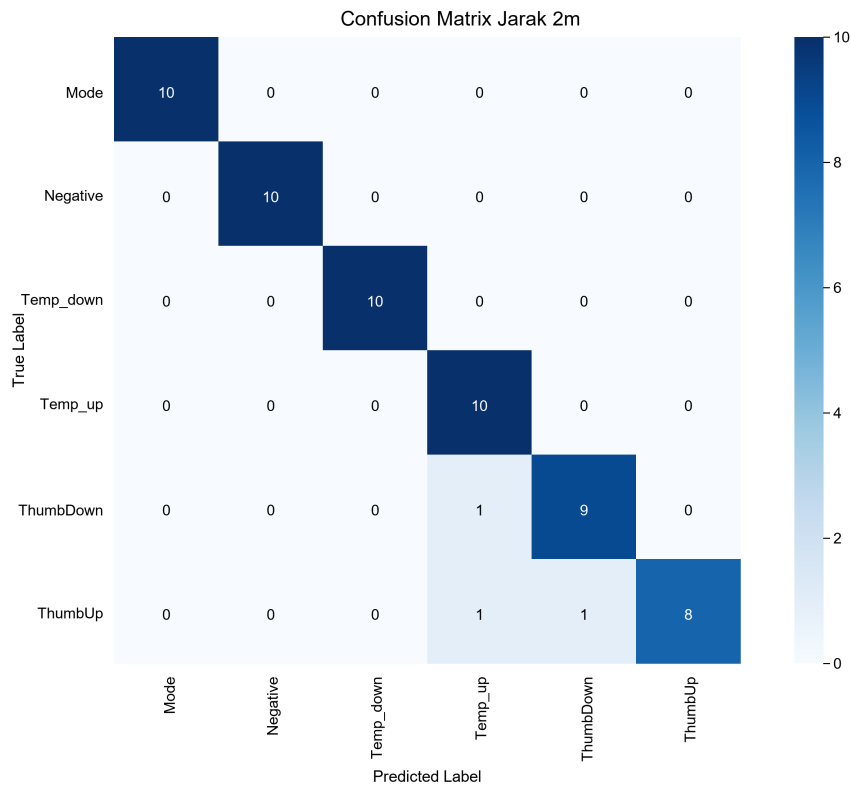
Gambar 4.6: Confusion matrix pengujian gestur pada jarak 1,5 meter

Tabel 4.6: Hasil klasifikasi model pada jarak 1,5 meter

Kelas	Precision	Recall	F1-score	Support
Mode	1.00	1.00	1.00	10
Negative	1.00	1.00	1.00	10
Temp_down	1.00	0.90	0.95	10
Temp_up	0.83	1.00	0.91	10
ThumbDown	0.90	0.90	0.90	10
ThumbUp	1.00	0.90	0.95	10
Accuracy			0.95	60
Macro Avg	0.96	0.95	0.95	60
Weighted Avg	0.96	0.95	0.95	60

Pada jarak 1,5 meter, model memperoleh akurasi sebesar 95,00%. Kelas Mode dan Negative masih dikenali sempurna. Kesalahan terjadi pada Temp\_down yang diprediksi sebagai ThumbDown, ThumbDown yang diprediksi sebagai Temp\_up, serta ThumbUp yang diprediksi sebagai Temp\_up. Pola ini menunjukkan bahwa ketika jarak bertambah, perubahan ukuran tangan pada citra mulai memengaruhi pembacaan arah gerak dan bentuk jari, terutama pada kelas yang memiliki kemiripan orientasi tangan.

### 4.4.3 Pengujian Gestur pada Jarak 2 Meter



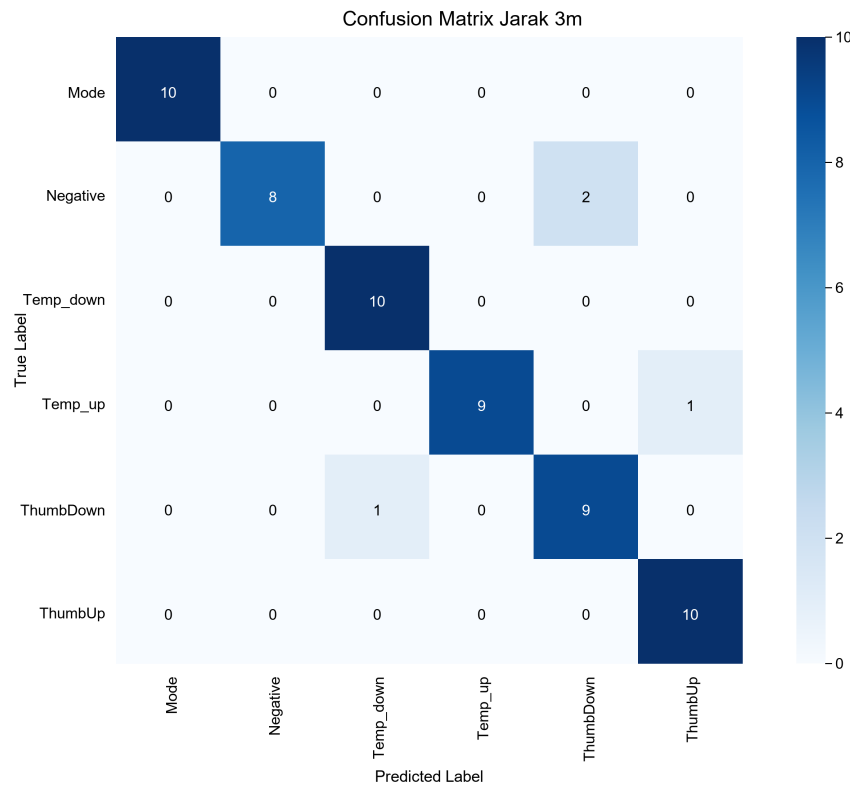
Gambar 4.7: Confusion matrix pengujian gestur pada jarak 2 meter

Tabel 4.7: Hasil klasifikasi model pada jarak 2 meter

Kelas	Precision	Recall	F1-score	Support
Mode	1.00	1.00	1.00	10
Negative	1.00	1.00	1.00	10
Temp_down	1.00	1.00	1.00	10
Temp_up	0.83	1.00	0.91	10
ThumbDown	0.90	0.90	0.90	10
ThumbUp	1.00	0.80	0.89	10
Accuracy			0.95	60
Macro Avg	0.96	0.95	0.95	60
Weighted Avg	0.96	0.95	0.95	60

Pada jarak 2 meter, model memperoleh akurasi sebesar 95,00%. Kelas Mode, Negative, Temp\_down, dan Temp\_up memiliki recall 1,00. Kesalahan utama terjadi pada kelas ThumbDown dan ThumbUp. Satu data ThumbDown diprediksi sebagai Temp\_up, sedangkan dua data ThumbUp masing-masing diprediksi sebagai Temp\_up dan ThumbDown. Hal ini menunjukkan bahwa gestur berbasis ibu jari mulai lebih sensitif terhadap perubahan skala dan orientasi tangan pada jarak menengah.

#### 4.4.4 Pengujian Gestur pada Jarak 3 Meter



Gambar 4.8: Confusion matrix pengujian gestur pada jarak 3 meter

Tabel 4.8: Hasil klasifikasi model pada jarak 3 meter

Kelas	Precision	Recall	F1-score	Support
Mode	1.00	1.00	1.00	10
Negative	1.00	0.80	0.89	10
Temp_down	0.91	1.00	0.95	10
Temp_up	1.00	0.90	0.95	10
ThumbDown	0.82	0.90	0.86	10
ThumbUp	0.91	1.00	0.95	10
Accuracy			0.93	60
Macro Avg	0.94	0.93	0.93	60
Weighted Avg	0.94	0.93	0.93	60

Pada jarak 3 meter, model memperoleh akurasi sebesar 93,33%, yaitu nilai terendah pada skenario jarak. Kesalahan yang paling menonjol adalah 2 data Negative yang diprediksi sebagai ThumbDown. Selain itu, terdapat kesalahan Temp\_up menjadi ThumbUp dan ThumbDown menjadi Temp\_down. Hasil ini menunjukkan bahwa jarak 3 meter masih dapat digunakan, tetapi risiko kesalahan meningkat karena ukuran tangan pada citra semakin kecil dan landmark menjadi lebih sulit diekstraksi secara stabil.

#### 4.4.5 Analisis Keseluruhan Pengaruh Jarak

Secara keseluruhan, pengujian jarak menghasilkan akurasi gabungan 95,83% dari 240 data uji. Performa terbaik terjadi pada jarak 1 meter dengan akurasi 96,67%, sedangkan performa terendah terjadi pada jarak 3 meter dengan akurasi 93,33%. Penurunan performa tidak bersifat ekstrem, tetapi pola kesalahan menunjukkan bahwa jarak yang lebih jauh meningkatkan risiko kekeliruan pada kelas berbasis arah dan ibu jari, seperti Temp\_up, Temp\_down, ThumbDown, dan ThumbUp. Pada jarak 3 meter, muncul pula kesalahan Negative yang diprediksi sebagai ThumbDown. Oleh karena itu, penggunaan threshold confidence, smoothing prediksi, dan batas jarak operasional tetap penting untuk mencegah false trigger pada implementasi nyata.

