



## **TUGAS AKHIR - EC224801**

# **KONTROL PRESENTASI BERBASIS POSE TANGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)**

**M. Dafa Raisya Rajwa**

NRP 0721 19 4000 0069

Dosen Pembimbing

**Ahmad Zaini, S.T., M.Sc.**

NIP 19750419 200212 1 003

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

NIP 19680601 199512 1 009

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

Departemen Teknik Komputer

Fakultas Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2023



## **TUGAS AKHIR - EC224801**

# **KONTROL PRESENTASI BERBASIS POSE TANGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)**

**M. Dafa Raisya Rajwa**

NRP 0721 19 4000 0069

Dosen Pembimbing

**Ahmad Zaini, S.T., M.Sc.**

NIP 19750419 200212 1 003

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

NIP 19680601 199512 1 009

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

Departemen Teknik Komputer

Fakultas Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2023

*[Halaman ini sengaja dikosongkan]*



**FINAL PROJECT - EC224801**

## **HAND POSE BASED PRESENTATION CONTROL USING *Convolutional Neural Network (CNN)***

**M. Dafa Raisya Rajwa**

NRP 0721 19 4000 0069

Advisor

**Ahmad Zaini, S.T., M.Sc.**

NIP 19750419 200212 1 003

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

NIP 19680601 199512 1 009

**Undergraduate Study Program of Computer Engineering**

Department of Computer Engineering

Faculty of Faculty of Intelligent Electrical and Informatics Technology

Sepuluh Nopember Institute of Technology

Surabaya

2023

*[Halaman ini sengaja dikosongkan]*

## **LEMBAR PENGESAHAN**

### **KONTROL PRESENTASI BERBASIS POSE TANGAN MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK (CNN)***

#### **TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat  
memperoleh gelar Sarjana Teknik pada  
Program Studi S-1 Teknik Komputer  
Departemen Teknik Komputer

Fakultas Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh: **M. Dafa Raisya Rajwa**

NRP. 0721 19 4000 0069

Disetujui oleh Tim Penguji Tugas Akhir:

Ahmad Zaini, S.T., M.Sc.  
NIP: 19750419 200212 1 003

(Pembimbing I)

.....

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

(Pembimbing II)

.....

Eko Pramunanto, S.T., M.T.  
NIP: 19661203 199412 1 001

(Penguji I)

.....

Reza Fuad Rachmadi, S.T., M.T., Ph.D..  
NIP: 19850403 201212 1 001

(Penguji II)

.....

Mengetahui,  
Kepala Departemen Teknik Komputer FTEIC - ITS

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T..  
NIP. 19700313 199512 1 001

**SURABAYA**  
**Agustus, 2023**

*[Halaman ini sengaja dikosongkan]*

## **APPROVAL SHEET**

### **HAND POSE BASED PRESENTATION CONTROL USING *Convolutional Neural Network (CNN)***

#### **FINAL PROJECT**

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

**By: M. Dafa Raisya Rajwa**

NRP. 0721 19 4000 0069

Approved by Final Project Examiner Team:

Ahmad Zaini, S.T., M.Sc. (Advisor I)  
NIP: 19750419 200212 1 003

.....

Dr. Eko Mulyanto Yuniarno, S.T., M.T. (Co-Advisor II)  
NIP: 19680601 199512 1 009

.....

Eko Pramunanto, S.T., M.T. (Examiner I)  
NIP: 19661203 199412 1 001

.....

Reza Fuad Rachmadi, S.T., M.T., Ph.D. (Examiner II)  
NIP: 19850403 201212 1 001

.....

Acknowledged,  
Head of Computer Engineering Department FTEIC - ITS

Dr. Supeno Mardi Susiki Nugroho, S.T., M.T..  
NIP. 19700313 199512 1 001

**SURABAYA**  
**August, 2023**

*[Halaman ini sengaja dikosongkan]*

## **PERNYATAAN ORISINALITAS**

Yang bertanda tangan dibawah ini:

Nama Mahasiswa / NRP : M. Dafa Raisya Rajwa / 0721 19 4000 0069  
Departemen : Teknik Komputer  
Dosen Pembimbing / NIP : Ahmad Zaini, S.T., M.Sc. / 19750419 200212 1 003

Dengan ini menyatakan bahwa Tugas Akhir dengan judul **"KONTROL PRESENTASI BERBASIS POSE TANGAN MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN)"** adalah hasil karya sendiri, berfsifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

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

Surabaya, August 2023

Mengetahui  
Dosen Pembimbing

Mahasiswa

Ahmad Zaini, S.T., M.Sc.  
NIP. 19750419 200212 1 003

M. Dafa Raisya Rajwa  
NRP. 0721 19 4000 0069

*[Halaman ini sengaja dikosongkan]*

## **STATEMENT OF ORIGINALITY**

The undersigned below:

Name of student / NRP : M. Dafa Raisya Rajwa / 0721 19 4000 0069  
Department : Computer Engineering  
Advisor / NIP : Ahmad Zaini, S.T., M.Sc. / 19750419 200212 1 003

Hereby declared that the Final Project with the title of "HAND POSE BASED PRESENTATION CONTROL USING *Convolutional Neural Network (CNN)*" is the result of my own work, is original, and is written by following the rules of scientific writing.

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

Surabaya, August 2023

Acknowledged  
Advisor

Student

Ahmad Zaini, S.T., M.Sc.  
NIP. 19750419 200212 1 003

M. Dafa Raisya Rajwa  
NRP. 0721 19 4000 0069

*[Halaman ini sengaja dikosongkan]*

## **ABSTRAK**

Nama Mahasiswa : M. Dafa Raisya Rajwa  
Judul Tugas Akhir : KONTROL PRESENTASI BERBASIS POSE TANGAN MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK* (CNN)  
Pembimbing : 1. Ahmad Zaini, S.T., M.Sc.  
                  2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

Dalam melakukan presentasi terdapat beberapa pilihan cara kontrol yang tersedia saat ini, salah satunya menggunakan *keyboard*. Namun, terdapat beberapa kekurangan dalam cara tersebut yaitu kontrol presentasi harus menekan tombol pada laptop secara langsung, sehingga tidak bisa dikendalikan dari jarak jauh dan dirasa kurang interaktif. Dalam penelitian ini, kontrol presentasi dilakukan menggunakan pose tangan sehingga tidak memerlukan kontak langsung lagi dengan perangkat yang digunakan untuk presentasi. Metode yang diterapkan sendiri menggunakan salah satu metode *machine learning* dalam mengolah citra yaitu *Convolutional Neural Network*. Berdasarkan pelaksanaan tugas akhir ini didapatkan hasil bahwa model dapat mendeteksi pose tangan dan terhubung dengan beberapa fungsi kontrol dalam aplikasi presentasi Microsoft PowerPoint menggunakan metode *Convolutional Neural Network* dengan tingkat akurasi tertinggi sebesar 99.00% pada jarak sekitar 40 cm dan kondisi cahaya minimal 40 lx.

Kata Kunci: *Pose Tangan, Presentasi, Convolutional Neural Network*.

*[Halaman ini sengaja dikosongkan]*

## **ABSTRACT**

*Name : M. Dafa Raisya Rajwa  
Title : HAND POSE BASED PRESENTATION CONTROL USING Convolutional Neural Network (CNN)  
Advisors : 1. Ahmad Zaini, S.T., M.Sc.  
2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.*

*When presenting, there are several control options currently available, one of which is using keyboard. However, there are several drawbacks in this method, namely the presentation control must press the buttons on the laptop directly, so it cannot be controlled remotely and feels less interactive. In this study, presentation control was carried out using hand poses so that it did not require direct contact with the device used for presentation. The self-implemented method uses one of the machine learning methods in processing images, namely Convolutional Neural Network. Based on the implementation of this final project, the results show that the model can detect hand poses and is connected to several control functions in the Microsoft PowerPoint presentation application using the Convolutional Neural Network method with an accuracy rate of 99% in ideal conditions. The ideal conditions are at a distance of about 40 cm, light conditions around 40 lx, and using the author's hand in the test.*

*Keywords: Hand pose, Presentation, Convolutional Neural Network.*

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

Puji syukur saya panjatkan atas kehadiran Allah SWT yang Maha Pengasih lagi Maha Penyayang, yang telah melimpahkan rahmat dan hidayahnya, sehingga penulis dapat menyelesaikan laporan tugas akhir tentang Kontrol Presentasi Berbasis Pose Tangan Menggunakan *Convolutional Neural Network (CNN)*.

Penelitian ini telah penulis susun dengan maksimal dengan melibatkan bantuan dari berbagai pihak sehingga dapat memperlancar pembuatan laporan ini. Untuk itu penulis ingin menyampaikan terima kasih kepada pihak-pihak yang telah berkontribusi dalam pembuatan laporan ini yaitu:

1. Bapak Ahmad Zaini, S.T., M.Sc., selaku dosen pembimbing 1, yang senantiasa selalu memberikan arahan dan masukan untuk dapat terus mengembangkan penelitian ini menjadi lebih baik.
2. Bapak Dr. Eko Mulyanto Yuniarno, S.T., M.T., sebagai dosen pembimbing 2, yang juga turut serta dalam memberikan arahan dan berbagai macam dukungan selama proses pengerjaan penelitian ini.
3. Kedua orang tua saya yang selalu memberikan semangat dan dukungan baik secara moril maupun materiil agar dapat menyelesaikan proses ini dengan baik.

Terlepas dari itu semua, penulis menyadari bahwa terdapat kekurangan dalam penelitian ini seperti dalam susunan kalimat atau tata bahasa. Oleh karena itu penulis menerima segala bentuk saran maupun kritik dari pembaca agar penulis dapat berkembang menjadi lebih baik lagi.

Akhir kata penulis berharap dengan adanya tugas akhir mengenai Kontrol Presentasi Berbasis Pose Tangan Menggunakan *Convolutional Neural Network (CNN)* ini dapat memberikan manfaat serta inspirasi terhadap pembaca.

Surabaya, Agustus 2023

M. Dafa Raisya Rajwa

*[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>ix</b>
<b>DAFTAR TABEL</b>	<b>xi</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Permasalahan . . . . .	2
1.3 Batasan Masalah . . . . .	2
1.4 Tujuan . . . . .	2
1.5 Manfaat . . . . .	2
<b>2 TINJAUAN PUSTAKA</b>	<b>5</b>
2.1 Hasil Penelitian Terdahulu . . . . .	5
2.1.1 <i>Power Point Slideshow Navigation Control with Hand Gestures using Hidden Markov Model Method</i> . . . . .	5
2.1.2 <i>Power Point Control Using Hand Gesture Recognition Based on Hog Feature Extraction And K-NN Classification</i> . . . . .	5
2.2 Dasar Teori . . . . .	5
2.2.1 Pose Tangan . . . . .	5
2.2.2 Estimasi Pose . . . . .	6
2.2.3 Kontrol Microsoft PowerPoint . . . . .	7
2.2.4 Convolutional Neural Network . . . . .	8
2.2.5 <i>Confusion Matrix</i> . . . . .	10
<b>3 METODOLOGI</b>	<b>13</b>
3.1 Dataset . . . . .	13

3.2	Estimasi Pose . . . . .	16
3.3	Model Klasifikasi . . . . .	18
3.3.1	Pembuatan Model . . . . .	18
3.3.2	Prediksi Model . . . . .	19
3.4	Kontrol PPT . . . . .	20
3.5	Model Kuesioner <i>User Experience</i> . . . . .	22
<b>4</b>	<b>PENGUJIAN DAN PEMBAHASAN</b>	<b>25</b>
4.1	Skenario Pengujian . . . . .	25
4.2	Hasil Pengujian . . . . .	25
4.2.1	Hasil Pengujian Model . . . . .	25
4.2.2	Hasil Pengujian Menggunakan Variasi Jarak . . . . .	26
4.2.3	Hasil Pengujian Menggunakan Variasi Pencahayaan . . . . .	30
4.2.4	Hasil Pengujian <i>Frame Rate</i> . . . . .	32
4.2.5	Hasil Pengujian dari Responden yang Berbeda . . . . .	33
4.2.6	Pengujian Waktu Respon . . . . .	36
4.2.7	Pengujian Menggunakan Model CNN yang Berbeda . . . . .	39
4.2.8	Pengujian <i>User Experience</i> . . . . .	45
<b>5</b>	<b>PENUTUP</b>	<b>51</b>
5.1	Kesimpulan . . . . .	51
5.2	Saran . . . . .	51
<b>DAFTAR PUSTAKA</b>		<b>59</b>
<b>BIOGRAFI PENULIS</b>		<b>61</b>

## DAFTAR GAMBAR

2.1	Hand Landmarks . . . . .	6
2.2	Ilustrasi Operasi Konvolusi . . . . .	8
2.3	Ilustrasi Max Pooling Layer . . . . .	9
2.4	Ilustrasi Fully connected . . . . .	10
3.1	Blok Diagram Metodologi . . . . .	13
3.2	Desain Model CNN yang digunakan . . . . .	19
3.3	Hasil Prediksi Model Klasifikasi . . . . .	20
3.4	Ilustrasi Penghubungan Model dengan Kontrol PPT . . . . .	21
3.5	Soal Kuesioner . . . . .	24
4.1	Hasil Pengujian Model . . . . .	25
4.2	Kondisi Sangat Terang(800 lx) . . . . .	30
4.3	Kondisi Terang(40 lx) . . . . .	31
4.4	Kondisi Redup(15 lx) . . . . .	31
4.5	Kondisi Gelap(5 lx) . . . . .	31
4.6	Distribusi Hasil Pengujian dengan Variasi Pencahayaan . . . . .	32
4.7	Grafik Pengujian <i>Frame Rate</i> . . . . .	33
4.8	Pengujian Menggunakan Data Citra dari Responden 1 . . . . .	34
4.9	Pengujian Menggunakan Data Citra dari Responden 2 . . . . .	34
4.10	Pengujian Menggunakan Data Citra dari Responden 3 . . . . .	35
4.11	Pengujian Menggunakan Data Citra dari Responden 4 . . . . .	35
4.12	Grafik Pengujian <i>Frame Rate</i> Model Resnet50 . . . . .	40
4.13	Grafik Pengujian <i>Frame Rate</i> Model VGG 16 . . . . .	43
4.14	Grafik Pengujian <i>Frame Rate</i> Model <i>MobileNet</i> . . . . .	45
4.15	Persebaran Data Gender . . . . .	46
4.16	Jawaban Rata-rata Setiap Item Soal . . . . .	47
4.17	Hasil Penilaian <i>User Experience</i> . . . . .	48

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

2.1 Keterangan Nama Setiap Titik <i>Landmark</i> . . . . .	7
3.1 Hasil Pengambilan Dataset . . . . .	13
3.1 Hasil Pengambilan Dataset . . . . .	14
3.1 Hasil Pengambilan Dataset . . . . .	15
3.2 Hasil Estimasi Pose . . . . .	17
3.2 Hasil Estimasi Pose . . . . .	18
3.3 Daftar Nomor Kelas dan Jenis Kontrolnya . . . . .	21
3.4 Daftar <i>Shortcut Keyboard</i> Microsoft PowerPoint . . . . .	21
3.4 Daftar <i>Shortcut Keyboard</i> Microsoft PowerPoint . . . . .	22
4.1 Gambaran Kondisi Pengujian Jarak . . . . .	26
4.1 Gambaran Kondisi Pengujian Jarak . . . . .	27
4.1 Gambaran Kondisi Pengujian Jarak . . . . .	28
4.2 Hasil Pengujian Tanpa Variasi Jarak Dataset . . . . .	28
4.3 Hasil Pengujian Dataset dengan Variasi Jarak . . . . .	29
4.4 Hasil Pengujian dengan Variasi Pencahayaan . . . . .	32
4.5 Hasil Pengujian dengan Variasi Ukuran Tangan . . . . .	36
4.6 Hasil Pengujian Waktu Respon Kelas <i>next slide</i> . . . . .	36
4.7 Hasil Pengujian Waktu Respon Kelas <i>previous slide</i> . . . . .	37
4.8 Hasil Pengujian Waktu Respon Kelas <i>Zoom</i> . . . . .	37
4.9 Hasil Pengujian Waktu Respon Kelas <i>Pointer</i> . . . . .	37
4.9 Hasil Pengujian Waktu Respon Kelas <i>Pointer</i> . . . . .	38
4.10 Hasil Pengujian Waktu Respon Kelas <i>Pen</i> . . . . .	38
4.11 Hasil Pengujian Waktu Respon Kelas <i>Erase</i> . . . . .	38
4.11 Hasil Pengujian Waktu Respon Kelas <i>Erase</i> . . . . .	39
4.12 Hasil Pengujian Menggunakan model ResNet50 . . . . .	39
4.12 Hasil Pengujian Menggunakan model ResNet50 . . . . .	40
4.13 Hasil Pengujian Menggunakan model VGG 16 . . . . .	41
4.13 Hasil Pengujian Menggunakan model VGG 16 . . . . .	42
4.14 Hasil Pengujian Menggunakan Model MobileNet . . . . .	43

4.14 Hasil Pengujian Menggunakan Model MobileNet . . . . .	44
4.15 Keterangan Setiap Nomor Item Soal . . . . .	47
4.15 Keterangan Setiap Nomor Item Soal . . . . .	48

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam berbagai aspek kehidupan, presentasi sangat memberikan pengaruh penting, mulai dari bangku sekolah, hingga berbagai macam bidang pekerjaan. Sebagai contoh kegiatan belajar mengajar di Sekolah pada era teknologi informasi seperti sekarang ini, sudah banyak yang memanfaatkan berbagai media pembelajaran sebagai upaya untuk memperbaiki proses pembelajaran menjadi lebih efektif dan fungsional (Rasmila et al., 2022). Presentasi menjadi cara berbagai orang untuk menyampaikan informasi, ide atau gagasan yang ingin disampaikan. Hal ini berarti bahwa presentasi merupakan salah satu bentuk komunikasi yang digunakan untuk menyampaikan pesan kepada pihak lain melalui tulisan atau lisan, dengan harapan orang dapat memahami apa yang disampaikan oleh pengirim pesan dengan baik. (Wardani, 2016). Namun, terkadang dalam melakukan presentasi dirasa terdapat beberapa kekurangan yang terjadi. Umumnya terdapat tiga cara yang dapat dilakukan dalam mengontrol presentasi yang tersedia saat ini. Cara itu mulai dari kontrol sendiri secara langsung melalui tombol *keyboard* dan *mouse*, meminta bantuan orang lain untuk mengontrol, atau menggunakan perangkat tambahan lain.(Idrees et al., 2021)

Kekurangan kontrol presentasi secara mandiri melalui tombol *keyboard* dan *mouse* adalah presentator harus berada di posisi yang berdekatan dengan laptop, yang berarti tidak bisa dikendalikan secara jarak jauh. Kemudian untuk kekurangan apabila meminta bantuan orang lain untuk mengontrol, adalah sering terjadi kesalahpahaman antara presentator dengan orang yang dimintai bantuan. Dan terakhir untuk kontrol menggunakan perangkat tambahan lain, terdapat kekurangan dari sisi biaya yang dikeluarkan dan sering terjadi kemungkinan kerusakan secara *hardware*. Selain itu, apabila menggunakan perangkat tambahan, maka perlu membawa-bawa perangkat tambahan yang merepotkan. Apalagi perangkat ini memerlukan pengisian atau penggantian baterai apabila perangkat tersebut kehabisan daya (Rahmad et al., 2022).

Padahal terdapat cara lain untuk dapat mengontrol presentasi, yaitu dengan menggunakan pose tangan. Karena penggunaan pose tangan dirasa lebih natural dan intuitif sebagai manusia dalam melakukan komunikasi atau interaksi, Seperti diantaranya adalah menunjuk pada bagian sesuatu yang menjadi *highlight*, atau juga pose lainnya yang menunjukkan keinginan untuk menavigasi dari *slide* satu ke yang lainnya. Selain itu, ini menjadi pilihan lain karena memang penggunaan pose tangan dalam interaksi manusia dengan komputer telah mulai banyak digunakan dalam beberapa tahun terakhir (Indriani et al., 2021). Disisi lain, cara ini tidak memerlukan alat tambahan ataupun memiliki kontak langsung dengan perangkat yang digunakan untuk presentasi (Soroni et al., 2021). Karena dalam penerapannya memanfaatkan *webcam* bawaan pada laptop yang umumnya memiliki tingkat resolusi 480p.

