

**IDENTIFIKASI EKSPRESI WAJAH PESERTA KELAS
ONLINE MENGGUNAKAN CONVOLUTIONAL
NEURAL NETWORK DENGAN TENSORFLOW PADA
PLATFORM ONLINE MEETING**



SKRIPSI

MU'TASHIM BILLAH

1710130010

Program Studi Ilmu Komputer

Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer ESQ

Jakarta

2021

**IDENTIFIKASI EKSPRESI WAJAH PESERTA KELAS
*ONLINE MENGGUNAKAN CONVOLUTIONAL
NEURAL NETWORK DENGAN TENSORFLOW PADA
PLATFORM ONLINE MEETING***



SKRIPSI

**Diajukan Sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Komputer**

**MU'TASHIM BILLAH
1710130010**

**Program Studi Ilmu Komputer
Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer ESQ
Jakarta
Juni 2021**

VISI & MISI STIMIK ESQ

VISI

Menjadi Perguruan Tinggi yang Terpercaya di Bidang Pendidikan, Penelitian, dan Pengabdian kepada Masyarakat untuk Mewujudkan Peradaban Emas.

MISI

Kami Perguruan Tinggi Berbasis Karakter yang Menyelenggarakan Pembelajaran, Penelitian, dan Pengabdian Masyarakat melalui Keseimbangan Spiritualitas, Kreativitas, dan Intelektualitas;

Kami Membentuk Entrepreneur dan Profesional yang Mampu Menjadi Pemimpin Perubahan, Beretika, Memiliki Kepekaan Sosial, dan Berkelas Dunia;

Kami Berjuang Menginspirasi Masyarakat untuk Mewujudkan Peradaban Emas.

5 VALUES STIMIK ESQ

Integrity

Passion

Creativity

Humility

Professionalism



PERNYATAAN ORISINALITAS

Saya menyatakan bahwa karya tulis berupa Skripsi yang berjudul "**Identifikasi Ekspresi Wajah Peserta Kelas Online Menggunakan Convolutional Neural Network dengan Tensorflow pada Platform Online Meeting**", adalah:

1. Sepenuhnya hasil karya saya sendiri berdasarkan dari pengetahuan yang didapat selama kuliah termasuk bahan pustaka yang dijadikan referensi.
2. Bukan merupakan duplikasi dari karya ilmiah orang lain yang diakui sebagai hasil tulisan ataupun pikiran sendiri melalui proses penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.
3. Bukan merupakan karya terjemahan dari kumpulan sumber referensi acuan.

Atas pernyataan ini, saya siap menerima sanksi yang dijatuhkan, apabila di kemudian hari terbukti bahwa karya ini merupakan hasil plagiat baik bersifat parsial maupun menyeluruh, atau adanya klaim dari pihak lain terhadap keaslian karya saya ini.

Bekasi, 22 Juni 2021

Yang membuat pernyataan,



Mu'tashim Billah

1710130010

LEMBAR PENGESAHAN

Skripsi yang diajukan oleh :

Nama : Mu'tashim Billah
NIM : 1710130010
Program Studi : Ilmu Komputer
Judul Skripsi : Identifikasi Ekspresi Wajah Peserta Kelas *Online*
Menggunakan *Convolutional Neural Network* dengan
Tensorflow pada Platform *Online Meeting*

Telah diuji dan berhasil dipertahankan dihadapan Dewan Pengaji dalam sidang pendadaran, dan dinyatakan L U L U S.

Ditetapkan di : Jakarta

Tanggal : 22 Juni 2021

Dosen Pembimbing,

Andika Sundawijaya, S.T., M.Kom

NIDN : 03-1303-8701

DEWAN PENGUJI

Ketua Penguji Ahlijati Nuraminah, S.Kom, M.T.I

NIDN: 03-1712-8404

Penguji 1 Erza Sofian S.Kom, M.Sc

NIDN: 03-2207-7002

Penguji 2 Desy Komalasari, S.Kom, M.Kom

NIDN: 03-2212-9202

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Saya yang bertanda tangan dibawah ini:

Nama : Mu'tashim Billah
NIM : 1710130010
Program Studi : Ilmu Komputer
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer ESQ Hak Bebas Royalti Non Eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

**“Identifikasi Ekspresi Wajah Peserta Kelas *Online* Menggunakan
Convolutional Neural Network dengan *Tensorflow* pada Platform *Online*
Meeting”**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif ini Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer ESQ berhak menyimpan, mengalih media/format, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Bekasi

Pada tanggal : 22 Juni 2021

Yang Menyatakan



(Mu'tashim Billah)

ABSTRAK

Nama	:	Mu'tashim Billah
Program Studi	:	Ilmu Komputer
Judul Skripsi	:	Identifikasi Ekspresi Wajah Peserta Kelas <i>Online</i> Menggunakan <i>Convolutional Neural Network</i> dengan <i>Tensorflow</i> pada Platform <i>Online Meeting</i>
Pembimbing	:	Andika Sundawijaya, S.T., M.Kom

Tahun 2020 memberikan kesan yang berat bagi dunia. Tercatat sampai saat ini Negara Indonesia masih dilanda pandemi COVID-19. Keadaan tersebut memaksa pelajar untuk melakukan pembelajaran secara daring sesuai Surat Edaran Nomor 4 Tahun 2020 tentang Pelaksanaan Kebijakan Pendidikan Dalam Masa Darurat Penyebaran Coronavirus Disease (COVID-19) yang dikeluarkan oleh Kementerian Pendidikan dan Kebudayaan yang terus berlanjut sampai saat ini. Pada pelaksanaan kegiatan belajar secara daring, banyak sekolah-sekolah dan perguruan tinggi yang memanfaatkan teknologi *online meeting* untuk kelangsungan proses belajar mengajar. Pertemuan secara daring dengan platform *online meeting* yang mendukung fitur video secara real-time mendasari dilaksanakannya penelitian ini. Pada penelitian kali ini akan menerapkan algoritma *deep learning* dengan metode *Convolutional Neural Network* dengan *Tensorflow* untuk melakukan klasifikasi ekspresi wajah dari peserta kelas *online*. Pengklasifikasian ekspresi wajah peserta dilakukan secara *real-time* atau berupa video rekaman kegiatan. Hasil yang diperoleh adalah akurasi terbaik dihasilkan oleh arsitektur CNN dengan 4 lapisan *convolutional layer*, 2 lapisan *fully connected layer*, penggunaan *dropout* sebesar 0.5, dan menggunakan *learning rate* sebesar 0.0001. Evaluasi model yang diperoleh yaitu *training accuracy* sebesar 0.7802, *validation accuracy* sebesar 0.6772, *training loss* sebesar 0.6021, dan *validation loss* sebesar 0.9832 yang menunjukkan bahwa model belum sepenuhnya dirancang dan dilatih dengan baik. Hasil dari tingkat keyakinan yang diperoleh dari pengujian model pada rekaman video kegiatan kelas *online* berkisar antara 0.60 sampai dengan 0.64 yang menunjukkan bahwa tingkat keyakinan model relatif rendah. Dari hasil tersebut peneliti memberikan rekomendasi untuk menambah jumlah dataset khususnya pada ekspresi *disgust* yang memiliki jumlah data yang sangat sedikit. Selain itu, perlu menambah analisis dengan menambahkan jumlah *filter* pada *convolution layer* serta penerapan *augmentation* untuk menambah keakuratan dalam prediksi citra.

Kata kunci: Covid-19, Ekspresi Wajah, *Convolutional Neural Network* (CNN), *Tensorflow*, *Online Meeting*

ABSTRACT

<i>Name</i>	:	Mu'tashim Billah
<i>Study Program</i>	:	<i>Computer Science</i>
<i>Title</i>	:	<i>Identification of Online Class Participant's Facial Expressions Using Convolutional Neural Network with Tensorflow on the Online Meeting Platform</i>
<i>Counsellor</i>	:	Andika Sundawijaya, S.T., M.Kom

The year 2020 gives a heavy impression on the world. Until now, Indonesia is still hit by the COVID-19 pandemic. This situation forces students to do online learning in accordance with Circular Letter No. 4 of 2020 concerning the Implementation of Educational Policies in the Emergency Period for the Spread of Coronavirus Disease (COVID-19) issued by the Ministry of Education and Culture which continues to this day. In the implementation of online learning activities, many schools and universities use online meeting technology for the continuity of the teaching and learning process. Online meetings with online meeting platforms that support real-time video features underlie this research. In this study, we will apply a deep learning algorithm with the Convolutional Neural Network method with Tensorflow to classify facial expressions from online class participants. Classification of participants' facial expressions is done in real-time or in the form of video recordings of activities. The results obtained are the best accuracy is generated by the CNN architecture with 4 layers of convolutional layers, 2 layers of fully connected layers, the use of dropout of 0.5, and using a learning rate of 0.0001. The evaluation of the model obtained is training accuracy of 0.7802, validation accuracy of 0.6772, training loss of 0.6021, and validation loss of 0.9832 which indicates that the model has not been fully designed and trained properly. The results of the level of confidence obtained from testing the model on video recordings of online class activities ranged from 0.60 to 0.64 which indicates that the confidence level of the model is relatively low. From these results, the researcher provides recommendations to increase the number of datasets, especially on disgust expressions which have very little data. In addition, it is necessary to increase the analysis by adding the number of filters on the convolution layer and the application of augmentation to increase accuracy in image prediction.

Keywords: Covid-19, Facial Expression, Convolutional Neural Network (CNN), Tensorflow, Online Meeting

KATA PENGANTAR

Segala puja dan puji syukur saya panjatkan kepada Ilahi Rabbi Allah Subhanahu Wata’ala atas segala limpahan rahmat dan karunia-Nya sampai detik ini hingga saya dapat menyelesaikan skripsi tepat pada waktunya. Shalawat serta salam saya limpahkan dan senandungkan kepada Baginda Rasulullah Shallallahu ‘Alaihi Wasallam yang atas jasa-jasa Beliau yang senantiasa memperjuangkan cahaya ilmu, sehingga kita dapat merasakan hangatnya *tholabul ‘ilmi*.

Penulisan skripsi ini dilaksanakan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Komputer di Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer ESQ. Saya sangat menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, mulai dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk dapat menuntaskan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- 1) Kedua orang tua saya yaitu Ayahanda Taupik Hidayat dan Ibunda Mumun Maemunah, beserta segenap keluarga saya yang telah memberikan dukungan serta bantuan baik secara material maupun moral tanpa henti-hentinya;
- 2) Bapak Andika Sundawijaya, S.T., M.Kom, selaku dosen pembimbing yang telah menyediakan waktu untuk mengarahkan saya dalam proses penyelesaian skripsi ini;
- 3) Segenap dosen Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer ESQ yang tidak dapat saya sebutkan satu persatu namun telah banyak membantu saya; dan
- 4) Seluruh sahabat saya khususnya mahasiswa program studi Ilmu Komputer angkatan 2017 yang tiada lelahnya menemani saya untuk berdiskusi dan mendengarkan keluh kesah selama proses penelitian hingga penulisan laporan.

Banyak rintangan dan hambatan dalam proses penelitian hingga penyelesaian laporan skripsi ini khususnya saat masa pandemi COVID-19 yang tentu membuat saya merasa berat dalam menjalannya. Mulai dari setiap prosesnya harus dilakukan secara daring, hingga kebutuhan akan sumber dan bahan olah yang terbatas. Namun tidak kalah dengan suasana dan keadaan, segala upaya dan usaha saya kerahkan dan perjuangkan agar terselesaikannya penelitian ini hingga tepat pada waktunya. Mudah-mudahan apa yang telah diperjuangkan sampai saat ini, dapat menjadi batu loncatan saya agar dapat meraih hasil yang Allah ridhai.

Akhir kata, saya berharap Allah Subhanahu Wata'ala berkenan membalaq segala kebaikan semua pihak yang telah membantu atas terselesaikannya skripsi ini. Saya ucapkan banyak-banyak terima kasih, *Jazakumullah Ahsanal Jaza*. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu dan senantiasa diridhoi dan diberkahi oleh Allah Subhanahu Wata'ala.

Bekasi, 22 Juni 2021



Mu'tashim Billah

DAFTAR ISI

VISI & MISI STIMIK ESQ	ii
5 VALUES STIMIK ESQ	iii
PERNYATAAN ORISINALITAS	iv
LEMBAR PENGESAHAN	v
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Rumusan Masalah	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
BAB 2 TINJAUAN LITERATUR.....	6
2.1 Landasan Teori.....	6
2.1.1 Ekspresi Wajah.....	6
2.1.2 Citra.....	8
2.1.3 <i>Artificial Intelligence</i>	10
2.1.4 <i>Computer Vision</i>	10
2.1.5 <i>Machine Learning</i>	12
2.1.6 <i>Artificial Neural Network</i>	14
2.1.7 <i>Backpropagation</i>	17
2.1.8 <i>Deep Learning</i>	19
2.1.9 <i>Convolutional Neural Network</i>	21
2.1.10 <i>Object Detection</i>	22
2.1.11 <i>Learning Rate</i>	23
2.1.12 <i>Dropout Regularization</i>	24

2.1.13 <i>Tensorflow</i>	25
2.2 Penelitian Terdahulu	26
2.3 Metodologi Pemecahan Masalah	30
2.4 Kerangka Pemikiran	32
BAB 3 METODE PENELITIAN.....	33
3.1 Alur Penelitian.....	33
3.2 Subjek dan Objek Penelitian	35
3.2.1 Subjek Penelitian.....	35
3.2.2 Objek Penelitian	36
3.3 Metode Pengumpulan Data	36
3.3.1 Jenis Data	36
3.3.2 Sumber Data.....	36
3.4 Instrumen Penelitian.....	37
3.5 Metode Analisis, Pengujian dan Interpretasi Data	38
3.5.1 Analisis Data	38
3.5.2 Pengujian Data	38
3.5.3 Proses <i>Convolution</i>	43
3.5.4 Proses <i>Pooling</i>	45
3.5.5 Proses <i>Classification (Flatten, Fully Connected, dan Output)</i> ...	46
3.5.6 Proses Pengujian Melalui Tampilan GUI	47
3.5.7 Interpretasi Data	48
BAB 4 ANALISIS DAN PEMBAHASAN	49
4.1 Analisis.....	49
4.1.1 Analisis <i>Dataset</i>	49
4.1.2 Persiapan <i>Tools</i> dan <i>Package</i>	51
4.1.3 <i>Preprocessing</i> dan Pembagian <i>Dataset</i>	56
4.1.4 Inisialisasi Arsitektur Model	56
4.2 Pembahasan.....	57
4.2.1 Hasil Pengujian Model.....	57
4.2.2 Arsitektur Akhir Model.....	62
4.2.3 Hasil Deteksi Video	64
BAB 5 PENUTUP.....	65
5.1 Simpulan.....	65
5.2 Saran.....	66
DAFTAR PUSTAKA	67

DAFTAR GAMBAR

Gambar 2.1 Macam-macam Ekspresi Wajah.....	8
Gambar 2.2 Model Sederhana Jaringan Saraf pada Manusia	15
Gambar 2.3 Model Sederhana <i>Artificial Neural Network</i>	15
Gambar 2.4 Arsitektur <i>Single Layer</i>	16
Gambar 2.5 Arsitektur <i>Multi Layer</i>	17
Gambar 2.6 Arsitektur <i>Backpropagation</i>	18
Gambar 2.7 Arsitektur <i>Convolutional Neural Network (CNN)</i>	22
Gambar 2.8 Visualisasi <i>Dropout Layer</i>	25
Gambar 2.9 Kerangka Pemikiran.....	32
Gambar 3.1 Alur Penelitian.....	33
Gambar 3.2 Alur Eksperimen Pengujian Data Secara Keseluruhan	38
Gambar 3.3 Pengujian Pertama dengan Menganalisis Pengaruh Jumlah Tumpukan <i>Convolutional Layer</i>	41
Gambar 3.4 Pengujian Kedua dengan Menganalisis Pengaruh Penggunaan Nilai <i>Dropout</i>	41
Gambar 3.5 Pengujian Ketiga dengan Menganalisis Pengaruh Besaran Nilai <i>Learning Rate</i>	42
Gambar 3.6 Pengujian Keempat dengan Menggunakan <i>Epochs</i> yang Lebih Besar yaitu 250.....	43
Gambar 3.7 Visualisasi Proses <i>Convolution</i>	43
Gambar 3.8 Perhitungan Manual Proses <i>Convolution</i>	44
Gambar 3.9 Pergeseran <i>Kernel</i> dengan 1 <i>Stride</i>	44
Gambar 3.10 Visualisasi Proses <i>Pooling</i> dengan <i>Max-Pooling</i>	46
Gambar 3.11 Visualisasi Proses <i>Flatten</i> , <i>Fully Connected</i> , dan <i>Output Classification</i>	47
Gambar 3.12 Alur Pengujian Video <i>Real-Time</i> Melalui Interaksi GUI.....	48
Gambar 4.1 Visualisasi Frekuensi Diagram <i>Donut</i> Dalam Bentuk Persentase	50
Gambar 4.2 Website Anaconda <i>Individual Edition</i>	52
Gambar 4.3 <i>Homescreen</i> Anaconda Navigator.....	52
Gambar 4.4 Mengecek versi Conda dan Python Melalui Anaconda Prompt	53
Gambar 4.5 Instalasi <i>Package</i> atau <i>Library</i> Numpy Menggunakan Pip.....	53
Gambar 4.6 Instalasi <i>Package</i> atau <i>Library</i> Pandas Menggunakan Pip	54
Gambar 4.7 Instalasi <i>Package</i> atau <i>Library</i> Matplotlib Menggunakan Pip.....	54
Gambar 4.8 Instalasi <i>Package</i> atau <i>Library</i> Opencv Menggunakan Pip	55

Gambar 4.9 Instalasi <i>Package</i> atau <i>Library Tensorflow</i> Menggunakan Pip.....	55
Gambar 4.10 Instalasi <i>Package</i> atau <i>Library Keras</i> Menggunakan Pip	55
Gambar 4.11 Instalasi <i>Package</i> atau <i>Library PySimpleGUI</i> Menggunakan Pip ..	56
Gambar 4.12 Visualisasi Perbandingan Evaluasi Pengaruh Jumlah Lapisan <i>Convolution Layer</i>	58
Gambar 4.13 Visualisasi Perbandingan Evaluasi Pengaruh Nilai <i>Dropout</i>	59
Gambar 4.14 Visualisasi Perbandingan Evaluasi Pengaruh Nilai <i>Learning Rate</i> ..	59
Gambar 4.15 Visualisasi Perbandingan Evaluasi Pengaruh Jumlah <i>Epochs</i>	60
Gambar 4.16 Analisis antara Evaluasi <i>Training</i> dan <i>Validation</i> pada <i>Learning</i> <i>Rate</i> Sebesar 0,001	61
Gambar 4.17 Analisis antara Evaluasi <i>Training</i> dan <i>Validation</i> pada <i>Learning</i> <i>Rate</i> Sebesar 0,0001	61
Gambar 4.18 Arsitektur Jaringan <i>Convolutional Neural Network</i> yang Digunakan	62

DAFTAR TABEL

Tabel 2.1 Kategori Emosi Negatif dan Positif pada Manusia	7
Tabel 2.2 Penelitian Terdahulu	28
Tabel 2.3 Metodologi Pemecah Masalah	30
Tabel 4.1 Frekuensi <i>Dataset Train</i> dan <i>Validation</i>	49
Tabel 4.2 Rincian Video yang Diperoleh dari Bagian Akademik STIMIK ESQ .	50
Tabel 4.3 Inisialisasi Arsitektur Model	57
Tabel 4.4 Evaluasi Pengaruh Jumlah Lapisan <i>Convolution Layer</i>	58
Tabel 4.5 Evaluasi Pengaruh Nilai <i>Dropout</i>	58
Tabel 4.6 Evaluasi Pengaruh Nilai <i>Learning Rate</i>	59
Tabel 4.7 Evaluasi Pengaruh Jumlah <i>Epochs</i>	60
Tabel 4.8 Hasil Pendekripsi Wajah dari Rekaman Video Kegiatan Kelas <i>Online</i> yang Berbeda dengan Durasi 120 Detik	64

DAFTAR LAMPIRAN

Lampiran 1: Arsitektur Lengkap Model	L1
Lampiran 2: Hasil Evaluasi Lengkap Pengaruh Lapisan <i>Convolution Layer</i>	L2
Lampiran 3: Hasil Evaluasi Lengkap Pengaruh Penggunaan Nilai <i>Dropout</i>	L6
Lampiran 4: Hasil Evaluasi Lengkap Pengaruh Besar Nilai <i>Learning Rate</i>	L8
Lampiran 5: Hasil Evaluasi Lengkap Pengaruh Jumlah <i>Epochs</i>	L10
Lampiran 6: Tampilan GUI.....	L15
Lampiran 7: Kode Python – Pelatihan Model.....	L16
Lampiran 8: Kode Python – Pengujian kepada <i>Image</i>	L18
Lampiran 9: Kode Python – Pengujian kepada <i>File Video</i>	L19
Lampiran 10: Kode Python – Pembuatan GUI menggunakan PySimpleGUI	L20
Lampiran 11: Surat Izin Penelitian	L23
Lampiran 12: Kartu Bimbingan Skripsi	L25
Lampiran 13: Lembar Pernyataan Perbaikan Hasil Sidang Skripsi	L30
Lampiran 14: Lembar Penilaian Pengaji Sidang	L31
Lampiran 15: Daftar Riwayat Hidup	L35

BAB 1

PENDAHULUAN

Pada bab ini berisi latar belakang penelitian, identifikasi masalah dari latar belakang yang diangkat, rumusan masalah, tujuan masalah, dan manfaat penelitian bagi beberapa pihak.

1.1 Latar Belakang

Tahun 2020 memberikan kesan yang berat bagi dunia. Tercatat sampai saat ini Negara Indonesia masih dilanda pandemi COVID-19. Keadaan tersebut memaksa pelajar untuk melakukan pembelajaran secara daring sesuai Surat Edaran Nomor 4 Tahun 2020 tentang Pelaksanaan Kebijakan Pendidikan Dalam Masa Darurat Penyebaran Coronavirus Disease (Covid-19) yang dikeluarkan oleh Kementerian Pendidikan dan Kebudayaan yang terus berlanjut sampai saat ini (Dwi et al., 2020). Kebijakan tersebut juga berlaku bagi seluruh mahasiswa yang disinggung melalui Surat Edaran Kementerian Pendidikan Kebudayaan Direktorat Pendidikan Tinggi No 1 Tahun 2020 tentang tentang pencegahan penyebaran Coronavirus Disease (Covid-19) di perguruan tinggi (Firman & Rahayu, 2020). Dengan adanya kebijakan ini, seluruh perguruan tinggi tak terkecuali Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer (STIMIK) ESQ pun turut mengeluarkan Surat Edaran Nomor 1 Tahun 2020 tentang penyelenggaraan perkuliahan dan kegiatan kemahasiswaan yang terus diperpanjang hingga saat ini.

Pada pelaksanaan kegiatan belajar secara daring, banyak sekolah-sekolah dan perguruan tinggi yang memanfaatkan teknologi *online meeting* untuk kelangsungan proses belajar mengajar. Di antara platform *online meeting* yang digunakan dalam proses belajar mengajar adalah Zoom Meeting dan Google Meet (Liu & Ilyas, 2020). Zoom Meeting dan Google Meet merupakan sarana modern untuk melakukan komunikasi dengan mengandalkan teknologi *cloud* yang andal (Abdillah, 2020). Dengan kedua platform *online meeting* tersebut memungkinkan adanya pertemuan secara *real-time* hingga lebih daripada 100 peserta dalam sekali pertemuan.

Diwajibkannya menjaga jarak dan adanya surat edaran yang mengharuskan segala kegiatan dilakukan secara daring juga menjadi sebuah penghalang bagi berbagai kegiatan masyarakat umumnya. Dengan melakukan kegiatan dengan secara daring melalui teknologi *online meeting* tentu tidak banyak hal yang bisa dilakukan layaknya pertemuan luring. Namun hal ini tidak mengurangi bobot dari pertemuan itu sendiri (Firman & Rahayu, 2020). Pertemuan secara daring dengan platform *online meeting* yang mendukung fitur video secara *real-time* mendasari dilaksanakannya penelitian ini. Masing-masing peserta yang mengikuti *online meeting* dapat mengaktifkan dan menonaktifkan fitur video *real-time* yang terhubung dengan kamera di perangkatnya dengan tujuan agar dapat melihat wajah satu sama lain layaknya pertemuan luring (Abdillah, 2020). Hal ini mendukung penelitian dengan metode pengenalan wajah yang memanfaatkan video *real-time* dari peserta sebagai sumber datanya.

Wajah adalah elemen yang digunakan untuk membedakan identitas manusia untuk dapat mengenali satu sama lain. Selain itu, wajah juga merupakan media penyampaian emosi seseorang melalui ekspresi yang dibuatnya (Lubis & Perdana, 2020). Emosi merupakan luapan perasaan yang timbul akibat keadaan seseorang pada situasi tertentu (Fitri & Adelya, 2017). Sedangkan ekspresi merupakan media untuk berkomunikasi secara nonverbal untuk menyampaikan emosi yang sedang dirasakan oleh seseorang. Ekspresi wajah yang ditimbulkan merupakan suatu umpan balik dalam proses interaksi dan dijadikan sebagai pengganti respons terhadap pesan yang ingin disampaikan (Fitri & Adelya, 2017).

Pengenalan wajah dalam sistem *biometric* memiliki dua tahapan yaitu pendektsian wajah seseorang dan melakukan klasifikasi terhadap wajahnya. Manusia dapat mengenali wajah seseorang dengan melakukan kedua tahapan ini dengan sangat cepat, namun berbeda dengan komputer yang perlu waktu yang lebih lama dan data yang lebih banyak agar hasil yang diperoleh memiliki tingkat akurasi yang lebih tinggi (Anggraini, 2020). Beberapa bagian pada wajah yang dapat dijadikan bahan untuk membedakan dalam proses pengenalan wajah antara lain adalah mata, hidung dan mulut (Lubis & Perdana, 2020). Namun, saat ini kebutuhan akan pengenalan wajah sudah sampai kepada pengenalan emosi manusia melalui

ekspresi wajah yang ditampilkannya maka perlu sebuah algoritma yang apik agar mendapatkan hasil yang cepat dan akurat (Mubarok, 2019).

Bagi manusia pada umumnya, mengolah citra dan mengklasifikasikannya adalah hal mudah. Namun bagi komputer itu adalah hal yang sulit karena komputer hanya dapat membaca deretan piksel dan angka dari citra yang diolah (Thohari & Hertantyo, 2018). Komputer mengadopsi sistem jaringan saraf pada manusia untuk dapat mengolah data dengan lebih baik. *Artificial Neural Network* (ANN) atau yang dapat diartikan Jaringan Saraf Tiruan (JST) adalah sebuah algoritma pembelajaran mesin yang merepresentasi sel saraf biologis pada manusia (Ramadhan, 2020). Untuk memperoleh hasil yang maksimal dalam pemrosesan suatu citra digunakan teknik pembelajaran lanjut menggunakan *deep learning* yang merupakan bagian dari *machine learning* berdasar pada jaringan saraf tiruan (Rena, 2019).

Deep Learning merupakan proses pembelajaran dalam yang berarti mengolah data dengan banyak tingkatan *layer* untuk mengolah data yang kompleks (Rena, 2019). *Deep Learning* menggabungkan proses *feature extraction* dan klasifikasi secara bersamaan dalam melakukan pencocokan *pattern recognition* (Mubarok, 2019). *Convolutional Neural Network* (CNN) merupakan salah satu contoh metode *deep learning* yang menghasilkan akurasi tertinggi dalam pengolahan berbasis citra (Thohari & Hertantyo, 2018). Hal ini dikarenakan metode CNN meniru teknik pengolahan informasi citra dari sistem pengenalan pada visual *cortex* manusia (Wayan et. al, 2016). Di samping kelebihannya tersebut, CNN memiliki kekurangan layaknya metode *deep learning* lainnya yaitu pelatihan model yang relatif lama. Seiring perkembangannya, kekurangan tersebut dapat ditangguhkan dengan menggunakan teknologi *Graphical Processing Unit* (GPU) disandingkan dengan spesifikasi komputer yang tinggi (Nurhikmat, 2018).

Pada penelitian kali ini akan menerapkan algoritma *deep learning* dengan metode CNN dengan *Tensorflow* untuk melakukan klasifikasi ekspresi wajah dari peserta kelas *online*. Pengklasifikasian ekspresi wajah peserta dilakukan secara *real-time* atau berupa video rekaman kegiatan. Penelitian ini dilaksanakan dengan harapan aplikasi yang dihasilkan dapat dijadikan alat bantu ukur bahan evaluasi kegiatan kelas *online* melalui kumpulan dan kesimpulan ekspresi wajah peserta.

1.2 Identifikasi Masalah

Mengacu pada latar belakang yang dijelaskan, peneliti memberikan informasi tentang masalah yang akan dibahas di dalam penelitian ini, yaitu:

- 1) Belum adanya sistem yang menerapkan pendekripsi ekspresi wajah ketika kegiatan *online meeting* berlangsung.
- 2) Belum adanya sarana pengelompokan ekspresi wajah peserta kelas *online* melalui rekaman video kegiatan kelas *online* pada platform *online meeting* yang nantinya dapat diambil langkah lebih lanjut.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah yang dijelaskan, maka dapat ditarik kesimpulan tentang masalah yang akan dibahas dan diteliti dalam tulisan ini, yaitu:

- 1) Bagaimana mengimplementasikan *deep learning* menggunakan *Tensorflow* dalam memproses *multi* klasifikasi ekspresi wajah?
- 2) Bagaimana penerapan algoritma *Convolutional Neural Network* (CNN) dengan menggunakan *Tensorflow* untuk melakukan klasifikasi ekspresi wajah secara *real-time*?
- 3) Bagaimana hasil analisis pengenalan wajah peserta kelas *online* melalui rekaman video kegiatan kelas *online* pada platform *online meeting*?

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam penelitian ini berdasarkan rumusan masalah yang telah disebutkan di atas adalah:

- 1) Untuk mengetahui langkah-langkah dalam implementasi *deep learning* menggunakan *Tensorflow* dalam memproses *multi* klasifikasi ekspresi wajah.
- 2) Untuk mengetahui langkah-langkah penerapan algoritma *Convolutional Neural Network* (CNN) menggunakan *Tensorflow* untuk melakukan klasifikasi ekspresi wajah secara *real-time*.
- 3) Untuk mengetahui hasil analisis pengenalan wajah peserta kelas *online* melalui rekaman video kegiatan kelas *online* pada platform *online meeting*.

1.5 Manfaat Penelitian

Berdasarkan tujuan penelitian tersebut, maka manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut:

1) Bagi Peneliti

Menambah ilmu, kreativitas dan pengalaman dalam bidang Ilmu Komputer, serta sebagai salah satu syarat untuk meraih gelar Sarjana Strata Satu (S-1) di Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer (STIMIK) ESQ. Dapat ikut serta membangun generasi Indonesia Emas dan Indonesia Adidaya sesuai dengan bidang yang ditempuh.

2) Bagi Lembaga

Hasil dari penelitian ini kiranya dapat digunakan sebagai tambahan informasi dalam meningkatkan kualitas *output* pendidikan, khususnya di Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer (SITMIK) ESQ.

3) Bagi Pengguna

Sistem yang dibuat dapat digunakan sebagaimana fungsinya yaitu sebagai alat bantu pengelompokan dan penilaian emosi peserta *online meeting* melalui ekspresi yang ditangkap oleh sistem.

4) Bagi Peneliti Selanjutnya

Peneliti meyakini bahwa penelitian ini dapat dilanjutkan dengan menggunakan berbagai metode lainnya dan dapat dijadikan referensi bagi peneliti selanjutnya. Selain itu, dari hasil penelitian ini dapat dijadikan pembahasan mengenai emosi yang ditampilkan oleh peserta *online meeting* selama proses kegiatan *online meeting* berlangsung, khususnya pada kegiatan belajar *online* di Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer (STIMIK) ESQ.

BAB 2

TINJAUAN LITERATUR

Pada bab ini berisi landasan teori dari berbagai literatur, penelitian terdahulu, metodologi pemecah masalah, dan kerangka pemikiran sebagai dasar keberlangsungan penelitian ini.

2.1 Landasan Teori

Pada subbab ini akan memaparkan teori-teori pendukung yang berkaitan dengan penelitian ini. Adapun teori-teori yang digunakan pada penelitian ini adalah sebagai berikut :

2.1.1 Ekspresi Wajah

Ekspresi wajah merupakan hasil dari gerakan yang mengubah posisi otot pada wajah. Dalam melakukan komunikasi secara langsung, ekspresi wajah adalah bentuk komunikasi nonverbal dalam mengungkapkan emosi yang ada pada seseorang baik berupa emosi yang positif maupun negatif (Prawitasllri, 2016). Pada umumnya ekspresi wajah timbul secara alami ataupun spontan sesuai perasaan dan emosi yang dialami seseorang, tetapi ekspresi wajah pun dapat diubah-ubah secara sengaja (Sari, 2019).

Pada hakikatnya, akan terasa sangat sulit jika manusia menyembunyikan emosi yang tergambar oleh ekspresi wajah. Walaupun manusia dapat menyembunyikan emosi yang ditampilkan oleh ekspresi wajah, namun pada saat-saat tertentu secara tidak sengaja akan muncul ekspresi wajah yang sesuai dengan emosinya saat itu (Lubis & Perdana, 2020). Emosi manusia sendiri terbagi menjadi dua bagian yaitu emosi primer dan sekunder. Adapun emosi primer terdiri dari senang, sedih, marah, takut, terkejut dan jijik, sedangkan emosi sekunder merupakan campuran dan gabungan dari emosi primer serta dipengaruhi oleh kondisi budaya dimana seseorang tersebut tinggal (Nadhiroh, 2015). Adapun kategori emosi secara ringkas dapat dilihat pada Tabel 2.1.

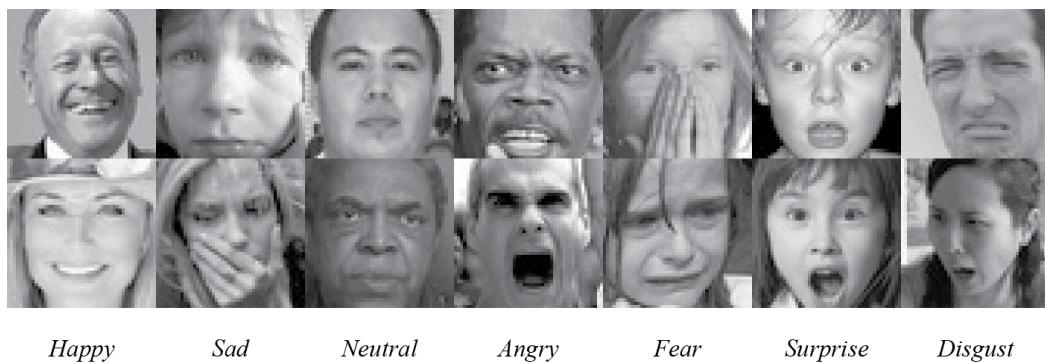
Tabel 2.1 Kategori Emosi Negatif dan Positif pada Manusia

Sumber: Yahdinil Firda Nadhiroh (2015) telah diolah kembali

Emosi Positif	Emosi Negatif
<ul style="list-style-type: none"> • Rela • Lucu • Gembira / Ceria • Senang / Nyaman • Rasa ingin tahu • Bahagia • Kesukaan • Cinta / Sayang • Ketertarikan 	<ul style="list-style-type: none"> • Tidak sabaran • Kebimbangan • Marah • Curiga • Cemas • Rasa bersalah • Cemburu • Jengkel • Takut • Depresi • Sedih • Benci

Dari Tabel 2.1 yang merupakan keseluruhan kategori emosi primer dan sekunder yang ada pada setiap manusia dapat ditimbulkan dan ditangkap melalui ekspresi wajah. Namun, hanya emosi primer dapat ditampakkan lewat ekspresi wajah. Ekspresi wajah seseorang menyimpan banyak informasi, seperti senang, sedih, takut, gugup, tidak percaya diri, dan lain sebagainya dapat terlihat dari wajah (Amda & Fitriyani, 2016). Saat melakukan komunikasi dengan orang lain, secara tidak langsung perasaan yang kita rasakan akan terungkap oleh ekspresi wajah kita. Dengan demikian, kita tidak bisa mengubah ekspresi permanen yang sudah ada pada wajah kita. Menurut Marono, Clarke, Navarro, dan Keatley (2017) terdapat gerakan-gerakan otot pada wajah yang terletak disekitar mata dan mulut yang memberikan informasi terkait perasaan yang kita rasakan dan dapat dipahami orang lain yang disebut *micro expression*.

Menurut (Abidin & Harjoko, 2011) dari keenam dasar ekspresi primer tersebut merupakan campuran dari ekspresi positif dan negatif tersebut akan ditambah satu ekspresi lagi yaitu netral. Macam-macam ekspresi wajah pada manusia tersebut antara lain yaitu ekspresi senang, sedih, netral, marah, takut, terkejut dan jijik. Contoh ekspresi dari ketujuh ekspresi wajah tersebut dapat dilihat pada Gambar 2.1 di bawah ini.



Gambar 2.1 Macam-macam Ekspresi Wajah

Sumber: website Kaggle, Jonathan Oheix (2019) telah diolah kembali

2.1.2 Citra

Pada pemrosesan gambar dua dimensi (2D) bidang ilmu komputer sering menyebutnya dengan pemrosesan atau pengolahan citra digital. Data yang diolah merupakan kode-kode warna pada setiap piksel dalam citra. Data tersebut direpresentasikan dalam bentuk *array* 2 dimensi yang berisi nilai-nilai yang kompleks (Zainuddin et al., 2017).