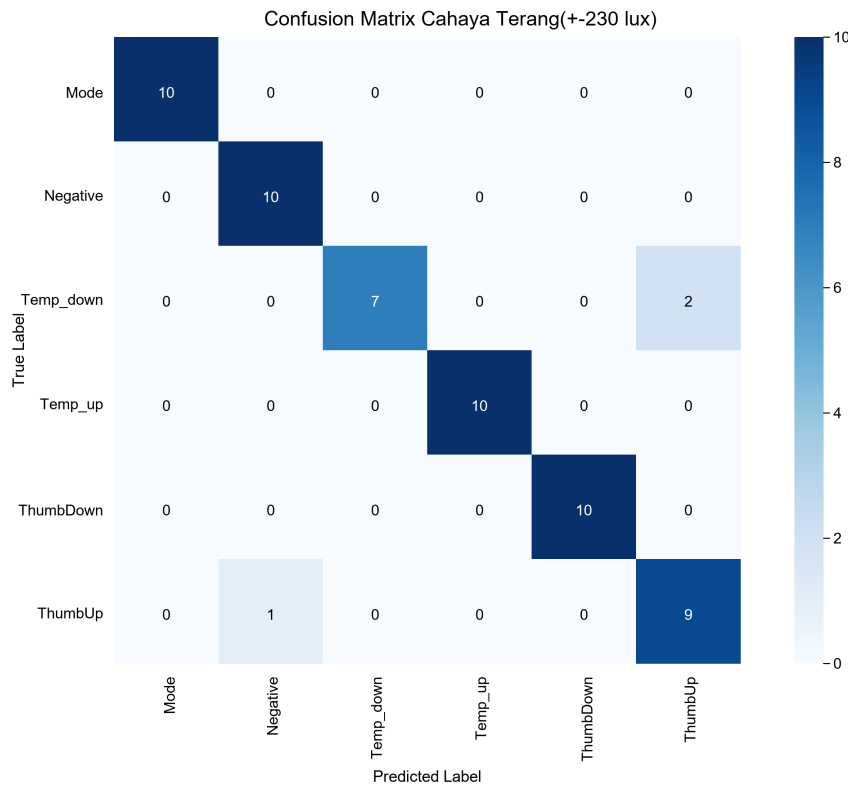
#### 4.5 Pengujian Performa Model Terhadap Pencahayaan

Pengujian performa model terhadap pencahayaan dilakukan untuk mengetahui pengaruh intensitas cahaya ruangan terhadap kestabilan deteksi landmark dan klasifikasi gestur. Pengujian dilakukan pada tiga kondisi, yaitu terang  $\pm 230$  lux, redup  $\pm 100$  lux, dan gelap  $\pm 55$  lux. Total data uji pada skenario pencahayaan adalah 180 data, dengan jumlah data pada setiap kondisi mengikuti hasil pengambilan data yang tersedia. Rekapitulasi hasil pengujian ditunjukkan pada Tabel 4.9.

Tabel 4.9: Rekapitulasi hasil pengujian gestur terhadap pencahayaan

Kondisi	Akurasi	Precision	Recall	F1-score	Support
Terang $\pm 230$ lux	0.95	0.95	0.95	0.95	59
Redup $\pm 100$ lux	0.92	0.92	0.92	0.92	61
Gelap $\pm 55$ lux	0.82	0.82	0.82	0.81	60

### 4.5.1 Pengujian Gestur pada Kondisi Terang 230 lux



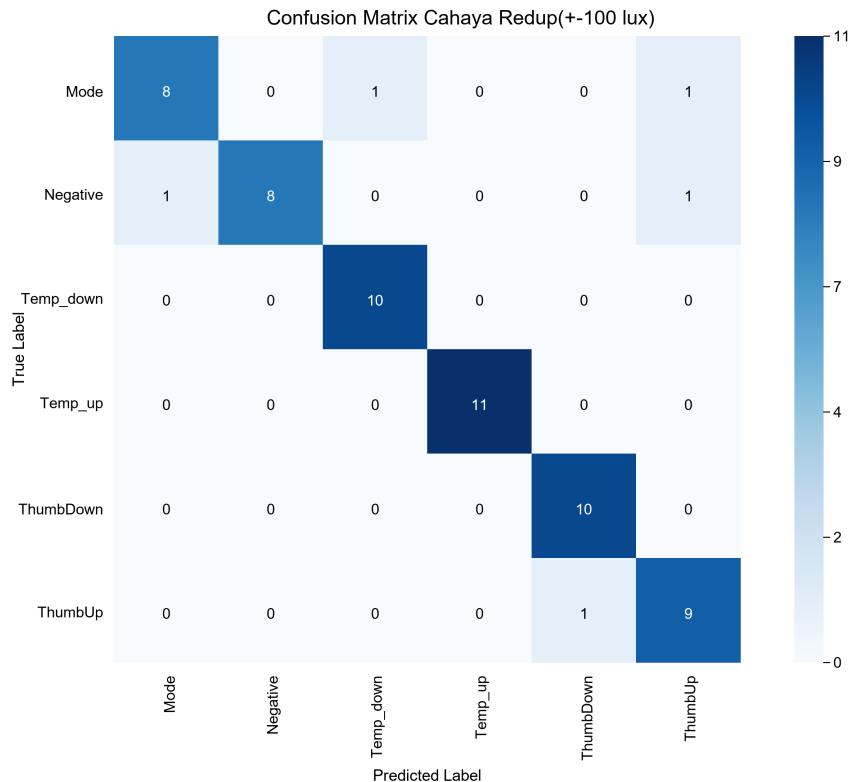
Gambar 4.9: Confusion matrix pengujian gestur pada kondisi terang 230 lux

Tabel 4.10: Hasil klasifikasi model pada kondisi terang  $\pm 230$  lux

Kelas	Precision	Recall	F1-score	Support
Mode	1.00	1.00	1.00	10
Negative	0.91	1.00	0.95	10
Temp_down	1.00	0.78	0.88	9
Temp_up	1.00	1.00	1.00	10
ThumbDown	1.00	1.00	1.00	10
ThumbUp	0.82	0.90	0.86	10
Accuracy			0.95	59
Macro Avg	0.95	0.95	0.95	59
Weighted Avg	0.95	0.95	0.95	59

Pada kondisi terang  $\pm 230$  lux, model memperoleh akurasi sebesar 94,92% dari 59 data uji. Kelas Mode, Negative, Temp\_up, dan ThumbDown dapat dikenali dengan recall 1,00. Kesalahan utama terjadi pada kelas Temp\_down, yaitu 2 data diprediksi sebagai ThumbUp, serta pada kelas ThumbUp, yaitu 1 data diprediksi sebagai Negative. Secara umum, kondisi terang memberikan kualitas citra yang baik sehingga proses ekstraksi landmark berjalan stabil.

## 4.5.2 Pengujian Gestur pada Kondisi Redup 100 lux



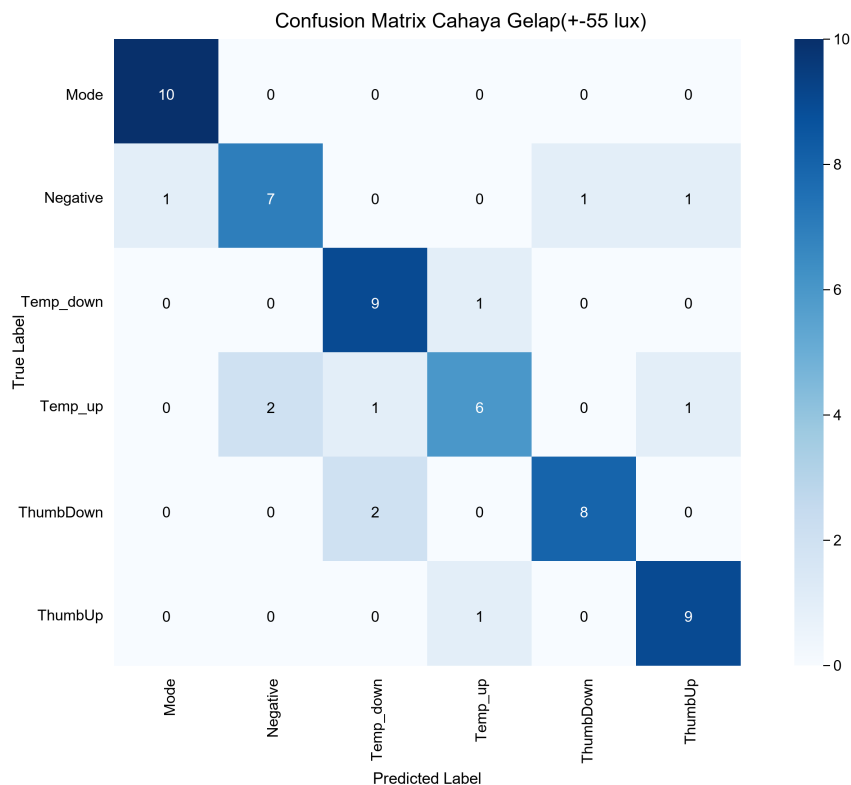
Gambar 4.10: Confusion matrix pengujian gestur pada kondisi redup 100 lux

Tabel 4.11: Hasil klasifikasi model pada kondisi redup  $\pm 100$  lux

Kelas	Precision	Recall	F1-score	Support
Mode	0.89	0.80	0.84	10
Negative	1.00	0.80	0.89	10
Temp_down	0.91	1.00	0.95	10
Temp_up	1.00	1.00	1.00	11
ThumbDown	0.91	1.00	0.95	10
ThumbUp	0.82	0.90	0.86	10
Accuracy			0.92	61
Macro Avg	0.92	0.92	0.92	61
Weighted Avg	0.92	0.92	0.92	61

Pada kondisi redup  $\pm 100$  lux, model memperoleh akurasi sebesar 91,80% dari 61 data uji. Kelas Temp\_down, Temp\_up, dan ThumbDown masih memiliki recall 1,00. Penurunan performa terutama terjadi pada kelas Mode dan Negative yang masing-masing memiliki recall 0,80. Beberapa data Mode diprediksi sebagai Temp\_down dan ThumbUp, sedangkan beberapa data Negative diprediksi sebagai Mode dan ThumbUp. Hal ini menunjukkan bahwa penurunan pencahayaan mulai memengaruhi kestabilan bentuk landmark dan meningkatkan risiko kesalahan antar-kelas.

