

**SISTEM DETEKSI MASKER PADA WAJAH MENGGUNAKAN
CONVOLUTIONAL NEURAL NETWORK ARSITEKTUR VGG16**

LAPORAN TUGAS AKHIR

Laporan ini Disusun untuk Memenuhi Salah Satu Syarat Memperoleh Gelar
Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika Fakultas Teknologi
Industri Universitas Islam Sultan Agung Semarang



DISUSUN OLEH :

MOHAMMAD USHULUDIN

NIM 32601601059

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2023

FINAL PROJECT

***FACE MASK DETECTION SYSTEM USING CONVOLUTIONAL NEURAL
NETWORK ARCHITECTURE VGG16***

*Proposed to complete the requirement to obtain a bachelor's degree (S-1) at
Informatics Engineering Department of Industrial Technology Faculty Sultan
Agung Islamic University*



ARRANGED BY :

MOHAMMAD USHULUDIN

NIM 32601601059

**MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG**

2023

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul “Sistem Deteksi Masker Pada Wajah Menggunakan *Convolutional Neural Network* Arsitektur *VGG16*” ini disusun oleh :

Nama : Mohammad Ushuludin

NIM : 32601601059

Program Studi : Teknik Informatika

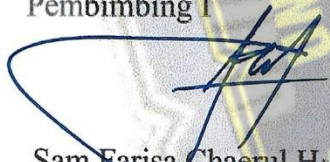
Telah disahkan oleh dosen pembimbing pada :

Hari : Senin

Tanggal : 21 Agustus 2023

Mengesahkan,

Pembimbing I



Sam Farisa Chaerul H; ST, M.Kom
NIDN. 0628028602

Pembimbing II



Imam Much Ibnu S, ST, M.Sc, Ph.D
NIDN. 0613037301

Mengetahui,

Ketua Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Islam Sultan Agung




Ir. Sri Mulyono, M.Eng
NIDN. 0626066601

LEMBAR PENGESAHAN PENGUJI

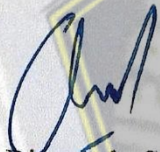
Laporan tugas akhir dengan judul “Sistem Deteksi Masker Pada Wajah Menggunakan *Convolutional Neural Network* Arsitektur *VGG16*” ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari : Senin

Tanggal : 14 Agustus 2023

TIM PENGUJI

Anggota I



Andi Rihsyah, ST, M.Kom
NIDN. 060108802

Anggota II



Ir. Sri Mulyono, M.Eng
NIDN. 0626066601



SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini :

Nama : Mohammad Ushuludin

NIM : 32601601059

Judul Tugas Akhir : Sistem Deteksi Masker Pada Wajah Menggunakan
Convolutional Neural Network Arsitektur *VGG16*

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 21 Agustus 2023

Yang Menyatakan,



SEPULUH RIBU RUPIAH
10000
TEL. 23
METERAI
TEMREL
DB4ADAKX573567314

(Mohammad Ushuludin)

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan dibawah ini :

Nama : Mohammad Ushuludin

NIM : 32601601059

Program Studi : Teknik Informatika

Fakultas : Teknologi industri

Alamat Asal : Ds. Sonorejo 05/02 Kec. Padangan Kab. Bojonegoro

Dengan ini menyatakan Karya Ilmiah berupa Tugas akhir dengan Judul : Sistem Deteksi Masker Pada Wajah Menggunakan *Convolutional Neural Network* Arsitektur *VGG16* Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan diinternet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 21 Agustus 2023

Yang menyatakan,



(Mohammad Ushuludin)

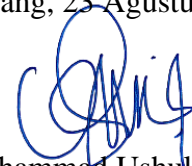
KATA PENGANTAR

Dengan mengucapkan rasa syukur Alhamdulillah atas kehadiran Allah SWT yang telah memberikan rahmat dan karunianya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Sistem Deteksi Masker Pada Wajah Menggunakan *Convolutional Neural Network* Arsitektur *VGG16*” ini untuk memenuhi salah satu syarat menyelesaikan studi serta dalam rangka memperoleh gelar sarjana (S-1) pada Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang. Dengan selesainya penyusunan Laporan Tugas Akhir ini penulis menyampaikan terimakasih kepada :