Dalam proses penerapannya nanti, cara pendekatan yang paling baik untuk proses pendeteksian pose tangan ini adalah menggunakan kamera sebagai input dan diproses menggunakan metode CNN. CNN merupakan jenis *neural network* yang didesain untuk mengolah data dua dimensi (Suartika et al., 2016). Penggunaan CNN dilakukan karena metode ini lebih efisien

dibandingkan metode *neural network* lainnya terutama untuk memori dan kompleksitas. Selain itu, terdapat penelitian sebelumnya yang berkaitan dengan topik ini dengan judul "*Power Point slideshow navigation control with hand gestures using Hidden Markov Model method*" yang dimana menghasilkan akurasi sebesar 76,47%. Berdasarkan data tersebut diharapkan penggunaan model CNN dapat menghasilkan akurasi yang lebih baik dibandingkan dengan model yang digunakan dalam penelitian tersebut (AlSaedi & AlAsadi, 2020).

Aplikasi yang dapat digunakan untuk presentasi sendiri ada berbagai macam, salah satunya adalah Microsoft PowerPoint. Aplikasi ini merupakan salah satu program berbasis multimedia yang dirancang khusus untuk menyampaikan presentasi yang mampu menjadikannya sebagai media komunikasi yang menarik (Muthoharoh, 2019). Dengan diadakannya penelitian ini diharapkan hasil klasifikasi tersebut dapat dapat dipetakan kedalam beberapa fungsi kontrol *slide* yang ada pada aplikasi Microsoft PowerPoint.

## 1.2 Rumusan Permasalahan

Berdasarkan apa yang telah dipaparkan dalam latar belakang pada Subbab 1.1, dapat diketahui bahwa terdapat beberapa kekurangan yang dimiliki dalam kontrol presentasi yang tersedia saat ini. Oleh karenanya, diperlukan opsi lain dalam mengontrol presentasi.

## 1.3 Batasan Masalah

Berdasarkan tujuan dari penelitian ini, dibutuhkan batasan-batasan untuk memperjelas penelitian yang dilakukan sebagai berikut:

1. Metode yang digunakan adalah *Convolutional Neural Network*.
2. *Library* yang digunakan adalah MediaPipe.
3. Diterapkan pada aplikasi Microsoft PowerPoint.
4. Diterapkan pada beberapa perintah tertentu yang digunakan dalam kontrol presentasi yaitu *next slide*, *previous slide*, *pointer*, *zoom*, *pen*, dan *erase*.
5. Input citra menggunakan *webcam* dengan resolusi 480p.

## 1.4 Tujuan

Tujuan yang ingin dicapai dari tugas akhir ini adalah dapat melakukan kontrol presentasi PowerPoint berdasarkan hasil klasifikasi pose tangan dengan menggunakan metode *Convolutional Neural Network*.

## 1.5 Manfaat

Terdapat beberapa manfaat yang didapatkan dari tugas akhir ini, yaitu :

1. Bagi Penulis  
Pengerjaan tugas akhir ini memiliki manfaat bagi penulis mulai dari menambah pengetahuan mengenai topik yang diangkat khususnya terkait dengan metode *Convolutional Neural Network*, hingga melatih kemampuan berpikir logis dan sistematis dalam mengatasi masalah.

2. Bagi Institusi

Tugas akhir ini, dapat menjadi referensi tambahan dalam pengembangan teknologi yang berkaitan dengan metode *Convolutional Neural Network* dan pemrosesan klasifikasi pose tangan.

3. Bagi Masyarakat Umum

Bagi masyarakat umum dapat menggunakan penelitian ini untuk diterapkan dalam berbagai macam kesempatan presentasi.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **TINJAUAN PUSTAKA**

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

Terdapat beberapa penelitian sebelumnya yang membahas mengenai topik kontrol PowerPoint menggunakan pose tangan. Berdasarkan penelitian terdahulu ini dapat dijadikan referensi serta dicari gap apa yang dapat ditingkatkan lagi dalam penelitian ini. Berikut ini adalah beberapa penelitian terdahulu tersebut.

##### **2.1.1 *Power Point Slideshow Navigation Control with Hand Gestures using Hidden Markov Model Method***

Dalam penelitian dengan judul "*Power Point slideshow navigation control with hand gestures using Hidden Markov Model method*", yang ditulis oleh C. Rahmad, A. Prasetyo, dan R. A. Baqy, dibahas mengenai kontrol PPT menggunakan pose tangan juga. Namun, dalam penelitian tersebut metode yang digunakan berbeda dengan penelitian ini, yaitu *Markov Model Method*. Selain itu, terdapat beberapa hal yang dapat ditingkatkan dari penelitian tersebut. Pertama adalah tingkat akurasinya diangka 76,47% dengan jarak maksimal 150 cm. Akurasi ini diharapkan dapat lebih ditingkatkan pada penelitian ini. Hal yang dapat ditingkatkan yang kedua adalah pada penelitian ini deteksi yang dilakukan terbatas pada jarak yang sesuai dengan dataset yang diberikan. Sehingga apabila jarak antara pengguna dengan kamera tidak sesuai dengan dataset yang diberikan, akurasinya menurun secara drastis (Rahmad et al., 2022).

##### **2.1.2 *Power Point Control Using Hand Gesture Recognition Based on Hog Feature Extraction And K-NN Classification***

Hasil yang diusulkan dalam penelitian ini adalah membuat program yang memungkinkan manusia untuk mengontrol presentasi PowerPoint tanpa menggunakan perangkat keras apa pun di antaranya, hanya menggunakan kamera dan gestur tangan. Penelitian ini melakukan proses ekstraksi fitur *Histogram-of-Gradient* dari masukan gambar dan mengklasifikasikannya kedalam salah satu dari empat isyarat menggunakan algoritma *K-NN Classification*. Terdapat empat klasifikasi yang dilakukan yaitu *next slide*, *previous slide*, *go to first slide*, dan *End of Presentation*. Keakuratan yang dicapai dalam penelitian ini berada pada kisaran 80% (Salunke & Bharkad, 2017).

#### **2.2 Dasar Teori**

Dasar teori berisi berbagai teori yang memiliki keterikatan dengan penelitian yang dilakukan sebagai dasar pemahaman mengenai konsep tertentu. Berikut adalah dasar teorinya.

##### **2.2.1 Pose Tangan**

Pose menurut Kamus Besar Bahasa Indonesia (KBBI) merupakan gaya atau sikap yang ditampilkan baik ketika dipotret, digambar atau dilukis. Sedangkan tangan merupakan anggota

badan dari siku sampai ke ujung jari atau dari pergelangan sampai ujung jari. Sehingga, dapat dinyatakan bahwa pose tangan merupakan gaya atau sikap yang ditampilkan bagian tubuh dari pergelangan sampai ujung jari dan kemudian dipotret untuk tujuan tertentu.

Pose tangan menjadi salah satu bentuk yang dapat digunakan untuk berinteraksi antar manusia untuk mengirimkan pesan tertentu secara efektif dan mudah dipahami. Oleh karena itu, pose tangan juga menginspirasi untuk digunakan sebagai sarana komunikasi dalam berinteraksi antara manusia dengan komputer.

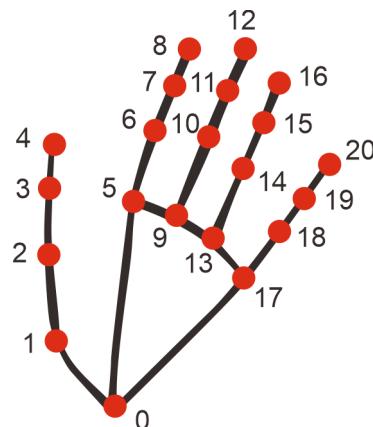
Interaksi Manusia Komputer atau lebih terkenal dengan istilah *Human Computer Interaction* (HCI), bertujuan untuk meningkatkan interaksi antara pengguna dan komputer dengan membuat komputer lebih bermanfaat dan menerima yang pengguna butuhkan. Bidang ini menjadi salah satu bidang yang berkembang pesat selama beberapa tahun terakhir.

Karena pesatnya perkembangan yang terjadi pada bidang ini, akhirnya banyak variasi dan cara yang digunakan sebagai sinyal masukan. Salah satunya adalah pose tangan. Hal ini memungkinkan karena perkembangan perangkat keras seperti kamera yang semakin berkualitas. Tidak hanya itu, bagian yang terpenting dalam perkembangan bidang ini adalah adanya algoritma *machine learning* yang semakin canggih untuk dapat membuat komputer memahami suatu citra yang diterima melalui input kamera.

Pose tangan juga menjadi sarana komunikasi yang dapat menggantikan banyak perangkat keras tambahan. Mekanisme input, yang biasanya diberikan oleh *mouse*, *keyboard*, *remote control* atau panel layar sentuh, bisa jadi diganti dengan menggunakan satu perangkat keras saja yaitu kamera. Karena, cara ini membuat pengguna menjadi lebih dapat leluasa memberikan perintah kontrol dengan isyarat dari pose tangan.

Namun, penerapan interaksi ini memiliki banyak tantangan seperti misalnya, deteksi yang dilakukan dalam intensitas cahaya yang berbeda. Selain itu, bagaimana dapat membuat deteksi dan pelacakan dengan komputasi yang rendah agar bisa menciptakan kesan *real time*. Pengenalan pose juga harus memiliki tingkat keakuratan yang tinggi, agar komputer tidak salah membaca pose yang tidak sesuai dengan keinginan. Terakhir, tantangan yang harus dihadapi adalah terkait dengan membuat keseluruhan kontrol ini bersifat ramah pengguna, agar nyaman dan aman digunakan.

### 2.2.2 Estimasi Pose



Gambar 2.1: Hand Landmarks

Tabel 2.1: Keterangan Nama Setiap Titik *Landmark*

Nomor titik	Keterangan
0	Pergelangan tangan
1	Karpal Jempol
2	Metakarpal Jempol
3	Inter Falang Jempol
4	Tuberocity Jempol
5	Metakarpal Telunjuk
6	Proksimal Palang Telunjuk
7	Distal Palang Telunjuk
8	Tuberocity Telunjuk
9	Metakarpal Jari Tengah
10	Proksimal Palang Jari Tengah
11	Distal Palang Jari Tengah
12	Tuberocity Jari Tengah
13	Metakarpal Jari Manis
14	Proksimal Palang Jari Manis
15	Distal Palang Jari Manis
16	Tuberocity Jari Manis
17	Metakarpal Kelingking
18	Proksimal Palang Kelingking
19	Distal Palang Kelingking
20	Tuberocity Kelingking

Dalam proses deteksi pose tangan ini, penulis menggunakan bantuan dari library *MediaPipe*. MediaPipe adalah kerangka kerja yang memungkinkan untuk membuat *landmark*, atau titik-titik bagian dari kerangka anggota tubuh. Semua titik koordinat dinormalisasi tiga dimensi. MediaPipe menggunakan model *landmark* pose, wajah dan tangan, namun yang digunakan pada penelitian ini hanyalah pada bagian tangan yang dimana terdapat 21 *landmark* dalam tiap *frame* sehingga menghasilkan citra kerangka tangan seperti yang ditunjukkan pada Gambar 2.1. Setiap titik dalam kerangka tersebut merupakan representasi dari setiap bagian anatomi kerangka tangan. Daftar nama bagian dari setiap titik *landmark* dapat dilihat pada Tabel 2.1. Dengan menggunakan library ini dapat langsung menghasilkan prediksi koordinat. Tiap *landmark* yang ada memiliki koordinat x, y, dan z. Dimana untuk nilai x dan y dinormalisasi menjadi lebar dan tinggi dari gambar input yang diterima. Sedangkan z merepresentasikan nilai kedalaman *landmark*, yang nilainya semakin kecil apabila tangan semakin mendekati kamera (Indriani et al., 2021).

### 2.2.3 Kontrol Microsoft PowerPoint

Microsoft Office PowerPoint merupakan sebuah aplikasi komputer untuk presentasi yang dikembangkan oleh Microsoft di dalam paket aplikasi kantor mereka yaitu Microsoft Office (Poerwanti & Mahfud, 2018). PowerPoint pada awalnya dirancang untuk menyediakan visual untuk presentasi kelompok dalam organisasi bisnis. Namun, kemudian berkembang pesat ke sektor lainnya, baik dalam bisnis maupun di luaranya. Didalamnya terdapat berbagai fitur yang menunjang presentasi seperti navigasi berpindah *slide*, *pen tool* untuk mencoret-coret *slide* beserta *erase* untuk dapat menghapusnya. *Pointer* untuk melakukan *highlight* pada bagian

tertentu. Hingga fitur *zoom* untuk memperbesar bagian visual tertentu pada presentasi.

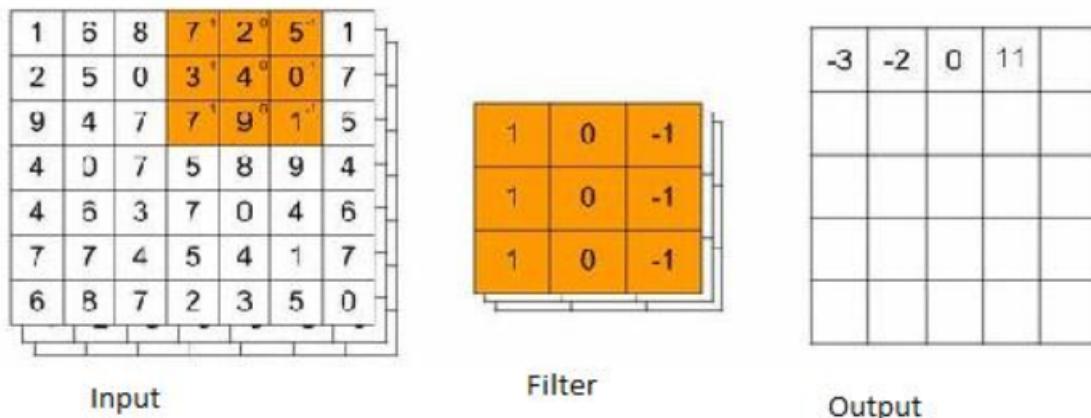
Fitur-fitur tersebut, dapat diakses salah satunya dengan menggunakan *library* bernama *pywin32*. *Pywin32* adalah *library* ekstensi *python* untuk *Windows* yang memungkinkan menggunakan fitur *Application Programming Interface (API) Win32* di *python*. Melalui *library* ini, penggunaan fitur dalam *PowerPoint* bisa diakses melalui *property* ataupun *method* yang ada dalam object yang diinginkan. Tersedia juga dokumentasi untuk objek apa saja beserta *property* dan *method*-nya yang dapat diakses. Selain menggunakan *library* ini, terdapat cara lain yaitu menggunakan *library* yang berfungsi untuk menjalankan fungsi-fungsi yang ada pada *keyboard*. Dikarenakan adanya keuntungan dalam menggunakan *PowerPoint* dimana tersedia opsi untuk mengakses fitur-fitur yang ada menggunakan shortcut tertentu pada *keyboard*, maka dapat digunakan *library* bernama *pyautogui*. Dengan menggunakan kedua cara tersebut, membuat proses menghubungkan program *python* yang dibuat bisa terhubung dengan aplikasi Microsoft *PowerPoint*.

## 2.2.4 Convolutional Neural Network

*Convolutional Neural Network* atau disingkat menjadi CNN merupakan jenis *neural network* yang digunakan dalam memproses data dua dimensi. Penggunaan nama *Convolutional Neural Network* digunakan, karena terdapat operasi konvolusi yang terjadi didalamnya. Proses operasi konvolusi ini, membuat CNN hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti citra (Suartika et al., 2016). Seperti jenis *neural network* lainnya, dalam CNN terdapat *hidden layer*. Namun, belum ada panduan yang betul-betul solid dalam menentukan jumlah *hidden layer* pada sebuah arsitektur model CNN (Bernico, 2018). Dalam *hidden layer* CNN sendiri, terdapat tiga jenis lapisan utama yaitu :

1. Convolution Layer
2. Pooling Layer
3. Fully-connected Layer

### 2.2.4.1 Convolution Layer



Gambar 2.2: Ilustrasi Operasi Konvolusi  
(Lakkavaram et al., 2019)

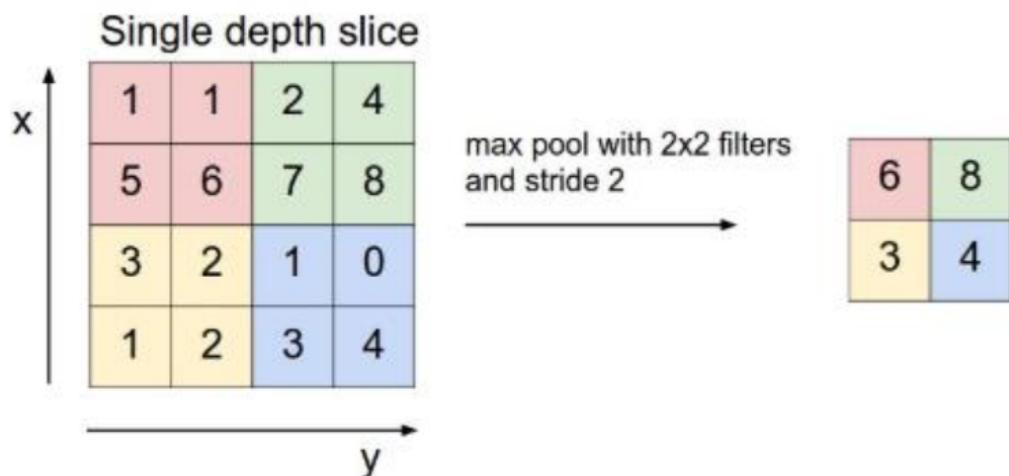
Pada bagian *layer* ini, sebagian besar komputasi terjadi. Hal ini terjadi karena citra input diproses dengan operasi konvolusi dalam rangka menghasilkan filter citra yang berbeda-beda.

Ada tiga komponen penting dalam *layer* ini, yaitu input data, *filter/kernel*, dan *feature map*. Sebagai contoh, Input data yang diberikan dapat memiliki tiga dimensi. Tiga dimensi ini adalah lebar, tinggi, dan kanal. Perlu diketahui juga, bahwa pendekatan deep learning yang digunakan CNN merupakan *supervised learning*. *Supervised learning* adalah teknik pembelajaran menggunakan input dataset yang berlabel (Alom et al., 2019). Kemudian, untuk *filter/kernel* memiliki tugas untuk bergerak melintasi citra dan melakukan operasi "dot" antara input citra dengan matriks dari filter. Hasil operasi ini adalah *output* yang biasa disebut dengan *feature map*. Keseluruhan proses inilah yang dikenal dengan istilah konvolusi. Ilustrasi proses operasi konvolusi ini dapat dilihat pada gambar 2.2.

Pada Gambar 2.2 dapat diketahui bahwa filter bisa berjumlah banyak dan memiliki bentuk berupa matriks. Nilai pada filter tersebut adalah *weight* yang dapat diperbarui nilainya saat proses pelatihan. Dalam *layer* ini terdapat *hyperparameter* yang bisa diatur. Berikut beberapa diantaranya :

1. Jumlah filter. Banyaknya filter yang digunakan berpengaruh terhadap banyaknya kanal pada *feature map*. Jadi, jika terdapat dua filter yang berbeda misalnya, maka *feature map* yang dihasilkan juga dua dengan nilai yang berbeda.
2. *Stride*. *Stride* merupakan besarnya jarak perpindahan yang dilalui filter saat melintasi citra saat proses konvolusi. Hal ini mengakibatkan apabila semakin besar nilai *stride*, maka ukuran *output* yang dihasilkan semakin kecil.
3. *Padding*. *Padding* adalah bagian yang ditambahkan pada elemen terluar citra. Umumnya, nilai yang ditambahkan adalah 0 dengan ukuran yang berbeda-beda. Terdapat tiga jenis *padding* yaitu *valid padding*, *same padding*, dan *full padding*.

#### 2.2.4.2 Pooling Layer



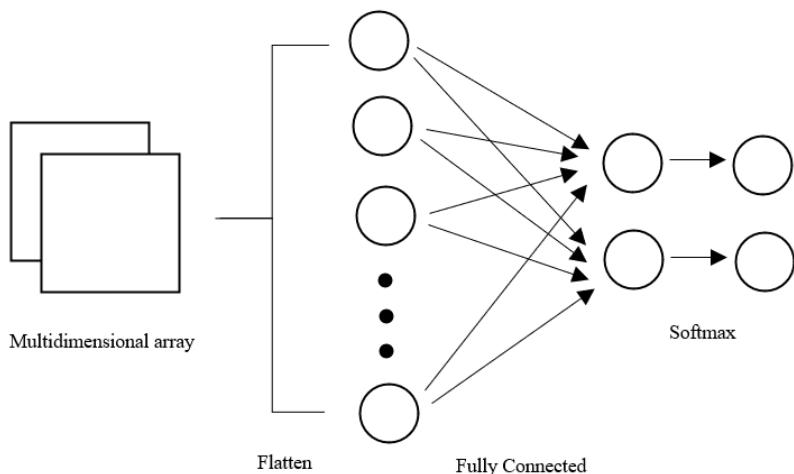
Gambar 2.3: Ilustrasi Max Pooling Layer  
(Lakkavaram et al., 2019)

*Pooling layer* adalah bagian *layer* yang memiliki fungsi untuk *downsampling*. *Downsampling* adalah pengurangan ukuran dimensi dan jumlah parameter. Hal ini dilakukan dengan tujuan mengurangi tingkat kompleksitas untuk *layer* selanjutnya karena proses ini membuat resolusi dari nilai matriks suatu citra menjadi lebih berkurang. Selain itu, *pooling layer* juga berfungsi untuk mengekstraksi fitur dominan. Umumnya, Ada dua jenis *pooling* yang dapat

digunakan yaitu *max pooling* dan *average pooling*. *Max pooling* mengambil nilai maksimum dari citra yang ada dalam suatu *kernel*. *Average pooling* mengambil nilai rata-rata dari citra yang dalam suatu *kernel*.