### 4.5.3 Pengujian Gestur pada Kondisi Gelap 55 lux



Gambar 4.11: Confusion matrix pengujian gestur pada kondisi gelap 55 lux

Tabel 4.12: Hasil klasifikasi model pada kondisi gelap  $\pm 55$  lux

Kelas	Precision	Recall	F1-score	Support
Mode	0.91	1.00	0.95	10
Negative	0.78	0.70	0.74	10
Temp_down	0.75	0.90	0.82	10
Temp_up	0.75	0.60	0.67	10
ThumbDown	0.89	0.80	0.84	10
ThumbUp	0.82	0.90	0.86	10
Accuracy			0.82	60
Macro Avg	0.82	0.82	0.81	60
Weighted Avg	0.82	0.82	0.81	60

Pada kondisi gelap  $\pm 55$  lux, model memperoleh akurasi sebesar 81,67%, yaitu hasil terendah pada skenario pencahayaan. Penurunan paling terlihat pada kelas Temp\_up dengan recall 0,60 dan kelas Negative dengan recall 0,70. Kelas Temp\_up mengalami kesalahan ke Negative, Temp\_down, dan ThumbUp, sedangkan kelas Negative mengalami kesalahan ke Mode, ThumbDown, dan ThumbUp. Kondisi ini menunjukkan bahwa pencahayaan rendah dapat menurunkan kualitas citra, mengurangi kejelasan landmark tangan, dan meningkatkan kesalahan antar-gestur.

#### 4.5.4 Analisis Keseluruhan Pengaruh Pencahayaan

Secara keseluruhan, pengujian pencahayaan menghasilkan akurasi gabungan 89,44% dari 180 data uji. Performa menurun seiring berkurangnya intensitas cahaya, dari 94,92% pada kondisi terang, menjadi 91,80% pada kondisi redup, dan 81,67% pada kondisi gelap. Penurunan paling besar terjadi pada kondisi gelap karena citra tangan menjadi kurang jelas sehingga deteksi landmark lebih tidak stabil. Berbeda dengan pengujian jarak yang masih berada di atas 93%, pengujian pencahayaan menunjukkan bahwa cahaya ruangan menjadi faktor lingkungan yang lebih kritis bagi sistem berbasis kamera. Oleh karena itu, penggunaan sistem sebaiknya dilakukan pada ruangan dengan pencahayaan cukup, atau dilengkapi pencahayaan tambahan agar proses deteksi tangan lebih stabil.

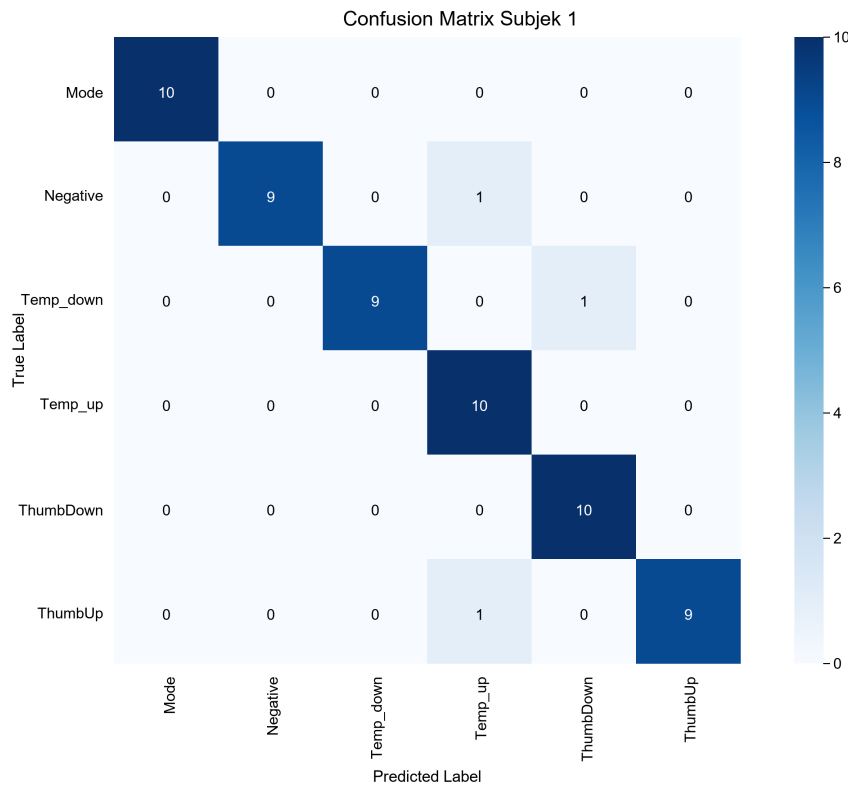
#### 4.6 Pengujian Terhadap Beberapa Subjek yang Berbeda

Pengujian terhadap variasi subjek dilakukan untuk mengetahui kemampuan generalisasi model terhadap pengguna yang berbeda. Secara keseluruhan, penelitian ini melibatkan 6 orang subjek yang terdiri dari 3 laki-laki dan 3 perempuan. Hasil pengujian subjek pada bagian ini disajikan dalam tiga set pengujian, yaitu Subjek 1, Subjek 2, dan Subjek 3, dengan masing-masing set berisi 60 data uji. Total data uji pada skenario subjek adalah 180 data. Rekapitulasi hasil pengujian ditunjukkan pada Tabel 4.13.

Tabel 4.13: Rekapitulasi hasil pengujian gestur terhadap variasi subjek

Set Pengujian	Akurasi	Precision	Recall	F1-score	Support
Subjek 1	0.95	0.96	0.95	0.95	60
Subjek 2	0.93	0.94	0.93	0.93	60
Subjek 3	0.97	0.97	0.97	0.97	60

### 4.6.1 Pengujian Gestur pada Subjek 1



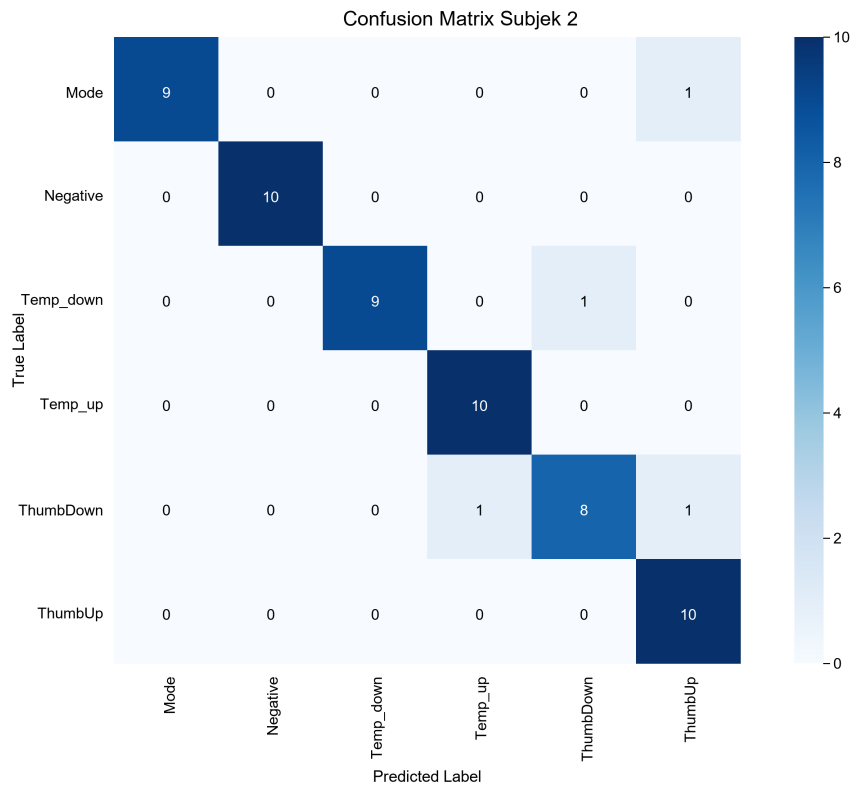
Gambar 4.12: Confusion matrix pengujian gestur pada subjek 1

Tabel 4.14: Hasil klasifikasi model pada Subjek 1

Kelas	Precision	Recall	F1-score	Support
Mode	1.00	1.00	1.00	10
Negative	1.00	0.90	0.95	10
Temp_down	1.00	0.90	0.95	10
Temp_up	0.83	1.00	0.91	10
ThumbDown	0.91	1.00	0.95	10
ThumbUp	1.00	0.90	0.95	10
Accuracy			0.95	60
Macro Avg	0.96	0.95	0.95	60
Weighted Avg	0.96	0.95	0.95	60

Berdasarkan Gambar 4.12 dan Tabel 4.14, pengujian pada Subjek 1 menghasilkan akurasi sebesar 95,00%. Kelas Mode, Temp\_up, dan ThumbDown dikenali sempurna dengan recall 1,00. Kesalahan terjadi pada kelas Negative yang diprediksi sebagai Temp\_up, Temp\_down yang diprediksi sebagai ThumbDown, serta ThumbUp yang diprediksi sebagai Temp\_up. Pola kesalahan ini menunjukkan bahwa perbedaan gaya gerakan dapat memengaruhi pembacaan arah gerak dan orientasi ibu jari.