Menurut Hotma Pangaribuan dan Joni Eka Candra (2018) citra adalah penyebutan lain dari gambar. Istilah citra sering didengar dalam dunia *machine learning* bidang pengolahan citra. Dalam bidang tersebut, citra didefinisikan sebagai fungsi yang memiliki dua variabel koordinat dan nilai dari fungsi tersebut atau $f(x, y)$ merupakan intensitas citra pada koordinat tersebut. Sedangkan citra yang telah mengalami proses digitalisasi pada proses pengolahan citra pada komputer disebut citra digital. Citra digunakan sebagai masukan atau *input* dan direpresentasikan sebagai matriks dengan tiap-tiap baris dan kolom dibentuk dari nilai piksel pada citra tersebut.

Menurut Anton Yudhana, Sunardi, dan Shoffan Saifullah (2016) citra adalah gambaran, kemiripan, atau imitasi dari objek yang diamati. Citra tidak selalu berupa gambar atau foto, citra juga dapat diperoleh dari sinyal video yang diperoleh dari TV atau penyimpanan media langsung. Sedangkan citra yang jelas dari objek yang dapat diolah menggunakan teknologi komputer disebut citra digital. Semakin rumit dan besar ukuran citra tersebut, maka semakin besar juga memori yang

dibutuhkan komputer untuk merepresentasikan citra. Adapun jenis-jenis citra dapat dibagi menjadi:

1) Citra Biner

Citra biner atau *binary image* adalah citra yang hanya memiliki dua nilai derajat keabuan yaitu putih dan hitam. Pada setiap piksel yang menggambarkan objek akan diberi nilai 1 selain itu (latar belakang) akan diberi nilai 0. Piksel-piksel yang memiliki nilai 0 melambangkan warna putih sedangkan yang memiliki nilai 1 melambangkan warna hitam saat menampilkan citra.

2) Citra Keabuan

Citra keabuan atau *grayscale* merupakan citra yang tiap-tiap piksel memiliki satu buah *layer*. Citra keabuan memiliki nilai intensitas di antara nilai 0 yang melambangkan warna hitam sampai dengan 255 yang melambangkan warna putih.

3) Citra Warna

Citra warna atau citra RGB merupakan citra digital yang memiliki tiga layer yang menginformasikan warna pada tiap pikselnya. Ada beberapa sistem pewarnaan yang ada pada gambar selain RGB adalah CMYK, HSV, dan lain sebagainya. RGB adalah model warna citra yang terdiri dari *red*, *green*, dan *blue* yang digabungkan untuk membentuk kombinasi susunan warna yang sangat luas. Pada layar monitor komputer, rentang nilai citra terletak pada nilai 0 sampai dengan 255.

Berdasarkan pemaparan di atas, citra merupakan istilah lain dari gambar. Istilah citra sering digunakan dalam bidang *machine learning* khususnya pengolahan gambar atau citra. Citra terdiri dari piksel-piksel yang sejatinya berisi angka-angka yang menentukan warna pada tiap-tiap pikselnya. Citra terdiri dari tiga jenis yaitu citra biner, citra keabuan atau *grayscale*, dan citra warna atau RGB. Pada citra biner dan keabuan, citra direpresentasikan sebagai *array* dua dimensi, sedangkan untuk citra warna atau RGB direpresentasikan sebagai *array* tiga dimensi di mana nilai ketiga dari dimensi citra merupakan warna dari citra tersebut.

2.1.3 *Artificial Intelligence*

Artificial Intelligence merupakan serangkaian langkah yang dapat ditempuh sistem komputer untuk dapat mampu melakukan serangkaian cara berpikir yang dimiliki manusia dan hewan. *Artificial Intelligence* juga merupakan bentuk dari puncak penerapan IT yang mampu menyatakan, mengumpulkan, merancang kecerdasan manusia dalam wujud sistem. Sistem akan terlebih dahulu mencari pola yang optimal dan kemudian mengaplikasikannya dalam suatu pekerjaan. Perkembangan robot hingga hologram atau *barcode* canggih tidak lepas dari perkembangan *Artificial Intelligence* yang semakin baik dan cepat. *Artificial Intelligence* dirancang tidak hanya untuk dapat mempermudah pekerjaan manusia, tetapi dirancang untuk dapat melakukan pekerjaan tersebut layaknya manusia normal (Pratikno, 2018).

Menurut Edo Dwi Christianto (2017) *Artificial Intelligence* atau Intelelegensi Buatan dapat didefinisikan sebagai pengenalan dan pembiasaan perlakuan terhadap komputer melalui sistem. Perlakuan tersebut yang membuat komputer mampu untuk dapat berpikir dan melakukan hal-hal lain yang manusia pada umumnya atau secara khusus dapat lakukan. Tujuan umum dari kecerdasan buatan antara lain:

- 1) Membuat mesin menjadi lebih pintar (tujuan utama)
- 2) Memahami apa itu kecerdasan (tujuan ilmiah)
- 3) Membuat mesin menjadi lebih bermanfaat (tujuan entrepreneurial)

Berdasarkan pemaparan di atas, dapat disimpulkan bahwa *Artificial Intelligence* merupakan serangkaian proses berpikir manusia atau hewan yang ditanam pada komputer. Sebelum proses berpikir itu ditanam pada komputer, perlu adanya pembelajaran dan pelatihan agar menghasilkan pola berpikir yang baik dan benar. Secara umum, *Artificial Intelligence* atau kecerdasan buatan dibuat dan dirancang sedemikian rupa untuk meminimalkan pekerjaan manusia beserta risiko kesalahan yang dibuatnya.

2.1.4 *Computer Vision*

Computer Vision merupakan suatu sistem yang dapat mengotomatisasi analisis terhadap citra dan video oleh komputer yang bertujuan untuk

mengumpulkan berbagai macam informasi dan pemahaman dari objek yang diamati. Umumnya tujuan utama dari pengembangan *Computer Vision* adalah untuk melakukan duplikasi kemampuan penglihatan manusia baik dalam hal penangkapan atau perekaman suatu citra, analisa citra dan pemahaman akan objek yang dideteksi itu ke dalam sebuah mesin atau komputer. Dapat disimpulkan bahwa *Computer Vision* adalah sebuah kemampuan yang luar biasa pada sebuah mesin atau komputer untuk melihat dan memahami sebuah citra dengan setara atau bahkan dapat melebihi kemampuan penglihatan manusia asli (Purnamajaya, 2019).

Menurut Duta Sayoga, Purba Daru Kusuma, Faisal Candrasyah Hasibuan (2020) *Computer Vision* merupakan salah satu bidang penelitian teknologi yang mengajarkan komputer untuk dapat memiliki kemampuan melihat dan mengenal objek layaknya manusia. Salah satu penerapan dari *Computer Vision* yang sudah banyak diterapkan adalah *Object Recognition* atau pengenalan suatu objek. *Internet of Things* (IoT) merupakan ladang penerapan *Computer Vision* yang memiliki konsep terkait objek yang memiliki kemampuan untuk mengirim data melalui jaringan tanpa interaksi antara manusia kepada manusia ataupun antara manusia kepada komputer.

Menurut Wulan Anggraini (2020) *Computer Vision* merupakan bidang ilmu pengetahuan yang mendalamai pengetahuan tentang pembuatan sistem yang dapat mengambil keputusan berdasarkan gambar atau citra yang diamati. *Computer Vision* memiliki sebuah misi yaitu menjadikan komputer “*Act like Human Sight*” sehingga komputer memiliki kemampuan yang setara atau bahkan melebihi manusia untuk mendapatkan berbagai macam informasi secara visual. Kelebihan yang dimiliki dalam *Computer Vision* antara lain adalah:

- 1) *Object Detection*, mengenali suatu benda yang terdapat pada *scene* untuk melihat batasannya.
- 2) *Recognition*, menempatkan label pada objek
- 3) *Description*, menugaskan properti pada objek
- 4) *3D Interface*, menerjemahkan macam-macam aksi 2D dari 3D yang dilihat.
- 5) *Interpreting Motion*, Menerjemahkan gerakan-gerakan.

Berdasarkan pemaparan di atas, peneliti menyimpulkan bahwa *Computer Vision* merupakan otomatisasi analisis citra dan video oleh sistem komputer yang bertujuan untuk mengumpulkan informasi layaknya manusia. Namun, *Computer Vision* memiliki kemampuan penglihatan yang jauh melebihi manusia karena tiap-tiap informasi yang ditangkap akan diubah menjadi angka-angka dan diproses sedemikian rupa sehingga dapat diperoleh kesimpulan yang lebih akurat dibandingkan manusia. *Computer Vision* kerap diterapkan pada produk *Internet of Things* (IoT) yang memiliki konsep kemampuan transfer data melalui jaringan tanpa interaksi langsung oleh manusia terhadap sistem di komputer.

2.1.5 *Machine Learning*

Machine Learning merupakan sebuah teknik analisis data yang mengajarkan komputer untuk mampu melakukan pekerjaan alamiah yang dilakukan manusia dan hewan. *Machine Learning* memberikan proses pembelajaran kepada komputer secara terus menerus layaknya pembelajaran dari pengalaman (data yang pernah dilihat). Algoritma *machine learning* menerapkan metode komputasi dalam melakukan pembelajaran informasi dari data yang tersedia tanpa adanya pengaruh daripada persamaan yang telah ditentukan sebagai model *machine learning*. Algoritma *machine learning* akan beradaptasi guna meningkatkan kinerja model saat jumlah sampel atau data latih yang tersedia meningkat (Mubarok, 2019).

Menurut Yusuf Umar Hanafi (2020) *machine learning* atau pembelajaran mesin merupakan bagian dari *Artificial Intelligence* yang memiliki tujuan untuk mengoptimalkan kriteria dengan menganalisa sampel data yang tersedia atau terekam sebelumnya. Teknik ini memiliki tujuan akhir yaitu agar model dapat mengambil sebuah prediksi untuk data baru. Oleh karena itu, dengan adanya teknik *machine learning* yang pada umumnya sudah banyak diterapkan pada kehidupan sehari-hari, manusia sudah tidak perlu lagi untuk melakukan identifikasi sebuah proses secara utuh. Dengan teknik *machine learning* komputer mampu membantu manusia untuk membuat pola dalam menarik sebuah keputusan dan kesimpulan.

Triano Nurhikmat (2018) berpendapat bahwa *machine learning* dapat didefinisikan sebagai metode komputasi yang bertindak berdasarkan pengalaman

untuk meningkatkan performa dalam membuat prediksi yang akurat dan tepat. Berbeda dalam pemahaman pengalaman pada umumnya, definisi pengalaman disini adalah data atau informasi yang sebelumnya telah tersedia dan dijadikan media belajar. Dalam pembelajaran *machine learning* terdapat beberapa skenario, antara lain:

1) *Supervised Learning*

Dalam pembelajaran mesin menggunakan skenario *supervised learning* mesin menerima masukan data yang sudah diberi label atau *class* pada tiap-tiap baris datanya. Setelah itu mesin melakukan pembelajaran dan pencarian pola yang optimal berdasarkan data yang telah diberi label. Contoh algoritma yang menerapkan pembelajaran *supervised learning* antara lain *Decision Tree*, *Nearest Neighbor Classifier*, *Naïve Bayes Classifier*, *Artificial Neural Network*, *Support Vector Machine*, *Fuzzy K-Nearest Neighbor*, dan lain sebagainya.

2) *Unsupervised Learning*

Dalam pembelajaran mesin menggunakan skenario *unsupervised learning* mesin menerima masukan data yang tidak memiliki label. Mesin melakukan pembelajaran dan pencarian pola yang optimal berdasarkan macam-macam karakteristik yang ditemui. Setelah pola tersebut ditemukan, data yang sebelumnya tidak memiliki label dapat diberi label sesuai pola karakteristik yang dimiliki. Contoh algoritma yang menerapkan pembelajaran *unsupervised learning* antara lain *K-Means*, *Hierarchical Clustering*, *DBSCAN*, *Fuzzy C-Means*, *Self-Organizing Map*, dan lain sebagainya.

3) *Reinforcement Learning*

Dalam pembelajaran mesin menggunakan skenario *reinforcement learning* mesin melakukan fase pembelajaran dan tes saling dicampur. Untuk mengumpulkan informasi pembelajar secara aktif dan efektif dengan berinteraksi ke lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari proses pembelajaran.

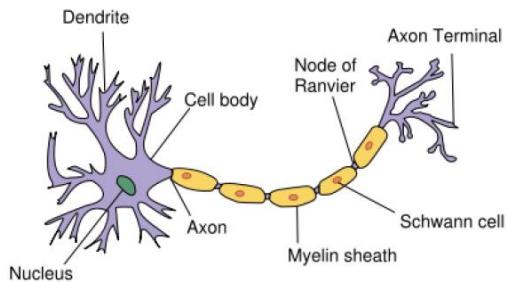
Berdasarkan pemaparan di atas, dapat disimpulkan bahwa *machine learning* atau pembelajaran mesin merupakan serangkaian teknik komputasi dan

pembelajaran dari serangkaian data yang menghasilkan sebuah aturan. Proses pembelajaran tidak dilakukan hanya sekali saja, layaknya manusia pembelajaran yang dilakukan oleh mesin juga dapat diperoleh dari pengalaman belajar sebelumnya. *Machine Learning* merupakan bagian daripada penerapan *Artificial Intelligence* yang memiliki tujuan untuk mengoptimalkan dalam proses analisa sampel data untuk dapat menghasilkan sebuah prediksi yang tepat dan akurat. Proses skenario dalam *machine learning* dibagi menjadi tiga yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.

2.1.6 Artificial Neural Network

Menurut John, Azhar, dan Supatmi (2013) *Artificial Neural Network* (ANN) atau yang dapat disebut Jaringan Saraf Tiruan (JST) adalah suatu teknik yang terinspirasi oleh sistem sel saraf biologi. Jaringan Saraf Tiruan menyerupai fungsi kerja otak manusia dalam dua hal, yaitu pengetahuan yang diperoleh jaringan dari proses belajar dan kekuatan dari hubungan antar sel saraf (*neuron*) yang digunakan untuk menyimpan pengetahuan. Jaringan Saraf Tiruan menerima masukan dari suatu data atau dari output sel saraf. Setiap pola-pola informasi yang diperoleh dari masukan yang datang akan diproses di dalam *neuron*. *Neuron-neuron* tersebut terkumpul di dalam lapisan-lapisan yang dapat disebut *neuron layers*.

Menurut Eka dan Edi (2017) Jaringan Saraf Tiruan atau *Artificial Neural Network* meniru kinerja otak manusia yang selalu mencoba melakukan simulasi proses pembelajaran. Dalam mempelajari sesuatu, otak menerima berbagai informasi yang diterima melalui jaringan saraf yang saling berhubungan. Istilah buatan dan tiruan ini digunakan untuk menandakan bahwa jaringan saraf ini serupa dengan kinerja jaringan saraf pada manusia yang diimplementasikan dengan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. Model sederhana *Artificial Neural Network* dapat dilihat pada Gambar 2.2.

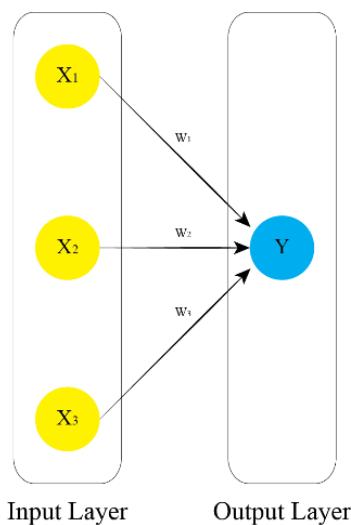


Gambar 2.2 Model Sederhana Jaringan Saraf pada Manusia

Sumber: Eka dan Edi (2017)

Jaringan Saraf Tiruan (*Artificial Neural Network*) atau dapat disingkat JST merupakan suatu proses pengolahan informasi yang mempunyai karakteristik mirip dengan jaringan saraf biologi (Hilmi, 2014). Jaringan Saraf Tiruan dibentuk sedemikian rupa sebagai generalisasi model matematika dari jaringan saraf biologi, dengan mengasumsikan:

- 1) Pemrosesan sebuah informasi dilakukan pada setiap node (*neuron*).
- 2) Sinyal dikirim di antara *neuron-neuron* melalui berbagai penghubung.
- 3) Penghubung antar *neuron* memiliki bobot yang berpengaruh pada kekuatan sinyal.
- 4) Dalam menentukan *output*, setiap *neuron* menggunakan fungsi aktivasi berdasarkan pada jumlah *input* yang diterima.



Gambar 2.3 Model Sederhana *Artificial Neural Network*

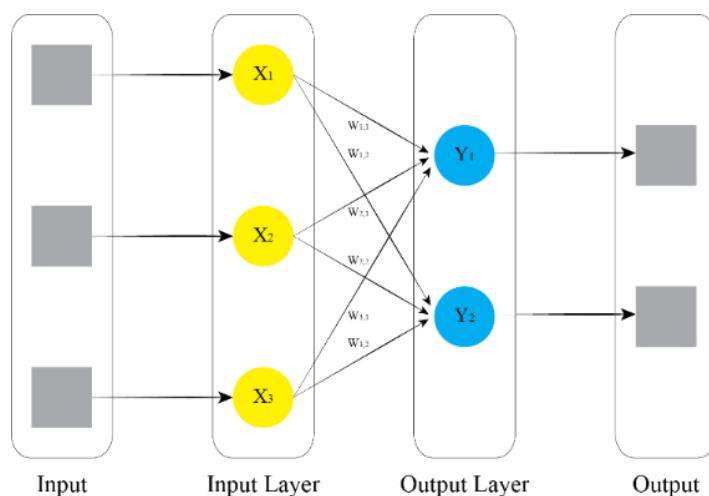
Sumber: Mubarok (2019) telah diolah kembali

Berdasarkan pemaparan di atas, peneliti menyimpulkan bahwa Jaringan Saraf Tiruan merupakan representasi buatan dari otak manusia. Pada dasarnya otak manusia terus mengalami proses pembelajaran untuk dapat menyelesaikan suatu permasalahan. Jaringan Saraf Tiruan ini terinspirasi oleh sistematika kerja otak manusia yaitu hubungan antar sel saraf (*neuron*) untuk menyimpan pengetahuan dari proses pembelajaran. Jaringan ini dibentuk berdasarkan teknik dari sistem sel saraf biologi yang digeneralisasikan sebagai model matematika.

Pada Gambar 2.3 dapat disimpulkan bahwa Y menerima masukan dari *neuron* x_1 , x_2 , dan x_3 dengan bobot yang menghubungkan masing-masing w_1 , w_2 , dan w_3 . Nilai impuls yang diterima oleh Y mengikuti fungsi aktivasi $y = f(\text{net})$. Apabila fungsi tersebut memiliki nilai yang kuat, maka sinyal akan diteruskan. Nilai dari fungsi aktivasi juga dapat digunakan sebagai dasar untuk perubahan bobot. Adapun arsitektur dari *Artificial Neural Network* dibagi menjadi dua, antara lain:

1) Single Layer

Sesuai dengan namanya, bentuk arsitektur *single layer* pada ANN hanya memiliki satu *layer input* dan satu *layer output*. Pada setiap *neuron* di *layer input* akan selalu terhubung kepada setiap *neuron* di *layer output*. Arsitektur ini hanya akan melakukan pengolahan secara langsung dari *input* yang diterima (Mubarok, 2019). Contoh arsitektur *single layer* dapat dilihat pada Gambar 2.4.

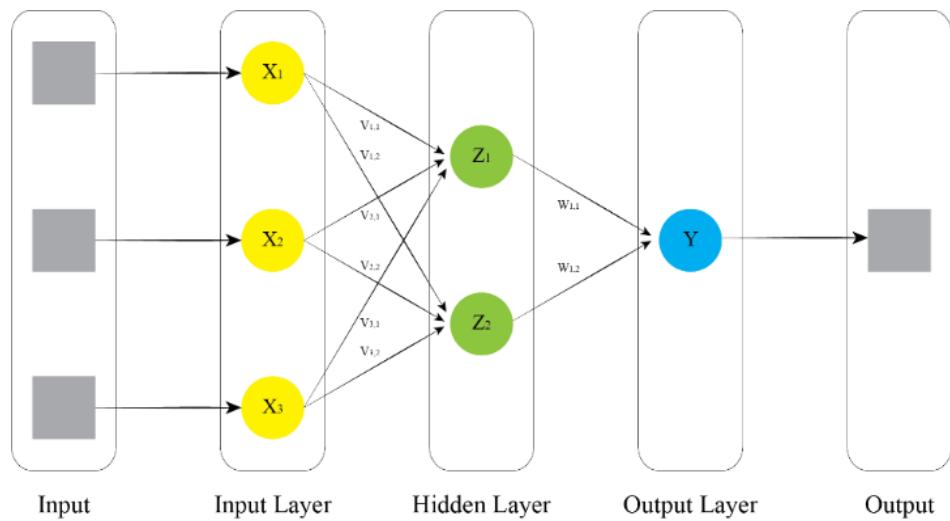


Gambar 2.4 Arsitektur Single Layer

Sumber: Mubarok (2019) telah diolah kembali

2) Multi Layer

Arsitektur *Multi Layer* pada ANN sering disebut dengan *Multi Layer Perceptron* (MLP). MLP memiliki banyak *layer* yang paling sedikit setidaknya satu *layer* di antara *input layer* dan *output layer*. Lapisan tambahan pada MLP dapat disebut dengan *hidden layer*. MLP biasanya digunakan untuk menyelesaikan permasalahan sulit yang tidak bisa diatasi oleh arsitektur *Single Layer* (Mubarok, 2019). Contoh arsitektur *multi layer* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Arsitektur *Multi Layer*

Sumber: Mubarok (2019) telah diolah kembali

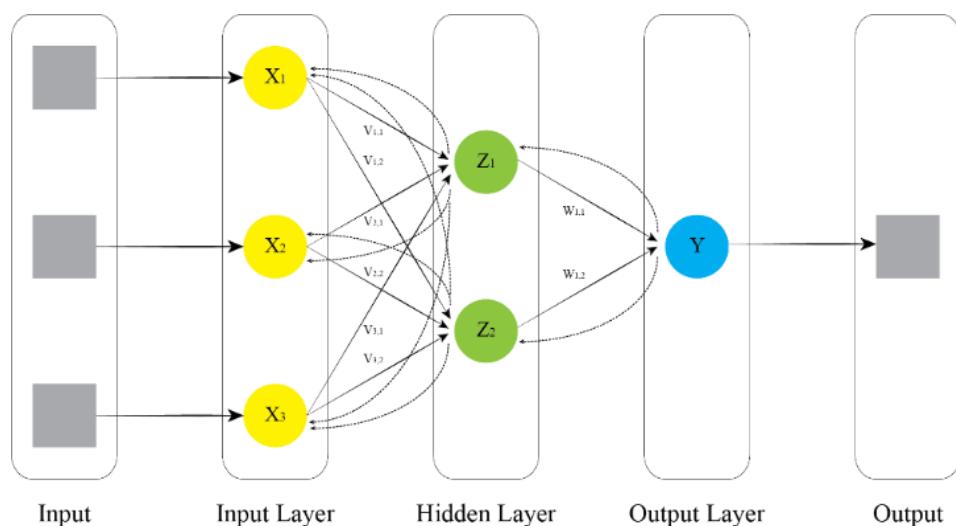
2.1.7 Backpropagation

Jaringan Saraf Tiruan dengan *layer* tunggal memiliki keterbatasan dalam melakukan pengenalan terhadap pola informasi. Keterbatasan ini dapat diatasi dengan menambahkan satu atau beberapa *layer* tersembunyi di antara *layer input* dan *layer output*. *Backpropagation* melatih jaringan dengan menyeimbangkan antara kemampuan jaringan untuk mengenali sebuah pola yang digunakan selama pelatihan data. *Backpropagation* juga memiliki kemampuan memberikan respons yang benar terhadap pola masukan yang serupa dengan pola yang dipakai selama pelatihan (Cynthia & Ismanto, 2017).

Menurut Faradiba (2017) *Backpropagation* merupakan Jaringan Saraf Tiruan dengan pembelajaran yang *supervised*. Data yang digunakan selama proses

pembelajaran terdiri dari vektor *input* dan target *output*. Algoritma *backpropagation* sering digunakan dalam pengenalan pola, klasifikasi data, pengolahan citra, dan penarikan kesimpulan. Pelatihan dengan menggunakan algoritma *backpropagation* meliputi dua tahap yaitu, perambat maju dan mundur.

Menurut Siang (2005) *Backpropagation* merupakan sebuah algoritma terawasi berupa *perceptron* yang mempunyai banyak lapisan yang digunakan untuk mengubah bobot yang melalui *neuron* tersembunyi sehingga menghasilkan suatu *output*. Metode *Backpropagation* menggunakan perhitungan yang disebut *error output* untuk mengubah bobot selanjutnya dengan alur mundur. *Backpropagation* digunakan melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan serta memberi respons yang benar terhadap pola masukan.



Gambar 2.6 Arsitektur Backpropagation

Sumber: Mubarok (2019) telah diolah kembali

Dari pemaparan di atas, peneliti menyimpulkan bahwa *Backpropagation* merupakan penerapan lanjutan dari Jaringan Saraf Tiruan dengan pembelajaran yang *supervised*. *Backpropagation* melatih jaringan dengan memperoleh keseimbangan antara kemampuan jaringan dalam proses pengenalan pola informasi dengan memanfaatkan *hidden layer* yang terdapat di antara *layer input* dan *layer output*. Algoritma ini menerapkan metode *error input* untuk mengubah bobot-bobot yang masuk dengan alur mundur. *Backpropagation* yang telah dilatih dengan baik akan menghasilkan *output* yang masuk akal sesuai dengan pola yang dipakai selama

proses pelatihan pola informasi. Pada Gambar 2.6 adalah arsitektur *Backpropagation* dengan satu *hidden layer*. Adapun macam-macam tipe dari fungsi aktivasi adalah sebagai berikut:

1) *Sigmoid Biner*

Fungsi ini umumnya digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode *Backpropagation* yang memiliki nilai antara 0 sampai 1. Fungsi *sigmoid biner* dirumuskan sebagai:

$$\varphi(s) = \frac{1}{1 + e^{-x}}$$

2) *Sigmoid Bipolar*

Fungsi ini memiliki *range* antara 1 sampai -1. Fungsi *sigmoid bipolar* dirumuskan sebagai:

$$\varphi(s) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

3) *ReLU*

ReLU merupakan fungsi aktivasi pada *Artificial Neural Network* yang saat ini banyak digunakan. Jika input lebih besar 0, *output*-nya sama dengan input. Fungsi *ReLU* lebih menyerupai sebuah *neuron* seperti pada tubuh manusia. Aktivasi *ReLU* pada dasarnya memiliki sebuah fungsi aktivasi non-linier yang paling sederhana. Bila mendapatkan hasil *input* positif, turunannya hanya 1, dengan kata lain fungsi aktivasi hanya melakukan *threshold* pada nilai nol. Dari berbagai penelitian yang menggunakan fungsi aktivasi *ReLU* menunjukkan bahwa *ReLU* menghasilkan pelatihan yang lebih cepat untuk jaringan besar. Fungsi *ReLU* dirumuskan sebagai:

$$f(x) = \max(x, 0)$$

2.1.8 Deep Learning

Deep Learning adalah sebuah teknik *machine learning* yang memungkinkan komputer dapat menyerupai kemampuan manusia melalui proses pengalaman dan belajar. Sesuai namanya, *deep learning* merupakan pembelajaran dalam yang melakukan proses pembelajarannya lebih banyak dan membutuhkan perangkat dengan performa yang tinggi juga. *Deep Learning* menggabungkan

proses *feature extraction* dan klasifikasi secara bersamaan dalam melakukan pencocokan *pattern recognition*. Di antara macam-macam produk yang menerapkan metode *deep learning* antara lain *voice recognition*, *face recognition*, dan *self-driving car*. *Deep Learning* dikenal dan sering digunakan oleh banyak orang karena performanya yang gemilang (Mubarok, 2019).

Menurut Hanafi (2020) *deep learning* merupakan implementasi lanjutan dari *machine learning*. *Deep Learning* mengadopsi arsitektur MLP dalam pola pembelajarannya. Kelebihan *deep learning* adalah mampu menemukan relasi tersembunyi atau pola yang rumit antara input dan output, yang tidak dapat diselesaikan menggunakan MLP. Selain itu, *deep learning* juga memiliki kemampuan untuk mengidentifikasi *decision boundary* berbentuk *non-linier* dan melakukan simulasi interaksinya antar fitur. *Deep Learning* mengadopsi proses yang diterapkan pada ANN *Backpropagation* dalam proses pembelajaran atau *training*.

Nurhikmat (2018) berpendapat bahwa *deep learning* adalah salah satu bidang *machine learning* yang menggunakan teknik *Artificial Neural Network* untuk memproses data dengan jumlah yang besar. *Deep Learning* termasuk kepada *supervised learning* yang sangat kuat karena arsitekturnya yang kompleks. Namun, metode *deep learning* pun masih memiliki kekurangan yaitu pemrosesannya yang relatif lama dan memerlukan data latih yang besar untuk mendapatkan akurasi yang maksimal.

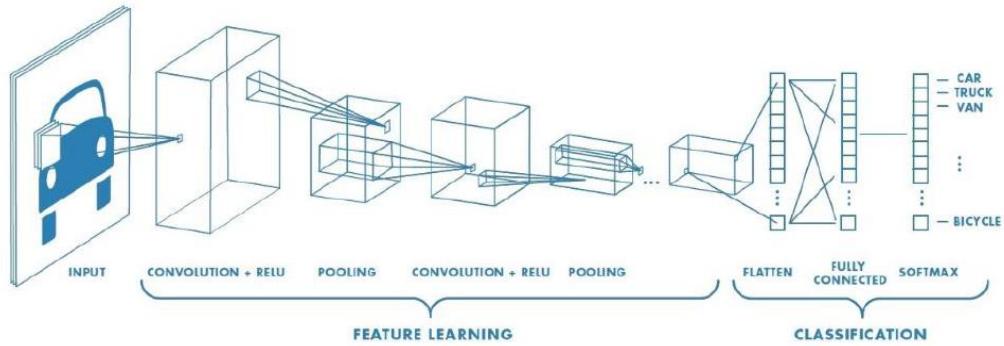
Dari beberapa pemaparan di atas, peneliti menyimpulkan bahwa *deep learning* adalah bagian dari *machine learning* yang fokus kepada pembelajaran dalam. Arsitektur *deep learning* pada umumnya menggunakan arsitektur MLP pada algoritma ANN. *Deep Learning* sendiri memiliki perbedaan yang signifikan dibandingkan *neural network* umumnya, yaitu *deep learning* menggabungkan proses *feature extraction* dan klasifikasi secara bersamaan. *Deep Learning* dinilai mampu dalam mengolah data citra, sinyal, teks, dan lain sebagainya dalam jumlah data yang besar. Namun dibalik keunggulannya, *deep learning* juga memiliki kekurangan yaitu proses komputasi yang relatif lama dan membutuhkan banyak data.

2.1.9 *Convolutional Neural Network*

Convolutional Neural Network (CNN) merupakan penerapan metode *deep learning* yang memungkinkan komputer dapat mendeteksi hingga mengenali suatu objek pada citra digital. CNN diklaim memiliki kemampuan terbaik dalam menyelesaikan masalah *object detection* dan *object recognition*. Dalam teknis pembelajarannya, CNN memiliki beberapa tahapan yang dilaluinya antara lain *input* dan *output* dari tiap-tiap tahapan yang berupa *array* atau *feature map*. Pada tiap tahapannya terdiri dari tiga *layer* yaitu konvolusi, fungsi aktivasi, dan *pooling*. Namun layaknya metode *Deep Learning* lainnya, CNN memiliki kekurangan dalam kompleksitas waktu yang cenderung besar ketika melakukan pelatihan model (Nurhikmat, 2018).

Menurut Mubarok (2019) CNN memiliki dasar yang meniru struktur ANN yang melakukan pembelajaran dari beberapa lapisan. Dapat dikatakan bahwa CNN merupakan penerapan lanjutan dari metode ANN. Walaupun demikian, dilihat pada strukturnya CNN sangatlah berbeda dengan ANN pada umumnya. CNN menggunakan teori konvolusi dalam proses *feature learning* yang juga merupakan ciri dari CNN itu sendiri. CNN juga memiliki kekurangan yaitu proses komputasi model yang relatif lama. Namun hal ini dapat diselesaikan dengan *Graphical Processing Unit* (GPU) dan spesifikasi komputer yang tinggi.

Menurut Mien Fatimah Az-Zahra (2019) CNN adalah arsitektur yang mampus mengenali informasi dari gambar, teks, suara, dan lain sebagainya. CNN merupakan pengembangan lanjutan dari MLP yang dibentuk sedemikian rupa sehingga mampu mengolah data citra. CNN termasuk ke dalam metode *Deep Learning* yang sudah banyak digunakan dalam memproses citra. CNN memiliki beberapa *layer* pada arsitekturnya antara lain *convolution layer*, *activation layer*, dan *fully connected layer*. Visualisasi dari arsitektur *Convolutional Neural Network* (CNN) dapat dilihat pada Gambar 2.7.



Gambar 2.7 Arsitektur Convolutional Neural Network (CNN)

Sumber: Mien Fatimah Az-Zahra (2019)

Dari beberapa pemaparan di atas, peneliti dapat menyimpulkan bahwa CNN merupakan algoritma yang fantastis. CNN merupakan penerapan dari *deep learning* yang mampu menyelesaikan permasalahan *object detection* dan *object recognition* secara cepat. Sesuai namanya CNN mengadopsi teori konvolusi dalam melakukan proses *feature learning*. Adapun arsitektur CNN antara lain *convolution layer*, *activation layer*, dan *fully connected layer*. Dibalik keunggulan yang sangat fantastis, CNN memiliki kekurangan dalam hal penggunaannya yang terbatas karena membutuhkan spesifikasi tinggi dan memanfaatkan teknologi GPU dalam menunjang pemrosesannya agar lebih baik dan cepat.

2.1.10 *Object Detection*

Object Detection merupakan salah satu teknik dari penerapan teknologi *Computer Vision* yang dapat mendeteksi objek pada sebuah citra. Pada konsep *Object Detection* terdapat dua bagian yang masing-masing fungsinya berbeda yaitu *soft detection* dan *hard detection*. Pada proses pendekripsi objek menggunakan *soft detection*, mesin hanya dapat mendekripsi sebuah objek secara khusus, sedangkan jika pendekripsi objek menggunakan *hard detection* mesin dapat mendekripsi objek tersebut serta lokasi atau latar belakang di mana tempat objek itu didekripsi (Of et al., 2020)

Menurut Syarifah Rosita Dewi (2018) *object detection* memiliki fungsi untuk menentukan ada tidaknya suatu objek dalam suatu ruang lingkup beserta

lokasinya pada sebuah citra. Dalam hal ini, *object detection* diperlakukan sebagai pengenalan objek dua kelas, di mana kelas yang pertama mewakili kelas dari objek secara khusus dan kelas lainnya mewakili non-objek. Dalam proses pendekalian, *object detection* dibagi lagi menjadi dua bagian yaitu *soft detection* dan *hard detection*. *Soft detection* mendekripsi objek secara khusus dengan mengabaikan hal lain di sekitarnya, sedangkan *hard detection* mendekripsi objek beserta lokasi objek.

Menurut Cahya Rahmad, Rawansyah, dan Tanggon Kalbu Rochastu (2018) *object detection* atau pendekalian objek merupakan bagian daripada penerapan *Computer Vision*. *Object Detection* mengacu kepada kemampuan sebuah mesin atau komputer dalam mendekripsi satu atau banyak objek pada suatu citra. *Object Detection* memanfaatkan beberapa *feature* untuk digunakan dalam proses pendekalian seperti garis, sudut, kontur, dan warna dari sebuah citra. Sehingga dapat disimpulkan bahwa *object detection* perlu diidentifikasi terlebih dahulu untuk dapat mendapatkan objek yang dicari.

Berdasarkan pemaparan di atas, *object detection* merupakan penerapan dari *Computer Vision* yang bergerak di bidang pendekalian suatu objek dalam sebuah citra. Dalam melakukan proses deteksi, *object detection* dibagi menjadi dua bagian yaitu *soft detection* dan *hard detection*. *Soft detection* berfokus kepada objek yang akan didekripsi dan menghiraukan objek-objek lainnya, sedangkan *hard detection* memiliki dua pernah yaitu pendekalian target objek secara khusus dan lokasi di mana objek itu didekripsi. *Object Detection* melakukan proses pengambilan informasi dari citra yang akan didekripsi dan akan menghasilkan sebuah *image feature* yang terdiri dari garis, sudut, kontur, dan warna.