Sebagai contoh, untuk *max pooling* dapat dilihat dalam gambar 2.3. Pada matriks tersebut, ukurannya dibagi tiap kotak 2x2. Tiap bagian kotak tersebut, dicari nilai paling besarnya dan dijadikan *output* dari nilai matriks yang baru. Begitu juga dengan *average pooling*, prosesnya mirip namun yang membedakan hanya pada nilai yang diambil dari bagian matriks yang dipotongnya. *Output* nilai matriks yang baru dalam *average pooling* didapatkan dari hasil rata-rata dalam kotak matriks yang dibagi.

#### 2.2.4.3 Fully Connected Layer



Gambar 2.4: Ilustrasi Fully connected

Dalam Gambar 2.4 terdapat sebuah ilustrasi mengenai serangkaian proses yang terjadi dalam layer ini. Proses awalnya dimulai dari hasil *feature map* pada convolution dan pooling layer yang memiliki bentuk *multidimensional array*. Bentuk ini perlu dilakukan reshape menjadi sebuah vektor. Tujuannya, agar dapat menjadi input dari *fully connected layer*. Vektor ini juga dapat disebut sebagai *flatten*, dimana hasilnya terhubung dengan beberapa neuron. Setiap koneksi pada neuron tersebut memiliki *weights* yang terkait. Besaran nilai *weights* ini sendiri didapatkan saat proses *training* model.

Saat proses sudah mencapai dalam neuron yang mengandung *weights* tersebut, diterapkanlah fungsi aktivasi. Fungsi dari diterapkannya fungsi aktivasi ini untuk menghitung probabilitas dalam setiap kelas yang ada pada model. Dalam melakukan klasifikasi, fungsi aktivasi yang digunakan biasanya adalah *softmax*. Hasil akhir dari *softmax* merupakan nilai probabilitas dengan rentang dari 0 sampai 1. Dari nilai probabilitas ini biasanya dipilih nilai yang paling besar diantara yang lain dengan juga menerapkan *threshold* yang menjadi nilai minimum untuk menyatakan suatu citra masuk dalam kelas tertentu.

#### 2.2.5 Confusion Matrix

*Confusion matrix* adalah parameter yang digunakan untuk mengukur kinerja model klasifikasi. Salah satu fungsi utama dari *confusion matrix* adalah untuk mengetahui distribusi atau

persebaran dari hasil klasifikasi. Dapat diketahui juga berapa jumlah data yang terdeteksi dalam kelas yang benar atau salah. *Confusion matrix* dapat diterapkan untuk klasifikasi biner dan klasifikasi multi-class. Terdapat beberapa istilah dalam Confusion matrix yaitu TP, TN, FP, dan FN. Namun, penggunaan istilah yang paling sering digunakan dalam menentukan akurasi adalah TP. Berikut adalah penjelasan tiap istilah tersebut (Heydarian et al., 2022).

1. True Positive (TP) TP menunjukkan jumlah sampel positif yang diprediksi secara akurat. Contohnya pada klasifikasi suatu gambar, sistem memprediksi bahwa gambar termasuk kelas A dan memang benar gambar tersebut termasuk kelas A.
2. True Negative (TN) TN menunjukkan jumlah sampel negatif yang diprediksi secara akurat. Contohnya pada klasifikasi suatu gambar, sistem memprediksi bahwa gambar tidak termasuk kelas A dan memang benar gambar tersebut tidak termasuk kelas A.
3. False Positive (FP) FP menunjukkan jumlah sampel yang sebenarnya negatif tetapi diprediksi positif. Contohnya pada klasifikasi suatu gambar, sistem memprediksi bahwa gambar termasuk kelas A, namun ternyata gambar tersebut tidak termasuk kelas A. Itu artinya prediksi tersebut salah.
4. False Negative (FN) FN menunjukkan jumlah sampel yang sebenarnya positif tetapi diprediksi negatif. Contohnya pada klasifikasi suatu gambar, sistem memprediksi bahwa gambar tidak termasuk kelas A, namun ternyata gambar tersebut termasuk kelas A. Itu artinya prediksi tersebut salah.

Hasil dari *confusion matrix* ini dapat ditunjukkan kedalam beberapa metrik untuk mengevaluasi klasifikasi yaitu *precision*, *recall*, dan *f1 score*. *Precision* adalah metrik yang digunakan untuk mengukur tingkat presisi hasil dari klasifikasi. Persamaan yang digunakan untuk mendapatkan nilai *precision* dapat dilihat pada Rumus 2.1.

$$Precision = \frac{TP}{TP + FP} \quad (2.1)$$

Sedangkan, *recall* merupakan metrik yang digunakan untuk mengetahui seberapa banyak objek yang dapat dideteksi oleh model. *Recall* juga bisa disebut *sensitivity* karena menggambarkan tingkat sensitifitas model saat mendekripsi. Melalui *recall* dapat diketahui seberapa besar peluang kasus kategori positif dapat tepat diprediksi secara positif juga. Nilai dari *recall* dapat dicari menggunakan Persamaan 2.2.

$$Recall = \frac{TP}{TP + FN} \quad (2.2)$$

Terakhir terdapat *f1 score*, yaitu metrik yang menunjukkan nilai perbandingan antara rata-rata dari nilai *precision* dengan nilai *recall*. Secara sederhana, dikarenakan *f1 score* memasukkan nilai *precision* dan *recall* kedalam perhitungannya, maka *f1 score* membuat nilai *False Positive* dan *False Negative* juga masuk kedalam penilaian. Sehingga, metrik *f1 score* juga menggambarkan resiko kesalahan dari *False Positive* dan *False Negative*. Rumus menghitung *f1 score* sendiri dapat dilihat pada Persamaan 2.3

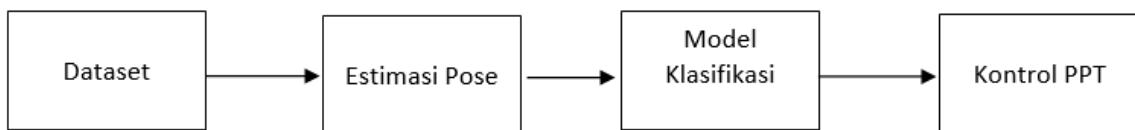
$$F1 Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2.3)$$

*[Halaman ini sengaja dikosongkan]*

## BAB III

### METODOLOGI

Metodologi berisi tentang langkah-langkah yang dilakukan untuk mencapai tujuan dari tugas akhir ini, yang dimana langkah-langkah tersebut digambar dalam blok diagram. Secara umum, gambaran blok diagram metodologi yang digunakan dalam pengerjaan tugas akhir ini adalah sebagai berikut.



Gambar 3.1: Blok Diagram Metodologi

#### 3.1 Dataset

Input citra digunakan sebagai dataset dalam proses *training* data. Proses pengambilan dataset dimulai dengan Input citra video. Proses pengambilannya melalui kamera *webcam*, dengan cara mengambil video dan membuatnya menjadi potongan *frame*. Video yang diambil disesuaikan dengan kelas apa yang ingin diperagakan. Misalnya untuk perintah seperti pengguna "*tool pen*" maka kita harus memperagakan pose *pen* yang telah ditentukan sebelumnya didepan kamera.

Cara memperagakannya adalah dengan mengambilnya dari beberapa jarak dari dengan kamera. Supaya nantinya deteksi perintah ini dapat terbaca dari berbagai sisi jarak kamera juga. Selain jarak juga dilakukan variasi bentuk pose tangan dan pengambilan angle dibandingkan dengan kamera. Karena, satu orang dalam memeragakan pose tangan tidak selalu sama persis bentuknya dengan yang orang lainnya.

Tabel 3.1: Hasil Pengambilan Dataset

No.	Kelas Dataset	Citra Dataset
1	Dataset <i>Next Slide</i>	

Tabel 3.1: Hasil Pengambilan Dataset

No.	Kelas Dataset	Citra Dataset
2	Dataset <i>Previous Slide</i>	
3	Dataset <i>Pointer</i>	
4	Dataset <i>Zoom Left</i>	
5	Dataset <i>Zoom Right</i>	
6	Dataset <i>Pen Left</i>	

Tabel 3.1: Hasil Pengambilan Dataset

No.	Kelas Dataset	Citra Dataset
7	Dataset <i>Pen Right</i>	
8	Dataset <i>Erase</i>	

Jumlah dan kualitas dari dataset ini juga perlu diperhatikan, karena sangat mempengaruhi kinerja atau hasil tingkat akurasi model yang didapat. Dalam tugas akhir ini banyak citra yang digunakan sebanyak 300 citra pada masing-masing kelas. Karena kelas yang dibuat terdapat delapan kelas, maka total banyaknya citra yang digunakan adalah 2400 citra. Hasil dari pengambilan dataset beserta bentuk pose tangan tiap kelasnya, dapat dilihat dalam Tabel 3.1.

Pada prinsipnya, semakin banyak data akan menambah tingkat akurasi dari sebuah model. Namun, belum ditemukan satu aturan khusus yang pasti mengenai jumlah batas minimal data yang dibutuhkan untuk melatih model dengan tingkat hasil akurasi tertentu. Jumlah data tersebut bergantung pada masalah yang diselesaikan (Cho et al., 2016). Oleh karenanya, penentuan berapa banyak jumlah data ini perlu dilihat pula hasil akurasinya. Apabila akurasi dirasa cukup maka jumlah dataset tersebut dapat dinyatakan cukup.

Dalam Tabel 3.1, bisa dilihat bahwa pose yang dibuat berjumlah delapan. Padahal jumlah fungsi yang ingin diterapkan berjumlah enam (*next slide*, *previous slide*, *pointer*, *zoom*, *pen*, dan *erase*). Hal ini dilakukan karena model yang dibuat, diharapkan dapat mendeteksi tidak hanya menggunakan salah satu tangan saja. Namun, bisa digunakan baik menggunakan tangan kanan maupun tangan kiri.

Tetapi, tidak semua pose perlu dataset dengan citra dari tangan kanan dan kiri. Karena beberapa pose memiliki kemiripan antara bentuk tangan kanan dan kiri. Dapat dilihat dalam Tabel 3.1 bahwa pose untuk fungsi *next slide*, *previous slide*, *pointer*, dan *erase* hanya memiliki masing-masing satu kelas. Apabila dilakukan *mirroring* pada citra tersebut hasilnya juga tidak jauh beda. Sehingga, model masih bisa mendeteksi.

Namun, untuk fungsi *pen* dan *zoom* diperlakukan berbeda. Karena bentuk posenya apabila dibandingkan antara menggunakan tangan kanan dengan tangan kiri, terlihat perbedaan yang cukup jauh. Oleh karena itu, cara termudah untuk model dapat mendeteksi tangan kanan maupun kiri dalam pose *pen* dan *zoom* adalah dengan memisahnya menjadi kelas dataset yang

terpisah. Hal ini lah yang membuat jumlah kelas dataset terdapat delapan, walaupun jumlah fungsi kontrol PowerPoint yang ingin diterapkan hanya berjumlah enam.

Ukuran *frame* yang terlihat pada Tabel 3.1 sendiri, perlu dipotong supaya hanya citra dari pose tangannya saja yang ditangkap. Namun, proses ini baru bisa dilakukan jika sudah dapat mendeteksi posisi tangan yaitu pada bagian proses estimasi pose. Citra yang digunakan input merupakan citra berwarna yang memiliki tiga buah kanal warna. Tiga buah kanal ini terdiri dari komponen warna RGB yaitu merah/*Red* (R), green/*Green* (G), dan biru/*Biru* (B) yang memiliki nilai dengan rentang mulai dari 0 sampai 255 (Hidayatullah, 2017).

Sebuah dataset perlu dibagi menjadi dua bagian yaitu *training set* dan *validation set*. Pembagian ini diperlukan agar dapat mencegah overfitting dan untuk mengevaluasi model. Overfitting yang dimaksud disini adalah kondisi yang terjadi saat model memiliki performa yang sangat baik namun hanya terjadi untuk dataset pelatihan saja. Apabila model diberi data yang belum pernah dilihat sebelumnya, maka performa yang dihasilkan menjadi buruk.

Sebagian besar data dari dataset merupakan Training set. Hal ini karena data pada *training set* digunakan untuk pelatihan. Apabila proses pelatihan ini selesai, dilakukan proses evaluasi kinerja dari model menggunakan data dari *validation set*. Jumlah, pembagian banyaknya data pada trainining set dan *validation set* dalam tugas akhir ini adalah 75% : 25%. Proses pengambilan *training set* dan *validation set* juga terpisah. Karena model idealnya tidak boleh dilatih menggunakan sample pada *validation set*, baik sebagian maupun seluruhnya. Sehingga antar *training set* dan *validation set* berisi data yang sepenuhnya berbeda.

## 3.2 Estimasi Pose

Estimasi pose memproses *frame* yang didapat dari hasil pengambilan dataset sebelumnya. Proses ini berguna untuk menentukan *keypoint* untuk membuat kerangka dari pose tangan atau yang disebut *landmark*. Citra tangan dari input kamera yang masih asli, diekstrak menjadi citra yang sudah terdapat kerangka posenya sehingga dapat dianalisa untuk melakukan pengklasifikasian. Dalam proses pendekripsi pose kerangka tangan, penulis menggunakan bantuan sebuah *library* bernama MediaPipe. *MediaPipe Hands* menggunakan *machine learning* untuk menemukan 21 *landmark* 3D tangan hanya dari satu *frame*. Tiap *frame* ini diproses lagi ketahapan selanjutnya bernama ekstraksi fitur.

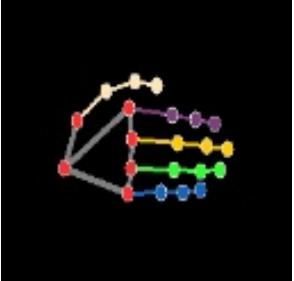
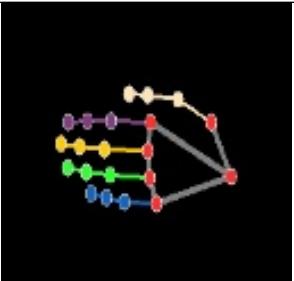
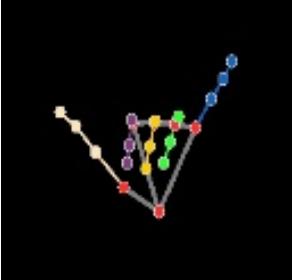
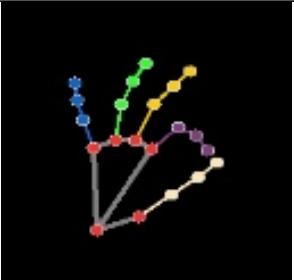
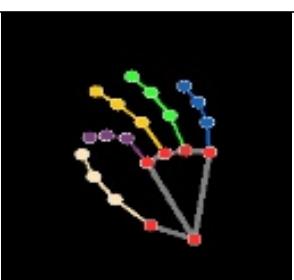
Ekstraksi fitur merupakan bagian dari tahapan yang untuk pengenalan pola. Proses ini dilakukan agar dapat memperoleh informasi yang terkandung dalam suatu citra. Jika informasi ini telah didapatkan, maka informasi tersebut digunakan sebagai acuan untuk membedakan antara satu citra dengan citra yang lainnya. Ekstraksi fitur ini dapat dilakukan, salah satunya dengan cara memisahkan antara objek dengan *background*. Dalam hal ini, deteksi *landmark* tangan dipisah dari *background* aslinya dan diganti dengan latar berwarna hitam. Sehingga, hanya menyisakan citra berupa *landmark* dari pose tangan saja tanpa ada citra tangan yang asli.

Hasil dari ekstraksi fitur tersebut, diproses lagi dengan proses *localization*. Proses ini menentukan lokasi objek dengan membuat kotak pembatas pada daerah citra yang terdapat pose tangannya. Kotak pembatas inilah yang menentukan seberapa besar bagian citra yang dipotong. Apabila gambar sudah diganti dengan *background* hitam, selanjutnya gambar dikumpulkan perkelas untuk dijadikan input proses *training*.

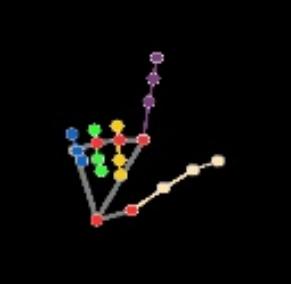
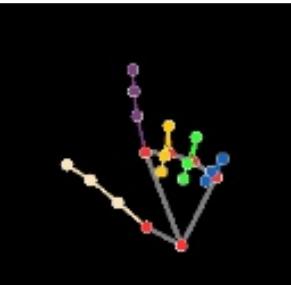
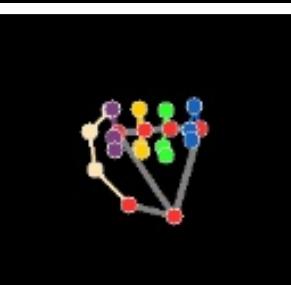
Berdasarkan proses *localization* yang dilakukan, citra dipotong menjadi ukuran *frame* sebesar 128 x 128 piksel dengan posisi menyesuaikan letak *landmark* tangan. Diberikan juga

*padding* sebesar 64 dikiri, atas, kanan, dan bawah citra. Tujuannya agar tidak ada informasi dari ujung-ujung *landmark* yang hilang karena terpotong. Hasilnya seperti yang terlihat pada gambar 3.2. Hasil dari proses ini kemudian disimpan lagi dalam folder sesuai dengan kelas dari pose yang dilakukan.

Tabel 3.2: Hasil Estimasi Pose

No.	Kelas Pose	Hasil Citra Estimasi Pose
1	<i>Next Slide</i>	
2	<i>Previous Slide</i>	
3	<i>Pointer</i>	
4	<i>Zoom Left</i>	
5	<i>Zoom Right</i>	

Tabel 3.2: Hasil Estimasi Pose

No.	Kelas Pose	Hasil Citra Estimasi Pose
6	<i>Pen Left</i>	
7	<i>Pen Right</i>	
8	<i>Erase</i>	

### 3.3 Model Klasifikasi

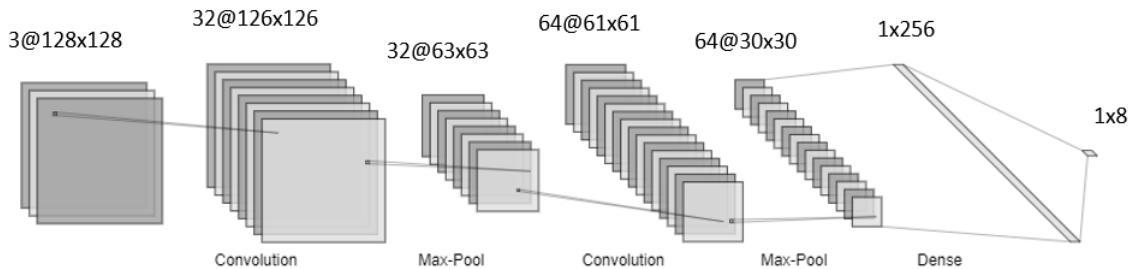
Tujuan dari pembuatan model adalah dapat menghasilkan model klasifikasi yang dapat memberikan *output* berupa hasil deteksi berdasarkan citra yang diinputkan. Berikut ini proses pembuatan model serta cara memprediksinya.

#### 3.3.1 Pembuatan Model

Proses pembuatan model dicapai dengan mengatur *layer* serta *hyperparameter*-nya. Umumnya pengaturan *layer* ini dibagi menjadi tiga bagian yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*. Dalam pembuatannya tipe model yang digunakan adalah *Sequential*. *Sequential* merupakan salah satu cara membuat model dalam *Keras*. Cara ini membuat model dapat dibangun dari satu *layer* ke *layer* lainnya.