## 4.6.2 Pengujian Gestur pada Subjek 2



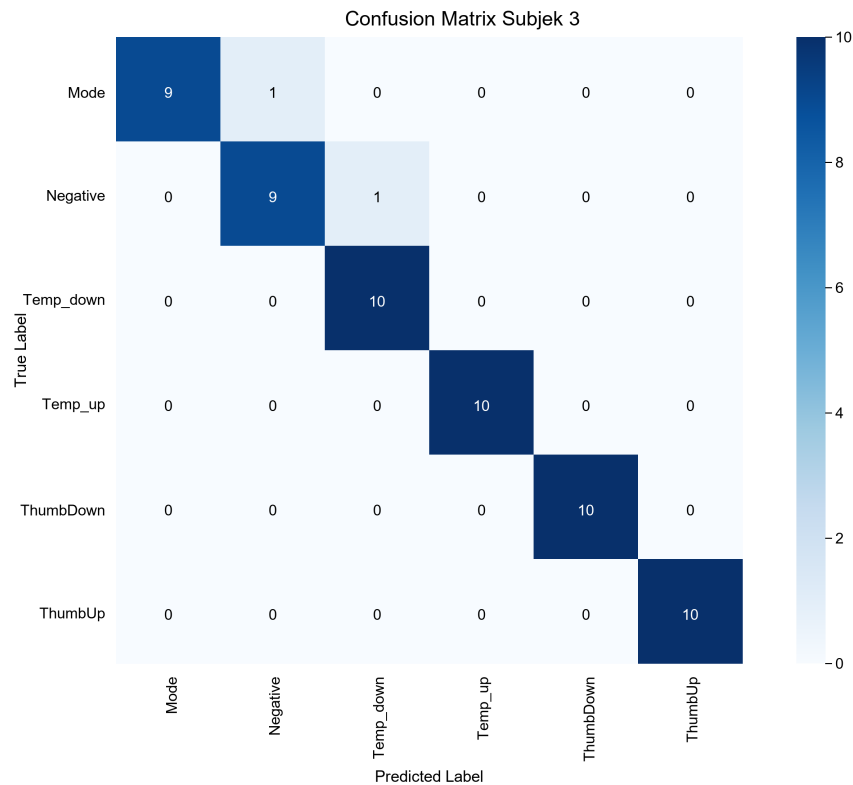
Gambar 4.13: Confusion matrix pengujian gestur pada subjek 2

Tabel 4.15: Hasil klasifikasi model pada Subjek 2

Kelas	Precision	Recall	F1-score	Support
Mode	1.00	0.90	0.95	10
Negative	1.00	1.00	1.00	10
Temp_down	1.00	0.90	0.95	10
Temp_up	0.91	1.00	0.95	10
ThumbDown	0.89	0.80	0.84	10
ThumbUp	0.83	1.00	0.91	10
Accuracy			0.93	60
Macro Avg	0.94	0.93	0.93	60
Weighted Avg	0.94	0.93	0.93	60

Pada Subjek 2, model memperoleh akurasi sebesar 93,33%. Kelas Negative, Temp\_up, dan ThumbUp memiliki recall 1,00, sedangkan kelas ThumbDown memiliki recall paling rendah, yaitu 0,80. Kesalahan terjadi pada Mode yang diprediksi sebagai ThumbUp, Temp\_down yang diprediksi sebagai ThumbDown, serta ThumbDown yang diprediksi sebagai Temp\_up dan ThumbUp. Hasil ini menunjukkan bahwa Subjek 2 memiliki variasi gerakan yang lebih menantang, terutama pada gestur berbasis ibu jari dan arah vertikal.

### 4.6.3 Pengujian Gestur pada Subjek 3



Gambar 4.14: Confusion matrix pengujian gestur pada subjek 3

Tabel 4.16: Hasil klasifikasi model pada Subjek 3

Kelas	Precision	Recall	F1-score	Support
Mode	1.00	0.90	0.95	10
Negative	0.90	0.90	0.90	10
Temp_down	0.91	1.00	0.95	10
Temp_up	1.00	1.00	1.00	10
ThumbDown	1.00	1.00	1.00	10
ThumbUp	1.00	1.00	1.00	10
Accuracy			0.97	60
Macro Avg	0.97	0.97	0.97	60
Weighted Avg	0.97	0.97	0.97	60

Pada Subjek 3, model memperoleh akurasi sebesar 96,67%, yaitu hasil tertinggi pada skenario subjek. Kelas Temp\_down, Temp\_up, ThumbDown, dan ThumbUp dapat dikenali sempurna dengan recall 1,00. Kesalahan hanya terjadi pada kelas Mode yang diprediksi sebagai Negative dan kelas Negative yang diprediksi sebagai Temp\_down. Hasil ini menunjukkan bahwa pola gerakan pada Subjek 3 relatif lebih konsisten terhadap pola data latih dibandingkan set pengujian subjek lainnya.

#### 4.6.4 Analisis Keseluruhan Pengaruh Perbedaan Subjek

Secara keseluruhan, pengujian variasi subjek menghasilkan akurasi gabungan 95,00% dari 180 data uji. Subjek 3 memperoleh akurasi tertinggi sebesar 96,67%, diikuti Subjek 1 sebesar 95,00%, dan Subjek 2 sebesar 93,33%. Perbedaan performa antar-subjek menunjukkan bahwa model cukup mampu melakukan generalisasi terhadap pengguna berbeda, tetapi tetap dipengaruhi oleh variasi gaya gerakan, ukuran tangan, sudut telapak tangan, dan orientasi ibu jari. Pola kesalahan paling sering muncul pada gestur yang memiliki kemiripan bentuk atau arah, seperti Temp\_up, Temp\_down, ThumbDown, dan ThumbUp. Dengan demikian, penambahan variasi data dari lebih banyak subjek dan variasi orientasi tangan tetap diperlukan untuk meningkatkan ketahanan model pada penggunaan nyata.

#### 4.7 Pengujian IR Blaster Terhadap Jarak, Arah, dan Desain Casing

Pada penelitian ini digunakan dua desain casing IR Blaster, yaitu casing lama dan casing baru. Casing lama merupakan desain awal dengan lubang LED IR menghadap lurus ke atas, sehingga arah pancaran inframerah cenderung vertikal dan lebih efektif ketika perangkat berada dekat atau tepat di bawah AC. Sementara itu, casing baru merupakan desain pengembangan dengan lubang LED IR dibuat miring ke arah luar sekitar 30 derajat, sehingga pancaran inframerah diharapkan lebih menyebar dan lebih mudah mengarah ke sensor penerima AC. Perbedaan desain ini diuji untuk mengetahui pengaruh bentuk casing dan arah pancaran LED IR terhadap keberhasilan pengiriman sinyal pada berbagai jarak dan arah hadap perangkat.

##### 4.7.1 Pengujian IR Blaster dengan Casing Lama Menghadap ke Atas pada Berbagai Jarak

Perintah	Dibawah AC		2 Meter		4,3 Meter		6,9 Meter	
	SHARP	Panasonic	SHARP	Panasonic	SHARP	Panasonic	SHARP	Panasonic
ON	1	1	1/0	1	0	0	0	0
OFF	1	1	1/0	1	0	0	0	0
SUHU NAIK	1	1	1/0	1	0	0	0	0
SUHU TURUN	1	1	1	1	0	0	0	0
GANTI MODE	1	1	0/1	1	0	0	0	0

Tabel 4.17: Pengujian IR Blaster dengan Casing Lama Menghadap ke Atas pada Berbagai Jarak

Berdasarkan Tabel 4.17, casing lama dengan posisi menghadap ke atas mampu mengirimkan seluruh perintah dengan baik ketika perangkat berada di bawah AC. Pada kondisi tersebut, seluruh perintah berhasil dikirimkan baik ke AC Sharp maupun Panasonic. Hal ini menunjukkan bahwa ketika jarak perangkat sangat dekat dengan AC, arah pancaran yang menghadap ke atas masih mampu menjangkau sensor penerima AC. Pada jarak 2 meter, performa mulai menurun terutama pada AC Sharp. Beberapa perintah pada AC Sharp hanya berhasil sebagian, seperti ditunjukkan oleh nilai 1/0 atau 0/1. Sementara itu, pada AC Panasonic, seluruh perintah masih berhasil dikirimkan. Kondisi ini menunjukkan bahwa keberhasilan pengiriman IR tidak hanya dipengaruhi oleh jarak, tetapi juga oleh sensitivitas atau posisi sensor IR pada masing-

masing merek AC. Pada jarak 4,3 meter dan 6,9 meter, seluruh perintah gagal dikirimkan pada AC Sharp maupun Panasonic. Hal ini menunjukkan bahwa posisi casing yang menghadap ke atas tidak efektif untuk jarak menengah hingga jauh. Arah pancaran IR yang tidak langsung mengarah ke sensor AC membuat sinyal tidak dapat diterima dengan baik ketika jarak semakin bertambah.

#### 4.7.2 Pengujian IR Blaster dengan Casing Baru Menghadap ke Atas pada Berbagai Jarak

Perintah	Dibawah AC		2 Meter		4,3 Meter		6,9 Meter	
	SHARP	Panasonic	SHARP	Panasonic	SHARP	Panasonic	SHARP	Panasonic
ON	1	1	0/1	1	0	0	0	0
OFF	1	1	1	1	0	0	0	0
SUHU NAIK	1	1	1	1	0	0	0	0
SUHU TURUN	1	1	1	1	0	0	0	0
GANTI MODE	1	1	1	1	0	0	0	0

Tabel 4.18: Pengujian IR Blaster dengan Casing Baru Menghadap ke Atas pada Berbagai Jarak

Berdasarkan Tabel 4.18, casing baru dengan posisi menghadap ke atas juga mampu bekerja dengan baik pada posisi di bawah AC. Seluruh perintah berhasil dikirimkan ke AC Sharp dan Panasonic. Hal ini menunjukkan bahwa desain casing baru masih dapat memancarkan sinyal IR dengan baik pada jarak sangat dekat. Pada jarak 2 meter, casing baru menunjukkan hasil yang sedikit lebih baik dibandingkan casing lama pada posisi menghadap ke atas. Hanya terdapat satu kondisi pengiriman sebagian pada AC Sharp, sedangkan perintah lainnya berhasil dikirimkan dengan baik. Pada AC Panasonic, seluruh perintah berhasil dikirimkan. Hasil ini menunjukkan bahwa casing baru masih cukup efektif pada jarak dekat hingga sekitar 2 meter apabila dibandingkan dengan casing lama pada arah yang sama. Namun, pada jarak 4,3 meter dan 6,9 meter, seluruh perintah gagal dikirimkan pada kedua merek AC. Dengan demikian, meskipun casing baru memiliki hasil lebih baik pada jarak 2 meter, posisi menghadap ke atas tetap tidak efektif untuk jarak lebih jauh. Arah pancaran IR perlu diarahkan langsung ke AC agar sinyal dapat diterima secara optimal.