2.1.11 *Learning Rate*

Pada proses pelatihan *Artificial Neural Network* terdapat penggunaan parameter *learning rate* yang memiliki pengaruh besar terhadap waktu dan kedalaman proses pembentukan model untuk tercapainya target yang diinginkan. Secara perlahan model akan mengoptimalkan nilai perubahan bobot dan akan terus menghasilkan *error* yang setiap iterasinya semakin mengecil (Fajri, 2011). Variabel yang terdapat pada penggunaan parameter *learning rate* berjarak antara 0,1 sampai

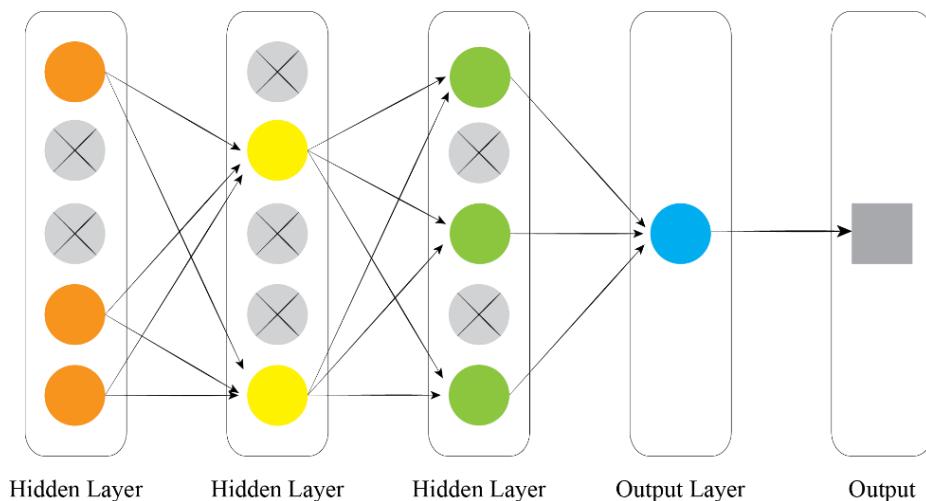
dengan 0,9. Nilai tersebut menunjukkan seberapa cepat sebuah model belajar dari jaringan.

Pada penggunaan parameter *learning rate* pada sebuah jaringan terdapat kondisi yang perlu diperhatikan. Jika nilai *learning rate* yang diinisialisasi terlalu kecil, maka *epoch* yang dibutuhkan juga semakin banyak agar dapat mencapai nilai target yang diinginkan, sehingga mengakibatkan proses pembelajaran membutuhkan lebih banyak waktu. Sebaliknya jika semakin besar nilai *learning rate* yang diinisialisasi maka proses pembelajaran dan pelatihan jaringan akan semakin cepat, namun jika nilai *learning rate* yang terlalu besar akan mengakibatkan proses pelatihan model jaringan menjadi tidak stabil dan menyebabkan *error* yang berulang. Oleh karena itu, pada proses inisialisasi parameter *learning rate* perlu nilai yang optimal agar didapatkan proses *training* yang cepat dan akurat (Hermawan, 2006).

2.1.12 Dropout Regularization

Regularization merupakan sebuah teknik yang digunakan pada saat pembuatan model yang bertujuan untuk mengurangi *overfitting* pada model yang dihasilkan. *Overfitting* adalah di mana kondisi suatu model *Artificial Neural Network* atau jaringan saraf tiruan mampu belajar dengan baik dari data pelatihan, namun tidak mampu melakukan generalisasi pada data uji. Terdapat beberapa teknik dalam penggunaan *regularization* seperti L2 *regularization* dan *dropout*.

Dropout merupakan proses pencegahan terjadinya *overfitting* pada suatu pelatihan dan pengujian model dan juga mempercepat proses pembelajaran atau *learning*. *Dropout* bekerja dengan cara menghilangkan beberapa *neuron* yang terletak pada *hidden layer* ataupun *visible layer* dalam suatu jaringan. Dengan terhapusnya suatu *neuron*, maka juga berpengaruh kepada jaringan yang ada. *Neuron* yang dihilangkan akan dipilih secara acak. Pada implementasi *dropout* pada *Artificial Neural Network*, perlu adanya parameter probabilitas berjarak antara 0 sampai dengan 1 yang merupakan jumlah persentase dari *neuron* yang akan dihapus sementara. Contoh visualisasi adanya penambahan *dropout* pada arsitektur *Artificial Neural Network* pada Gambar 2.8 di bawah ini.



Gambar 2.8 Visualisasi Dropout Layer

Sumber: Triano Nurhikmat (2018) telah diolah kembali

2.1.13 Tensorflow

Tensorflow merupakan kumpulan kode atau *library package* perangkat lunak yang mampu membantu dalam pemrosesan *deep learning*. *Tensorflow* dikembangkan secara khusus oleh Tim Google Brain dalam organisasi Mesin Cerdas Google. Pembuatan *package* *Tensorflow* bertujuan untuk melakukan pembelajaran *machine learning* dan penelitian *deep learning* semakin mudah dan cepat. *Tensorflow* menggabungkan komputasi teknik pengoptimalan komputasi, mempermudah penghitungan ekspresi matematika di mana masalah terbesarnya adalah waktu yang cukup banyak saat melakukan perhitungan yang kompleks. *Tensorflow* memiliki fitur-fitur utama yang meliputi:

- 1) Mendefinisikan *layer-layer*, mengoptimalkan hubungan antara *layer* satu ke *layer* berikutnya, dan menghitung secara cepat dan singkat ekspresi matematis yang melibatkan *array* berdimensi besar atau *multidimensional* (*tensors*).
- 2) Mendukung pembangunan model *Artificial Neural Network* dan *deep learning* serta teknik *machine learning*.
- 3) Penggunaan teknologi GPU yang mampu mengoptimalkan manajemen dan penggunaan memori saat melakukan pembelajaran. *Tensorflow* bisa menjalankan kode yang sama baik menggunakan CPU ataupun GPU.

- Terlebih lagi khususnya, *Tensorflow* mampu memberikan informasi bagian perhitungan mana yang perlu dipindahkan ke GPU agar proses lebih cepat.
- 4) Mampu memproses skala komputasi yang tinggi di seluruh mesin dan kumpulan himpunan data yang besar.

2.2 Penelitian Terdahulu

Syarifah Rosita Dewi (2018) telah menyelesaikan penelitiannya dengan judul pada proses klasifikasi gambar meja dan kursi motif ukiran Jepara. Penelitian ini berfokus kepada penerapan CNN dan *Tensorflow* untuk melakukan identifikasi dan klasifikasi. Langkah-langkah yang dilakukan antara lain pengumpulan data, perancangan *preprocessing* dengan pelabelan, konversi himpunan data berekstensi *xml* ke *csv*, konversi himpunan data berekstensi *csv* ke *TFRecord* dan *resizing* citra dan label map. Hasil yang diperoleh adalah klasifikasi meja dan kursi pada suatu citra digital menggunakan *Convolutional Neural Network* dapat dinilai bekerja dengan baik. Tingkat akurasi model yang didapatkan dari hasil pendekripsi klasifikasi citra meja dan kursi motif ukiran Jepara pada suatu citra digital menggunakan *Convolutional Neural Network* berkisar antara 70% hingga 99%.

Hamdani Mubarok (2019) pernah melakukan penelitian dengan ekspresi wajah sebagai objek utamanya. Penelitian ini menggunakan metode *Convolutional Neural Network* (CNN) dalam melakukan identifikasi dan klasifikasi. Langkah-langkah yang dilakukan dalam identifikasi dan klasifikasi ekspresi wajah antara lain pengumpulan dan, pembagian data menjadi data latih dan data uji, *preprocessing* data berupa *cropping* dan *grayscaled* serta *augmented*, *image processing* dengan CNN, dan *output* berupa *probability* ekspresi wajah. Hasil yang diperoleh bahwa algoritma CNN mampu melakukan identifikasi ekspresi wajah. Akurasi *training* yang diperoleh sebesar 99.6% dengan 1000 *epoch* dan *learning rate* sebesar 0.001. Pada saat melakukan *testing* akurasi yang diperoleh sebesar 88.89%.

Chairisni Lubis dan Novario Jaya Perdana (2020) sudah menyelesaikan penelitiannya dengan melakukan kombinasi antara metode YOLO dan CNN untuk pendekripsi dan pengenalan ekspresi wajah. Adapun langkah-langkah yang diterapkan dengan metode CNN yaitu pembagian data latih dan data uji, normalisasi, *convolution layer*, *ReLU*, *pooling layer*, *flatten*, *fully connected layer*,

softmax dan *output* berupa hasil pengenalan ekspresi wajah. Pengujian YOLO menggunakan *webcam* untuk melakukan *real-time* identifikasi dan pengenalan ekspresi wajah manusia.

Hasil yang diperoleh yaitu model YOLO yang dilatih dengan 150 epoch menghasilkan tingkat keyakinan sebesar 85% yang berarti model sudah dapat mengenali wajah dengan baik. Kemudian model CNN yang berarsitektur ResNet50 memiliki performa kurang baik yaitu akurasi sebesar 46.42%. Adanya ketidakseimbangan jumlah data saat melakukan proses pelatihan dan pembelajaran model menjadi faktor utamanya. Pada akhirnya, dari 6 kelas ekspresi wajah yang dilatih hanya 3 kelas yang berhasil dilakukan identifikasi dan pengenalan sesuai dengan ekspresi aslinya.

Wulan Anggraini (2020) pernah melakukan penelitian pada proses deteksi wajah berhijab dengan algoritma CNN dengan *Tensorflow*. Langkah-langkah yang dilakukan untuk melakukan deteksi wajah berhijab antara lain pengumpulan data wajah berhijab dengan rincian 800x650 piksel resolusi 500dpi berjumlah 300 data, pembagian data latih dan data uji, *preprocessing* dengan melakukan *resize* citra, melakukan klasifikasi dengan CNN. Hasil yang diperoleh adalah pemodelan CNN yang telah dibuat yaitu data latih sebesar 92% dan data uji sebesar 87%, tingkat keakuratan yang paling tinggi yaitu dengan menggunakan nilai *epoch* 100 sebesar 90%, jumlah nilai *learning rate* yang menghasilkan tingkat keakuratan yang tinggi yaitu 0,0001 dengan keakuratan 89%. Sistem membutuhkan waktu yang cukup lama untuk memproses 300 data citra.

Tabel 2.2 Penelitian Terdahulu

Judul	Peneliti	Tahun	Metode	Kesimpulan	Relevansi
Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network	Syarifah Rosita Dewi	2018	<i>Convolutional Neural Network (CNN)</i> dan <i>Tensorflow</i>	<ul style="list-style-type: none"> 1. Pendekripsi klasifikasi meja dan kursi pada suatu citra digital menggunakan CNN dapat dinilai bekerja dengan baik. 2. Akurasi yang diperoleh berkisar antara 70% hingga 99%. 	Metode yang diterapkan pada penelitian ini berkaitan
Identifikasi Ekspresi Wajah Berbasis Citra Menggunakan Algoritma <i>Convolutional Neural Network (CNN)</i>	Hamdani Mubarok	2019	<i>Convolutional Neural Network (CNN)</i>	<ul style="list-style-type: none"> 1. Algoritma CNN mampu melakukan identifikasi ekspresi wajah. 2. Akurasi <i>training</i> yang diperoleh sebesar 99,6% dengan 1000 <i>epoch</i> dan <i>learning rate</i> sebesar 0,001. 3. Pada saat melakukan testing akurasi yang diperoleh sebesar 88,89%. 	Metode yang diterapkan pada penelitian ini berkaitan
Sistem Pendekripsi Dan Pengenalan Ekspresi Wajah Dengan Algoritma Yolo Dan Convolutional Neural Network	Chairisni Lubis dan Novario Jaya Perdana	2020	<i>Convolutional Neural Network (CNN)</i> dan <i>YOLO</i>	<ul style="list-style-type: none"> 1. YOLO yang dilatih dengan 150 <i>epoch</i> menghasilkan tingkat keyakinan sebesar 85%. 2. Model CNN yang berarsitektur ResNet50 memiliki performa kurang baik yaitu akurasi sebesar 46,42%. 3. Faktor utama penyebab kurang baiknya hasil CNN yaitu adanya ketidakseimbangan jumlah data saat melakukan proses <i>training</i>. 4. Dari 6 kelas ekspresi wajah yang dilatih hanya 3 kelas yang berhasil dilakukan identifikasi sesuai dengan aslinya. 	Metode yang diterapkan pada penelitian ini berkaitan

Judul	Peneliti	Tahun	Metode	Kesimpulan	Relevansi
Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (CNN) Dengan Tensorflow	Wulan Anggraini	2020	<i>Convolutional Neural Network (CNN)</i> dan <i>Tensorflow</i>	<ul style="list-style-type: none"> 1. Akurasi yang dihasilkan oleh model CNN yaitu data latih sebesar 92% dan data uji sebesar 87%. 2. Tingkat keakuratan yang paling tinggi yaitu dengan nilai <i>epoch</i> 100 sebesar 90%. 3. Sistem membutuhkan waktu yang cukup lama untuk memproses 300 data citra. 4. Jumlah nilai <i>learning rate</i> yang menghasilkan tingkat keakuratan yang tinggi yaitu 0,0001 dengan keakuratan 89%. 	Metode yang diterapkan pada penelitian ini berkaitan

2.3 Metodologi Pemecahan Masalah

Dari penelitian terdahulu yang sudah dijabarkan sebelumnya, terdapat beberapa metode yang sama. Adapun rangkuman dari berbagai metodologi yang digunakan untuk memecahkan masalah pada tiap-tiap penelitian dapat dilihat pada Tabel 2.3.

Tabel 2.3 Metodologi Pemecah Masalah

Nama Metodologi	Relevansi
<i>Artificial Neural Network</i> (ANN)	Arsitektur yang digunakan dalam membuat model pembelajaran <i>machine learning</i>
<i>Deep Learning</i>	Teknik pembelajaran dalam yang digunakan dalam penelitian ini guna mampu menghasilkan model yang akurat dalam jumlah data yang besar
<i>Convolutional Neural Network</i> (CNN)	Teknik konvolusi yang diterapkan pada pemrosesan citra dengan menggerakan sebuah kernel konvolusi yang mampu meningkatkan kualitas identifikasi dan prediksi data citra
<i>You Only Look Once</i> (YOLO)	Teknik <i>deep learning</i> yang memanfaatkan jaringan saraf konvolusional (CNN) yang membagi citra ke dalam <i>grid</i> kemudian tiap <i>grid</i> akan memprediksi <i>bounding box</i> serta peta kelas masing-masing <i>grid</i>
<i>Dropout Regularization</i>	Metode yang digunakan untuk mengurangi <i>overfitting</i> pada model yang akan dihasilkan
<i>Learning Rate</i>	Parameter yang digunakan selama pelatihan model untuk menghitung nilai bobot (<i>weight</i>)
<i>Transfer Learning</i>	Metode yang digunakan untuk mempercepat proses pembelajaran dengan memanfaatkan model yang sudah ada sebelumnya
<i>Tensorflow</i>	<i>Library</i> atau <i>package</i> yang membantu dalam proses pembangunan arsitektur serta pembelajaran model <i>Neural Network</i>

Dari keseluruhan metodologi yang digunakan di penelitian terdahulu, peneliti mengambil beberapa metodologi untuk digunakan dalam memecahkan masalah yang diangkat dalam penelitian ini. Adapun metode yang digunakan dalam penelitian ini beserta alasannya adalah sebagai berikut:

1) *Deep Learning*

Penggunaan teknik *deep learning* merupakan hal utama yang dibahas pada penelitian ini khususnya dengan menggunakan metode *Convolutional Neural Network*. Berdasarkan penelitian sebelumnya, teknik ini memberikan hasil evaluasi yang sangat baik dengan syarat yaitu jumlah data yang besar serta proses pelatihannya yang akan memakan *resource* dan waktu yang cukup besar.

2) *Convolutional Neural Network*

Arsitektur yang dibangun menggunakan mekanisme *Convolutional Neural Network* memiliki hasil deteksi dan klasifikasi yang baik. Arsitektur ini biasa digunakan saat pelatihan model dengan data yang digunakan berbasis citra.

3) *Dropout Regularization*

Penggunaan parameter ini sangat berpengaruh pada evaluasi akhir model. Pada setiap proses pelatihan model khususnya pada *neural network* ini sangat berpotensi menghasilkan model yang *overfitting*. Penggunaan parameter *dropout* ini diharapkan dapat mengatasi kendala dan masalah tersebut.

4) *Learning Rate*

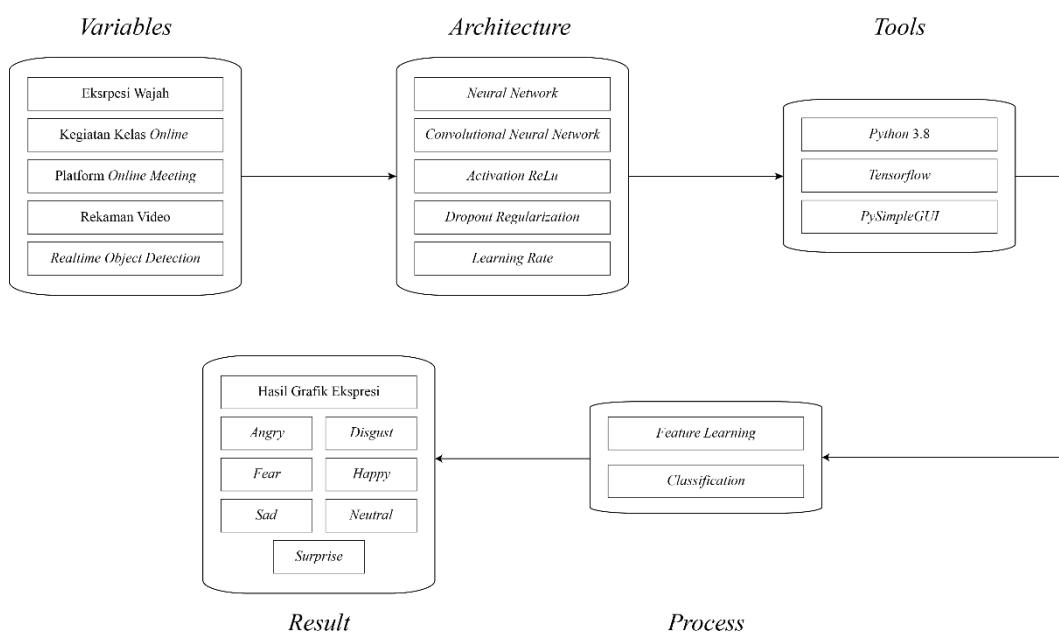
Penggunaan besar nilai *learning rate* yang berbeda dapat berpengaruh kepada hasil yang diperoleh dari setiap *epoch* saat pelatihan model. Semakin besar nilai *learning rate* maka akan semakin cepat mesin belajar yang tentu juga akan mengakibatkan beberapa momen penting dalam perhitungan bobot terlewat. Maka penggunaan besar nilai *learning rate* juga sangat berpengaruh kepada hasil akhir model.

5) *Tensorflow*

Tensorflow merupakan *framework* yang andal dalam membentuk dan membangun arsitektur *Neural Network* dengan sangat mudah. *Tensorflow* memiliki banyak sekali fungsi matematika yang dengan mudah diimplementasikan dalam proses pembelajaran mesin.

2.4 Kerangka Pemikiran

Hubungan antara setiap variabel dengan faktor-faktor lainnya yang menjadi alur logis dalam penyelesaian masalah dalam penelitian ini. Adapun variabel yang digunakan antara lain ekspresi wajah, kegiatan kelas *online*, platform *online meeting*, rekaman video dan *real-time object detection* yang kemudian dihubungkan dengan pembangunan arsitektur menggunakan teknik pendekatan *Deep Learning*. Pembangunan arsitektur tersebut meliputi *Neural Network*, *Convolutional Neural Network*, *activation ReLu*, *Dropout*, dan *Learning Rate*. Arsitektur tersebut dibangun menggunakan *tools Python 3.8*, *library Tensorflow*, dan *PySimpleGUI*. Kemudian dilakukan proses *feature learning* dan klasifikasi yang kemudian menghasilkan grafik *accuracy* dan *loss* sebagai evaluasi model. Adapun bagan kerangka pemikiran dapat dilihat pada Gambar 2.9.



Gambar 2.9 Kerangka Pemikiran

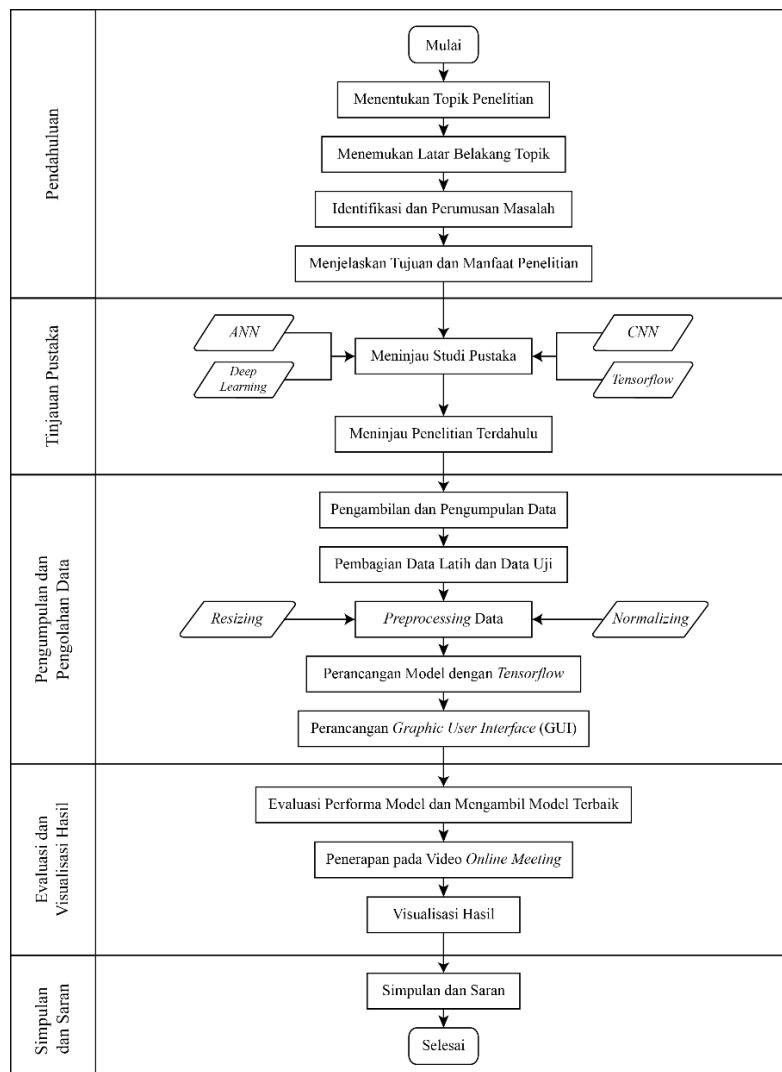
BAB 3

METODE PENELITIAN

Pada bab ini berisi alur penelitian, subjek dan objek penelitian, metode pengumpulan data, instrumen penelitian, dan metode analisis, pengujian serta interpretasi data yang digunakan selama proses penelitian ini berlangsung.

3.1 Alur Penelitian

Dalam melakukan eksperimen pada penelitian ini, perlu dilakukan beberapa tahapan. Gambar 3.1 merupakan representasi dari tiap-tiap tahapan yang dilakukan dalam pelaksanaan penelitian ini.



Gambar 3.1 Alur Penelitian

Berdasarkan pada gambar di atas, proses alur penelitian ini dibagi menjadi lima bagian agar pada pelaksanaan penelitian ini dapat terukur dan terarah. Pada proses yang pertama peneliti memulai penelitian dengan mencari topik pembahasan dengan mulai mencari permasalahan disekitar yang dapat diselesaikan dengan metode-metode yang dipelajari di ilmu komputer. Setelah menentukan topik, peneliti perlu menemukan latar belakang dari permasalahan tersebut dan mengidentifikasi permasalahan serta menuliskannya dalam rumusan masalah. Hal ini dilakukan peneliti agar penelitian ini hanya memiliki satu titik fokus permasalahan saja. Selain itu, peneliti juga menjelaskan tujuan dan manfaat yang diperoleh dari hasil penelitian nanti.

Proses yang kedua berfokus kepada studi literatur ke berbagai buku, jurnal, skripsi, dan *website*. Studi yang dilakukan guna mempelajari pengetahuan lebih lanjut dari penelitian terkait dan menjadi referensi tulisan. Studi literatur yang dilakukan meliputi berbagai materi tentang *Artificial Neural Network*, *Deep Learning*, *Convolutional Neural Network* (CNN), dan *Tensorflow*. Selain melakukan studi literatur terhadap berbagai materi yang sudah disebutkan, peneliti juga melakukan peninjauan ke beberapa peneliti terdahulu yang juga sempat melakukan penelitian dengan topik atau metode yang sama.

Pada proses yang ketiga waktu yang dibutuhkan cukup lama karena pada proses ini peneliti melakukan pengumpulan dan pengolahan data. Proses pengambilan dan pengumpulan data berasal dari dua sumber yang berbeda. Untuk pelatihan dan pengujian model data yang digunakan berasal dari *website* Kaggle dengan pemilik himpunan data adalah Jonathan Oheix yang terakhir diunggah pada tahun 2019. Sedangkan untuk implementasi secara pada rekaman video kegiatan kelas *online* diperoleh dari data rekaman kegiatan kelas *online* melalui izin dari pihak Bagian Akademik STIMIK ESQ. Selain itu pada proses yang ketiga juga terdapat proses pelatihan model menggunakan parameter yang berbeda yang akan dibandingkan performanya untuk dijadikan dasar dari proses identifikasi ekspresi wajah pada sistem ini.

Proses Keempat merupakan proses evaluasi performa model dari tiap-tiap model dengan parameter yang berbeda yang telah diterapkan pada proses ketiga.

Proses evaluasi dilakukan untuk menguji seberapa besar akurasi dan *loss* yang dihasilkan dalam mengidentifikasi ekspresi wajah. Setiap *epoch* yang dijalani selama pelatihan model akan disimpan dengan performa yang paling baik. Setelah itu setiap model akan dibandingkan untuk mendapatkan model yang terbaik untuk digunakan dalam pendekripsi ekspresi wajah dalam sistem. Selain melakukan evaluasi performa, pada proses ini juga menampilkan hasil dari penerapan pada rekaman video kegiatan kelas *online* pada platform *online meeting* yang ditampilkan berupa grafik. Menurut peneliti hal ini sangat penting, mengingat tidak hanya evaluasinya saja yang ditampilkan melainkan visualisasi dari hasilnya pun perlu ditampilkan.

Pada proses terakhir atau proses kelima, peneliti melakukan penarikan simpulan yang diperoleh dari proses keempat. Proses ini menjadi proses yang penting mengingat hal ini yang menjadi sorotan bagi setiap penelitian yang dilakukan. Simpulan berisi apakah metode yang digunakan menghasilkan akurasi yang bagus dan data yang digunakan cocok dengan metode tersebut. Tentunya pada setiap penelitian pun akan ada pengembangan tingkat lanjut, maka pada proses terakhir ini peneliti juga memberikan saran kepada peneliti selanjutnya dalam pengembangan penelitian bertema pendekripsi ekspresi wajah melalui rekaman video pada platform *online meeting*.

3.2 Subjek dan Objek Penelitian

Pada subbab ini menjelaskan tentang subjek dan objek penelitian yang dijadikan sasaran dalam penelitian ini. Adapun subjek dan objek penelitian tersebut adalah sebagai berikut:

3.2.1 Subjek Penelitian

Subjek penelitian merupakan sasaran atau tempat di mana suatu variabel melekat. Menurut Arikunto (2010) subjek penelitian adalah tempat diperolehnya suatu variabel data yang digunakan dalam penelitian. Secara umum, subjek penelitian ini adalah manusia. Namun jika digambarkan secara rinci berdasarkan topik dan tujuan penelitian ini, subjek penelitian meliputi manusia atau orang yang mengikuti kegiatan kelas secara *online* melalui rekaman video kegiatan kelas *online* pada platform *online meeting*.

3.2.2 Objek Penelitian

Objek penelitian merupakan variabel yang diteliti pada yang melekat pada sebuah subjek penelitian. Sugiyono (2012) berpendapat bahwa objek penelitian adalah atribut inti dari suatu kegiatan, orang ataupun objek tertentu yang memiliki variansi yang kemudian dipelajari dan ditarik simpulannya. Pada penelitian ini, fokus utama objek penelitian adalah emosi manusia yang ditampilkan melalui ekspresi wajah saat mengikuti kegiatan kelas secara *online* melalui rekaman video kegiatan kelas *online* pada platform *online meeting*.

3.3 Metode Pengumpulan Data

Pada subbab ini menjelaskan tentang metode pengumpulan data yang meliputi jenis data dan sumber data. Adapun rincian dari metode pengumpulan data tersebut adalah sebagai berikut:

3.3.1 Jenis Data

Pada penelitian ini, jenis data yang digunakan adalah data kualitatif. Data kualitatif adalah data yang berbentuk kata, kalimat atau gambar (Sugiyono, 2015). Penelitian ini menggunakan citra sebagai data utamanya. Data citra yang digunakan dibagi menjadi dua yaitu data citra yang tersaji sebagai sebuah gambar dan data citra yang masih dalam berbentuk video.

3.3.2 Sumber Data

Data yang akan diolah pada penelitian ini bersumber dari dua jenis data yaitu premier dan sekunder. Adapun rincian dari karakteristik data tersebut adalah sebagai berikut:

1) Data Primer

Data primer pada penelitian ini dibagi menjadi dua jenis data berdasarkan kebutuhan pengolahannya. Data yang digunakan untuk membuat dan menguji model diperoleh dari *website Kaggle* dengan pemilik himpunan data adalah Jonathan Oheix yang terakhir diunggah pada tahun 2019. Serta data yang digunakan untuk pengujian secara langsung diperoleh dengan cara pengolektifan data rekaman dari kegiatan kelas *online* melalui izin

dari pihak Bagian Akademik STIMIK ESQ. Adapun ketentuan dari data rekaman video belajar mengajar adalah sebagai berikut:

- Durasi minimal 30 menit
- Jumlah partisipan minimal 4 orang
- Rekaman yang digunakan baik dari platform Zoom Meeting ataupun Google Meet adalah rekaman dengan tampilan *grid mode*
- Jumlah sampel video sebanyak 12 kelas yang terdiri dari:
 - Kelas Umum (3 sampel)
 - Kelas Program Studi Ilmu Komputer (3 sampel)
 - Kelas Program Studi Sistem Informasi (3 sampel)
 - Kelas Program Studi Manajemen Bisnis (3 sampel)

2) Data Sekunder

Data diperoleh dari berbagai karya ilmiah berupa buku dan literatur yang mendukung teori-teori dan konsep yang digunakan dalam penelitian ini. Data sekunder yang dimanfaatkan dalam penelitian ini antara lain langkah-langkah dalam mendeteksi objek dan evaluasi kinerja model.

3.4 Instrumen Penelitian

Instrumen Penelitian merupakan sebuah *tools* atau alat yang membantu peneliti dalam pengumpulan dan pengolahan data pada penelitian yang dilaksanakannya (Hasibuan, 2007). Adapun instrumen penelitian yang digunakan dalam pengumpulan dan pengolahan data pada penelitian ini antara lain:

- 1) Dokumentasi Foto Ekspresi Wajah
- 2) Dokumentasi Video Kegiatan Kelas *Online*
- 3) Perangkat Lunak
 - Sistem Operasi Windows 10
 - Python
 - Jupyter Notebook Anaconda dan Google Collab
 - Microsoft Office
- 4) Perangkat Keras
 - Laptop Asus VivoBook A442U
 - Processor Intel Core i5 1,8 Ghz

- RAM 16GB
- HDD 1TB dan SSD 240GB

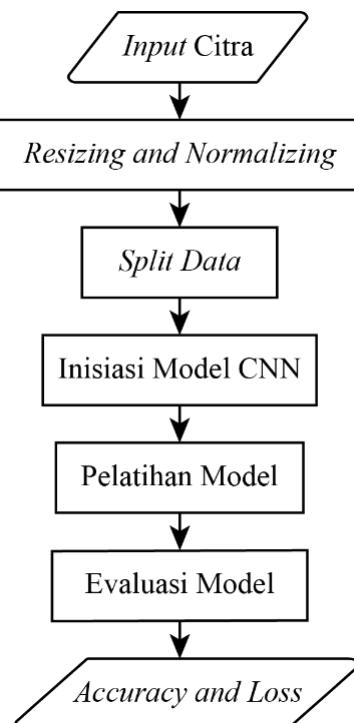
3.5 Metode Analisis, Pengujian dan Interpretasi Data

Pada subbab ini menjelaskan tentang metode analisis, pengujian dan interpretasi data yang digunakan selama penelitian ini. Adapun rincian dari berbagai metode tersebut adalah sebagai berikut:

3.5.1 Analisis Data

Ekspresi wajah yang akan dijadikan kelas yaitu senang, sedih, takut, terkejut, marah, dan menjijikkan. Data dari setiap kelas tersebut akan dilakukan proses normalisasi terlebih dahulu agar ada keseimbangan dari proses klasifikasi. Kemudian data akan dilakukan uji coba dengan menerapkan beberapa parameter terlebih dahulu untuk menguji arsitektur mana yang akan digunakan untuk melakukan identifikasi wajah peserta kelas *online*.

3.5.2 Pengujian Data



Gambar 3.2 Alur Eksperimen Pengujian Data Secara Keseluruhan

Pada dapat kita pada Gambar 3.2 merupakan lihat alur umum dari pengujian data yang akan menentukan arsitektur model yang selanjutnya akan digunakan dalam proses deteksi pada rekaman video secara *real-time*. Data diolah dan diuji menggunakan *tools* Python versi 3.7 dengan Jupyter Notebook Anaconda serta menggunakan *package* atau *library* *Tensorflow*. Adapun tahapan-tahapan dari pengolahan hingga pengujian secara keseluruhan dijabarkan sebagai berikut:

- 1) Membaca Data Citra

Citra yang akan diolah memiliki 7 kelas yang berbeda yang dikelompokkan dan disatukan pada setiap *folder* yang diberi nama sesuai dengan ekspresinya. Data citra tersebut akan dihitung terlebih dahulu dengan memanfaatkan fungsi yang ada pada *package* atau *library* *Tensorflow* yaitu *ImageGenerator*.

- 2) *Preprocessing Data*

Sebelum proses pelatihan dan pengujian data dilakukan, perlu adanya proses *preprocessing* agar pengolahan data yang dilakukan sesuai dan bersih. Pada pemrosesan citra pada data latih, tahap *preprocessing* terdiri dari dua proses tahapan, yaitu proses *resizing* dengan mengubah ukuran citra menjadi 48x48 dan *normalization* data citra dengan membagi pada setiap data citra dengan 255. Proses ini perlu dilakukan sebagai syarat dan ketentuan untuk memudahkan proses identifikasi. Selain itu, pada proses ini juga dilakukan *labeling* berdasarkan folder yang sudah tersusun rapi dan diakses menggunakan *package* atau *library* *Image Generator* dari *Tensorflow* untuk memudahkan proses *preprocessing* dan *labeling* seluruh citra.

- 3) Pembagian Data Latih dan Data Validasi

Untuk kebutuhan evaluasi model yang akan dilatih perlu melakukan pembagian data latih dan data validasi secara acak. Setelah dibagi menjadi dua bagian, kemudian data latih akan diacak kembali urutannya agar pada saat proses pelatihan, model yang akan dibuat terbiasa menerima inputan secara acak. Namun pada data validasi urutan data tidak perlu diacak. Pembagian data latih dan data validasi ini dilakukan satu kali setiap pelatihan dan pengujian seluruh model yang akan dievaluasi.

4) Perancangan model dengan *Tensorflow*

Merancang model *deep learning* menggunakan *package* atau *library* *Tensorflow* dengan menguji evaluasi model terbaik dari beberapa pengujian diantaranya yaitu jumlah lapisan *convolutional layer*, dilanjutkan dengan penggunaan *dropout layer*, dilanjutkan lagi dengan mengubah nilai *learning rate*, dan yang terakhir diuji menggunakan nilai *epochs* yang lebih besar.

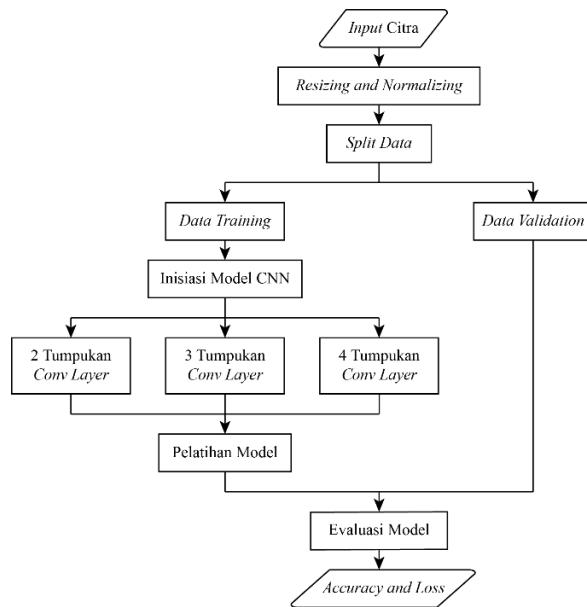
5) Evaluasi performa model dan mengambil model terbaik

Proses ini akan menampilkan diagram dari besarnya nilai akurasi dan loss masing-masing arsitektur model pada setiap *epoch*-nya. Model dengan nilai *validation accuracy* tertinggi dan *validation loss* terendah yang akan dipilih sebagai model terbaik. Model tersebut akan digunakan untuk mendeteksi ekspresi wajah dan juga menghitung total ekspresi wajah yang ditampilkan peserta kelas *online*. Secara otomatis pada setiap model akan selalu menyimpan modelnya pada setiap *epoch* dan hanya akan menyimpan model dengan nilai akurasi dan *loss* terbaik.