1. Bapak Prof. Dr. H. Gunarto, SH., M.Hum selaku Rektor Universitas Islam Sultan Agung.
2. Ibu Dr. Novi Marlyana, ST, MT. selaku Dekan Fakultas Teknologi Industri Universitas Islam Sultan Agung.
3. Bapak Ir. Sri Mulyono, M.Eng selaku Ketua Program Studi Teknik Informatika Universitas Islam Sultan Agung.
4. Bapak Sam Farisa Chaerul Haviana, ST, M.Kom selaku Dosen Pembimbing I.
5. Bapak Imam Much Ibnu Subroto, ST, M.Sc, Ph.D selaku Dosen Pembimbing II.
6. Orang tua dan keluarga yang selalu memberi supportnya kepada penulis.
7. Teman – teman dan semua pihak yang tidak dapat disebutkan satu persatu.

Penulis menyadari dalam penyusunan Laporan Tugas Akhir ini masih dari kata sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran untuk pengembangan dan perbaikan tugas akhir ini di masa yang akan datang. Apabila ada uraian dan penjelasan yang kurang berkenan, penulis mengucapkan permohonan maaf yang sebesar-besarnya.

Semarang, 23 Agustus 2023



(Mohammad Ushuludin)

DAFTAR ISI

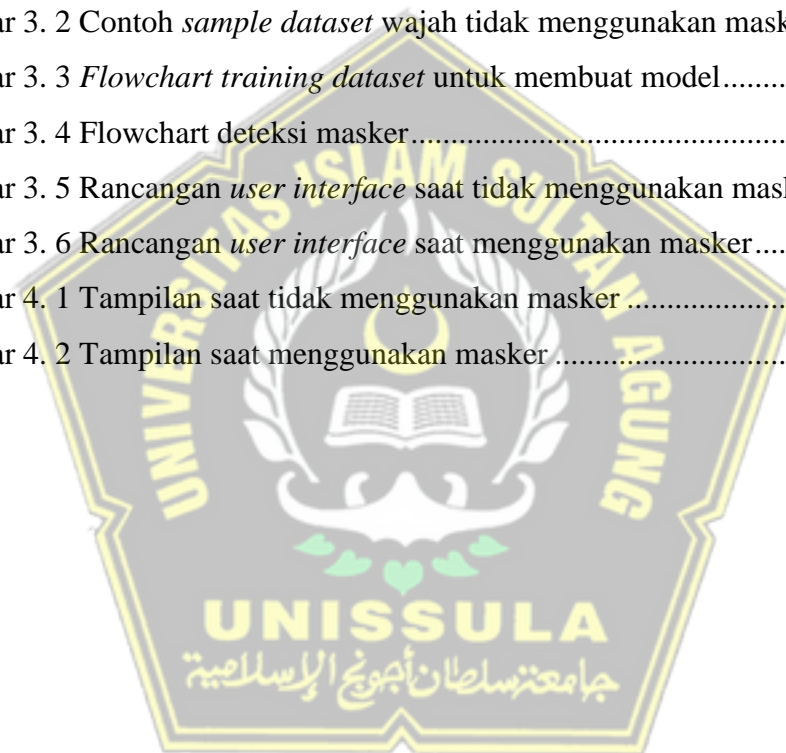
COVER.....	i
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	v
PERNYATAAN PERSETUUAN PUBLIKASI KARYA ILMIAH.....	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
ABSTRAK.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan	4
1.5 Manfaat	5
1.6 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	7
2.1 Tinjauan Pustaka.....	7
2.2 Dasar Teori.....	9
2.2.1 Citra Digital	9
2.2.2 Konvolusi Citra.....	10
2.2.3 <i>Convolutional Neural Network</i>	11
2.2.3.1 <i>Convolution Layer</i>	13
2.2.3.2 <i>Pooling Layer</i>	13
2.2.3.3 <i>Fully Connected Layer</i>	14
2.2.4 VGG16.....	15
2.2.5 <i>Optimizer</i>	16
2.2.6 Metriks Evaluasi	17
BAB III METODE PENELITIAN DAN PERANCANGAN SISTEM	18

3.1	Studi dan Pengumpulan Data.....	18
3.2	Pengembangan Model Sistem Deteksi Masker.....	20
BAB IV HASIL DAN EVALUASI.....		26
4.1	Hasil	26
4.2	Evaluasi	28
BAB V KESIMPULAN DAN SARAN		41
5.1	Kesimpulan	41
5.2	Saran.....	41
DAFTAR PUSTAKA		
LAMPIRAN		



DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi operasi konvolusi.....	11
Gambar 2. 2 Contoh arsitektur <i>convolutional neural network</i>	12
Gambar 2. 3 Proses <i>convolution layer</i>	13
Gambar 2. 4 Proses <i>pooling layer</i> metode <i>max polling</i>	14
Gambar 2. 5 Arsitektur VGG16.....	15
Gambar 3. 1 Contoh <i>sample dataset</i> wajah menggunakan masker.....	19
Gambar 3. 2 Contoh <i>sample dataset</i> wajah tidak menggunakan masker.....	19
Gambar 3. 3 <i>Flowchart training dataset</i> untuk membuat model.....	20
Gambar 3. 4 <i>Flowchart deteksi masker</i>	22
Gambar 3. 5 Rancangan <i>user interface</i> saat tidak menggunakan masker.....	24
Gambar 3. 6 Rancangan <i>user interface</i> saat menggunakan masker.....	25
Gambar 4. 1 Tampilan saat tidak menggunakan masker	26
Gambar 4. 2 Tampilan saat menggunakan masker	27



DAFTAR TABEL

Tabel 2. 1 Contoh <i>confusion matrix</i>	17
Tabel 4. 1 Hasil uji coba sistem	29
Tabel 4. 2 Hasil evaluasi	40



ABSTRAK

Pandemi COVID-19 telah mengubah paradigma global terhadap kesehatan dan kebersihan lingkungan. Penyakit yang disebabkan oleh SARS CoV-2 telah menimbulkan dampak serius di seluruh dunia. Di Indonesia, penanganan pandemi ini melibatkan berbagai peraturan dan upaya pencegahan, termasuk penggunaan masker sebagai langkah penting. Selain karena pandemi tersebut penggunaan masker juga sangat penting dilakukan di area tertentu misalnya rumah sakit. Penggunaan masker oleh dokter dan perawat saat merawat pasien sangatlah krusial untuk melindungi mereka dan mencegah penyebaran droplet penyakit. Meskipun penting, tingkat kepatuhan terhadap penggunaan masker di rumah sakit masih bervariasi. Penelitian ini mengembangkan sistem deteksi masker pada wajah menggunakan metode *Convolutional Neural Network (CNN)* arsitektur *VGG16*. Sistem deteksi masker ini dapat memberikan alternatif teknologi bagi petugas dalam mengawasi dan memantau penggunaan masker pada individu di lingkungan rumah sakit atau tempat-tempat lain yang memerlukan penggunaan masker sehingga membantu efisiensi pengecekan serta pengurangan beban kerja. Melalui pengujian dan evaluasi, ditemukan bahwa penggunaan threshold rendah sebesar 0.5 pada sistem deteksi masker memberikan hasil prediksi yang lebih akurat dengan tingkat akurasi mencapai 90% dan *f1-score* 0,909. Pengaturan threshold ini memungkinkan sistem menjadi lebih sensitif dalam mengenali apakah seseorang menggunakan masker atau tidak.

Kata Kunci: COVID-19, Masker, *Convolutional Neural Network*, *VGG16*

ABSTRACT

The COVID-19 pandemic has changed the global paradigm towards health and environmental hygiene. The disease caused by SARS-CoV-2 has had serious impacts worldwide. In Indonesia, managing this pandemic involves various regulations and preventive efforts, including the use of masks as a crucial measure. Apart from the pandemic, mask usage is also highly important in specific areas, such as hospitals. The use of masks by doctors and nurses when treating patients is crucial to protect them and prevent the spread of disease droplets. Despite its importance, compliance with mask usage in hospitals varies. This study develops a mask detection system on faces using the Convolutional Neural Network (CNN) method with the VGG16 architecture. This mask detection system can provide a technological alternative for personnel to monitor and supervise mask usage in individuals within hospital environments or other places that require mask usage, thus aiding in checking efficiency and reducing workload. Through testing and evaluation, it was found that using a low threshold of 0.5 in the mask detection system provides more accurate prediction results with an accuracy rate of 90% and an F1-score of 0.909. This threshold setting allows the system to become more sensitive in recognizing whether someone is wearing a mask or not.