### 4.7.3 Pengujian IR Blaster dengan Casing Lama Menghadap ke Arah AC pada Berbagai Jarak

Perintah	Dibawah AC		2 Meter		4,3 Meter		6,9 Meter	
	SHARP	Panasonic	SHARP	Panasonic	SHARP	Panasonic	SHARP	Panasonic
ON	1	1	1	1	1	1	1	1
OFF	1	1	1	1	1	1	1	1
SUHU NAIK	1	1	1	1	1	1	1	1
SUHU TURUN	1	1	1	1	1	1	1	1
GANTI MODE	1	1	1	1	1	1	1	1

Tabel 4.19: Pengujian IR Blaster dengan Casing Lama Menghadap ke Arah AC pada Berbagai Jarak

Berdasarkan Tabel 4.19, casing lama dengan posisi menghadap langsung ke arah AC memberikan hasil paling baik dibandingkan skenario pengujian lainnya. Seluruh perintah berhasil dikirimkan pada semua variasi jarak, yaitu di bawah AC, 2 meter, 4,3 meter, dan 6,9 meter. Hasil ini berlaku untuk kedua merek AC, yaitu Sharp dan Panasonic. Keberhasilan pada seluruh jarak menunjukkan bahwa casing lama memiliki arah pancaran IR yang efektif ketika diarahkan langsung ke sensor AC. Pada jarak terjauh, yaitu 6,9 meter, seluruh perintah masih berhasil dikirimkan. Hal ini menunjukkan bahwa desain casing lama tidak terlalu menghalangi pancaran sinyal IR dan mampu mempertahankan jangkauan pengiriman sinyal secara stabil. Hasil ini juga menunjukkan bahwa faktor arah hadap perangkat memiliki pengaruh yang sangat besar. Ketika IR Blaster diarahkan langsung ke AC, sinyal IR dapat diterima dengan jauh lebih baik dibandingkan posisi casing yang menghadap ke atas.

### 4.7.4 Pengujian IR Blaster dengan Casing Baru Menghadap ke Arah AC pada Berbagai Jarak

Perintah	Dibawah AC		2 Meter		4,3 Meter		6,9 Meter	
	SHARP	Panasonic	SHARP	Panasonic	SHARP	Panasonic	SHARP	Panasonic
ON	1	1	1	1	0	1	0	0
OFF	1	1	1	1	0	1	0	0
SUHU NAIK	1	1	1	1	0	1	0	0
SUHU TURUN	1	1	1	1	0	1	0	0
GANTI MODE	1	1	1	1	0	1	0	0

Tabel 4.20: Pengujian IR Blaster dengan Casing Baru Menghadap ke Arah AC pada Berbagai Jarak

Berdasarkan Tabel 4.20, casing baru dengan posisi menghadap langsung ke arah AC menunjukkan peningkatan performa dibandingkan ketika casing menghadap ke atas. Pada jarak di

bawah AC dan 2 meter, seluruh perintah berhasil dikirimkan ke AC Sharp dan Panasonic. Hal ini menunjukkan bahwa casing baru masih efektif digunakan pada jarak dekat hingga menengah. Pada jarak 4,3 meter, terdapat perbedaan hasil antara AC Sharp dan Panasonic. Seluruh perintah pada AC Panasonic berhasil dikirimkan, sedangkan seluruh perintah pada AC Sharp gagal. Perbedaan ini menunjukkan bahwa keberhasilan pengiriman IR dapat dipengaruhi oleh posisi sensor penerima, sensitivitas sensor, atau sudut penerimaan pada masing-masing merek AC. Pada jarak 6,9 meter, seluruh perintah gagal dikirimkan pada kedua merek AC. Hasil ini menunjukkan bahwa casing baru memiliki keterbatasan jangkauan pada jarak jauh, meskipun sudah diarahkan langsung ke AC. Kemungkinan penyebabnya adalah bukaan casing, posisi LED IR, atau sudut pancaran sinyal yang kurang optimal sehingga intensitas sinyal yang sampai ke sensor AC menjadi tidak cukup kuat.

#### **4.7.5 Analisis Keseluruhan Pengujian IR Blaster terhadap Jarak, Arah, dan Desain Casing**

Berdasarkan seluruh hasil pengujian, faktor yang paling berpengaruh terhadap keberhasilan pengiriman sinyal IR adalah arah hadap casing. Posisi casing yang menghadap langsung ke arah AC memberikan hasil yang jauh lebih baik dibandingkan posisi casing yang menghadap ke atas. Ketika casing menghadap ke atas, pengiriman sinyal hanya efektif pada jarak dekat, terutama saat perangkat berada di bawah AC dan sebagian pada jarak 2 meter. Pada jarak 4,3 meter dan 6,9 meter, posisi menghadap ke atas tidak mampu mengirimkan sinyal dengan baik. Dari sisi desain casing, casing lama memiliki performa terbaik ketika diarahkan langsung ke AC. Casing lama berhasil mengirimkan seluruh perintah pada semua jarak hingga 6,9 meter untuk kedua merek AC. Sebaliknya, casing baru masih bekerja baik pada jarak di bawah AC dan 2 meter, tetapi performanya menurun pada jarak 4,3 meter dan gagal sepenuhnya pada jarak 6,9 meter.

Hasil ini menunjukkan bahwa desain fisik casing sangat memengaruhi arah dan jangkauan pancaran IR. Casing yang terlalu membatasi bukaan LED IR atau tidak mengarahkan pancaran secara optimal dapat menurunkan keberhasilan pengiriman sinyal, terutama pada jarak jauh. Oleh karena itu, desain casing final sebaiknya memperhatikan posisi LED IR, arah bukaan casing, sudut pancaran, dan orientasi pemasangan perangkat agar sinyal inframerah dapat menjangkau sensor AC secara stabil.

#### **4.8 Pengujian IR Blaster Terhadap Pencahayaan**

Pengujian IR Blaster terhadap pencahayaan dilakukan untuk mengetahui apakah perubahan intensitas cahaya ruangan memengaruhi keberhasilan pengiriman sinyal inframerah ke AC. Pengujian dilakukan pada tiga kondisi pencahayaan, yaitu gelap  $\pm 55$  lux, redup  $\pm 100$  lux, dan terang  $\pm 230$  lux. Setiap kondisi diuji sebanyak 15 percobaan, yang berasal dari 5 perintah aktif dengan masing-masing perintah dilakukan sebanyak 3 kali. Hasil pengujian tersebut ditunjukkan pada Tabel 4.21.

Percobaan ke-	Gelap ( $\pm 55$ lux)	Redup ( $\pm 100$ lux)	Terang ( $\pm 230$ lux)
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1
7	1	1	1
8	1	1	1
9	1	1	1
10	1	1	1
11	1	1	1
12	1	1	1
13	1	1	1
14	1	1	1
15	1	1	1

Tabel 4.21: Hasil pengujian IR Blaster terhadap kondisi pencahayaan

Berdasarkan Tabel 4.21, ketiga kondisi pencahayaan, yaitu gelap  $\pm 55$  lux, redup  $\pm 100$  lux, dan terang  $\pm 230$  lux, memperoleh hasil keberhasilan penuh, yaitu 15 dari 15 percobaan berhasil atau setara dengan tingkat keberhasilan 100%. Hal ini menunjukkan bahwa pada seluruh kondisi pencahayaan yang diuji, sinyal inframerah tetap dapat diterima oleh AC dengan baik. Dengan demikian, perubahan intensitas cahaya tampak pada rentang tersebut tidak memberikan gangguan berarti terhadap proses pengiriman sinyal IR.

Secara keseluruhan, hasil pengujian menunjukkan bahwa pencahayaan ruangan bukan faktor utama yang memengaruhi keberhasilan pengiriman sinyal IR. Hal ini karena inframerah tidak bergantung pada cahaya tampak seperti sistem berbasis kamera. Performa IR Blaster lebih banyak dipengaruhi oleh faktor fisik seperti arah hadap casing, jarak terhadap AC, posisi LED IR, desain casing, serta arah sensor penerima AC.

## 4.9 Pengujian Kecepatan Pengiriman Perintah

Pengujian kecepatan pengiriman perintah dilakukan untuk mengetahui waktu respons sistem sejak gestur berhasil terdeteksi hingga perintah inframerah dikirimkan oleh ESP32. Pengujian dilakukan sebanyak 15 percobaan yang dibagi ke dalam 5 gestur aktif, dengan masing-masing gestur diuji sebanyak 3 kali. Gestur Negative tidak dimasukkan dalam pengujian ini karena tidak menghasilkan perintah kendali dan tidak dikirimkan ke ESP32. Pada pengujian ini, parameter yang diamati meliputi detection to publish, detection to status, dan publish latency. Detection to publish menunjukkan waktu dari gestur terdeteksi hingga Raspberry Pi melakukan publish perintah. Detection to status menunjukkan waktu total dari gestur terdeteksi hingga status pengiriman diterima, yang menandakan bahwa perintah telah diproses sampai tahap pengiriman IR oleh ESP32. Sementara itu, publish latency menunjukkan waktu yang berkaitan dengan proses publish perintah. Hasil pengujian ditunjukkan pada Tabel 4.22.

