



PROYEK AKHIR – VE 190834

SISTEM ABSENSI NEW NORMAL MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK UNTUK DETEKSI MASKER

**Muhammad Shalahuddin Al Ayyubi
NRP 10311710000005**

**Dosen Pembimbing:
Joko Priambodo, S.T.M.T.
Ir. Joko Susila, MT.**

**DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

---Halaman ini sengaja dikosongkan---



PROYEK AKHIR - VE 190834

**SISTEM ABSENSI NEW NORMAL MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK UNTUK
DETEKSI MASKER**

MUHAMMAD SHALAHUDDIN AL AYYUBI
NRP 10311710000005

Dosen Pembimbing:
Joko Priambodo, S.T.M.T.
Ir. Joko Susila, MT.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2021

---Halaman ini sengaja dikosongkan----

HALAMAN PENGESAHAN

SISTEM ABSENSI NEW NORMAL MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK UNTUK DETEKSI MASKER

PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Sains Terapan
Pada
Program Studi Teknologi Rekayasa Otomasi
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui,
Dosen Pembimbing

Pembimbing 1,

Pembimbing 2,

Joko Priambodo, S.T.,M.T
NPP.1992202011021

Ir. Joko Susila, MT.
NPP. 196606061991021001

SURABAYA

JULI, 2021

PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini penulis menyatakan bahwa isi sebagian maupun keseluruhan buku Proyek Akhir dengan judul “**Sistem Absensi New Normal Menggunakan Convolutional Neural Network Untuk Masker**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang penulis akui sebagai karya sendiri.

Seluruh data hasil pengujian yang ditulis benar-benar asli tanpa penambahan atau pengurangan dan mampu dipertanggungjawabkan. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka dan telah terbukti validitasnya.

Apabila ternyata pernyataan ini tidak benar, penulis bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 9 Juli 2021

Muhammad Shalahuddin

Al Ayyubi

10311710000005

SISTEM ABSENSI NEW NORMAL MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK UNTUK DETEKSI MASKER

Nama Mahasiswa : Muhammad Shalahuddin Al Ayyubi
NRP : 10311710000005
Dosen Pembimbing 1 : Joko Priambodo, S.T.M.T.
NPP : 1985201912087
Dosen Pembimbing 2 : Ir. Joko Susila, MT.
NPP : 196606061991021001

ABSTRAK

Pada masa pandemi absensi manual, amano, dan sidik jari tidak lagi relevan dikarenakan salah satu cara penyebaran virus Covid-19 adalah melalui permukaan yang terkontaminasi. Penelitian ini bertujuan untuk membangun sebuah sistem absensi yang mengimplementasikan protokol kesehatan dengan catatan fokus yang diteliti lebih dalam adalah peforma sistem deteksi masker yang mempengaruhi lama waktu proses absensi. Fitur yang digunakan dalam sistem absensi ini meliputi deteksi masker, deteksi suhu tubuh, idetifikasi wajah. Metode yang digunakan meliputi *Infrared Sensor*, *Convolutional Neural Network* (CNN), haarcascade. Pengujian sistem akan berfokus pada sistem deteksi masker dimana pada peforma deteksi masker akan membandingkan peforma dari 4 arsitektur yang meliputi MobileNetV2, NASNetMobile, InceptionV3, dan VGG16. Dihasilkan arsitektur MobileNetV2 memiliki nilai rata- rata *f1-score* dan *K-Fold Cross Validation* tertinggi yaitu 99,75% pada uji coba menggunakan *dataset training*. Pada uji coba menggunakan *dataset* percobaan arsitektur MobileNetV2 mendapat nilai tertinggi dari arsitektur lainnya dengan rincian rata - rata FPS sebesar 10,17 dan *f1-score* sebesar 93,6%. Dengan keseluruhan sistem, proses absensi memakan waktu rata rata 25,11 detik.

Kata Kunci : Absensi, COVID-19, Protokol Kesehatan, Deteksi masker, *Convolution Neural Network*(CNN)

----Halaman ini sengaja dikosongkan-----

NEW NORMAL ABSENCE SYSTEM USING CONVOLUTIONAL NEURAL NETWORK METHOD FOR MASK DETECTION

Student Name : Muhammad Shalahuddin Al Ayyubi
ID Number : 10311710000005
Supervisor 1 : Joko Priambodo, S.T. M.T.
NPP : 1985201912087
Supervisor 2 : Ir. Joko Susila, MT.
NPP : 196606061991021001

ABSTRACT

During the pandemic, manual attendance, amano, and fingerprints are no longer relevant because one of the ways the Covid-19 virus spreads is through contaminated surfaces. This study aims to build an attendance system that implements health protocols with a note that the focus that is studied more deeply is the performance of the mask detection system that affects the length of time the attendance process. Features used in this attendance system include mask detection, body temperature detection, face identification. The methods used include Infrared Sensor, Convolutional Neural Network (CNN), haarcascade. System testing will focus on the mask detection system where the mask detection performance will compare the performance of 4 architectures which include MobileNetV2, NASNetMobile, InceptionV3, and VGG16. The result is that the MobileNetV2 architecture has the highest average f1-score and K-Fold Cross Validation, which is 99.75% in trials using the training dataset. In the trial using the experimental dataset, the MobileNetV2 architecture got the highest score from other architectures with details of an average FPS of 10.17 and an f1-score of 93.6%. With the whole system, the attendance process takes an average of 25.11 seconds.

Keywords : Attedance, COVID-19, health protocol, Facemask detection, Convolution Neural Network(CNN)

---Halaman ini sengaja dikosongkan---

KATA PENGANTAR

Puji syukur kami panjatkan kepada Allah SWT atas limpahan rahmat dan hidayah dari-Nya, sehingga kami dapat menyelesaikan Proyek Akhir ini dengan baik, begitu pula pembuatan buku Proyek Akhir ini.

Proyek Akhir ini dilakukan untuk memenuhi satuan kredit semester (SKS) yang harus ditempuh sebagai persyaratan akademis di Departemen Teknik Elektro Otomasi Fakultas Vokasi Institut Sepuluh Nopember Surabaya untuk menyelesaikan program pendidikan Diploma di Teknik Elektro dengan judul :

“Sistem Absensi New Normal Menggunakan Convolutional Neural Network Untuk Deteksi Masker”

Dengan terselesainya Proyek Akhir ini Penulis mengucapkan terima kasih yang sebesar – besarnya kepada :

1. Orang tua dan kakak – kakak yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti.
2. Bapak Imam Arifin selaku Kepala Departemen Teknik Elektro Otomasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya.
3. Bapak Joko Priambodo, S.ST., M.T selaku dosen pembimbing 1 yang telah meluangkan waktu, pikiran, serta memberikan pengarahan dan bimbingan dalam proses penyelesaian Proyek Akhir ini.
4. Bapak Ir. Joko Susila, MT. selaku dosen pembimbing 2 yang telah banyak memberikan masukan ilmu, waktu dan pengarahan dalam penyusunan Proyek Akhir ini.
5. Bapak Siswanto selaku pembimbing industri yang telah meluangkan waktu, pikiran serta memberikan pengarahan dan bimbingan ilmu industri dalam penyelesaian Proyek Akhir ini.

6. Teman – teman D4 Teknik Rekayasa Otomasi Angkatan 2017 yang selalu memberikan doa, bantuan, semangat, dan dukungannya.
7. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhir kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juli 2021

Muhammad Shalahuddin Al Ayyubi

DAFTAR ISI

HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN PROYEK AKHIR	v
ABSTRAK.....	vi
<i>ABSTRACT</i>	viii
KATA PENGANTAR	x
DAFTAR ISI	xii
DAFTAR GAMBAR	xvi
DAFTAR TABEL	xviii
BAB I PENDAHULUAN	21
1.1 Latar Belakang	21
1.2 Tujuan dan Manfaat	22
1.3 Rumusan Masalah	22
1.4 Batasan Masalah.....	23
1.5 Sistematika Penulisan.....	23
BAB II TINJAUAN PUSTAKA	25
2.1 Tinjauan Teori	25
2.2 Dasar Teori	28
2.2.1 Covid-19	28
2.2.2 Protokol Kesehatan	29
2.2.3 Citra Digital	30
2.2.3.1Citra Biner	31
2.2.3.2Citra <i>Grayscale</i>	32
2.2.3.3Citra Bewarna	32
2.2.4 <i>Convolutional Neural Network</i> (CNN)	33
2.2.4.1 Arsitektur Convolution Neural Network.....	40

2.2.4.2 <i>Pooling Layer</i>	37
2.2.4.3 <i>Rectified Linear Unit (ReLU)</i>	38
2.2.4.4 <i>Epoch, Batch Size, Dan Learning Rate</i>	40
2.2.4.5 <i>Confusion Matrix</i>	41
2.2.4.6 <i>K-Fold Cross Validation</i>	43
2.2.5 OpenCV	44
2.2.6 MLX90614.....	45
2.2.7 <i>Encoding dan Decoding</i>	45
2.2.7.1 <i>Encoder base64</i>	46
2.2.7.2 <i>Decoder base64</i>	47
2.2.7.3 <i>Bitmap</i>	48
2.2.8 Blob.....	49
BAB III DESAIN SISTEM DAN IMPLEMENTASI	50
3.1 Arsitektur Sistem Absensi New Normal	50
3.2 Diagram Alir Proses Absensi New Normal	51
3.3 Registrasi Mandiri.....	54
3.3.1 Aplikasi Registrasi Mandiri	54
3.3.2 <i>Base64 Encoder</i> Aplikasi Registrasi Mandiri	55
3.4 Protokol Kesehatan	55
3.4.1 Temperatur Tubuh	56
3.4.2 Deteksi Masker	58
3.4.2.1 <i>Convolusi Neural Network</i> Arsitektur <i>MobileNetV2</i> ...	60
3.4.2.2 <i>Convolusi Neural Network</i> Arsitektur <i>NasNet</i>	62
3.4.2.3 <i>Convolusi Neural Network</i> Arsitektur <i>InceptionV3</i>	63
3.4.2.4 <i>Convolusi Neural Network</i> Arsitektur <i>VGG16</i>	64
3.5 Dataflow Sistem Absensi New Normal	65

BAB IV PENGUJIAN DAN ANALISIS	66
4.1 Pengujian Sistem Absensi New Normal	66
4.1.1 Uji Coba Aplikasi Registrasi Mandiri.....	66
4.1.1.1 Uji Coba Input Data Diri Pada Registrasi Mandiri	66
4.2 Uji Coba Sistem Deteksi Masker	68
4.2.1 <i>Dataset</i> Uji Coba Sistem Deteksi Masker.....	69
4.2.2 Peforma Deteksi Masker Arsitektur MobileNetV2.....	70
4.2.2.1 Peforma MobileNetV2 Pada <i>Dataset Training</i>	71
4.2.2.2 <i>K-Fold Cross Validation</i> MobileNetV2 Pada <i>Data Training</i>	73
4.2.2.3 Peforma MobileNetV2 Pada <i>Dataset Percobaan</i>	73
4.2.2.4 Uji Coba Pengaruh Pencahayaan Terhadap Sistem Deteksi Masker Arsitektur MobileNetV2	75
4.2.2.5 Uji Coba Pengaruh Resolusi Terhadap <i>Frame Per Second</i> Arsitektur MobileNetV2.....	76
4.2.3 Peforma Deteksi Masker Arsitektur NASNetMobile.....	77
4.2.3.1 Peforma MobileNetV2 Menggunakan <i>Dataset Training</i>	77
4.2.3.2 <i>K-Fold Cross Validation</i> NASNetMobile Pada <i>Data Training</i>	79
4.2.3.3 Peforma NASNetMobile Menggunakan <i>Dataset Percobaan</i>	80
4.2.3.4 Uji Coba Pengaruh Pencahayaan Terhadap Sistem Deteksi Masker Arsitektur NASNetMobile	81
4.2.3.5 Uji Coba Pengaruh Resolusi Terhadap <i>Frame Per Second</i> Arsitektur NASNetMobile.....	82
4.2.4 Peforma Deteksi Masker Arsitektur InceptionV3	83

4.2.4.1 Peforma InceptionV3 Menggunakan <i>Dataset Training</i>	83
4.2.4.2 <i>K-Fold Cross Validation</i> InceptionV3 Pada Data <i>Training</i>	85
4.2.4.3 Peforma InceptionV3 Menggunakan <i>Dataset Percobaan</i>	86
4.2.4.4 Uji Coba Pengaruh Pencahayaan Terhadap Sistem Deteksi Masker Arsitektur InceptionV3	87
4.2.4.5 Uji Coba Pengaruh Resolusi Terhadap <i>Frame Per Second</i> (FPS) Arsitektur InceptionV3	88
4.2.5 Peforma Deteksi Masker Arsitektur VGG16	89
4.2.5.1 Peforma VGG16 Menggunakan <i>Dataset Training</i>	89
4.2.5.2 <i>K-Fold Cross Validation</i> VGG16 Pada Data <i>Training</i>	91
4.2.5.3 Peforma VGG16 Menggunakan <i>Dataset Percobaan</i>	92
4.2.5.4 Uji Coba Pengaruh Pencahayaan Terhadap Sistem Deteksi Masker Arsitektur VGG16	93
4.2.5.5 Uji Coba Pengaruh Resolusi Terhadap <i>Frame Per Second</i> (FPS) Arsitektur VGG16	94
4.3 Peforma Keseluruhan Sistem Absensi New Normal	95
4.4 Evaluasi Sistem Deteksi Masker	96
BAB V PENUTUP	98
5.1 Kesimpulan	98
5.2 Saran	99
DAFTAR PUSTAKA	100
LAMPIRAN	103

DAFTAR GAMBAR

Gambar 2.1 Citra Fungsi Dua Variable	30
Gambar 2.2 Elemen Citra atau Pixel.....	31
Gambar 2.3 <i>Grayscale Pixel</i>	32
Gambar 2.4 <i>Single Layer Feedforward Network</i>	34
Gambar 2.5 <i>Multi – Layer Feedforward Network</i>	34
Gambar 2. 6 Proses Operasi Konvolusi	37
Gambar 2.7 <i>Average Pooling</i> dengan <i>Stride 2</i>	38
Gambar 2.8 <i>Max Pooling</i> dengan <i>Stride 2</i>	38
Gambar 2.9 Rectified Linear Unit(ReLU)	39
Gambar 2.10 Arsitektur VGG16.....	41
Gambar 2.11 <i>Confusion Matrix</i>	42
Gambar 2.12 <i>K-Fold Cross Validation</i>	43
Gambar 2.13 MLX90614.....	45
Gambar 2.14 List Base64	46
Gambar 2.15 Bitmap dan Vektor	48
Gambar 3.1 Blok Diagram Sistem Absensi New Normal.....	50
Gambar 3.2 Arsitektur 2 Sistem Absensi New Normal	51
Gambar 3.3 Diagram Alir Proses Absensi New Normal	52
Gambar 3.4 UML Aplikasi Registrasi Mandiri	54
Gambar 3.5 Wiring Sistem Protokol Kesehatan Temperature Tubuh. 56	
Gambar 3.6 Diagram Alir <i>Infrared</i> Sensor	57
Gambar 3.7 Diagram Alir Pembuatan Deteksi Masker pada Absensi 58	
Gambar 3.8 Proses Pembuatan Model CNN Deteksi Masker.....	59
Gambar 3.9 Arsitektur <i>MobileNetV2</i>	60
Gambar 3.10 Depthwise dan Pointwise Conv pada <i>MobileNetV2</i>	61
Gambar 3.11 <i>Sequential MobileNetV2</i>	61
Gambar 3.12 Arsitektur NasNet	62
Gambar 3.13 Arsitektur InceptionV3	63
Gambar 3. 14 Arsitektur VGG16.....	64
Gambar 3.15 Dataflow Sistem Absensi New Normal	65

Gambar 4.1 Sampel Dokumentasi Input Data Diri Pada Aplikasi Registrasi Mandiri	67
Gambar 4.2 Hasil Pengiriman Data Diri dari Aplikasi Ke Database	68
Gambar 4.3 Grafik <i>Training Loss</i> dan <i>Accuracy</i> MobileNetV2 Epoch 100	71
Gambar 4.4 Nilai TN, FP, FN dan TP Arsitektur MobileNetV2 Epoch 100	71
Gambar 4.5 Grafik <i>Training Loss</i> dan <i>Accuracy</i> MobileNetV2 Epoch 30	72
Gambar 4. 6 Nilai TN, FP, FN dan TP Arsitektur MobileNetV2 Epoch 30	72
Gambar 4.7 Grafik <i>Training Loss</i> dan <i>Accuracy</i> NASNetMobile Epoch 100	77
Gambar 4.8 Nilai TN, FP, FN dan TP Arsitektur NASNetMobile Epoch 100	78
Gambar 4.9 Grafik <i>Training Loss</i> dan <i>Accuracy</i> NASNetMobile Epoch 30	78
Gambar 4.10 Nilai TN, FP, FN dan TP Arsitektur NASNetMobile Epoch 30	79
Gambar 4.11 Grafik <i>Training Loss</i> dan <i>Accuracy</i> InceptionV3 Epoch 100	83
Gambar 4.12 Nilai TN, FP, FN dan TP Arsitektur InceptionV3 Epoch 100	84
Gambar 4.13 Nilai TN, FP, FN dan TP Arsitektur InceptionV3 Epoch 30	84
Gambar 4.14 Nilai TN, FP, FN dan TP Arsitektur InceptionV3 Epoch 30	85
Gambar 4.15 Grafik <i>Training Loss</i> dan <i>Accuracy</i> VGG16 Epoch 10089	
Gambar 4.16 Nilai TN, FP, FN dan TP Arsitektur VGG16 Epoch 100	90
Gambar 4.17 Gambar 4.18 Grafik <i>Training Loss</i> dan <i>Accuracy</i> VGG16 Epoch 30	90
Gambar 4.19 Nilai TN, FP, FN dan TP Arsitektur VGG16 Epoch 30	91

DAFTAR TABEL

Tabel 4.1 Uji Coba Pengiriman Data Diri Ke Database.....	67
Tabel 4.2 Spesifikasi <i>Hardware</i>	68
Tabel 4.3 Confusion Matrix Deteksi Penggunaan Masker	69
Tabel 4.4 <i>Dataset Training</i>	70
Tabel 4.5 <i>Dataset Percobaan</i>	70
Tabel 4.6 Peforma Arsitektur MobileNetV2 <i>Epoch</i> 30 dan <i>Epoch</i> 100 menggunakan <i>Dataset Training</i>	73
Tabel 4.7 <i>K-Fold Cross Validation</i> MobileNetV2 <i>Epoch</i> 30 dan 50..	73
Tabel 4.8 Rincian Uji Coba Deteksi Masker Arsitektur MobileNetV2	74
Tabel 4.9 Peforma Deteksi Masker Arsitektur MobileNetV2 Menggunakan <i>Dataset Percobaan</i>	75
Tabel 4.10 Pengaruh Pencahayaan Terhadap Sistem Deteksi Penggunaan Masker Arsitektur MobileNetv2.....	75
Tabel 4.11 Rincian Pengaruh Resolusi Terhadap <i>Frame Per Second</i> Deteksi Penggunaan Masker Arsitektur MobileNetV2.....	76
Tabel 4.12 Peforma Arsitektur NASNetMobile <i>Epoch</i> 30 dan <i>Epoch</i> 100 menggunakan <i>Dataset Training</i>	79
Tabel 4.13 <i>K-Fold Cross Validation</i> NASNetMobile <i>Epoch</i> 30 dan 50	79
Tabel 4.14 Rincian Uji Coba Peforma Deteksi Masker Arsitektur NASNetMobile	80
Tabel 4.15 Peforma Deteksi Masker Arsitektur NASNetMobile Menggunakan <i>Dataset Percobaan</i>	81
Tabel 4.16 Pengaruh Pencahayaan Terhadap Sistem Deteksi Penggunaan Masker Arsitektur NASNetMobile.....	81
Tabel 4.17 Rincian Pengaruh Resolusi Terhadap <i>Frame Per Second</i> Deteksi Penggunaan Masker Arsitektur NASNetMobile	82
Tabel 4.18 Peforma Arsitektur InceptionV3 <i>Epoch</i> 30 dan <i>Epoch</i> 100 menggunakan <i>Dataset Training</i>	85
Tabel 4.19 <i>K-Fold Cross Validation</i> InceptionV3 <i>Epoch</i> 30 dan 50..	85
Tabel 4.20 Peforma Deteksi Masker Arsitektur InceptionV3.....	86

Tabel 4.21 Peforma Deteksi Masker Arsitektur InceptionV3 Menggunakan <i>Dataset</i> Percobaan.....	87
Tabel 4.22 Pengaruh Pencahayaan Terhadap Sistem Deteksi Penggunaan Masker Arsitektur InceptionV3	87
Tabel 4.23 Rincian Pengaruh Resolusi Terhadap <i>Frame Per Second</i> Deteksi Penggunaan Masker Arsitektur InceptionV3	88
Tabel 4.24 Peforma Arsitektur VGG16 <i>Epoch</i> 30 dan <i>Epoch</i> 100 menggunakan <i>Dataset Training</i>	91
Tabel 4.25 <i>K-Fold Cross Validation</i> VGG16 <i>Epoch</i> 30 dan 50	91
Tabel 4.26 Peforma Deteksi Masker Arsitektur VGG16	92
Tabel 4.27 Peforma Deteksi Masker Arsitektur VGG16 menggunakan <i>Dataset</i> Percobaan	93
Tabel 4.28 Pengaruh Pencahayaan Terhadap Sistem Deteksi Penggunaan Masker Arsitektur VGG16	94
Tabel 4.29 Rincian Pengaruh Resolusi Terhadap <i>Frame Per Second</i> Deteksi Penggunaan Masker Arsitektur InceptionV3	95
Tabel 4.30 Rincian Lama Waktu Uji Coba Proses Absensi.....	96
Tabel 4.31 Peforma 4 Arsitektur Pada <i>Dataset Training</i>	96
Tabel 4.32 Peforma 4 Arsitektur Pada <i>Dataset</i> Percobaan	96

----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

Latar belakang dilakukan penelitian tentang implementasi protokol kesehatan pada sistem absensi new normal berawal dari upaya mengurangi penyebaran COVID-19 pada sistem absensi sidik jari. Protokol kesehatan menghimbau untuk tidak melakukan kontak fisik secara langsung seperti berjabat tangan ataupun kontak fisik secara tidak langsung seperti menggunakan barang/alat bersama, pada PT. Galena Perkasa masih menggunakan absensi sidik jari yang dapat mengakibatkan penyebaran COVID-19. Penelitian ini diawali dengan studi literatur, pemodelan sistem, desain mekanik. Sistematika laporan ini penelitian ini terdiri dari 5 Bab

1.1 Latar Belakang

PT. Galena Perkasa Sidoarjo adalah perusahaan yang bergerak dalam bidang jasa transportasi niaga darat, terbatas hanya dalam pengiriman barang, dengan menggunakan unit kendaraan box tertutup (bukan bak terbuka). PT. Galena Perkasa Sidoarjo didirikan oleh Bapak Krisna Didit Wiwaha pada tahun 1996 dalam bentuk Perseoran Komanditer (*Commanditaire Vennootschaap, CV*) di Pandaan (Pasuruan). Seiring dengan pertambahan jumlah konsumen, maka di awal tahun 2006 berkembang menjadi Perseroan Terbatas (*Naamloze Vennootschaap, NV*) dengan pertambahan 2 kantor cabang operasional lagi di Solo dan Karawang. Hingga saat ini PT. Galena Perkasa tetap menjadi jasa transportasi niaga darat (non kereta api) terbatas dalam pengiriman barang yang telah memiliki 200 lebih armada mulai truk tronton fuso hingga colt diesel fullbox. Terdapat 2 jenis pegawai pada PT. Galena Perkasa yaitu pegawai tetap dan pegawai harian yang dimana mayoritas pegawai harian adalah pengemudi armada. Pegawai harian pada PT. Galena Perkasa tidak selalu bekerja dalam jangka waktu tertentu, perhitungan pekerjaan dihitung perpengiriman barang, maka dari itu pada banyak pegawai harian yang keluar dan masuk.