6) Visualisasi hasil

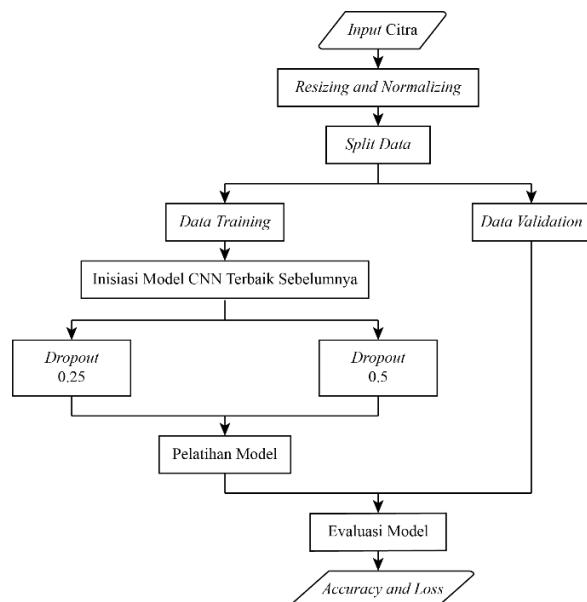
Proses ini merupakan akhir dari siklus satu kali pendekripsi ekspresi wajah dari satu *file* rekaman kegiatan kelas *online*. Proses ini terjadi ketika pemutaran video telah selesai atau *user* sendiri yang menghentikan pemutaran video maka secara otomatis sistem akan mengkalkulasikan seluruh hasil yang dihitung selama video diputar. Hasil tersebut kemudian ditampilkan melalui diagram batang dan lingkaran yang dapat disimpan sebagai gambar.

Perancangan model yang sudah disebutkan sebelumnya akan diuji terlebih dahulu dengan menggunakan 4 parameter yaitu pengaruh penggunaan jumlah tumpukan *convolutional layer*, penggunaan nilai *dropout*, besaran nilai *learning rate*, dan jumlah *epochs*. Pada pengujian model dengan jumlah tumpukan *Convolutional Layer* yang berbeda akan dilakukan sebanyak 3 kali yaitu dengan 2 lapisan, 3 lapisan, dan 4 lapisan *convolutional layer*. Pada pengujian ini akan menentukan berapa tumpukan *convolutional layer* yang terbaik dan akan digunakan dalam pendekripsi melalui rekaman video. Adapun alur dari pengujian model dengan jumlah tumpukan *convolutional layer* dapat dilihat pada Gambar 3.3.



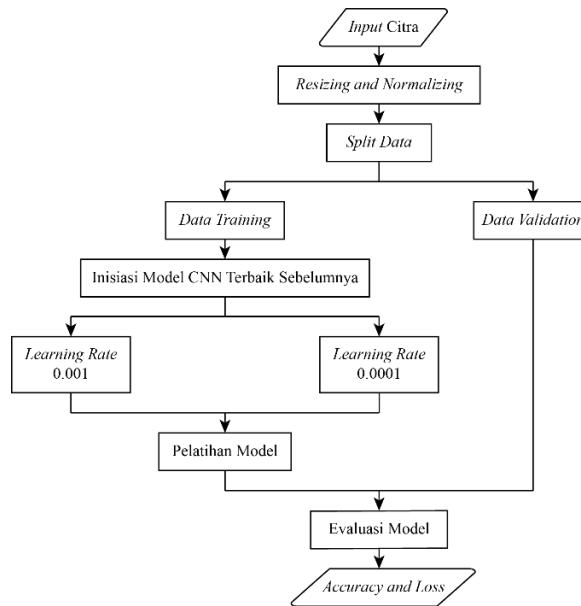
Gambar 3.3 Pengujian Pertama dengan Menganalisis Pengaruh Jumlah Tumpukan Convolutional Layer

Pada pengujian model kali ini akan menguji dengan penggunaan nilai *dropout* sebesar 0,25 dan 0,5 dengan menggunakan arsitektur terbaik pada pengujian sebelumnya. Pada pengujian ini juga akan menentukan berapa nilai *dropout* yang terbaik dan akan digunakan dalam pendekripsi melalui rekaman video. Adapun alur dari pengujian model dengan penggunaan nilai *dropout* yang berbeda dapat dilihat pada Gambar 3.4.



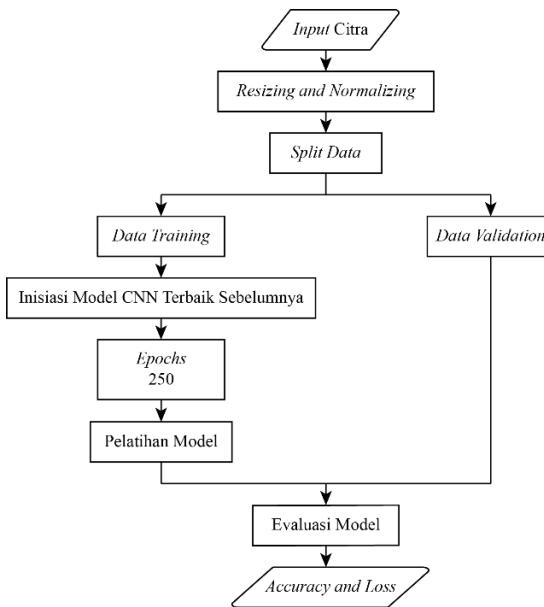
Gambar 3.4 Pengujian Kedua dengan Menganalisis Pengaruh Penggunaan Nilai Dropout

Pada pengujian model selanjutnya yaitu dengan menguji model dengan nilai besaran *learning rate* sebesar 0,001 dan 0,0001 dengan menggunakan arsitektur terbaik pada pengujian tumpukan *convolutional layer* dan penggunaan nilai *dropout*. Pada pengujian ini akan menentukan berapa besaran nilai *learning rate* yang terbaik dan akan digunakan dalam pendekripsi melalui rekaman video. Adapun alur dari pengujian model besaran nilai *learning rate* yang berbeda dapat dilihat pada Gambar 3.5.



Gambar 3.5 Pengujian Ketiga dengan Menganalisis Pengaruh Besaran Nilai *Learning Rate*

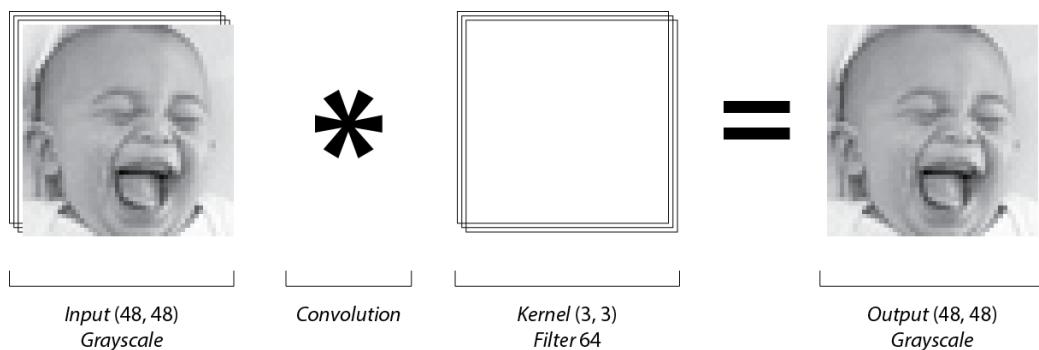
Pada pengujian model yang terakhir ini akan menguji dan mengambil model terbaik dengan *epochs* selama pelatihan model sebanyak 250 kali dengan arsitektur yang sudah diperoleh pada pengujian sebelumnya. Pada pengujian ini juga akan menentukan model mana yang terbaik dan akan digunakan dalam pendekripsi melalui rekaman video berdasarkan jumlah *epochs*. Adapun alur dari pengujian model dengan jumlah *epochs* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Pengujian Keempat dengan Menggunakan *Epochs* yang Lebih Besar yaitu 250

3.5.3 Proses *Convolution*

Berdasarkan penguraian dari arsitektur jaringan *Convolutional Neural Network*, berikut adalah pembahasan secara lengkap dari proses *convolution*.



Gambar 3.7 Visualisasi Proses *Convolution*

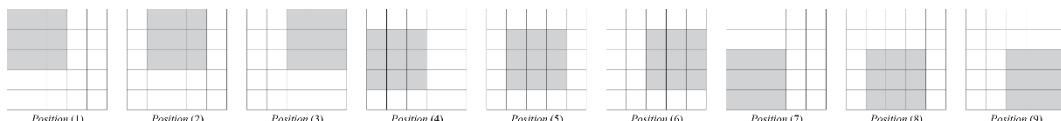
Convolution merupakan proses yang bertujuan menghasilkan satu buah matriks hasil dari kombinasi dua buah matriks *input* dan *kernel*. Pada penelitian ini *input* citra memiliki dimensi 48x48 dengan *color mode grayscale* yang berarti hanya memiliki 1 *channel* gambar. Selanjutnya *input* citra akan dilakukan *convolution* dengan *filter* yang sudah diinisialisasi. *Filter* berfungsi untuk menentukan pola deteksi citra yang akan dilakukan proses *convolution*. Jumlah *filter* yang diinisialisasi pada *convolution* yang pertama sebesar 64 yang akan

menghasilkan 64 *feature map*. Visualisasi proses *convolution* dapat dilihat pada Gambar 3.7.

Agar mudah memahami proses apa yang terjadi pada *convolution* peneliti akan menggunakan *sample input* dari citra yang akan diproses. Namun, karena ukuran citra yang cukup besar yaitu 48x48, maka peneliti hanya akan mengambil sebagian matriks saja untuk menjelaskan perhitungan dari proses *convolution* ini. Adapun perhitungan pada proses *convolution* dapat dilihat pada Gambar 3.8.

Gambar 3.8 Perhitungan Manual Proses *Convolution*

Pada di atas, besar pergeseran atau *stride* yang digunakan pada *kernel* adalah sebesar 1 dengan ukuran 3x3. Artinya pada proses *convolution* akan dihitung berdasarkan jumlah kernel 3x3 dengan pergeseran satu-satu hingga seluruhnya telah selesai buat matriks baru. Berikut adalah visualisasi pergeseran kernel.



Gambar 3.9 Pergeseran *Kernel* dengan 1 *Stride*

Pada Gambar 3.9 menunjukkan perhitungan *convolution* pada setiap posisi dengan ukuran *kernel* 3x3 dan *padding same*. Adapun masing-masing perhitungan *convolution* dijabarkan sebagai berikut:

$$\begin{aligned} \text{Position (1)} &= (7 \times 1) + (7 \times (-1)) + (5 \times 1) + (3 \times (-1)) + (4 \times 1) + (2 \times (-1)) + \\ &\quad (8 \times 1) + (6 \times (-1)) + (7 \times (-1)) = -1 \end{aligned}$$

$$\begin{aligned} \text{Position (2)} &= (3 \times 1) + (4 \times (-1)) + (2 \times 1) + (8 \times (-1)) + (6 \times 1) + (7 \times (-1)) + \\ &\quad (2 \times 1) + (1 \times (-1)) + (6 \times (-1)) = -13 \end{aligned}$$

$$\begin{aligned}Position (3) &= (8 \times 1) + (6 \times (-1)) + (7 \times 1) + (2 \times (-1)) + (1 \times 1) + (6 \times (-1)) + \\&\quad (5 \times 1) + (7 \times (-1)) + (4 \times (-1)) = -4\end{aligned}$$

$$\begin{aligned}Position (4) &= (7 \times 1) + (5 \times (-1)) + (8 \times 1) + (4 \times (-1)) + (2 \times 1) + (9 \times (-1)) + \\&\quad (6 \times 1) + (7 \times (-1)) + (1 \times (-1)) = -3\end{aligned}$$

$$\begin{aligned}Position (5) &= (4 \times 1) + (2 \times (-1)) + (9 \times 1) + (6 \times (-1)) + (7 \times 1) + (1 \times (-1)) + \\&\quad (1 \times 1) + (6 \times (-1)) + (3 \times (-1)) = 3\end{aligned}$$

$$\begin{aligned}Position (6) &= (6 \times 1) + (7 \times (-1)) + (1 \times 1) + (1 \times (-1)) + (6 \times 1) + (3 \times (-1)) + \\&\quad (7 \times 1) + (4 \times (-1)) + (2 \times (-1)) = 3\end{aligned}$$

$$\begin{aligned}Position (7) &= (5 \times 1) + (8 \times (-1)) + (9 \times 1) + (2 \times (-1)) + (9 \times 1) + (4 \times (-1)) + \\&\quad (7 \times 1) + (1 \times (-1)) + (7 \times (-1)) = 8\end{aligned}$$

$$\begin{aligned}Position (8) &= (2 \times 1) + (9 \times (-1)) + (4 \times 1) + (7 \times (-1)) + (1 \times 1) + (7 \times (-1)) + \\&\quad (6 \times 1) + (3 \times (-1)) + (3 \times (-1)) = -16\end{aligned}$$

$$\begin{aligned}Position (9) &= (7 \times 1) + (1 \times (-1)) + (7 \times 1) + (6 \times (-1)) + (3 \times 1) + (3 \times (-1)) + \\&\quad (4 \times 1) + (2 \times (-1)) + (8 \times (-1)) = 1\end{aligned}$$

Sebelum berlanjut ke proses berikutnya, seluruh hasil dari perhitungan *convolution* akan ditambahkan proses *activation* dengan *ReLU* yang mengubah nilai negatif menjadi 0 dan yang lainnya sampai *infinity*. *Output* dari proses ini memiliki ukuran yang sama yaitu 48x48.

3.5.4 Proses *Pooling*

Proses *pooling* merupakan proses pengurangan dimensi dari matriks citra dengan proses penggabungan. Pada penelitian ini fungsi *pooling* yang diterapkan adalah *max-pooling* yang memiliki hasil lebih baik dibandingkan metode atau fungsi lainnya. Proses *pooling* pada model ini menggunakan *kernel* sebesar 2x2. Adapun visualisasi dari proses *pooling* dapat dilihat pada Gambar 3.10.

1	13	4
3	3	3
8	16	1

 $=$

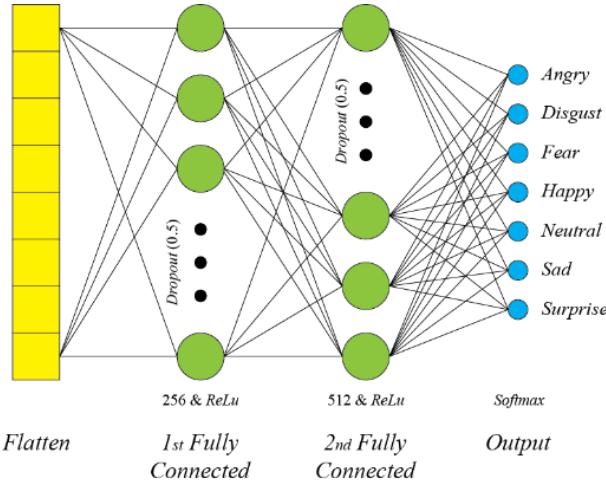
13	13
16	16

Gambar 3.10 Visualisasi Proses *Pooling* dengan *Max-Pooling*

Proses *pooling* dilakukan dengan cara mengambil nilai maksimum dari setiap matriks berukuran 2×2 sesuai *kernel size* dari *pooling layer*. Pada Gambar 3.10 *output* dari proses *pooling* ini berupa matriks yang berisi nilai maksimum yang diperoleh dari matriks *feature map* hasil dari proses *convolution*. Pada akhir proses *pooling* ini juga diterapkan proses *dropout* sebesar 0,5 yang bertujuan menghilangkan sebagian *neuron* agar model tidak menjadi *overfitting*. Proses *convolution* dan *pooling* ini akan dilakukan sebanyak 4 kali karena pada arsitektur model terdapat 4 lapisan *convolution* sebelum masuk kepada proses *classification*.

3.5.5 Proses *Classification* (*Flatten*, *Fully Connected*, dan *Output*)

Proses *fully connected* perlu merupakan proses *classification* yang menerima masukan berupa *vector*. Hasil dari proses *feature learning* akan terlebih dahulu dijadikan *vector* dengan *flatten layer*. Proses *fully connected* ini memiliki 2 *hidden layer* yang masing-masingnya memiliki *neuron* sebanyak 256 dan 512 dengan menggunakan fungsi *activation ReLu*. Pada setiap akhir proses *fully connected* ini juga diterapkan proses *dropout* sebesar 0,5. Setelah proses *fully connected* selesai akan baru akan dihasilkan *output* yang memiliki 7 *neuron* berdasarkan banyaknya jumlah kelas yaitu *angry*, *disgust*, *fear*, *happy*, *neutral*, *sad*, dan *surprise* menggunakan fungsi *activation softmax*. Visualisasi proses *classification* dapat dilihat pada Gambar 3.11.

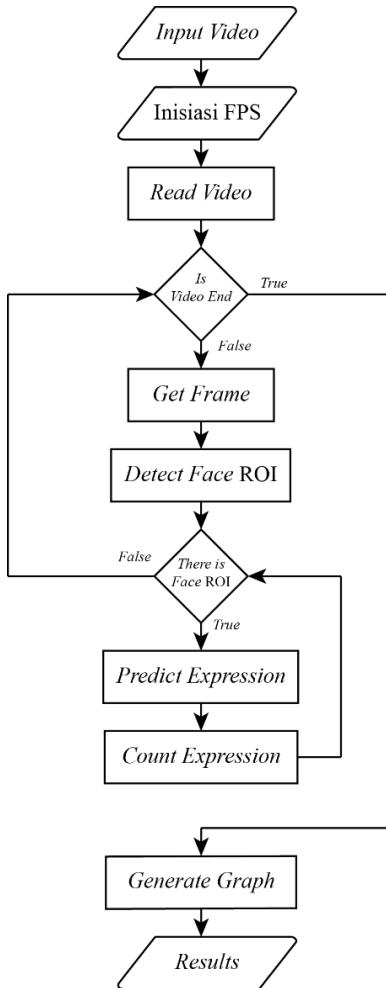


Gambar 3.11 Visualisasi Proses *Flatten*, *Fully Connected*, dan *Output Classification*

3.5.6 Proses Pengujian Melalui Tampilan GUI

Proses ini sebagai sarana yang dapat mempermudah user untuk melakukan interaksi secara langsung dengan sistem. Pembuatan GUI ini memanfaatkan package atau library PySimpleGUI dengan banyak sekali fitur yang dapat diterapkan dengan mudah untuk membuat perancangan GUI. Pengujian model menggunakan tampilan GUI terhadap rekaman video kegiatan kelas *online* melalui platform *online meeting*. Masing-masing video yang sebelumnya berukuran panjang dipotong dan hanya akan digunakan ketika video menampilkan *grid mode*. Masing-masing video tersebut berukuran 120 detik yang kemudian akan diuji dengan model yang sudah dibuat dengan ketentuan FPS yang diinisiasi oleh *user*.

Secara gambaran umum, *user* akan memilih *file* rekaman video kegiatan kelas *online* yang akan dideteksi serta melakukan input besaran nilai *Frame Per Second* (FPS) yang akan diterapkan selama pendekripsi berlangsung. Video akan diputar dengan kecepatan sesuai FPS yang dipilih sebelumnya hingga selesai. Selama pemutaran video, secara otomatis sistem akan mendekripsi ekspresi wajah dengan menandakan wajah tersebut dengan *box* beserta nama ekspresi yang telah diprediksi. Selain itu sistem juga melakukan perhitungan pada setiap kategori ekspresi seberapa banyak ekspresi tersebut muncul. Adapun alur proses pendekripsi ekspresi wajah melalui video yang dilakukan melalui interaksi GUI dapat dilihat pada Gambar 3.12.



Gambar 3.12 Alur Pengujian Video *Real-Time* Melalui Interaksi GUI

3.5.7 Interpretasi Data

Pada tahapan interpretasi data ini, peneliti terlebih dahulu mempelajari hasil yang diperoleh dari pengujian data. Hasil tersebut dikaitkan dengan analisis dan teori dari studi literatur sebelumnya. Selain itu, perlu juga pengujian berkala untuk memastikan bahwa hasil yang diperoleh sudah benar-benar valid. Kemudian hasil tersebut juga dikaitkan dengan penelitian terdahulu guna membandingkan hasil yang diperoleh dengan hasil dari penelitian terdahulu. Variabel yang diambil untuk menentukan model mana yang terbaik adalah dengan mempertimbangkan besaran nilai *accuracy* dan *loss* pada tiap-tiap model. Kemudian peneliti akan menuliskan kesimpulan dan saran dari interpretasi hasil pengujian data yang dilakukan dan hasil dari penerapan langsung dengan video kegiatan kelas *online*.

BAB 4

ANALISIS DAN PEMBAHASAN

Pada bab berisi kajian-kajian analisis dan pembahasan yang telah dilakukan selama proses penelitian.

4.1 Analysis

Pada subbab ini menjelaskan keseluruhan proses analisis yang sudah dilakukan selama proses penelitian. Adapun rincian dari hasil analisis tersebut adalah sebagai berikut:

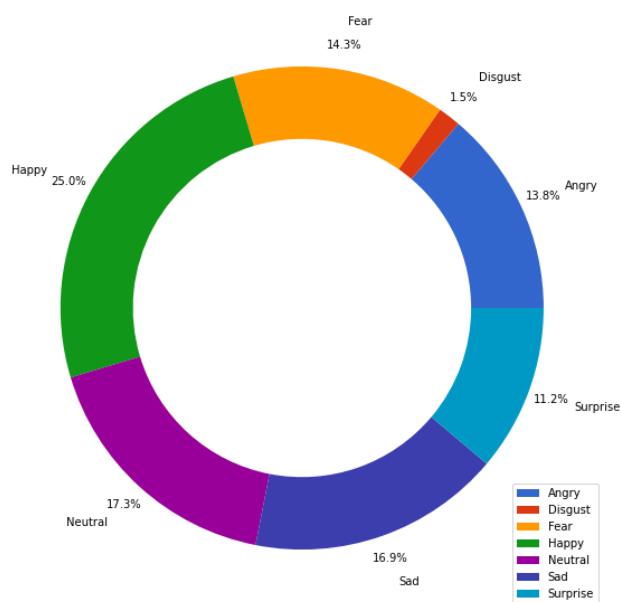
4.1.1 Analisis Dataset

Pada penelitian ini *dataset* yang digunakan berasal dari *website* Kaggle milik Jonathan Oheix yang terakhir diunggah pada tahun 2019. Adapun karakteristik yang terdapat pada *dataset* tersebut terdiri dari 7 macam ekspresi yang terdiri dari *angry* (marah), *disgust* (menjijikkan), *fear* (takut), *happy* (senang), *neutral* (netral), *sad* (sedih), dan *surprise* (terkejut). *Dataset* tersebut dapat diunduh pada *link* <https://www.kaggle.com/jonathanoheix/face-expression-recognition-dataset>. Rincian frekuensi pada *dataset* ekspresi wajah dapat dilihat pada Tabel 4.1 dan visualisasi frekuensi diagram *donut* dalam bentuk persentase dapat dilihat pada Gambar 4.1.

Tabel 4.1 Frekuensi Dataset Train dan Validation

Sumber: website Kaggle, Jonathan Oheix (2019) telah diolah kembali

No	Dataset	Frekuensi
1	<i>Angry</i>	4953
2	<i>Disgust</i>	547
3	<i>Fear</i>	5121
4	<i>Happy</i>	8989
5	<i>Neutral</i>	6189
6	<i>Sad</i>	6077
7	<i>Surprise</i>	4002



Gambar 4.1 Visualisasi Frekuensi Diagram Donut Dalam Bentuk Persentase

Sumber: website Kaggle, Jonathan Oheix (2019) telah diolah kembali

Selain itu, pada data video yang diperoleh dari bagian akademik STIMIK ESQ berdasarkan kriteria yang sudah ditentukan terdapat beberapa rekaman video yang tidak sesuai dengan kriteria. Dari 12 data video yang diajukan tersebut hanya 2 video yang masuk kriteria. Adapun rincian dari hasil video yang diperoleh dapat dilihat pada Tabel 4.2.

Tabel 4.2 Rincian Video yang Diperoleh dari Bagian Akademik STIMIK ESQ

No	Video	Durasi > 30 menit	Partisipan > 4 Orang	Grid Mode	Keterangan
1	Kelas Umum (1)	Ya	Ya	Ya	Digunakan
2	Kelas Umum (2)	Ya	Ya	Ya	Digunakan
3	Kelas Umum (3)	Ya	Ya	Tidak	Tidak Digunakan
4	Kelas Program Studi Ilmu Komputer (1)	Ya	Ya	Tidak	Tidak Digunakan
5	Kelas Program Studi Ilmu Komputer (2)	Ya	Ya	Tidak	Tidak Digunakan
6	Kelas Program Studi Ilmu Komputer (3)	Ya	Ya	Tidak	Tidak Digunakan
7	Kelas Program Studi Sistem Informasi (1)	Ya	Ya	Tidak	Tidak Digunakan
8	Kelas Program Studi Sistem Informasi (2)	Ya	Ya	Tidak	Tidak Digunakan
9	Kelas Program Studi Sistem Informasi (3)	Ya	Ya	Tidak	Tidak Digunakan

No	Video	Durasi > 30 menit	Partisipan > 4 Orang	Grid Mode	Keterangan
10	Kelas Program Studi Manajemen Bisnis (1)	Ya	Ya	Tidak	Tidak Digunakan
11	Kelas Program Studi Manajemen Bisnis (2)	Ya	Ya	Tidak	Tidak Digunakan
12	Kelas Program Studi Manajemen Bisnis (3)	Ya	Ya	Tidak	Tidak Digunakan

Dari 2 video yang sesuai dengan kriteria yang sudah ditentukan maka akan dipotong menjadi 5 video yang berdurasi 120 detik yang akan digunakan dalam pengujian secara langsung. Pemilihan momen video yang akan dipotong adalah ketika dosen sedang tidak menampilkan slide presentasi, kemudian ketika keseluruhan peserta terlihat pada saat *grid mode*, dan peserta yang menghidupkan fitur kameranya melebihi 50%. Hal ini dilakukan guna mendapatkan hasil yang maksimal dalam pengujian secara langsung menggunakan video *real-time*.

4.1.2 Persiapan Tools dan Package

Penelitian ini membutuhkan beberapa persiapan *tools* dan *package* untuk dapat menjalankan program agar berjalan dengan baik dan benar. Selain itu, persiapan ini juga menghindari adanya *error* atau *bug* tertentu yang akan muncul sewaktu-waktu jika tidak sesuai dengan persyaratan sistem. Adapun beberapa persiapan yang dibutuhkan antara lain:

- 1) Anaconda

Anaconda merupakan sebuah aplikasi yang merupakan salah satu di antara beberapa cara agar dapat mengoperasikan *python* pada suatu komputer. Anaconda adalah aplikasi *open source* yang dapat menjalankan aplikasi standar yang terdapat pada Anaconda Individual Edition. Namun, untuk edisi yang lain seperti *Commercial*, *Team*, *Enterprise*, dan *Professional Edition* terdapat *premium package* yang membutuhkan *subscribe* tiap periodenya. Website Anaconda *Individual Edition* dapat dilihat pada Gambar 4.2 dan untuk aplikasinya sendiri dapat diunduh pada *link* <https://www.anaconda.com/products/individual>.



Individual Edition

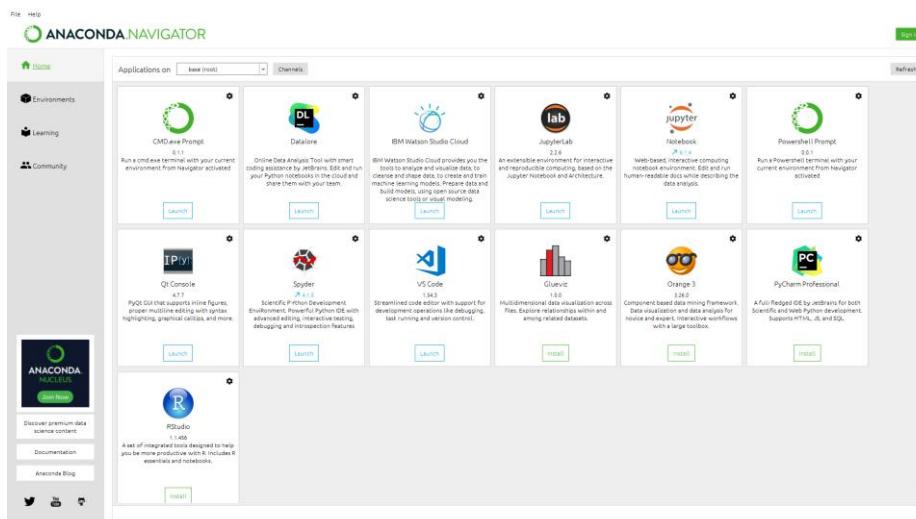
Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

[Download](#)

Gambar 4.2 Website Anaconda *Individual Edition*

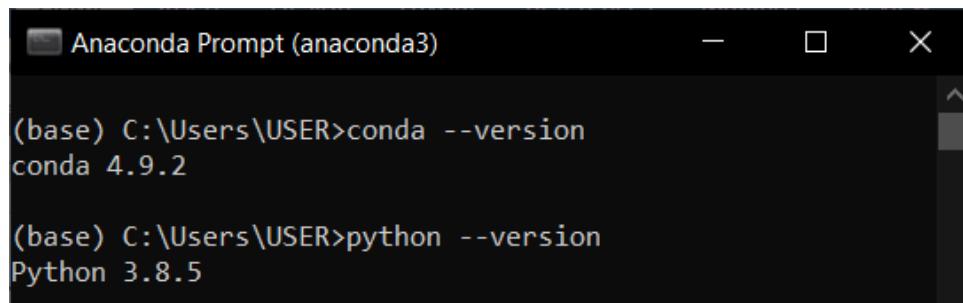
Setelah selesai mengunduh Anaconda *installer* yang sesuai dengan operasi sistem pada komputer yang digunakan, selanjutnya adalah proses instalasi. Dalam proses instalasi, kita tidak perlu melakukan konfigurasi tambahan apapun karena pada dasarnya instalasi Anaconda sudah melakukan konfigurasi standarnya. Namun, jika kita ingin menambahkan *custom configuration* maka kita perlu sedikit mengubah parameter saat melakukan instalasi Anaconda. Aplikasi yang terdapat pada Anaconda Navigator cukup beragam seperti Spyder, Jupyter Notebook, VS Code dan lain sebagainya yang dapat digunakan dalam pengolahan *python*. Adapun tampilan *home screen* Anaconda Navigator dapat dilihat pada Gambar 4.3.



Gambar 4.3 Homescreen Anaconda Navigator

2) Python

Setelah sebelumnya kita melakukan instalasi Anaconda *Individual Edition*, maka sekarang kita perlu mengecek versi dari Anaconda dan *python* yang terpasang pada komputer. Untuk melakukan pengecekan versi Anaconda dan *python* kita perlu membuka Anaconda Prompt atau CMD pada umumnya kemudian memasukkan kode seperti pada Gambar 4.4.



```
Anaconda Prompt (anaconda3)
-
(base) C:\Users\USER>conda --version
conda 4.9.2

(base) C:\Users\USER>python --version
Python 3.8.5
```

Gambar 4.4 Mengecek versi Conda dan Python Melalui Anaconda Prompt

3) Numpy

Numpy merupakan sebuah *package* atau *library open source* pada *python* yang mampu menangani *array*, *vector*, dan *matrix* dengan dimensi yang sangat besar. Struktur data yang digunakan Numpy mirip dengan *list* pada *python*, namun perbedaannya adalah membutuhkan memori yang lebih kecil dan memiliki performa yang cepat. Untuk proses instalasi Numpy dapat melihat dokumentasi PyPi pada link <https://pypi.org/> atau dapat dilakukan dengan melalui Anaconda Prompt dengan memasukkan kode seperti pada Gambar 4.5.



Gambar 4.5 Instalasi Package atau Library Numpy Menggunakan Pip

4) Pandas

Pandas merupakan sebuah *package* atau *library open source* pada *python* yang menangani permasalahan *structuring data* dan analisis data terhadap suatu *dataset* baik yang berukuran kecil maupun besar. Pandas mampu melakukan visualisasi data menjadi sebuah tabel, melakukan deskripsi

statistik, mengubah dimensi data, dan lain sebagainya. Untuk proses instalasi Pandas dapat dilakukan melalui Anaconda Prompt dengan memasukkan kode seperti pada Gambar 4.6.



Gambar 4.6 Instalasi *Package* atau *Library* Pandas Menggunakan Pip

5) Matplotlib

Matplotlib merupakan sebuah *package* atau *library open source* pada *python* yang spektakuler dalam menangani visualisasi data seperti visualisasi data 2D atau 3D maupun membuat diagram. Matplotlib dibangun melalui pemanfaatan *array* Numpy. Di antara fungsi pembangunan visualisasi diagram yang disediakan oleh Matplotlib adalah *bar chart*, *pie chart*, *histogram*, *scatter plot*, *polar plot*, dan lain sebagainya. Untuk proses instalasi Matplotlib dapat dilakukan melalui Anaconda Prompt dengan memasukkan kode seperti pada Gambar 4.7.



Gambar 4.7 Instalasi *Package* atau *Library* Matplotlib Menggunakan Pip

6) OpenCV

OpenCV merupakan sebuah *package* atau *library open source* pada *python* yang dibangun sedemikian rupa khusus agar mampu menangani permasalahan *Computer Vision*. OpenCV mampu mengolah data gambar dan video menjadi sebuah informasi yang dimuat dalam *array* yang berisi angka-angka. OpenCV mampu memanipulasi data citra seperti *resizing*, *cropping*, *converting*, dan lain sebagainya. Untuk proses instalasi OpenCV dapat dilakukan melalui Anaconda Prompt dengan memasukkan kode seperti pada Gambar 4.8.



Gambar 4.8 Instalasi Package atau Library Opencv Menggunakan Pip

7) *Tensorflow*

Tensorflow merupakan *package* atau *library python* yang menjadi kunci utama dari sistem ini. *Tensorflow* menyediakan fungsi-fungsi alur kerja *neural network* untuk membuat sebuah model. Selain itu *Tensorflow* juga dibangun agar *machine learning* dapat diimplementasikan pada seluruh platform aplikasi. Untuk proses instalasi *Tensorflow* dapat dilakukan melalui Anaconda Prompt dengan memasukkan kode seperti pada Gambar 4.9.



Gambar 4.9 Instalasi Package atau Library Tensorflow Menggunakan Pip

8) *Keras*

Keras merupakan *package* atau *library python* yang sama pentingnya seperti *Tensorflow*. Sama seperti *Tensorflow*, *Keras* juga menyediakan fungsi-fungsi yang mudah digunakan untuk pembangunan model *deep learning*. Untuk proses instalasi *Keras* dapat dilakukan melalui Anaconda Prompt dengan memasukkan kode seperti pada Gambar 4.10.



Gambar 4.10 Instalasi Package atau Library Keras Menggunakan Pip

9) *PySimpleGUI*

PySimpleGUI merupakan *package* atau *library* pada *python* yang dapat digunakan untuk membuat *Graphic User Interface* (GUI) melalui kode

python. PySimpleGUI memiliki fungsi yang dapat membuat *field* seperti layaknya *form input*, *button*, *dropdown*, dan lain sebagainya. Untuk proses instalasi PySimpleGUI dapat dilakukan melalui Anaconda Prompt dengan memasukkan kode seperti pada Gambar 4.11.



Gambar 4.11 Instalasi *Package* atau *Library* PySimpleGUI Menggunakan Pip

4.1.3 *Preprocessing* dan Pembagian *Dataset*

Seluruh data citra akan dilakukan proses *preprocessing* terlebih dahulu sebelum diolah dan dibagi menjadi data *training* dan *validation*. Adapun citra yang menjadi bahan *dataset* disama ratakan ukurannya menjadi 48x48 dan diubah menjadi *grayscale* yang sebelumnya memiliki warna RGB. Setelah itu, seluruh citra akan dilakukan *normalization* dengan membagi seluruh pikselnya dengan 255 agar memiliki *range* yang sama antara satu piksel dengan piksel lainnya. Seluruh data akan dimuat dengan melabelkan secara otomatis melalui ImageGenerator pada *Tensorflow* dan membaginya menjadi 90% data *training* yaitu sebanyak 32302 citra dan 10% data *validation* sebanyak 3585 citra.

4.1.4 Inisialisasi Arsitektur Model

Model yang digunakan untuk pelatihan data *training* dan *validation* terbagi menjadi 5 bagian grup antara lain *input layer*, *convolution layer*, *flattening*, *fully connected layer*, dan *output layer*. *Input layer* memiliki *input shape* sebesar (48, 48) dengan *color mode grayscale*. Pada *convolution layer* terdiri dari *convolution* 2D, *batch normalization*, *activation ReLu*, *max pooling* 2D, dan ditutup dengan *dropout*. Setelah seluruh proses *convolution layer* dilakukan kemudian akan dijadikan satu baris melalui *flatten layer*.

Setelah seluruh *input shape* dijadikan satu baris, maka akan dilakukan *fully connected* melalui *fully connected layer* yang terdiri dari *dense*, *batch normalization*, *activation ReLu*, dan ditutup dengan *dropout*. Setelah seluruh proses *fully connected layer* dilalui maka arsitektur model ditutup dengan *output layer*

berupa *dense* dengan *activation softmax* sebesar 7 sesuai dengan jumlah kelas ekspresi wajah merupakan rangkuman dari *input shape*, *total hidden layer*, dan *output shape* pada model yang akan dianalisis dan diuji. Adapun rincian inisiasi arsitektur model dapat dilihat pada Tabel 4.3.