Percobaan ke-	Detection to publish (s)	Detection to status (s)	Publish latency (s)
1	0.297832	1.110731	0.297794
2	0.277105	0.960502	0.277073
3	0.259975	0.953407	0.259944
4	0.234583	0.889648	0.234551
5	0.233504	0.881018	0.233472
6	0.239674	0.845153	0.239640
7	0.247922	1.027222	0.247889
8	0.266462	1.063288	0.266429
9	0.241951	1.019858	0.241925
10	0.234768	1.094240	0.234740
11	0.235050	0.898344	0.235024
12	0.233041	0.937335	0.233012
13	0.224155	0.840029	0.224129
14	0.233740	0.943711	0.233714
15	0.255062	0.877478	0.255036

Tabel 4.22: Hasil pengujian 15 percobaan waktu pengiriman perintah

Berdasarkan Tabel 4.22, waktu detection to publish berada pada rentang 0,224155 detik hingga 0,297832 detik. Nilai ini menunjukkan bahwa proses dari gestur terdeteksi hingga perintah dipublikasikan oleh Raspberry Pi berlangsung cukup cepat dan relatif stabil. Rata-rata detection to publish berdasarkan rekapitulasi adalah 0,247655 detik.

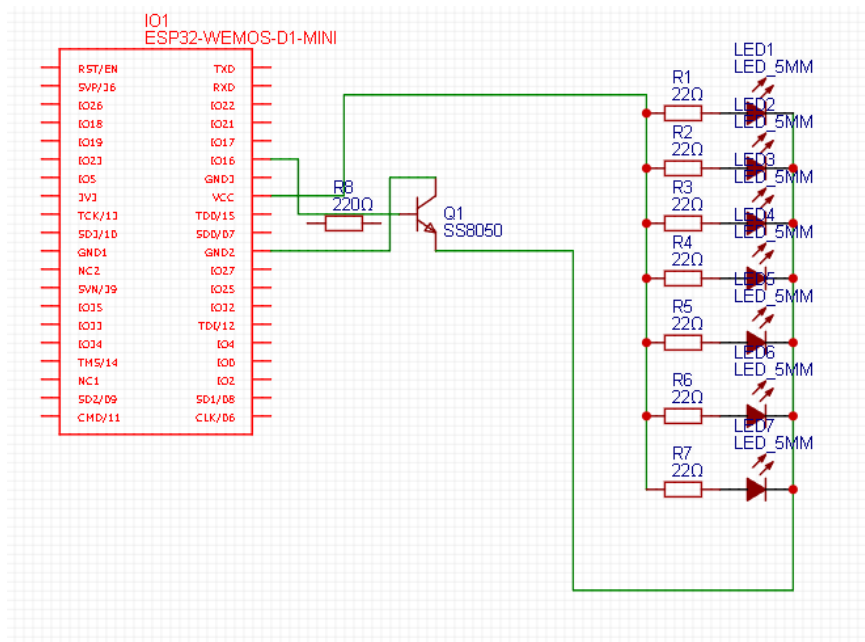
Sementara itu, waktu detection to status berada pada rentang 0,840029 detik hingga 1,110731 detik. Waktu tercepat terjadi pada percobaan ke-13, sedangkan waktu terlama terjadi pada percobaan ke-1. Berdasarkan rekapitulasi, rata-rata waktu detection to status adalah 0,956131 detik. Hasil ini menunjukkan bahwa secara rata-rata sistem membutuhkan waktu kurang dari 1 detik sejak gestur terdeteksi hingga proses pengiriman perintah IR selesai atau status diterima. Total waktu deteksi ke status adalah 14,341964 detik, dengan minimum 0,840029 detik dan maksimum 1,110731 detik.

#### 4.10 Hasil Perancangan Alat

Hasil perancangan alat pada penelitian ini mencakup realisasi rangkaian IR Blaster, desain PCB, desain casing, serta hasil perakitan perangkat secara keseluruhan. Perangkat ini berperan sebagai aktuator pada sistem kendali AC berbasis gestur tangan. Setelah hasil klasifikasi gestur dikirimkan ke ESP32, perangkat akan menerjemahkan perintah tersebut menjadi sinyal inframerah dan memancarkannya ke unit AC melalui rangkaian IR Blaster. Perancangan alat dilakukan dengan mempertimbangkan beberapa aspek, yaitu kestabilan rangkaian, kemudahan integrasi dengan ESP32, ukuran perangkat yang ringkas, serta arah pancaran sinyal inframerah. Selain itu, desain casing juga menjadi bagian penting karena posisi lubang LED IR dapat memengaruhi arah pancaran dan jangkauan sinyal terhadap sensor penerima AC.

### 4.10.1 Hasil Rangkaian IR Blaster

Rangkaian IR Blaster dirancang menggunakan ESP32 D1 Mini sebagai mikrokontroler utama. ESP32 berfungsi untuk menerima perintah kendali dan menghasilkan sinyal inframerah melalui pin keluaran yang telah ditentukan. Pada rancangan ini, sinyal IR dikendalikan melalui GPIO 17. Sumber daya perangkat berasal dari koneksi USB, sehingga perangkat dapat dioperasikan menggunakan catu daya standar 5 V. Rangkaian pemancar inframerah menggunakan 7 buah LED IR TSAL6400. Jumlah LED yang lebih dari satu digunakan untuk meningkatkan intensitas pancaran dan memperluas area jangkauan sinyal. Karena pin GPIO ESP32 tidak dirancang untuk memberikan arus besar secara langsung ke beberapa LED IR, maka digunakan transistor SS8050 sebagai saklar elektronik sekaligus penguat arus. Transistor ini memungkinkan arus dari sumber daya mengalir ke rangkaian LED IR ketika GPIO 17 memberikan sinyal kendali. Pada rangkaian ini, resistor 220  $\Omega$  dipasang pada jalur basis transistor untuk membatasi arus yang masuk ke transistor. Sementara itu, masing-masing LED IR diberi resistor 22  $\Omega$  pada jalur positif LED untuk membatasi arus yang mengalir melalui LED. Penggunaan resistor tersebut bertujuan menjaga komponen tetap bekerja dalam batas aman dan mengurangi risiko kerusakan akibat arus berlebih.



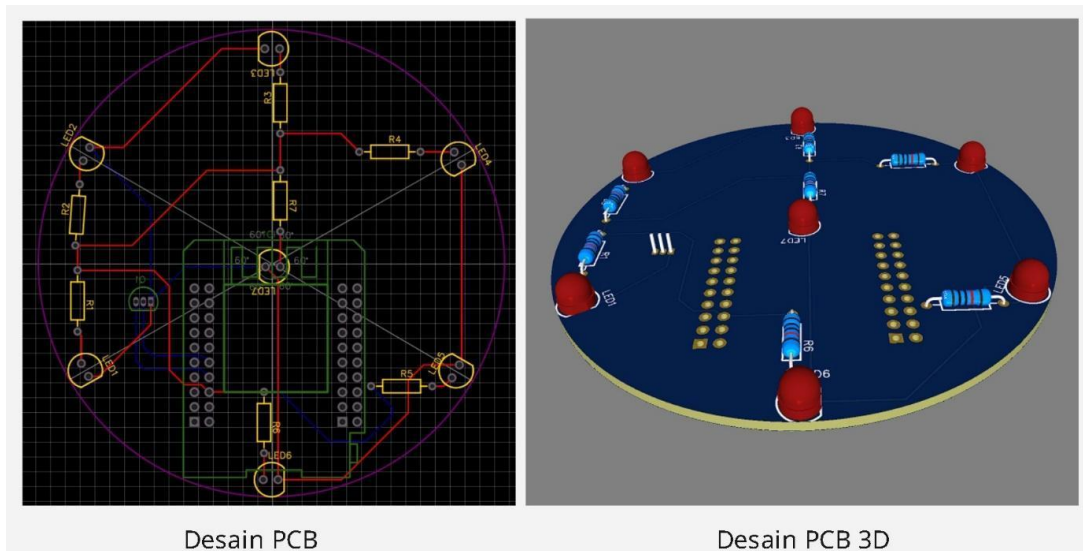
Gambar 4.15: Skematik rangkaian IR Blaster

Berdasarkan Gambar 4.15, ESP32 memberikan sinyal kendali melalui GPIO 17 menuju basis transistor SS8050. Ketika transistor aktif, arus akan mengalir ke tujuh LED IR sehingga seluruh LED dapat memancarkan sinyal inframerah secara bersamaan. Dengan konfigurasi ini, sinyal inframerah yang dihasilkan menjadi lebih kuat dibandingkan penggunaan satu LED IR. Rancangan ini diharapkan mampu meningkatkan keberhasilan pengiriman perintah ke AC, terutama ketika perangkat digunakan pada jarak dan arah yang bervariasi.