Pada awal 2020 COVID-19 masuk ke Indonesia membuat pemerintah mulai menerapkan protokol kesehatan yang dimana setiap warga diwajibkan menggunakan masker, menjaga suhu tubuh, cuci tangan, serta mengurangi kontak fisik secara langsung dan tidak langsung. Dengan mulai diimplementasikan protokol kesehatan, PT. Galena Perkasa yang masih menggunakan absensi konvensional yang merupakan bersentuhan secara tidak langsung dapat menjadi media penyebaran COVID-19 dan perlu upaya pembaharuan.

Maka dari itu proyek akhir ini akan membahas bagaimana membuat sebuah sistem absensi new normal pada perkantoran khususnya PT. Galena Perkasa dengan mengimplementasikan protokol kesehatan. Absensi new normal ini akan mengedepankan sebuah sistem yang minim akan kontak fisik mulai dari registrasi data hingga teknis proses absensi. Harapannya dengan implementasinya sistem ini para seluruh pegawai dapat merasa aman dan nyaman tanpa takut tertular COVID-19 dan para pemilik perusahaan dapat memaksimalkan produktivitas dari industri mereka

1.2 Tujuan dan Manfaat

Menciptakan sistem absensi yang minim akan kontak fisik mulai dari registrasi data hingga ke teknis absensi tanpa mengurangi keabsahan data sebagai upaya mengurangi penyebaran COVID-19. Pada sistem absensi new normal ini memiliki fitur mengidentifikasi wajah, mendeteksi masker pada wajah dan juga temperature tubuh sebagai upaya implementasi protokol kesehatan serta meningkatkan produktivitas pegawai PT. Galena Perkasa.

1.3 Rumusan Masalah

Permasalahan yang akan dibahas dalam proyek akhir ini adalah sistem absensi yang digunakan pada PT. Galena Perkasa masih menggunakan model konvensional atau sidik jari yang sudah tidak relevan dengan keadaan pandemi saat ini. Berdasarkan hal tersebut maka perlunya sistem absensi untuk meminimalisir kontak fisik guna

mencegah penyebaran COVID-19 tanpa mengurangi keabsahan dari data absensi pegawai PT. Galena Perkasa.

1.4 Batasan Masalah

Adapun batasan masalah pada Proyek Akhir ini memiliki ruang lingkup sebagai berikut:

1. Tidak membahas detail mengenai proses rekognisi wajah
2. Tidak membahas secara rinci mengenai database dan pengolahan data absensi new normal
3. Tidak membahas mengenai arsitektur CNN selain MobileNetV2, InceptionV3, NASNetMobile, dan VGG16.
4. Tidak membahas mengenai proses recognisi dengan penggunaan masker pada wajah

1.5 Sistematika Penulisan

Sistematika penulisan dalam laporan proyek akhir ini adalah :

BAB I PENDAHULUAN

Bab ini meliputi latar belakang dari masalah sistem absensi milik PT. Galen perkasa, tujuan yang akan dicapai, dan sistematika penulisan Proyek Akhir.

BAB II TINJAUAN PUSTAKA

Bab ini berisi mengenai tinjauan pustaka yang dijadikan acuan dalam proses pengerjaan proyek akhir ini, antara lain Penyebaran Covid-19, protokol kesehatan, *base64*, sensor *infrared*, pengolahan citra, operasi konvolusi, dan model *Convolution Neural Network*.

BAB III DESAIN SISTEM DAN IMPLEMENTASI

Bab ini menjelaskan mengenai perancangan sistem mulai dari operasi input data dengan android ke database, implementasi protokol kesehatan khususnya temperatur tubuh dan deteksi masker menggunakan *Convolution Neural Network*.

BAB IV UJI COBA DAN EVALUASI SISTEM

Bab ini membahas tentang pengujian sistem yang telah terimplementasi dengan melakukan studi kasus pada sistem absensi dengan memperhatikan kinerja sistem terhadap implementasi protokol kesehatan dan performa deteksi masker menggunakan 4 arsitektur yang berbeda.

BAB V PENUTUP

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari proyek akhir dan saran terkait evaluasi proyek akhir dan pengembangan alat ini lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Teori

Berdasarkan hasil laporan world health organization (WHO) dari 96 negara yang melaporkan terdapat lebih dari 2.7 juta jiwa dan kematian mencapai 180.000 diakibatkan oleh Covid-19. Saat ini semakin banyak orang-orang yang khawatir mengenai kesehatan dan pemerintah sudah memutuskan bahwa keamanan kesehatan publik merupakan prioritas utama untuk diselesaikan. Memutus jalur penyebaran Covid-19, World Health Organization (WHO) menyarankan untuk setiap warga perlu memakai masker bagi yang memiliki gejala penyakit pernafasan atau orang yang bersosialisasi langsung dengan penderita penyakit pernafasan. Setelah melakukan berbagai upaya pemerintah mulai menerapkan protokol kesehatan sebagai upaya mengurangi penyebaran Covid-19[1]. Salah satu aspek penting dalam protokol kesehatan adalah mewajibkan seluruh warga memakai masker ketika beraktifitas diluar rumah serta mencuci tangan serta suhu tubuh normal ketika mulai memasuki tempat-tempat umum layaknya rumah sakit, pusat perbelanjaan hingga perkantoran. Mayoritas perkantoran masih menggunakan sistem absensi sidik jari ataupun timestap yang sangat rentan akan penyebaran Covid-19 dikarenakan alat absensi model seperti ini digunakan oleh banyak orang secara bergantian. Maka dari itu penelitian ini bertujuan untuk menciptakan sebuah sistem absensi yang minim akan kontak fisik mulai dari registrasi hingga proses absensi guna mengurangi penyebaran COVID-19. Dengan adanya penelitian sebelumnya, dapat mempermudah pengerjakan proyek akhir ini, berikut merupakan *paper/jurnal/proyek* akhir sebelumnya

- a) Pada tugas akhir yang berjudul Convolutional Neural Network Untuk Pengenalan Wajah Secara Real-Time (Zufar Muhammad, 2016) Metode lokal yaitu metode ekstraksi fitur dari beberapa daerah wajah seperti mulut, mata, dan hidung yang kemudian digunakan untuk klasifikasi. Beberapa metode lokal seperti *Local*

Binary Pattern merupakan metode ekstraksi fitur dengan cara mendapatkan nilai biner piksel pada pusat citra dengan 8 nilai piksel disekelilingnya sehingga didapatkan suatu nilai matrik baru yang akan dirubah kesuatu histogram untuk memperoleh fitur vektor wajah.

- b) Pada jurnal yang berjudul WebLogo-2M: Scalable Logo Detection by Deep Learning From the Web (Su hang, Shaogang Gong, Xiatian Zhu, , Adrian Popescu, Alexandru Ginsca, Herve Le Borgne, Yuen Peng Loh et al, 2015) dalam proses pengolahan citra untuk mempercepat sebuah proses pengenalan objek pada suatu gambar terdapat sebuah metode yang mana mengurangi ukuran pixel dari sebuah gambar pada *dataset*.
- c) Pada jurnal yang berjudul Towards an Analytical Age-Dependent Model of Contrast Sensitivity Functions for an Ageing Society (Joulán K, Brémond R, Hautière N, 2015) algoritma dalam pengenalan pola dibutuhkan sebuah metode segmentasi yang dapat mendeskripsikan sebuah objek, dalam studi kasus pada jurnal diatas warna merupakan aspek penting dalam pengenalan pola yang diinginkan maka digunakan metode Contrast Function(CSF) untuk mempercepat proses pengenalan pola.
- d) Pada jurnal yang berjudul Deep learning based super-resolution for improved action recognition (Nasrollahi, Kamal, Sergio Escalera, Pejman Rasti, Gholamreza Anbarjafari, Xavier Baro, Hugo Jair Escalante, and Thomas B. Moeslund, 2015) ukuran gambar pada *dataset* mempengaruhi performa dan akurasi dari pengenalan objek, dalam beberapa kasus pengguna pengolahan citra mengurangi ukuran pixel sebagai upaya mempercepat proses pengenalan objek namun dengan mengorbankan akurasi. Dengan metode image illumination enhancement pada color pixel correction dapat meningkatkan akurasi sebuah gambar yang ukuran pixelnya telah diperkecil.

- e) Pada tugas akhir yang berjudul sistem pengenalan wajah menggunakan metode metode Hidden Markov Model (Sepritahara, 2012) menyebutkan bahwa file ekstensi datasheet mempengaruhi kecepatan sistem untuk mengenali wajah seseorang. Dari berbagai banyak file ekstensi gambar seperti .jpg, .jpeg, .gif, .bitmap, .png namun dalam pembuatan datasheet pengolahan citra file ekstensi terbaik adalah .bitmap dikarenakan pada format .bitmap citra disimpan dalam sebuah metode bit-per-pixel. Semakin besar ukuran bit-per-pixel dari suatu gambar maka dapat memberikan informasi yang lebih akurat karena sistem pengolahan citra menggunakan vektor.
- f) Pada jurnal yang berjudul Rational use of face masks in the Covid-19 pandemic (S. Feng, C. Shen, N. Xia, W. Song, M. Fan, and B. J. Cowling, 2020) menunjukkan bahwa penggunaan masker dapat memutus penyebaran virus COVID-19. Pada waktu yang sama World Health Organization (WHO) juga menyarankan untuk menggunakan masker bagi yang mengalami gejala pernafasan ataupun yang berhubungan langsung dengan penderita penyakit pernafasan.
- g) Pada jurnal yang berjudul Masked facerecognition *dataset* and application (Z.Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei, 2020.) deteksi masker wajah merujuk pada orang yang menggunakan masker atau tidak serta mencari salah satu bagian dari wajah yaitu mata dan dahi.
- h) Pada peraturan pemerintah No. 21 Tahun 2020 tentang Pembatasan Sosial Berskala Besar dalam Rangka Percepatan Penanganan Corona Virus Disease 2019 (COVID-19) Salah satu aspek penting dalam protokol kesehatan adalah mewajibkan seluruh warga memakai masker ketika beraktifitas diluar rumah serta mencuci tangan serta suhu tubuh normal ketika mulai memasuki tempat tempat umum layaknya rumah sakit, pusat perbelanjaan hingga perkantoran.

- i) Pada tugas akhir yang berjudul Rancang Bangun New Normal Covid-19 Masker Detektor dengan Notifikasi Telegram (Musakkarul Mu'minin, 2020) pengembangan penggunaan mask detektor dapat dilakukan dengan mengintegrasikan dengan database untuk memantau penggunaan sistem serta kontrol gerbang atau pintu dengan syarat penggunaan masker.
- j) Pada jurnal yang berjudul Wireless Attendance Management System Based on Iris Recognition (Kardy S, 2010) sistem absensi yang mengautentikasi pengguna dengan mengenali iris mata individu untuk menyatakan kehadiran dapat lebih efektif dalam hal mencatat kehadiran umum dari para karyawan serta mengurangi kecurangan yang terjadi.

2.2 Dasar Teori

2.2.1 Covid-19

Corona Virus Disease 2019 atau Covid-19 merupakan keluarga besar virus yang menyebabkan penyakit pada manusia dan hewan. Pada manusia biasanya menyebabkan penyakit infeksi saluran pernapasan, mulai flu biasa hingga penyakit yang serius seperti Middle East Respiratory Syndrome (MERS) dan Sindrom Pernapasan Akut Berat/ Severe Acute Respiratory Syndrome (SARS). Penyakit ini terutama menyebar di antara orang-orang melalui tetesan pernapasan dari batuk dan bersin. Virus ini dapat tetap bertahan hingga tiga hari dengan plastik dan stainless steel SARS CoV-2 dapat bertahan hingga tiga hari, atau dalam aerosol selama tiga jam[9].

COVID-19 paling utama ditransmisikan oleh tetesan aerosol penderita dan melalui kontak langsung. Aerosol kemungkinan ditransmisikan ketika orang memiliki kontak langsung dengan penderita dalam jangka waktu yang terlalu lama. Konsentrasi aerosol di ruang yang relatif tertutup akan semakin tinggi sehingga penularan akan semakin mudah[8].

2.2.2 Protokol Kesehatan

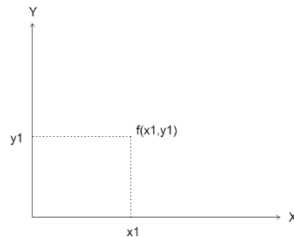
Penyebaran virus Covid-19 memiliki berbagai jalur transmisi mulai dari udara ataupun permukaan. Berawal dari metode penyebaran COVID-19 maka pemerintah mulai menerapkan protokol kesehatan sebagai upaya mengurangi penyebaran Covid-19 serta menjaga keberlangsungan usaha sektor jasa dan perdagangan pada masa *Pandemic Corono Virus Disease* 2019. Protokol kesehatan ini ditujukan kepada pengelola tempat kerja, pelaku usaha, pekerja, pelanggan/konsumen dan masyarakat yang terlibat pada sektor jasa dan perdagangan (area publik) melalui adaptasi perubahan pola hidup pada situasi COVID-19 (*New Normal*)[9]. Dengan menerapkan protokol ini diharapkan dapat meminimalisir risiko dan dampak pandemi Covid-19 pada usaha sektor jasa dan perdagangan (area publik) dimana terdapat potensi penularan COVID-19 akibat berkumpulnya sejumlah/banyak orang dalam satu lokasi.

Penerapan protokol kesehatan selain upaya mengurangi penyebaran Covid-19 juga dalam rangka percepatan penanganan Covid-19 setelah masa Pembatasan Sosial Berskala Besar(PSSB). Pemerintah menghimbau untuk seluruh pelaku usaha pada sektor jasa dan perdagangan untuk menerapkan kesiapan teknis dalam mengurangi penyebaran Covid-19 diantaranya[6] :

- a. Melakukan pembersihan dan disinfeksi secara berkala di area kerja dan area publik (mendisinfeksi fasilitas yang sering disentuh publik setiap 4 jam sekali)
- b. Menyediakan fasilitas cuci tangan
- c. Pekerja memahami perlindungan diri dari penularan Covid-19 dengan Perilaku Hidup Bersih dan Sehat(PHBS).
- d. Melakukan pengecekan suhu badan sebelum mulai bekerja di pintu masuk. Jika ditemukan pekerja dengan suhu $> 37,5$ C tidak diperkenankan masuk.
- e. Mewajibkan pekerja dan pengunjung menggunakan masker.
- f. Melakukan pembatasan jarak fisik minimal 1 meter.
- g. Mencegah terjadinya kerumunan

2.2.3 Citra Digital

Citra merupakan representasi, kemiripan atau imitasi dari suatu objek atau benda. Secara matematis, citra dinyatakan sebagai sebuah fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Fungsi $f(x,y)$ dapat dilihat sebagai fungsi dengan dua unsur. Unsur yang pertama adalah kekuatan sumber cahaya yang melingkupi sistem optik manusia ataupun kamera terhadap objek (*illumination*). Unsur kedua merupakan besarnya cahaya yang direfleksikan oleh objek kedalam sistem optik (*reflectance component*). Keduanya dituliskan sebagai fungsi $i(x,y)$ dan $r(x,y)$ yang digabungkan sebagai perkalian fungsi untuk membentuk fungsi $f(x,y)$ [17].



Gambar 2.1 Citra Fungsi Dua Variable
Dimana fungsi dari $f(x,y)$ dapat ditulis sebagai persamaan:

$$f(x, y) = i(x, y) * r(x, y) \text{ di mana } 0 < i(x, y) < \infty \text{ dan } 0 < r(x, y) < 1 \quad (2.1)$$

Citra yang terlihat oleh sistem optik merupakan cahaya yang direfleksikan dari sebuah objek oleh cahaya, dengan kata lain pencahayaan mempengaruhi isi dan perspektif dari sistem optik manusia atau kamer. Citra dibedakan menjadi dua yaitu citra kontinu diperoleh dari sistem optik yang menerima sinyal analog(mata manusia dan kamera analog) dan citra diskrit(digital) dihasilkan melalui proses digitalisasi terhadap citra kontinu. Proses digitalisasi pada citra digital dibagi menjadi dua proses yakni sampling dan kuantisasi. Proses sampling merupakan proses pengambilan nilai diskrit koordinat ruang(x,y) secara periodik dengan periode sampling T . Proses kuantisasi

merupakan proses pengelompokkan nilai tingkat keabuan citra kontinu kedalam beberapa level atau merupakan proses membagi skala keabuan (0,L) menjadi G buah level yang dinyatakan dengan suatu harga bilangan bulat(*integer*), dinyatakan sebagai $G = 2^m$ dengan G adalah derajat keabuan dan m adalah bilangan bulat positif[17].

Citra digital juga disebut suatu matriks yang terdiri dari baris dan kolom, dimana setiap baris dan kolom menyatakan titik pada citra. Nilai matriksnya menyatakan nilai kecerahan titik tersebut. Titik titik tersebut dalam citra digital sering dinamakan sebagai elemen citra atau pixel.

		kolom/lebar → y			
		0	1	2	3
baris/tinggi ↓ x	0	1	0	0	1
	1	0	1	1	0
	2	0	1	1	0
	3	1	0	0	1

→ pixel

Gambar 2.2 Elemen Citra atau Pixel

Dengan demikian citra digital dapat didefinisikan suatu matriks A berukuran M (baris) x N (kolom) dimana indeks baris dan kolomnya menyatakan suatu titik atau pixel pada citra tersebut dan elemen matriksnya menyatakan tingkat keabuan pada titik tersebut.

$$A = \begin{bmatrix} a_{0,0} & \dots & a_{0,N-1} \\ a_{1,0} & \dots & a_{1,N-1} \\ \vdots & \vdots & \vdots \\ a_{M-1,0} & \dots & a_{M-1,N-1} \end{bmatrix} \quad (2.2)$$

2.2.3.1 Citra Biner

Binary Image atau citra biner adalah citra yang hanya mempunyai dua nilai derajat keabuan yaitu hitam dan putih. Pixel yang

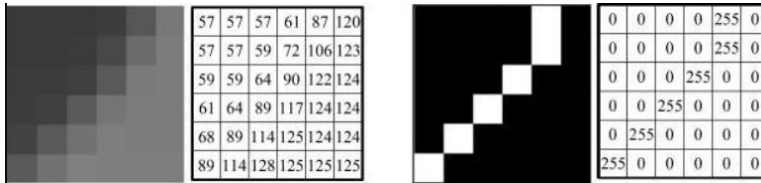
bernilai 1 melambangkan warna hitam, sedangkan pixel yang bernilai 0 adalah warna putih. Citra biner ini dapat dinyatakan dalam suatu fungsi sebagai berikut :

$$f(x,y) \in \{0,1\} \quad (2.3)$$

Sebagai contoh gambar yang dapat merepresentasikan citra biner terdapat pada gambar 2.2 yang mana biner yang bernilai 1 membentuk huruf “X” jika dibandingkan dengan biner yang bernilai 0.

2.2.3.2 Citra Grayscale

Citra yang terdiri dari satu layer warna dengan derajat keabuan tertentu. Citra grayscale diukur berdasarkan skala intensitas kecerahan yang bernilai dari rentan 0 – 255. Untuk citra *grayscale* yang bernilai 0 merepresentasikan warna hitam pekat sedangkan 255 merepresentasikan warna putih.



Gambar 2.3 *Grayscale Pixel*

Dari gambar 2.3 merupakan contoh dari gambar pixel yang telah diubah menjadi *grayscale* dengan intensitas kecahayaan atau hasil akan mempengaruhi nilai dengan rentan 0 – 255. Citra *grayscale* dapat dinyatakan dalam suatu fungsi sebagai berikut :

$$f(x,y) \in [0...255] \quad (2.4)$$

2.2.3.3 Citra Bewarna

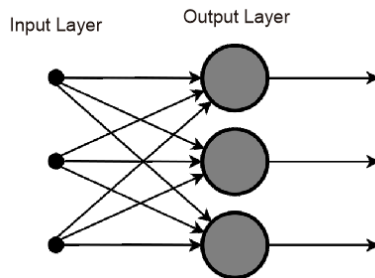
Red, Green, Blue (RGB) adalah suatu model warna yang terdiri tiga warna dan menjadikan suatu susunan warna yang cukup beragam. Pada masing masing warna dasar RGB memiliki rentang nilai dari 0 hingga 255 layaknya *grayscale*. Skala ini didasari oleh cara membaca 8 digit bilangan biner yang digunakan oleh komputer. Masing masing warna dasar memiliki skala 256 maka apabila warna campuran yang bisa didapatkan sebesar $256 \times 256 \times 256 = 16.777.216$ jenis warna.