Tabel 4.3 Inisialisasi Arsitektur Model

No	Layer Name	Layer Type	Output Shape	Group
1	conv2d	Conv2D	(None, 48, 48, 64)	<i>Input Layer and 1st Convolution Layer</i>
2	batch_normalization	BatchNormalization	(None, 48, 48, 64)	
3	activation	Activation	(None, 48, 48, 64)	
4	max_pooling2d	MaxPooling2D	(None, 24, 24, 64)	
5	dropout	Dropout	(None, 24, 24, 64)	
6	conv2d_1	Conv2D	(None, 24, 24, 128)	<i>2nd Convolution Layer</i>
7	batch_normalization_1	BatchNormalization	(None, 24, 24, 128)	
8	activation_1	Activation	(None, 24, 24, 128)	
9	max_pooling2d_1	MaxPooling2D	(None, 12, 12, 128)	
10	dropout_1	Dropout	(None, 12, 12, 128)	
11	flatten	Flatten	(None, 18432)	<i>Flattening</i>
12	dense	Dense	(None, 256)	<i>1st Fully Connected Layer</i>
13	batch_normalization_2	BatchNormalization	(None, 256)	
14	activation_2	Activation	(None, 256)	
15	dropout_2	Dropout	(None, 256)	
16	dense_1	Dense	(None, 512)	<i>2nd Fully Connected Layer</i>
17	batch_normalization_3	BatchNormalization	(None, 512)	
18	activation_3	Activation	(None, 512)	
19	dropout_3	Dropout	(None, 512)	
20	dense_2	Dense	(None, 7)	<i>Output Layer</i>

4.2 Pembahasan

Pada subbab ini menjelaskan keseluruhan pembahasan dari analisis dan pengujian yang telah dilakukan selama proses penelitian. Adapun rincian dari pembahasan tersebut adalah sebagai berikut:

4.2.1 Hasil Pengujian Model

Dalam proses pengujian model pada penelitian ini, peneliti berfokus kepada tumpukan *convolution layer*, nilai *dropout*, nilai *learning rate*, dan *epochs* dengan menggunakan *optimizer compiler adam*. Adapun ringkasan analisis pengujian model adalah sebagai berikut:

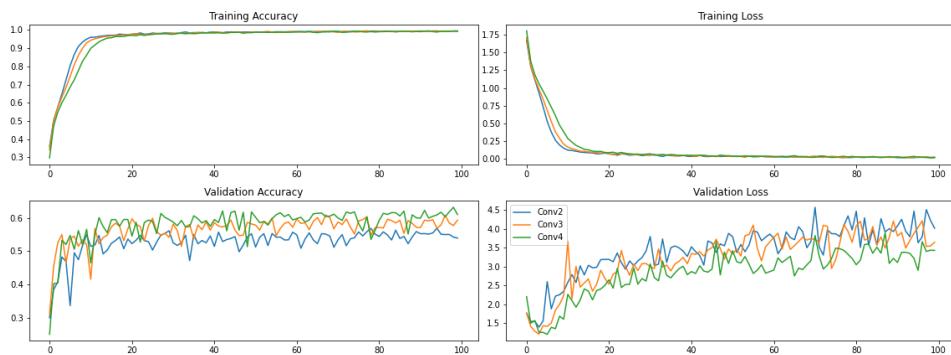
1) Pengaruh jumlah lapisan *convolution layer*

Pada pengujian ini dilakukan sebanyak 3 kali pengujian di mana pada masing-masing pengujian akan ditambahkan tumpukan *convolution layer*

pada arsitektur jaringan. Hasil dari pengaruh jumlah lapisan *convolution layer* dapat dilihat pada Tabel 4.4 dan visualisasi lengkap berupa grafik perbandingan dapat dilihat pada Gambar 4.12.

Tabel 4.4 Evaluasi Pengaruh Jumlah Lapisan *Convolution Layer*

Tumpukan <i>Convolution Layer</i>	<i>Evaluation</i>			
	<i>Training Accuracy</i>	<i>Training Loss</i>	<i>Validation Accuracy</i>	<i>Validation Loss</i>
2	0,9947	0,0141	0,5689	4,468
3	0,9951	0,0120	0,6088	3,7782
4	0,9948	0,0128	0,6317	3,4322



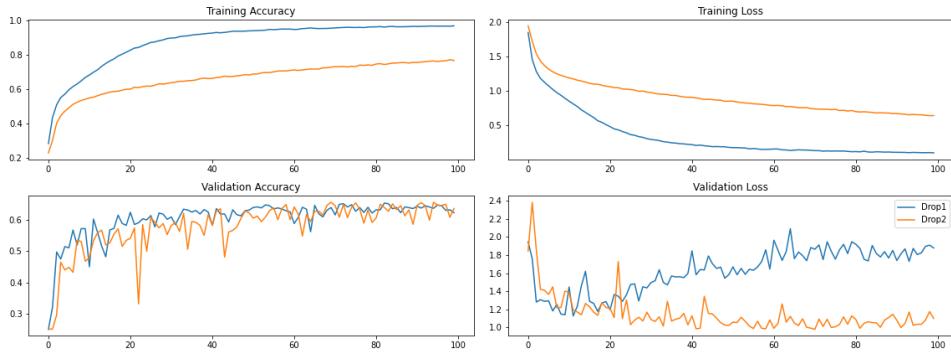
Gambar 4.12 Visualisasi Perbandingan Evaluasi Pengaruh Jumlah Lapisan *Convolution Layer*

2) Pengaruh penggunaan nilai *dropout*

Pada pengujian ini dilakukan sebanyak 2 kali pengujian di mana pada masing-masing pengujian akan ditambahkan nilai *dropout* pada arsitektur jaringan sebesar 0,25 dan 0,5. Hasil dari pengaruh nilai *dropout* dapat dilihat pada Tabel 4.5 dan visualisasi lengkap berupa grafik perbandingan dapat dilihat pada Gambar 4.13.

Tabel 4.5 Evaluasi Pengaruh Nilai *Dropout*

Nilai <i>Dropout</i>	<i>Evaluation</i>			
	<i>Training Accuracy</i>	<i>Training Loss</i>	<i>Validation Accuracy</i>	<i>Validation Loss</i>
0,25	0,9667	0,0968	0,6535	1,7545
0,5	0,7698	0,6404	0,6560	0,9937



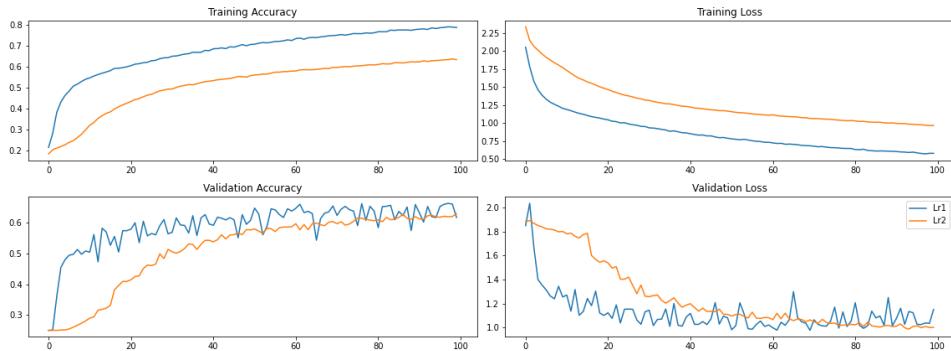
Gambar 4.13 Visualisasi Perbandingan Evaluasi Pengaruh Nilai *Dropout*

3) Pengaruh besar nilai *learning rate*

Pada pengujian ini dilakukan sebanyak 2 kali pengujian di mana pada masing-masing pengujian akan digunakan nilai *learning rate* yang berbeda pada arsitektur jaringan sebesar 0,001 dan 0,0001. Hasil dari pengaruh nilai *learning rate* dapat dilihat pada Tabel 4.6 dan visualisasi lengkap berupa grafik perbandingan dapat dilihat pada Gambar 4.14.

Tabel 4.6 Evaluasi Pengaruh Nilai *Learning Rate*

Nilai <i>Learning Rate</i>	Evaluation			
	<i>Training Accuracy</i>	<i>Training Loss</i>	<i>Validation Accuracy</i>	<i>Validation Loss</i>
0,001	0,7902	0,5705	0,6638	1,0387
0,0001	0,6374	0,9654	0,6303	1,0021



Gambar 4.14 Visualisasi Perbandingan Evaluasi Pengaruh Nilai *Learning Rate*

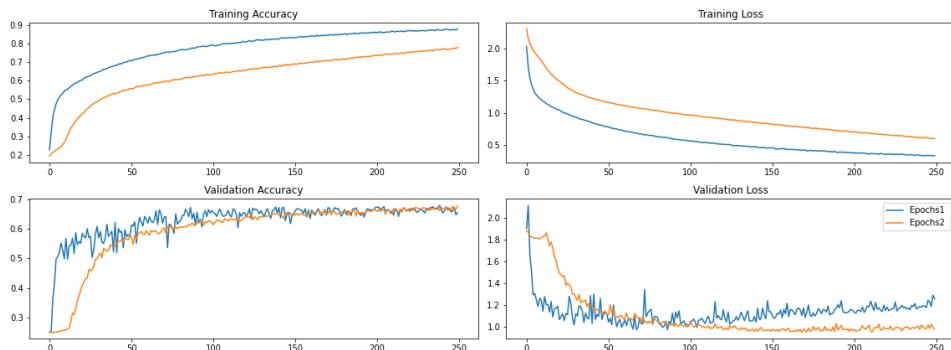
4) Pengaruh jumlah *epochs*

Pada pengujian ini dilakukan sebanyak 2 kali pengujian di mana pada masing-masing pengujian akan dilakukan sebanyak 250 *epochs*. Hasil dari pengaruh banyaknya jumlah *epochs* dapat dilihat pada Tabel 4.7 dan

visualisasi lengkap berupa grafik perbandingan dapat dilihat pada Gambar 4.15.

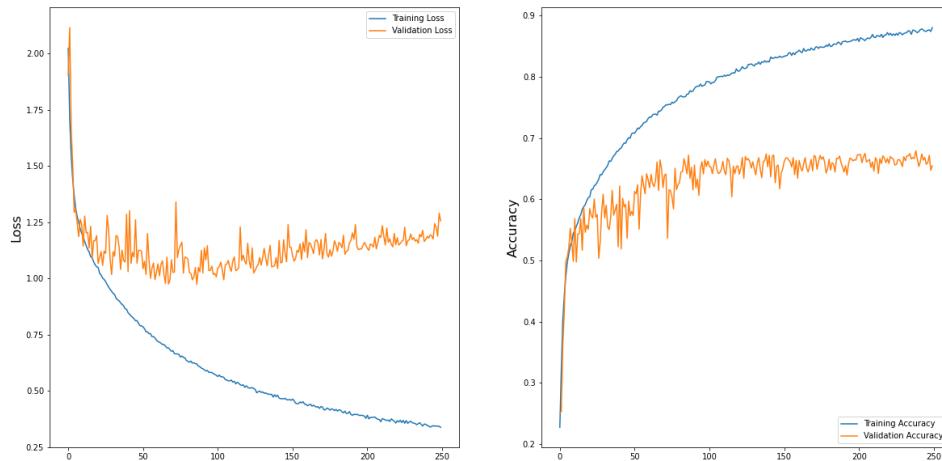
Tabel 4.7 Evaluasi Pengaruh Jumlah Epochs

Jumlah Epochs	Learning Rate 0,001				Learning Rate 0,0001			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
25	0,6239	0,9958	0,6004	1,0582	0,4547	1,4070	0,4621	1,3787
50	0,7084	0,7895	0,6217	1,0302	0,5583	1,1646	0,5751	1,1145
75	0,7551	0,6645	0,6643	0,9752	0,5993	1,0669	0,6127	1,0314
100	0,7923	0,5698	0,6722	0,9737	0,6344	0,9714	0,6278	1,0101
125	0,8167	0,5129	0,6722	0,9737	0,6636	0,8938	0,6476	0,9708
150	0,8338	0,4599	0,6741	1,0568	0,6918	0,8314	0,6537	0,9606
175	0,8492	0,7118	0,6741	1,0568	0,7176	0,7591	0,6632	0,9492
200	0,8627	0,3780	0,6741	1,0568	0,7383	0,7058	0,6663	0,9814
225	0,8723	0,3568	0,6749	1,1391	0,7578	0,6509	0,6719	0,9747
250	0,8805	0,3379	0,6794	1,1646	0,7802	0,6021	0,6772	0,9832



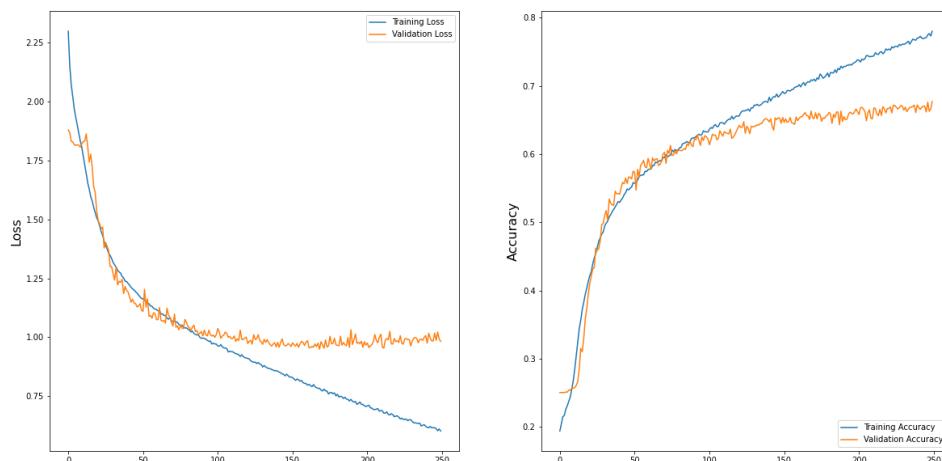
Gambar 4.15 Visualisasi Perbandingan Evaluasi Pengaruh Jumlah Epochs

Pada pengujian terakhir yaitu pengaruh jumlah *epochs* pada saat pelatihan data menunjukkan yang hampir sama rata, di mana hasil evaluasi *validation accuracy* terbaik pada *learning rate* 0,001 adalah sebesar 0,6794 dan pada *learning rate* 0,0001 adalah sebesar 0,6772 yang perbedaannya hanya sebesar 0,0022. Sedangkan hasil evaluasi *validation loss* terbaik pada *learning rate* 0,001 adalah sebesar 1,1646 dan pada *learning rate* 0,0001 adalah sebesar 0,9832 yang perbedaannya juga tidak terlalu jauh yaitu sebesar 0,1814. Namun perlu adanya analisis tambahan di mana perlu membandingkan antara evaluasi *training* dan *validation* untuk melihat apakah ada jarak yang menyebabkan model tersebut dikatakan *overfitting*.



Gambar 4.16 Analisis antara Evaluasi *Training* dan *Validation* pada *Learning Rate* Sebesar **0,001**

Pada Gambar 4.16 dapat dilihat perbandingan antara *training loss* dan *validation loss* yang cukup jauh. Selain itu, perbandingan antara *training accuracy* dan *validation accuracy* yang perbedaannya sebesar 0,2. Hal ini menyebabkan model dengan *learning rate* sebesar 0,001 yang dilatih sebanyak 250 kali *epochs* menjadi model yang *overfitting*.



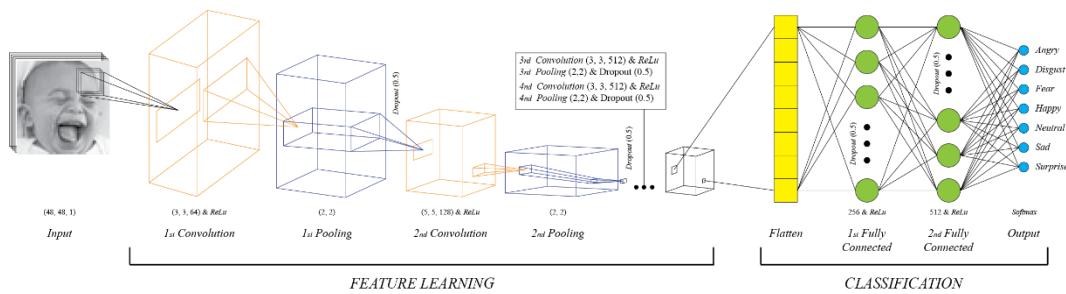
Gambar 4.17 Analisis antara Evaluasi *Training* dan *Validation* pada *Learning Rate* Sebesar **0,0001**

Sedangkan pada evaluasi *training* dan *validation* pada *learning rate* sebesar 0,0001 yang dapat dilihat pada Gambar 4.17 menunjukkan perbandingan *training loss* dan *validation loss* yang jauh lebih baik dibandingkan pengujian sebelumnya. Selain itu, perbandingan antara *training accuracy* dan *validation accuracy* yang

perbedaannya juga jauh lebih baik yaitu sebesar 0,11. Hal ini membuktikan bahwa model dengan *learning rate* sebesar 0,0001 yang dilatih sebanyak 250 kali *epochs* lebih baik dibandingkan pengujian sebelumnya walaupun masih ada jarak yang signifikan antara *training accuracy* dengan *validation accuracy*.

4.2.2 Arsitektur Akhir Model

Sebelumnya kita telah menguji dan melakukan analisis terhadap beberapa parameter seperti jumlah *convolution layer*, nilai *dropout*, nilai *learning rate*, dan jumlah *epochs*. Dari hasil tersebut didapatkan bahwa model terbaik dihasilkan menggunakan 4 lapisan *Convolution Layer*, nilai *dropout* sebesar 0,5, nilai *learning rate* sebesar 0,0001 yang dilatih dengan 250 *epochs*. Pada Gambar 4.18 adalah visualisasi arsitektur jaringan *Convolutional Neural Network* yang digunakan dalam penelitian ini untuk mendeteksi ekspresi wajah peserta kelas *online*.



Gambar 4.18 Arsitektur Jaringan *Convolutional Neural Network* yang Digunakan

Dari visualisasi pada Gambar 4.18 merupakan arsitektur lengkap yang digunakan pada proses pelatihan model yang telah diuji menghasilkan model yang cukup optimal. Adapun penjelasan dari alur arsitektur tersebut antara lain:

- 1) Proses *input* data citra yang akan dilakukan *feature learning* dan *classification* terlebih dahulu dilakukan *resize* atau *reshape* menjadi berukuran 48x48 dengan *color mode grayscale*.
- 2) Proses *convolution* pertama menggunakan *kernel* berukuran 3x3 dengan 64 filter dan *padding same* yang melakukan proses kombinasi antara dua matriks yang akan menghasilkan matriks baru sebagai *output* proses ini. Setelah berhasil melakukan perhitungan *convolution*, hasilnya akan terlebih dahulu ditambahkan proses *activation* dengan *ReLU* yang mengubah nilai

negatif menjadi 0 pada hasil *convolution*. *Output* dari proses ini memiliki ukuran yang sama yaitu 48x48.

- 3) Proses *pooling* pertama akan mengompres matriks dengan *kernel pooling* yang berukuran 2x2 menggunakan fungsi *max-pooling*. Proses *pooling* dengan fungsi *max-pooling* merupakan proses yang mengambil nilai terbesar dari setiap pergeseran pada area *feature map* yang akan menghasilkan matriks baru yang sudah dikurangi ukurannya. *Output* dari proses ini adalah berupa matriks berukuran 24x24 sesuai nilai *stride* pada *kernel* yang melakukan pergeseran sebanyak 2. Pada akhir proses *pooling* ini juga diterapkan proses *dropout* sebesar 0,5 yang bertujuan menghilangkan sebagian *neuron* yang aktif agar model tidak menjadi *overfitting*.
- 4) Proses *convolution* kedua menggunakan *kernel* berukuran 5x5 dengan 128 *filter* dan *padding same*. Proses ini sama halnya seperti proses *convolution* pertama yang menjadi perbedaan hanyalah pada ukuran *input* yang sudah dilakukan pooling menjadi ukuran 24x24. Pada akhir proses ini juga ditambahkan proses *activation* dengan *ReLU*.
- 5) Proses *pooling* kedua juga menggunakan *kernel* berukuran 2x2 yang menghasilkan *output* berukuran separuh *input* yaitu 12x12. Pada akhir proses ini juga diterapkan proses *dropout* sebesar 0,5.
- 6) Proses *convolution* ketiga menggunakan *kernel* berukuran 3x3 dengan 512 *filter* dan *padding same*. Pada akhir proses ini juga ditambahkan proses *activation* dengan *ReLU*.
- 7) Proses *pooling* ketiga juga menggunakan *kernel* berukuran 2x2 yang menghasilkan *output* berukuran separuh *input* yaitu 6x6. Pada akhir proses ini juga diterapkan proses *dropout* sebesar 0,5.
- 8) Proses *convolution* keempat juga menggunakan *kernel* berukuran 3x3 dengan 512 *filter* dan *padding same*. Pada akhir proses ini juga ditambahkan proses *activation* dengan *ReLU*.
- 9) Proses *pooling* keempat juga menggunakan *kernel* berukuran 2x2 yang menghasilkan *output* berukuran separuh *input* yaitu 6x6. Pada akhir proses ini juga diterapkan proses *dropout* sebesar 0,5.

- 10) Proses *flatten* adalah meratakan matriks menjadi satu baris atau *vector*. Proses ini memiliki *input* berukuran $3 \times 3 \times 512$ sesuai dengan *filter* terakhir pada proses *convolution* keempat. Hasil dari proses *flatten* ini menghasilkan *vector* dengan panjang 4608.
- 11) Proses *fully connected* pertama menggunakan *hidden layer* dengan *neuron* sebanyak 256 dan menggunakan *activation ReLu*. Proses ini merupakan bagian dari proses *classification* untuk melakukan prediksi citra. Pada akhir proses ini juga diterapkan proses *dropout* sebesar 0,5.
- 12) Proses *fully connected* kedua menggunakan *hidden layer* dengan *neuron* sebanyak 512 dan menggunakan *activation ReLu*. Proses ini merupakan bagian dari proses *classification* untuk melakukan prediksi citra. Pada akhir proses ini juga diterapkan proses *dropout* sebesar 0,5.
- 13) Proses *output* dengan *neuron* sebanyak jumlah ekspresi wajah yaitu 7 dan menggunakan *activation softmax*. *Output* dari proses ini adalah berupa label kelas yang dihasilkan dari keseluruhan proses *feature learning* dan *classification*.

4.2.3 Hasil Deteksi Video

Hasil dari pengujian dari rekaman video kelas *online* melalui platform *online meeting* memiliki banyak variasi dari jumlah wajah yang terdeteksi. Dari 5 pengujian dengan video yang berbeda ekspresi paling banyak terdeteksi 1006 kali dan paling sedikit adalah 401 kali selama pemutaran video yang berdurasi 120 detik. Selain itu juga memiliki tingkat keyakinan tertinggi pada ekspresi yang ditampilkan dengan rata-rata yaitu sebesar 0,64 dan terendah yaitu sebesar 0,60. Hal ini menunjukkan bahwa tingkat keyakinan yang dihasilkan relatif rendah. Hasil dari deteksi ekspresi wajah melalui rekaman video dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Pendekripsi Wajah dari Rekaman Video Kegiatan Kelas *Online* yang Berbeda dengan Durasi 120 Detik

Pengujian	Total Ekspresi yang Terdeteksi	Rata-rata Tingkat Keyakinan
1	1006	0,60
2	401	0,62
3	368	0,64
4	988	0,60
5	433	0,62

BAB 5

PENUTUP

Pada bab ini berisi simpulan dari keseluruhan hasil penelitian serta saran-saran untuk pengembangan dan penelitian lebih lanjut.

5.1 Simpulan

Berdasarkan hasil dari analisis dan eksperimen yang telah dilakukan, dapat diperoleh simpulan sebagai berikut:

- 1) Implementasi *deep learning* dengan menerapkan algoritma *Convolutional Neural Network* untuk melakukan klasifikasi wajah secara *real-time* berhasil dilakukan dengan memanfaatkan *package* atau *library* *Tensorflow*, *Keras*, dan *OpenCV*.
- 2) Akurasi terbaik dihasilkan oleh arsitektur *Convolutional Neural Network* dengan 4 lapisan *convolutional layer* yang terdiri dari *conv2d* dan *max pooling*, 2 lapisan *Fully Connected Layer*, penggunaan *dropout* sebesar 0,5, dan menggunakan *learning rate* sebesar 0,0001.
- 3) Evaluasi model yang diperoleh yaitu *training accuracy* sebesar 0,7802 dan *validation accuracy* sebesar 0,6772. Selain itu juga menghasilkan *training loss* sebesar 0,6021 dan *validation loss* sebesar 0,9832 yang menunjukkan bahwa model belum sepenuhnya dirancang dan dilatih dengan baik.
- 4) Penggunaan *learning rate* sebesar 0,0001 menghasilkan evaluasi yang lebih baik. Terbukti karena jarak antara *training* dan *validation accuracy* tidak terlalu jauh atau dapat dikatakan tidak *overfitting*.
- 5) Penggunaan *epoch* sebesar 250 menghasilkan akurasi yang lebih baik. Terbukti sampai pada iterasi *epoch* yang ke 250 masih ada peningkatan nilai *validation accuracy* yang dihasilkan.
- 6) Hasil dari tingkat keyakinan yang diperoleh dari pengujian model pada rekaman video kegiatan kelas *online* pada platform *online meeting* berkisar antara 0,60 sampai dengan 0,64 yang menunjukkan bahwa tingkat keyakinan model relatif rendah.

5.2 Saran

Berdasarkan hasil yang telah diperoleh dari penelitian ini, peneliti memberikan saran kepada peneliti selanjutnya dalam bidang dan objek penelitian yang sama, antara lain:

- 1) Menambah jumlah *dataset* dari seluruh ekspresi khususnya pada ekspresi *disgust* karena pada penelitian ini memiliki jumlah *dataset* yang sangat sedikit. Menambah jumlah *dataset* dapat dilakukan salah satunya dengan menggabungkan *dataset facial expression recognition* yang tersedia dan terbuka untuk umum.
- 2) Menambah analisis dengan menambahkan jumlah *filter* pada *convolution layer* serta penerapan *augmentation* untuk menambah keakuratan dalam prediksi citra.
- 3) Menguji dan melakukan pelatihan model dengan *optimizer compiler* yang berbeda atau pun dengan parameter yang berbeda agar mendapatkan evaluasi model yang lebih baik lagi.
- 4) Memfokuskan target kegiatan yang spesial seperti kegiatan *training*, seminar, dan lain sebagainya agar manfaat dari hasil penelitian tersebut dapat diperluas lagi.
- 5) Penelitian ini didasari oleh suasana pandemi yang dirasakan pada tahun 2021. Oleh karena itu, penelitian ini dapat dilanjutkan dan diterapkan pada kamera pengawas ataupun perangkat lainnya yang dapat digunakan saat kegiatan kelas secara luring.

DAFTAR PUSTAKA

- Abdillah, L. A. (2020). *Online Learning Menggunakan Zoom Teleconference*. <http://eprints.binadarma.ac.id/4162/2/Abdillah2020 %5BOnline Learning Menggunakan Zoom Teleconference%5D 2020415.pdf>
- Abidin, Z., & Harjoko, A. (2011). Facial Expression Recognition By Using Fisherface Methode With Backpropagation Neural Network. *Ijcss*, 5(1), 83-91.
- Amda, K., & Fitriyani, R. (2016). Membaca ekspresi wajah. Huta Publisher.
- Anggara, M. R. P. (2020). Transfer Learning Dengan Model Arsitektur Vgg16 Dan Inception-V3 Untuk Klasifikasi Penyakit Daun Kentang.
- Arikunto, S. (2010). Metode penelitian. Jakarta: Rineka Cipta.
- Dwi, B., Amelia, A., Hasanah, U., & Putra, A. M. (2020). Analisis Keefektifan Pembelajaran Online di Masa Pandemi Covid-19. *Jurnal Pendidikan Guru Sekolah Dasar*, 2(1), 3.
- Edi Ismanto, E. P. C. (2017). Jaringan Saraf Tiruan Algoritma Backpropagation Dalam Memprediksi Ketersediaan Komoditi Pangan Provinsi Riau. *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, 2(2), 196–209. <https://doi.org/10.36341/rabit.v2i2.152>
- Eka Putra, W. S. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, 5(1). <https://doi.org/10.12962/j23373539.v5i1.15696>
- Firman, F., & Rahayu, S. (2020). Pembelajaran Online di Tengah Pandemi Covid-19. *Indonesian Journal of Educational Science (IJES)*, 2(2), 81–89. <https://doi.org/10.31605/ijes.v2i2.659>
- Fitri, N. F., & Adelya, B. (2017). Kematangan emosi remaja dalam pengentasan masalah. *Jurnal Penelitian Guru Indonesia*, 2(2), 30–39. <https://jurnal.iicet.org/index.php/jpgi/article/view/225>

- Hamdani M. (2019). Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (Cnn) Dengan Tensorflow
- Hasibuan, Z. A. (2007). Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi. *Konsep, Teknik, Dan Aplikasi, Universitas Indonesia*, 194.
- Hilmi M. A (2014) Identifikasi Suara Menggunakan Fft Dan Neural Network. *SSRN Electronic Journal*, 5(564), 1–19. <https://doi.org/10.4324/9781315853178>
- Liu, A. N. A. M., & Ilyas, I. (2020). Pengaruh Pembelajaran Online Berbasis Zoom Cloud Meeting Terhadap Hasil Belajar Mahasiswa Fisika Universitas Flores. *Jurnal Pendidikan Fisika Dan Keilmuan (JPFK)*, 6(1), 34. <https://doi.org/10.25273/jpfk.v6i1.7303>
- Lubis, C. & Perdana, N. J. (2020). Sistem Pendekripsi Dan Pengenalan Ekspresi Wajah Dengan Algoritma Yolo Dan Convolutional Neural Network <https://doi.org/10.1016/j.tmaid.2020.101607%0A>
- Marono, A., Clarke, D. D., Navarro, J., & Keatley, D. A. (2017). A behaviour sequence analysis of nonverbal communication and deceit in different personality clusters. *Psychiatry, Psychology and Law*, 24(5), 730-744.
- Nadhiroh, Y. F. (2015). Pengendalian Emosi. *Jurnal Saintifika Islamica*, 2(1), 53–63.
- Nurhikmat, T. (2018). Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (Cnn) Pada Citra Wayang Golek.
- Pratikno, A. S. (2018). Implementasi Artificial Intelligence dalam Memetakan Karakteristik, Kompetensi, dan Perkembangan Psikologi Siswa Sekolah Dasar Melalui Platform Offline Conference. *Isbn, September 2017*, 18–27.
- Prawitasllri, J. E. (2016). Mengenal Emosi Melalui Komunikasi Nonverbal. *Buletin Psikologi*, 3(1), 27–43. <https://doi.org/10.22146/bpsi.13384>
- Purnamajaya, I. A. (2019). Pembangunan Aplikasi Gesture To Text Dan Text To

- Speech Untuk Penderita Tunawicara (Studi Kasus Di Slb B Sukapura) Program Studi Teknik Informatika Fakultas Teknik Dan Ilmu Komputer Universitas Komputer Indonesia 2019. (2019). 10115045.
- Ramadhan, J. (2020). Implementasi Algoritma Artificial Neural Network (Ann) Pada Pendekripsi Uang Kertas Laporan Skripsi.
- Rena, P. N. (2019). Penerapan Metode Convolutional Neural Network Pada Pendekripsi Gambar Notasi Balok (Vol. 87, Issue 1,2).
- Sari, M. Y. (2019). Komunikasi Antarbudaya Studi Negosiasi Wajah Dalam Interaksi Etnik Batak Dan Etnik Minang Di Duri Kelurahan Gadjah Sakti Kecamatan Mandau Kabupaten Bengkalis. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.
- Sayoga, D., Kusuma, P. D., & Hasibuan, F. C. (2020). Pengembangan Sistem Deteksi Occupancy Menggunakan Computer Vision Untuk Smart Building Dan Automation. 7(2), 4751–4758
- Siang, J. (2005). Jaringan Saraf Tiruan dan Pemrogramannya Menggunakan Matlab. Andi Yogyakarta. Yogyakarta.
- Sugiyono. (2012). Metode Penelitian Administrasi dan R&D. Bandung: Alfabeta.
- Sugiyono, P. (2015). Metode penelitian kombinasi (mixed methods). Bandung: Alfabeta, 28.
- Thohari, A., & Hertantyo, G. B. (2018). Implementasi Convolutional Neural Network untuk Klasifikasi Pembalap MotoGP Berbasis GPU. *Proceedings on Conference on Electrical Engineering, Telematics, Industrial Technology, and Creative Media*, 50–55.
- Wulan A. (2020). Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network Dengan Tensorflow
- Zainuddin, M., Sianturi, L. T., & Hondro, R. K. (2017). Implementasi Metode Robinson Operator 3 Level Untuk Mendekripsi Tepi Pada Citra Digital. *Jurnal Riset Komputer (JURIKOM)*, 4(4), 1–5.