### 4.10.2 Hasil Desain PCB

Setelah rangkaian skematik selesai dirancang, tahap berikutnya adalah pembuatan desain Printed Circuit Board (PCB). PCB digunakan untuk menempatkan komponen secara lebih rapi,

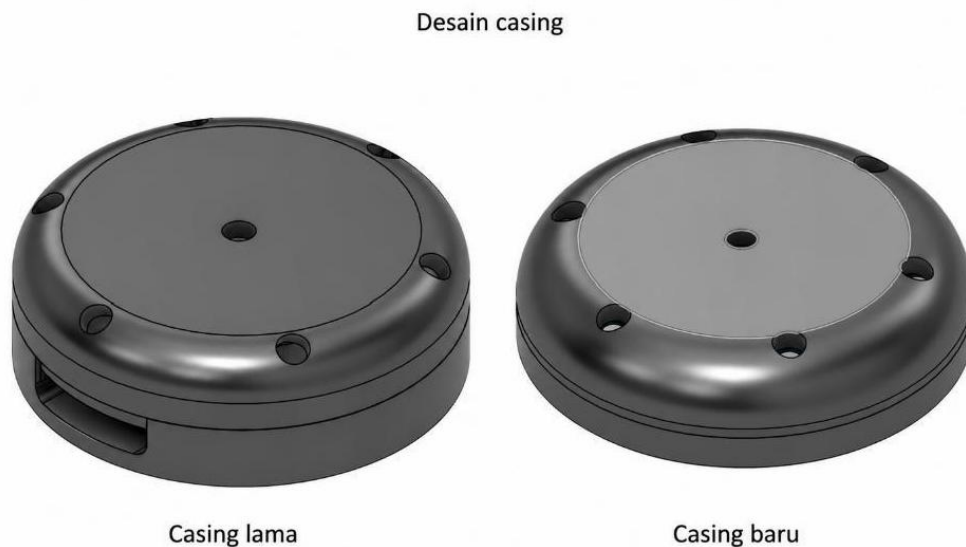
mengurangi penggunaan kabel jumper, serta meningkatkan kestabilan koneksi antar komponen. Desain PCB dibuat dalam bentuk melingkar agar menyesuaikan bentuk casing dan mendukung penempatan LED IR secara menyebar. Pada desain PCB, tujuh LED IR ditempatkan di bagian tepi PCB dengan pola melingkar. Penempatan ini bertujuan agar pancaran inframerah dapat menjangkau area yang lebih luas. Resistor LED ditempatkan berdekatan dengan masing-masing LED untuk mempermudah jalur rangkaian dan mengurangi kompleksitas routing. Bagian tengah PCB digunakan untuk penempatan koneksi modul ESP32 dan jalur pengendali transistor.



Gambar 4.16: Desain PCB IR Blaster

Berdasarkan Gambar 4.16, tata letak komponen dibuat agar mendukung bentuk alat yang ringkas dan simetris. LED IR yang ditempatkan secara melingkar memungkinkan arah pancaran tidak hanya terpusat pada satu titik. Selain itu, penggunaan PCB membuat perangkat lebih kokoh dibandingkan rangkaian yang disusun secara manual menggunakan kabel. Dengan desain ini, proses perakitan ke dalam casing juga menjadi lebih mudah karena posisi komponen telah disesuaikan dengan lubang LED pada casing.

### 4.10.3 Hasil Desain Casing



Gambar 4.17: Desain casing lama dan casing baru

Casing dirancang sebagai pelindung rangkaian elektronik sekaligus sebagai struktur mekanik yang menentukan posisi dan arah LED IR. Bahan yang digunakan untuk casing adalah PLA, karena bahan ini mudah dicetak menggunakan printer 3D, ringan, dan cukup kuat untuk

kebutuhan prototipe. Bentuk casing dibuat bulat agar sesuai dengan bentuk PCB dan susunan LED IR yang ditempatkan secara melingkar. Pada penelitian ini terdapat dua desain casing, yaitu casing lama dan casing baru. Casing lama merupakan desain awal dengan posisi lubang LED IR yang menghadap lurus ke atas. Desain ini membuat pancaran inframerah cenderung diarahkan secara vertikal. Sementara itu, casing baru merupakan hasil pengembangan dari desain sebelumnya. Pada casing baru, enam lubang LED di bagian tepi dibuat miring ke arah luar dengan sudut sekitar 30 derajat, sedangkan satu LED di bagian tengah tetap menghadap lurus ke atas. Perubahan ini dilakukan agar pancaran inframerah lebih menyebar dan jangkauan sinyal menjadi lebih merata.

Berdasarkan Gambar 4.17, desain casing baru memiliki perbedaan utama pada orientasi lubang LED IR. Pada casing lama, lubang LED yang menghadap ke atas membuat pancaran lebih efektif ketika alat berada dekat atau tepat di bawah AC. Namun, desain tersebut berpotensi kurang optimal ketika alat berada pada posisi yang tidak sejajar dengan sensor penerima AC. Oleh karena itu, casing baru dirancang dengan sudut kemiringan LED ke arah luar untuk memperluas area pancaran. Dengan desain ini, sinyal IR diharapkan lebih mudah mengenai sensor AC pada berbagai posisi pemasangan perangkat.

### 4.10.4 Hasil Perakitan Alat

Tahap akhir dari perancangan alat adalah proses perakitan seluruh komponen menjadi perangkat IR Blaster yang siap digunakan. Proses perakitan meliputi pemasangan LED IR, resistor, transistor SS8050, dan modul ESP32 D1 Mini pada PCB. Setelah seluruh komponen elektronik terpasang, rangkaian dimasukkan ke dalam casing PLA yang telah dicetak.



Gambar 4.18: Hasil perakitan alat IR Blaster

Berdasarkan Gambar 4.18, perangkat berhasil dirakit menjadi satu unit IR Blaster yang ringkas. Seluruh LED IR dapat menyala dan aktif ketika perintah dikirimkan oleh ESP32. Casing juga berhasil terpasang dengan baik sehingga rangkaian elektronik terlindungi dan posisi LED IR tetap sesuai dengan rancangan. Hasil pengujian awal menunjukkan bahwa alat berhasil menyala dan dapat digunakan untuk mengontrol beberapa merek AC. Hal ini menunjukkan bahwa rangkaian elektronik, PCB, firmware ESP32, dan casing telah terintegrasi dengan baik. Perubahan desain casing dari versi lama ke versi baru dilakukan sebagai upaya untuk meningkatkan pemerataan arah pancaran inframerah. Dengan demikian, perangkat IR Blaster yang dirancang telah memenuhi fungsi utama sebagai aktuator kendali AC berbasis sinyal inframerah. Secara keseluruhan, hasil perancangan alat menunjukkan bahwa perangkat berhasil direalisasikan sesuai dengan kebutuhan sistem. Perangkat dapat diberi daya melalui USB, dikendalikan oleh ESP32, dan memancarkan sinyal inframerah melalui tujuh LED IR. Hasil ini menjadi dasar untuk melanjutkan tahap pengujian performa alat, khususnya terhadap variasi jarak, arah casing, desain casing, dan kondisi pencahayaan.

#### 4.11 Pembahasan dan Analisis Hasil

Berdasarkan seluruh hasil pengujian yang telah dilakukan, sistem kendali AC berbasis gestur tangan menggunakan LSTM pada perangkat edge menunjukkan performa yang baik, baik dari sisi pengenalan gestur maupun dari sisi pengiriman perintah ke AC melalui IR Blaster. Sistem mampu menjalankan alur kerja mulai dari akuisisi citra tangan, ekstraksi landmark menggunakan MediaPipe, klasifikasi gestur menggunakan model LSTM, pengiriman perintah melalui MQTT, hingga eksekusi sinyal inframerah oleh ESP32. Dataset yang digunakan terdiri dari 2.740 raw sample yang tersebar pada enam kelas gestur. Evaluasi model dilakukan menggunakan 501 data uji. Pada pengujian performa model, model Keras memperoleh akurasi sebesar 98,80% dengan precision, recall, dan F1-score keseluruhan sebesar 98,80%. Setelah dikonversi ke format TFLite, model tetap memperoleh akurasi yang sama, yaitu 98,80%. Hal ini

menunjukkan bahwa proses konversi model ke TFLite tidak menurunkan performa klasifikasi, sehingga model layak digunakan pada perangkat edge seperti Raspberry Pi. Pola kesalahan utama terdapat pada kelas Negative dan ThumbUp, terutama ketika beberapa data Negative diprediksi sebagai ThumbUp dan beberapa data ThumbUp diprediksi sebagai Negative.

Pengujian terhadap variasi jarak menunjukkan bahwa model masih mampu mengenali gestur dengan baik pada jarak 1 meter, 1,5 meter, 2 meter, dan 3 meter. Akurasi gabungan pada skenario jarak adalah 95,83%. Akurasi tertinggi diperoleh pada jarak 1 meter sebesar 96,67%, sedangkan akurasi terendah terjadi pada jarak 3 meter sebesar 93,33%. Pada jarak 1,5 meter dan 2 meter, model memperoleh akurasi yang sama, yaitu 95,00%. Pola kesalahan menunjukkan bahwa jarak yang lebih jauh meningkatkan risiko kekeliruan pada kelas yang bergantung pada arah gerak dan orientasi ibu jari, seperti Temp\_up, Temp\_down, ThumbDown, dan ThumbUp. Pada jarak 3 meter, terdapat pula data Negative yang diprediksi sebagai ThumbDown, sehingga penggunaan threshold confidence dan smoothing prediksi tetap diperlukan untuk menekan risiko false trigger.

Pengujian terhadap pencahayaan menunjukkan bahwa intensitas cahaya memiliki pengaruh paling besar terhadap performa pengenalan gestur. Akurasi gabungan pada skenario pencahayaan adalah 89,44%. Pada kondisi terang  $\pm 230$  lux, model memperoleh akurasi 94,92%. Pada kondisi redup  $\pm 100$  lux, akurasi menurun menjadi 91,80%, dan pada kondisi gelap  $\pm 55$  lux, akurasi turun menjadi 81,67%. Penurunan performa paling terlihat pada kelas Temp\_up dan Negative pada kondisi gelap. Hal ini menunjukkan bahwa kualitas citra dan kestabilan deteksi landmark sangat dipengaruhi oleh pencahayaan ruangan. Dengan demikian, sistem paling ideal digunakan pada kondisi pencahayaan cukup, atau dilengkapi pencahayaan tambahan ketika digunakan pada ruangan gelap.