Sebuah jenis warna, dapat direpresentasikan sebagai sebuah vektor di tiga dimensi yang digunakan dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalkan sebuah vektor dituliskan sebagai $r = (x,y,z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen R (*Red*), G (*Green*), B (*Blue*). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = RGB(xxx, xxx, xxx). Putih = RGB(255,255,255), sedangkan untuk hitam= RGB(0,0,0).

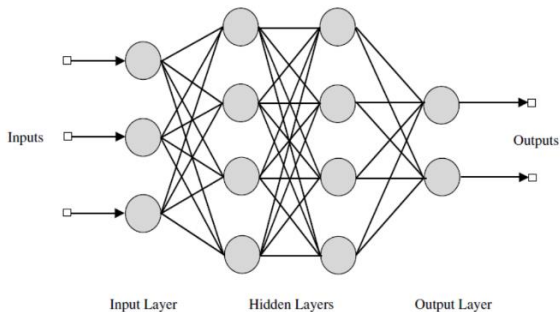
2.2.4 Convolutional Neural Network(CNN)

Convolutional Neural Network(CNN) adalah sebuah variasi dari *multilayer perception* yang terinspirasi dari jaringan syaraf manusia. Keuntungan terbesar dari pengimplementasian CNN adalah mengurangi parameter yang ada pada Artificial Neural Network(ANN). Penggunaan dari CNN sudah dibuktikan oleh peneliti dan pengembang yang telah menyelesaikan berbagai macam permasalahan kompleks menggunakan metode ini. Penelitian awal yang mendasari penemuan ini pertama kali dilakukan oleh Hubel dan Wiesel yang melakukan penelitian visual cortex pada indera penglihatan kucing. Visual cortex adalah bagian dari otak yang berfungsi untuk memproses informasi visual yang didalamnya berisi susunan kompleks dari sel. Sel-sel ini sensitif terhadap bagian daerah kecil(sub-regions) pada bidang visual disebut sebagai bidang reseptif(*receptive field*). Daerah bagian berbentuk seperti ubin berfungsi untuk menutupi seluruh bidang visual. Sel-sel ini bertindak sebagai filter lokal atas ruang input dan dapat dieksploitasi pada daerah spasial lokal gambar. Selain itu dari hasil penelitian teridentifikasi 2 tipe sel dasar, sel tipe sederhana merespon maksimal terhadap pola tertentu seperti tepi didalam bidang reseptifnya sedangkan sel tipe kompleks memiliki bidang reseptif yang lebih besar dan lokal invariant terhadap posisi yang sesuai pada pola. Visual cortex pada hewan sangat kuat dalam sistem pemrosesan visual yang pernah ada. Hingga banyak penelitian yang terinspirasi dari cara kerjanya dan menghasilkan Piksel citra Kernel konvolusi Nilai piksel baru seperti 15 model-model baru diantaranya seperti Xception , ResNet, dan LeNet-5[4].

Gambaran Umum *Neural Network* dapat direpresentasikan dengan Gambar 2.4 dan Gambar 2.5, ketika nilai input yang ada pada input layer diproses dengan semua neuron. Jumlah neuron dan jumlah layer pada proses CNN tidak ada hasil yang pasti atau hasil paten untuk menyelesaikan semua kasus, karena itu CNN bersifat eksperimental. Terdapat dua bentuk dalam Neural Network jika melihat *hidden layer* yang dimiliki, ketika neural network input layer terhubung dengan output layer disebut dengan *single layer-Feedforward Network*. Sedangkan apabila sebuah neural network memiliki beberapa hidden layer maka disebut *Multi – Layer Feedforward Network*.



Gambar 2.4 *Single Layer Feedforward Network*



Gambar 2.5 *Multi – Layer Feedforward Network*

Pada *Multilayer Neural Networks*, setiap *hidden layer* tersusun dari kumpulan neuron, dimana setiap neuron terhubung secara penuh(*fully connected*) ke semua neuron pada layer sebelumnya, berbeda dengan *Single Layer Neural Networks* yang hanya ada neuron yang terhubung secara penuh pada layer terakhir tanpa adanya pembagian koneksi antar neuron(tidak ada *hidden layer*). Input layer menerima berupa vektor tunggal sebagai input dan ditransformasikan sepanjang *hidden layer*. Layer terakhir pada keseluruhan jaringan bersifat terhubung secara penuh dan merupakan representasi dari nilai kelas klasifikasi.

Pada *Convolutional Neural Networks* memiliki suatu *layer* yang memiliki susunan *neuron* 3D (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran *layer* sedangkan kedalaman mengacu pada jumlah *layer*. Secara umum jenis *layer* pada CNN dibedakan menjadi dua yaitu:

a. *Layer* ekstraksi fitur gambar, letaknya berada pada awal arsitektur tersusun atas beberapa *layer* dan setiap *layer* tersusun atas *neuron* yang terkoneksi pada daerah lokal (*local region*) *layer* sebelumnya. *Layer* jenis pertama adalah *layer* konvolusi dan *layer* kedua adalah *layer pooling*. Setiap *layer* diberlakukan fungsi aktivasi. Posisinya berselang-seling antara jenis pertama dengan jenis kedua. *Layer* ini menerima input gambar secara langsung dan memprosesnya hingga menghasilkan keluaran berupa vektor untuk diolah pada *layer* berikutnya

b. *Layer* klasifikasi, tersusun atas beberapa *layer* dan setiap *layer* tersusun atas *neuron* yang terkoneksi secara penuh (*fully connected*) dengan *layer* lainnya. *Layer* ini menerima input dari hasil keluaran *layer* ekstraksi fitur gambar berupa vektor kemudian ditransformasikan seperti *Multi Neural Networks* dengan tambahan beberapa *hidden layer*. Hasil keluaran berupa skoring kelas untuk klasifikasi.

2.2.4.1 Convolutional Layer

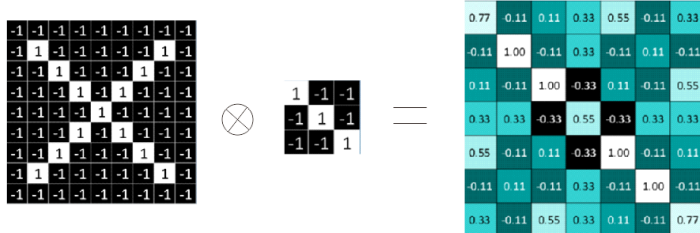
Konvolusi adalah sebuah operasi matematika dari dua fungsi f dan g yang menghasilkan sebuah fungsi ketiga h . Dalam pengolahan citra digital konvolusi adalah operasi pengendalian sebuah citra dengan sebuah kernel/filter yang akan menghasilkan sebuah citra baru. Untuk fungsi diskrit, konvolusi didefinisikan sebagai berikut

$$\begin{aligned} h(x, y) &:= (f * g)(x, y) \\ &:= \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b)g(x - a, y - b) \end{aligned} \quad (2.5)$$

Fungsi $f(x, y)$ adalah sebuah fungsi yang merepresentasikan citra sedangkan fungsi $g(x, y)$ adalah kernel/filter konvolusi. Kernel $g(x, y)$ merupakan suatu jendela/matriks yang dioperasikan secara bergeser pada sinyal masukan $f(x, y)$, yang dalam hal ini, jumlah perkalian kedua fungsi pada setiap titik merupakan hasil konvolusi dinyatakan sebagai keluaran fungsi $h(x)$. Karena itu ukuran citra terbatas pada ukuran $M \times N$ dengan filter berukuran $m \times n$ maka operasi fungsi konvolusi didefinisikan sebagai berikut ini:

$$(f * g)(x, y) := \sum_{a=x-h}^{x+h} \sum_{b=y-w}^{y+h} f(a, b)g(x - a, y - b) \quad (2.6)$$

Dimana $m = 2h + 1$ adalah tinggi filter dan $n = 2w + 1$ adalah lebar filter. Operasi dilakukan dengan melakukan kombinasi linear dari mengambil bagian citra yang sama atau fitur dari sebuah citra dan hasil operasi disimpan berupa elemen nilai matriks kemudian dilanjutkan dengan menggeser filter *pixel* per *pixel* sampai terhimpun nilai *pixel* baru. Proses konvolusi ini dapat dilihat pada Gambar 2.6 dengan citra berukuran $M = 9$ $N = 9$ dan filter berukuran $m = 3$ $n = 3$.



Gambar 2. 6 Proses Operasi Konvolusi

Operasi yang pertama dilakukan adalah mengalikan bagian dari elemen citra pertama berukuran 3x3 (ukuran menyesuaikan dengan filter) dengan tiap elemen filter yang bersesuaian, kemudian dijumlahkan pada akhir perhitungan.

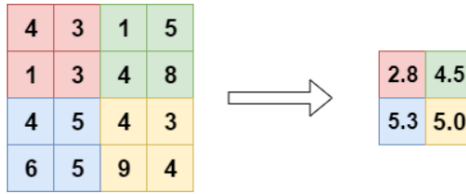
2.2.4.2 Pooling Layer

Mayoritas proses konvolusi ataupun arsitektur CNN selalu memiliki *pooling layer* setelah konvolusi, meskipun penempatan dari pooling layer beragam secara algoritma. Tujuan *pooling layer* sendiri adalah mereduksi jumlah dimensi dari feature map (down sampling) sehingga dapat mengurangi proses komputasi. *Pooling layer* bekerja dengan blok spasial sepanjang ukuran feature map dengan stride (pergeseran) yang telah ditentukan. Terdapat 2 jenis pooling layer yang sering digunakan dalam proses CNN yaitu average pooling dan max pooling.

a. Average Pooling

Menghitung rata nilai dari blok spasial pada feature map di setiap pergerakan area pooling.

$$u_{i,j}^{(k)} = \frac{1}{H^2} \sum_{(p,q) \in P_{ij}} z_{p,q}^{(k)} \quad (2.7)$$

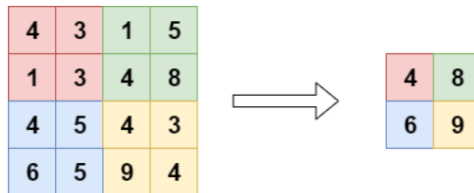


Gambar 2.7 *Average Pooling* dengan *Stride 2*

b. *Max Pooling*

Keluaran dari max pooling adalah nilai tertinggi dari blok spasial pada feature map pada setiap pergerakan area pooling.

$$u_{i,j}^{(k)} = \max_{p,q \in P_{i,j}} z_{p,q}^{(k)} \quad (2.8)$$



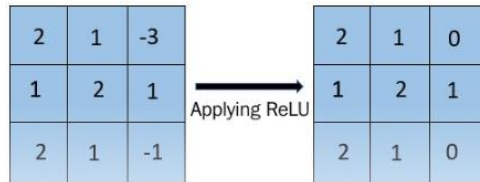
Gambar 2.8 *Max Pooling* dengan *Stride 2*

Selain mereduksi ukuran dari *feature map* (down sampling), pooling layer juga berguna untuk mengatasi *translation invariance* atau dapat mendeteksi objek dimanapun objek itu berada pada frame tanpa harus menentukan atau memberi batasan untuk objek harus berada pada suatu tempat agar dapat teridentifikasi. Selain posisi atau tempat, pooling layer juga dapat mengatais objek terbalik ataupun miring yang dikarenakan proses pooling ini adalah sebuah fitur mencari pola dan dapat meningkatkan peforma dalam CNN.

2.2.4.3 *Rectified Linear Unit (ReLU)*

Pada arsitektur *Convolution Neural Network* terdapat sebuah fungsi yang berguna untuk mengubah fungsi linear menjadi non linear jika berbicara pada proses *classifier*. Fungsi tersebut disebut *activation function*, pada proses *classifier* CNN terdapat sebuah logistic regression yang di dialamnya terdapat regresi linear yang dimasukkan ke *activation*

function agar model mengklasifikasi data lebih dinamis, dimana ReLu merupakan salah satu *activation function*.



Gambar 2.9 Rectified Linear Unit(ReLu)

Selain pada proses classifier, ReLu bisa juga diletakkan setelah proses konvolusi atau pooling untuk menghasilkan suatu pola fitur. Karakteristik ReLu adalah dimana nilai yang berada dibawah 0 akan diubah menjadi 0.

2.2.4.4 Softmax

Softmax adalah sebuah proses pengambilan keputusan akhir dari proses klasifikasi setelah *feature extraction*. Hasil dari dari proses klasifikasi mayoritas berupa angka yang mendeskripsikan nilai prediksi dari model, namun nilai prediksi ini tidak berupa nilai probabilitas. Sebagai contoh dalam koin terdapat 2 sisi, maka kemungkinan untuk mendapat salah satu sisi koin maka nilainya adalah 0,5 dan karena terdapat 2 sisi maka sisi satu lainnya juga bernilai 0,5. Maka dari itu *softmax* berguna untuk mengubah hasil prediksi output klasifikasi menjadi probabilitas 0 hingga 1 sesuai dengan jumlah output.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.9)$$

Sebagai contoh apabila output dari proses klasifikasi adalah 8 dan 6 lalu dua angka ini akan menjadi input softmax. Hasil akhir dari softmax pada contoh input diatas adalah 0,88 dan 0,12.

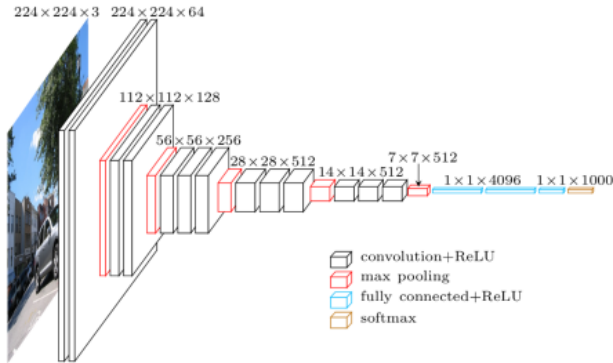
2.2.4.5 *Epoch, Batch Size, Dan Learning Rate*

Epoch merupakan jumlah dari berapa kali atau berapa lama model belajar dalam satu proses keseluruhan data *training* masuk ke proses *training*. Dalam 1 *epoch* akan terjadi beberapa kali iterasi sesuai jumlah *batch size* yang disetting dan *dataset* yang ada, sebagai contoh apabila terdapat 1000 *dataset* dan *batch size* diatur 50 maka akan terjadi 20 kali iterasi dalam 1 *epoch*.

Banyak proses *training* model CNN menggunakan *gradient descent* sebagai metode update weight pada setiap kali *epoch* atau iterasi berdasarkan loss yang keluar setiap *epoch* atau iterasi. Proses update weight ini sering disebut sebagai backpropagation. Learning rate ini akan mengontrol seberapa cepat model belajar dengan melihat gradient descent, dalam beberapa kasus semakin kecil learning rate maka perlu semakin banyak *epoch* untuk mendapat hasil yang lebih baik bergitupun sebaliknya. Nilai learning rate yang terlalu besar dapat membuat model tidak dapat menemukan nilai optimal atau melewati nilai optimal jika dibandingkan dengan learning rate kecil, sedangkan semakin apabila learning rate terlalu kecil dapat membuat model *stuck* atau tidak berubah pada suatu nilai yang belum tentu nilai itu adalah nilai paling optimal. Maka dari itu proses setting parameter pembuatan model CNN ini bersifat eksperimental, tidak ada nilai parameter yang pasti untuk semua studi kasus.

2.2.4.6 *Arsitektur Convolution Neural Network*

Arsitektur CNN sangat beragam, berbeda arsitektur dapat menghasilkan hasil dan beban proses yang berbeda. Peneliti dan pengembang banyak melakukan riset untuk membuat sebuah arsitektur CNN yang ringan dan memiliki hasil yang baik. Sebagai contoh terdapat arsitektur VGG16 buatan salah satu laboratorium oxford yang menjadi *runner up* pada perlombaan ILSVRC(*Imagenet Large Scale Visual Recognition Challenge*) pada tahun 2014.



Gambar 2.10 Arsitektur VGG16

Arsitektur VGG16 melakukan proses konvolusi tiga kali diawal dan menghasilkan sebuah data ukuran 224×224 dengan 64 channel. Proses selanjutnya melakukan *max polling* hingga data menjadi 112×112 dengan 128 channel lalu dilakukan konvolusi sebanyak dua kali. *Max polling* yang berguna untuk menfokuskan data menjadi lebih kecil dengan cara mencari nilai tertinggi/dominan dari beberapa pixel. Proses max polling dan dua kali konvolusi dilakukan hingga data menjadi 1×1 dengan 4096 channel. Lalu dilakukan fully connected+ReLU dan softmax pada proses akhir. ReLu kepanjangan dari Rectified Linear Unit yang berguna untuk menfilter nilai dari data yang dibawah nol menjadi 0 agar dapat menfokuskan gambar dari nilai nilai data yang tidak dipakai. Softmax memiliki fungsi hampir sama dengan ReLu namun softmax lebih berfokus pada proses klasifikasi dengan membuat seluruh data dan channel dibuat dibuat menjadi data ukuran $1 \times 1 \times 1000$ dan lalu dilakukan proses perbandingan atau klasifikasi dengan *data training* atau *validation data*.

2.2.4.7 Confusion Matrix

Confusion Matrix adalah salah satu metode untuk mengukur performa dari suatu model klasifikasi adalah dengan mencari nilai *precision*, *recall*, serta nilai akurasi dari suatu model.

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 2.11 *Confusion Matrix*

Terdapat 4 istilah klasifikasi dari hasil proses *training* model CNN pada confusion matrix yaitu *True Positive*(TP) dimana data dikategorikan benar, model memprediksi benar. *False Positive*(FP) adalah ketika data dikategorikan benar dan model memprediksi salah. *False Negative*(FN) adalah data salah diprediksi benar, dan *True Negative*(TN) adalah data salah diprediksi salah oleh model. Dari confusion matrix dapat dihasilkan nilai *precision*, *recall*, *accuracy*, dan *f1-score*.

$$\text{Precision} = \frac{tp}{tp + fp} \quad (2.9)$$

$$\text{Recall} = \frac{tp}{tp + fn} \quad (2.10)$$

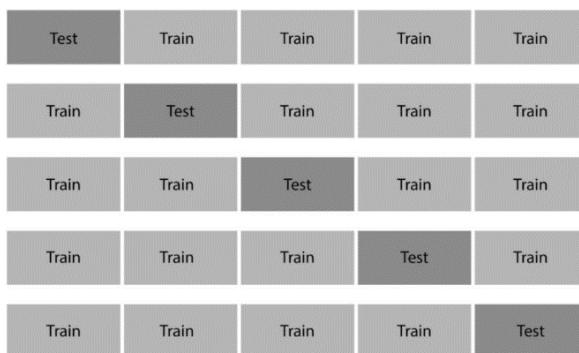
$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \quad (2.11)$$

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (2.12)$$

Precision adalah rasio dari prediksi benar positif dengan keseluruhan hasil yang diprediksi positif, dari seluruh data yang telah di prediksi, berapa data yang benar benar positif. *Recall* atau sensitivitas adalah rasio prediksi benar positif dibandingkan dengan seluruh data yang benar positif. Akurasi adalah rasio prediksi benar (negatif atau positif) dengan keseluruhan data. *F1-score* adalah perbandingan rata rata antara presisi dan *recall* yang dibobotkan, pada penilaian deteksi penggunaan masker ini akan menggunakan nilai akan menggunakan *f1-score* dan akurasi sebagai penilaian bobot tertinggi.

2.2.4.10 K-Fold Cross Validation

K-Fold Cross Validation merupakan sebuah metode untuk membagi data menjadi beberapa bagian sebanyak ‘k’ dan akan digunakan sebagai proses testing model.



Gambar 2.12 *K-Fold Cross Validation*

Sebagai contoh apabila terdapat 1000 gambar pada *dataset* dan *k-fold* disetting menjadi 5 maka *dataset* akan dibagi menjadi 5. 200 gambar pertama akan digunakan sebagai *testing data* dan 800 gambar lainnya akan digunakan sebagai *training data*, selanjutnya data ke 201 hingga ke 400 akan digunakan sebagai *testing data* dan 800 data lainnya akan digunakan sebagai *training data* dan seterusnya.

2.2.5 OpenCV

library pemrograman opencv adalah library python yang ditujukan untuk *computer vision*. Awalnya dikembangkan oleh pusat penelitian Intel di Nizhny Novgorod (Rusia), kemudian didukung oleh Willow Garage dan sekarang dikelola oleh Itseez. Library OpenCV di bawah lisensi BSD *open-source* gratis dan *cross-platform* untuk digunakan. Didalamnya terdapat ratusan algoritma *computer vision*. OpenCV memiliki struktur modul, dengan modulnya sebagai berikut:

- *Core functionality* - mendefinisikan struktur data dasar, termasuk multi-dimensi array (Mat) dan fungsi dasar yang digunakan oleh semua modul lainnya.
- *Image Processing* - modul pengolahan citra yang mencakup pengolahan gambar linear dan *non-linear filtering*, transformasi gambar geometri (mengubah ukuran, *affine* dan *perspective warping*, *generic table-based remapping*), konversi ruang warna, histogram, dan sebagainya.
- Video - modul analisis video yang mencakup estimasi gerakan, *background subtraction*, dan algoritma pelacakan obyek.
- *calib3d* - dasar algoritma geometri *multiple-view geometry*, kalibrasi kamera tunggal dan stereo, estimasi posisi obyek, algoritma korespondensi stereo, dan unsur-unsur rekonstruksi 3D.
- *features2d* - *salient feature detectors*, deskripsi, dan *matchers descriptor*.
 - *objdetect* - deteksi objek dan beberapa contoh obyek yang sudah didefinisikan (misalnya, wajah, mata, mulut, orang, mobil, dan sebagainya).
 - *highgui* - antarmuka yang digunakan untuk UI sederhana.
 - *videoio* - antarmuka yang digunakan untuk menangkap video dan *codec* video.
 - *gpu* - algoritma GPU-*accelerated* dari modul OpenCV 2.5
 - dan beberapa modul pembantu lainnya, seperti FLANN dan *Google test wrappers*, *Python bindings*, dan lain-lain.