Gambaran arsitektur *layer* model CNN yang digunakan dapat dilihat pada Gambar 3.2. *Stride* atau jarak perpindahan konvolusi pada semua *layer* konvolusi sama yaitu satu. Artinya, filter bergeser sebesar satu kotak ukuran matriks pada setiap operasinya. *Padding* yang digunakan pada semua *layer* konvolusi dimodel ini juga sama, yaitu *valid padding*. *Valid padding* berarti tanpa menggunakan *padding* sama sekali. Sehingga, *output* yang dihasilkan (bisa disebut juga dengan *feature map*) berkurang ukurannya masing-masing satu pada kiri, kanan, atas, dan bawah. Jadi, jika ukuran input citranya 128 x 128 maka setelah dikonvolusi hasilnya men-

jadi  $126 \times 126$ . Ukuran filter atau kernelnya yang digunakan tiap konvolusi *layer* dalam model ini sendiri sama yaitu berukuran  $3 \times 3$ . Pada *layer* pertama jumlah filter yang digunakan sebesar 32.



Gambar 3.2: Desain Model CNN yang digunakan

Pada *layer* kedua, dilakukan proses *max pooling*. Ukuran max pooling yang digunakan sebesar  $2 \times 2$ . Karena ukuran yang digunakan  $2 \times 2$ , maka hasil dari pooling *layer* ini menghasilkan matriks dengan ukuran sebesar setengah dari sebelumnya. Apabila *layer* yang dilakukan pooling awalnya sebesar  $126 \times 126$ , maka hasil setelah dilakukan pooling adalah sebesar  $63 \times 63$ .

Dalam *layer* ketiga, dilakukan proses konvolusi lagi. *Stride*, ukuran filter dan *padding* yang digunakan sama dengan *layer* pertama, yang membedakan hanya pada jumlah filternya yang sebesar 64 filter. Layer keempat dilakukan pooling lagi dengan pengaturan sama persis dengan pooling pada *layer* kedua. Hasilnya berupa feature map berukuran  $30 \times 30$ .

*Feature map* yang dihasilkan dari ekstraksi fitur perlu melakukan flatten karena masih berbentuk multidimensional array. Dari proses tersebut didapatkan sebuah vector. Hal ini bertujuan agar bisa digunakan sebagai input dari *fully connected layer*. Lapisan *Fully connected* adalah lapisan dimana semua neuron aktivitas dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya. Lapisan ini digunakan dengan tujuan untuk mengolah data sehingga bisa diklasifikasikan nantinya. Oleh karena itu, pada Gambar 3.2 bagian paling kanan adalah hasil akhir dimana terdapat array berukuran  $1 \times 8$ . Dimana, angka delapan ini menunjukkan jumlah kelas yang ada dalam model ini.

Selain dilakukan pengaturan *layer* yang digunakan, terdapat hyperparameter yang perlu dikonfigurasi, sebelum melakukan *training*. Seperti misalnya adalah jumlah epochs, yang digunakan. Epochs sendiri adalah hyperparameter yang memiliki fungsi dalam menentukan banyaknya pengulangan dalam menjalankan algoritma learning dalam proses *training*. Banyaknya epochs yang digunakan dalam model ini adalah 20 epochs.

### 3.3.2 Prediksi Model

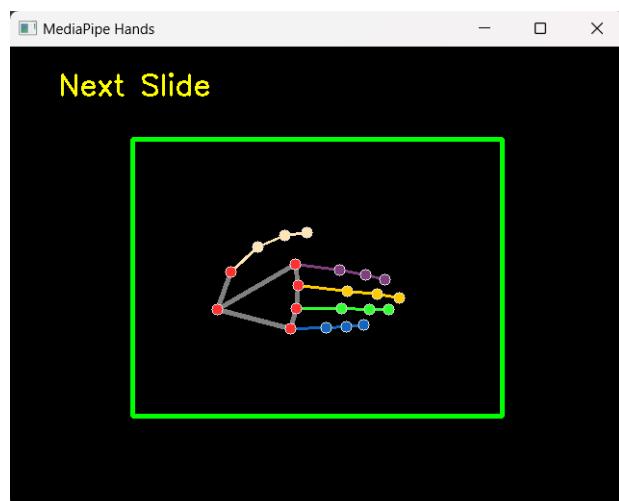
Fungsi dari prediksi model klasifikasi ini adalah untuk menyatakan suatu objek kedalam kategori yang sudah ditentukan dengan cara membandingkan input kamera dengan hasil *training* model mana yang sesuai. Bagian awal dari proses ini adalah deteksi objek tangan yang dimana prosesnya sudah ditangani oleh *library* MediaPipe. Apabila objek tangan terdeteksi,

dilanjut dengan proses yang namanya *localization*. *localization* menjadi bagian dari klasifikasi yang menambahkan citra dengan menentukan lokasi objek dengan bantuan *bounding box* atau kotak pembatas.

Tujuan utama deteksi objek adalah memprediksi lokasi objek dengan *bounding box* atau kotak pembatas dan melakukan klasifikasi objek yang ada pada setiap *bounding box*. Input pada deteksi objek adalah citra yang mengandung satu objek tangan. Sedangkan *output*-nya adalah hasil prediksi lokasi objek dengan *bounding box* dan klasifikasi kelasnya (Elgendi, 2020). Ukuran dari kotak ini disamakan dengan dataset yang di-*training*, yaitu 128 x 128. Termasuk juga, didalamnya diberikan *padding* 64 piksel disemua sisinya seperti yang dilakukan pada Subbab 3.2.

Dalam memprediksi pose yang diinputkan melalui kamera, digunakan sebuah fungsi yang ada pada *library keras*. Output dari fungsi ini adalah array dengan jumlah index sebanyak jumlah kelas yang ada pada model. Sehingga, dari hasil array ini dapat dipilih nilai prediksi yang paling memenuhi untuk klasifikas kelas kita.

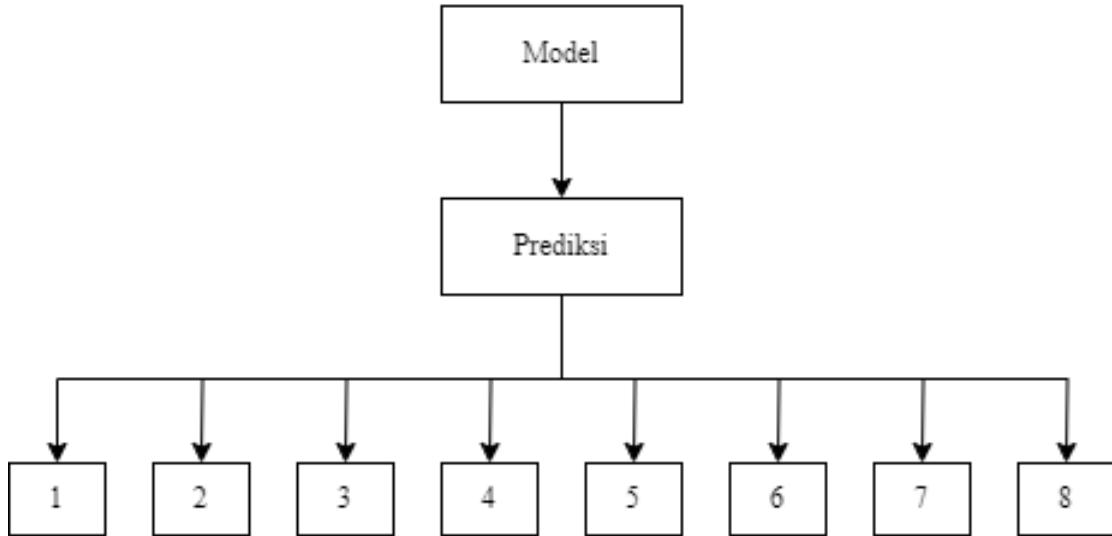
Tampilan secara *real time* hasil prediksi pada program seperti terlihat pada gambar 3.3. Jika input dari kamera mendeteksi ada pose atau gerakan yang sesuai dengan daftar klasifikasi yang ada, maka proses selanjutnya adalah mengimplementasikan hasil klasifikasi tersebut kedalam aplikasi presentasi yang digunakan.



Gambar 3.3: Hasil Prediksi Model Klasifikasi

### 3.4 Kontrol PPT

Tahapan ini menghubungkan hasil klasifikasi yang didapatkan dari model, kedalam aplikasi presentasi yang digunakan. Proses implementasi ini membuat program *python* yang men-erapkan aksi kontrol kedalam *slide* presentasi sesuai dengan pose yang terdeteksi. Seperti yang dijelaskan pada Subbab 3.3.2, Hasil keluaran dari model adalah *array* yang dimana panjangnya sebanyak kelas yang dimiliki. Berdasarkan nilai pada tiap *index* dalam *array* tersebut dicari nilai tertingginya yang berada diatas *threshold*. Sehingga, menghasilkan prediksi citra memasuki kelas antara 1 sampai 8. Gambarannya dapat dilihat pada Gambar 3.4.



Gambar 3.4: Ilustrasi Penghubungan Model dengan Kontrol PPT

Keluaran yang didapatkan dari prediksi model pada Gambar 3.4, berfungsi untuk dijadikan input dari kontrol PPT. Masing-masing angka merepresentasikan jenis kontrol yang diterapkan dalam PPT. Daftar lengkap representasi dari tiap angka dan jenis kontrolnya dapat dilihat dalam Tabel 3.3.

Tabel 3.3: Daftar Nomor Kelas dan Jenis Kontrolnya

Nomor Kelas	Kontrol PPT
1	<i>Erase</i>
2	<i>Next Slide</i>
3	<i>Pen Left</i>
4	<i>Pen Right</i>
5	<i>Pointer</i>
6	<i>Previous Slide</i>
7	<i>Zoom Left</i>
8	<i>Zoom Right</i>

Keluaran prediksi model dan jenis kontrol PPT dalam Tabel 3.3, dihubungkan dengan berbagai macam *shortcut keyboard* yang tersedia dalam fitur Microsoft PowerPoint untuk mengontrol *slide*. Sebagai contoh, untuk mengakses fungsi "pen tool" dapat menekan tombol "ctrl" dan "p" secara bersamaan. Daftar lengkap *shortcut keyboard* untuk kontrol presentasi yang digunakan dalam tugas akhir ini, dapat dilihat pada tabel 3.4. Perlu diketahui juga, dalam penerapan cara ini dibutuhkan *library pyautogui* supaya dapat mengakses *keyboard* pada perangkat komputer yang digunakan.

Tabel 3.4: Daftar *Shortcut Keyboard* Microsoft PowerPoint

Perintah Kontrol	Shortcut Keyboard
<i>Erase</i>	'e'
<i>Next slide</i>	Panah Kanan
<i>Previous slide</i>	Panah Kiri

Tabel 3.4: Daftar *Shortcut Keyboard* Microsoft PowerPoint

Perintah Kontrol	Shortcut Keyboard
Pointer	'ctrl + l'
Pen	'ctrl' + 'p'
Zoom	'ctrl' + 'mouse wheel up'

Beberapa perintah tertentu juga perlu menggunakan *cursor* dalam penggunaannya. Salah satunya adalah tool untuk *pointer*. Hal ini berarti, dalam penerapannya perlu menghubungkan dengan *library cursor* dan mengubah titik koordinatnya supaya bergerak menyesuaikan gerakan tangan. Oleh karena itu, perlu ditentukan satu titik landmark yang menjadi acuan gerak cursor tersebut.

Titik landmark yang dipilih adalah titik ke-8, yaitu bagian ujung telunjuk tangan (daftar titik *landmark* dapat dilihat pada Tabel 2.1). Titik ini dipilih karena secara natural dapat menjadi acuan dalam menunjuk sesuatu. Penggunaan ujung telunjuk dapat dirasa lebih memudahkan pengguna untuk menentukan di titik mana pose ingin diarahkan. Penggunaan titik acuan ini juga diterapkan untuk fungsi yang lainnya, yaitu zoom dan pen.

Penghubungan antara input gambar dengan titik koordinat pergerakan *cursor* dilakukan dengan memanfaatkan kemampuan fitur bawaan dari MediaPipe. Dalam MediaPipe, dapat diambil data koordinat posisi tangan relatif terhadap ukuran panjang dan lebar kamera. Nilai koordinat yang didapatkan memiliki rentang mulai dari 0 sampai 1. Namun, data ini bersifat *mirror* atau terbalik karena diambil melalui kamera. Jadi, misalnya ukuran kamera yang digunakan adalah 640 x 480. Maka koordinat (0, 0) dalam MediaPipe berada pada titik piksel (640, 0). Sedangkan untuk koordinat (1, 1), maka letak pikselnya berada pada (0, 480).

Koordinat yang didapat dari MediaPipe ini dihubungkan dengan ukuran layar dari komputer, supaya dapat menggerakkan *cursor* pada titik yang diinginkan dalam layar. Cara dalam menghubungkannya melalui operasi perkalian antara rentang koordinat yang didapat dari MediaPipe dengan ukuran layar. Hasil ini yang menjadi input untuk menentukan titik *cursor* perlu digerakkan kemana.

Dalam penerapan fungsi lain seperti *pen* dan *zoom* penentuan titik koordinatnya sama persis dengan yang fungsi *pointer*. Pembedanya ada proses perubahan *state*. Perubahan *state* yang dilakukan memiliki tujuan tertentu. Pertama, dapat menghindari perubahan hasil deteksi ketika sedang menggerakkan tangan. Selain itu, saat berubah *state* proses prediksi model untuk klasifikasi juga tidak dijalankan. Sehingga saat sedang menggambar misalnya, diharapkan dapat menurunkan proses komputasi agar tidak menurunkan *frame rate*.

Pada penerapan *pen*, kondisi *state* ini hanya mengaktifkan fiturnya, namun tidak menekan tombol klik. Penekanan tombol klik baru dilakukan ketika bagian ujung jempol dihubungkan dengan jari tengah. Tepatnya pada titik *landmark* ke-10 dan ke-4 (Detail letak titik ini dapat dilihat pada Tabel 2.1). Jika jaraknya lebih kecil dari 0,05 maka program mengirimkan input untuk menekan klik kiri pada *mouse*. Mekanisme tersebut dibuat agar penggunaan fitur *pen* terasa lebih mudah seperti menggunakan *mouse*.

### 3.5 Model Kuesioner *User Experience*

Terdapat beberapa cara dalam menguji *user experience*, Salah satunya dengan *User Experience Questionare* (UEQ). UEQ berisi kuesioner yang mudah serta efisien, dan berguna untuk

mengukur *user experience* pada sebuah produk yang interaktif. *User Experience* disini didefinisikan sebagai persepsi dan tanggapan seseorang yang dihasilkan saat suatu produk tersebut digunakan. Dengan demikian, *User Experience* dipandang sebagai konsep yang mencakup semua jenis reaksi emosional, kognitif atau fisik mengenai produk jadi yang konkret atau bahkan hanya produk yang masih diasumsikan pembuatannya. Penilaian penggunaan produk ini dapat dinilai baik saat sebelum, selama dan setelah digunakan (Hinderks et al., 2019).

UEQ mengukur menggunakan kuesioner yang berisi skala dan mencakup kesan komprehensif tentang (*User Experience*). Terdapat enam skala penilaian akhir dalam UEQ, berikut rinciannya.

1. *Attractiveness*. *Attractiveness* atau daya tarik menilai kesan secara keseluruhan dari suatu produk. Penilaian ini mengukur apakah pengguna cenderung menyukai atau tidak menyukai produk yang diuji. Oleh karenanya, produk harus dibuat untuk terlihat menarik, menyenangkan, ramah pengguna dan menyenangkan.
2. *Perspicuity*. *Perspicuity* atau kejelasan adalah bagian yang menilai seberapa mudah produk untuk dikenali dan dipelajari cara penggunaannya. Sehingga produk harus dibuat untuk bisa melakukan tugas tertentu dengan cepat, efisien, dan dengan cara yang pragmatis.
3. *Efficiency*. *Efficiency* merupakan penilaian yang mengukur apakah pengguna bisa menyelesaikan fungsi atau fitur tertentu dengan melalui usaha yang sedikit mungkin. Selain itu, reaksi dari produk apakah dapat bereaksi dengan cepat atau tidak. Karena itu, produk harus sederhana, dan mudah dipelajari.
4. *Dependability*. *Dependability* atau ketepatan menilai apakah interaksi dengan produk ada dibawah kendali dari pengguna. Penilaian ini berkaitan dengan tingkat keakuratan sehingga interaksi yang dihasilkan sesuai dengan prediksi dari pengguna. Hal ini juga berhubungan dengan tingkat keamanan yang dihasilkan dari produk, karena apabila *output* yang terjadi tidak sesuai dengan yang diharapkan, dapat menimbulkan kesalahan prediksi yang tidak dapat diantisipasi oleh pengguna. Sehingga, item ini menilai apakah produk dapat diprediksi, aman, dan memenuhi harapan pengguna atau tidak.
5. *Stimulation*. *Stimulation* adalah bagian penilaian yang dapat memiliki efek rangsangan perasaan positif. Perasaan yang timbul ini bisa perasaan menarik, menyenangkan, memotivasi, ataupun seru saat menggunakan produk. Oleh karena itu, produk harus dirancang sedemikian rupa hingga dapat menimbulkan efek rangsangan perasaan menarik dan menyenangkan.
6. *Novelty*. *Novelty* atau kebaruan merupakan bagian pengukuran yang menilai seberapa kreatif dan inovatifnya suatu produk. Apabila suatu produk itu memiliki nilai inovasi dan kreatif didalamnya, tentu dapat menimbulkan kesan menarik minat bagi pengguna. Sehingga, produk harus memiliki ide awal yang dirancang secara inovatif dan kreatif, serta menggunakan teknologi yang sesuai dengan perkembangan zaman.

Pertanyaan yang ada pada kuesioner ini berjumlah 26 item. Setiap item ini menunjukkan skala satu sampai tujuh, dimana penilaian positifnya ini tergantung pada letak item tiap nomor soalnya. Sebagai contoh, dapat dilihat pada gambar 3.5 dimana pada item nomor satu menyenangkan ada pada sisi kanan. Sehingga, semakin skalanya kearah kanan, maka semakin positif penilaianya. Namun, pada soal nomor tiga, penilaian kreatif ada pada sisi kiri. Hal ini membuat penilaian positifnya justru ada pada skala yang semakin kecil.

Tujuan dari dibuatnya pertanyaan seperti ini, agar pengguna yang mengisi kuesioner bisa dapat menilai lebih objektif dan teliti. Apabila penilaian positifnya semuanya ada pada sisi

Kuesioner User Experience Tugas Akhir dengan judul  
 "Kontrol Presentasi Berbasis Pose Tangan Menggunakan Convolutional Neural Network"

Nama : ..... Usia : ..... Jenis Kelamin : .....

Saat ini silakan evaluasi produk dengan memilih satu lingkaran tiap baris item.

	1	2	3	4	5	6	7	
menyusahkan	<input type="radio"/>	menyenangkan 1						
tak dapat dipahami	<input type="radio"/>	dapat dipahami 2						
kreatif	<input type="radio"/>	monoton 3						
mudah dipelajari	<input type="radio"/>	sulit dipelajari 4						
bermanfaat	<input type="radio"/>	kurang bermanfaat 5						
membosankan	<input type="radio"/>	seru 6						
tidak menarik	<input type="radio"/>	menarik 7						
tidak dapat diprediksi	<input type="radio"/>	dapat diprediksi 8						
cepat	<input type="radio"/>	lambat 9						
ide baru	<input type="radio"/>	ide lama 10						
menghalangi	<input type="radio"/>	mendukung 11						
baik	<input type="radio"/>	buruk 12						
rumit	<input type="radio"/>	sederhana 13						
tidak disukai	<input type="radio"/>	menggembirakan 14						
lazim	<input type="radio"/>	teknologi terdepan 15						
tidak nyaman	<input type="radio"/>	nyaman 16						
aman	<input type="radio"/>	tidak aman 17						
mengasyikkan	<input type="radio"/>	Tidak mengasyikkan 18						
memenuhi ekspektasi	<input type="radio"/>	tidak memenuhi ekspektasi 19						
tidak efisien	<input type="radio"/>	efisien 20						
jelas	<input type="radio"/>	membingungkan 21						
tidak praktis	<input type="radio"/>	praktis 22						
terorganisasi	<input type="radio"/>	berantakan 23						
attraktif	<input type="radio"/>	tidak attraktif 24						
ramah pengguna	<input type="radio"/>	tidak ramah pengguna 25						
konservatif	<input type="radio"/>	inovatif 26						

Gambar 3.5: Soal Kuesioner

kanan, dikhawatirkan beberapa responden tidak melihat item pada tiap soalnya dan hanya mengisi di sisi yang positif semua atau negatif semua.

Berdasarkan pertanyaan dalam tiap item kuesioner, dikumpulkan dan dianalisis penilaian-nya seperti apa. Dalam penggunaan UEQ, sudah terdapat *tools* yang dapat digunakan untuk menilai *output* dari pengujian ini. *Output*-nya berupa diagram yang dengan nilai tertentu dari setiap enam skala penilaian dalam UEQ yaitu daya tarik, kejelasan, efisiensi, ketepatan, stimu-lasi, dan kebaruan.