Keywords: COVID-19, Masks, *Convolutional Neural Network*, *VGG16*.

BAB I PENDAHULUAN

1.1 Latar Belakang

Coronavirus Disease 2019 atau COVID-19 merupakan penyakit menular yang disebabkan oleh *Severe Acute Respiratory Syndrome Coronavirus 2 (SARS CoV-2)* serta dapat mengakibatkan infeksi saluran pernapasan, mulai dari flu biasa sampai dengan penyakit yang serius seperti *MERS (Middle East Respiratory Syndrome)* serta *SARS (Severe Acute Respiratory Syndrome)* (Sutari dkk., 2022). Berdasarkan Data dari Infeksi Emerging Kementerian Kesehatan Republik Indonesia hingga bulan Juli tahun 2023, angka total kasus yang telah dikonfirmasi terkait COVID-19 di seluruh dunia mencapai 767 juta kasus, dengan jumlah kematian sebanyak 6,9 juta orang di 234 negara yang terdampak. Di sisi lain, pemerintah Republik Indonesia telah mencatat bahwa terdapat 6,8 juta individu yang telah terkonfirmasi positif terinfeksi COVID-19, serta tercatat 161 ribu kematian akibat COVID-19 dan sebanyak 6,6 juta pasien telah dinyatakan sembuh dari penyakit tersebut.

Saat terjadi pandemi COVID-19 membuat seluruh penduduk di dunia berada pada keadaan yang mengkhawatirkan. Dampak pandemi tersebut mengakibatkan kematian dan menurunnya sistem perekonomian, serta paling mengkhawatirkan adalah penularannya yang sangat cepat. Melihat dari penyebarannya yang tinggi, WHO yang merupakan Badan Kesehatan Dunia menilai bahwa resiko dari virus ini masuk dalam kategori yang tinggi di tingkat global sehingga ditetapkan status *Public Health Emergency of International Concern (PHEIC)*. Pemerintah Republik Indonesia kemudian menerbitkan Peraturan Pemerintah (PP) yang membahas tentang Pembatasan Sosial Berskala Besar (PSBB) dalam upaya percepatan penanganan COVID-19. Kementerian Pendidikan dan Kebudayaan kemudian juga mengeluarkan surat edaran untuk langkah-langkah pencegahan virus ini, termasuk edaran mengenai upaya pencegahan COVID-19 di lingkungan Kemendikbud dan juga panduan tentang penggunaan masker sebagai salah satu langkah pencegahan penyebaran virus. Penggunaan masker menjadi sebuah bagian yang penting dalam

rangkaian pencegahan dan pengendalian penyebaran penyakit virus pada saluran pernapasan, terutama penyakit seperti COVID-19. Masker dapat digunakan bukan hanya pelindung agar melindungi individu itu sendiri, tetapi juga digunakan untuk mengendalikan potensi penularan dari individu yang telah terinfeksi virus kepada orang lain. (Simanjuntak dkk., 2022).

Dalam lingkungan rumah sakit kewajiban penggunaan masker dianggap sebuah hal yang penting saat akan memeriksa pasien, perawat dan dokter memiliki peran krusial dalam merawat pasien dan menjaga keamanan serta kebersihan lingkungan perawatan. Penggunaan masker saat akan memeriksa pasien membantu melindungi diri mereka sendiri dari paparan droplet atau partikel yang dapat mengandung patogen penyakit yang dihasilkan oleh pasien saat batuk, bersin, atau mengeluarkan cairan tubuh. Beberapa penyakitnya yakni seperti flu, tuberkulosis, atau infeksi saluran pernapasan, sehingga masker merupakan salah satu langkah penting dalam rangka menjaga kebersihan dan keamanan lingkungan rumah sakit serta melindungi kesehatan perawat, dokter, dan pasien.