2.2.6 MLX90614

infrared thermometer MLX90614 adalah sebuah sensor suhu yang dapat mengukur suhu dari 4cm – 6cm tanpa melakukan kontak langsung dengan objek yang akan diukur.



Gambar 2.13 MLX90614

Sensor ini menggunakan inframerah untuk mengukur atau mendeteksi radiasi panas benda. Prinsip kerjanya dengan menangkap energi panas yang dihasilkan dari pancaran inframerah yang dimiliki setiap benda kemudian dikonversikan dalam bentuk besaran suhu. Sensor ini mampu mendeteksi radiasi pada temperatur objek antara -70 C hingga 380 C. Keluaran dari Sensor ini berbentuk digital.

2.2.7 *Encoding dan Decoding*

Informasi yang dapat dimengerti oleh manusia dan komputer berbeda. Komputer hanya memahami angka biner '0' dan '1'. Ketika akan mendeskripsikan sebuah angka, huruf, atau simbol kepada komputer, manusia akan membuat sebuah pola biner yang terdiri dari '0' dan '1'. Sebagai contoh huruf 'A' adalah '000000' dalam biner 6-bit dan 'a' adalah '011010' dalam biner 6-bit.

Perpindahan informasi antar komputer atau antar sistem dilakukan dengan mengirimkan sebuah pola biner ke tujuan, namun terdapat kekurangan dimana keamanan yang kurang dan tidak efisien. Maka dari itu dibuat sebuah metode encoding dan decoding sebagai upaya meningkatkan keamanan data yang dikirim serta meningkatkan efisiensi pengiriman data. Macam macam algoritma encoding dan decoding sangat beragam, sesuai kebutuhan jenis data yang dikirim/diterima dan metode pengiriman/penerimaan.

Encoding adalah proses mengubah data desimal atau char seperti angka 0 sampai 9, huruf A – Z ke biner. Sedangkan decoding merupakan sebuah proses merubah pola biner ke data desimal atau char. Perlu diketahui bahwa bahwa proses encoding dan decoding tidak selalu char/desimal ke biner atau biner ke char/desimal.

2.2.7.1 Encoder *base64*

Proses encoding dan decoding *base64* berasal dari jumlah tipe data yang dapat dikirim yaitu 64 tipe data. 64 tipe data berawal dari huruf kapital “A” sampai “Z”, huruf tidak kapital “a” sampai “z”, angka “0” sampai “9” dan 2 tambahan yaitu “+” dan “/” serta sebuah padding.

Index	Binary	Char	Index	Binary	Char	Index	Binary	Char	Index	Binary	Char
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u	62	111110	+
15	001111	P	31	011111	f	47	101111	v	63	111111	/

Gambar 2.14 List Base64

Proses encoding *base64* memiliki algoritma yang mengharuskan untuk seluruh data diubah menjadi 8-bit biner terlebih dahulu sebelum diolah oleh *encoder base64*. Algoritma encoding *base64* berawal dari mendapatkan data 8-biner dari data, sebagai contoh string “gajah”.

Tabel 2.1 Proses Encoding String ke Base64

Proses 1	gajah = 01100111 01100001 01101010 01100001 01101000
Proses 2	011001 110110 000101 101010 011000 010110 1000
Proses 3	011001 110110 000101 101010 011000 010110 100000
Proses 4	Z2FqYWg=

Proses 1 dari encoding base64 didapatkan biner 8-bit dari “gajah”. Proses 2 pola 8-bit biner itu diurutkan kembali menjadi pola 6-bit biner, dan menyisakan 4-bit biner dibagian akhir. Pola 4-bit biner ini akan dibuat menjadi 6-bit biner dengan menambahkan padding 2-bit biner “00” dan satu set padding . Perlu diketahui bahwa pengiriman encoding base64 pada proses 3 harus memiliki total kelipatan 8 dengan 1 set pola 6 - bit, dan setiap kali akan memasukkan padding harus menggunakan sebuah filler 8-bit hingga mendapat total kelipatan 8. Proses ketiga mendapatkan sebuah pola 6-bit biner dengan jumlah 7 set dengan tambahan 1 set padding, dengan total 48-bit biner. Pada proses 4 akan didapatkan “Z2FqYWg=” hasil dari encoding “gajah”.

2.2.7.2 Decoder *base64*

Data proses dencoding base64 harus melalui encoding base64 pula, apabila tidak output akan tidak dapat terbaca oleh sistem atau error. Proses decoding base64 memiliki proses yang hampir sama dengan proses encoding dengan membalik prosesnya.

Tabel 2.2 Proses Decoding Base64 ke String

Proses 1	Z2FqYWg=
Proses 2	011001 110110 000101 101010 011000 010110 100000
Proses 3	01100111 01100001 01101010 01100001 01101000
Proses 4	gajah

Proses 1 mengambil data base64 yaitu “Z2FqYWg=”. Pada proses 2 mengubah “Z2FqYWg=” menjadi 6-bit biner, dengan catatan “=” sebagai padding tidak diubah menjadi 6-bit biner. Proses 3 6-bit biner diurutkan kembali menjadi 8-bit biner dan menghasilkan 5 set 8-bit biner, dengan catatann biner “g” yang mendapat padding 2 biner terakhir dihapus agar dapat menjadi 8-bit biner dengan biner sebelumnya. Proses 4 akan didapat string “gajah” dari 5 set 8-bit biner.

2.2.7.3 Bitmap

Berbeda dengan ekstensi file gambar seperti .JPG atau .PNG yang memiliki fitur compres agar ukuran file dapat diperkecil, file bitmap atau .BMP memiliki ukuran yang lebih besar dikarenakan gambar tidak dicompres. Pixel pada bitmap yang membawa sebuah data atau warna, dan apabila dilakukan perubahan ukuran maka ukuran pixel juga ikut berubah bukan menggunakan penskalaan. berbeda dengan vektor yang menggunakan skala garis untuk membentuk gambarnya.



Gambar 2.15 Bitmap dan Vektor

Seperti pada gambar diatas , perbedaan gambar bitmap dan vektr terlihat jelas ketika dilakukan pembesaran. Gambar bitmap akan terlihat

pola pixel persegi yang membentuk lingkaran sedangkan gambar vektor akan tetap ketika dilakukan pembesaran.

Dengan tidak adanya fitur *compres* pada gambar bitmap membuat data atau *feature* yang ada pada gambar lebih terlihat dan dapat representasikan oleh komputer menggunakan pola sehingga proses perubahan data (*segmentasi*, *grayscale*) atau proses pengolahan citra dapat dilakukan dengan lebih baik..

2.2.8 Blob

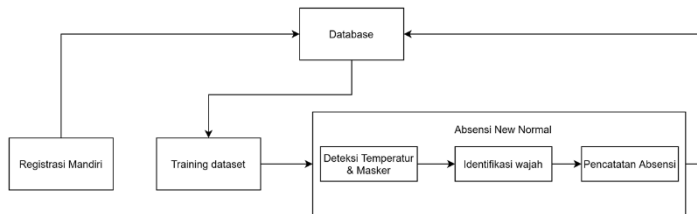
Binary Large Object atau disingkat Blob merupakan sebuah data dalam satu entitas yang dapat berupa gambar, data, *audio*, ataupun program. Blob sering kali ditemukan pada database atau didalam sebuah *runtime* program. Menurut Jim Starkey dalam bukunya yang berjudul “The True Story of BLOBs” blob merupakan sebuah file atau data yang besar yang tidak dapat dibaca oleh sebuah file manager.

Blob merupakan binary data berbeda dengan beberapa jenis file lainnya seperti *integers*, *floating point number*, *characters*, dan *strings*. Karakteristik blob salah satunya adalah ukuran file yang lebih besar dari pada ukuran asli dari file itu sendiri ketika masih berupa file yang file manager masih dapat membacanya seperti gambar, video, ataupun data lainnya.

BAB III

DESAIN SISTEM DAN IMPLEMENTASI

Bab desain sistem dan implementasi berisi tentang desain perangkat lunak dan perangkat keras. Perangkat lunak yang dibuat pada proyek akhir ini adalah deteksi penggunaan masker, deteksi temperature tubuh pada sistem absensi bagian implementasi protokol kesehatan, serta penginputan data diri melalui gawai pengguna untuk masukan data sistem absensi. Pada bab ini akan menjelaskan secara garis besar mengenai cara kerja dari sistem absensi pada bagian identifikasi wajah serta pencatatan data absensi. Arsitektur sistem absensi beserta dataflow juga akan dibahas pada bab ini.

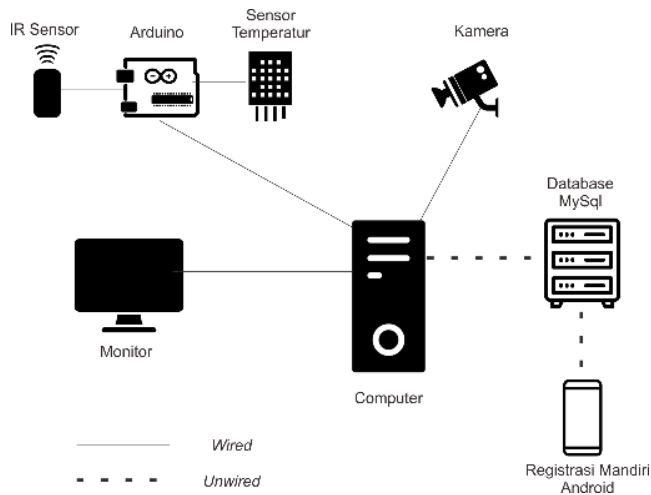


Gambar 3.1 Blok Diagram Sistem Absensi New Normal

Secara garis besar terdapat lima proses besar pada sistem absensi new normal ini. Fokus pembahasan paper ini akan berada pada bagian registrasi mandiri, dan performa sistem deteksi untuk proses absensi. Masing masing proses yang saya sampaikan sebelumnya dan komunikasi antar bagian ini akan dibahas lebih rinci pada bagian sub-bab desain sistem dan implementasi.

3.1 Arsitektur Sistem Absensi New Normal

Salah satu faktor yang dapat mempengaruhi performa sistem absensi new normal adalah perangkat keras yang digunakan. Pada proyek akhir ini sistem absensi new normal ini memiliki arsitektur menggunakan sebuah computer sebagai pengolahan data serta arduino sebagai controller dari sensor yang ada.



Gambar 3.2 Arsitektur Sistem Absensi New Normal

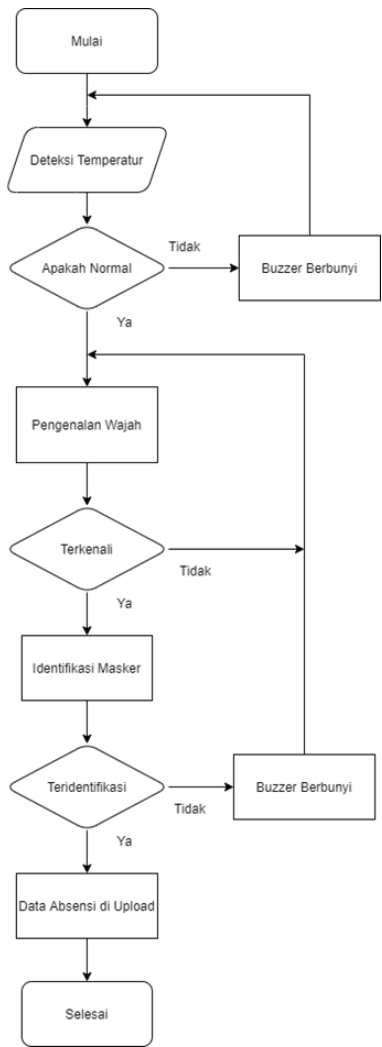
Pengolahan data serta komunikasi sistem absensi new normal dapat dikerjakan dengan lebih baik menggunakan komputer dikarenakan terdapat VGA yang membantu imageprocessing agar memiliki peforma yang lebih baik. Komunikasi dari aplikasi registrasi mandiri dengan database akan menggunakan tcp/ip.

3.2 Diagram Alir Proses Absensi New Normal

New normal adalah mengimplementasikan protokol kesehatan pada setiap aktivitas sehari – hari. Protokol kesehatan yang kita implementasikan kepada sistem absensi new normal ini adalah deteksi temperatur tubuh serta deteksi penggunaan masker. Sebagai upaya mengimplementasikan protokol kesehatan pada sistem absensi new normal yang kita buat, maka sebagai proses awal adalah proses deteksi temperatur tubuh dan masker.

Parameter yang digunakan sebagai acuan apakah pengguna boleh lanjut ke proses selanjutnya adalah penggunaan masker. Sesuai peraturan yang dikeluarkan pemerintah mengenai COVID–19, pemerintah mewajibkan untuk seluruh pelaku usaha, institusi, dan juga perusahaan yang melakukan kegiatan secara offline untuk memastikan temperatur setiap pekerja diatas 38C. Apabila pengguna absensi new

normal tidak memenuhi salah satu parameter protokol kesehatan yang ada nantinya absensi akan membunyikan buzzer dan tidak memperbolehkan untuk melakukan absensi.



Gambar 3.3 Diagram Alir Proses Absensi New Normal

Pengguna absensi new normal yang telah diperbolehkan melanjutkan proses setelah melakukan pengecekan protokol kesehatan akan melakukan proses absensi dengan menunjukkan wajah ke kamera dan sistem absensi new normal akan mencocokkan data yang adalah database. Apabila setelah melakukan proses pencocokan data dan pengguna teridentifikasi sebagai pegawai maka akan melakukan proses deteksi masker sebagai konfirmasi telah menyelesaikan proses absensi. Jika proses identifikasi wajah tidak berhasil dan sistem tidak dapat mengenali pengguna maka sistem absensi akan mengulangi kembali proses identifikasi sebanyak dua kali, apabila proses identifikasi tetap gagal atau sistem tidak dapat mengidentifikasi pengguna maka sistem absensi akan memberitahukan pengguna bahwa tidak dapat melanjutkan proses absensi dan kembali ke proses awal yaitu temperatur tubuh.

Dari kedua proses protokol kesehatan pada sistem absensi new normal, deteksi temperatur tubuh ditempatkan pada bagian pertama agar sistem tidak memakan banyak memori ketika sistem berada pada kondisi *idle*. Kondisi *idle* adalah kondisi ketika tidak ada pengguna yang melakukan proses absensi. Apabila proses deteksi masker ditempatkan pada proses pertama maka sistem akan menjalankan banyak proses deteksi masker yang membuat perangkat keras akan panas dan membuat sistem absensi berhenti berjalan dan mempengaruhi performa keseluruhan (akan dibahas pada bab IV). Dari permasalahan tersebut yang mendasari urutan proses awal pada sistem absensi new normal.

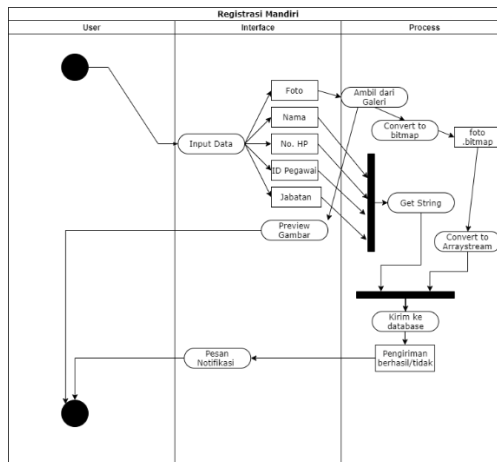
Selain permasalahan performa, alasan lain yang mendasari urutan proses absensi khususnya pada bagian awal yaitu deteksi temperatur tubuh dikarenakan adalah *looping* proses setelah melakukan absensi. Pada Gambar 3.4 diketahui bahwa proses akhir dari absensi adalah deteksi masker, sedangkan apabila deteksi masker ditempatkan pada awal proses maka akan terjadi proses *looping* dengan pengguna yang sama. Proses *looping* yang dimaksud adalah ketika pengguna sudah selesai melakukan absensi, sistem akan berjalan dari awal lagi dan apabila deteksi masker berada di awal, sistem absensi akan mengira bahwa sudah ada pengguna baru padahal masih pengguna yang sama.

3.3 Registrasi Mandiri

Proses penginputan data pengguna absensi dapat dilakukan melalui gawai masing masing pegawai. Penginputan data pengguna akan melalui sebuah aplikasi yang bernama registrasi mandiri. Pada aplikasi registrasi mandiri ini pegawai akan diwajibkan untuk memasukkan data diri yang meliputi nama lengkap, nomor pegawai, jabatan, nomor *handphone*, dan foto wajah. Data diri pegawai akan dikirim ke absensi melewati database menggunakan *TCP/IP*.

3.3.1 Aplikasi Registrasi Mandiri

Proses pembuatan aplikasi registrasi mandiri menggunakan android studio sebagai *IDE* dan java sebagai bahasa pemogramannya. Dalam proses pembuatan aplikasi registrasi mandiri terdapat sebuah *grandle* yang berguna sebagai *build-tool* tambahan apabila pada pemograman java tidak ada. Sebagai contoh *grandle* yang digunakan dalam pembuatan aplikasi registrasi mandiri ini adalah “mysql – mysql-connector-java” sebagai koneksi aplikasi dengan database, dan “com.google.android.gms:play-services-vision” sebagai machine learning kit.



Gambar 3.4 UML Aplikasi Registrasi Mandiri

Registrasi mandiri terdiri memiliki 5 input data diri diantaranya adalah input foto, input nama lengkap, input nomor *handphone*, input nomor pegawai, dan input jabatan. Pada Unified Modeling Diagram (UML) dari aplikasi registrasi mandiri, empat input selain foto akan dimasukkan kedalam variable string sebagai persiapan data untuk dikirim ke database. Input foto akan pecah menjadi dua bagian, bagian yang pertama akan menjadi dikembalikan ke interface awal aplikasi sebagai *image preview*. Bagian kedua foto yang berekstensi bitmap akan diubah menjadi *arraystream* oleh *base64* aplikasi. Ketika pengguna sudah memasukkan data diri beserta foto maka data bisa dikirim ke database ketika pengguna menekan tombol “kirim” pada bagian bawah gambar.

3.3.2 Base64 Encoder Aplikasi Registrasi Mandiri

Aplikasi registrasi mandiri menggunakan base64 dalam merubah data input (Nama lengkap, jabatan, id pegawai, nomor handphone, dan foto) sebelum dikirim ke database.

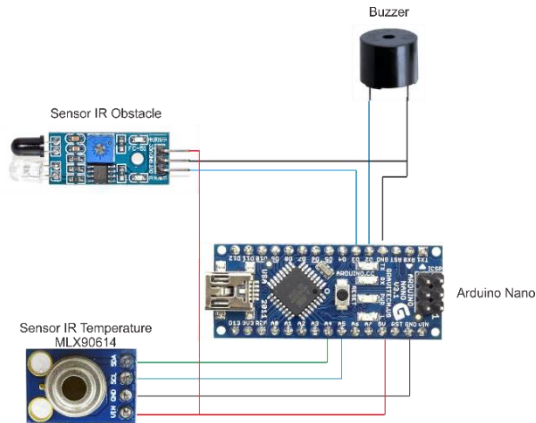
Tabel 3.1 Proses Encoder String to Base64 Aplikasi

	String	Base64
Nama	Muhammad Shalahuddin Al Ayyubi	TXVoYW1tYWQgU2hhbGFodWRkaW4gQWwgQXI5dWJp
Jabatan	Mahasiswa	TWFoYXNpc3dh
ID Pegawai	10311710000005	MTAzMTE3MTAwMDAwMDU=
Nomor HP	877346360770	ODc3MzQzMzYwNzcwCg==

Input nama lengkap, jabatan, id pegawai, nomor handphone yang berupa string akan diubah menggunakan base64 encoder seperti tabel diatas.

3.4 Protokol Kesehatan

Sistem absensi new normal mengimplementasikan protokol kesehatan dalam proses absensi seperti deteksi temperatur tubuh dan deteksi masker. Terdapat pula tujuan lain secara yang tidak langsung sejalan dengan protokol kesehatan yaitu *Sosial Distancing*, dengan memberikan sebuah sistem absensi dengan performa yang memadai maka tidak akan terjadi penumpukkan pengguna absensi.

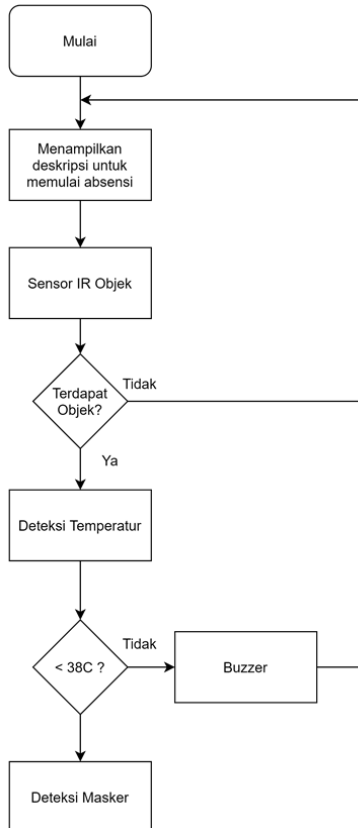


Gambar 3.5 Wiring Sistem Protokol Kesehatan Temperature Tubuh

Implementasi protokol kesehatan pada sistem absensi new normal menggunakan beberapa sensor *infrared* seperti sensor ir *obstacle* dan sensor ir *temperature* mlx9064 dan buzzer yang terhubung arduino nano sebagai *controller*. Bahasa pemrograman pada arduino nano menggunakan bahasa C, dan akan berkomunikasi dengan sistem absensi new normal yang berbasis bahasa pemrograman python menggunakan *serial communication*.