## BAB IV

# PENGUJIAN DAN PEMBAHASAN

Dalam bab ini dijelaskan mengenai pengujian yang dilakukan berdasarkan hasil pelaksanaan pada bab 3. Selain itu, juga dibahas dan dianalisa hasil dari pengujiannya. Melalui pengujian dan analisa mengenai hasil penelitian, diharapkan dapat menjadi acuan dalam pengambilan kesimpulan dari pelaksanaan tugas akhir ini.

### 4.1 Skenario Pengujian

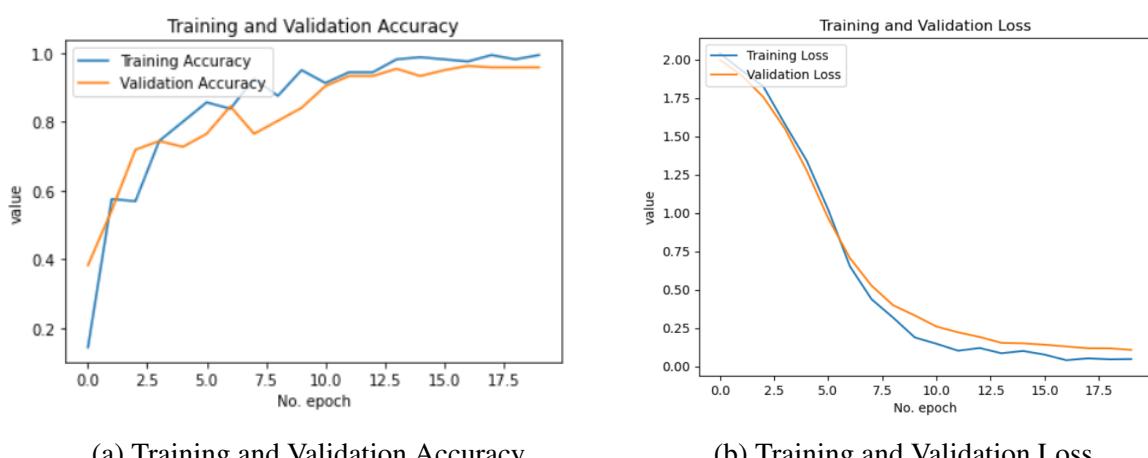
Pada tugas akhir ini dilakukan empat skenario pengujian. Pengujian ini diperlukan untuk mengetahui tingkat keberhasilan atau kesalahan agar dapat ditarik kesimpulan yang sesuai. Berikut ini adalah daftar skenario pengujianya.

1. Pengujian hasil *training* dan validasi model CNN
2. Pengujian dengan menggunakan variasi jarak dataset
3. Pengujian dengan menggunakan variasi Pencahayaan
4. Pengujian tingkat *Frame Rate*
5. Pengujian dengan menggunakan tangan dari responden yang berbeda
6. Pengujian *response time*
7. Pengujian dengan membandingkan model lain (*Resnet50*, *VGG16*, dan *MobileNet*)
8. Pengujian *user experience*

### 4.2 Hasil Pengujian

Hasil pengujian adalah implementasi dari skenario pengujian yang sudah dijelaskan pada bab 4.2, untuk hasil lengkapnya dapat dilihat pada penjelasan berikut ini.

#### 4.2.1 Hasil Pengujian Model



(a) Training and Validation Accuracy

(b) Training and Validation Loss

Gambar 4.1: Hasil Pengujian Model

Pengujian model bertujuan untuk mengukur akurasi dari model yang dibuat berdasarkan hasil *training* yang dilakukan menggunakan *library tensorflow*. Akurasi ini dapat diukur dari hasil perhitungan tiap *epoch* yang dilakukan. Grafik pengujian model perbandingan antara akurasi *training* dengan *validation* pada tiap *epoch*, dapat dilihat pada gambar 4.1. Training yang dimaksud disini adalah hasil model dari dataset yang dilatih. Sedangkan *validation* menggunakan dataset yang belum pernah dilihat sebelumnya (berbeda dengan dataset pada *training*).

Sedangkan untuk perbedaan akurasi dan loss adalah akurasi merupakan banyaknya hasil prediksi yang diklasifikasikan dengan benar. Sedangkan, loss adalah nilai yang menunjukkan perbedaan dari target yang diharapkan. Melalui Grafik 4.1 didapatkan informasi bahwa nilai akurasi tiap *epoch* menunjukkan adanya peningkatan dari waktu ke waktu. Nilai akurasi *training* akhirnya ada pada 99%. Sedangkan, untuk akurasi *validation* akhirnya ada pada kisaran 96%. Performa akurasi model yang hampir sama antara *training* dan validasi adalah hasil yang baik, karena artinya model dapat memprediksi data yang belum pernah dilihat sebelumnya.

Pada *training loss* didapatkan nilai dikisaran 0,03. Sedangkan, *validation loss* dikisaran 0,12. Jarak antara *training* dan *validation* dengan nilai *loss* yang semakin kecil ini menunjukkan prediksi yang error pada model juga kecil.

#### 4.2.2 Hasil Pengujian Menggunakan Variasi Jarak

Skenario pengujian menggunakan variasi jarak bertujuan mengukur akurasi dari setiap jarak yang ditentukan. Hal ini dapat menunjukkan seberapa baik kemampuan model dalam mendekripsi klasifikasi pose tangan pada jarak tertentu. Akurasi pengujian model ini menggunakan data pengujian yang diambil diluar dataset. Variasi jarak yang digunakan sendiri dapat dilihat pada Tabel 4.1 dimana pengujian dilakukan pada jarak 40 cm, 60 cm, 80 cm, 100 cm, 150 cm, 200 cm, 250 cm, dan 300 cm. Data yang telah didapat dapat dihitung berapa banyak data yang benar sesuai hasil prediksi dan berapa banyak yang tidak sesuai hasil prediksi.

Tabel 4.1: Gambaran Kondisi Pengujian Jarak

No.	Jarak (cm)	Gambar Pengujian
1	40	
2	60	

Tabel 4.1: Gambaran Kondisi Pengujian Jarak

No.	Jarak (cm)	Gambar Pengujian
3	80	
4	100	
5	150	
6	200	
7	250	

Tabel 4.1: Gambaran Kondisi Pengujian Jarak

No.	Jarak (cm)	Gambar Pengujian
8	300	

Pengujian variasi jarak ini diuji dengan menggunakan dua model yang memiliki perbedaan pada datasetnya. Model yang pertama menggunakan kumpulan dataset yang dimana posisi pengambilan datanya bersifat statis disatu jarak tertentu. Jarak yang digunakan dalam model pertama ini adalah sejauh 40 cm. Selain model tersebut, dibuat pula model yang kedua. Model kedua ini menggunakan kumpulan dataset yang posisi pengambilannya bervariasi mulai dari 40 cm hingga 100 cm. Tujuan dilakukannya variasi pengambilan dataset ini untuk diuji apakah model ini dapat mendeteksi lebih baik dibandingkan dengan model yang hanya menggunakan dataset dengan satu variasi jarak saja. Berikut ini hasil pengujian menggunakan variasi jarak.

#### 4.2.2.1 Pengujian Menggunakan Dataset Satu Jarak

Hasil akurasi pengujian ini dapat dilihat pada Tabel 4.2. Tingkat akurasi yang didapatkan mengalami penurunan setiap jaraknya diperlebar. Saat jarak 40 cm, akurasi yang didapatkan sebesar 96,00%, saat jarak 60 cm didapatkan akurasi 95,00%. Sedangkan saat jarak 80 cm akurasinya mulai menurun menjadi 75,50 %, dan semakin menurun setiap jarak diperlebar. Tingkat paling parahnya terjadi saat mulai lebih jauh dari 150 cm, akurasi yang dihasilkan dibawah 50% yaitu sebesar 43,00% yang berarti mayoritas data citra yang diprediksi sudah salah.

Tabel 4.2: Hasil Pengujian Tanpa Variasi Jarak Dataset

No.	Jarak(cm)	Precision(%)	Recall(%)	F1 Score(%)	Akurasi(%)
1	40	97	97	96	96,00
2	60	97	96	95	95,00
3	80	79	73	68	75,50
4	100	60	62	56	68,00
5	150	59	61	55	64,00
6	200	50	43	36	43,00
7	250	2	12	3	12,50
8	300	14	13	4	14,00

Hasil distribusinya sendiri cukup beragam namun hampir semuanya terdapat kesalahan prediksi yang masuk kedalam kelas *erase*. Penurunan Berkurangnya akurasi seiring bertambahnya jarak terjadi karena berkurangnya keakuratan titik *landmark* yang dapat dideteksi. Sedangkan dikondisi lebih dari 200 cm, hampir semua prediksinya masuk kedalam kelas *pointer* yang berarti model sudah sama sekali tidak dapat mengklasifikasikan citra. Selain itu, bisa diketahui

juga bahwa jarak yang paling ideal adalah jarak yang sesuai dengan pengambilan dataset yaitu sebesar 40 cm.

#### 4.2.2.2 Pengujian Menggunakan Dataset dengan Jarak Bervariasi

Dalam pengujian ini, model yang digunakan memiliki dataset yang jaraknya berbeda-beda. Penulis mengambil dataset tiap kelasnya dengan cara bergerak mulai dari jarak sekitar 40 cm terhadap kamera, dan perlahan-lahan mundur sampai jarak sekitar 100 cm. Dalam pengujian dengan jarak dataset yang bervariasi ini didapatkan ringkasan hasil data seperti pada Tabel 4.3.

Tabel 4.3: Hasil Pengujian Dataset dengan Variasi Jarak

No.	Jarak(cm)	Precision(%)	Recall(%)	F1 Score(%)	Akurasi(%)
1	40	99	99	99	99,00
2	60	98	97	97	97,50
3	80	98	98	97	97,00
4	100	96	95	94	94,50
5	150	94	91	89	90,50
6	200	77	76	75	77,00
7	250	21	33	23	32,50
8	300	9	13	6	13,00

Akurasi yang dihasilkan secara umum, selalu lebih besar dibandingkan dengan model yang pertama. Saat jarak 40 cm didapatkan akurasi sebesar 99.00%. Sedangkan, pada jarak 60 cm didapatkan akurasi 97.50%. Akurasi yang didapatkan pada saat jarak 80 cm juga tidak terjadi penurunan yang signifikan dengan tingkat akurasinya sebesar 97.00%. Begitu juga saat jarak 100 cm, masih didapatkan akurasi sebesar 94.50%.

Pada jarak 150 cm, akurasi yang dihasilkan sebesar 90,50%. Hal ini menunjukkan bahwa model berhasil mendapatkan akurasi yang lebih besar dari model pada penelitian sebelumnya pada Subbab 2.1.1, yang dimana dengan jarak yang sama hasil akurasinya diangka 76,47%. Data ini sekaligus menjawab permasalahan yang diangkat pada Subbab 1.2 dimana peningkatan akurasi berhasil dilakukan sampai 90,50%.

Sedangkan pada pengujian dengan posisi jarak 200 cm, akurasi yang dihasilkan turun diangka 77,00%. Walaupun terjadi penurunan akurasi yang cukup signifikan, namun model masih menunjukkan bahwa mayoritas data citra dapat diprediksi dengan benar. Performa akurasi model mulai tidak dapat mendeteksi mayoritas data citra, terjadi dijarak lebih dari 200 cm. Terlihat dari pengujian saat jarak 250 cm, akurasi yang didapatkan hanya 32,50%. Begitu juga ketika jarak 300 cm, akurasinya semakin kecil menjadi 13,00%.

Pada penilaian *f1 score* juga menunjukkan nilai yang tidak jauh dari akurasi. Hal ini menunjukkan bahwa jumlah data yang termasuk kedalam *false negative* ataupun *false positif* tergolong cukup kecil. Distribusi persebaran prediksi klasifikasinya juga lebih stabil dan umumnya berada pada kelas yang benar apabila dibandingkan dengan dataset tanpa variasi jarak. Namun, pada jarak lebih jauh dari 200 cm, distribusi prediksinya mayoritas masuk kedalam satu kelas saja yaitu *pointer*. Hal ini menunjukkan bahwa pada jarak tersebut, model memang sudah tidak mampu memprediksi citra dengan baik. Karena hasil kerangka tangan dari estimasi pose Mediapipe juga sudah tidak akurat, dan variasi jarak dataset yang diambil hanya berkisar dijarak 40 cm sampai 100 cm.

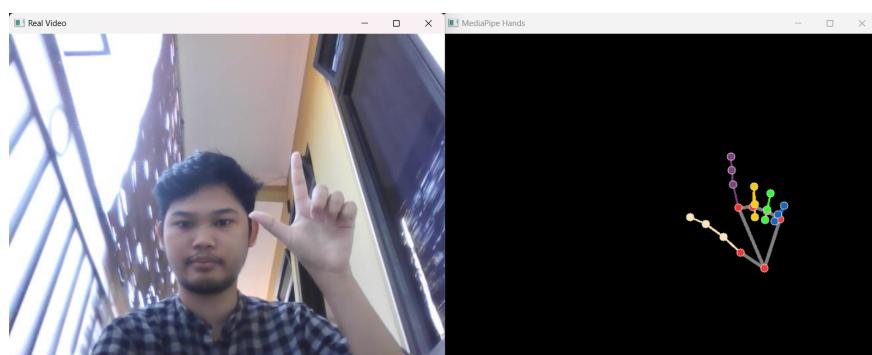
Berdasarkan analisa tersebut diketahui bahwa tingkat akurasi mengalami penurunan setiap jaraknya diperlebar. Namun, akurasi yang dihasilkan masih tetap lebih tinggi dibanding model yang datasetnya tidak ada variasi jarak sama sekali. Jarak ideal yang didapatkan dengan model ini ada pada 40 cm. Sedangkan, batasan terjauh dimana model masih memiliki kemampuan memprediksi mayoritas data citra dengan benar ada pada jarak 200 cm.

#### 4.2.3 Hasil Pengujian Menggunakan Variasi Pencahayaan

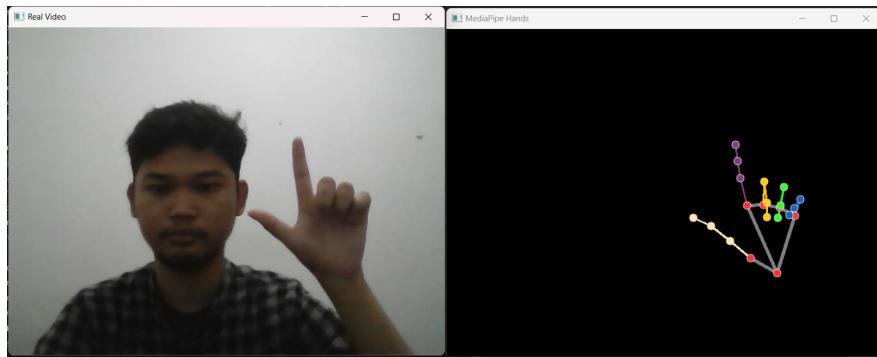
Pengujian menggunakan variasi Pencahayaan yang berbeda-beda ini dilakukan dengan tujuan untuk menguji performa tingkat keterbacaan pose tangan dalam berbagai kondisi cahaya di tempat pengambilan citra. Dalam mengukur pencahayaan ini digunakan alat ukur bernama *Digital Lux Meter*. Perangkat ini berfungsi untuk mengukur banyaknya cahaya masuk menggunakan satuan lux dalam perhitungannya. Satuan lux sendiri adalah besarnya arus cahaya yang datang persatuan luas permukaan. *Digital Lux Meter* yang digunakan sendiri memiliki spesifikasi untuk dapat mengukur cahaya dengan rentang 0 - 200.000 Lux, dengan tingkat akurasi kurang lebih 4% dan resolusi 1 lux. Sehingga, alat ini dapat menjadi acuan tiap kondisi pencahayaan dalam skenario pengujian kali ini.

Variasi pencahayaannya dibagi menjadi empat kondisi. Kondisi pertama adalah sangat terang, dimana pengambilan data dilakukan di luar ruangan pada saat siang hari atau dikisaran 800 lx jika diukur dengan *lux meter*. Kondisi kedua adalah didalam ruangan yang mendapat cahaya lampu dengan cerah. Jika diukur dengan *lux meter* nilainya pada kisaran 40 lx. Kondisi ketiga adalah pencahayaan yang redup, dimana pengujian diambil tetap dalam ruangan namun kondisi lampu mati dan hanya memanfaatkan cahaya matahari dari celah-celah jendela dan pintu ruangan. Apabila diukur dengan *lux meter* didapatkan angka sebesar 15 lx. Kondisi terakhir adalah kondisi pencahayaan gelap, dimana pengujian dilakukan dalam ruangan tanpa cahaya lampu dan matahari. Sumber cahaya yang ada hanya dari layar laptop. Dalam kondisi ini didapatkan angka pada *lux meter* sebesar 5 lx. Gambaran kondisi pengambilan data pengujian untuk kondisi sangat terang dapat dilihat Gambar 4.2, kondisi terang pada gambar 4.3, kondisi redup pada gambar 4.4, dan kondisi gelap pada gambar 4.5.

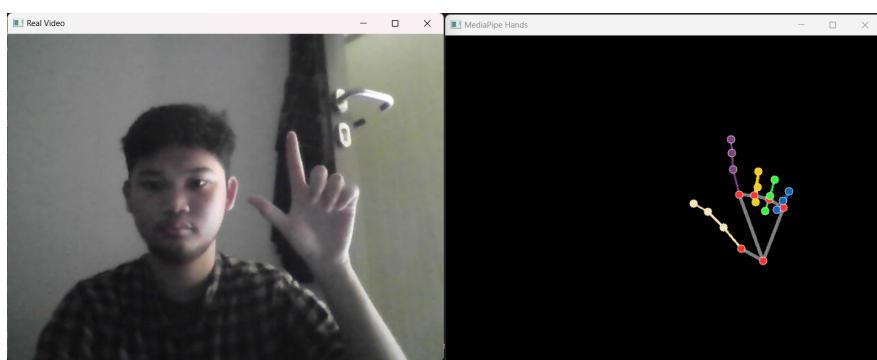
Persebaran hasil klasifikasi yang benar dan salah untuk pencahayaan kondisi gelap ada pada Gambar 4.6d. Hasil klasifikasi dimana citra yang sebenarnya termasuk klasifikasi '*pointer*' tapi salah terdeteksi kedalam '*erase*'. Hal ini bisa terjadi karena dikondisi gelap terkadang ada bagian yang terjadi kesalahan prediksi. Pada citra '*pointer*' misalnya, bagian jari kelingking dan jempol yang tidak terdeteksi. Sehingga, pose pointer jadi terlihat seperti pose '*erase*'. Hal inilah yang menyebabkan terdapat 14 citra '*pointer*' yang salah terdeteksi kedalam kelas '*erase*'.