Berdasarkan hasil penelitian yang dilakukan (Kustriyani dkk., 2018) kepada 153 responden di suatu rumah sakit di Kudus Jawa Tengah, didapatkan bahwa tingkat ketaatan perawat pada penggunaan alat pelindung diri seperti handscoon dan masker paada saat melaksanakan tindakan keperawatan diperoleh sebanyak 57 orang (37,3%) menunjukkan tingkat ketaatan yang tidak penuh, sementara 96 orang (62,7%) menunjukkan tingkat ketaatan penuh. Kemudian penelitian lain yang dilakukan oleh (Maulana, 2020) juga disebutkan bahwa menggunakan masker adalah salah satu upaya untuk mencegah penularan tuberculosi. Penggunaan masker bisa mencegah sebaran droplet dari udara, tetapi masih banyak penderita tuberculosi yang tidak tahu pentingnya mengenakan masker. Dari 28 orang yang diuji, sebanyak 16 orang (57,1%) tidak menggunakan masker, 10 orang (35,7%) menggunakan masker namun kurang benar, dan hanya 2 orang (7,1%) menggunakan masker dengan benar. Hal-hal tersebut menunjukkan bahwa masih ada perawat maupun pasien di lingkungan rumah sakit yang tidak patuh sepenuhnya dengan aturan masker yang ada.

Dalam rumah sakit juga, area terbatas atau kawasan khusus seperti tempat operasi/bedah dalam rumah sakit diwajibkan untuk menggunakan masker serta masuk zona steril yang tinggi. Studi kesehatan menunjukkan bahwa dengan menggunakan masker, penyebaran virus dapat disaring hingga batas tertentu ketika diuji menggunakan partikel kurang dari 0,072 mm, serta hasil masker bedah akan lebih baik untuk digunakan (Taufiq dkk., 2022). Ruang *Intensive Care Unit* (ICU) dan ruang operasi termasuk dalam kategori ruangan yang memerlukan perhatian khusus dalam menjaga kebersihannya, agar ruang tersebut tidak jadi sarana penularan terhadap pasien melalui mikroba. Sebagai lingkungan di mana pasien kondisi lemah serta kritis dirawat, ruang tersebut harus dijaga agar tidak terkontaminasi terhadap bakteri patogen. Mikroorganisme berpotensi ada di ruang-ruang tersebut, karena dapat ditularkan melalui tangan petugas medis yang terkontaminasi atau melalui partikel udara seperti aerosol, bahkan dapat berasal dari pasien sendiri. Mikroba tersebut bertahan di udara, dan ada di permukaan benda pada ruang operasi maupun ICU (Tuntun, 2022). Area khusus dan terbatas tersebut yang mewajibkan syarat pengunjung menggunakan masker agar dapat masuk ruang. Pengawasan terhadap syarat ini sering dilakukan oleh petugas keamanan, yang terkadang dapat merasa kelelahan, lupa, atau jenuh untuk memeriksa setiap orang. Situasi semacam ini dapat menyebabkan kejenuhan, terutama ketika harus melakukan pemeriksaan orang satu persatu.

Saat ini, kehidupan manusia sangat erat kaitannya dengan teknologi di era digital. Manusia terus berupaya mengembangkan teknologi guna mempermudah berbagai tugasnya. Salah satu bidang yang tengah mengalami perkembangan yakni *artificial intelligence* (AI). Penerapan AI, terutama melalui pendekatan *deep learning* yang merupakan bagian *machine learning*, telah menghasilkan kemajuan signifikan mengenai deteksi citra serta telah sering digunakan dalam mengatasi berbagai masalah mengenai *object detection* salah satunya adalah pendeteksian masker ini.

Dari permasalahan tersebut penelitian ini memberikan suatu solusi untuk memecahkan masalah yang ada dengan pembuatan sistem deteksi masker wajah otomatis menggunakan pemanfaatan *artificial intelligence* menggunakan metode

convolutional neural network (CNN) arsitektur *VGG16*. Metode tersebut juga telah terbukti efektif dalam tugas-tugas pengenalan objek. Teknologi deteksi masker pada wajah dapat menjadi solusi untuk memantau dan mengawasi kepatuhan individu terhadap penggunaan masker di situasi, kondisi, dan tempat tertentu saat ini misalnya dalam lingkungan rumah sakit yang masih ditemukan beberapa orang yang harusnya menggunakan masker namun tidak menggunakannya.

1.2 Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah bagaimana memberikan alternatif teknologi yang dapat mendeteksi masker menggunakan algoritma *Convolutional Neural Network* (CNN) arsitektur *VGG16* untuk mempermudah pendeteksian masker.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Penelitian ini berfokus dan terbatas pada pembuatan sistem untuk mendeteksi seseorang memakai masker atau tidak
2. *Dataset* yang digunakan adalah gambar wajah tanpa menggunakan masker dan wajah yang menggunakan masker
3. Penelitian ini akan memfokuskan pada penerapan algoritma *Convolutional Neural Network* arsitektur *VGG16* sebagai metode dalam pendeteksian penggunaan masker.