3.4.1 Temperatur Tubuh

Terdapat sebuah proses sebelum deteksi temperatur tubuh pada awal proses absensi new normal, yaitu proses deteksi objek yang dilakukan oleh *infrared* sensor. Proses deteksi objek ini berguna untuk memutus proses looping absensi jika deteksi masker berada diawal dan diakhir proses absensi. Arsitektur absensi new normal yang menggunakan *computer* sebagai pengolahan data dan arduino sebagai penghubung sensor *infrared* dan temperatur ke *computer*. Berikut adalah diagram alir dari sensor temperatur dan *infrared* :

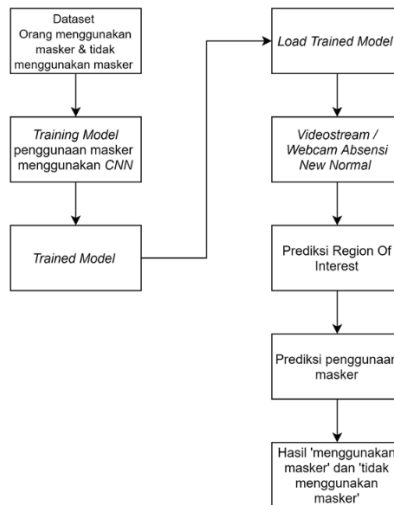


Gambar 3.6 Diagram Alir *Infrared* Sensor

Sistem absensi akan menampilkan sebuah kalimat yang mengindikasikan bahwa sistem absensi dapat digunakan. Ketika tidak ada objek yang terdeteksi pada sensor *infrared*, sistem absensi akan mengeluarkan kalimat indikasi terus menerus. Ketika sensor *infrared* mendeteksi objek maka sistem absensi akan memulai deteksi temperatur. Parameter temperatur pada absensi new normal adalah ketika pengguna diatas 38C , pengguna tidak diperbolehkan untuk melakukan absen dan seketika itu buzzer sistem absensi akan menyala. Ketika temperatur pengguna kurang dari 38C pengguna dapat masuk ke proses selanjutnya yaitu deteksi masker.

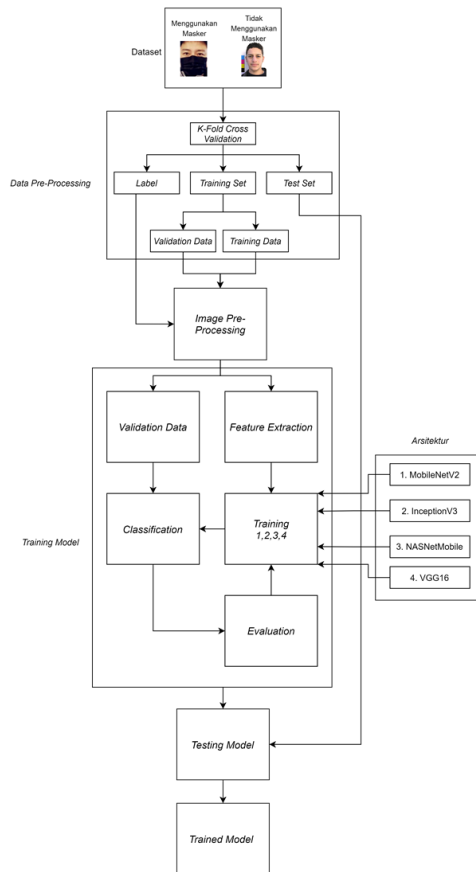
3.4.2 Deteksi Masker

Implementasi protokol kesehatan pada sistem absensi new normal selanjutnya adalah deteksi masker. Metode yang digunakan dalam untuk mendeteksi penggunaan masker pada wajah pada absensi new normal adalah *Convolution Neural Network*(CNN) dengan arsitektur *MobileNetV2*, *Nasnet*, *InceptionV3*, *VGG16*. Input dari proses CNN adalah *dataset* yang terdiri dari 100 foto wajah tanpa masker, dan 100 foto wajah menggunakan masker.



Gambar 3.7 Diagram Alir Pembuatan Deteksi Masker pada Absensi

Model hasil *training* menggunakan CNN akan dimasukkan kedalam sistem absensi new normal. Proses penentuan pengguna menggunakan masker atau tidak berada pada program absensi. Dimulai dari memprediksi *Region Of Interest* pada *videostream* absensi, setelah mendapat hasil yang diinginkan proses prediksi penggunaan masker berjalan dan menghasilkan klasifikasi apakah pengguna menggunakan masker atau tidak menggunakan masker.



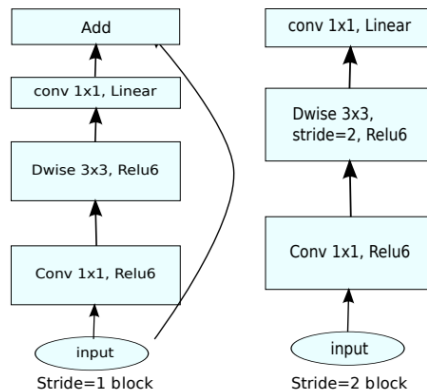
Gambar 3.8 Proses *Training* 4 Arsitektur Model Deteksi Masker

Pembuatan model CNN deteksi Masker dimulai dari pembagian *dataset* menjadi beberapa bagian diantaranya label (menggunakan masker dan tidak menggunakan masker), membagi gambar *dataset* menjadi *Training set* dan *Test set*. *Training set* akan dibagi menjadi dua yaitu *validation data* dan *training data* dimana *validation data* berguna untuk melakukan proses validasi pada prediksi model. *Training data* pada *training set* berguna untuk bahan belajar dari model CNN. Langkah berikutnya adalah *image pre-processing* dimana *training set* akan diubah

menjadi gambar ukuran 224x224 untuk arsitektur selain InceptionV3 dan 299x299 untuk arsitektur InceptionV3, masing masing gambar akan diberi label menggunakan masker atau tidak menggunakan masker sebagai acuan model CNN belajar.

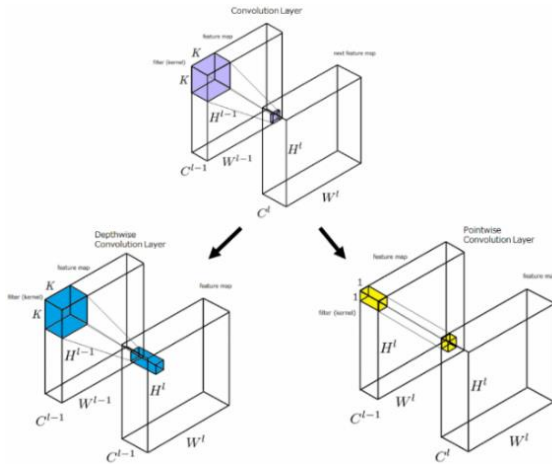
3.4.2.1 *Convulasi Neural Network Arsitektur MobileNetV2*

Proses *training* deteksi masker menggunakan beberapa arsitektur seperti *Convolution Neural Network(CNN) MobileNetV2*. *Dataset* yang telah terbagi menjadi dua bagian yaitu tanpa masker dan menggunakan masker akan akan menghasilkan sebuah model dari hasil *training*.



Gambar 3.9 Arsitektur *MobileNetV2*

MobileNetV2 memiliki proses *training* yang cukup berbeda dengan proses CNN lainnya dikarenakan *MobileNetV2* membagi proses konvolusi menjadi dua bagian yaitu *dephtwise convolution* dan *pointwise convolution*.



Gambar 3.10 Depthwise dan Pointwise Conv pada *MobileNetV2*

Depthwise convolution adalah proses konvolusi menggunakan satu kernel/filter pada masing masing channel input, sedangkan CNN pada umumnya melakukan konvolusi dengan seluruh kernel/filter ke input. Input dari pointwise convolution adalah output dari depthwise convolution yang disebut sebagai convolution layer. Convolution layer hasil dari depthwise convolution akan dikonvolusikan dengan seluruh filter/kernel 1×1 . Pointwise convolution akan menghasilkan sebuah fully connected layer dan kembali lagi ke convolution layer. Proses ini akan berjalan hingga seluruh input selesai diproses dan menghasilkan sebuah model yang berguna untuk deteksi masker pada sistem absensi new normal.

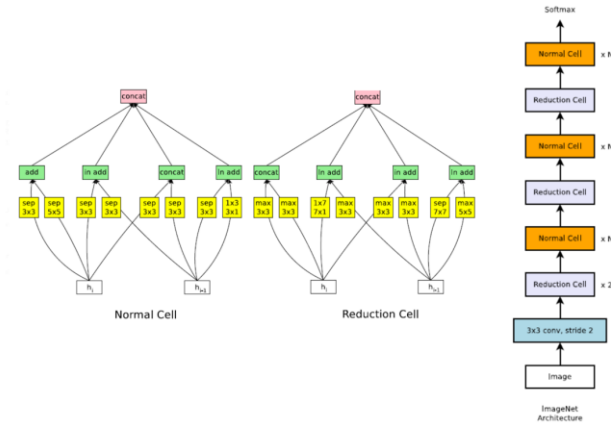
```
(classifier): Sequential(
  (0): Dropout(p=0.2, inplace=False)
  (1): Linear(in_features=1280, out_features=1000, bias=True)
```

Gambar 3.11 *Sequential MobileNetV2*

Sequential model merupakan sebuah model dengan urutan atau algoritma yang telah dibuat menjadi standard operasi dan dapat juga dapat diubah sesuai kebutuhan selama pilihan konfigurasi tersedia. Proses pembuatan deteksi masker pada absensi new normal menggunakan perintah beberapa setting yang merubah setting

sequential model dari *Dropout* = 0.2 menjadi *Dropout* = 0.5. Pembuatan model deteksi penggunaan masker menggunakan arsitektur MobileNetV2 dilakukan dua kali percobaan yaitu menggunakan *epoch* 30 dan *epoch* 100.

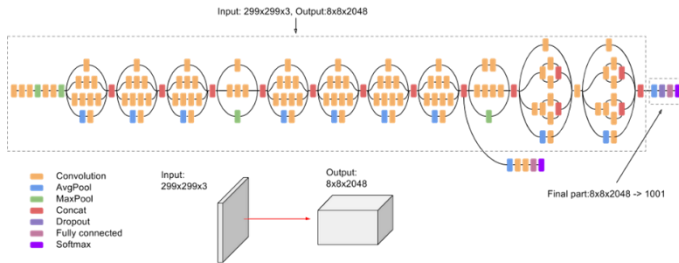
3.4.2.2 Convulusi Neural Network Arsitektur NASNetMobile



Gambar 3.12 Arsitektur NasNet

Convolution Neural Network arsitektur Nasnet adalah arsitektur CNN yang mengimplementasikan *reinforcement learning* pada proses *Training CNN* yang dikategorikan sebagai *supervised learning*. Nasnet melakukan memiliki proses inti yaitu *search space*, *search algorithm*, *evaluation strategy*. Proses *search space* dilakukan dengan melakukan proses konvolusi, maxpolling, dan fully connected) dalam satu kesatuan yang telah didefinisikan dalam setting awal. Proses *search algorithm* dilakukan dengan mencari proses *search space* dengan peforma terbaik dan menggunakan sistem reward ketika mesin dapat menemukan proses dengan peforma terbaik. Proses *search algorithm* ini adalah satu *reduction cell* dan satu *normal cell* yang diulang sebanyak n kali. *Evaluation strategy* adalah proses memprediksi menghitung pola pefroma terbaik untuk diimplementasikan. Pada arsitektur Nasnet proses *sequential* yang diubah adalah dari *Dropout* = 0.2 menjadi *Dropout* = 0.5 dan fitur yang dihasilkan adalah 1000 features.

3.4.2.3 Convolusi Neural Network Arsitektur InceptionV3



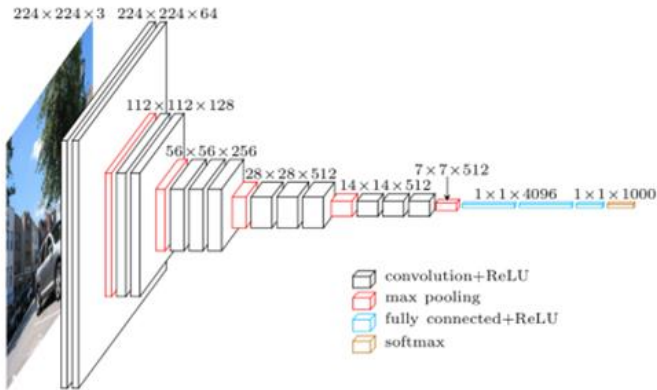
Gambar 3.13 Arsitektur InceptionV3

Pembuatan deteksi penggunaan masker menggunakan arsitektur InceptionV3 melakukan tiga kali proses convolution dengan kernel 3x3 dan dilakukan maxpooling dengan stride = 2 lalu dilakukan dua kali proses convolution dan maxpooling hingga menghasilkan data sebesar 64x64 dengan 80 channel.

Proses percabangan pada arsitektur InceptionV3 disebut proses InceptionA hingga InceptionE. Proses InceptionA merupakan 3 ranting pertama proses pada gambar arsitektur InceptionV3, setiap ranting melakukan sembilan kali proses konvolusi dengan input 64x64x192 pada awal InceptionA hingga 64x64x288 pada output InceptionA. Proses InceptionB hanya melakukan empat kali proses konvolusi dengan input 64x64x288 dan output 96x96x96. InceptionC memiliki 4 ranting dengan input 96x96x96 dan output 192x192x768. Lalu dilakukan proses InceptionAux dengan input 192x192x768 dan output 192x192x768. InceptionE memiliki 2 ranting yang didalamnya terdapat sembilan kali konvolusi dengan input 192x192x768 dan output 192x192x2048.

Proses terakhir yaitu proses klasifikasi dimana didalamnya terdapat proses *averagepolling*, *dropout*, *fully connected layer*, dan *softmax* yang menghasilkan data 1x1 dengan 1000 *features*. Data ini akan menjadi model pembandingan yang digunakan pada alat sistem absensi new normal.

3.4.2.4 *Convulusi Neural Network* Arsitektur VGG16



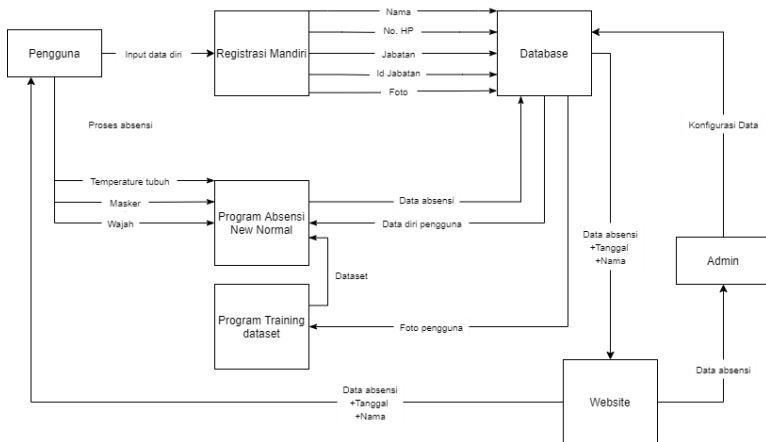
Gambar 3. 14 Arsitektur VGG16

Arsitektur VGG16 adalah salah satu arsitektur yang memiliki proses komputasi yang cukup berat jika dibandingkan dengan arsitektur lainnya. Proses komputasi yang berat ini dikarenakan pada arsitektur VGG16 memproses ekstraksi fitur hingga mendapat 4096 channel *features*, berbeda dengan arsitektur lainnya yang hanya 2048 channel atau 1000 channel saja. Pada proses *training* penggunaan masker menggunakan arsitektur VGG16 tidak ada proses sequential dari arsitektur yang diubah.

Arsitektur VGG16 melakukan proses konvolusi tiga kali diawal dan menghasilkan sebuah data ukuran 224x224 dengan 64 channel yang sebelumnya hanya 3 channel saja. Proses selanjutnya melakukan *max polling* hingga data menjadi 112x112 dengan 128 channel lalu dilakukan konvolusi sebanyak dua kali. Proses max polling dan dua kali konvolusi dilakukan hingga data menjadi 1x1 dengan 4096 channel. Lalu dilakukan fully connected+ReLU dan softmax pada proses akhir untuk mengubah 4096 *features* menjadi 1000 *features*.

3.5 Dataflow Sistem Absensi New Normal

Arsitektur sistem absensi new normal mengintegrasikan berbagai macam proses agar dapat berjalan sesuai rencana. Proses yang berjalan menggunakan berbagai macam interpreter serta perubahan data dan perpindahan data antar proses menggunakan berbagai macam metode. Dimulai dari aplikasi registrasi mandiri yang menggunakan java, pengiriman data diri menggunakan TCP/IP, fungsi penerimaan data dataase menggunakan PHP, bahasa pemograman C untuk sensor *infrared*, python sebagai penggerak sistem absensi new normal. Pada gambar 3.15 merupakan dataflow dari sistem absensi new normal.



Gambar 3.15 Dataflow Sistem Absensi New Normal

Perubahan dan perindahan data pada sistem absensi new normal dimulai dari input data diri pengguna melalui aplikasi registrasi mandiri. Input data diri foto akan dilabelkan dengan nama lengkap dan akan masuk ke proses *training dataset* untuk proses identifikasi wajah. Hasil *training* akan masuk ke program absensi new normal. Pengguna akan melakukan proses absensi dengan melakukan proses deteksi temperature tubuh , deteksi masker pada wajah, dan identifikasi wajah. Pencatatan absensi akan masuk ke database dan ditampilkan pada website. Pengguna dapat melihat riwayat absensi pada webiste.

BAB IV

PENGUJIAN DAN ANALISIS

Pada bab ini dijelaskan tahap-tahap uji coba berdasarkan desain sistem yang telah dibuat. Hasil uji coba absensi new normal dan registrasi mandiri akan dianalisis melalui berdasarkan waktu penggunaan sehingga dapat dilakukan evaluasi sistem absensi new normal mengacu pada protokol kesehatan. Hasil uji coba pada sistem arsitektur *Convolution Neural Network* deteksi masker akan dianalisis melalui proses klasifikasi gambar yang berhasil dideteksi dari *dataset* dan data percobaan sendiri sehingga dapat dilakukan evaluasi sistem.

4.1 Pengujian Sistem Absensi New Normal

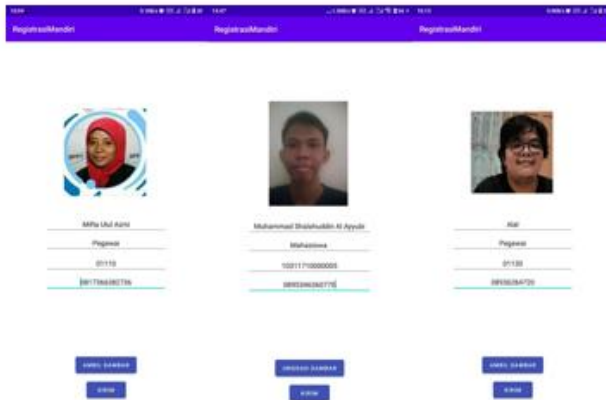
Data uji coba sistem absensi new normal dikumpulkan dari relawan yang diawali dengan memasukkan data diri melalui aplikasi registrasi mandiri yang meliputi nama lengkap, jabatan, id pegawai, nomor handphone, dan foto wajah. Berlanjut dengan proses absensi new normal dengan fokus waktu dalam satu kali proses absensi, lama waktu sistem temperatur tubuh dan deteksi masker absensi new normal.

4.1.1 Uji Coba Aplikasi Registrasi Mandiri

Uji coba aplikasi registrasi mandiri akan menggunakan 5 orang relawan yang akan menginputkan data diri diantaranya nama lengkap, jabatan, id pegawai, nomor *handphone*, dan foto.

4.1.1.1 Uji Coba Input Data Diri Pada Registrasi Mandiri

Data diri yang diperlukan sistem absensi new normal adalah nama lengkap, jabatan, id pegawai, nomor *handphone*, dan foto. Untuk input foto terbagi menjadi 2 cara yaitu dengan proses mengunggah dari gallery atau melakukan foto mandiri dari kamera.



Gambar 4.1 Sampel Dokumentasi Input Data Diri Pada Aplikasi Registrasi Mandiri

Aplikasi registrasi mandiri absensi new normal memiliki 4 kolom input yang digunakan untuk pengisian data diri. Terdapat 2 tombol yang berguna untuk memilih opsi pengambilan gambar dan tombol untuk mengirim data ke database.

Tabel 4.1 Uji Coba Pengiriman Data Diri Ke Dapatabase

No	Nama	Jabatan	ID Pegawai	Nomor Hp	Foto	Keterangan
1	Ayubi	Mahasiswa	10311710000072	000123458888	(Kamera)	Berhasil dikirim
2	Kahfi	Mahasiswa	10311710000005	000123456888	(Kamera)	Berhasil dikirim
3	Mifta	Pegawai	1113	000092629392	(Galeri)	Berhasil dikirim
4	Alal	Pegawai	1120	089384731204	(Galeri)	Berhasil dikirim
5	Kamila saadah	Akutansi	6769	089667563421	(Kamera)	Berhasil dikirim

Pada tabel diatas merupakan sampel data dari pengiriman data diri ke database. Dari ke 5 data yang ditampilkan, 2 diantaranya menggunakan kamera untuk mengambil foto wajah dan 3 diantaranya menggunakan galeri. Data sampel yang ditampilkan secara keseluruhan data ini dapat dikirim dan berhasil diterima oleh *database*.

+ Options									
+ - T →									
				Id	Nama	Foto	jabatan	nomor_pegawai	nomor_hp
<input type="checkbox"/>	Edit	Copy	Delete	52	Lila	[BLOB - 15.6 KiB]	HRD	117802033117	012345888
<input type="checkbox"/>	Edit	Copy	Delete	53	Kahli	[BLOB - 12.3 KiB]	Mahasiswa	10311710000072	0123456888
<input type="checkbox"/>	Edit	Copy	Delete	54	Ayubi	[BLOB - 12.7 KiB]	Mahasiswa	10311710000005	0123458888
<input type="checkbox"/>	Edit	Copy	Delete	55	Pak Joko	[BLOB - 15.1 KiB]	Dosen	1992202011021	0123456789
<input type="checkbox"/>	Edit	Copy	Delete	56	Pak Fauzi	[BLOB - 15.7 KiB]	Dosen	121991201711057	0123456788
<input type="checkbox"/>	Edit	Copy	Delete	57	Annisa Synthia	[BLOB - 14.7 KiB]	HRD	101711233053	0123456888
<input type="checkbox"/>	Edit	Copy	Delete	59	Meiza	[BLOB - 14.3 KiB]	Public Relation	10311710000072	0123456888
<input type="checkbox"/>	Edit	Copy	Delete	62	Mifta	[BLOB - 14.8 KiB]	pegawai	1113	092629392
<input type="checkbox"/>	Edit	Copy	Delete	65	kamila saadah	[BLOB - 17.5 KiB]	akutansi	6769	08967563421

Gambar 4.2 Hasil Pengiriman Data Diri dari Aplikasi Ke Database

Data diri yang dikirimkan dari aplikasi registrasi mandiri akan dibedakan menjadi 2 tipe data yaitu string dan blob. Data diri yang berupa string adalah nama lengkap, id pegawai, jabatan, dan nomor hp sedangkan untuk foto akan berubah menjadi .blob apabila masuk ke database. maka dari itu perlu base64 encoder pada proses aplikasi agar foto dapat diubah kembali menjadi foto sama seperti pengguna aplikasi mengirimkan foto.