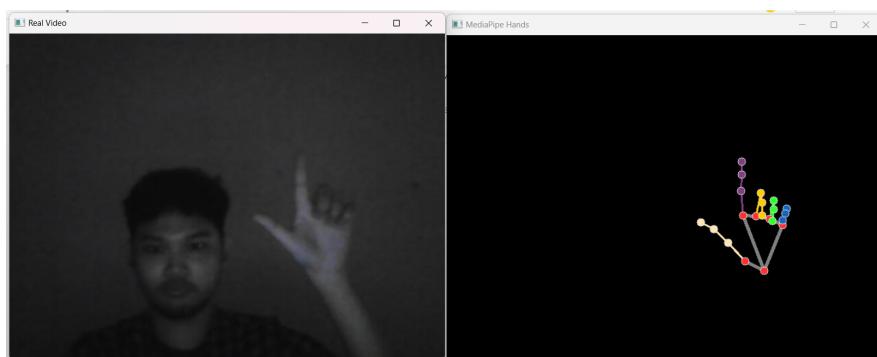
Gambar 4.2: Kondisi Sangat Terang(800 lx)



Gambar 4.3: Kondisi Terang(40 lx)



Gambar 4.4: Kondisi Redup(15 lx)

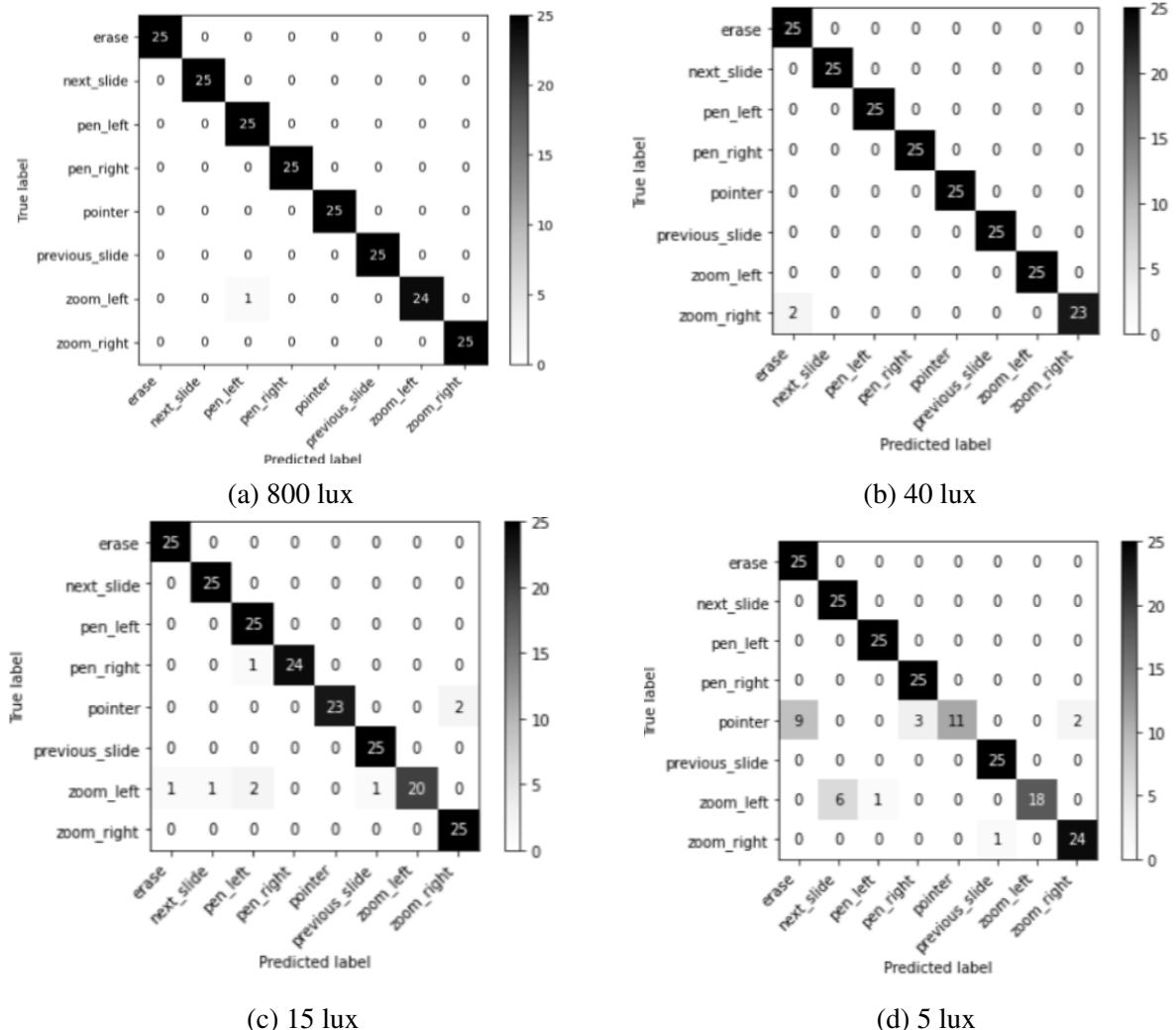


Gambar 4.5: Kondisi Gelap(5 lx)

Berdasarkan pengujian yang dilakukan, didapatkan data seperti yang tertera dalam tabel 4.4. Dalam data tersebut menunjukkan bahwa dalam kondisi cahaya terang dan sangat terang, tingkat akurasi yang dicapai dikisaran 99%. Sedangkan untuk kondisi cahaya redup akurasinya yang didapatkan sebesar 96.00%. Terakhir, dalam kondisi pencahayaan gelap akurasinya sebesar 89.00%. Melalui data tersebut memang terjadi penurunan akurasi saat setiap kondisi pencahayaan semakin menurun.

Namun, penurunan akurasi yang terjadi tidak terlalu signifikan, bahkan sampai dalam kondisi pencahayaan sangat gelap sekalipun. Hal ini lebih banyak dipengaruhi faktor kemampuan dalam *library* MediaPipe. Selama MediaPipe masih dapat mendeteksi *landmark* tangan dengan tepat, maka model klasifikasi pun juga tetap mampu memprediksi pose dengan tepat juga. Karena, yang dibaca oleh model adalah *landmark* tangan yang sudah diproses MediaPipe,

bukan gambar pose tangan langsung dari kamera.



Gambar 4.6: Distribusi Hasil Pengujian dengan Variasi Pencahayaan

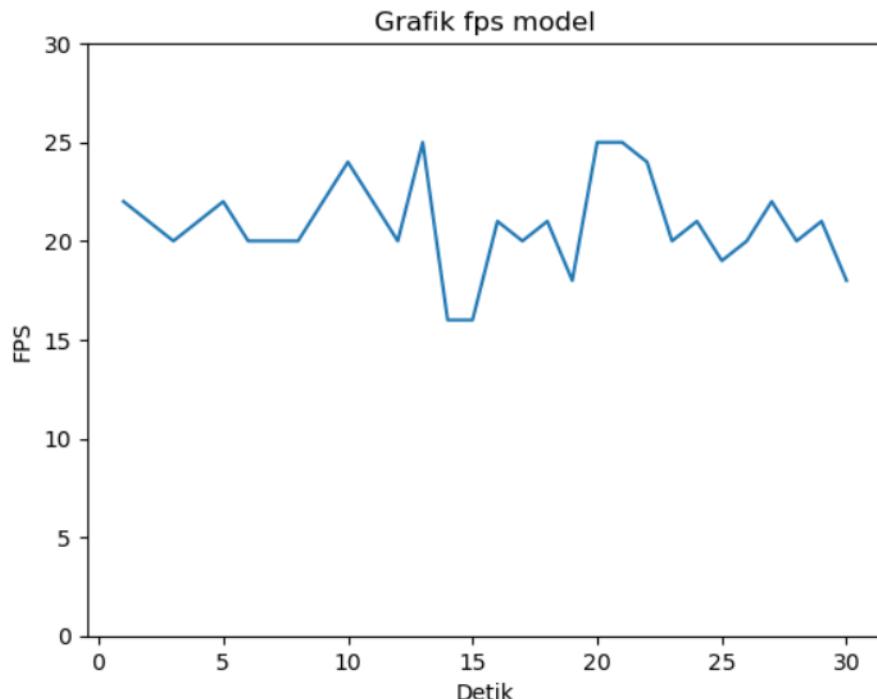
Tabel 4.4: Hasil Pengujian dengan Variasi Pencahayaan

No.	Cahaya	Data Terdeteksi Benar	Data Terdeteksi Salah	Akurasi(%)
1	Sangat Terang (800 lx)	199	1	99.50%
2	Terang (40 lx)	198	2	99.00%
3	Redup (15 lx)	192	8	96.00%
4	Gelap (5 lx)	178	22	89.00%

#### 4.2.4 Hasil Pengujian *Frame Rate*

*Frame Rate* pada umumnya dinyatakan dalam satuan per detik atau disebut sebagai *Frame Per Second (FPS)*. FPS ini merupakan pengukuran seberapa banyak jumlah *frame* yang ada dalam satu detik. Pengujian *Frame Rate* dianggap penting karena menjadi salah satu penilaian

performa dari model yang dibuat. Selain itu, *Frame Rate* juga sangat mempengaruhi penilaian *user experience*. Apabila semakin kecil nilainya maka gerakan semakin terlihat patah-patah, dan respon yang diberikan semakin lambat. Sebaliknya, semakin besar nilainya maka gerakan yang dihasilkan lebih lancar dan responnya semakin cepat.



Gambar 4.7: Grafik Pengujian *Frame Rate*

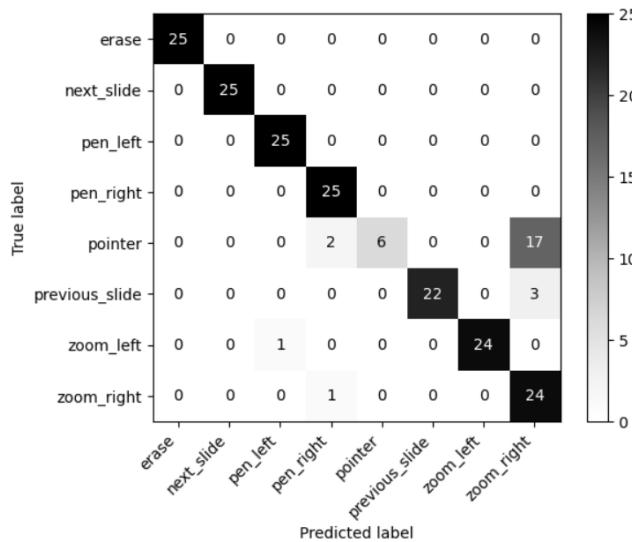
Pada pengujian kali ini, perhitungan *frame rate* diuji dengan mengambil data nilai FPS setiap detiknya selama 30 detik pada kondisi ideal yaitu cahaya terang dan jarak sekitar 40 cm - 60 cm. Hasil pengujian *frame rate* dapat dilihat pada gambar 4.7. Berdasarkan grafik pada Gambar 4.7, didapatkan nilai rata-rata FPS yang didapatkan sebesar 20.86 FPS, dengan rincian nilai tertingginya bisa mencapai 25 FPS, dan titik terendahnya mendekati 15 FPS. Berdasarkan data ini dapat dinyatakan bahwa performa model memungkinkan untuk masih dapat mendeteksi pose tangan dengan baik.

#### 4.2.5 Hasil Pengujian dari Responden yang Berbeda

Saat pengambilan dataset untuk proses training model, data yang digunakan hanya menggunakan pose tangan penulis saja. Sehingga, perlu dilakukan pengujian apabila pose tangan yang dideteksi merupakan tangan selain penulis yang mungkin secara ukuran berbeda. Hal ini bertujuan untuk mengetahui tingkat akurasi model apabila tangan yang dideteksi memiliki ukuran atau bentuk yang tidak sama dengan tangan penulis yang dijadikan dataset.

Dalam pengujian ini, kondisi pengambilan data responden dilakukan dikondisi ideal. Kondisi ideal yang dimaksud disini adalah kondisi dengan tingkat akurasi paling tinggi yang sudah diuji dalam skenario pengujian pada subbab 4.2.2 dan 4.2.3. Berdasarkan data tersebut, didapatkan bahwa kondisi paling ideal adalah saat jarak tangan dengan kamera diangka 40 cm. Sedangkan, untuk kondisi cahayanya dilakukan didalam ruangan yang terang (sekitar 40 lx). Berikut ini adalah hasil pengujian.

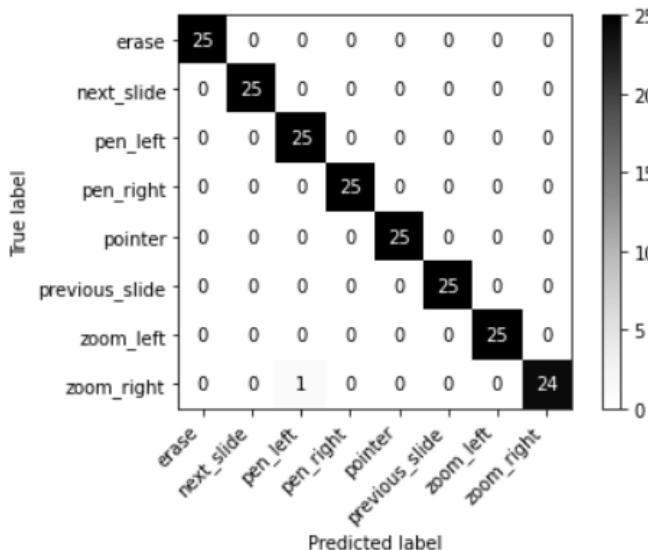
#### 4.2.5.1 Pengujian Menggunakan Data Citra dari Responden 1



Gambar 4.8: Pengujian Menggunakan Data Citra dari Responden 1

Pada saat pengambilan data citra, responden 1 diposisikan untuk berada pada kondisi dengan tingkat akurasi paling ideal, dan diambil data citra sebanyak 25 frame pada tiap kelasnya, yang berarti total data citra yang diambil sebanyak 200 data. Berdasarkan gambar 4.8 didapatkan tingkat akurasi yang dihasilkan sebesar 88.00%. Hal ini berarti menunjukkan adanya penurunan akurasi dari model dengan input citra dari penulis. Namun, penurunan yang terjadi tidak terlalu jauh, sehingga masih dapat terdeteksi dengan akurasi cukup baik.

#### 4.2.5.2 Pengujian Menggunakan Data Citra dari Responden 2

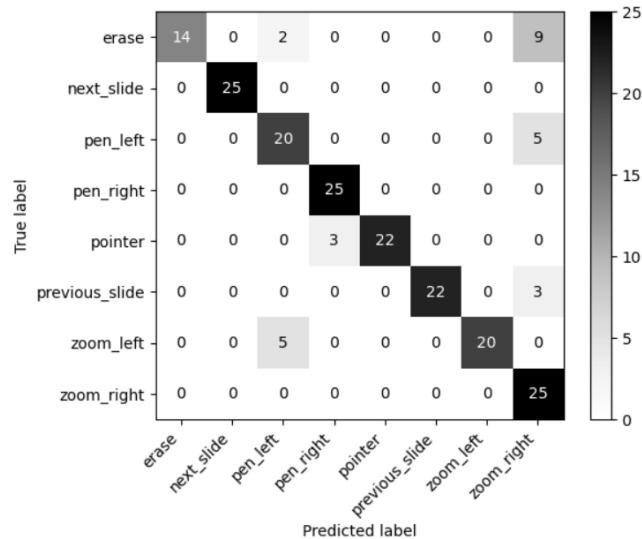


Gambar 4.9: Pengujian Menggunakan Data Citra dari Responden 2

Kondisi pengambilan data citra untuk responden 2, sama dengan responden satu yaitu diposisikan untuk berada pada kondisi dengan tingkat akurasi paling ideal. Jumlah data citra

yang diambil juga sebanyak 25 *frame* pada tiap kelasnya. Hasilnya dapat dilihat pada gambar 4.9. Didapatkan tingkat akurasi yang dihasilkan sebesar 99.50%. Akurasi pada responden 2 ini ternyata mirip dengan input citra dari tangan penulis. Sehingga, dapat dinyatakan model bisa memprediksi dengan baik walaupun diberi input citra tangan yang berbeda.

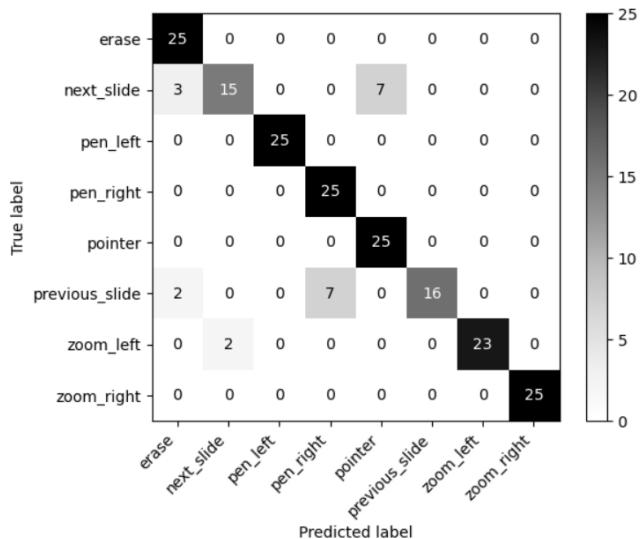
#### 4.2.5.3 Pengujian Menggunakan Data Citra dari Responden 3



Gambar 4.10: Pengujian Menggunakan Data Citra dari Responden 3

Pengujian pada responden 3 didapatkan hasil seperti yang terlihat pada gambar 4.10. Didapatkan tingkat akurasi yang dihasilkan sebesar 86,50%. Sama dengan responden 1 terdapat adanya penurunan akurasi dari model dengan input citra dari penulis, walaupun masih lebih besar dari 80%.

#### 4.2.5.4 Pengujian Menggunakan Data Citra dari Responden 4



Gambar 4.11: Pengujian Menggunakan Data Citra dari Responden 4

Proses pengambilan data citra responden 4 masih sama responden sebelumnya, yang dilakukan pada kondisi dengan tingkat akurasi paling ideal. Jumlah data yang digunakan juga sama persis. Hasilnya dapat dilihat pada gambar 4.11. Didapatkan tingkat akurasi yang dihasilkan sebesar 89,50%. Hal ini berarti menunjukkan adanya penurunan akurasi dari model dengan input citra dari penulis, walaupun akurasinya masih mendekati 90%.

Berdasarkan data dari keempat responden yang diambil, dapat diambil ringkasan seperti yang terlihat pada tabel 4.5. Memang terjadi adanya penurunan akurasi apabila dibandingkan antara tangan penulis dengan menggunakan tangan responden. Namun, penurunan yang terjadi tidak signifikan, dimana saat menggunakan tangan penulis akurasi yang didapatkan sebesar 99,00%, sedangkan saat menggunakan tangan responden lain didapatkan rata-rata akurasi sebesar 90,00%.

Tabel 4.5: Hasil Pengujian dengan Variasi Ukuran Tangan

No.	Responden	Data Terdeteksi Benar	Data Terdeteksi Salah	Akurasi(%)
1	Responden 1	179	21	88,00
2	Responden 2	168	32	99,50
3	Responden 3	190	10	86,50
4	Responden 4	190	10	89,50

## 4.2.6 Pengujian Waktu Respon

Pengujian *response time* dilakukan dengan tujuan untuk mengetahui seberapa cepat jarak waktu antara prediksi tangan dengan jalannya fungsi pada PowerPoint. Hal ini penting karena dapat mempengaruhi *user experience* saat mengontrol presentasi. Hasil pengujian dibagi dalam tiap kelas. Pada kelas *next slide* hasilnya dapat dilihat pada Tabel 4.6, sedangkan untuk kelas *previous slide* ada pada Tabel 4.7. Pada kelas *next slide* waktu respon rata-ratanya sebesar 138,4 *milliseconds*. Kelas *previous slide* didapatkan rata-rata waktunya sebesar 139,7. Rata-rata waktu respon *next slide* dan *previous slide* tidak terjadi perbedaan yang jauh karena penerapan fungsinya yang mirip.

Tabel 4.6: Hasil Pengujian Waktu Respon Kelas *next slide*

No.	Jenis Kontrol	Waktu Respon (ms)
1	<i>Next Slide</i>	267
2	<i>Next Slide</i>	100
3	<i>Next Slide</i>	133
4	<i>Next Slide</i>	117
5	<i>Next Slide</i>	100
6	<i>Next Slide</i>	183
7	<i>Next Slide</i>	100
8	<i>Next Slide</i>	167
9	<i>Next Slide</i>	117
10	<i>Next Slide</i>	100
Rata-Rata		138,4

Tabel 4.7: Hasil Pengujian Waktu Respon Kelas *previous slide*

No.	Jenis Kontrol	Waktu Respon (ms)
1	<i>Previous Slide</i>	100
2	<i>Previous Slide</i>	117
3	<i>Previous Slide</i>	150
4	<i>Previous Slide</i>	183
5	<i>Previous Slide</i>	100
6	<i>Previous Slide</i>	167
7	<i>Previous Slide</i>	117
8	<i>Previous Slide</i>	183
9	<i>Previous Slide</i>	180
10	<i>Previous Slide</i>	100
Rata-Rata		139,7

Pada kelas *zoom*, hasil waktu responnya dapat dilihat pada Tabel 4.8. Rata-rata waktu respon yang didapatkan adalah sebesar *238,7 miliseconds*. Hasil ini lebih lama dibandingkan kelas *next slide* ataupun *previous slide* karena penerapannya yang lebih memerlukan beberapa fungsi untuk dijalankan.

Tabel 4.8: Hasil Pengujian Waktu Respon Kelas *Zoom*

No.	Jenis Kontrol	Waktu Respon (ms)
1	<i>Zoom</i>	300
2	<i>Zoom</i>	250
3	<i>Zoom</i>	217
4	<i>Zoom</i>	233
5	<i>Zoom</i>	150
6	<i>Zoom</i>	267
7	<i>Zoom</i>	220
8	<i>Zoom</i>	250
9	<i>Zoom</i>	283
10	<i>Zoom</i>	217
Rata-Rata		238,7

Dalam waktu respon kelas *pointer*, hasil yang didapatkan dapat dilihat dalam Tabel 4.9. Rata-rata yang didapatkan untuk waktu responnya adalah sebesar *171 miliseconds*. Hasil ini tidak jauh beda dengan waktu respon kontrol navigasi karena proses penerapannya menerapkan jumlah fungsi yang mirip sehingga secara komputasi tidak terlalu berat.