1.4 Tujuan

Tujuan dari penelitian ini adalah membangun suatu sistem agar dapat mendeteksi seseorang memakai masker atau tidak dengan implementasi metode *Convolutional Neural Network* arsitektur *VGG16*.

1.5 Manfaat

Manfaat yang diharapkan dari pembuatan tugas akhir ini adalah dapat menghasilkan sistem untuk mendeteksi masker secara otomatis untuk digunakan di tempat tertentu seperti contoh di lingkungan rumah sakit.

1.6 Sistematika Penulisan

Adapun sistematika penulisan tugas akhir ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini akan dijelaskan latar belakang dari permasalahan yang diteliti yaitu tentang penggunaan masker, rumusan dan batasan masalah, tujuan dari penelitian, manfaat, dan yang terakhir sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bab ini menjelaskan mengenai beberapa penelitian yang terdahulu dan relevan dengan topik penelitian saat ini. Penelitian-penelitian tersebut meliputi hasil temuan, metode penelitian yang digunakan, serta kesimpulan yang dihasilkan dari tinjauan dari berbagai penelitian yang ada sebelumnya. Kemudian dasar teori, akan membahas teori yang digunakan untuk mendukung penelitian ini. Teori-teori tersebut akan memberikan pemahaman yang mendalam pada konsep kunci yang dipakai pada penelitian. Beberapa teori yang penting untuk penelitian akan diuraikan secara rinci.

BAB III : METODE PENELITIAN DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan mengenai metode penelitian yang digunakan pada penelitian ini, termasuk studi dan pengumpulan data serta pengembangan model sistem deteksi masker. Dalam bab ini akan dijelaskan secara rinci langkah yang dipakai pada penelitian agar dapat membuat sistem yang dibutuhkan.

BAB IV : HASIL DAN EVALUASI

Pada bab ini menjelaskan hasil penelitian dan pengembangan sistem deteksi masker yang telah dilakukan sebelumnya. Pada bagian ini, akan dijelaskan temuan yang didapatkan saat pengumpulan data dan implementasi model sistem deteksi masker. Selain itu, hasil evaluasi kinerja sistem deteksi masker juga akan dijelaskan secara rinci.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini merupakan bagian penutup pada laporan. Pada bagian ini, akan dijelaskan secara singkat kesimpulan pada hasil dan evaluasi sistem deteksi masker yang dilakukan sebelumnya. Selain itu, di bagian akhir bab ini, akan diberikan saran yang dapat menjadi acuan untuk pengembangan lebih lanjut atau implementasi sistem deteksi masker ini.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada penelitian sebelumnya (Doga, 2019) merancang sebuah sistem untuk mengidentifikasi nominal uang logam menggunakan *tensorflow* dan *convolutional neural network* berbasis RaspberryPi. Penelitian ini terdiri menjadi dua fase utama, yakni proses *training* model dan proses *testing* model. *Dataset* yang dipakai untuk penelitian adalah berupa 511 gambar, kemudian di bagi 80% (511 gambar) untuk data *training*, sedangkan 20% (97 gambar) untuk data *testing*. Tahapan awal melibatkan *preprocessing*, dimana objek pada setiap gambar diberikan label untuk mengklasifikasikannya ke dalam salah satu dari empat kelas: Rp.100, Rp.200, Rp.500, dan Rp.1000. Selanjutnya, tahap pelatihan model dilakukan dengan menggunakan *dataset testing*, dan model yang dihasilkan dari pelatihan akan di uji di perangkat RaspberryPi yang diintegrasikan dengan webcam. Dilakukan uji dengan mengvariasikan jarak kamera, banyaknya objek, kondisi pencahayaan, dan kemiringan kamera. Setiap rangkaian pengujian dilakukan 50 kali untuk memastikan hasil akurat dan konsisten. Hasil dari pengujian memperlihatkan bahwa tingkat keberhasilan dalam mengidentifikasi 1-3 objek berhasil mencapai 100%. Namun, tingkat keberhasilan ini menunjukkan penurunan yang berbanding lurus ketika jumlah objek yang harus diidentifikasi bertambah. Selanjutnya, dalam pengujian terkait kemiringan kamera dan jarak, ditemukan bahwa jarak optimal antara objek dan kamera berada dalam rentang 12 hingga 16 cm, sementara kemiringan dari kamera efektif antara 0 hingga 20 derajat. Dalam kondisi ini, rata-rata 87% dari objek dapat diidentifikasi dengan benar.