4.2 Uji Coba Sistem Deteksi Masker

Pengujian sistem deteksi masker akan dilakukan dengan menganalisis performa dari 4 arsitektur CNN diantaranya MobileNetV2, NasNetMobile, InceptionV3, VGG16. Berikut adalah spesifikasi *hardware* yang digunakan untuk melakukan proses uji coba sistem absensi new normal :

Tabel 4.2 Spesifikasi *Hardware*

Hardware	Processor	AMD Athlon x4 640 (4 core / 4 thread 3.0 Ghz)
	RAM	4GB
	VGA	GTX 670 2GB
	Kamera	720P/30fps

Beberapa studi kasus yang dilakukan untuk mengukur keberhasilan masing masing arsitektur dalam mengklasifikasikan gambar orang tidak menggunakan masker dan tidak menggunakan masker pada *dataset*, mengklasifikasikan orang menggunakan masker 3 motif diantaranya masker bermotif warna dasar, masker bermotif batik,

dan masker bermotif bagian wajah serta pengaruh pencahayaan terhadap berjalannya sistem deteksi masker.

Tabel 4.3 Confusion Matrix Deteksi Penggunaan Masker

	Diprediksi Menggunakan Masker	Diprediksi Tidak Masker
Data Aktual Menggunakan Masker	TN	FP
Data Aktual Tidak Masker	FN	TP

Penilaian performa deteksi penggunaan masker pada 4 arsitektur akan menggunakan 4 parameter yaitu TP untuk gambar tidak menggunakan masker dan prediksi model benar, FN untuk gambar tidak menggunakan masker dan prediksi salah, TN untuk gambar menggunakan masker dan prediksi benar, FP untuk gambar menggunakan masker dan prediksi salah.

4.2.1 Dataset Uji Coba Sistem Deteksi Masker

Dataset uji coba sistem deteksi masker terbagi menjadi *dataset training* dan *dataset* uji coba. *Dataset training* dikumpulkan dari sebuah website penyedia *dataset* bernama kaggle.com, didalam *dataset* terbagi menjadi 2 yaitu orang menggunakan masker satu warna dan orang tidak menggunakan masker. *Dataset* uji coba dikumpulkan dari pembuat laporan yang menggunakan berbagai macam motif masker.

Tabel 4.4 *Dataset Training*

Jenis Gambar	Jumlah Gambar
Orang Menggunakan Masker	100
Orang Tidak Menggunakan Masker	100
Total	200

Tabel 4.5 *Dataset Percobaan*

Jenis Motif Masker	Jumlah Gambar
Masker Motif Bagian wajah	5
Masker Motif Warna Dasar	7
Masker Motif Acak	9
Masker Gas Respirator	4
Total	25

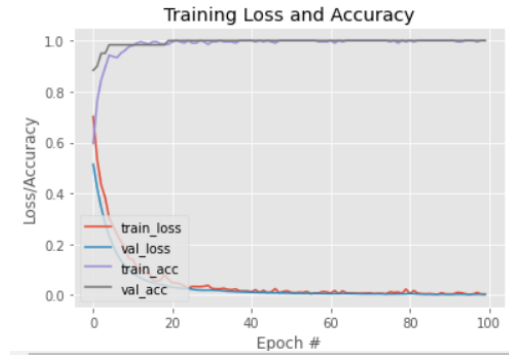
Dataset `percobaan terbagi menjadi 4 kategori yaitu masker warna dasar, masker motif acak, masker gas respirator. Sedangkan *dataset training* dibedakan menjadi dua yaitu orang menggunakan masker berwarna cerah dan orang tidak menggunakan masker

4.2.2 Peforma Deteksi Masker Arsitektur MobileNetV2

Uji coba peforma deteksi masker arsitektur MobileNetV2 dilakukan dengan melihat grafik proses *training* model, hasil *confusion matrix* menggunakan *dataset training*, *confusion matrix* menggunakan *dataset percobaan* dan FPS yang dihasilkan pada beberapa ukuran resolusi yang digunakan.

4.2.2.1 Peforma MobileNetV2 Pada *Dataset Training*

Proses *training* model dilakukan dengan parameter : *Batch Size* = 10, *Epoch* = 100. Dengan parameter tersebut dihasilkan grafik *training loss* dan *accuracy* sebagai berikut



Gambar 4.3 Grafik *Training Loss* dan *Accuracy* MobileNetV2 *Epoch* 100

Train_acc atau *training accuracy* didapatkan dari tingkat akurasi klasifikasi model *training* dengan *dataset training*. Val_acc atau *validation accuracy* didapatkan dari proses klasifikasi model *training* dengan data validasi. Train_loss atau *training loss* didapatkan dari tingkat error klasifikasi model *training* dengan *dataset training*, begitu juga dengan val_loss atau *validation loss*.

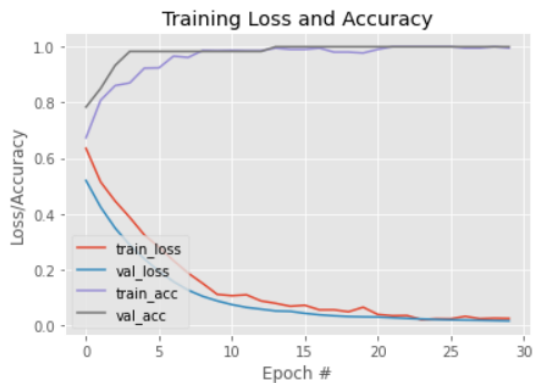
Berdasarkan grafik model *training* arsitektur MobileNetV2 *training accuracy* dan *validation accuracy* mulai berada dipuncak ketika *epoch* diantara 20 hingga 40 dan *training loss* dan *validation loss* mulai berada pada nilai terendah pada *epoch* 60 hingga 80.

```
True Negatives: 20
False Positives: 0
False Negatives: 0
True Positives: 40
```

Gambar 4.4 Nilai TN, FP, FN dan TP Arsitektur MobileNetV2 *Epoch* 100

Nilai *True Negatives*(TN), *False Negatives*(FN), *False Positives*(FP), dan *True Positives*(TP) didapatkan dari proses *testing*

akhir model *training* menggunakan *dataset testing* . Dari 60 *dataset training* yang digunakan didapatkan bahwa TN bernilai 20 yang mengindikasikan bahwa dari 20 gambar orang menggunakan masker model dapat memprediksi ke 20 gambar dengan benar. Sedangkan dari 40 gambar orang tidak menggunakan masker, model dapat memprediksi ke 40 gambar dengan benar.



Gambar 4.5 Grafik *Training Loss* dan *Accuracy* MobileNetV2 *Epoch* 30

Pada gambar diatas nilai *train_error* dan *val_loss* mulai menurun dengan seiring berjalannya *epoch*. Pada *epoch* ke 25 hingga 30 nilai *train_loss* dan *val_loss* mulai bergerak dinilai 0.0. *Train_acc* dan *val_acc* mulai bergerak naik ke batas maksimum pada *epoch* ke 5 dan hingga *epoch* 30.

True Negatives:	20
False Positives:	0
False Negatives:	0
True Positives:	40

Gambar 4. 6 Nilai TN, FP, FN dan TP Arsitektur MobileNetV2 *Epoch* 30

Tidak berbeda dengan model *training* arsitektur yang sama namun *epoch* yang berbeda, arsitektur MobileNetV2 menghasilkan hasil model *training* yang sama. Hal ini ini membuktikan bahwa dalam kasus deteksi penggunaan masker menggunakan arsitektur MobileNetV2,

menggunakan parameter *Batch Size* = 10, *Epoch* = 30 sudah cukup membuat model *training* dengan tingkat akurasi 100%.

Tabel 4.6 Peforma Arsitektur MobileNetV2 *Epoch* 30 dan *Epoch* 100 menggunakan *Dataset Training*

Arsitektur	Epoch	TN	TP	FN	FP	Accuracy	Precision	Recall	F1-Score
MobileNetV2	30	20	40	0	0	100,0	100,0	100,0	100,0
MobileNetV2	100	20	40	0	0	100,0	100,0	100,0	100,0

Peforma CNN arsitektur MobileNetV2 menghasilkan nilai yang sama dengan parameter *epoch* berbeda. Dengan *epoch* 30 memberikan komputasi lebih ringan dari pada *epoch* 100 dan memiliki nilai *training* yang lebih cepat dengan hasil *accuracy*, *precision*, *recall*, dan *F1-score* bernilai 100%.

4.2.2.2 K-Fold Cross Validation MobileNetV2 Pada Data Training

Berdasarkan 200 *dataset* yang dibagi menjadi 2 kategori yaitu menggunakan masker dan tidak menggunakan masker, *K-Fold Cross Validation* disetting sebesar 5. *Epoch* 30 dan *epoch* 50 saja yang akan menggunakan proses k-fold cross validation. Berikut adalah rincian dari hasil *K-Fold Cross Validation* pada arsitektur MobileNetV2 :

Tabel 4.7 *K-Fold Cross Validation* MobileNetV2 *Epoch* 30 dan 50

Epoch	Fold					Rata - Rata
	1	2	3	4	5	
30	98,3%	98,3%	100,0%	100,0%	100,0%	99,3%
50	100,0%	100,0%	98,3%	100,0%	100,0%	99,7%

Pada proses validasi menggunakan *k-flod cross validation* arsitektur MobileNetV2 *Epoch* 30 dan 50 didapatkan rata-rata akurasi 99,3% pada *epoch* 30 dan 99,7% pada *Epoch* 50. Rata – rata dari kedua *epoch* pada *K-Fold Cross Validation* didapatkan akurasi sebesar 99,5%.

4.2.2.3 Peforma MobileNetV2 Pada Dataset Percobaan

Berdasarkan percobaan penggunaan masker menggunakan *dataset* percobaan, didapatkan MobileNetV2 *Epoch* = 30 dan *Epoch* = 100 mendapatkan nilai 88%. Rincian hasil percobaan deteksi masker arsitektur MobileNetV2 menggunakan *dataset* percobaan ada pada tabel berikut :

Tabel 4.8 Rincian Uji Coba Deteksi Masker Arsitektur MobileNetV2

Jenis Motif Masker	Gambar ke -	MobileNetV2			
		Epoch = 30		Epoch = 100	
		Prediksi	Prediksi salah	Prediksi	Prediksi salah
Masker motif bagian wajah	1	0	1	0	1
	2	0	1	0	1
	3	0	1	0	1
	4	1	0	1	0
	5	1	0	1	0
Masker motif acak	1	1	0	1	0
	2	1	0	1	0
	3	1	0	1	0
	4	1	0	1	0
	5	1	0	1	0
	6	1	0	1	0
	7	1	0	1	0
Masker warna dasar	1	1	0	1	0
	2	1	0	1	0
	3	1	0	1	0
	4	1	0	1	0
	5	1	0	1	0
	6	1	0	1	0
	7	1	0	1	0
	8	1	0	1	0
	9	1	0	1	0
Masker gas respirator	1	1	0	1	0
	2	1	0	1	0
	3	1	0	1	0
	4	1	0	1	0
Rata - rata		88%	12%	88%	12%

Hasil akurasi yang didapatkan arsitektur MobileNetV2 *Epoch* 100 dan *epoch* 30 sebesar 88% dengan rincian dimana masing masing bagian tidak dapat melakukan prediksi dengan benar pada bagian maskeer motif bagian wajah. Dari 5 gambar masker motif bagian wajah, arsitektur MobileNetV2 *epoch* 100 dan *epoch* 30 hanya dapat memprediksi benar 2 gambar benar. Secara garis besar arsitektur MobileNetV2 dalam penggunaan deteksi masker, arsitektur ini dapat memprediksi benar 22 gambar dari 25 gambar *dataset* percobaan yang disediakan.

Tabel 4.9 Peforma Deteksi Masker Arsitektur MobileNetV2
Menggunakan *Dataset* Percobaan

Arsitektur	Epoch	TN	TP	FN	FP	Accuracy	Precision	Recall	F1-Score
MobileNetV2	30	22	25	0	3	94,0	88,0	100,0	93,6
MobileNetV2	100	22	25	0	3	94,0	88,0	100,0	93,6

Dilain sisi pada proses percobaan peforma arsitektur MobileNetV2 mendapat nilai rata rata 10,17 *Frame Per Second*(FPS) selama proses prediksi penggunaan masker, nilai fps ini merepresentasikan beban komputasi model *training* arsitektur dengan *unseen data* atau *videostream*. Peforma deteksi masker menggunakan *dataset* percobaan menghasilkan bahwa arsitektur MobileNetV2 *epoch* 100 dan 30 menghasilkan nilai yang sama mulai dari *accuracy* sebesar 94% , *precision* 88%, *recall* 100%, dan *f1-score* 93%.

4.2.2.4 Uji Coba Pengaruh Pencahayaan Terhadap Sistem Deteksi Masker Arsitektur MobileNetV2

Terdapat beberapa faktor yang dapat mempengaruhi kinerja dan hasil dari deteksi penggunaan masker salah satunya adalah pencahayaan. Pada bagian ini akan tampilkan hasil uji coba pengaruh pencahayaan terhadap berjalannya sistem deteksi penggunaan masker

Tabel 4.10 Pengaruh Pencahayaan Terhadap Sistem Deteksi
Penggunaan Masker Arsitektur MobileNetv2

No	Kondisi	Pencahayaan	Terdeteksi	Tidak Terdeteksi
1	Menggunakan masker	50 - 100 lux	0	1
2			0	1
3			0	1
4	Tidak Menggunakan Masker		0	1
5			0	1
6			0	1
1	Menggunakan masker	101 - 200 lux	1	0
2			1	0
3			1	0
4	Tidak Menggunakan Masker		1	0
5			1	0
6			1	0
1	Menggunakan masker	201 - 350 lux	1	0
2			1	0
3			1	0
4	Tidak Menggunakan Masker		1	0
5			1	0
6			1	0

Pada uji coba pencahayaan dengan intensitas cahaya 50 lux hingga 100 lux sistem tidak dapat mendeteksi wajah yang mengakibatkan proses deteksi penggunaan masker tidak dapat bekerja. Sedangkan diatas parameter tersebut hingga intensitas cahaya 350 lux, sistem dapat berjalan dengan normal seperti mendeteksi penggunaan masker dan mendeteksi tidak menggunakan masker.

4.2.2.5 Uji Coba Pengaruh Resolusi Terhadap *Frame Per Second* Arsitektur MobileNetV2

Komputasi yang dilakukan oleh sistem deteksi penggunaan masker dipengaruhi oleh frame atau resolusi yang digunakan, semakin besar resolusi yang digunakan maka input gambar yang harus diolah akan semakin besar pula begitupun sebaliknya apabila resolusi yang digunakan kecil. Maka dari itu bagian ini akan melakukan proses uji coba pengaruh besar kecilnya resolusi terhadap FPS yang didapat pada sistem deteksi penggunaan masker.

Tabel 4.11 Rincian Pengaruh Resolusi Terhadap *Frame Per Second* Deteksi Penggunaan Masker Arsitektur MobileNetV2

MobileNetV2			
No	Kondisi	Resolusi	Frame Per Second (FPS)
1	Menggunakan masker	426 x 240 (420p)	10,90
2			10,72
3			10,59
4	Tidak		10,54
5	Menggunakan Masker		10,90
6	Masker		10,99
1	Menggunakan masker	640 x 360 (360p)	10,76
2			10,10
3			10,97
4	Tidak		10,10
5	Menggunakan Masker		10,61
6	Masker		10,67
1	Menggunakan masker	854 x 480 (480p)	10,20
2			10,63
3			10,20
4	Tidak		10,63
5	Menggunakan Masker		10,00
6	Masker		10,14
1	Menggunakan masker	1280 x 720 (720p)	9,11
2			9,10
3			10,94
4	Tidak		9,11
5	Menggunakan Masker		10,41
6	Masker		10,64
1	Menggunakan masker	1920 x 1080 (1080p)	9,72
2			9,07
3			9,41
4	Tidak		9,49
5	Menggunakan Masker		9,31
6	Masker		9,04
Average FPS			10,17

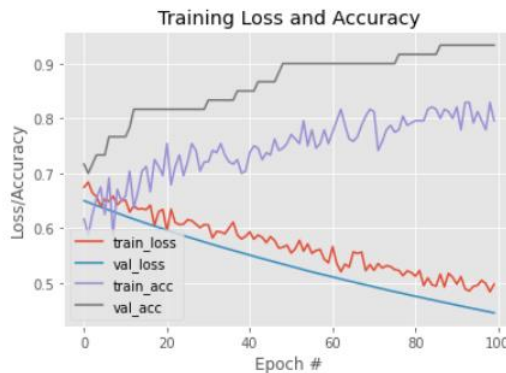
Didapatkan kesimpulan bahwa semakin kecil resolusi maka performa *Frame Per Second* (FPS) dari sistem deteksi penggunaan masker akan semakin besar. Pada arsitektur MobileNetV2 didapatkan rata-rata *Frame Per Second* (FPS) sebesar 10,17.

4.2.3 Peforma Deteksi Masker Arsitektur NASNetMobile

Uji coba peforma deteksi masker arsitektur NASNetMobile dilakukan dengan melihat grafik proses *training* model, hasil *confusion matrix* menggunakan *dataset training*, *confusion matrix* menggunakan *dataset percobaan* dan fps yang dihasilkan pada beberapa ukuran resolusi yang digunakan.

4.2.3.1 Peforma NASNetMobile Menggunakan Dataset Training

Proses *training* model dilakukan dengan parameter : *Batch Size* = 10, *Epoch* = 100 dan *Epoch* = 30. *Training* model dengan Parameter *Epoch* = 100 menghasilkan grafik *training loss* dan *acuracy* sebagai berikut



Gambar 4.7 Grafik *Training Loss* dan *Accuracy* NASNetMobile *Epoch* 100

Pada proses *training* deteksi penggunaan masker menggunakan arsitektur NASNetMobile memakan waktu rata-rata 30 detik per *epoch* dan untuk 100 *epoch* memerlukan waktu sekitar 50 menit. *Train_acc* dan *val_acc* masih berada disekitar 0,7 hingga 0.9 yang menandakan hasil yang tidak cukup bagus. *Train_loss* dan *val_loss* menunjukkan grafik

yang menurun dari *epoch* ke *epoch*. Model *training* arsitektur NASNetMobile ini menghasilkan *confusion matriks* sebagai berikut:

True Negatives:	16
False Positives:	4
False Negatives:	0
True Positives:	40

Gambar 4.8 Nilai TN, FP, FN dan TP Arsitektur NASNetMobile
Epoch 100

Dari 20 *dataset training* menggunakan masker, model *training* menggunakan arsitektur NASNetMobile berhasil mengklasifikasikan ke 16 *dataset* menggunakan masker dengan benar dan 4 gambar lainnya terdeteksi namun salah dalam proses klasifikasi. Sedangkan arsitektur ini berhasil mengklasifikasikan 40 gambar dari 40 gambar orang tidak memakai masker yang tersedia.



Gambar 4.9 Grafik *Training Loss* dan *Accuracy* NASNetMobile *Epoch* 30

Pada gambar diatas nilai *train_error* dan *val_loss* mulai menurun dengan seiring berjalannya *epoch*. Pada *epoch* ke 25 hingga 30 nilai *val_loss* mulai bergerak dinilai 0.0. *Train_acc* dan *val_acc* mulai bergerak naik ke batas maksimum pada *epoch* ke 10 dan hingga *epoch* 30. Model *training* arsitektur NASNetMobile ini menghasilkan *confusion matriks* sebagai berikut:

True Negatives:	20
False Positives:	0
False Negatives:	1
True Positives:	39

Gambar 4.10 Nilai TN, FP, FN dan TP Arsitektur NASNetMobile
Epoch 30

Dari 20 *dataset training* menggunakan masker, model *training* menggunakan arsitektur NASNetMobile berhasil mengklasifikasikan ke 20 *dataset* menggunakan masker dengan benar. Sedangkan arsitektur ini berhasil mengklasifikasikan 39 dari 40 gambar orang tidak memakai masker, dan 1 gambar tidak berhasil diklasifikasikan dan dideteksi.

Tabel 4.12 Peforma Arsitektur NASNetMobile *Epoch 30* dan *Epoch 100* menggunakan *Dataset Training*

Arsitektur	Epoch	TN	TP	FN	FP		Accuracy	Precision	Recall	F1-Score
NasNetMobile	30	20	39	1	0		98,3	100,0	97,5	98,7
NasNetMobile	100	16	40	0	4		93,3	80,0	100,0	88,9

Peforma CNN arsitektur NASNetMobile menghasilkan nilai yang berbeda tiap *setting parameter epoch*. Dengan *epoch 30* memberikan hasil yang lebih baik dengan nilai *F1-score* 98,7% dari pada dengan *epoch 100* yang hanya menghasilkan nilai 88,9%.

4.2.3.2 K-Fold Cross Validation NASNetMobile Pada Data Training

Berdasarkan 200 *dataset* yang dibagi menjadi 2 kategori yaitu menggunakan masker dan tidak menggunakan masker, *K-Fold Cross Validation* disetting sebesar 5. *Epoch 30* dan *epoch 50* saja yang akan menggunakan proses k-fold cross validation. Berikut adalah rincian dari hasil *K-Fold Cross Validation* pada arsitektur NASNetMobile:

Tabel 4.13 *K-Fold Cross Validation* NASNetMobile *Epoch 30* dan 50

Epoch	Fold					Rata - Rata
	1	2	3	4	5	
30	100,0%	100,0%	98,3%	100,0%	100,0%	99,7%
50	98,3%	100,0%	100,0%	98,3%	98,3%	99,0%

Pada proses validasi menggunakan *k-flod cross validation* arsitektur NASNetMobile *Epoch 30* dan 50 didapatkan ata – rata dari kedua *epoch* pada *K-Fold Cross Validation* didapatkan akurasi sebesar 99,3%.