Tabel 4.9: Hasil Pengujian Waktu Respon Kelas *Pointer*

No.	Jenis Kontrol	Waktu Respon (ms)
1	<i>Pointer</i>	117
2	<i>Pointer</i>	100
3	<i>Pointer</i>	133
4	<i>Pointer</i>	283

Tabel 4.9: Hasil Pengujian Waktu Respon Kelas *Pointer*

No.	Jenis Kontrol	Waktu Respon (ms)
5	<i>Pointer</i>	110
6	<i>Pointer</i>	167
7	<i>Pointer</i>	250
8	<i>Pointer</i>	183
9	<i>Pointer</i>	267
10	<i>Pointer</i>	100
Rata-Rata		171

Waktu respon untuk kelas *pen*, dapat dilihat dalam Tabel 4.10, yang dimana hasil rata-rata waktu responnya diangka 292 *miliseconds*. Kelas ini memiliki waktu respon paling lama dibandingkan yang lain karena proses komputasi yang terjadi lebih banyak. Terdapat beberapa fungsi yang dipanggil serta dilakukan juga proses *tracking* titik saat sedang menggambar.

Tabel 4.10: Hasil Pengujian Waktu Respon Kelas *Pen*

No.	Jenis Kontrol	Waktu Respon (ms)
1	<i>Pen</i>	333
2	<i>Pen</i>	217
3	<i>Pen</i>	220
4	<i>Pen</i>	383
5	<i>Pen</i>	350
6	<i>Pen</i>	217
7	<i>Pen</i>	283
8	<i>Pen</i>	267
9	<i>Pen</i>	317
10	<i>Pen</i>	333
Rata-Rata		292

Pada kelas *erase*, waktu responnya dapat dilihat dalam Tabel 4.11. Waktu rata-rata responnya adalah sebesar 131,4 *miliseconds*. Kelas ini menjadi kelas dengan waktu respon tercepat karena memiliki fungsi yang jauh lebih simpel sehingga tidak banyak terjadi proses komputasi apabila dibandingkan dengan kelas yang lain.

Tabel 4.11: Hasil Pengujian Waktu Respon Kelas *Erase*

No.	Jenis Kontrol	Waktu Respon (ms)
1	<i>Erase</i>	100
2	<i>Erase</i>	112
3	<i>Erase</i>	105
4	<i>Erase</i>	117
5	<i>Erase</i>	150
6	<i>Erase</i>	167
7	<i>Erase</i>	180
8	<i>Erase</i>	100
9	<i>Erase</i>	150

Tabel 4.11: Hasil Pengujian Waktu Respon Kelas *Erase*

No.	Jenis Kontrol	Waktu Respon (ms)
10	<i>Erase</i>	133
	Rata-Rata	131,4

#### 4.2.7 Pengujian Menggunakan Model CNN yang Berbeda

Pengujian ini bertujuan untuk menguji performa apabila menggunakan model yang lain. Terdapat tiga model yang digunakan yaitu Resnet50, VGG16, dan MobileNet. Pengujian ini dilakukan dengan menguji semua variasi yang sama dengan model yang penulis buat sendiri. Berikut adalah hasil pengujian tiap modelnya.

##### 4.2.7.1 Pengujian Menggunakan Model Resnet50

Pada pengujian pertama, model yang digunakan untuk diuji performanya adalah ResNet50. Model ResNet diusulkan dengan penggunaan metode bernama *Deep Residual Learning for Image Recognition* oleh Kaiming He, Xiangyu Zhang, Shaoqing Ren dan Jian Sun. Model ini memiliki jumlah *layer* yang jauh lebih banyak dibandingkan model yang dibuat penulis, yaitu berjumlah 50 *layer* (He et al., 2015). Hasil pengujinya dapat dilihat pada Tabel 4.12.

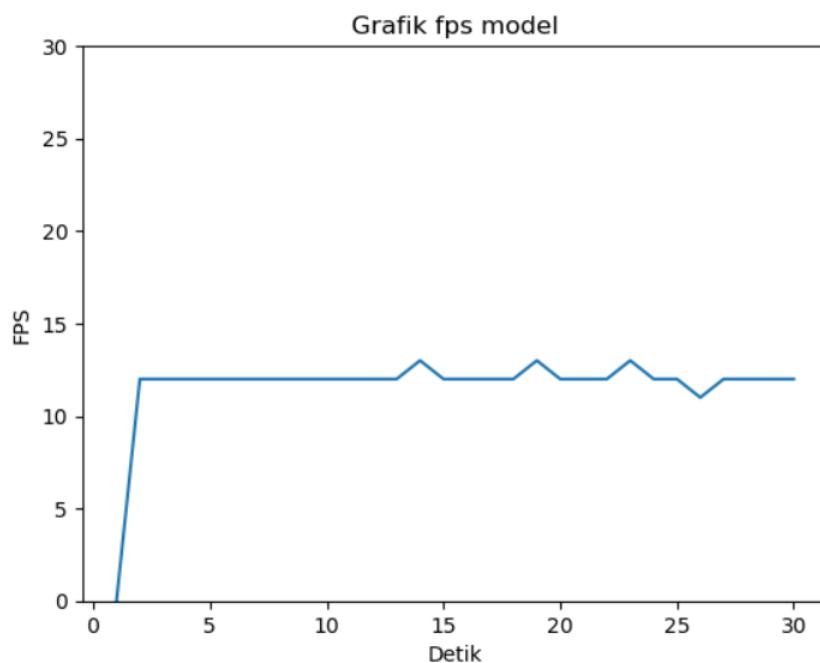
Tabel 4.12: Hasil Pengujian Menggunakan model ResNet50

Variasi Pengujian	Total Data untuk testing	Data Terprediksi Benar	Data Terprediksi Salah	Akurasi(%)
Pengujian Jarak 40 cm	200	195	5	97.5
Pengujian Jarak 60 cm	200	190	10	95
Pengujian Jarak 80 cm	200	195	5	97.5
Pengujian Jarak 100 cm	200	190	10	95
Pengujian Jarak 150 cm	200	166	34	83,00
Pengujian Jarak 200 cm	200	117	83	58.50
Pengujian Jarak 250 cm	200	50	150	25.00
Pengujian Jarak 300 cm	200	32	168	16.00
Pengujian Cahaya Terang	200	189	11	97.5
Pengujian Cahaya Redup	200	190	10	95
Pengujian Cahaya Gelap	200	195	5	94.5

Tabel 4.12: Hasil Pengujian Menggunakan model ResNet50

Pengujian Menggunakan Tangan Responden 1	200	183	17	91.5
Pengujian Menggunakan Tangan Responden 2	200	174	26	87
Pengujian Menggunakan Tangan Responden 3	200	187	13	93.5
Pengujian Menggunakan Tangan Responden 4	200	186	14	93

Berdasarkan Tabel 4.12, dalam pengujian jarak mendapatkan hasil akurasi rata-rata sebesar 96,25%, dimana akurasi tertingginya ada pada angka 97,5% dan terendahnya 95%. Hasil ini tidak jauh dibandingkan dengan model yang penulis buat yang tercantum dalam Tabel 4.3, dimana rata-rata yang didapatkan sebesar 97,5%.



Gambar 4.12: Grafik Pengujian *Frame Rate* Model Resnet50

Sedangkan, untuk pengujian cahaya akurasi sama dengan model yang penulis buat yang dimana akurasinya mengalami penurunan seiring dengan kurangnya cahaya yang masuk. Namun untuk kondisi yang sangat gelap, akurasinya terhitung lebih baik dibandingkan model yang

dibuat penulis dimana akurasinya sebesar 94.5% dibandingkan dengan 89.00%.

Pada jenis pengujian menggunakan tangan responden yang berbeda-beda. Akurasi yang dihasilkan lebih besar dari model yang penulis buat. Rata-rata yang didapatkan model ini adalah sebesar 91,25% dibandingkan dengan 90.00%.

Selain pengujian yang disebutkan pada Tabel 4.12, terdapat pengujian *frame rate* untuk mengukur performa model dalam memprediksi citra. Hasilnya terdapat pada Gambar 4.12, dimana rata-rata nilai FPS-nya berada pada angka 11.66 FPS. Hasil ini lebih kecil dibandingkan dengan model yang dibuat penulis dimana FPS yang didapatkan sebesar 20.86 FPS.

Berdasarkan analisa diatas bisa disimpulkan bahwa akurasi yang dihasilkan model *ResNet-50* jika dibandingkan dengan model yang penulis buat dan diuji di kondisi ideal hasilnya masih lebih kecil walaupun jaraknya tidak jauh, yaitu 97.5% dibanding 99.00%. Namun, untuk performa *frame rate* model ini tergolong cukup rendah diangka 11.66 FPS.

#### 4.2.7.2 Pengujian Menggunakan Model VGG 16

VGG16 merupakan arsitektur Convolutional Neural Network (CNN) yang digunakan untuk memenangkan kompetisi ILSVR (Imagenet) 2014. Model ini dianggap sebagai salah satu arsitektur model untuk visi komputer terbaik hingga saat ini. Angka 16 dalam nama model ini mengartikan bahwa model ini memiliki 16 lapisan pembobot. Hal yang menjadi kelebihan tentang VGG16 adalah penggunaan hyperparameter yang tidak banyak. Mereka fokus menggunakan lapisan konvolusi filter 3x3 pada langkah 1, dan selalu menggunakan *padding* yang sama dan lapisan filter maxpool 2x2 pada langkah 2. Ini mengikuti pengaturan ini agar konsisten di seluruh lapisan arsitektur convolutional dan max pooling. Pada akhir layernya sendiri memiliki 2 *fully connected layer* diikuti oleh softmax untuk *output*.

Tabel 4.13: Hasil Pengujian Menggunakan model VGG 16

Variasi Pengujian	Total Data untuk testing	Data Terprediksi Benar	Data Terprediksi Salah	Akurasi(%)
Pengujian Jarak 40 cm	200	178	22	89
Pengujian Jarak 60 cm	200	193	7	96.5
Pengujian Jarak 80 cm	200	193	7	96.5
Pengujian Jarak 100 cm	200	199	1	99.5
Pengujian Jarak 150 cm	200	196	4	98.00
Pengujian Jarak 200 cm	200	172	28	86.00
Pengujian Jarak 250 cm	200	141	59	70.50
Pengujian Jarak 300 cm	200	101	99	50.50

Tabel 4.13: Hasil Pengujian Menggunakan model VGG 16

Variasi Pengujian	Total Data untuk testing	Data Terprediksi Benar	Data Terprediksi Salah	Akurasi(%)
Pengujian Cahaya Terang	200	176	24	89
Pengujian Cahaya Redup	200	179	21	89.5
Pengujian Cahaya Gelap	200	178	22	88
Pengujian Menggunakan Tangan Responden 1	200	167	33	83.5
Pengujian Menggunakan Tangan Responden 2	200	176	24	88
Pengujian Menggunakan Tangan Responden 3	200	135	65	67.5
Pengujian Menggunakan Tangan Responden 4	200	184	16	92

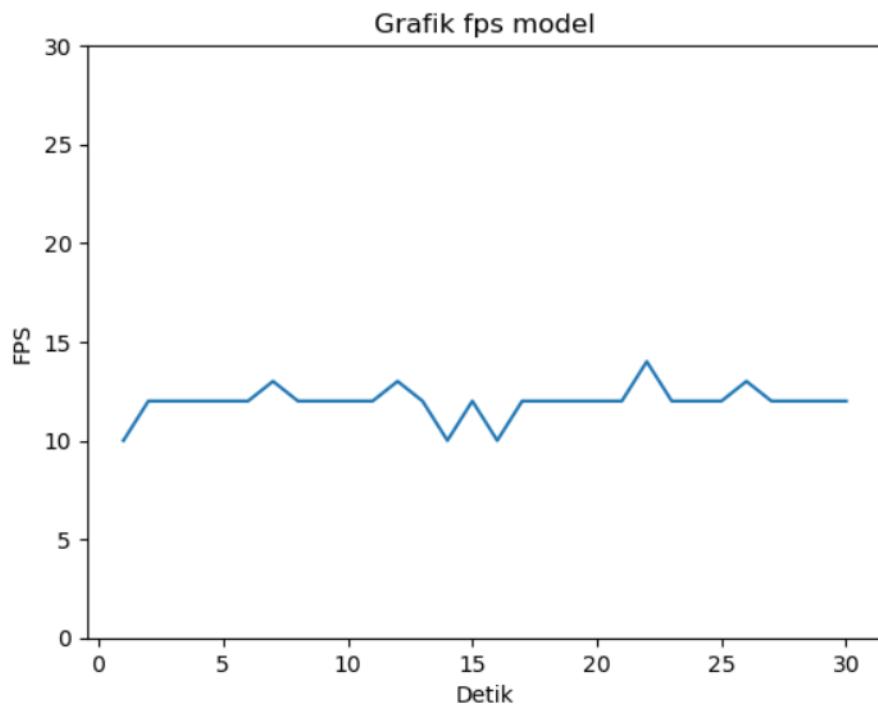
Data dalam tabel 4.13, menunjukkan bahwa dalam pengujian jarak, akurasi malah turun saat jarak menjadi semakin dekat. Sebagai contoh, ketika jarak 100 cm akurasinya sebesar 96.5%, namun ketika jarak 40 cm akurasinya menjadi 89%. Walaupun rata-rata akurasinya masih diangka 94,62 %. Namun, akurasinya lebih kecil dibandingkan dengan model yang penulis buat yang ada diangka 97.5%.

Pada pengujian cahaya, akurasinya juga lebih kecil dibanding model yang dibuat penulis. Akurasi model pada pengujian cahaya ini dikondisi terangnya ada pada angka 89%, sedangkan model yang penulis buat ketika kondisi terangnya ada pada 99%. Walaupun pada kondisi yang benar-benar gelap akurasi yang dihasilkan tidak jauh beda diangka sekitar 88%.

Dalam pengujian menggunakan tangan dari beberapa responden, hasil tingkat akurasi yang didapatkan kurang stabil atau konsisten. Sebagai contoh, pada responden ke-4 akurasi yang dihasilkan sebesar 92%. Namun, begitu diterapkan pada tangan dari responden ke-3, akurasinya turun pada angka 67.5%.

Penurunan performa model ini juga terjadi pada pengujian *frame rate*. Berdasarkan grafik pada Gambar 4.13, didapatkan data bahwa rata-rata FPS yang dihasilkan sebesar 11,96 FPS. Hasil ini lebih kecil dibandingkan model yang dibuat penulis dimana mendapatkan 20,86 FPS.

Melalui analisa diatas, dapat dikatakan bahwa akurasi yang dihasilkan model *VGG 16*



Gambar 4.13: Grafik Pengujian *Frame Rate* Model VGG 16

angkanya lebih kecil jika dibandingkan dengan model yang penulis buat, terutama pada kondisi saat menggunakan tangan dari responden yang tidak ada pada dataset. Performa *frame rate* juga lebih kecil hasilnya karena hanya dapat memproses sampai 11.96 FPS.

#### 4.2.7.3 Pengujian Menggunakan Model MobileNet

MobileNets merupakan salah satu model CNN yang seperti namanya, dapat digunakan pada perangkat mobile. Peneliti dari Google telah menciptakan arsitektur CNN yang dapat mengatasi kebutuhan resource komputasi yang berlebihan. Perbedaan mendasar antara arsitektur MobileNet dengan arsitektur CNN biasanya adalah penggunaan *layer* atau lapisan konvolusional dengan ketebalan filter yang sesuai dengan ketebalan citra masukan.

Tabel 4.14: Hasil Pengujian Menggunakan Model MobileNet

Variasi Pengujian	Total Data untuk Testing	Data Terprediksi Benar	Data Terprediksi Salah	Akurasi(%)
Pengujian Jarak 40 cm	200	192	8	96
Pengujian Jarak 60 cm	200	200	0	100
Pengujian Jarak 80 cm	200	200	0	100
Pengujian Jarak 100 cm	200	200	0	100

Tabel 4.14: Hasil Pengujian Menggunakan Model MobileNet

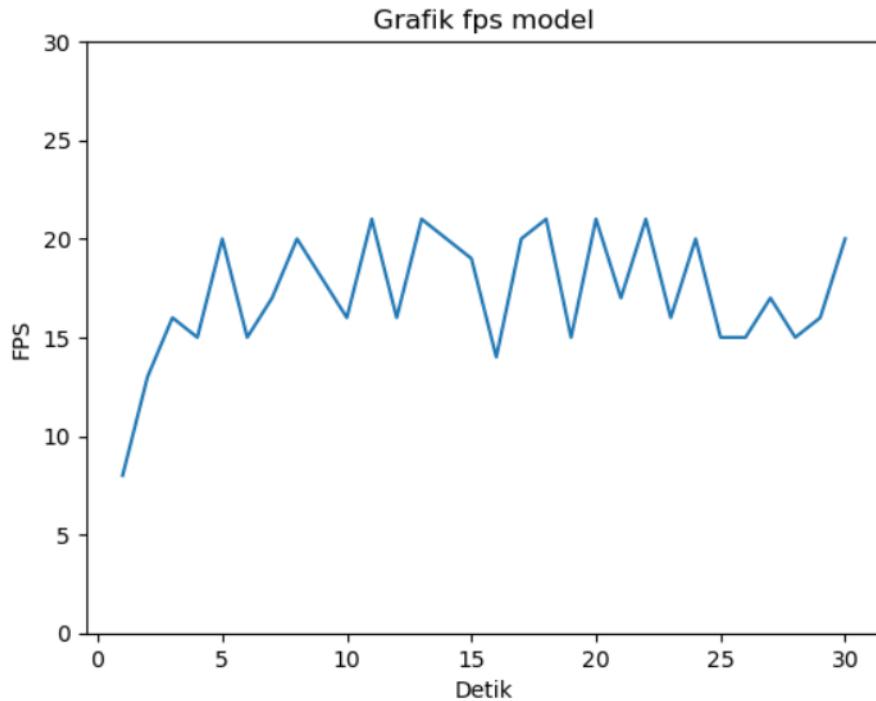
Variasi Pengujian	Total Data untuk Testing	Data Terprediksi Benar	Data Terprediksi Salah	Akurasi(%)
Pengujian Jarak 150 cm	200	188	12	94.00
Pengujian Jarak 200 cm	200	139	61	69.50
Pengujian Jarak 250 cm	200	124	76	62.00
Pengujian Jarak 300 cm	200	118	82	59.00
Pengujian Cahaya Terang	200	193	7	96.5
Pengujian Cahaya Redup	200	193	7	96.5
Pengujian Cahaya Gelap	200	192	8	96
Pengujian Menggunakan Tangan Responden 1	200	199	1	99.5
Pengujian Menggunakan Tangan Responden 2	200	180	20	90
Pengujian Menggunakan Tangan Responden 3	200	191	9	95.5
Pengujian Menggunakan Tangan Responden 4	200	198	2	99

Pada pengujian jarak, model mobileNet memiliki akurasi paling tinggi dibanding model yang lain. Berdasarkan Tabel 4.14, hasil akurasi rata-ratanya sebesar 99%, dimana akurasi tertingginya ada pada angka 100% dan terendahnya 96%. Hasil ini tentu lebih besar dibandingkan dengan model yang penulis buat yang rata-rata akurasi jaraknya sebesar 97.5%.

Dalam pengujian cahaya, akurasi model ini juga masih menjadi yang tertinggi dibandingkan model yang lain. Model ini dapat memprediksi secara stabil dengan tingkat akurasi selalu berada pada kisaran 96,5%. Bahkan dalam kondisi paling gelap sekalipun tidak terjadi penurunan yang signifikan.

Pengujian yang terakhir yaitu menggunakan variasi dari tangan responden, didapatkan

hasil yang juga tertinggi diantara model lain. Rata-rata akurasinya diangka 96%, dengan nilai akurasi paling kecilnya masih diangka 90%.



Gambar 4.14: Grafik Pengujian *Frame Rate* Model *MobileNet*

Sedangkan, dalam pengujian *frame rate*, model *MobileNet* ini masih lebih tinggi dibandingkan model *Resnet50* maupun *VGG 16*. Dapat dilihat dalam grafik Gambar 4.14, rata-rata yang dihasilkan sebesar 17.26 FPS. Namun, FPS dari model ini masih lebih kecil dibandingkan model yang dibuat oleh penulis. Karena memang dari sisi jumlah *layer* dan hasil akhir ukuran file model jauh lebih kecil. Pada model *MobileNet* ukuran file modelnya sebesar 136 MB. Sedangkan model yang dibuat penulis ukurannya hanya sebesar 43 MB saja.

Berdasarkan penyajian data dan analisa ini dapat dikatakan bahwa model *MobileNet* memiliki performa yang paling baik diantara semua model yang diuji pada tugas akhir ini. Tingkat akurasi yang dihasilkan baik pada kondisi yang ideal maupun kurang ideal, model ini masih menghasilkan semua akurasi diangka lebih besar dari 90% semuanya. FPS yang dihasilkan juga lebih tinggi dibanding model *Resnet50* dan *VGG 16* dengan angka 17,26 FPS.