LAMPIRAN

Lampiran 1: Arsitektur Lengkap Model

No	Layer Name	Layer Type	Output Shape	Param	Group
1	conv2d	Conv2D	(None, 48, 48, 64)	640	Input and 1 st Conv Layer
2	batch_normalization	BatchNormalization	(None, 48, 48, 64)	256	
3	activation	Activation	(None, 48, 48, 64)	0	
4	max_pooling2d	MaxPooling2D	(None, 24, 24, 64)	0	
5	dropout	Dropout	(None, 24, 24, 64)	0	
6	conv2d_1	Conv2D	(None, 24, 24, 128)	204928	2 nd Conv Layer
7	batch_normalization_1	BatchNormalization	(None, 24, 24, 128)	512	
8	activation_1	Activation	(None, 24, 24, 128)	0	
9	max_pooling2d_1	MaxPooling2D	(None, 12, 12, 128)	0	
10	dropout_1	Dropout	(None, 12, 12, 128)	0	3 rd Conv Layer
11	conv2d_2	Conv2D	(None, 12, 12, 512)	590336	
12	batch_normalization_2	BatchNormalization	(None, 12, 12, 512)	2048	
13	activation_2	Activation	(None, 12, 12, 512)	0	
14	max_pooling2d_2	MaxPooling2D	(None, 6, 6, 512)	0	4 th Conv Layer
15	dropout_2	Dropout	(None, 6, 6, 512)	0	
16	conv2d_3	Conv2D	(None, 6, 6, 512)	2359808	
17	batch_normalization_3	BatchNormalization	(None, 6, 6, 512)	2048	
18	activation_3	Activation	(None, 6, 6, 512)	0	1 st Fully Conn Layer
19	max_pooling2d_3	MaxPooling2D	(None, 3, 3, 512)	0	
20	dropout_3	Dropout	(None, 3, 3, 512)	0	
21	flatten	Flatten	(None, 4608)	0	
22	dense	Dense	(None, 256)	1179904	2 nd Fully Conn Layer
23	batch_normalization_4	BatchNormalization	(None, 256)	1024	
24	activation_4	Activation	(None, 256)	0	
25	dropout_4	Dropout	(None, 256)	0	
26	dense_1	Dense	(None, 512)	131584	Output
27	batch_normalization_5	BatchNormalization	(None, 512)	2048	
28	activation_5	Activation	(None, 512)	0	
29	dropout_5	Dropout	(None, 512)	0	
30	dense_2	Dense	(None, 7)	3591	

Lampiran 2: Hasil Evaluasi Lengkap Pengaruh Lapisan *Convolution Layer*

Epoch	2 Lapisan Convolution Layer				3 Lapisan Convolution Layer				4 Lapisan Convolution Layer			
	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	0,3585	0,3413	0,2974	1,6730	1,7134	1,8063	0,3005	0,3119	0,2508	1,7641	1,7668	2,1993
2	0,5050	0,4992	0,4729	1,3010	1,3046	1,3791	0,3850	0,4554	0,4026	1,5441	1,4126	1,4973
3	0,5759	0,5743	0,5483	1,1143	1,1280	1,1865	0,4082	0,5257	0,4051	1,5506	1,2861	1,5681
4	0,6482	0,6304	0,6001	0,9328	0,9824	1,0602	0,4827	0,5499	0,5329	1,3943	1,2076	1,2579
5	0,7257	0,6833	0,6414	0,7346	0,8417	0,9537	0,4690	0,4654	0,5204	1,5566	1,4319	1,2545
6	0,8036	0,7451	0,6868	0,5281	0,6845	0,8441	0,3368	0,5366	0,5474	2,5994	1,4143	1,2032
7	0,8659	0,8082	0,7261	0,3751	0,5243	0,7275	0,4955	0,5458	0,5059	1,8772	1,5046	1,3887
8	0,9093	0,8606	0,7767	0,2618	0,3787	0,6084	0,4746	0,5148	0,5617	2,2159	1,8438	1,3543
9	0,9337	0,8989	0,8274	0,1932	0,2871	0,4741	0,5165	0,5343	0,5201	2,2527	1,9990	1,6836
10	0,9497	0,9288	0,8618	0,1492	0,2059	0,3799	0,5497	0,5193	0,5656	2,3431	2,2282	1,6042
11	0,9600	0,9435	0,8987	0,1212	0,1650	0,2861	0,5145	0,4163	0,4648	2,5791	3,6703	2,2634
12	0,9602	0,9510	0,9167	0,1163	0,1420	0,2356	0,5187	0,5681	0,5491	2,7847	2,1454	2,0964
13	0,9640	0,9569	0,9350	0,1052	0,1270	0,1885	0,5472	0,5229	0,6007	2,5775	3,0037	1,9223
14	0,9671	0,9626	0,9458	0,0942	0,1082	0,1609	0,4919	0,5410	0,5742	3,0175	2,4532	2,1112
15	0,9698	0,9643	0,9549	0,0885	0,1030	0,1324	0,5033	0,5483	0,5555	2,7777	2,5655	2,4083
16	0,9710	0,9677	0,9571	0,0843	0,0971	0,1252	0,5237	0,5728	0,5943	3,0321	2,6695	2,3471
17	0,9711	0,9690	0,9637	0,0821	0,0900	0,1070	0,5321	0,5840	0,5951	2,9664	2,3461	2,1184
18	0,9769	0,9699	0,9636	0,0675	0,0883	0,1018	0,5427	0,5748	0,5753	2,9792	2,5300	2,3638
19	0,9747	0,9729	0,9639	0,0716	0,0805	0,1040	0,5075	0,5315	0,5951	3,1828	2,9025	2,4042
20	0,9728	0,9697	0,9692	0,0797	0,0898	0,0874	0,5366	0,5664	0,5951	3,1880	2,6850	2,5007
21	0,9766	0,9771	0,9700	0,0678	0,0691	0,0874	0,5229	0,5974	0,5452	3,1878	2,5335	2,6465
22	0,9775	0,9802	0,9682	0,0659	0,0572	0,0922	0,5329	0,5804	0,5868	3,1259	2,7985	2,4268
23	0,9836	0,9763	0,9742	0,0490	0,0681	0,0756	0,5474	0,5739	0,5273	3,3576	2,8606	2,9496
24	0,9770	0,9763	0,9695	0,0679	0,0665	0,0893	0,5318	0,5310	0,5859	3,1013	3,4254	2,4434
25	0,9775	0,9731	0,9744	0,0648	0,0794	0,0735	0,5287	0,5765	0,5935	2,9287	3,0060	2,5257
26	0,9833	0,9801	0,9753	0,0492	0,0576	0,0699	0,5036	0,5988	0,5890	3,1479	2,7670	2,5263
27	0,9804	0,9801	0,9782	0,0552	0,0587	0,0643	0,5276	0,5583	0,5527	3,0261	3,0572	2,9607

Epoch	2 Lapisan Convolution Layer				3 Lapisan Convolution Layer				4 Lapisan Convolution Layer			
	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
28	0,9813	0,9800	0,9778	0,0554	0,0544	0,0655	0,5480	0,5485	0,6133	3,1477	2,8935	2,5332
29	0,9835	0,9779	0,9814	0,0446	0,0610	0,0564	0,5533	0,5539	0,6016	3,2349	3,0739	2,6799
30	0,9826	0,9828	0,9769	0,0513	0,0507	0,0649	0,5614	0,5416	0,5949	3,4126	3,0782	2,6143
31	0,9818	0,9801	0,9760	0,0496	0,0569	0,0699	0,5251	0,5815	0,5876	3,7940	2,9980	3,0689
32	0,9796	0,9831	0,9810	0,0593	0,0506	0,0535	0,5173	0,5619	0,5765	3,0249	2,9438	2,7058
33	0,9855	0,9820	0,9814	0,0425	0,0499	0,0559	0,5254	0,5145	0,5924	3,0615	3,4636	2,6219
34	0,9887	0,9789	0,9803	0,0314	0,0594	0,0577	0,5480	0,5737	0,5725	3,7294	2,9850	3,1448
35	0,9836	0,9834	0,9806	0,0492	0,0463	0,0536	0,4718	0,5801	0,5739	3,1080	3,0378	2,7654
36	0,9853	0,9820	0,9793	0,0443	0,0534	0,0590	0,5499	0,5374	0,6063	3,4983	2,8827	2,6997
37	0,9855	0,9856	0,9808	0,0405	0,0413	0,0546	0,5223	0,5787	0,5974	3,5439	3,0673	2,8290
38	0,9835	0,9869	0,9846	0,0461	0,0364	0,0448	0,5321	0,5647	0,5845	3,4945	3,1370	2,9424
39	0,9837	0,9840	0,9843	0,0469	0,0443	0,0444	0,5254	0,5879	0,5918	3,4070	3,2431	3,0077
40	0,9875	0,9822	0,9856	0,0354	0,0517	0,0438	0,5458	0,5633	0,5748	3,2677	3,0734	2,7842
41	0,9888	0,9867	0,9828	0,0308	0,0385	0,0499	0,5142	0,5857	0,5949	3,5231	3,3446	2,8654
42	0,9868	0,9886	0,9842	0,0366	0,0313	0,0459	0,5572	0,5753	0,5862	3,3967	3,3222	2,8157
43	0,9848	0,9834	0,9826	0,0469	0,0453	0,0493	0,5290	0,5728	0,6208	3,2949	3,3926	3,0183
44	0,9885	0,9849	0,9834	0,0329	0,0418	0,0489	0,5349	0,5770	0,5611	3,0363	3,2761	2,8817
45	0,9891	0,9866	0,9861	0,0301	0,0364	0,0409	0,5382	0,5935	0,6161	3,6754	3,4389	2,8499
46	0,9885	0,9866	0,9861	0,0339	0,0375	0,0397	0,5438	0,5633	0,6203	3,6236	3,4981	2,9326
47	0,9876	0,9894	0,9873	0,0343	0,0293	0,0346	0,4983	0,5469	0,5628	3,3737	3,7244	3,6604
48	0,9889	0,9858	0,9816	0,0309	0,0414	0,0513	0,5187	0,5480	0,6172	3,5981	3,4739	2,7774
49	0,9881	0,9845	0,9859	0,0347	0,0431	0,0413	0,5421	0,5865	0,5134	3,5485	3,1986	3,3805
50	0,9891	0,9883	0,9843	0,0324	0,0315	0,0437	0,5525	0,5818	0,6180	3,8763	3,4646	3,0818
51	0,9874	0,9892	0,9886	0,0354	0,0317	0,0343	0,5126	0,5784	0,6016	3,7909	3,3388	3,4154
52	0,9888	0,9874	0,9874	0,0304	0,0357	0,0349	0,5419	0,5633	0,5857	3,3702	3,4598	3,1245
53	0,9885	0,9895	0,9883	0,0346	0,0332	0,0315	0,5527	0,5904	0,5887	3,5823	3,2786	3,0934
54	0,9895	0,9920	0,9860	0,0277	0,0206	0,0420	0,5170	0,5815	0,5784	3,3919	3,8151	3,2645
55	0,9882	0,9865	0,9876	0,0353	0,0399	0,0351	0,5530	0,5991	0,5977	3,5739	3,8503	3,0544
56	0,9897	0,9852	0,9880	0,0265	0,0410	0,0328	0,5421	0,5466	0,6094	3,9468	4,0954	2,8167
57	0,9901	0,9890	0,9874	0,0279	0,0301	0,0369	0,5424	0,5684	0,6152	3,9104	3,4828	2,9044

Epoch	2 Lapisan Convolution Layer				3 Lapisan Convolution Layer				4 Lapisan Convolution Layer			
	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
58	0,9894	0,9871	0,9877	0,0331	0,0352	0,0340	0,5338	0,5834	0,6038	3,6827	3,1468	3,0460
59	0,9901	0,9898	0,9873	0,0277	0,0282	0,0364	0,5340	0,5876	0,6102	3,7768	3,5196	2,8233
60	0,9900	0,9910	0,9882	0,0289	0,0236	0,0348	0,5569	0,5717	0,5938	3,8585	3,6493	2,8646
61	0,9904	0,9914	0,9888	0,0246	0,0242	0,0323	0,5491	0,5965	0,5979	3,7566	3,7753	2,9068
62	0,9911	0,9891	0,9896	0,0261	0,0306	0,0288	0,5089	0,5831	0,6016	3,3381	3,8814	3,2268
63	0,9901	0,9893	0,9889	0,0256	0,0318	0,0325	0,5366	0,5857	0,5876	3,5038	3,4613	3,1075
64	0,9919	0,9893	0,9908	0,0223	0,0307	0,0270	0,5552	0,5647	0,5935	3,9165	3,7786	3,1887
65	0,9907	0,9889	0,9872	0,0261	0,0295	0,0359	0,5338	0,5893	0,6122	3,7658	3,4517	3,2701
66	0,9916	0,9906	0,9846	0,0249	0,0251	0,0430	0,5226	0,6010	0,6136	3,7095	3,5537	2,7500
67	0,9907	0,9909	0,9887	0,0250	0,0266	0,0303	0,5527	0,5795	0,6144	4,0546	3,6611	2,9586
68	0,9920	0,9923	0,9890	0,0221	0,0211	0,0317	0,5513	0,6088	0,6038	3,9211	3,7782	2,9193
69	0,9893	0,9920	0,9907	0,0305	0,0206	0,0265	0,5458	0,6010	0,6052	3,4976	3,7002	3,0417
70	0,9918	0,9889	0,9928	0,0225	0,0324	0,0186	0,5117	0,5672	0,6110	3,6625	3,7383	3,1562
71	0,9911	0,9864	0,9923	0,0246	0,0382	0,0214	0,5343	0,5876	0,5996	4,5683	3,7165	3,8206
72	0,9896	0,9920	0,9866	0,0284	0,0211	0,0379	0,5499	0,5795	0,5924	3,5194	3,6371	3,2769
73	0,9905	0,9926	0,9876	0,0290	0,0186	0,0379	0,5165	0,5968	0,6175	3,3023	4,0916	2,9338
74	0,9908	0,9905	0,9897	0,0264	0,0269	0,0283	0,5371	0,5725	0,6136	3,9723	4,0645	3,0370
75	0,9931	0,9889	0,9925	0,0182	0,0341	0,0196	0,5405	0,5513	0,6177	3,8702	2,9465	3,2108
76	0,9915	0,9889	0,9927	0,0238	0,0312	0,0188	0,5578	0,5907	0,5831	3,8650	3,1802	3,4866
77	0,9911	0,9941	0,9893	0,0241	0,0153	0,0299	0,5458	0,5949	0,5608	3,7933	3,6263	3,4721
78	0,9924	0,9951	0,9887	0,0199	0,0120	0,0335	0,5586	0,6030	0,6035	4,0867	4,1116	3,4272
79	0,9934	0,9906	0,9908	0,0164	0,0278	0,0253	0,5684	0,5477	0,5354	4,3392	3,9310	3,3221
80	0,9942	0,9914	0,9884	0,0173	0,0255	0,0319	0,5511	0,5703	0,5628	3,9704	3,6389	3,2111
81	0,9913	0,9920	0,9902	0,0226	0,0225	0,0267	0,5444	0,5711	0,5929	4,4636	4,0826	3,0466
82	0,9929	0,9918	0,9908	0,0190	0,0206	0,0258	0,5580	0,5698	0,6136	3,5048	4,1972	3,1684
83	0,9933	0,9910	0,9916	0,0169	0,0252	0,0225	0,5485	0,5960	0,6102	4,2899	3,7007	3,5822
84	0,9905	0,9923	0,9917	0,0274	0,0209	0,0230	0,5335	0,5918	0,6158	3,5708	3,7313	3,5948
85	0,9917	0,9888	0,9902	0,0261	0,0322	0,0265	0,5631	0,5848	0,6152	3,9735	4,0578	3,3497
86	0,9923	0,9925	0,9911	0,0205	0,0216	0,0246	0,5382	0,5926	0,6057	3,5665	3,5758	3,5115
87	0,9937	0,9924	0,9919	0,0175	0,0195	0,0208	0,5444	0,5536	0,6203	4,2670	3,7831	3,3541

Epoch	2 Lapisan Convolution Layer				3 Lapisan Convolution Layer				4 Lapisan Convolution Layer			
	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
88	0,9916	0,9921	0,9929	0,0214	0,0239	0,0197	0,5226	0,5921	0,5516	3,9166	3,4881	3,8916
89	0,9924	0,9936	0,9896	0,0220	0,0170	0,0296	0,5427	0,5834	0,6091	4,0008	3,7676	3,0811
90	0,9928	0,9927	0,9916	0,0218	0,0196	0,0223	0,5589	0,5717	0,6002	3,9252	4,2040	3,2162
91	0,9941	0,9919	0,9903	0,0142	0,0236	0,0268	0,5525	0,5709	0,5991	4,0770	3,8079	3,1077
92	0,9932	0,9925	0,9937	0,0170	0,0215	0,0161	0,5527	0,5932	0,6225	4,2488	3,9006	3,3761
93	0,9929	0,9927	0,9948	0,0215	0,0192	0,0128	0,5513	0,5910	0,5993	3,9323	3,4875	3,3747
94	0,9940	0,9940	0,9906	0,0166	0,0146	0,0269	0,5547	0,5946	0,6046	3,7895	3,6851	3,3482
95	0,9947	0,9920	0,9886	0,0141	0,0241	0,0325	0,5689	0,5684	0,6088	4,4680	3,8793	3,2173
96	0,9918	0,9930	0,9922	0,0229	0,0177	0,0239	0,5508	0,5792	0,6169	3,6278	4,0368	2,8981
97	0,9924	0,9921	0,9923	0,0203	0,0205	0,0212	0,5499	0,6071	0,6032	3,7879	4,2116	3,6568
98	0,9934	0,9910	0,9924	0,0175	0,0276	0,0211	0,5502	0,5845	0,6180	4,5090	3,5323	3,3993
99	0,9931	0,9934	0,9943	0,0179	0,0166	0,0129	0,5424	0,5770	0,6317	4,2234	3,5417	3,4322
100	0,9923	0,9941	0,9943	0,0220	0,0153	0,0159	0,5393	0,5932	0,6099	4,0150	3,6423	3,4275

Lampiran 3: Hasil Evaluasi Lengkap Pengaruh Penggunaan Nilai *Dropout*

Epoch	Dropout 0,25				Dropout 0,5			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
1	0,2833	0,2295	1,8531	1,9521	0,2503	0,2506	1,9483	1,8436
2	0,4350	0,3023	1,4616	1,7278	0,3211	0,2514	1,7592	2,3837
3	0,5082	0,4010	1,2843	1,5441	0,4972	0,2960	1,2812	1,8571
4	0,5504	0,4451	1,1842	1,4377	0,4749	0,4646	1,3077	1,4222
5	0,5701	0,4715	1,1279	1,3714	0,5145	0,4400	1,2892	1,4113
6	0,5957	0,4911	1,0749	1,3206	0,5106	0,4481	1,2953	1,3670
7	0,6140	0,5112	1,0240	1,2806	0,5675	0,4325	1,1833	1,4496
8	0,6295	0,5231	0,9770	1,2512	0,5195	0,5349	1,2553	1,2191
9	0,6472	0,5335	0,9382	1,2292	0,5720	0,5304	1,1492	1,2214
10	0,6671	0,5406	0,8935	1,2093	0,5717	0,4676	1,1416	1,4026
11	0,6815	0,5484	0,8518	1,1914	0,4501	0,4802	1,4482	1,3881
12	0,6974	0,5528	0,8122	1,1768	0,6024	0,5338	1,1280	1,1923
13	0,7118	0,5612	0,7739	1,1591	0,5597	0,5589	1,2353	1,1717
14	0,7317	0,5693	0,7249	1,1461	0,5159	0,5667	1,4613	1,1410
15	0,7484	0,5750	0,6876	1,1301	0,4819	0,5195	1,6215	1,2662
16	0,7629	0,5818	0,6496	1,1130	0,5678	0,5268	1,2889	1,2294
17	0,7747	0,5858	0,6125	1,1008	0,5725	0,5541	1,2662	1,1693
18	0,7913	0,5872	0,5664	1,0990	0,6150	0,5714	1,1784	1,1349
19	0,8021	0,5931	0,5426	1,0840	0,5884	0,5151	1,2708	1,2749
20	0,8138	0,5985	0,5093	1,0707	0,5823	0,5343	1,2870	1,2224
21	0,8253	0,5994	0,4795	1,0632	0,6242	0,5410	1,1961	1,2111
22	0,8376	0,6102	0,4475	1,0494	0,5851	0,5742	1,3653	1,1117
23	0,8410	0,6082	0,4327	1,0487	0,5907	0,3318	1,3465	1,7297
24	0,8519	0,6134	0,4087	1,0311	0,6032	0,5848	1,2886	1,0972
25	0,8605	0,6169	0,3897	1,0267	0,5999	0,4994	1,3620	1,3047
26	0,8700	0,6162	0,3634	1,0223	0,6136	0,6077	1,4757	1,0322
27	0,8735	0,6221	0,3536	1,0137	0,5773	0,5921	1,4829	1,0819
28	0,8802	0,6303	0,3344	0,9954	0,6217	0,5745	1,2936	1,1134
29	0,8849	0,6283	0,3234	0,9997	0,6180	0,5882	1,4505	1,0727
30	0,8928	0,6321	0,3052	0,9840	0,6016	0,5530	1,4384	1,1692
31	0,8957	0,6367	0,2920	0,9774	0,6091	0,5820	1,4969	1,0880
32	0,8969	0,6398	0,2858	0,9653	0,5843	0,5912	1,5170	1,0677
33	0,9038	0,6447	0,2765	0,9561	0,6102	0,5631	1,6389	1,1181
34	0,9069	0,6449	0,2607	0,9517	0,6339	0,6205	1,4984	1,0159
35	0,9086	0,6481	0,2540	0,9480	0,6314	0,5059	1,4715	1,2901
36	0,9129	0,6492	0,2443	0,9362	0,6247	0,5949	1,5697	1,0711
37	0,9159	0,6525	0,2397	0,9358	0,6297	0,5921	1,5596	1,0918
38	0,9173	0,6597	0,2315	0,9227	0,6186	0,5818	1,5612	1,1038
39	0,9204	0,6632	0,2273	0,9136	0,6328	0,5502	1,5519	1,1561
40	0,9230	0,6604	0,2216	0,9094	0,6247	0,6166	1,5968	1,0304
41	0,9245	0,6617	0,2172	0,9071	0,5823	0,5826	1,8471	1,1290
42	0,9280	0,6664	0,2052	0,8990	0,6359	0,6289	1,5832	0,9860
43	0,9261	0,6686	0,2108	0,8891	0,6183	0,6350	1,6410	0,9932
44	0,9284	0,6745	0,2006	0,8771	0,6186	0,4810	1,6358	1,3450
45	0,9315	0,6724	0,1956	0,8796	0,5932	0,5608	1,7904	1,1574
46	0,9349	0,6730	0,1872	0,8748	0,6320	0,5664	1,7041	1,1490
47	0,9355	0,6762	0,1892	0,8676	0,6164	0,5773	1,6533	1,1030
48	0,9350	0,6789	0,1866	0,8669	0,6127	0,6044	1,6676	1,0542
49	0,9360	0,6836	0,1853	0,8502	0,6289	0,6278	1,5451	1,0275
50	0,9374	0,6810	0,1771	0,8529	0,6311	0,6222	1,5850	1,0239

Epoch	Dropout 0,25				Dropout 0,5			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
51	0,9381	0,6872	0,1734	0,8516	0,6395	0,6052	1,6702	1,0608
52	0,9390	0,6871	0,1738	0,8401	0,6412	0,6119	1,5835	1,0547
53	0,9402	0,6934	0,1705	0,8322	0,6381	0,5938	1,6538	1,1139
54	0,9416	0,6956	0,1680	0,8247	0,6479	0,6094	1,5880	1,0682
55	0,9458	0,6953	0,1558	0,8219	0,6448	0,6297	1,6434	1,0161
56	0,9442	0,6997	0,1606	0,8131	0,6345	0,6381	1,6333	0,9897
57	0,9461	0,7044	0,1543	0,8105	0,6381	0,6004	1,6691	1,0698
58	0,9478	0,7049	0,1473	0,8051	0,6345	0,6359	1,7281	0,9917
59	0,9477	0,7048	0,1492	0,7998	0,6289	0,6490	1,8588	0,9865
60	0,9475	0,7083	0,1500	0,7909	0,6250	0,6004	1,6463	1,0839
61	0,9455	0,7106	0,1533	0,7882	0,5882	0,6412	1,9647	0,9914
62	0,9465	0,7082	0,1531	0,7890	0,6080	0,6180	1,8537	1,0479
63	0,9506	0,7107	0,1415	0,7847	0,6401	0,5483	1,7414	1,2604
64	0,9519	0,7136	0,1393	0,7728	0,6339	0,6172	1,8392	1,0592
65	0,9546	0,7161	0,1318	0,7743	0,5619	0,5946	2,0921	1,1237
66	0,9518	0,7157	0,1358	0,7668	0,6462	0,6272	1,7609	1,0495
67	0,9508	0,7161	0,1416	0,7568	0,6191	0,6275	1,8341	1,0225
68	0,9513	0,7226	0,1388	0,7552	0,6094	0,6152	1,7947	1,0963
69	0,9516	0,7233	0,1377	0,7575	0,6306	0,6462	1,7389	1,0027
70	0,9530	0,7270	0,1347	0,7442	0,6392	0,6560	1,8834	0,9937
71	0,9545	0,7298	0,1293	0,7384	0,6158	0,6451	1,8647	0,9801
72	0,9557	0,7291	0,1293	0,7360	0,6493	0,6083	1,9133	1,0932
73	0,9570	0,7309	0,1220	0,7331	0,6515	0,6487	1,7505	1,0120
74	0,9581	0,7276	0,1259	0,7345	0,6409	0,6069	1,9478	1,0916
75	0,9573	0,7317	0,1225	0,7271	0,6482	0,6398	1,8426	1,0008
76	0,9567	0,7306	0,1247	0,7324	0,6272	0,6540	1,7559	1,0087
77	0,9577	0,7388	0,1230	0,7148	0,6390	0,6339	1,8571	1,0334
78	0,9562	0,7369	0,1240	0,7188	0,6236	0,5893	1,9198	1,1183
79	0,9595	0,7398	0,1186	0,7077	0,6401	0,6317	1,8172	1,0369
80	0,9604	0,7374	0,1134	0,7155	0,6217	0,5918	1,9480	1,1289
81	0,9606	0,7451	0,1153	0,6992	0,6317	0,6057	1,9198	1,0888
82	0,9622	0,7469	0,1118	0,6933	0,6317	0,6496	1,8737	0,9918
83	0,9591	0,7424	0,1213	0,6969	0,6535	0,6395	1,7545	1,0472
84	0,9622	0,7458	0,1096	0,6922	0,6512	0,6272	1,7357	1,0637
85	0,9631	0,7502	0,1076	0,6875	0,6353	0,6509	1,9062	1,0543
86	0,9609	0,7509	0,1127	0,6795	0,6392	0,6325	1,8183	1,0524
87	0,9611	0,7533	0,1111	0,6832	0,6228	0,6451	1,7797	1,0031
88	0,9613	0,7534	0,1079	0,6792	0,6412	0,6127	1,8382	1,0821
89	0,9620	0,7506	0,1097	0,6793	0,6390	0,6314	1,7680	1,1103
90	0,9638	0,7549	0,1061	0,6706	0,6359	0,5857	1,8521	1,1467
91	0,9624	0,7535	0,1053	0,6713	0,6434	0,6465	1,7407	1,0805
92	0,9634	0,7569	0,1048	0,6661	0,6376	0,6546	1,8132	1,0004
93	0,9636	0,7586	0,1037	0,6610	0,6448	0,6451	1,8693	1,0451
94	0,9652	0,7615	0,1001	0,6528	0,6409	0,5988	1,7309	1,1715
95	0,9652	0,7626	0,1039	0,6581	0,6367	0,6560	1,8740	1,0246
96	0,9648	0,7602	0,1019	0,6535	0,6473	0,6468	1,8060	1,0367
97	0,9650	0,7626	0,1006	0,6522	0,6445	0,6456	1,8257	1,0354
98	0,9648	0,7638	0,0992	0,6451	0,6303	0,6498	1,8972	1,0822
99	0,9646	0,7698	0,1015	0,6404	0,6320	0,6080	1,9100	1,1767
100	0,9667	0,7656	0,0968	0,6411	0,6225	0,6362	1,8791	1,1001

Lampiran 4: Hasil Evaluasi Lengkap Pengaruh Besar Nilai *Learning Rate*

Epoch	Learning Rate 0,001				Learning Rate 0,0001			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
1	0,2163	0,1856	2,0539	2,3385	0,2506	0,2506	1,8480	1,8835
2	0,2825	0,2051	1,7827	2,1494	0,2514	0,2506	2,0338	1,8902
3	0,3836	0,2132	1,5830	2,0675	0,3613	0,2506	1,6712	1,8683
4	0,4326	0,2204	1,4627	2,0136	0,4556	0,2517	1,4003	1,8489
5	0,4638	0,2288	1,3868	1,9598	0,4802	0,2522	1,3543	1,8378
6	0,4846	0,2404	1,3321	1,9139	0,4950	0,2550	1,3137	1,8212
7	0,5077	0,2484	1,2891	1,8762	0,4978	0,2609	1,2628	1,8188
8	0,5176	0,2630	1,2602	1,8387	0,5134	0,2673	1,2396	1,8122
9	0,5300	0,2789	1,2334	1,8061	0,4980	0,2737	1,3440	1,7964
10	0,5408	0,3000	1,2034	1,7717	0,5086	0,2812	1,2547	1,7994
11	0,5473	0,3222	1,1879	1,7310	0,5050	0,2905	1,2712	1,7803
12	0,5559	0,3351	1,1713	1,6929	0,5622	0,2944	1,1372	1,7857
13	0,5631	0,3543	1,1514	1,6533	0,4732	0,3161	1,3166	1,7596
14	0,5693	0,3673	1,1338	1,6210	0,5829	0,3189	1,1028	1,7451
15	0,5752	0,3784	1,1214	1,5996	0,5706	0,3225	1,1351	1,7738
16	0,5817	0,3855	1,1042	1,5708	0,5273	0,3318	1,2420	1,7838
17	0,5922	0,3990	1,0885	1,5525	0,5561	0,3825	1,1826	1,5995
18	0,5930	0,4097	1,0788	1,5289	0,5053	0,3965	1,3038	1,5674
19	0,5959	0,4187	1,0680	1,5036	0,5745	0,4096	1,1216	1,5439
20	0,5999	0,4267	1,0547	1,4839	0,5748	0,4096	1,1017	1,5556
21	0,6055	0,4351	1,0441	1,4655	0,5795	0,4152	1,1231	1,5386
22	0,6125	0,4438	1,0252	1,4425	0,6004	0,4255	1,0771	1,4951
23	0,6149	0,4492	1,0189	1,4243	0,5357	0,4277	1,1890	1,5072
24	0,6194	0,4573	1,0023	1,4051	0,6055	0,4517	1,0388	1,4030
25	0,6215	0,4649	1,0038	1,3876	0,5578	0,4629	1,1531	1,4028
26	0,6289	0,4691	0,9871	1,3771	0,5656	0,4612	1,1547	1,4201
27	0,6308	0,4778	0,9765	1,3631	0,5611	0,4660	1,1523	1,3506
28	0,6386	0,4855	0,9682	1,3459	0,5910	0,4994	1,0633	1,2822
29	0,6424	0,4891	0,9531	1,3366	0,6108	0,4838	1,0280	1,3538
30	0,6432	0,4937	0,9505	1,3181	0,5645	0,5140	1,1318	1,2614
31	0,6503	0,4943	0,9326	1,3147	0,5695	0,5059	1,1449	1,2569
32	0,6514	0,5017	0,9294	1,2988	0,6164	0,5017	1,0171	1,2667
33	0,6559	0,5077	0,9214	1,2885	0,5932	0,5070	1,0684	1,2722
34	0,6604	0,5110	0,9113	1,2786	0,5918	0,5167	1,0709	1,2268
35	0,6624	0,5153	0,9026	1,2677	0,5670	0,5315	1,1548	1,2037
36	0,6690	0,5145	0,8850	1,2654	0,6236	0,5299	1,0103	1,2239
37	0,6689	0,5195	0,8909	1,2554	0,5589	0,5137	1,2008	1,2485
38	0,6690	0,5246	0,8772	1,2450	0,6164	0,5304	1,0198	1,2037
39	0,6781	0,5290	0,8653	1,2335	0,6270	0,5432	1,0125	1,1694
40	0,6766	0,5315	0,8624	1,2307	0,5963	0,5432	1,0889	1,1877
41	0,6857	0,5341	0,8509	1,2220	0,5924	0,5385	1,1187	1,1990
42	0,6867	0,5383	0,8399	1,2080	0,6186	0,5455	1,0274	1,1672
43	0,6898	0,5402	0,8312	1,2052	0,6152	0,5614	1,0277	1,1374
44	0,6865	0,5424	0,8333	1,1981	0,6097	0,5480	1,0505	1,1615
45	0,6950	0,5447	0,8203	1,1908	0,6169	0,5603	1,0255	1,1347
46	0,6934	0,5482	0,8208	1,1843	0,6102	0,5619	1,0731	1,1355
47	0,6992	0,5538	0,8061	1,1801	0,5516	0,5664	1,2087	1,1327
48	0,7055	0,5533	0,7956	1,1719	0,6270	0,5619	1,0292	1,1537
49	0,7001	0,5508	0,7987	1,1721	0,5954	0,5781	1,0952	1,1110
50	0,7061	0,5575	0,7881	1,1656	0,6080	0,5773	1,0822	1,1104

Epoch	Learning Rate 0,001				Learning Rate 0,0001			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
51	0,7075	0,5611	0,7799	1,1574	0,6482	0,5804	0,9824	1,0958
52	0,7120	0,5623	0,7734	1,1489	0,6292	0,5734	1,0156	1,1093
53	0,7155	0,5653	0,7672	1,1443	0,5614	0,5661	1,2068	1,1118
54	0,7137	0,5650	0,7736	1,1408	0,5882	0,5823	1,1158	1,0938
55	0,7168	0,5698	0,7631	1,1339	0,6459	0,5812	0,9911	1,0804
56	0,7203	0,5743	0,7527	1,1269	0,6420	0,5717	0,9867	1,1289
57	0,7209	0,5743	0,7463	1,1222	0,6272	0,5843	1,0267	1,0785
58	0,7236	0,5771	0,7415	1,1182	0,6175	0,5865	1,0548	1,0766
59	0,7281	0,5773	0,7317	1,1119	0,6459	0,5865	1,0103	1,0846
60	0,7250	0,5796	0,7312	1,1085	0,6384	0,5873	1,0223	1,0826
61	0,7341	0,5800	0,7232	1,1139	0,6476	0,5974	0,9993	1,0663
62	0,7364	0,5849	0,7157	1,1018	0,6604	0,5776	0,9792	1,1196
63	0,7310	0,5863	0,7180	1,0968	0,6334	0,5946	1,0451	1,0738
64	0,7379	0,5856	0,7043	1,0929	0,6378	0,5778	1,0194	1,1209
65	0,7395	0,5858	0,7067	1,0889	0,6300	0,5971	1,0674	1,0735
66	0,7386	0,5875	0,7027	1,0858	0,5438	0,5996	1,2990	1,0586
67	0,7421	0,5899	0,6983	1,0842	0,6138	0,5940	1,0842	1,0716
68	0,7442	0,5918	0,6886	1,0745	0,6317	0,5910	1,0481	1,0677
69	0,7473	0,5920	0,6860	1,0725	0,6359	0,6024	1,0374	1,0497
70	0,7485	0,5970	0,6832	1,0636	0,6551	0,6044	0,9775	1,0643
71	0,7500	0,5968	0,6770	1,0621	0,6244	0,5974	1,0654	1,0500
72	0,7534	0,5991	0,6695	1,0614	0,6445	0,6038	1,0265	1,0357
73	0,7505	0,6011	0,6725	1,0553	0,6537	0,5926	1,0145	1,0697
74	0,7542	0,6000	0,6647	1,0535	0,6431	0,5963	1,0126	1,0426
75	0,7581	0,6028	0,6575	1,0518	0,6384	0,6052	1,0532	1,0353
76	0,7578	0,6042	0,6560	1,0444	0,5910	0,6158	1,1708	1,0383
77	0,7574	0,6052	0,6528	1,0408	0,6627	0,6150	0,9961	1,0182
78	0,7612	0,6079	0,6472	1,0353	0,6083	0,6116	1,1309	1,0207
79	0,7598	0,6092	0,6465	1,0309	0,6540	0,6091	1,0097	1,0235
80	0,7617	0,6088	0,6456	1,0345	0,6387	0,6071	1,0545	1,0276
81	0,7668	0,6110	0,6307	1,0291	0,5845	0,6102	1,2071	1,0237
82	0,7666	0,6148	0,6291	1,0221	0,6529	0,6038	1,0199	1,0403
83	0,7666	0,6134	0,6351	1,0231	0,6540	0,6180	0,9945	1,0100
84	0,7743	0,6138	0,6186	1,0152	0,6574	0,6102	1,0154	1,0458
85	0,7733	0,6195	0,6146	1,0131	0,6099	0,6124	1,1374	1,0146
86	0,7753	0,6192	0,6096	1,0098	0,6373	0,6177	1,0808	1,0101
87	0,7749	0,6180	0,6113	1,0110	0,6236	0,6278	1,0963	1,0045
88	0,7755	0,6210	0,6106	1,0027	0,6518	0,6147	1,0194	1,0150
89	0,7735	0,6236	0,6079	0,9994	0,5751	0,6119	1,2506	1,0191
90	0,7773	0,6228	0,6060	1,0011	0,6599	0,6203	1,0148	1,0089
91	0,7790	0,6234	0,6034	0,9949	0,6323	0,6130	1,0784	1,0128
92	0,7803	0,6284	0,5978	0,9916	0,6027	0,6122	1,1611	1,0309
93	0,7777	0,6251	0,5954	0,9919	0,6535	0,6230	1,0276	0,9990
94	0,7853	0,6289	0,5898	0,9832	0,6200	0,6253	1,1342	0,9874
95	0,7817	0,6295	0,5952	0,9829	0,6175	0,6194	1,1245	1,0094
96	0,7859	0,6309	0,5842	0,9766	0,6537	0,6186	1,0273	1,0151
97	0,7876	0,6330	0,5766	0,9756	0,6604	0,6217	1,0260	1,0000
98	0,7902	0,6340	0,5705	0,9697	0,6638	0,6200	1,0387	1,0095
99	0,7890	0,6374	0,5785	0,9654	0,6607	0,6208	1,0355	1,0017
100	0,7867	0,6340	0,5783	0,9657	0,6177	0,6303	1,1514	1,0021

Lampiran 5: Hasil Evaluasi Lengkap Pengaruh Jumlah *Epochs*

Epoch	Learning Rate 0.001				Learning Rate 0.0001			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
1	0,2275	0,1942	2,0246	2,2993	0,2506	0,2506	1,9029	1,8803
2	0,3221	0,2043	1,7119	2,1534	0,2531	0,2506	2,1163	1,8690
3	0,4029	0,2150	1,5420	2,0706	0,3630	0,2506	1,6617	1,8315
4	0,4436	0,2170	1,4406	2,0226	0,4166	0,2503	1,5061	1,8298
5	0,4749	0,2260	1,3680	1,9709	0,4983	0,2514	1,2941	1,8177
6	0,4969	0,2313	1,3111	1,9338	0,5064	0,2511	1,3058	1,8145
7	0,5113	0,2376	1,2768	1,9023	0,5206	0,2542	1,2470	1,8171
8	0,5242	0,2442	1,2496	1,8672	0,5525	0,2539	1,1863	1,8159
9	0,5305	0,2555	1,2227	1,8349	0,5285	0,2570	1,2634	1,8073
10	0,5445	0,2680	1,2012	1,8085	0,4986	0,2567	1,2412	1,8196
11	0,5490	0,2854	1,1841	1,7692	0,5684	0,2581	1,1439	1,8309
12	0,5556	0,3055	1,1646	1,7328	0,4975	0,2612	1,2779	1,8329
13	0,5618	0,3267	1,1465	1,6955	0,5432	0,2667	1,2032	1,8637
14	0,5698	0,3444	1,1287	1,6558	0,5441	0,2863	1,2051	1,8096
15	0,5747	0,3567	1,1183	1,6300	0,5684	0,3153	1,1357	1,7437
16	0,5838	0,3725	1,0974	1,5971	0,5167	0,3103	1,2324	1,7774
17	0,5880	0,3829	1,0918	1,5786	0,5868	0,3301	1,0898	1,7235
18	0,5912	0,3946	1,0742	1,5526	0,5460	0,3583	1,1675	1,6396
19	0,5978	0,4032	1,0608	1,5292	0,5578	0,3730	1,1686	1,6152
20	0,6025	0,4132	1,0523	1,5078	0,5511	0,3940	1,1909	1,5382
21	0,6039	0,4218	1,0490	1,4937	0,5848	0,4104	1,0669	1,4947
22	0,6155	0,4273	1,0232	1,4729	0,5734	0,4210	1,0937	1,4806
23	0,6164	0,4394	1,0149	1,4482	0,5695	0,4316	1,1198	1,4597
24	0,6220	0,4474	1,0017	1,4324	0,6004	0,4336	1,0582	1,4681
25	0,6239	0,4547	0,9958	1,4070	0,5709	0,4621	1,1216	1,3787
26	0,6280	0,4631	0,9864	1,3906	0,5748	0,4590	1,1118	1,4024
27	0,6329	0,4723	0,9734	1,3750	0,5036	0,4612	1,2808	1,3862
28	0,6404	0,4772	0,9622	1,3565	0,5307	0,4729	1,2154	1,3677
29	0,6401	0,4827	0,9516	1,3474	0,5812	0,4967	1,0879	1,2994
30	0,6449	0,4859	0,9429	1,3345	0,6083	0,4983	1,0179	1,2999
31	0,6482	0,4947	0,9343	1,3180	0,5695	0,5103	1,1196	1,2688
32	0,6516	0,4989	0,9285	1,3073	0,5904	0,5173	1,0991	1,2427
33	0,6554	0,5016	0,9125	1,2988	0,5547	0,5028	1,1914	1,2976
34	0,6628	0,5079	0,9052	1,2837	0,5505	0,5343	1,1860	1,2222
35	0,6619	0,5113	0,9009	1,2775	0,5700	0,5282	1,1414	1,2377
36	0,6667	0,5158	0,8921	1,2724	0,6147	0,5257	1,0404	1,2334
37	0,6710	0,5190	0,8846	1,2579	0,5739	0,5248	1,1264	1,2462
38	0,6734	0,5226	0,8787	1,2525	0,5823	0,5458	1,0800	1,1851
39	0,6780	0,5265	0,8658	1,2377	0,5918	0,5419	1,0734	1,2156
40	0,6788	0,5305	0,8628	1,2370	0,5223	0,5421	1,2863	1,1985
41	0,6825	0,5293	0,8482	1,2285	0,6217	0,5413	1,0302	1,1923
42	0,6837	0,5329	0,8407	1,2210	0,5190	0,5505	1,3017	1,1698
43	0,6884	0,5357	0,8342	1,2117	0,6016	0,5586	1,0674	1,1486
44	0,6920	0,5406	0,8279	1,2055	0,5876	0,5566	1,1162	1,1621
45	0,6916	0,5436	0,8208	1,2008	0,5915	0,5650	1,0954	1,1454
46	0,6973	0,5490	0,8119	1,1929	0,5363	0,5569	1,2612	1,1431
47	0,7006	0,5475	0,8130	1,1877	0,5971	0,5667	1,0677	1,1285
48	0,6998	0,5502	0,8006	1,1791	0,5731	0,5661	1,1255	1,1332
49	0,7071	0,5517	0,7899	1,1719	0,5804	0,5636	1,1257	1,1424
50	0,7084	0,5583	0,7895	1,1646	0,5737	0,5751	1,1216	1,1145