4.2.3.3 Peforma NASNetMobile Menggunakan *Dataset Percobaan*

Berdasarkan percobaan penggunaan masker menggunakan *dataset* percobaan, didapatkan NASNetMobile *Epoch* = 30 dan *Epoch* = 100 mendapatkan nilai 84%.. Rincian peforma deteksi penggunaan masker arsitektur NASNetMobile akan ditampilkan pada tabel berikut :

Tabel 4.14 Rincian Uji Coba Peforma Deteksi Masker Arsitektur NASNetMobile

Jenis Motif Masker	Gambar ke -	NASNetMobile			
		Epoch = 30		Epoch = 100	
		Prediksi Benar	Prediksi salah	Prediksi Benar	Prediksi salah
Masker motif bagian wajah	1	0	1	0	1
	2	0	1	0	1
	3	0	1	0	1
	4	1	0	1	0
	5	1	0	1	0
Masker motif acak	1	1	0	0	1
	2	1	0	1	0
	3	1	0	1	0
	4	1	0	1	0
	5	1	0	1	0
	6	0	1	1	0
	7	1	0	1	0
Masker warna dasar	1	1	0	1	0
	2	1	0	1	0
	3	1	0	1	0
	4	1	0	1	0
	5	1	0	1	0
	6	1	0	1	0
	7	1	0	1	0
	8	1	0	1	0
	9	1	0	1	0
Masker gas respirator	1	1	0	1	0
	2	1	0	1	0
	3	1	0	1	0
	4	1	0	1	0
rata - rata		84%	16%	84%	16%

Arsitektur NASNetMobile pada deteksi masker masih belum bisa mendeteksi 3 dari 5 masker dengan motif bagian wajah dan 1 dari

7 masker motif acak. Secara garis besar dari 25 *dataset* percobaan arsitektur NASNetMobile dapat mendeteksi 21 dari 25 gambar orang menggunakan masker dan mendapat nilai akurasi 84%. *Frame Per Second* yang didapat disaat melakukan proses deteksi wajah menggunakan arsitektur NASNetMobile sebesar 7.19, yang menandakan bahwa saat melakukan proses deteksi penggunaan masker, sistem ini melakukan komputasi yang berat.

Tabel 4.15 Peforma Deteksi Masker Arsitektur NASNetMobile Menggunakan *Dataset* Percobaan

Arsitektur	Epoch	TN	TP	FN	FP		Accuracy	Precision	Recall	F1-Score
NASNetMobile	30	21	25	0	4		92,0	84,0	100,0	91,3
NASNetMobile	100	21	25	0	4		92,0	84,0	100,0	91,3

Peforma deteksi masker menggunakan *dataset* percobaan menghasilkan bahwa arsitektur NASNetMobile *epoch* 100 dan 30 menghasilkan nilai yang sama mulai dari *accuracy* sebesar 92% , *precision* 84%, *recall* 100%, dan *f1-score* 91,3%.

4.2.3.4 Uji Coba Pengaruh Pencahayaan Terhadap Sistem Deteksi Masker Arsitektur NASNetMobile

Terdapat beberapa faktor yang dapat mempengaruhi kinerja dan hasil dari deteksi penggunaan masker salah satunya adalah pencahayaan. Pada bagian ini akan tampilkan hasil uji coba pengaruh pencahayaan terhadap berjalannya sistem deteksi penggunaan masker.

Tabel 4.16 Pengaruh Pencahayaan Terhadap Sistem Deteksi Penggunaan Masker Arsitektur NASNetMobile

No	Kondisi	Pencahayaan	Terdeteksi	Tidak Terdeteksi
1	Menggunakan masker	50 - 100 lux	0	1
2			0	1
3			0	1
4	Tidak		0	1
5	Menggunakan Masker		0	1
6			0	1
1	Menggunakan masker	101 - 200 lux	1	0
2			1	0
3			1	0
4	Tidak		1	0
5	Menggunakan Masker		1	0
6			1	0
1	Menggunakan masker	201 - 350 lux	1	0
2			1	0
3			1	0
4	Tidak		1	0
5	Menggunakan Masker		1	0
6			1	0

Pada uji coba pencahayaan dengan intensitas cahaya 50 lux hingga 100 lux sistem tidak dapat mendeteksi wajah yang mengakibatkan proses deteksi penggunaan masker tidak dapat bekerja. Sedangkan diatas parameter tersebut hingga intensitas cahaya 350 lux, sistem dapat berjalan dengan normal seperti mendeteksi penggunaan masker dan mendeteksi tidak menggunakan masker.

4.2.3.5 Uji Coba Pengaruh Resolusi Terhadap *Frame Per Second* Arsitektur NASNetMobile

Komputasi yang dilakukan oleh sistem deteksi penggunaan masker dipengaruhi oleh frame atau resolusi yang digunakan, semakin besar resolusi yang digunakan maka input gambar yang harus diolah akan semakin besar pula begitupun sebaliknya apabila resolusi yang digunakan kecil.

Tabel 4.17 Rincian Pengaruh Resolusi Terhadap *Frame Per Second* Deteksi Penggunaan Masker Arsitektur NASNetMobile

NASNetMobile			
No	Kondisi	Resolusi	Frame Per Second (FPS)
1	Menggunakan masker	426 x 240 (420p)	7,78
2			7,83
3			7,11
4	Tidak Menggunakan Masker		7,81
5			7,27
6			7,79
1	Menggunakan masker	640 x 360 (360p)	7,68
2			7,18
3			7,2
4	Tidak Menggunakan Masker		7,12
5			7,78
6			7,67
1	Menggunakan masker	854 x 480 (480p)	7,6
2			7
3			7,68
4	Tidak Menggunakan Masker		6,67
5			6,89
6			6,79
1	Menggunakan masker	1280 x 720 (720p)	7,29
2			6,7
3			6,7
4	Tidak Menggunakan Masker		6,79
5			7,09
6			7,28
1	Menggunakan masker	1920 x 1080 (1080p)	6,87
2			6,7
3			6,84
4	Tidak Menggunakan Masker		6,71
5			6,82
6			6,97
Average FPS			7,19

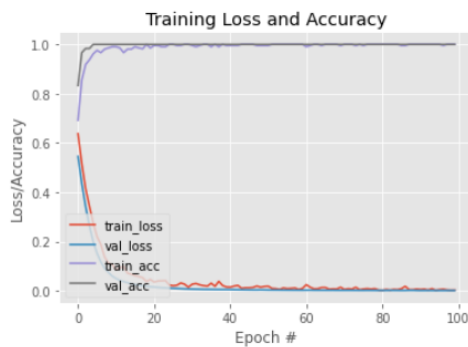
Didapatkan kesimpulan bahwa semakin kecil resolusi maka performa *Frame Per Second* (FPS) dari sistem deteksi penggunaan masker akan semakin besar. Pada arsitektur NASNetMobile didapatkan rata-rata *Frame Per Second* (FPS) sebesar 7,19 FPS.

4.2.4 Peforma Deteksi Masker Arsitektur InceptionV3

Uji coba peforma deteksi masker arsitektur InceptionV3 dilakukan dengan melihat grafik proses *training* model, hasil *confusion matrix* menggunakan *dataset training*, *confusion matrix* menggunakan *dataset percobaan* dan fps yang dihasilkan pada beberapa ukuran resolusi yang digunakan.

4.2.4.1 Peforma InceptionV3 Menggunakan Dataset Training

Proses *training* model dilakukan dengan parameter : *Batch Size* = 10, *Epoch* = 100 dan *Epoch* = 30. Pada proses *training* deteksi penggunaan masker menggunakan arsitektur InceptionV3 memakan waktu rata-rata 120 detik per *epoch* dan untuk 100 *epoch* memerlukan waktu sekitar 3 jam 20 menit. Dengan parameter tersebut dihasilkan grafik *training loss* dan *accuracy* sebagai berikut :



Gambar 4.11 Grafik *Training Loss* dan *Accuracy* InceptionV3 *Epoch* 100

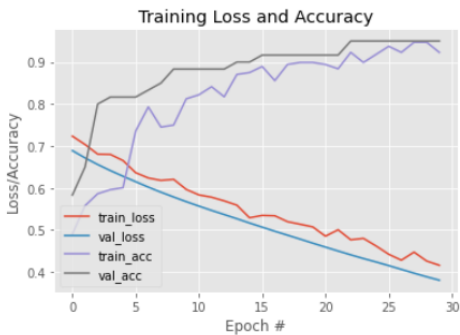
Train_acc dan val_acc sudah mulai bergerak dinilai maksimum pada *epoch* ke 40 serta mengalami perubahan yang minim ketika dibatas maksimum .Train_loss dan val_loss menunjukkan grafik yang menurun dari *epoch* ke *epoch* dan perubahan nilainya mulai minim pada *epoch* ke

80. Model *training* arsitektur InceptionV3 ini menghasilkan *confusion matriks* sebagai berikut:

True Negatives:	20
False Positives:	0
False Negatives:	0
True Positives:	40

Gambar 4.12 Nilai TN, FP, FN dan TP Arsitektur InceptionV3 *Epoch* 100

Dari 60 *dataset training* yang digunakan didapatkan bahwa TN bernilai 20 yang mengindikasikan bahwa dari 20 gambar orang menggunakan masker model dapat memprediksi ke 20 gambar dengan benar. Sedangkan dari 40 gambar orang tidak menggunakan masker, model dapat memprediksi ke 40 gambar dengan benar sama seperti hasil model *training* MobileNetV2.



Gambar 4.13 Nilai TN, FP, FN dan TP Arsitektur InceptionV3 *Epoch* 30

Pada proses *training* deteksi penggunaan masker menggunakan arsitektur InceptionV3 memakan waktu rata-rata 120 detik per *epoch* dan untuk 30 *epoch* memerlukan waktu sekitar 60 menit. Train_acc dan val_acc bergerak dengan dinamis namun masih mengalami kenaikan hingga *epoch* ke 30 meskipun masih belum menyentuh nilai 1 sebagai batas maksimal. Train_loss dan val_loss menunjukkan grafik yang menurun dari *epoch* ke *epoch* dan nilai yang didapatkan pada *epoch* ke

30 berada disekitar 0.4 hingga 0.5. Model *training* arsitektur InceptionV3 ini menghasilkan *confusion matriks* sebagai berikut:

True Negatives:	17
False Positives:	3
False Negatives:	0
True Positives:	40

Gambar 4.14 Nilai TN, FP, FN dan TP Arsitektur InceptionV3 *Epoch* 30

Dari 60 *dataset training* yang digunakan didapatkan bahwa TN bernilai 20 yang mengindikasikan bahwa dari 20 gambar orang menggunakan masker model dapat memprediksi ke 17 gambar dengan benar dan 3 gambar lainnya salah prediksi. Sedangkan dari 40 gambar orang tidak menggunakan masker, model dapat memprediksi ke 40 gambar dengan benar.

Tabel 4.18 Peforma Arsitektur InceptionV3 *Epoch* 30 dan *Epoch* 100 menggunakan *Dataset Training*

Arsitektur	Epoch	TN	TP	FN	FP		Accuracy	Precision	Recall	F1-Score
InceptionV3	30	17	40	0	3		95,0	85,0	100,0	91,9
InceptionV3	100	20	40	0	0		100,0	100,0	100,0	100,0

Peforma CNN arsitektur InceptionV3 menghasilkan nilai yang berbeda tiap *setting parameter epoch*. Dengan *epoch* 100 memberikan hasil yang lebih baik dengan nilai *F1-score* 100 % dari pada dengan *epoch* 30 yang hanya menghasilkan nilai 90,9%.

4.2.4.2 K-Fold Cross Validation InceptionV3 Pada Data Training

Berdasarkan 200 *dataset* yang dibagi menjadi 2 kategori yaitu menggunakan masker dan tidak menggunakan masker, *K-Fold Cross Validation* disetting sebesar 5. *Epoch* 30 dan *epoch* 50 saja yang akan menggunakan proses k-fold cross validation. Berikut adalah rincian dari hasil *K-Fold Cross Validation* pada arsitektur NASNetMobile:

Tabel 4.19 *K-Fold Cross Validation* InceptionV3 *Epoch* 30 dan 50

Epoch	Fold					Rata - Rata
	1	2	3	4	5	
30	98,6%	100,0%	93,6%	100,0%	100,0%	98,4%
50	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%

Pada proses validasi menggunakan *k-fold cross validation* arsitektur InceptionV3 *Epoch* 30 dan 50 didapatkan rata – rata dari kedua *epoch* pada *K-Fold Cross Validation* didapatkan akurasi sebesar 99,2%.

4.2.4.3 Peforma InceptionV3 Menggunakan Dataset Percobaan

Berdasarkan percobaan yang dilakukan terhadap deteksi masker arsitektur InceptionV3 menggunakan *dataset* percobaan dihasilkan nilai yang berbeda antara *Epoch* = 100 dengan *Epoch* = 30 dengan selisih 4% lebih tinggi di sisi *Epoch* = 100. Pada proses percobaan peforma arsitektur VGG16 mendapat nilai rata rata 6 *Frame Per Second*(FPS) selama proses prediksi penggunaan masker, nilai fps ini merepresentasikan beban komputasi model *training* arsitektur dengan *unseen data* atau *videostream*.

Tabel 4.20 Peforma Deteksi Masker Arsitektur InceptionV3

Jenis Motif Masker	Gambar ke -	InceptionV3			
		Epoch = 30		Epoch = 100	
		Prediksi Benar	Prediksi salah	Prediksi Benar	Prediksi salah
Masker motif bagian wajah	1	0	1	0	1
	2	0	1	0	1
	3	0	1	0	1
	4	0	1	0	1
	5	1	0	0	1
Masker motif acak	1	1	0	1	0
	2	1	0	1	0
	3	1	0	1	0
	4	1	0	1	0
	5	1	0	1	0
	6	1	0	1	0
	7	1	0	1	0
Masker warna dasar	1	1	0	1	0
	2	1	0	1	0
	3	1	0	1	0
	4	1	0	1	0
	5	1	0	1	0
	6	1	0	1	0
	7	1	0	1	0
	8	1	0	1	0
	9	1	0	1	0
Masker gas respirator	1	1	0	1	0
	2	1	0	1	0
	3	1	0	1	0
	4	1	0	1	0
rata - rata		84%	16%	80%	20%

Hasil yang didapatkan arsitektur InceptionV3 *Epoch* = 30 lebih tinggi 4% yang dihasilkan oleh prediksi benar pada masker kategori motif bagian wajah. Arsitektur InceptionV3 *Epoch* = 100 tidak berhasil memprediksi ke 5 gambar masker motif bagian wajah, sedangkan untuk *Epoch* = 30 berhasil melakukan prediksi benar pada 1 dari 5 gambar masker motif wajah.

Tabel 4.21 Peforma Deteksi Masker Arsitektur InceptionV3 Menggunakan *Dataset* Percobaan

Arsitektur	Epoch	TN	TP	FN	FP		Accuracy	Precision	Recall	F1-Score
InceptionV3	30	21	25	0	4		92,0	84,0	100,0	91,3
InceptionV3	100	20	25	0	5		90,0	80,0	100,0	88,9

Peforma deteksi masker menggunakan *dataset* percobaan menghasilkan bahwa arsitektur InceptionV3 *epoch* 30 menghasilkan nilai yang lebih baik dari pada *epoch* 100 dengan rincian *accuracy* sebesar 92% untuk *epoch* 30 lebih besar 2% dibandingkan *epoch* 100 , *precision* 84% untuk *epoch* 30 lebih besar 4% dibandingkan *epoch* 100, *recall* bernilai sama yaitu 100%, dan *f1-score* 91,3% untuk *epoch* 30 lebih besar 2,4% dibandingkan *epoch* 100.

4.2.4.4 Uji Coba Pengaruh Pencahayaan Terhadap Sistem Deteksi Masker Arsitektur InceptionV3

Terdapat beberapa faktor yang dapat mempengaruhi kinerja dan hasil dari deteksi penggunaan masker salah satunya adalah pencahayaan. Pada bagian ini akan tampilan hasil uji coba pengaruh pencahayaan terhadap berjalannya sistem deteksi penggunaan masker.

Tabel 4.22 Pengaruh Pencahayaan Terhadap Sistem Deteksi Penggunaan Masker Arsitektur InceptionV3

No	Kondisi	Pencahayaan	Terdeteksi	Tidak Terdeteksi
1	Menggunakan masker	50 - 100 lux	0	1
2			0	1
3			0	1
4	Tidak Menggunakan Masker		0	1
5			0	1
6			0	1
1	Menggunakan masker	101 - 200 lux	1	0
2			1	0
3			1	0
4	Tidak Menggunakan Masker		1	0
5			1	0
6			1	0
1	Menggunakan masker	201 - 350 lux	1	0
2			1	0
3			1	0
4	Tidak Menggunakan Masker		1	0
5			1	0
6			1	0

Pada uji coba pencahayaan dengan intensitas cahaya 50 lux hingga 100 lux sistem tidak dapat mendeteksi wajah yang mengakibatkan proses deteksi penggunaan masker tidak dapat bekerja. Sedangkan diatas parameter tersebut hingga intensitas cahaya 350 lux, sistem dapat berjalan dengan normal seperti mendeteksi penggunaan masker dan mendeteksi tidak menggunakan masker.

4.2.4.5 Uji Coba Pengaruh Resolusi Terhadap *Frame Per Second* (FPS) Arsitektur InceptionV3

Komputasi yang dilakukan oleh sistem deteksi penggunaan masker dipengaruhi oleh frame atau resolusi yang digunakan, semakin besar resolusi yang digunakan maka input gambar yang harus diolah akan semakin besar pula begitupun sebaliknya apabila resolusi yang digunakan kecil. Maka dari itu bagian ini akan melakukan proses uji coba pengaruh besar kecilnya resolusi terhadap FPS yang didapat pada sistem deteksi penggunaan masker.

Tabel 4.23 Rincian Pengaruh Resolusi Terhadap *Frame Per Second* Deteksi Penggunaan Masker Arsitektur InceptionV3

InceptionV3			
No	Kondisi	Resolusi	Frame Per Second (FPS)
1	Menggunakan masker	426 x 240 (420p)	4,08
2			3,95
3			3,18
4	Tidak		4,96
5	Menggunakan Masker		4,19
6			4,33
1	Menggunakan masker	640 x 360 (360p)	4,62
2			4,08
3			4,68
4	Tidak		4,37
5	Menggunakan Masker		6,07
6			4,26
1	Menggunakan masker	854 x 480 (480p)	4,76
2			5,78
3			6,04
4	Tidak		4,43
5	Menggunakan Masker		5,65
6			4,31
1	Menggunakan masker	1280 x 720 (720p)	3,54
2			5,29
3			5,34
4	Tidak		4,41
5	Menggunakan Masker		5,76
6			4,72
1	Menggunakan masker	1920 x 1080 (1080p)	3,81
2			6,31
3			3,57
4	Tidak		3,92
5	Menggunakan Masker		3,02
6			2,96
Average FPS			4,55

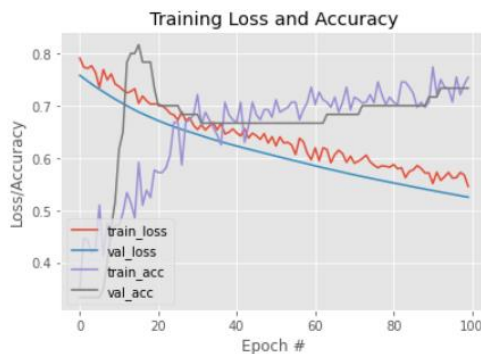
Didapatkan kesimpulan bahwa semakin kecil resolusi maka performa *Frame Per Second* (FPS) dari sistem deteksi penggunaan masker akan semakin besar. Pada arsitektur InceptionV3 didapatkan rata-rata *Frame Per Second* sebesar 4,55 FPS.

4.2.5 Peforma Deteksi Masker Arsitektur VGG16

Uji coba peforma deteksi masker arsitektur VGG16 dilakukan dengan melihat grafik proses *training* model, hasil *confusion matrix* menggunakan *dataset training*, *confusion matrix* menggunakan *dataset percobaan* dan fps yang dihasilkan pada beberapa ukuran resolusi yang digunakan.

4.2.5.1 Peforma VGG16 Menggunakan Dataset Training

Proses *training* model dilakukan dengan parameter : *Batch Size* = 10, *Epoch* = 100 dan *Epoch* = 30. Dengan parameter tersebut dihasilkan grafik *training loss* dan *accuracy* sebagai berikut



Gambar 4.15 Grafik *Training Loss* dan *Accuracy* VGG16 *Epoch* 100

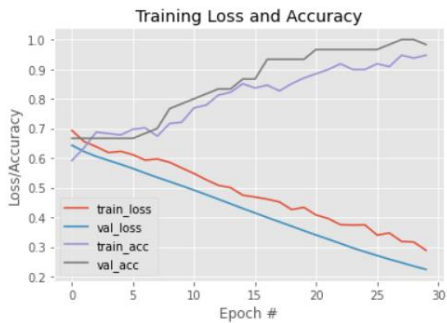
Pada proses *training* deteksi penggunaan masker menggunakan arsitektur VGG16 memakan waktu rata-rata 200 detik per *epoch* dan untuk 100 *epoch* memerlukan waktu sekitar 6 jam. Train_acc dan val_acc mengalami peningkatan drastis pada *epoch* 0 hingga 20 lalu mengalami penurunan sampai *epoch* ke 40. Pada *epoch* ke 60 train_acc mulai naik hingga ke nilai antara 0,7 hingga 0,8 pada *epoch* ke 100 .Train_loss dan val_loss menunjukkan grafik yang menurun dari *epoch* ke *epoch* namun belum menyentuh batas maksimum pada *epoch* ke 100.