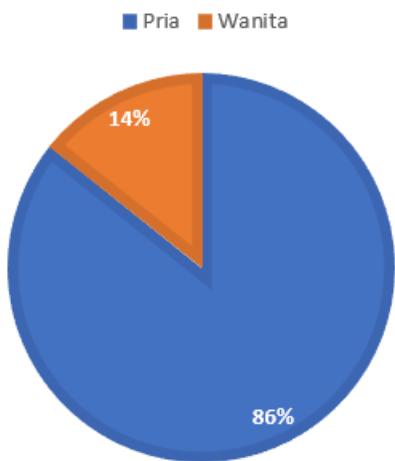
#### 4.2.8 Pengujian *User Experience*

Skenario pengujian yang terakhir adalah pengujian *user experience* dimana dalam pengujian ini terdapat beberapa responden yang diminta untuk mencoba program yang dibuat. Mereka kemudian diminta untuk menjawab beberapa pertanyaan mengenai seberapa baik pengalaman penggunaan mereka dalam bentuk kuesioner. Melalui data tersebut dapat menguji apakah program ini dapat lebih baik dibandingkan dengan cara kontrol presentasi lain yang sudah tersedia saat ini.

##### 4.2.8.1 Distribusi Responden

Pengambilan data kuesioner dilakukan pada area kampus, dimana umumnya responden memiliki latar belakang sebagai pelajar atau mahasiswa. Alasan pengambilan responden den-

## PERSEBARAN DATA GENDER



Gambar 4.15: Persebaran Data Gender

gan latar belakang tersebut karena mereka adalah salah satu bagian masyarakat yang paling sering melakukan kegiatan presentasi. Sehingga, sasaran responden dianggap tepat dalam pen-gujian kali ini.

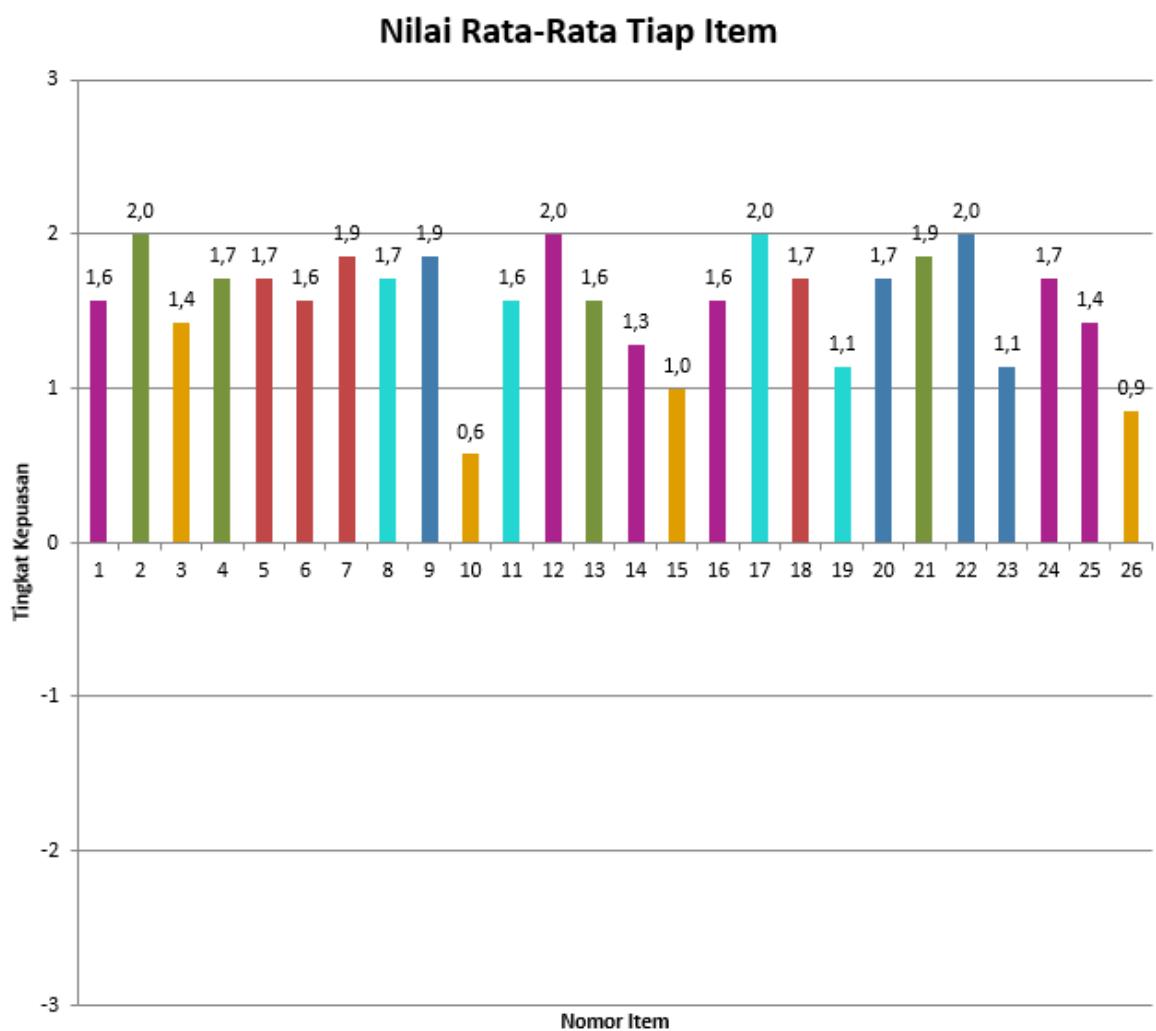
Jika dilihat berdasarkan usia, responden yang mengisi kuesioner berada pada rentang usia antara 21 sampai 22 tahun. Sedangkan berdasarkan jenis kelamin didominasi oleh pria dengan rincian 86% pria dan 14% wanita seperti yang terlihat pada Gambar 4.15

### 4.2.8.2 Hasil Analisa Kuesioner

Berdasarkan pengumpulan kuesioner dari responden yang disebutkan pada 4.2.8.1, data dimasukkan kedalam '*tool*' yang disediakan dari metode UEQ. *Output* yang dihasilkan '*tool*' ini berupa data grafis yang menunjukkan seberapa positif atau negatif evaluasi yang diberikan oleh responden.

Nilai untuk satu item digambarkan dalam skala antara 1 sampai 3. Nilai antara -0,8 dan 0,8 mewakili evaluasi yang kurang lebih dianggap netral. Sedangkan, nilai lebih dari 0,8 mewakili evaluasi atau penilaian yang positif. Sebaliknya, nilai kurang dari -0,8 mewakili evaluasi atau penilaian negatif. Jadi, apabila hasil penilaian ada pada kisaran skala antara -3 berarti sangat buruk dan +3 berarti sangat baik.

*Ouput* dari grafis yang pertama adalah terkait dengan rata-rata jawaban responden pada setiap item soal yang tertera pada Gambar 4.16. Sedangkan, untuk keterangan dari item tiap soalnya dapat dilihat pada Tabel 4.15. Rata-rata jawaban dari responden memberikan penilaian yang bersifat positif karena nilainya lebih besar dari 0,8. Terdapat beberapa item penilaian yang dinilai paling positif yaitu item kemudahan untuk dapat dipahami, baiknya program, keamanan saat digunakan, dan kepraktisan penggunaan.



Gambar 4.16: Jawaban Rata-rata Setiap Item Soal

Tabel 4.15: Keterangan Setiap Nomor Item Soal

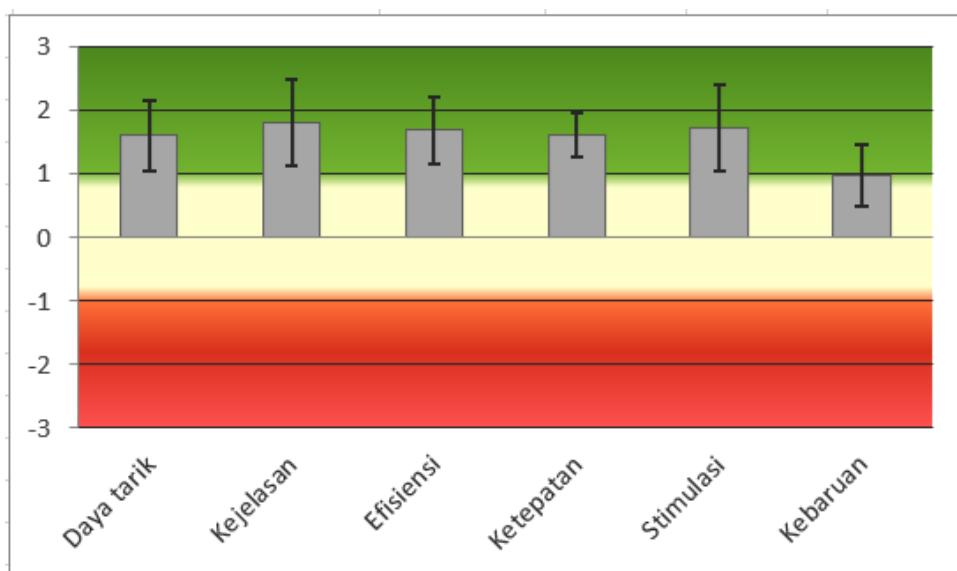
Nomor Item Soal	Isi Item Soal
1	menyusahkan/menyenangkan
2	tak dapat dipahami/dapat dipahami
3	monoton/kreatif
4	sulit dipelajari/mudah dipelajari
5	kurang bermanfaat/bermanfaat
6	membosankan/mengasyikkan
7	tidak menarik/menarik
8	tak dapat diprediksi/dapat diprediksi
9	lambat/cepat
10	konvensional/berdaya cipta
11	menghalangi/mendukung
12	buruk/baik

Tabel 4.15: Keterangan Setiap Nomor Item Soal

Nomor Item Soal	Isi Item Soal
13	rumit/sederhana
14	tidak disukai/menggembirakan
15	lazim/terdepan
16	tidak nyaman/nyaman
17	tidak aman/aman
18	tidak memotivasi/memotivasi
19	tidak memenuhi ekspektasi/memenuhi ekspektasi
20	tidak efisien/efisien
21	membingungkan/jelas
22	tidak praktis/praktis
23	berantakan/terorganisasi
24	tidak atraktif/atratif
25	tidak ramah pengguna/ramah pengguna
26	konservatif/inovatif

Namun, terdapat penilaian yang terhitung bukan positif namun masih ada pada rentang netral. Penilaian itu terkait dengan item daya cipta dan inovasi. Maksud dari item ini adalah terkait dengan seberapa inovatif dan baru ide dari topik tugas akhir ini. Berdasarkan penilaian ini dapat dinyatakan bahwa inovasi dan ide dari tugas akhir ini memang bukan sesuatu yang sangat baru, namun tidak dapat dinyatakan tertinggal juga.

Berdasarkan semua item soal dalam kuesioner, dapat dikelompokkan menjadi enam penilaian utama yaitu *Attractiveness*, *Perspicuity*, *Efficiency*, *Dependability*, *Stimulation*, dan *Novelty*. Hasilnya dapat dilihat pada Gambar 4.17.



Gambar 4.17: Hasil Penilaian *User Experience*

Pada penilaian pertama terdapat *Attractiveness* atau daya tarik yang mendapatkan poin

1,595. Hal ini menunjukkan bahwa program yang dibuat dapat dinilai terlihat menarik, menyenangkan, serta ramah untuk pengguna yang baru mencoba.

Penilaian yang kedua adalah *Perspicuity* atau kejelasan, dimana didapatkan poin sebesar 1,786. Nilai tersebut menunjukkan bahwa program bisa melakukan tugasnya dengan cepat, efisien, dan dengan cara yang pragmatis.

Penilaian yang ketiga adalah *Efficiency*. Poin yang didapatkan untuk efisiensi sebesar 1,679 yang berarti pengguna bisa menggunakan suatu fungsi atau fitur tertentu dengan melalui usaha yang seminimal mungkin.

Item penilaian yang keempat disebut sebagai *Dependability* atau ketepatan. Didapatkan nilai poin 1,607 yang berarti nilai tersebut menunjukkan bahwa interaksi antara pengguna dengan program yang dijalankan masih ada dibawah kendali dan sesuai dengan perkiraan atau harapan pemakainya.

Dalam item penilaian yang kelima ada *Stimulation*. Poin yang didapatkan sebesar 1,714. Hasil penilaian ini menunjukkan bahwa pengguna mendapatkan efek rangsangan perasaan yang positif, baik itu perasaan menarik, menyenangkan, dan seru.

Item Penilaian yang terakhir adalah *Novelty* atau kebaruan. Dalam item penilaian ini didapatkan poin sebesar 0,964. Memang angka ini tidak terhitung cukup tinggi dibandingkan dengan penilaian sebelumnya, namun masih termasuk dalam rentang penilaian positif. Oleh karenanya, program ini masih bisa dikatakan sebagai produk kreatif dan inovatifnya, walaupun memang dinilai oleh beberapa pengguna bukan sebagai suatu ide yang benar-benar baru.

Berdasarkan analisa pada pengujian *user experience* dapat diambil penilaian bahwa program yang dicoba memberikan pengalaman pengguna dengan hasil reaksi evaluasi yang bersifat positif, terutama dari aspek kemudahan untuk dipahami, kepraktisan dalam penggunaan, dan memberikan pengalaman perasaan menarik dan menyenangkan.

*[Halaman ini sengaja dikosongkan]*

## **BAB V**

### **PENUTUP**

Dalam bab ini berisi mengenai kesimpulan berdasarkan penelitian yang dilakukan dengan menyesuaikan tujuan yang berusaha dicapai. Pada bab ini juga, terdapat saran-saran yang diberikan dalam rangka meningkatkan dan mengembangkan penelitian selanjutnya yang terkait dengan topik tugas akhir ini. Berikut ini adalah kesimpulan dan saran yang dapat diambil.

#### **5.1 Kesimpulan**

Berdasarkan penelitian yang sudah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut.

1. Besar nilai akurasi yang didapatkan dalam pengujian model sebesar 99,37%, sedangkan nilai *loss* yang didapatkan sebesar 0,03%.
2. Berdasarkan pengujian jarak, didapatkan hasil bahwa nilai akurasi tertingginya pada jarak 40 cm dengan akurasi sebesar 99,00%. Pada jarak 60 cm akurasinya sebesar 97,50%, pada jarak 80 cm akurasinya 97,00%, pada jarak 100 cm akurasinya 94,50%, pada jarak 150 cm akurasinya sebesar 90,50%, dan saat jarak 200 cm akurasinya sebesar 77,50%. Sedangkan, batas jarak dimana model mulai menghasilkan akurasi lebih kecil dari 50% dimulai saat jarak 250 cm, dimana akurasi yang didapatkan sebesar 32,50%. Akurasi terendahnya sendiri ada pada jarak terjauh yaitu 300 cm, dengan akurasi 13,00%.
3. Hasil nilai akurasi tertinggi dalam pengujian variasi kondisi cahaya adalah saat kondisi terang atau lebih dari 40 lx, dengan tingkat akurasi sebesar 99,00%. Sedangkan nilai akurasi terendahnya pada kondisi gelap disekitar 5 lx dengan tingkat akurasi sebesar 89,00%
4. Hasil akurasi pada pengujian dengan responden yang berbeda, didapatkan rata-rata sebesar 90,87%. Dimana pada responden pertama akurasinya sebesar 88,00%, pada responden kedua sebesar 99,50%, pada responden ketiga sebesar 86,50%, dan pada responden keempat sebesar 89,50%.
5. Hasil rata-rata waktu respon yang didapatkan sebesar 185,2 *milliseconds*, dengan rata-rata waktu respon tercepat pada kelas *erase* sebesar 131,4 *milliseconds*, dan waktu respon terlama pada kelas *pen* sebesar 292 *milliseconds*.
6. Kontrol presentasi menggunakan pose tangan ini juga memberikan respon *user experience* yang positif dan dinilai menarik serta mudah untuk digunakan oleh pengguna.

#### **5.2 Saran**

Dalam rangka pengembangan penelitian kedepan terutama dalam topik mengenai pose tangan ataupun kontrol presentasi, dapat diusulkan beberapa saran yang sebagai berikut.

1. Menggunakan metode yang berbeda selain *Convolutional Neural Network* untuk dapat membandingkan performa yang dihasilkan.
2. Menambahkan tahapan *preprocessing* data untuk membandingkan pengaruhnya terhadap proses pengenalan citra.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Essen, B. C. V., Awwal, A. A. S., & Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics*.
- AlSaedi, A. K. H., & AlAsadi, A. H. H. (2020). A new hand gestures recognition system. *Indonesian Journal of Electrical Engineering and Computer Science*.
- Bernico, M. (2018). Deep learning quick reference: Useful hacks for training and optimizing deep neural networks with tensorflow and keras.
- Cho, J., Lee, K., Shin, E., Choy, G., & Do, S. (2016). How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *Department of Radiology, Massachusetts General Hospital and Harvard Medical School*.
- Deshpande, A. (2016). Cnn. <https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529>
- Elgendi, M. (2020). Deep learning for vision systems. *Shelter Island. NY: Manning Publication Co.*
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *Microsoft Research*.
- Heydarian, M. R., Doyle, T. E., & Samavi, R. (2022). Mlcm: Multi-label confusion matrix. *IEEE Access*.
- Hidayatullah, P. (2017). Pengolahan citra digital : Teori dan aplikasi nyata.
- Hinderks, A., Schrepp, M., Mayo, F. J. D., Escalona, M., & Thomaschewski, M. E. J. (2019). Developing a ux kpi based on the user experience questionnaire. *Computer Standards and Interfaces*.
- Idrees, M., Ahmad, A., Butt, M. A., & Danish, H. M. (2021). Controlling powerpoint using hand gestures in python. *Department of Data Science, University of the Punjab, Lahore, Pakistan*.
- Indriani, Moh.Harris, & Agoes, A. S. (2021). Applying hand gesture recognition for user guide application using mediapipe.
- Lakkavaram, V. S., Raghuveer, L. V. S., Kumar, C. S., Sri, G. S., & Habeeb, S. (2019). Max pooling layer. [https://www.researchgate.net/publication/332570921\\_A\\_review\\_on\\_Practical\\_Diagnostic\\_of\\_Tomato\\_Plant\\_Diseases](https://www.researchgate.net/publication/332570921_A_review_on_Practical_Diagnostic_of_Tomato_Plant_Diseases)
- Muthoharoh, M. (2019). Media powerpoint dalam pembelajaran. *Tasyri: Jurnal Tarbiyah Syariah-Islamiyah*.
- Poerwanti, J. I. S., & Mahfud, H. (2018). Optimalisasi penggunaan media pembelajaran interaktif dengan microsoft power point pada guru-guru sekolah dasar. *JPPM (Jurnal Pengabdian Dan Pemberdayaan Masyarakat)*.
- Rahmad, C., Prasetyo, A., & Baqy, R. A. (2022). Powerpoint slideshow navigation control with hand gestures using hidden markov model method. *Matrix: Jurnal Manajemen Teknologi dan Informatika*.
- Rasmila, Huda, N., Jemakmun, Mukti, A. R., Amalia, R., Hadinata, N., Kurniawan, Putra, A., & Nainggolan, C. E. (2022). Pelatihan presentasi menggunakan microsoft power point pada smp patra mandiri 2 palembang. *ABSYARA: Jurnal Pengabdian Pada Masyarakat*.

- Salunke, T. P., & Bharkad, S. . (2017). Power point control using hand gesture recognition based on hog feature extraction and k-nn classification. *Institute of Electrical and Electronics Engineers*.
- Soroni, F., Sajid, S. A., & Bhuiyan, M. N. H. (2021). Hand gesture based virtual blackboard using webcam. *Conference: 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)At: University of British Columbia, Vancouver, Canada.*
- Suartika, I. W., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi citra menggunakan convolutional neural network (cnn) pada caltech 101.
- Wardani, H. (2016). Pengembangan instrumen asesmen presentasi ilmiah di sma.

## **BIOGRAFI PENULIS**



M. Dafa Raisya Rajwa, lahir di Semarang pada tanggal 19 Mei 2001. Merupakan anak terakhir dari tiga bersaudara. Penulis menempuh pendidikan di SMPN 2 Kendal (2013-2016), dan SMAN 1 Kendal (2016-2019). Saat ini penulis melanjutkan pendidikan sarjana S1 di Institut Teknologi Sepuluh Nopember Surabaya dalam Program Studi Teknik Komputer.

Penulis memiliki keterkaitan dalam bidang komputer, khususnya pengembangan aplikasi. Selama kuliah, penulis mengikuti beberapa kegiatan yang berhubungan dengan ilmu dibidang terkait seperti acara MAGE, UKM Robotika ITS, dan menjadi Asisten Laboratorium Telematika B201.

*[Halaman ini sengaja dikosongkan]*