Epoch	Learning Rate 0.001				Learning Rate 0.0001			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
51	0.7080	0.5573	0.7848	1,1633	0.6127	0.5734	1,0433	1,1102
52	0.7114	0.5576	0.7767	1,1578	0.6091	0.5472	1,0772	1,2037
53	0.7164	0.5621	0.7633	1,1517	0.6236	0.5778	1,0181	1,1279
54	0.7153	0.5669	0.7625	1,1451	0.5513	0.5670	1,2004	1,1633
55	0.7187	0.5690	0.7545	1,1421	0.6094	0.5834	1,0826	1,0890
56	0.7207	0.5694	0.7545	1,1335	0.6384	0.5862	1,0009	1,0925
57	0.7250	0.5697	0.7411	1,1309	0.6228	0.5910	1,0464	1,0807
58	0.7256	0.5751	0.7415	1,1237	0.6091	0.5812	1,0689	1,1042
59	0.7274	0.5745	0.7359	1,1200	0.6417	0.5801	0.9950	1,1011
60	0.7318	0.5773	0.7254	1,1174	0.6325	0.5938	1,0281	1,0751
61	0.7345	0.5780	0.7201	1,1123	0.6189	0.5862	1,0651	1,0748
62	0.7340	0.5784	0.7162	1,1131	0.6406	0.5778	1,0121	1,1269
63	0.7379	0.5831	0.7123	1,0984	0.6258	0.5946	1,0628	1,0713
64	0.7390	0.5834	0.7060	1,0993	0.6205	0.5901	1,0790	1,0679
65	0.7392	0.5883	0.7064	1,0923	0.6311	0.5926	1,0191	1,0718
66	0.7375	0.5876	0.6993	1,0928	0.6610	0.5935	0.9784	1,0601
67	0.7439	0.5892	0.6913	1,0814	0.6144	0.5834	1,0949	1,1234
68	0.7439	0.5909	0.6914	1,0767	0.6643	0.5848	0.9752	1,0937
69	0.7460	0.5898	0.6838	1,0838	0.6504	0.5896	0.9927	1,0848
70	0.7504	0.5951	0.6765	1,0675	0.6191	0.6004	1,0837	1,0641
71	0.7517	0.5948	0.6789	1,0689	0.6395	0.6027	1,0270	1,0477
72	0.7545	0.5993	0.6657	1,0669	0.6412	0.5865	1,0289	1,0982
73	0.7542	0.5962	0.6646	1,0654	0.5363	0.6057	1,3407	1,0426
74	0.7551	0.5976	0.6645	1,0634	0.6152	0.6013	1,0929	1,0483
75	0.7547	0.5988	0.6618	1,0543	0.6144	0.6127	1,1281	1,0314
76	0.7587	0.6061	0.6508	1,0504	0.6035	0.5982	1,1482	1,0517
77	0.7571	0.6037	0.6551	1,0517	0.5843	0.5988	1,1612	1,0391
78	0.7593	0.6070	0.6492	1,0436	0.6507	0.6052	1,0235	1,0370
79	0.7612	0.6051	0.6466	1,0393	0.6158	0.6004	1,0957	1,0758
80	0.7657	0.6062	0.6355	1,0397	0.6239	0.6057	1,0929	1,0636
81	0.7676	0.6092	0.6315	1,0345	0.6275	0.6057	1,0865	1,0454
82	0.7692	0.6096	0.6277	1,0304	0.6445	0.6052	1,0263	1,0399
83	0.7675	0.6152	0.6326	1,0225	0.6454	0.6055	1,0223	1,0277
84	0.7672	0.6159	0.6239	1,0260	0.6663	0.6083	0.9943	1,0429
85	0.7679	0.6168	0.6250	1,0162	0.6554	0.6069	1,0069	1,0514
86	0.7715	0.6185	0.6212	1,0121	0.6417	0.6147	1,0490	1,0245
87	0.7723	0.6166	0.6200	1,0098	0.6722	0.6127	0.9737	1,0284
88	0.7775	0.6180	0.6097	1,0097	0.6342	0.6194	1,0494	1,0113
89	0.7757	0.6229	0.6078	1,0000	0.6523	0.6130	1,0300	1,0175
90	0.7769	0.6239	0.6016	0.9955	0.6152	0.6110	1,1254	1,0244
91	0.7811	0.6237	0.5988	0.9965	0.6560	0.6225	1,0247	1,0056
92	0.7837	0.6257	0.5937	0.9954	0.6189	0.6211	1,1367	1,0307
93	0.7850	0.6278	0.5887	0.9914	0.6328	0.6278	1,0829	1,0101
94	0.7845	0.6293	0.5909	0.9869	0.6066	0.6205	1,1459	1,0059
95	0.7883	0.6277	0.5828	0.9826	0.6504	0.6127	1,0353	1,0301
96	0.7845	0.6287	0.5827	0.9844	0.6613	0.6253	1,0358	1,0110
97	0.7860	0.6344	0.5818	0.9714	0.6415	0.6200	1,0553	1,0054
98	0.7866	0.6326	0.5769	0.9751	0.6649	0.6272	1,0179	1,0090
99	0.7923	0.6328	0.5731	0.9715	0.6537	0.6253	1,0223	0.9998
100	0.7923	0.6324	0.5698	0.9711	0.6624	0.6225	1,0071	1,0123
101	0.7918	0.6364	0.5638	0.9626	0.6535	0.6138	1,0471	1,0371
102	0.7886	0.6390	0.5703	0.9625	0.6526	0.6205	1,0518	1,0209

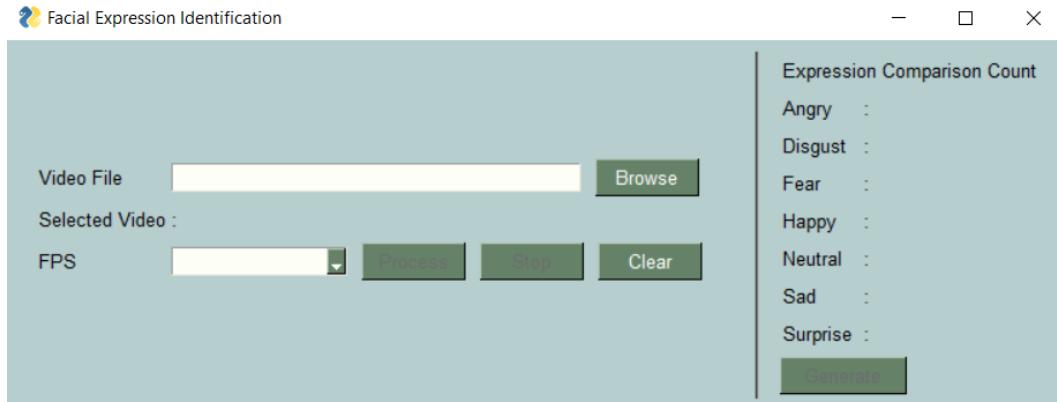
Epoch	Learning Rate 0.001				Learning Rate 0.0001			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
103	0.7908	0.6379	0.5628	0.9687	0.6417	0.6292	1.0731	0.9960
104	0.7910	0.6417	0.5632	0.9598	0.6562	0.6278	1.0380	1.0051
105	0.7938	0.6435	0.5601	0.9553	0.6663	0.6281	0.9944	1.0098
106	0.7990	0.6403	0.5505	0.9537	0.6490	0.6211	1.0603	1.0224
107	0.8002	0.6406	0.5482	0.9523	0.6420	0.6230	1.0644	1.0158
108	0.8004	0.6456	0.5444	0.9375	0.6437	0.6342	1.0804	0.9996
109	0.8010	0.6463	0.5446	0.9409	0.6487	0.6300	1.0504	1.0039
110	0.8003	0.6459	0.5485	0.9388	0.6621	0.6264	1.0318	0.9938
111	0.8027	0.6441	0.5393	0.9382	0.6462	0.6331	1.0768	0.9952
112	0.8011	0.6471	0.5428	0.9336	0.6401	0.6311	1.0959	1.0055
113	0.8032	0.6496	0.5315	0.9306	0.6671	0.6306	1.0348	0.9828
114	0.8022	0.6513	0.5383	0.9290	0.6629	0.6297	1.0394	0.9929
115	0.8039	0.6510	0.5338	0.9260	0.6487	0.6367	1.0472	0.9863
116	0.8060	0.6557	0.5275	0.9185	0.6041	0.6230	1.2300	1.0301
117	0.8067	0.6516	0.5245	0.9261	0.6543	0.6253	1.0799	0.9916
118	0.8064	0.6540	0.5271	0.9177	0.6417	0.6286	1.1041	0.9898
119	0.8130	0.6561	0.5158	0.9135	0.6529	0.6334	1.0766	0.9972
120	0.8094	0.6555	0.5227	0.9122	0.6549	0.6381	1.0637	1.0006
121	0.8095	0.6565	0.5158	0.9100	0.6409	0.6423	1.1570	0.9734
122	0.8125	0.6609	0.5126	0.9080	0.6504	0.6390	1.0788	0.9830
123	0.8167	0.6634	0.5129	0.9003	0.6682	0.6398	1.0455	0.9869
124	0.8136	0.6634	0.5151	0.8962	0.6342	0.6476	1.1364	0.9708
125	0.8138	0.6636	0.5104	0.8938	0.6699	0.6306	1.0244	1.0113
126	0.8149	0.6646	0.5067	0.8947	0.6641	0.6409	1.0363	0.9904
127	0.8203	0.6661	0.4917	0.8885	0.6666	0.6373	1.0530	0.9940
128	0.8204	0.6631	0.4965	0.8930	0.6613	0.6300	1.0620	1.0109
129	0.8212	0.6680	0.4994	0.8864	0.6509	0.6409	1.0940	0.9878
130	0.8194	0.6677	0.4934	0.8837	0.6613	0.6398	1.0595	0.9808
131	0.8179	0.6719	0.4955	0.8738	0.6501	0.6412	1.0923	0.9702
132	0.8212	0.6716	0.4893	0.8803	0.6401	0.6409	1.1343	0.9916
133	0.8210	0.6731	0.4910	0.8736	0.6590	0.6431	1.0926	0.9788
134	0.8242	0.6712	0.4870	0.8706	0.6680	0.6426	1.0562	0.9899
135	0.8205	0.6718	0.4855	0.8670	0.6493	0.6448	1.1067	0.9777
136	0.8245	0.6735	0.4844	0.8669	0.6350	0.6448	1.1473	0.9686
137	0.8237	0.6760	0.4849	0.8630	0.6613	0.6490	1.0521	0.9626
138	0.8264	0.6774	0.4726	0.8610	0.6632	0.6515	1.0540	0.9713
139	0.8241	0.6771	0.4830	0.8574	0.6741	0.6521	1.0568	0.9771
140	0.8256	0.6758	0.4746	0.8587	0.6532	0.6504	1.0931	0.9590
141	0.8243	0.6834	0.4807	0.8557	0.6719	0.6473	1.0445	0.9785
142	0.8324	0.6778	0.4681	0.8525	0.6367	0.6518	1.1693	0.9574
143	0.8301	0.6822	0.4654	0.8490	0.6705	0.6406	1.0698	0.9777
144	0.8305	0.6843	0.4655	0.8447	0.6722	0.6350	1.0715	0.9932
145	0.8323	0.6839	0.4645	0.8424	0.6270	0.6532	1.1727	0.9692
146	0.8311	0.6882	0.4671	0.8363	0.6537	0.6507	1.1110	0.9636
147	0.8319	0.6837	0.4611	0.8433	0.6560	0.6532	1.0745	0.9653
148	0.8338	0.6860	0.4599	0.8369	0.6172	0.6529	1.2396	0.9732
149	0.8319	0.6888	0.4618	0.8308	0.6426	0.6454	1.1391	0.9690
150	0.8326	0.6918	0.4581	0.8314	0.6493	0.6537	1.1389	0.9606
151	0.8334	0.6891	0.4634	0.8287	0.6616	0.6465	1.1004	0.9766
152	0.8345	0.6894	0.4567	0.8234	0.6674	0.6518	1.0773	0.9681
153	0.8350	0.6933	0.4457	0.8166	0.6677	0.6465	1.0886	0.9627
154	0.8392	0.6932	0.4425	0.8227	0.6641	0.6482	1.1258	0.9739

Epoch	Learning Rate 0.001				Learning Rate 0.0001			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
155	0.8393	0.6925	0.4439	0.8199	0.6616	0.6537	1.1168	0.9573
156	0.8350	0.6946	0.4508	0.8135	0.6529	0.6459	1.1545	0.9727
157	0.8394	0.6950	0.4469	0.8165	0.6680	0.6454	1.0837	0.9828
158	0.8365	0.6979	0.4517	0.8069	0.6403	0.6437	1.1617	0.9771
159	0.8401	0.6986	0.4417	0.8049	0.6635	0.6546	1.1195	0.9654
160	0.8408	0.6994	0.4401	0.8032	0.6345	0.6493	1.2168	0.9802
161	0.8436	0.7020	0.4340	0.7966	0.6635	0.6523	1.0928	0.9853
162	0.8411	0.6986	0.4380	0.8000	0.6521	0.6549	1.1600	0.9556
163	0.8396	0.7025	0.4405	0.7944	0.6462	0.6549	1.1545	0.9567
164	0.8464	0.7050	0.4318	0.7906	0.6576	0.6582	1.1344	0.9614
165	0.8421	0.7007	0.4364	0.7992	0.6680	0.6610	1.1107	0.9574
166	0.8427	0.7048	0.4315	0.7902	0.6515	0.6565	1.1636	0.9719
167	0.8448	0.7063	0.4279	0.7858	0.6445	0.6562	1.1732	0.9605
168	0.8432	0.7055	0.4298	0.7832	0.6537	0.6509	1.1568	0.9751
169	0.8471	0.7090	0.4219	0.7834	0.6652	0.6632	1.1047	0.9492
170	0.8450	0.7079	0.4307	0.7762	0.6465	0.6532	1.1737	0.9810
171	0.8439	0.7099	0.4279	0.7718	0.6716	0.6521	1.0875	0.9678
172	0.8486	0.7072	0.4155	0.7797	0.6691	0.6607	1.1312	0.9536
173	0.8492	0.7118	0.4164	0.7724	0.6490	0.6518	1.2003	0.9889
174	0.8464	0.7099	0.4233	0.7720	0.6621	0.6599	1.1103	0.9583
175	0.8492	0.7176	0.4206	0.7591	0.6716	0.6576	1.0952	0.9635
176	0.8484	0.7151	0.4142	0.7648	0.6655	0.6482	1.1283	1.0040
177	0.8484	0.7113	0.4200	0.7660	0.6694	0.6532	1.0993	0.9723
178	0.8487	0.7128	0.4180	0.7599	0.6613	0.6523	1.1215	0.9744
179	0.8517	0.7155	0.4121	0.7638	0.6429	0.6560	1.1679	0.9956
180	0.8497	0.7190	0.4178	0.7509	0.6496	0.6574	1.1496	0.9677
181	0.8539	0.7133	0.4103	0.7577	0.6655	0.6613	1.1205	0.9658
182	0.8495	0.7198	0.4157	0.7462	0.6479	0.6588	1.1604	0.9637
183	0.8503	0.7193	0.4146	0.7501	0.6596	0.6451	1.1537	1.0044
184	0.8535	0.7191	0.4073	0.7474	0.6574	0.6585	1.1365	0.9663
185	0.8535	0.7238	0.4026	0.7399	0.6445	0.6537	1.1923	0.9779
186	0.8534	0.7196	0.4100	0.7452	0.6669	0.6618	1.1076	0.9683
187	0.8568	0.7256	0.3998	0.7377	0.6663	0.6618	1.1216	0.9638
188	0.8556	0.7238	0.4000	0.7360	0.6708	0.6431	1.1240	0.9986
189	0.8525	0.7290	0.4080	0.7298	0.6599	0.6632	1.1582	0.9570
190	0.8574	0.7262	0.3955	0.7364	0.6529	0.6479	1.1688	1.0320
191	0.8579	0.7285	0.3912	0.7286	0.6445	0.6551	1.2029	0.9695
192	0.8585	0.7287	0.3960	0.7244	0.6549	0.6504	1.1751	0.9882
193	0.8569	0.7303	0.3947	0.7275	0.6395	0.6515	1.2395	1.0143
194	0.8573	0.7303	0.3959	0.7149	0.6576	0.6585	1.1418	0.9740
195	0.8579	0.7316	0.3928	0.7188	0.6674	0.6616	1.1286	0.9783
196	0.8588	0.7305	0.3904	0.7253	0.6657	0.6602	1.1317	0.9728
197	0.8604	0.7329	0.3921	0.7149	0.6627	0.6588	1.1313	0.9829
198	0.8606	0.7343	0.3902	0.7148	0.6646	0.6618	1.1455	0.9567
199	0.8596	0.7356	0.3880	0.7096	0.6641	0.6663	1.1601	0.9814
200	0.8627	0.7383	0.3780	0.7058	0.6727	0.6624	1.1489	0.9670
201	0.8579	0.7377	0.3918	0.7090	0.6727	0.6582	1.1342	0.9755
202	0.8639	0.7351	0.3757	0.7112	0.6733	0.6677	1.1255	0.9561
203	0.8626	0.7394	0.3823	0.7009	0.6596	0.6688	1.1535	0.9641
204	0.8610	0.7382	0.3816	0.6990	0.6694	0.6660	1.1499	0.9680
205	0.8593	0.7402	0.3829	0.6937	0.6749	0.6624	1.1391	0.9821
206	0.8620	0.7445	0.3834	0.6931	0.6610	0.6490	1.1835	1.0117

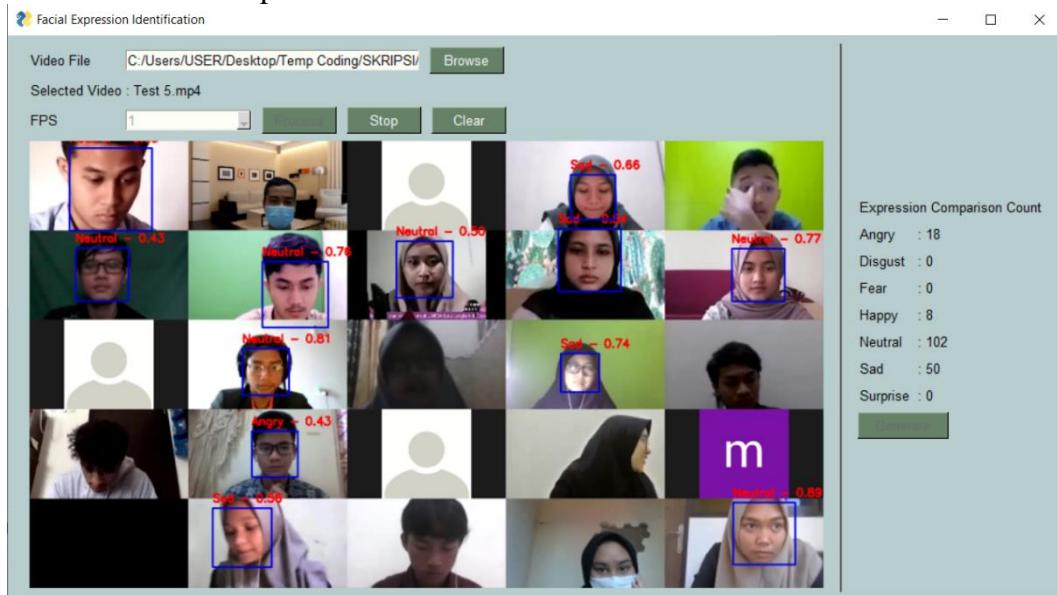
Epoch	Learning Rate 0.001				Learning Rate 0.0001			
	Train Acc	Train Loss	Val Acc	Val Loss	Train Acc	Train Loss	Val Acc	Val Loss
207	0,8636	0,7444	0,3779	0,6947	0,6629	0,6607	1,1318	0,9866
208	0,8624	0,7427	0,3736	0,6970	0,6652	0,6649	1,1594	0,9769
209	0,8644	0,7433	0,3728	0,6862	0,6521	0,6526	1,1917	0,9994
210	0,8692	0,7433	0,3629	0,6892	0,6610	0,6518	1,1825	1,0132
211	0,8647	0,7443	0,3769	0,6880	0,6459	0,6680	1,2352	0,9593
212	0,8642	0,7488	0,3741	0,6775	0,6680	0,6669	1,1887	0,9535
213	0,8675	0,7466	0,3710	0,6782	0,6629	0,6590	1,1493	0,9880
214	0,8670	0,7462	0,3681	0,6830	0,6420	0,6565	1,2248	0,9951
215	0,8657	0,7494	0,3704	0,6723	0,6666	0,6657	1,1613	0,9896
216	0,8695	0,7504	0,3640	0,6702	0,6585	0,6568	1,1768	1,0252
217	0,8643	0,7471	0,3758	0,6755	0,6710	0,6696	1,1513	0,9719
218	0,8680	0,7492	0,3696	0,6714	0,6674	0,6663	1,1410	0,9715
219	0,8683	0,7543	0,3653	0,6627	0,6599	0,6604	1,2003	0,9948
220	0,8723	0,7528	0,3568	0,6653	0,6666	0,6669	1,1503	0,9813
221	0,8685	0,7541	0,3671	0,6662	0,6588	0,6669	1,1811	0,9856
222	0,8712	0,7527	0,3620	0,6611	0,6699	0,6694	1,1488	0,9779
223	0,8677	0,7574	0,3696	0,6529	0,6671	0,6604	1,1556	1,0069
224	0,8702	0,7558	0,3593	0,6567	0,6657	0,6719	1,1657	0,9747
225	0,8676	0,7578	0,3687	0,6509	0,6560	0,6710	1,1707	0,9706
226	0,8710	0,7583	0,3559	0,6510	0,6568	0,6688	1,2284	0,9933
227	0,8681	0,7573	0,3678	0,6528	0,6646	0,6663	1,1458	0,9946
228	0,8724	0,7609	0,3558	0,6458	0,6646	0,6682	1,1687	0,9898
229	0,8711	0,7595	0,3615	0,6499	0,6610	0,6713	1,1729	0,9906
230	0,8687	0,7612	0,3652	0,6501	0,6532	0,6705	1,1980	0,9631
231	0,8708	0,7612	0,3585	0,6421	0,6696	0,6646	1,1731	0,9842
232	0,8717	0,7616	0,3567	0,6367	0,6629	0,6688	1,1817	0,9958
233	0,8726	0,7637	0,3548	0,6368	0,6708	0,6624	1,1660	0,9991
234	0,8754	0,7656	0,3495	0,6352	0,6710	0,6657	1,1728	0,9956
235	0,8746	0,7613	0,3543	0,6345	0,6747	0,6685	1,1569	0,9765
236	0,8695	0,7650	0,3587	0,6353	0,6657	0,6688	1,1966	0,9907
237	0,8765	0,7686	0,3504	0,6234	0,6682	0,6699	1,1863	0,9923
238	0,8760	0,7676	0,3438	0,6264	0,6666	0,6719	1,2142	0,9864
239	0,8742	0,7671	0,3524	0,6288	0,6794	0,6610	1,1646	1,0000
240	0,8721	0,7702	0,3517	0,6216	0,6660	0,6722	1,1736	1,0069
241	0,8742	0,7705	0,3462	0,6178	0,6535	0,6682	1,1874	0,9754
242	0,8770	0,7728	0,3437	0,6160	0,6613	0,6657	1,1793	1,0169
243	0,8784	0,7702	0,3390	0,6216	0,6741	0,6685	1,1945	0,9965
244	0,8765	0,7693	0,3421	0,6156	0,6641	0,6705	1,1890	0,9846
245	0,8746	0,7703	0,3452	0,6162	0,6682	0,6702	1,1756	0,9842
246	0,8743	0,7709	0,3437	0,6154	0,6571	0,6618	1,2445	1,0179
247	0,8763	0,7748	0,3438	0,6117	0,6599	0,6763	1,2273	0,9868
248	0,8767	0,7768	0,3433	0,6034	0,6719	0,6629	1,1880	1,0223
249	0,8742	0,7733	0,3422	0,6107	0,6473	0,6663	1,2907	0,9930
250	0,8805	0,7802	0,3379	0,6021	0,6546	0,6772	1,2563	0,9832

Lampiran 6: Tampilan GUI

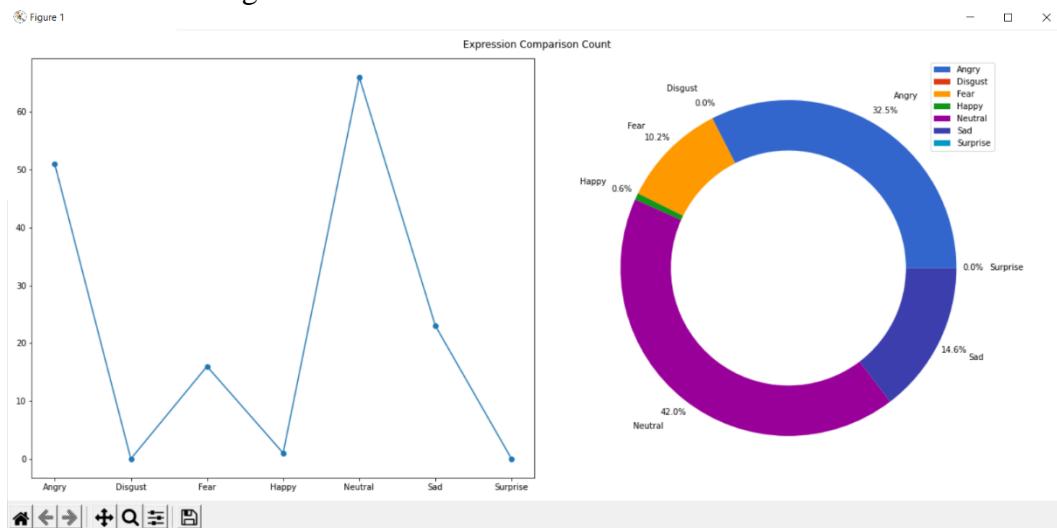
Halaman Utama GUI



Proses Prediksi Ekspresi dari Video



Hasil Prediksi dengan Grafik



Lampiran 7: Kode Python – Pelatihan Model

```

# Import Packages and Libraries
import tensorflow as tf
from keras.layers import Conv2D, BatchNormalization, Activation, MaxPooling2D, Dropout, Flatten, Dense
from keras.models import Model, Sequential
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint
from keras.preprocessing.image import ImageDataGenerator
import pandas as pd
import numpy as np

%matplotlib inline
import matplotlib.pyplot as plt

# Initiation Number of Dropout, Classes, and Learning Rate
nb_dropout = 0.5
nb_classes = 7
nb_learning_rate = 0.0001

# Creating Model Neural Network
model = Sequential()

# 1st - Convolution Layer
model.add(Conv2D(64, (3,3), padding = 'same', input_shape = (48, 48, 1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Dropout(nb_dropout))

# 2nd - Convolution Layer
model.add(Conv2D(128, (5,5), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Dropout(nb_dropout))

# 3rd - Convolution Layer
model.add(Conv2D(512, (3,3), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Dropout(nb_dropout))

# 4th - Convolution Layer
model.add(Conv2D(512, (3,3), padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2, 2)))
model.add(Dropout(nb_dropout))

# Flattening
model.add(Flatten())

# 1st - Fully Connected Layer
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(nb_dropout))

# 2nd - Fully Connected Layer
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(nb_dropout))

# Output as Much as the Number of Classes
model.add(Dense(nb_classes, activation = 'softmax'))

# Summary Model
model.summary()

# Compile and Create Checkpoint
opt = Adam(learning_rate = nb_learning_rate)
model.compile(optimizer = opt, loss = 'categorical_crossentropy', metrics = ['accuracy'])

model_name = "model-facial-expression.h5"
checkpoint = ModelCheckpoint(model_name, monitor = 'val_accuracy', verbose = 1, save_best_only = True, mode = 'max')
callbacks_list = [checkpoint]

# Initiation Variable for ImageDataGenerator
rescale = 1./255.
validation_split = 0.1
batch_size = 128
target_size = (48, 48)
data_dir = "/dataset"

```

```

# Create Data Generator, Split It into Data Training and Validation
datagen = ImageDataGenerator(rescale = rescale, validation_split = validation_split)

train_generator = datagen.flow_from_directory(data_dir,
                                             batch_size = batch_size,
                                             color_mode = 'grayscale',
                                             class_mode = 'categorical',
                                             shuffle = True,
                                             target_size = target_size,
                                             subset = 'training')

validation_generator = datagen.flow_from_directory(data_dir,
                                                 batch_size = batch_size,
                                                 color_mode = 'grayscale',
                                                 class_mode = 'categorical',
                                                 shuffle = False,
                                                 target_size = target_size,
                                                 subset = 'validation')

# Fitting Model
history = model.fit(train_generator,
                     steps_per_epoch = train_generator.n // train_generator.batch_size,
                     epochs = nb_epochs,
                     validation_data = validation_generator,
                     validation_steps = validation_generator.n // validation_generator.batch_size,
                     callbacks = callbacks_list)

# Plotting a Evaluation Model and Saving it as PNG
fig = plt.figure(figsize = (20,10))
plt.subplot(1, 2, 1)
plt.suptitle('Optimizer : Adam', fontsize = 10)
plt.ylabel('Loss', fontsize = 16)
plt.plot(history.history['loss'], label = 'Training Loss')
plt.plot(history.history['val_loss'], label = 'Validation Loss')
plt.legend(loc = 'upper right')

plt.subplot(1, 2, 2)
plt.ylabel('Accuracy', fontsize = 16)
plt.plot(history.history['accuracy'], label = 'Training Accuracy')
plt.plot(history.history['val_accuracy'], label = 'Validation Accuracy')
plt.legend(loc = 'lower right')
plt.show()

fname = 'Evaluation.png'
fig.savefig(fname, transparent = True)

# Create DataFrame of Evaluation Model and Saving it as CSV
columns = ['accuracy', 'loss', 'val_accuracy', 'val_loss']

df = pd.DataFrame()
for column in columns:
    df[column] = ['%.4f' % elem for elem in history.history[column]]

fname = 'Evaluation.csv'
df.to_csv(fname, header = col, index = False)

```

Lampiran 8: Kode Python – Pengujian kepada *Image*

```

# Import Packages and Libraries
import tensorflow as tf
import numpy as np
import cv2
import matplotlib.pyplot as plt

# Initiation Path and Categories of Classes
parent_path = 'test-image/'
categories = ["Angry", "Disgust", "Fear", "Happy", "Neutral", "Sad", "Surprise"]

# Load Face Cascade and Model
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
new_model = tf.keras.models.load_model('model/model-face-detection.h5')

# Function Predict Expression per Face
def predict_expression(face):
    final_image = cv2.resize(face, (48, 48))
    final_image = cv2.cvtColor(final_image, cv2.COLOR_BGR2GRAY)
    final_image = np.expand_dims(final_image, axis = -1)
    final_image = np.expand_dims(final_image, axis = 0)
    final_image = final_image / 255.0

    predictions = new_model.predict(final_image)
    max_percent = "{:.2f}".format(max(predictions[0]))
    class_prediction = np.argmax(predictions)
    result = '{} - {}'.format(categories[class_prediction], max_percent)

    return result

# Function Detect Face from Image
def detect_face(frame):
    results = []
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(frame, 1.1, 4)

    for (x, y, w, h) in faces:
        roi_gray = gray[y : y + h, x : x + w]
        roi_color = frame[y : y + h, x : x + w]
        faces = face_cascade.detectMultiScale(roi_gray)
        if len(faces) == 0:
            continue
        else:
            face = cv2.rectangle(frame, (x, y), (x + w, y + h), (225, 0, 0), 2)
            for (ex, ey, ew, eh) in faces:
                face_roi = roi_color[ey : ey + eh, ex : ex + ew]

            result = predict_expression(face_roi)
            results.append((face_roi, result))
            cv2.putText(face, result, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
                        fontScale = 0.6, color = (0, 0, 255), thickness = 2)

    return results

# Function Predict Image, Counting of Number of Detected Face, and Display the Image
def predict_image(frame):
    results = detect_face(frame)
    plt.figure()
    text = 'Num of Detected as Face = {}'.format(len(results))
    plt.title(text)
    plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))

    if len(results) == 0:
        print("Face not Detected!")
        return None

    for face, result in results:
        plt.figure()
        plt.title(result)
        plt.imshow(cv2.cvtColor(face, cv2.COLOR_BGR2RGB))

    # Testing Image (Happy)
    frame_1 = cv2.imread(parent_path + "happy-boy.jpg")
    predict_image(frame_1)

    # Testing Image (Surprised)
    frame_2 = cv2.imread(parent_path + "surprised-woman.jpg")
    predict_image(frame_2)

    # Testing Image (Angry)
    frame_3 = cv2.imread(parent_path + "angry-man.jpg")
    predict_image(frame_3)

    # Testing Image (Multiple Expression)
    frame_4 = cv2.imread(parent_path + "multiple-expression.jpg")
    predict_image(frame_4)

    # Testing Image (No Face)
    frame_5 = cv2.imread(parent_path + "no-face-image.jpg")
    predict_image(frame_5)

```

Lampiran 9: Kode Python – Pengujian kepada *File* Video

```

# Import Packages and Libraries
import tensorflow as tf
import numpy as np
import time
import cv2
import matplotlib.pyplot as plt

# Initiation Path and Categories of Classes
parent_path = 'test-image/'
categories = ["Angry", "Disgust", "Fear", "Happy", "Neutral", "Sad", "Surprise"]

# Load Face Cascade and Model
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
new_model = tf.keras.models.load_model('model/model-face-detection.h5')

# Function Predict Expression per Face
def predict_expression(face):
    final_image = cv2.resize(face, (48, 48))
    final_image = cv2.cvtColor(final_image, cv2.COLOR_BGR2GRAY)
    final_image = np.expand_dims(final_image, axis = -1)
    final_image = np.expand_dims(final_image, axis = 0)
    final_image = final_image / 255.0

    predictions = new_model.predict(final_image)
    max_percent = "{:.2f}".format(max(predictions[0]))
    class_prediction = np.argmax(predictions)
    counts[categories[class_prediction]] += 1
    result = '{} - {}'.format(categories[class_prediction], max_percent)

    return result

# Function Predict Image and Detect Faces From Image
def predict_image(frame):
    results = []
    try:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(frame, 1.1, 4)

        for (x, y, w, h) in faces:
            roi_gray = gray[y : y + h, x : x + w]
            roi_color = frame[y : y + h, x : x + w]
            facess = face_cascade.detectMultiScale(roi_gray)
            if len(facess) == 0:
                continue
            else:
                frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (225, 0, 0), 2)
                for (ex, ey, ew, eh) in facess:
                    face_roi = roi_color[ey : ey + eh, ex : ex + ew]

                    result = predict_expression(face_roi)
                    results.append((face_roi, result))
                    frame = cv2.putText(frame, result, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
                                        fontScale = 0.6, color = (0, 0, 255), thickness = 2)
    except:
        pass
    return frame, results

# Initiation Video Path and Video Capture
file_name = "test-video/Online Class Video.mp4"
cap = cv2.VideoCapture(file_name)

# Open the Video Capture and Predict All Frame
frame_rate = 1
prev = 0

while(cap.isOpened()):
    time_elapsed = time.time() - prev
    ret, frame = cap.read()

    if ret:
        if time_elapsed > 1./frame_rate:
            prev = time.time()
            try:
                frame, results = predict_image(frame)
                cv2.imshow('frame', frame)

            except:
                pass

        else:
            break

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

```

Lampiran 10: Kode Python – Pembuatan GUI menggunakan PySimpleGUI

```

# Import Packages and Libraries
import tensorflow as tf
import PySimpleGUI as sg
import numpy as np
import os.path
import time
import cv2

import matplotlib
import matplotlib.pyplot as plt
matplotlib.use('TkAgg')

# Initiation Path and Categories of Classes
parent_path = 'test-image/'
categories = ["Angry", "Disgust", "Fear", "Happy", "Neutral", "Sad", "Surprise"]

# Load Face Cascade and Model
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
new_model = tf.keras.models.load_model('model/model-face-detection.h5')

# Function Predict Expression per Face
def predict_expression(face):
    final_image = cv2.resize(face, (48, 48))
    final_image = cv2.cvtColor(final_image, cv2.COLOR_BGR2GRAY)
    final_image = np.expand_dims(final_image, axis = -1)
    final_image = np.expand_dims(final_image, axis = 0)
    final_image = final_image / 255.0

    predictions = new_model.predict(final_image)
    max_percent = "{:.2f}".format(max(predictions[0]))
    class_prediction = np.argmax(predictions)
    counts[categories[class_prediction]] += 1
    result = '{} - {}.'.format(categories[class_prediction], max_percent)

    return result

# Function Predict Image and Detect Faces From Image
def predict_image(frame):
    results = []
    try:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(frame, 1.1, 4)

        for (x, y, w, h) in faces:
            roi_gray = gray[y : y + h, x : x + w]
            roi_color = frame[y : y + h, x : x + w]
            faces = face_cascade.detectMultiScale(roi_gray)
            if len(faces) == 0:
                continue
            else:
                frame = cv2.rectangle(frame, (x, y), (x + w, y + h), (225, 0, 0), 2)
                for (ex, ey, ew, eh) in faces:
                    face_roi = roi_color[ey : ey + eh, ex : ex + ew]

                    result = predict_expression(face_roi)
                    results.append((face_roi, result))
                    frame = cv2.putText(frame, result, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
                                        fontScale = 0.6, color = (0, 0, 255), thickness = 2)
    except:
        pass

    return frame, results

#Create Template GUI with PySimpleGUI
def template():
    video = [
        [
            sg.Text("Video File", size=(10, 1)),
            sg.In(size=(40, 1), enable_events = True, key="-VIDEO FILE-"),
            sg.FileBrowse(size=(8, 1), file_types=(("Video Files", "*.mp4"),))
        ],
        [
            sg.Text(size=(60, 1), key="-VIDEO NAME-")
        ],
        [
            sg.Text("FPS", size=(10, 1)),
            sg.Combo([1, 2, 3, 4, 5], size=(15, 1),
                    enable_events = True, key="-FPS-", disabled = True),
            sg.Button("Process", size=(8, 1), key="-PROCESS-", disabled = True),
            sg.Button("Stop", size=(8, 1), key="-STOP-", disabled = True),
            sg.Button("Clear", size=(8, 1), key="-CLEAR-", disabled = True)
        ],
        [
            sg.Image(filename = "", key="-IMAGE-")
        ]
    ]