Model *training* arsitektur VGG16 ini menghasilkan *confusion matriks* sebagai berikut:

True Negatives:	4
False Positives:	16
False Negatives:	0
True Positives:	40

Gambar 4.16 Nilai TN, FP, FN dan TP Arsitektur VGG16 *Epoch* 100

Dari 60 *dataset training* yang digunakan didapatkan bahwa TN bernilai 4 yang mengindikasikan bahwa dari 20 gambar orang menggunakan masker model hanya dapat memprediksi ke 4 gambar dengan benar dan 16 lainnya salah prediksi. Sedangkan dari 40 gambar orang tidak menggunakan masker, model dapat memprediksi ke 40 gambar dengan benar.



Gambar 4.17 Gambar 4.18 Grafik *Training Loss* dan *Accuracy* VGG16 *Epoch* 30

Pada proses *training* deteksi penggunaan masker menggunakan arsitektur VGG16 memakan waktu rata-rata 200 detik per *epoch* dan untuk 30 *epoch* memerlukan waktu sekitar 2 jam. Train_acc dan val_acc mengalami peningkatan secara berkesinambungan dari *epoch* 0 hingga 30 dan sempat menyentuh batas maksimal pada *epoch* diantara 25 sampai 30. Train_loss dan val_loss menunjukkan grafik menurun dari *epoch* ke *epoch* namun belum menyentuh batas maksimum pada *epoch* ke 30. Model *training* arsitektur VGG16 ini menghasilkan *confusion matriks* sebagai berikut:

```

True Negatives: 19
False Positives: 1
False Negatives: 0
True Positives: 40

```

Gambar 4.19 Nilai TN, FP, FN dan TP Arsitektur VGG16 *Epoch* 30

Dari 60 *dataset training* yang digunakan didapatkan bahwa TN bernilai 19 yang mengindikasikan bahwa dari 20 gambar orang menggunakan masker model dapat memprediksi ke 19 gambar dengan benar dan 1 gambar salah prediksi. Sedangkan dari 40 gambar orang tidak menggunakan masker, model dapat memprediksi ke 40 gambar dengan benar.

Tabel 4.24 Peforma Arsitektur VGG16 *Epoch* 30 dan *Epoch* 100 menggunakan *Dataset Training*

Arsitektur	Epoch	TN	TP	FN	FP	Accuracy	Precision	Recall	F1-Score
VGG16	30	19	40	0	1	98,3	95,0	100,0	97,4
VGG16	100	4	40	0	16	73,3	20,0	100,0	33,3

Peforma CNN arsitektur VGG16 menghasilkan nilai yang berbeda tiap *setting parameter epoch*. Dengan *epoch* 30 memberikan hasil yang lebih baik dengan nilai *F1-score* 97,4 % dari pada dengan *epoch* 100 yang hanya menghasilkan nilai 33,3%.

4.2.5.2 K-Fold Cross Validation VGG16 Pada Data Training

Berdasarkan 200 *dataset* yang dibagi menjadi 2 kategori yaitu menggunakan masker dan tidak menggunakan masker, *K-Fold Cross Validation* disetting sebesar 5. *Epoch* 30 dan *epoch* 50 saja yang akan menggunakan proses k-fold cross validation. Berikut adalah rincian dari hasil *K-Fold Cross Validation* pada arsitektur VGG16:

Tabel 4.25 *K-Fold Cross Validation* VGG16 *Epoch* 30 dan 50

Epoch	Fold					Rata - Rata
	1	2	3	4	5	
30	85,0%	80,0%	94,9%	91,6%	86,6%	87,9%
50	31,6%	25,0%	36,6%	53,3%	63,3%	42,0%

Pada proses validasi menggunakan *k-flod cross validation* arsitektur VGG16 *Epoch* 30 dan 50 didapatkan ata – rata dari kedua *epoch* pada *K-Fold Cross Validation* didapatkan akurasi sebesar 64,9%.

4.2.5.3 Peforma VGG16 Menggunakan *Dataset* Percobaan

Berdasarkan percobaan yang dilakukan terhadap deteksi masker arsitektur VGG16 menggunakan *dataset* percobaan dihasilkan nilai yang berbeda antara *Epoch* = 100 dengan *Epoch* = 30 dengan selisih 32% lebih tinggi di sisi *Epoch* = 100. Rincian peforma deteksi masker arsitektur VGG16 akan ada pada tabel berikut :

Tabel 4.26 Peforma Deteksi Masker Arsitektur VGG16

Jenis Motif Masker	Gambar ke -	VGG16			
		Epoch = 30		Epoch = 100	
		Prediksi Benar	Prediksi salah	Prediksi Benar	Prediksi salah
Masker motif bagian wajah	1	0	1	0	1
	2	0	1	0	1
	3	0	1	0	1
	4	0	1	0	1
	5	0	1	0	1
Masker motif acak	1	0	1	0	1
	2	0	1	0	1
	3	0	1	0	1
	4	0	1	0	1
	5	0	1	1	0
	6	0	1	0	1
	7	0	1	0	1
Masker warna dasar	1	0	1	1	0
	2	1	0	1	0
	3	0	1	1	0
	4	1	0	1	0
	5	1	0	1	0
	6	0	1	1	0
	7	0	1	1	0
	8	1	0	1	0
	9	1	0	1	0
Masker gas respirator	1	0	1	1	0
	2	0	1	1	0
	3	0	1	0	1
	4	0	1	1	0
Rata - rata		20%	80%	52%	48%

Deteksi masker arsitektur VGG16 *epoch* 30 hanya dapat memprediksi benar pada masker warna dasar dengan keterangan dari 9 gambar masker warna dasar , sistem ini hanya dapat melakukan prediksi benar sebanyak 4 gambar saja. Sedangkan pada arsitektur VGG16 *epoch* 100 berhasil memprediksi benar 13 gambar dari 25 gambar dengan keterangan masker warna dasar berhasil diprediksi benar secara keseluruhan, masker gas respirator 3 dari 4 gambar saja, masker motif acak berhasil memprediksi 1 dari 7 gambar, dan pada masker motif bagian wajah tidak berhasil memprediksi benar dari ke 5 gambar. Peforma deteksi masker arsitektur VGG16 ini hanya mencapai mendapat akurasi sebesar 20% pada *epoch* 30 dan 52% pada *epoch* 100. Pada proses percobaan peforma arsitektur VGG16 mendapat nilai rata – rata 5,21 *Frame Per Second*(FPS) selama proses prediksi penggunaan masker, nilai fps ini merepresentasikan beban komputasi model *training* arsitektur dengan *unseen data* atau *videostream*.

Tabel 4.27 Peforma Deteksi Masker Arsitektur VGG16 menggunakan *Dataset Percobaan*

Arsitektur	Epoch	TN	TP	FN	FP		Accuracy	Precision	Recall	F1-Score
VGG16	30	5	25	0	20		60,0	20,0	100,0	33,3
VGG16	100	13	25	0	12		76,0	52,0	100,0	68,4

Peforma deteksi masker menggunakan *dataset* percobaan menghasilkan bahwa arsitektur VGG16 *epoch* 100 menghasilkan nilai yang lebih baik dari pada *epoch* 30 dengan rincian *accuracy* sebesar 76% untuk *epoch* 100 lebih besar 16% dibandingkan *epoch* 30 , *precision* 52% untuk *epoch* 100 lebih besar 32% dibandingkan *epoch* 30, *recall* bernilai sama yaitu 100%, dan *f1-score* 68,4% untuk *epoch* 100 lebih besar 35.1% dibandingkan *epoch* 30.

4.2.5.4 Uji Coba Pengaruh Pencahayaan Terhadap Sistem Deteksi Masker Arsitektur VGG16

Terdapat beberapa faktor yang dapat mempengaruhi kinerja dan hasil dari deteksi penggunaan masker salah satunya adalah pencahayaan. Pada bagian ini akan tampilkan hasil uji coba pengaruh pencahayaan terhadap berjalannya sistem deteksi penggunaan masker.

Tabel 4.28 Pengaruh Pencahayaan Terhadap Sistem Deteksi Penggunaan Masker Arsitektur VGG16

No	Kondisi	Pencahayaan	Terdeteksi	Tidak Terdeteksi
1	Menggunakan masker	50 - 100 lux	0	1
2			0	1
3			0	1
4	Tidak		0	1
5	Menggunakan		0	1
6	Masker		0	1
1	Menggunakan masker	101 - 200 lux	1	0
2			1	0
3			1	0
4	Tidak		1	0
5	Menggunakan		1	0
6	Masker		1	0
1	Menggunakan masker	201 - 350 lux	1	0
2			1	0
3			1	0
4	Tidak		1	0
5	Menggunakan		1	0
6	Masker		1	0

Pada uji coba pencahayaan dengan intensitas cahaya 50 lux hingga 100 lux sistem tidak dapat mendeteksi wajah yang mengakibatkan proses deteksi penggunaan masker tidak dapat bekerja. Sedangkan diatas parameter tersebut hingga intensitas cahaya 350 lux, sistem dapat berjalan dengan normal seperti mendeteksi penggunaan masker dan mendeteksi tidak menggunakan masker.

4.2.5.5 Uji Coba Pengaruh Resolusi Terhadap *Frame Per Second* (FPS) Arsitektur VGG16

Komputasi yang dilakukan oleh sistem deteksi penggunaan masker dipengaruhi oleh frame atau resolusi yang digunakan, semakin besar resolusi yang digunakan maka input gambar yang harus diolah akan semakin besar pula begitupun sebaliknya apabila resolusi yang digunakan kecil. Maka dari itu bagian ini akan melakukan proses uji coba pengaruh besar kecilnya resolusi terhadap FPS yang didapat pada sistem deteksi penggunaan masker.

Tabel 4.29 Rincian Pengaruh Resolusi Terhadap *Frame Per Second* Deteksi Penggunaan Masker Arsitektur InceptionV3

VGG16			
No	Kondisi	Resolusi	Frame Per Second (FPS)
1	Menggunakan masker	426 x 240 (420p)	5,28
2			6,07
3			5,13
4	Tidak		5,31
5	Menggunakan Masker		5,89
6			5,8
1	Menggunakan masker	640 x 360 (360p)	4,32
2			5,76
3			4,31
4	Tidak		5,45
5	Menggunakan Masker		5,02
6			4,87
1	Menggunakan masker	854 x 480 (480p)	5,21
2			4,45
3			5,71
4	Tidak		4,87
5	Menggunakan Masker		4,31
6			5,49
1	Menggunakan masker	1280 x 720 (720p)	5,97
2			5,4
3			6,18
4	Tidak		6,04
5	Menggunakan Masker		6,07
6			5,56
1	Menggunakan masker	1920 x 1080 (1080p)	4,91
2			4,23
3			4,96
4	Tidak		4,72
5	Menggunakan Masker		4,74
6			4,33
Average FPS			5,21

Didapatkan kesimpulan bahwa semakin kecil resolusi maka performa *Frame Per Second* (FPS) dari sistem deteksi penggunaan masker akan semakin besar. Pada arsitektur VGG16 didapatkan rata-rata *Frame Per Second* sebesar 5,21 FPS.

4.3 Peforma Keseluruhan Sistem Absensi New Normal

Penilaian peforma keseluruhan dari sistem absensi new normal akan dievaluasi berdasarkan lama waktu dari proses absensi. Pada uji coba proses absensi ini akan melibatkan 5 orang yang akan melakukan 5 kali proses absensi. Berikut adalah rincian dari hasil lama waktu dari masing masing percobaan proses absensi :

Tabel 4.30 Rincian Uji Coba Lama Waktu Proses Absensi

Subject No.	Pengujian Ke - 1	Pengujian Ke - 2	Pengujian Ke - 3	Pengujian Ke - 4	Pengujian Ke - 5	Rata - Rata
1	23,40	25,60	22,40	28,50	27,40	25,46
2	28,60	27,30	28,20	25,20	25,60	26,98
3	24,40	24,60	23,20	24,90	24,50	24,32
4	26,50	24,10	22,40	23,90	23,20	24,02
5	25,50	24,10	24,30	26,20	23,70	24,76
Rata Rata total						25,11

Pada pengujian lama waktu proses absensi, terdapat 5 subjek yang melakukan 5 kali percobaan proses absensi. 5 *subject* ini adalah adek pertama kahfi, adek kedua kahfi, kahfi, saudara perempuan ayubi, dan ayubi. Didapatkan dalam hasil pengujian lama waktu dari proses absensi ini rata rata selama 25,11 detik.

4.4 Evaluasi Sistem Deteksi Masker

Dari seluruh hasil uji coba deteksi penggunaan masker *dataset training* dan *dataset* percobaan pada ke-4 arsitektur didapatkan data peforma sebagai berikut :

Tabel 4.31 Peforma 4 Arsitektur Pada *Dataset Training*

Arsitektur	Epoch	Dataset Training					Peforma Arsitektur
		Accuracy	Precision	Recall	F1-Score	K-Fold CV	
MobileNetV2	30	100,0%	100,0%	100,0%	100,0%	99,3%	99,75%
	100	100,0%	100,0%	100,0%	100,0%	99,7%	
NASNetMobile	30	98,3%	100,0%	97,5%	98,7%	99,7%	96,58%
	100	93,3%	80,0%	100,0%	88,9%	99,0%	
InceptionV3	30	95,0%	85,0%	100,0%	91,9%	98,4%	97,58%
	100	100,0%	100,0%	100,0%	100,0%	100,0%	
VGG16	30	98,3%	95,0%	100,0%	97,0%	87,9%	65,05%
	100	73,0%	20,0%	100,0%	33,3%	42,0%	

Tabel 4.32 Peforma 4 Arsitektur Pada *Dataset Percobaan*

Arsitektur	Epoch	Dataset Percobaan				Average FPS
		Accuracy	Precision	Recall	F1-Score	
MobileNetV2	30	94,0%	88,0%	100,0%	93,6%	10,2
	100	94,0%	88,0%	100,0%	93,6%	
NASNetMobile	30	92,0%	84,0%	100,0%	91,3%	7,2
	100	92,0%	84,0%	100,0%	91,3%	
InceptionV3	30	92,0%	84,0%	100,0%	91,3%	4,6
	100	90,0%	80,0%	100,0%	88,9%	
VGG16	30	60,0%	20,0%	100,0%	33,0%	5,2
	100	76,0%	52,0%	100,0%	68,4%	

Peforma ke-4 arsitektur pada *dataset training* didapatkan bahwa nilai akurasi tertinggi yaitu 100% pada InceptionV3 *epoch* 100, MobileNetV2 *epoch* 30 & 100 dan berlaku juga pada nilai *f1-score*. Pada *K-Fold Cross Validation* nilai tertinggi dimiliki oleh inceptionV3 *epoch* 100 dengan nilai 100% dan disusul oleh NASNetMobile *epoch* 30 dan MobileNetV2 *epoch* 100 dengan nilai 99,7%. Maka nilai akhir dari peforma arsitektur yang didapatkan dari rata rata *f1-score* dan *K-Fold Cross Validation* dihasilkan bahwa arsitektur MobileNetV2 memiliki nilai tertinggi sebesar 99,75%.

Pada *dataset* percobaan peforma ke- 4 aarsitektur dinilai dengan *f1-score* dan *Frame Per Second*(FPS) sebagai acuan peforma. Hubungan FPS dengan peforma ini adalah beban komputasi yang berimbas ke waktu yang dibutuhkan untuk absensi. Dari ke-4 arsitektur MobileNetV2 memiliki nilai *f1-score* tertinggi dengan nilai 93,6% dan FPS tertinggi rata – rata 10,2 FPS.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah melakukan perancangan dan pembuatan absensi new normal serta melalui uji coba yang berfokus pada performa arsitektur pada deteksi penggunaan masker yang telah dilakukan maka dapat diambil kesimpulan antara lain :

- a. Deteksi penggunaan masker pada *dataset* percobaan menggunakan arsitektur MobileNetV2 menghasilkan akurasi prediksi tertinggi diantara arsitektur lainnya sebesar 94% sedangkan arsitektur VGG16 mendapat nilai akurasi terendah sebesar 60%.
- b. Penambahan *epoch* pada masing masing arsitektur tidak memastikan dapat meningkatkan nilai *f1-score* sebagai contoh pada arsitektur VGG16 tidak mengalami peningkatan nilai *f1-score* pada penambahan *epoch*, dan juga pada arsitektur MobileNetV2 penambahan *epoch* tidak memberikan perubahan nilai *f1-score*.
- c. Hasil uji coba deteksi penggunaan masker pada *dataset training* diperoleh nilai *f1-score* terendah pada arsitektur VGG16 sebesar 33% dan tertinggi pada arsitektur MobileNetV2 sebesar 100%.
- d. Penggunaan *K-Fold Cross Validation* pada proses *training* data deteksi penggunaan masker ke-4 arsitektur CNN memberikan nilai akurasi lebih spesifik hingga 10% jika dibandingkan tanpa menggunakan *k-fold cross validation*.
- e. Hasil uji coba deteksi penggunaan masker menggunakan *dataset* percobaan pada sistem absensi new normal didapatkan dari ke empat arsitektur nilai tertinggi di diperoleh oleh MobileNetV2 dengan akurasi 94,0% dan f1-score sebesar 93,6% sedangkan nilai terendah diperoleh oleh VGG16 dengan akurasi 60% dan f1-score sebesar 33,0%.
- f. Hasil uji coba deteksi penggunaan masker menggunakan *dataset* percobaan pada sistem absensi new normal ke empat arsitektur didapatkan nilai tertinggi *Frame Per Second* pada arsitektur MobileNetV2 dengan rata-rata 10,17fps dan nilai terendah pada arsitektur InceptionV3 dengan rata rata 4,6fps.

- g. Pada hasil pengujian lama waktu yang diperlukan dalam satu kali proses absensi menggunakan absensi new normal, didapatkan rata rata untuk satu kali proses absensi memakan waktu selama 25,11 detik.

5.2 Saran

Sistem absensi new normal dapat dikembangkan pada aspek deteksi cuci tangan dan deteksi jarak pada proses absensi. Dengan pengembangan ini harapannya dapat menghasilkan sebuah sistem absensi yang mengimplementasikan protokol kesehatan lebih menyeluruh sebagai upaya mengurangi penyebaran virus COVID-19.

DAFTAR PUSTAKA

- [1] W. H. Organization *et al.*(2020). *Coronavirus disease 2019 (Covid-19): situation report*, 96.
- [2] Y. Fang, Y. Nie, and M. Penny. (2020). *Transmission dynamics of the Covid-19 outbreak and effectiveness of government interventions: A data-driven analysis*. *Journal of medical virology*, vol. 92, no. 6, pp. 645–659.
- [3] N. H. Leung, D. K. Chu, E. Y. Shiu, K.-H. Chan, J. J. McDevitt, B. J. Hau, H.-L. Yen, Y. Li, D. KM, J. Ip *et al.* *Respiratory virus shedding in exhaled breath and efficacy of face masks*.
- [4] Zufar Muhammad.(2016). *Convolutional Neural Netwrok Untuk Pengenalan Wajah Secara Real-Time*, Institut Teknologi Sepuluh Nopember.
- [5] M. Khalil Hani. (2014). *Convolutional Neural Network for Face Recognition with Pose and Illumination Variation*. VeChad Research Laboratory, Faculty of Electrical Engineering, Universiti Teknologi Malaysia.
- [6] Mas’udi Wawan. *Tata Kelola Penganan Covid-19 di Indonesia Kajian Awal* , Universitas Gajah Mada
- [7] Aarsal Muhammad. (2020). *Face Recognition untuk Akses Pegawai Bank Menggunakan Deeplearning Dengan metode CNN*. Universitas Gunadarma.
- [8] PP No. 21 Tahun 2020 tentang Pembatasan Sosial Berskala Besar dalam Rangka Percepatan Penanganan Corona Virus Disease 2019 (COVID-19).
- [9] Surat Edaran No. HK.02.01/MENKES/335/2020 tentang Protokol Pencegahan Penularan Corona Virus Disease (COVID-19) di Tempat Kerja Sektor Jasa dan Pedagangan (Area Publik) dalam Mendukung Keberlangsungan Usaha.
- [10] Catherine Ley. (2020). *Decrasing human body temperature in United States since Industrial Revolutions*. School of Medicine, Stanford University.
- [11] Meenpal, T., Balakrishnan, A., & Verma, A. (2019). *Facial Mask Detection using Semantic Segmentation*. *2019 4th International Conference on Computing, Communications and Security*.
- [12] Yusuf, Eddy. “2019 Novel Coronavirus Disease (Covid-19): Thermal Imaging System for Covid-19 Symptom Detection Using Iot Technology”. *DE CLÍNICA PSICOLÓGICA, REVISTA ARGENTINA*.

- [13] Zahagir, Alom. "Computer Vision Based Employee Activities Analysis". University of Dayton.
- [14] Su, Hang, Shaogang Gong, Xiatian Zhu, Adrian Popescu, Alexandru Ginsca, Herve Le Borgne, Yuen Peng Loh et al . (2015). *WebLogo-2M: Scalable Logo Detection by Deep Learning From the Web*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
- [15] Nasrollahi, Kamal, Sergio Escalera, Pejman Rasti, Gholamreza Anbarjafari, Xavier Baro, Hugo Jair Escalante, and Thomas B. Moeslund. (2015). *Deep learning based super-resolution for improved action recognition*. In Image Processing Theory, Tools and Applications (IPTA), International Conference, IEEE.
- [16] Joulan K, Brémond R, Hautière N (2015). *Towards an Analytical Age-Dependent Model of Contrast Sensitivity Functions for an Ageing Society*. The Scientific World Journal.
- [17] Sepritahara (2012). *Sistem Pengenalan Wajah Menggunakan Metode Hidden Markov Model (HMM)*. Program Studi Teknik Elektro, Fakultas Teknik Universitas Indonesia.
- [18] Barret Zoph, Quoc V. Le. *Neural architecture search with reinforcement learning*. ICLR 2017.
- [19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens. 2016. *Rethinking the Inception Architecture for Computer Vision*. University College London. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [20] Karen Simonyan, Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. University of Oxford, UK.
- [21] Marufur Rahman, Hossen Manik, Milon Islam (2020). *An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network*. IEEE.
- [22] David Sussilo, L.F Abbott.(2014). Random Walk Initialization for Training Very Deep Feedforward Convolution. ICLR 2015.

LAMPIRAN