Pengujian terhadap variasi subjek melibatkan 6 orang, yang terdiri dari 3 laki-laki dan 3 perempuan. Hasil pengujian disajikan dalam tiga set pengujian subjek dengan total 180 data uji. Akurasi gabungan pada skenario subjek adalah 95,00%. Subjek 3 memperoleh akurasi tertinggi sebesar 96,67%, Subjek 1 memperoleh akurasi 95,00%, dan Subjek 2 memperoleh akurasi 93,33%. Perbedaan akurasi antar-subjek menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap pengguna berbeda, tetapi performanya tetap dipengaruhi oleh variasi gaya gerakan, ukuran tangan, sudut telapak tangan, dan orientasi ibu jari. Kelas yang paling sensitif terhadap variasi subjek adalah kelas yang memiliki kemiripan bentuk atau arah, terutama Temp\_up, Temp\_down, ThumbDown, dan ThumbUp.

Dari sisi aktuasi, pengujian IR Blaster terhadap jarak, arah, dan desain casing menunjukkan bahwa keberhasilan pengiriman sinyal inframerah sangat dipengaruhi oleh arah hadap casing dan desain fisik perangkat. Posisi casing yang menghadap langsung ke arah AC memberikan hasil lebih baik dibandingkan posisi menghadap ke atas. Casing lama menunjukkan performa terbaik karena mampu mengirimkan sinyal secara stabil hingga jarak 6,9 meter, sedangkan casing baru masih efektif pada jarak dekat hingga menengah, tetapi performanya menurun pada jarak yang lebih jauh. Hasil ini menunjukkan bahwa desain casing, bukaan LED IR, dan sudut pancaran LED sangat berpengaruh terhadap keberhasilan pengiriman sinyal inframerah. Pengujian IR Blaster terhadap pencahayaan menunjukkan bahwa pencahayaan ruangan tidak menjadi faktor utama dalam keberhasilan pengiriman sinyal IR. Pada kondisi gelap, redup, dan terang, tingkat keberhasilan pengiriman mencapai 100%.

Pada pengujian kecepatan pengiriman perintah, sistem memperoleh rata-rata waktu dari gestur terdeteksi hingga status pengiriman diterima sebesar 0,956131 detik. Nilai ini menun-

jukkan bahwa sistem mampu memberikan respons dalam waktu kurang dari satu detik secara rata-rata. Waktu dari deteksi hingga publish MQTT juga relatif kecil, yaitu sekitar 0,247655 detik, sehingga keterlambatan utama lebih banyak terjadi setelah proses publish, seperti komunikasi MQTT, pemrosesan perintah oleh ESP32, proses IR encoding, pengiriman sinyal IR, dan pengiriman status kembali.

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem telah mampu menjalankan fungsi utama sebagai kendali AC berbasis gestur tangan. Model LSTM mampu mengenali gestur dengan akurasi tinggi, model TFLite tetap mempertahankan performa setelah konversi, dan ESP32 mampu menjalankan perintah melalui IR Blaster. Namun, sistem masih memiliki beberapa keterbatasan, terutama pada kondisi pencahayaan rendah, variasi subjek tertentu, serta desain casing IR Blaster pada jarak jauh. Oleh karena itu, peningkatan variasi dataset, optimasi pencahayaan, dan perbaikan desain casing menjadi aspek penting untuk pengembangan sistem selanjutnya.

# BAB V

## PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, sistem kendali AC berbasis gestur tangan menggunakan LSTM pada perangkat edge berhasil dibangun dan mampu menjalankan alur kerja mulai dari akuisisi citra tangan, ekstraksi landmark menggunakan MediaPipe, klasifikasi gestur menggunakan model LSTM, pengiriman perintah melalui MQTT, hingga eksekusi sinyal inframerah oleh ESP32. Model yang digunakan menunjukkan performa sangat baik dengan akurasi model Keras sebesar 98,80% dan tetap mempertahankan akurasi yang sama setelah dikonversi ke TFLite, sehingga layak digunakan pada Raspberry Pi sebagai perangkat edge. Pengujian terhadap jarak, pencahayaan, dan subjek berbeda menunjukkan bahwa sistem masih mampu mengenali gestur dengan baik, meskipun performa dapat menurun pada kondisi tertentu seperti pencahayaan gelap atau variasi gerakan antar pengguna. Dari sisi aktuasi, IR Blaster mampu mengirimkan perintah ke AC, tetapi keberhasilannya dipengaruhi oleh arah hadap perangkat, jarak, dan desain casing. Secara keseluruhan, sistem yang dirancang telah mampu menggantikan fungsi dasar remote AC konvensional melalui gestur tangan, dengan respons rata-rata pengiriman perintah kurang dari satu detik dan tingkat akurasi pengenalan gestur yang tinggi.

### 5.2 Saran

Berdasarkan hasil pengujian dan keterbatasan yang ditemukan, saran untuk pengembangan selanjutnya adalah sebagai berikut:

1. Dataset dapat ditambah dengan variasi kondisi yang lebih luas, terutama pada pencahayaan rendah, jarak lebih jauh, dan jumlah subjek yang lebih banyak, agar model memiliki kemampuan generalisasi yang lebih baik pada kondisi penggunaan nyata.
2. Sistem dapat dikembangkan dengan aplikasi mobile sebagai antarmuka tambahan. Aplikasi ini dapat digunakan untuk konfigurasi perangkat, pemilihan merek AC, pemantauan status koneksi, serta kendali manual apabila pengguna tidak ingin menggunakan gestur.
3. Desain casing IR Blaster perlu disempurnakan agar pancaran inframerah lebih optimal. Salah satu pengembangan yang dapat dilakukan adalah menggunakan penutup casing berbahan mika atau transparan pada bagian LED IR, sehingga pancaran sinyal inframerah tidak terlalu terhalang dan dapat menyebar lebih luas.
4. Pengujian IR Blaster dapat diperluas pada lebih banyak merek AC dan posisi pemasangan perangkat, karena setiap AC memiliki posisi sensor dan sensitivitas penerimaan inframerah yang berbeda.

5. Proses komunikasi MQTT dan pemrosesan pada ESP32 masih dapat dioptimalkan agar waktu respons sistem menjadi lebih cepat dan konsisten, terutama pada proses penerimaan perintah, encoding IR, pengiriman sinyal, dan pengiriman status kembali.

## DAFTAR PUSTAKA

- [1] H. R. B and A. Deepak. Review on infrared communication. (6):82–83, 2016.
- [2] B. I. Alabdullah et al. Smart home automation-based hand gesture recognition using feature fusion and recurrent neural network. *Sensors*, 23(17), 2023.
- [3] D. Park, S. Kim, Y. An, and J. Y. Jung. Lired: A light-weight real-time fault detection system for edge computing using lstm recurrent neural networks. *Sensors*, 18(7), 2018.
- [4] M. H. Akbar. Perancangan dan pengembangan sistem pengendali ac menggunakan gerakan tangan berbasis lstm dengan implementasi pada esp32, 2025. Vol. 1.
- [5] J. Zhou, L. Qiu, Y. Li, Z. Wang, and Y. Guo. Pupil control intelligent wheelchair based on mediapipe. In *Proceedings of the 35th Chinese Control and Decision Conference (CCDC 2023)*, pages 1656–1660, 2023.
- [6] H. Yoo, I. Goncharenko, and Y. Gu. Real-time dynamic sign language recognition using lstm based on mediapipe hand data. In *2023 International Conference on Consumer Electronics – Taiwan (ICCE-Taiwan)*, pages 17–18, 2023.
- [7] A. Dhiran, A. Kumbhare, A. Patil, M. Vichare, and D. Patel. Real-time hand gesture recognition system using mediapipe, spiking convolution neural network and bi-lstm. In *2nd International Conference on Electronics, Computing, Communication and Control Technology (ICECCC 2025)*, pages 32–35, 2025.
- [8] P. J. Werbos. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [9] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2017.
- [10] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
- [11] C. Lugaresi et al. Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172, 2019.
- [12] P. Priyanka and D. Kumar. Meta-heuristic optimization based convolutional neural network for medical image classification: A survey. In *IEEE 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN 2020)*, pages 752–756, 2020.
- [13] G. Chandrashekar, P. Vaishnavi, D. Neeharika, N. Suchitha, and T. Rohith. Smart building management system using esp32. pages 1962–1966, 2025.
- [14] T. Kim, H. Lee, and Y. Chung. Advanced universal remote controller for home automation and security. *IEEE Transactions on Consumer Electronics*, 56:2537–2542, 2025.

- [15] S. Wolfgang. Raspberry pi based computing prototypes: Design, implementation and performance analysis. In *2023 IEEE International Conference on Electro Information Technology*, pages 59–66, 2023.
- [16] F. Zhang et al. Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214, 2020.

## BIOGRAFI PENULIS



Ali Akbar Alhabsyi, lahir di Ternate pada 17 Februari 2004, merupakan mahasiswa Program Studi S-1 Teknik Komputer, Departemen Teknik Komputer, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember. Penulis menempuh pendidikan menengah atas di SMAN 1 Kota Ternate sebelum melanjutkan studi sarjana di Teknik Komputer ITS.

Pada tugas akhir ini, penulis mengangkat topik kendali AC berbasis gesture tangan menggunakan *Long Short-Term Memory* (LSTM) pada perangkat edge. Selama masa studi, penulis memiliki minat pada bidang pengembangan aplikasi Internet of Things (IoT) dan *web development*. Melalui tugas akhir ini, penulis berfokus pada pengembangan sistem pengenalan gesture berbasis *computer vision*, implementasi model LSTM pada Raspberry Pi, serta integrasi komunikasi MQTT dan IR Blaster berbasis ESP32 untuk kendali perangkat AC.

*[Halaman ini sengaja dikosongkan]*