```

```

graph = [
    [sg.Text("Expression Comparison Count", justification='center')],
    [sg.Text("Angry\t:", size=(20, 1), key="-EXP ANGRY-")],
    [sg.Text("Disgust\t:", size=(20, 1), key="-EXP DISGUST-")],
    [sg.Text("Fear\t:", size=(20, 1), key="-EXP FEAR-")],
    [sg.Text("Happy\t:", size=(20, 1), key="-EXP HAPPY-")],
    [sg.Text("Neutral\t:", size=(20, 1), key="-EXP NEUTRAL-")],
    [sg.Text("Sad\t:", size=(20, 1), key="-EXP SAD-")],
    [sg.Text("Surprise\t:", size=(20, 1), key="-EXP SURPRISE-")],
    [sg.Button("Generate", size=(10, 1), key="-GENERATE-", disabled = True)]
]

layout = [
    [
        sg.Column(video),
        sg.VSeparator(),
        sg.Column(graph)
    ]
]

return layout

# Create Event GUI Interaction and Launch It
def main():
    sg.theme("LightGreen")

    layout = template()
    window = sg.Window("Facial Expression Identification", layout,
                        resizable = True, finalize = True)
    cap = None
    prev = 0

    while True:
        event, values = window.read(timeout=20)
        if event == "Exit" or event == sg.WIN_CLOSED:
            break

        elif event == "-VIDEO FILE-":
            file = values["-VIDEO FILE-"]
            fnames = 'Selected Video : {}'.format(os.path.basename(file))
            window["-VIDEO NAME-"].update(fnames)
            window["-FPS-"].update(disabled = False)
            window["-CLEAR-"].update(disabled = False)

        elif event == "-FPS-":
            window["-PROCESS-"].update(disabled = False)

        elif event == "-PROCESS-":
            file_name = values["-VIDEO FILE-"]
            cap = cv2.VideoCapture(file_name)
            window["-FPS-"].update(disabled = True)
            window["-PROCESS-"].update(disabled = True)
            window["-STOP-"].update(disabled = False)

        elif event == "-STOP-":
            window["-STOP-"].update(disabled = True)
            window["-GENERATE-"].update(disabled = False)
            cap = None

        elif event == "-GENERATE-":
            colors = ['#3366CC', '#DC3912', '#FF9900', '#109618', '#990099', '#3B3EAC', '#0099C6']

            # Plotting a Evaluation of Video Detection
            fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 8))
            fig.suptitle('Expression Comparison Count')
            ax1.plot(counts.keys(), counts.values(), 'o-')
            ax2.pie(counts.values(),
                     labels = counts.keys(),
                     colors = colors,
                     autopct = '%.1f%%',
                     startangle=0,
                     pctdistance=1.1,
                     labeldistance=1.2,
                     wedgeprops=dict(width=0.3))

            plt.tight_layout()
            plt.legend()
            plt.show()

        elif event == "-CLEAR-":
            window["-VIDEO FILE-"].update('')
            window["-VIDEO NAME-"].update('')
            window["-FPS-"].update('', disabled = True)
            window["-PROCESS-"].update(disabled = True)
            window["-STOP-"].update(disabled = True)
            window["-IMAGE-"].update(data = None)
            window["-GENERATE-"].update(disabled = True)
            cap = None
    
```

```

if cap:
    frame_rate = values["-FPS-"]
    time_elapsed = time.time() - prev
    ret, frame = cap.read()

    if ret:
        if time_elapsed > 1./ frame_rate:
            prev = time.time()
            try:
                frame, results = predict_image(frame)

                window["-EXP ANGRY-"].update("Angry\t: {}".format(counts['Angry']))
                window["-EXP DISGUST-"].update("Disgust\t: {}".format(counts['Disgust']))
                window["-EXP FEAR-"].update("Fear\t: {}".format(counts['Fear']))
                window["-EXP HAPPY-"].update("Happy\t: {}".format(counts['Happy']))
                window["-EXP NEUTRAL-"].update("Neutral\t: {}".format(counts['Neutral']))
                window["-EXP SAD-"].update("Sad\t: {}".format(counts['Sad']))
                window["-EXP SURPRISE-"].update("Surprise\t: {}".format(counts['Surprise']))

                scale_percent = 60 # percent of original size
                width = int(frame.shape[1] * scale_percent / 100)
                height = int(frame.shape[0] * scale_percent / 100)
                dim = (width, height)

                # resize image
                resized = cv2.resize(frame, dim, interpolation = cv2.INTER_AREA)

                imgbytes = cv2.imencode(".png", resized)[1].tobytes()
                window["-IMAGE-"].update(data = imgbytes)

            except:
                pass

        else:
            cap = None
            window["-STOP-"].update(disabled = True)
            window["-GENERATE-"].update(disabled = False)

    window.close()

# Run the GUI
main()

```

Lampiran 11: Surat Izin Penelitian



Jakarta, 04 Maret 2021

No : 002/PRODI/IK/III/2021
Hal : Permohonan Penelitian

Kepada Yth.
Ibu Leli Deswindi., S.E., M.T
Ketua STIMIK ESQ
Kampus Menara 165, Lantai 18 & 19,
Jl. Tb. Simatupang Kav. 1, Cilandak Timur, Pasar Minggu, RT.3/RW.3,
Jakarta 12560

Assalamu'alaikum Wr.Wb

Dengan hormat,

Sehubungan dengan mata kuliah Skripsi yang diwajibkan bagi mahasiswa tingkat akhir Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer (STIMIK ESQ) program studi **Ilmu Komputer** tahun akademik 2020/2021, bersama surat ini kami bermaksud mengajukan permohonan kegiatan **penelitian** untuk mahasiswa berikut:

No	NIM	Nama	Topik Penelitian	Pengumpulan Data
1	1710130001	Abdul Fattah Kusnandar	PENGEMBANGAN ALTERNATIF DESAIN INTERAKSI AIS MOBILE STIMIK ESQ	Observasi sistem dan wawancara
2	1710130005	Iwan Sinanto Ate	SELEKSI PENERIMAAN BEASISWA GENERASI EMAS MENGGUNAKAN METODE SIMPLE ADDITIVE WEIGHTING (SAW) DAN ANALYTICAL HIERARCHY PROCESS (AHP) PADA STMIK ESQ	Studi dokumen dan wawancara
3	1710130007	M Fu'ad Zikri	IMPLEMENTASI WEB SERVICE UNTUK INTEGRASI FITUR LMS DAN AIS STIMIK ESQ	Observasi sistem dan wawancara
4	1710130008	Mita Nurul Yatimah	KOMPARASI ALGORITMA KLASIFIKASI DATA MINING UNTUK PREDIKSI KELULUSAN MAHASISWA TEPAT WAKTU DI STIMIK ESQ	Studi dokumen dan wawancara
5	1710130010	Mu'tashim Billah	IDENTIFIKASI EKSPRESI WAJAH PESERTA KELAS ONLINE MENGGUNAKAN TENSORFLOW DAN CONVOLUTIONAL NEURAL NETWORK PADA PLATFORM ONLINE MEETING	Studi dokumen
6	1710130011	Risnawati	EVALUASI USABILITY PADA E-LEARNING MANAGEMENT SYSTEM STIMIK ESQ MENGGUNAKAN SYSTEM USABILITY SCALE (SUS)	Kuesioner dan wawancara

MENARA 165 Lt. 19

Jl. TB. Simatupang Kav. 1, Cilandak - Jakarta Selatan - 12560, Indonesia.
T. +62-21-2940-6999, 2940-7000 | F. +62-21-7883-4165 | E.info@esqbs.ac.id
www.esqbs.ac.id



Besar harapan kami, Ibu dapat memberikan dukungan dan kesempatan kepada mahasiswa tersebut. Demikian surat permohonan ini kami sampaikan, atas perhatian dan kerjasamanya kami ucapan terima kasih.

Wassalamu'alaikum Wr.Wb

Hormat Kami,

Ahlijati Nuraminah, S.Kom., M.T.I.

Kepala Program Studi Ilmu Komputer
STIMIK ESQ

MENARA 165 Lt. 19

Jl. TB. Simatupang Kav. 1, Cilandak - Jakarta Selatan - 12560, Indonesia.
T. +62-21-2940-6999, 2940-7000 | F. +62-21-7883-4165 | E.info@esqbs.ac.id
www.esqbs.ac.id

Lampiran 12: Kartu Bimbingan Skripsi

 STIMIK - ESQ STIMIK ESPRESSO SAINS DAN KONSEP SEMARANG	BERITA ACARA PEMBIMBINGAN SKRIPSI <i>Minutes of Meeting – Thesis Advisement</i> Semester Genap Tahun Akademik 2020/2021	No Dok : F-A.4.03.07 Rev : 0 Hal : 1
---	---	--

NIM : 1710130010
Student ID
 Nama : Mu'tashim Billah
Name
 Program Studi : Ilmu Komputer
Program Study
 Judul Skripsi : IDENTIFIKASI EKSPRESI WAJAH PESERTA KELAS ONLINE
Title of Thesis/Final Project
 MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DENGAN
 TENSORFLOW PADA PLATFORM ONLINE MEETING

Tanggal Mulai : 06 Maret 2021
 Bimbingan

Tanggal Selesai : 29 Mei 2021
 Bimbingan

Pembimbing 1 : Andika Sundawijaya, S.Kom, M. Kom
Advisor 1

Pembimbing 2 :
Advisor 2

PEMBIMBINGAN:

**). dalam mengisi kegiatan pembimbingan, mohon diisi hanya yang memiliki progress sesuai dengan target.*

Tanggal/ Pertemuan	Kegiatan Pembimbingan *)		Paraf Pembimbing
1 <hr/> 06 Mar 2021	Topik Bahasan	1. Pembahasan mengenai data wajah yang digunakan 2. Pembahasan mengenai tensorflow dan penggunaannya 3. Pembahasan mengenai proposal revisi 4. Pembahasan mengenai rumus dan hitungan manual CNN	Telah diperiksa dosen pembimbing 1
	Saran Perbaikan (KPI/Target)	1. Memulai code metode CNN dengan Tensorflow terlebih dahulu agar tahu komputasi dan kemampuan perangkat 2. Merevisi beberapa hal yang diubah dalam proposal sebelumnya	
	Catatan Target (KPI):	Penyelesaian : 10%	

 STIMIK - ESQ STIMULUS INSTITUTE OF MANAGEMENT & SCIENCE	BERITA ACARA PEMBIMBINGAN SKRIPSI <i>Minutes of Meeting – Thesis Advisement</i> Semester Genap Tahun Akademik 2020/2021	No Dok : F-A.4.03.07 Rev : 0 Hal : 2
--	---	--

Tanggal/ Pertemuan	Kegiatan Pembimbingan *)		Paraf Pembimbing
2 <hr/> 20 Mar 2021	Topik Bahasan	1. Menjelaskan program yang sudah kerjakan selama kurun waktu kurang lebih 2 minggu meliputi - Pembuatan model tensorflow berserta hyperparameternya (Hasil: dengan epoch 3 didapatkan akurasi yang masih rendah yaitu 0.58) - Pengujian model terhadap data citra secara langsung (Hasil: 3 dari 5 pengujian sukses) - Pengujian model terhadap data video realtime menggunakan webcam (Hasil: program dapat berjalan dan memprediksi namun belum sepenuhnya tepat dalam melakukan prediksi ekspresi) 2. Menjelaskan kesulitan dalam penulisan kode dan pembuatan model yaitu - Membutuhkan spesifikasi yang lebih tinggi dibanding resource yang dimiliki - Membutuhkan waktu training yang relatif lama (1 epoch sekitar 40-60 menit, sedangkan proses training membutuhkan epoch sebanyak 25)	Telah diperiksa dosen pembimbing 1
	Saran Perbaikan	1. Melanjutkan penulisan program (sementara dengan model yang ada) dan menerapkannya dengan membaca data dari file (rekaman video) 2. Saat ini masih terpikirkan untuk menyewa/meminjam komputer lain untuk mempermudah dan mempercepat proses training data	
	Catatan Target (KPI):	Penyelesaian : 15%	
3 <hr/> 27 Mar 2021	Topik Bahasan	1. Hasil training model dengan 5, 15, 30 epoch 2. Pelatihan model perlu dilakukan di tempat lain karena resources pribadi dapat memakan waktu yang sangat lama 3. Penyesuaian proposal dengan judul baru	Telah diperiksa dosen pembimbing 1
	Saran Perbaikan	1. Hasil training sudah berhasil namun hasil yang di keluarkan belum maksimal, dari 7 ekspresi masih 2-3 ekspresi saja yang berhasil di klasifikasi, perlu adanya layer tambahan dan eksperimen lagi ke model tensorflownya 2. Pelatihan model dilakukan menggunakan google colab agar lebih ringan	

 <p>BERITA ACARA PEMBIMBINGAN SKRIPSI <i>Minutes of Meeting – Thesis Advisement</i></p> <p>Semester Genap Tahun Akademik 2020/2021</p>	No Dok : F-A.4.03.07 Rev : 0 Hal : 3
---	--

Tanggal/ Pertemuan	Kegiatan Pembimbingan *)		Paraf Pembimbing
		3. Penyesuaian proposal dengan judul baru dan penambahan bab 2 yang dimulai dari umum s/d khusus minggu depan diselesaikan	
	Catatan Target (KPI):	Penyelesaian : 30%	
4 <hr/> 03 Apr 2021	Topik Bahasan	1. Penyesuaian proposal skripsi khususnya bab 2 yang penjabaran teorinya lebih mendetail (dari teori umum ke khusus) 2. Pengujian program dengan menambahkan beberapa layer tambahan dan penggunaan dropout serta image generator 3. Pengujian program dengan read file video dengan video simulasi pembelajaran online 4. Data video kelas online sudah diterima dari bagian BAA dan IT ESQ	Telah diperiksa dosen pembimbing 1
	Saran Perbaikan	1. Mulai merancang model dan program agar overfitting tidak terlalu besar 2. Mulai menerapkan layer tambahan seperti dropout dan image generator	
	Catatan Target (KPI):	Penyelesaian : 40%	
5 <hr/> 10 Apr 2021	Topik Bahasan	Pembuatan model berulang kali dengan 30 - 100 epoch dengan rincian <ul style="list-style-type: none"> 1. Menggunakan Pre-trained Model (Transfer Learning) MobileNetV1 2. Menggunakan Pre-trained Model (Transfer Learning) MobileNetV2 3. Menggunakan CNN 5 Lapis Hasil yang didapatkan ketiga modelnya overfitting dengan akurasi training 85-90% akurasi testing 55-65% Masih mencari model terbaik agar hasil akurasi meningkat dan loss berkurang 	Telah diperiksa dosen pembimbing 1
	Saran Perbaikan	1. Menggunakan layer dropout 2. Menggunakan augmentation 3. Dicoba kembali dengan menambah-kurangkan layer yang ada	
	Catatan Target (KPI):	Penyelesaian : 55%	

 <p>BERITA ACARA PEMBIMBINGAN SKRIPSI <i>Minutes of Meeting – Thesis Advisement</i></p> <p>Semester Genap Tahun Akademik 2020/2021</p>	No Dok : F-A.4.03.07 Rev : 0 Hal : 4
--	--

Tanggal/ Pertemuan	Kegiatan Pembimbingan *)			Paraf Pembimbing	
6 <hr/> 24 Apr 2021	Topik Bahasan <ul style="list-style-type: none"> 1. Codingan backend sudah selesai (model, multiple detection, grafik) 2. Ada penambahan teori bab 2 3. Arsitektur Tensorflownya perlu di cantumkan 4. Pembuatan GUI dengan python 			Telah diperiksa dosen pembimbing 1	
	Saran Perbaikan <ul style="list-style-type: none"> 1. Perlu perbaikan arsitektur dan perbandingan arsitektur tensorflow 2. Melanjutkan penulisan bagian-bagian yang kurang 				
	Catatan Target (KPI): Penyelesaian : 60%				
7 <hr/> 01 Mei 2021	Topik Bahasan <ul style="list-style-type: none"> 1. Visualisasi Model Tensorflow (Ada kendala untuk menampilkan visualisasi model yang advance, untuk yang standar sudah bisa akan di attach gambarnya, untuk yang advance masih mencari tools terbaik) 2. Pembuatan GUI 3. Interaksi GUI seperti - Browse Video File - Setting FPS - Proses Identifikasi Ekspresi Wajah - Interaktif penjumlahan dan total Ekspresi Wajah yang ditangkap - Generate Grafik (Pie dan Plot) 			Telah diperiksa dosen pembimbing 1	
	Saran Perbaikan <ul style="list-style-type: none"> 1. Masih ada beberapa fungsi di GUI yang belum dibuat untuk menghandle beberapa kondisi dan perlu pengujian lebih lanjut 				
	Catatan Target (KPI): Penyelesaian : 70%				
8 <hr/> 08 Mei 2021	Topik Bahasan <ul style="list-style-type: none"> 1. Revisi penulisan Proposal Skripsi dengan format Skripsi 2. Menyesuaikan template Skripsi 3. Pengecekan Typo dan Plagiarism 4. Penyesuaian Interaksi GUI 			Telah diperiksa dosen pembimbing 1	
	Saran Perbaikan <ul style="list-style-type: none"> 1. Perlu ada beberapa tambahan konfigurasi di GUI agar semakin lebih nyaman untuk digunakan user 				
	Catatan Target (KPI): Penyelesaian : 80%				
9	Topik Bahasan <ul style="list-style-type: none"> 1. Penulisan Laporan Bab 4 2. Penulisan Lampiran 3. Program GUI 			Telah diperiksa dosen pembimbing 1	

 STIMIK - ESQ <small>Sekolah Tinggi Ilmu Komunikasi & Seni Pertunjukan</small>	BERITA ACARA PEMBIMBINGAN SKRIPSI <i>Minutes of Meeting – Thesis Advisement</i> Semester Genap Tahun Akademik 2020/2021	No Dok : F-A.4.03.07 Rev : 0 Hal : 5
--	---	--

Tanggal/ Pertemuan	Kegiatan Pembimbingan *)		Paraf Pembimbing
21 Mei 2021	Saran Perbaikan	1. Masih perlu menyelesaikan beberapa bagian Bab 4 yang belum terisi dengan gambar-gambar referensi 2. Target penyelesaian Bab 4 dan bab 5 adalah minggu depan sebelum bimbingan terakhir 3. Penulisan serba-serbi (kata pengantar, abstrak, dsb) bisa dilakukan di akhir	
	Catatan Target (KPI):	Penyelesaian : 90%	
10 29 Mei 2021	Topik Bahasan	1. Pengujian model dengan 2, 3, 4 convolution layer 2. Pengujian model dengan 0.25 dan 0.5 dropout layer 3. Pengujian model dengan 0.001 dan 0.0001 learning rate 4. Pengujian model dengan 250 epochs 5. Penulisan bab 4 6. Penulisan beberapa tambahan lampiran	 Telah diperiksa dosen pembimbing 1
	Saran Perbaikan	1. Melanjuti bab 4 tahap akhir 2. Penulisan bab 5 penyelesaian maksimal minggu depan 3. Pendaftaran sidang saat tulisan sudah selesai 100% 4. Revisi beberapa penulisan yang masih kurang	
	Catatan Target (KPI):	Penyelesaian : 100%	

Saya menyatakan bahwa skripsi dengan judul IDENTIFIKASI EKSPRESI WAJAH PESERTA KELAS ONLINE MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK DENGAN TENSORFLOW PADA PLATFORM ONLINE MEETING atas nama Sdr. Mu'tashim Billah layak untuk disidangkan.

Jakarta, 01 Juni 2021

Pembimbing 1



NIDN : 03-1303-8701

Nama : Andika Sundawijaya, S.Kom, M. Kom

Lampiran 13: Lembar Pernyataan Perbaikan Hasil Sidang Skripsi

 STIMIK-ESQ STIMULUS INOVASI DAN KINERJA DALAM KEGIATAN	LEMBAR PERNYATAAN PERBAIKAN HASIL SIDANG SKRIPSI Semester Ganjil/Genap *) Tahun Akademik 2020/2021	No Dok : F-SPMI Rev : 0
---	---	----------------------------

Saya yang bertanda tangan dibawah ini :

Nama : Mu'tashim Billah
NIM : 1710130010
Program Studi : Ilmu Komputer
Judul Skripsi : Identifikasi Ekspresi Wajah Peserta Kelas Online Menggunakan *Convolutional Neural Network* dengan *Tensorflow* pada Platform *Online Meeting*
Tanggal Sidang : 22 Juni 2021

Menyatakan dengan sebenarnya, bahwa saya telah melakukan perbaikan hasil sidang dan telah diperiksa dan disetujui oleh :

Nama Pembimbing dan Team Penguji	Persetujuan Perbaikan Hasil Sidang (Tanggal & Tanda tangan)
Dosen Pembimbing Skripsi 1 Nama & NIDN : Andika Sundawijaya, S.T., M.Kom / NIDN : 03-1303-8701	
Ketua Penguji Nama & NIDN : Ahlijati Nuraminah, S.Kom, M.T.I / NIDN : 03-1712-8404	 30/6/21
Anggota Penguji -1 : Nama & NIDN : Erza Sofian S.Kom., M.Sc / NIDN : 03-2207-7002	
Anggota Penguji-2 : Nama & NIDN : Desy Komalasari., S.Kom, M.Kom / NIDN : 03-2212-9202	

Jakarta, 28 Juni 2021

Yang membuat pernyataan,



Mu'tashim Billah

*) Coret yang tidak perlu

Lampiran 14: Lembar Penilaian Penguji Sidang

 <p>STIMIK-ESQ SCHOOL OF COMPUTER & INFORMATION SYSTEM</p>	<p>LEMBAR PERBAIKAN PENGUJI SIDANG</p> <p>Semester Ganjil/Genap *) Tahun Akademik 2020 / 2021</p>	<p>No Dok : F-SPMI-1-D.03.15 Rev : 0</p> <p>Hal : 1</p>
--	--	---

Identitas Mahasiswa

Nama : Mu'tashim Billah
 NIM : 1710130010
 Program Studi : Ilmu Komputer
 Pembimbing 1 : Andika Sundawijaya, S.T., M.Kom / NIDN : 03-1303-8701
 Pembimbing 2 : --
 Judul Skripsi : Identifikasi Ekspresi Wajah Peserta Kelas *Online* Menggunakan *Convolutional Neural Network* dengan Tensorflow pada *Platform Online Meeting*

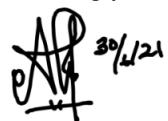
LEMBAR PENGUJIAN/PERTANYAAN

1. Apakah kesimpulan menjawab pertanyaan dan tujuan penelitian?
2. Berapa banyak video training? Berapa video data testing? apakah jumlah video testing mempengaruhi?
3. Apa syarat video yang digunakan utk training dan testing? dapat dimasukkan ke bagian saran
4. Metode mana yang dipilih dan apa alasannya? masukkan ke dalam sub bab 2.3
5. Apa kendala terbesar dalam penelitian ini

*LEMBAR CATATAN / REVISI YANG HARUS DILAKUKAN

1. Tambahkan penjelasan metode yang dipilih dalam penelitian dan alasannya
2. Setiap gambar seharusnya ada penjelasannya. contoh gambar kerangka pemikiran
3. Jarak antar judul bab dengan tulisan di bawahnya 3 spasi
4. Perbaikan penulisan (istilah asing, kata baku, penggunaan kata depan) langsung di dokumen
5. Lampiran kartu bimbingan dan surat izin dibuat full page

Jakarta, 22 Juni 2021
Ketua Penguji Sidang,



30/6/21

(Ahlijati Nuraminah, S.Kom, M.T.I / NIDN : 03-1712-8404)

**Diberikan kepada peserta sidang
Berkas dilampirkan bersama dengan penyerahan hardcopy skripsi*

 <p>STIMIK - ESQ SCHOOL OF COMPUTER SCIENCE & INFORMATION SYSTEM</p>	<p>LEMBAR PERBAIKAN PENGUJI SIDANG</p> <p>Semester Ganjil/Genap *) Tahun Akademik 2020 / 2021</p>	No Dok : F-SPMI-1-D.03.15 Rev : 0 Hal : 1
--	---	---

Identitas Mahasiswa

Nama : Mu'tashim Billah
NIM : 1710130010
Program Studi : Ilmu Komputer
Pembimbing 1 : Andika Sundawijaya, S.T., M.Kom / NIDN : 03-1303-8701
Pembimbing 2 : -
Judul Skripsi : Identifikasi Ekspresi Wajah Peserta Kelas *Online* Menggunakan *Convolutional Neural Network* dengan *Tensorflow* pada Platform *Online Meeting*

LEMBAR PENGUJIAN/PERTANYAAN

- Yang memotivasi kamu dalam memilih topik pada penelitian kamu
- Bagaimana Proses penambahan jumlah filtering pada convolution layernya
- Penerapan Augmentation data untuk menambah keakuratan dalam prediksi citra
- Bagaimana proses untuk membedakan rentang nilai yang signifikan pada citra masukan pada Normalisasi layer ?

***LEMBAR CATATAN / REVISI YANG HARUS DILAKUKAN**

Penambahan saran untuk penelitian berikutnya, untuk di masukan event yang spesial yang akan dapat masukan data citra dari objek target.
Menambahkan referensi untuk dataset train dan validation pada table.
Dapat dikembangkan untuk penelitian selanjutnya

Jakarta, 22 Juni 2021
Penguji Sidang 1,



(Erza Sofian S.Kom., M.Sc / NIDN : 03-2207-7002)

**Diberikan kepada peserta sidang*
Berkas dilampirkan bersama dengan penyerahan hardcopy skripsi

 <p>STIMIK - ESQ Sekolah Tinggi Ilmu Komputer & Sosial Komunitas</p>	<p>LEMBAR PERBAIKAN PENGUJI SIDANG</p> <p>Semester Ganjil/Genap *) Tahun Akademik 2020 / 2021</p>	<p>No Dok : F-SPMI-1-D.03.15 Rev : 0</p> <p>Hal : 1</p>
--	--	---

Identitas Mahasiswa

Nama	: Mu'tashim Billah
NIM	: 1710130010
Program Studi	: Ilmu Komputer
Pembimbing 1	: Andika Sundawijaya, S.T., M.Kom / NIDN : 03-1303-8701
Pembimbing 2	: -
Judul Skripsi	: Identifikasi Ekspresi Wajah Peserta Kelas Online Menggunakan Convolutional Neural Network Dengan Tensorflow Pada Platform Online Meeting

LEMBAR PENGUJIAN/PERTANYAAN

1. Apa sebenarnya tujuan utama mengidentifikasi wajah secara real time pada saat pembelajaran daring.
2. Bagaimana proses dari video menjadi citra yang nantinya akan diproses oleh CNN juga tidak terlihat.
3. Lalu bagaimana pra proses data, komposisi data, dll?
4. Kenapa 7 emoji? Interpretasinya apa?
5. Skenario percobaan? Dan tujuannya?
6. Apa arti dari angka2 tabel hasil pendekatan?

***LEMBAR CATATAN / REVISI YANG HARUS DILAKUKAN**

1. Penulisan Bahasa tasing yang tidak dimiringkan.
2. Apabila Bahasa asing masih bisa diartikan dalam Bahasa Indonesia, sebaiknya menggunakan Bahasa Indonesia saja.
3. Cek kembali apakah (titik) pada angka itu penulisan dalam format Bahasa Indonesia, Contoh: Sekolah-sekolah (?)
4. Paragraf satu dengan dua terbalik, hal ini dikarenakan pada abstrak topik utama yang diangkat dari isu covid yang menyebabkan pembelajaran harus daring
5. Tidak ada sinkronisasi antara wajah dengan covid pada pendahuluan di paragraf 1 dan 2.
6. Pembahasan tentang ekspresi sangat kurang.
7. Pendekan emosi yang digunakan juga tidak ada.
8. Alur penelitian tidak terlihat, harusnya breakdown secara jelas dari mulai input-proses-output. Selain itu baiknya dijelaskan dalam bentuk grafik bagan.
9. Bagaimana proses dari video menjadi citra yang nantinya akan diproses oleh CNN juga tidak terlihat.
10. Proses2 deep learning CNN harusnya diletakkan di bab 3.
11. hanya mendeskripsikan hasil yang diperoleh dari training dan testing deep learning.
12. Hanya mendeskripsikan hasil yang diperoleh dari training dan testing deep learning.

 <p>STIMIK - ESQ SCHOOL OF MANAGEMENT & LAW KOMASARI</p>	<p>LEMBAR PERBAIKAN PENGUJI SIDANG</p> <p>Semester Ganjil/Genap *) Tahun Akademik 2020 / 2021</p>	<p>No Dok : F-SPMI-1-D.03.15 Rev : 0</p> <p>Hal : 2</p>
--	--	---

Jakarta, 22 Juni 2021
Penguji Sidang 2,



(Desy Komalasari., S.Kom, M.Kom / NIDN : 03-2212-9202)

**Diberikan kepada peserta sidang
Berkas dilampirkan bersama dengan penyerahan hardcopy skripsi*

Lampiran 15: Daftar Riwayat Hidup

Mu'tashim Billah

Kp. Karang Congok RT. 001/001, Karang Satria, Tambun Utara,

Kabupaten Bekasi

082134888583 (WA) / 081317714843

mutashimb7@gmail.com

<https://www.linkedin.com/in/mu-tashim-billah-733283187>



*I am very interested in developing myself with deeper expertise in **Web Programming** and **Data Science**. In web programming I am experienced in creating **Web-Based Applications** using the **Laravel** (PHP) and **VueJs** (JavaScript) collaboration frameworks. In data science I am experienced in creating **Deep Learning** models using **Tensorflow** and **Keras** for image classification and recognition. Passion for work and discipline in time and really likes challenges. I also have a passion for **Machine Learning**. Have a strong desire to learn something new in the field of **Programming** and **Project Implementation**. Very good at working **individually** and in **teams**, both **offline** and **online***

PERSONAL DETAILS

Place & Date of Birth : Bekasi, 07 November 1998
Religion : Islam

EDUCATION

- SD Negri 01 Karang Satria 2005 - 2010
- SMP Daarul Qur'an International 2010 - 2013
- SMA Daarul Qur'an International. Science major 2013 - 2016
- STIMIK ESQ. Major in Computer Science 2017 - Now
 - Average GPA **3.83** (in scale of 4, from 1st semester to 7th semester)
 - **Highest GPA** (GPA: 4.00) 1st and 7th semester

PERSONAL CERTIFICATION

- Oracle Database Design 2020
- Oracle Database Programming 2020
- Oracle Java Fondation 2020
- Oracle Java Programming 2020
- SanberCode Python – Data Science 2020
- Google IT Automation Professional Certificate 2021
- Google IT Support Professional Certificate 2021
- Mathematics for Machine Learning 2021
- DeepLearning.AI TensorFlow Developer 2021
- Structuring Machine Learning Projects 2021
- TensorFlow: Data and Deployment 2021

WORK EXPERIENCE

- Internship as a programmer at PT Bank Syariah Mandiri 2020
Build applications Internal Control Dashboard and Advanced Monitoring List using Laravel 8.9 and design the database management system

LEADERSHIP EXPERIENCE

- Chairman of OSDAQU Daarul Qur'an Islamic Boarding School 2015 - 2016
Manage and supervise team members and students to carry out daily activities in the Islamic boarding school
- BEM (Badan Eksekutif Mahasiswa) ESQ Business School 2018 - 2019
Minister of Social and Relations.
Providing human resource development services in organizations and carrying out social activities to spread benefits among students
- The Head Organizer Of ESQ Activities Cares About Children's Education 2019
Manage volunteers in conducting social teaching activities for underprivileged children in the area of south Jakarta
- Chairman Of The Daarul Quran Alumni Market 2019
Managing members in conditioning alumni meeting activities which are held once a year

PROJECT EXPERIENCE

- Savings application for Nurul Hidayah Tours and Travel, a web-based application for recording savings balances and paying off departures using Native PHP
Role in the project: carry out the entire project analysis, design, development and testing
- Website Perkemi (Persaudaraan Shorinji Kempo Indonesia), a web-based application for displaying profiles and information on Perkemi using Native PHP
Role in the project: carry out the entire project analysis, design, development and testing
- Website Alumni ESQ, a web-based application to display profiles and information about ESQ alumni including administrator settings using Native PHP
Role in the project: design, development and testing
- Lunch Habits, classification of the lunch habits of ESQ students using the Naive Bayes Classification and Java
Role in the project: development and testing
- Website Research and Community Service, a web-based application to display profiles and information about Website Research and Community Service in ESQ including administrator settings using Native PHP and Waterfall methodology.
Role in the project: analysis, back-end development and testing
- Mobile Application Research and Community Service, a mobile-application to display profiles and information about Website Research and Community Service in ESQ including administrator settings using Java and Waterfall methodology.
Role in the project: analysis, back-end development and testing
- Prediction Model for Hajj Departure Packages, analyze the performance of the Decision Tree Classification class and K-Nearest Neighbor Classification in predicting Umrah departure packages using Python
Role in the project: carry out the entire project analysis, design, development and testing

- E-Commerce Larashop, a web-based application for conducting online transactions including administrator and store settings using VueJs as front-end framework, Laravel as back-end framework and the Scrum Agile methodology.
Role in the project: analysis, design, front-end developer, and testing
- Voice Signal Recognition, recognizing sound signals to recognize "OK" commands using Fast Fourier Transform, Artificial Neural Networks Back Propagation and Python
Role in the project: development and testing
- Learning Japanese, a web-based application to practice Japanese language skills with Hiragana, Katakana and Personal Dictionary features using VueJs
Role in the project: carry out the entire project analysis, design, development and testing
- Internal Control Dashboard and Advanced Monitoring List PT. Bank Mandiri Syariah using Laravel
Role in the project: database design, development and testing
- Application of Collaborative Filtering, PCA and K-Means in the Development of the Recommendation System for Ongoing and Upcoming Japanese Animation Films
Role in the project: development and testing

ACHIEVEMENTS

- Sanad Al-Jazariyah 2016
Sanad Tajweed Al-Jazariyah Certification
- Daarul Qur'an National Tahfidz Graduation 2016
Certification of Al-Qur'an Memorization
- Generasi Emas Scholarship 2017
Scholarship for Bachelor of Computer Science for four years

PUBLICATIONS

- SENAMIKA 2021
Application of Collaborative Filtering, Principal Component Analysis and K-Means in the Development of the Recommendation System for Ongoing and Upcoming Japanese Animation Films
- SENAMIKA 2021
Application of Collaborative Filtering, Principal Component Analysis and K-Means in the Development of a Film Recommendation System

SEMINAR, TRAINING AND WORKSHOP

- Training Basic ESQ. September 2017, as participant.
- 3D Animation & Augmented Reality. July 2018, as speaker.
- ITB Cloud Computing Bootcamp (AWS). October 2018, as participant.
- Training ESQ QX. March 2018, as participant.
- Training ESQ MCB. September 2018, as participant.
- Training ESQ SSC. February 2019, as participant.
- Training ESQ Total Action. October 2019, as participant.
- Training Data Mining. Mei 2020, as participant.
- SanberCode Data Science Bootcamp. June 2020, as participant.
- Bangkit 2021 – Machine Learning Path. Feb 2021, as participant.

SKILLS

Computer and Programming Skill:

- Programming language: C, C++, C#, Java, Python
- Script language: PHP, CSS, JavaScript, jQuery, AJAX, XML
- Web Framework: Bootstrap, Laravel, VueJs
- Data Science Framework: Tensorflow, Keras, ScikitLearn
- Database Management System: MySQL
- Modelling Notation: Entity Relationship Diagram (ERD), Data Flow Diagram (DFD), Unified Modelling Language (UML)
- Development Tools: Anaconda 3, VS Code, NetBeans
- Graphical Design Tools: Adobe Photoshop CS, Adobe Illustrator
- Operating System: Microsoft Windows 10
- Microsoft Product: MS Office
- Content Management System: WordPress
- SDLC and Methodology: Waterfall, Scrum

Language Skill:

Native Indonesian, Passive in English & Arabic (both oral and written) and Japanese (N5)

Soft Skill:

Critical thinking, logical thinking, public speaking, leadership, prioritize the process, time management, hard worker, teamwork and collaboration.

PERSONAL INTEREST

Interested in counting, analyzing, memorizing, and working in a conducive and communicative environment. And interested in things that can help the work of others by utilizing technology applications