Di penelitian lain yang dilakukan oleh (Wulan, 2020) yaitu pembuatan sistem untuk deteksi wajah yang berhijab menggunakan algoritma *convolutional neural network* (CNN) dengan *tensorflow*. Dalam penelitian digunakan input berupa gambar dengan ukuran 80x80 piksel, serta filter berukuran 3x3. Data pelatihan terdiri dari 250 gambar, sedangkan data pengujian 50 gambar. Sebanyak 300 gambar wajah dengan penggunaan hijab digunakan, dengan variasi ekspresi

wajah senyum dan ekspresi datar. Dari penelitian tersebut kemudian menghasilkan nilai akurasi. Dalam pemodelan *convolutional neural network* digunakan data *training* 92% serta data *testing* 87%. Akurasi tertinggi tercapai pada nilai epoch 100, mencapai 90%. Kenaikan nilai epoch di atas 100 tidak berdampak positif pada peningkatan akurasi, bahkan dapat menyebabkan penurunan akurasi. Hal ini disebabkan oleh ketidakseimbangan antara jumlah epoch yang lebih tinggi dengan jumlah dataset yang terbatas. Dengan hasil presentase yang diperoleh, dapat disimpulkan bahwa model yang sudah dikembangkan dapat secara efektif dalam deteksi wajah berhijab. Karena itu, semakin banyak jumlah data pelatihan, potensi peningkatan akurasi semakin besar, karena komputer dapat memahami pola lebih lanjut dari gambar yang dimasukkan.

Kemudian penelitian (Nurona dkk., 2021) tentang klasifikasi penyakit mata menggunakan *convolutional neural network* (CNN). Penelitian ini menjelaskan tentang klasifikasi antara mata yang normal, mata dengan katarak, glaukoma, dan penyakit retina. Proses klasifikasi ini dapat memberikan manfaat dan bantuan yang signifikan dalam mendeteksi penyakit mata dengan tepat dan akurat. Pada penelitian tersebut menggunakan dataset berjumlah 610 yang terbagi dari 4 kelas yakni normal, katarak, glaukoma, dan penyakit retina. Setelahnya, proses dilanjutkan pada tahap *pre-processing* di mana citra-citra tersebut diubah ukurannya menjadi 224x224 piksel. Pelatihan dilakukan selama 150 *epoch* pada tahap implementasi CNN. Hasil penelitian dapat mencapai akurasi sebesar 98.37%.

Selanjutnya penelitian yang lain oleh (Maharil, 2022) yakni perbandingan arsitektur VGG16 dan ResNet50 untuk rekognisi tulisan tangan aksara Lampung. Data yang dipakai yakni tulisan tangan aksara Lampung sebanyak 20 gambar. Hasilnya menunjukkan model arsitektur VGG16 menghasilkan tingkat akurasi sebesar 91% dan waktu pelatihan yang lebih efisien. Di sisi lain, model ResNet50 menghasilkan tingkat akurasi sebesar 65%. Kemudian dapat disimpulkan dalam pengenalan aksara Lampung, VGG16 terbukti menjadi pilihan baik berdasarkan nilai akurasi yang dihasilkan dan jumlah parameter yang digunakan. Sementara untuk ResNet50, meskipun memberikan hasil yang kurang memuaskan dalam

proses pelatihan, namun arsitektur ini memiliki keunggulan dalam hal waktu komputasi yang lebih efisien.

Dan yang terakhir penelitian (Setiawan dkk., 2022) tentang perancangan deteksi emosi manusia berdasarkan ekspresi wajah menggunakan algoritma VGG16. Pada penelitian ini menggunakan data dari kaggle yaitu FER2013. Dataset FER2013 berbentuk gambar. Pada gambar yang telah ada ini dilakukan *Preprocessing* yang artinya dilakukan eksplorasi data dengan memperhatikan data yang sudah ada dilakukan pengecekan. Pada data FER2013 ini berisi data mulai dari ekspresi wajah yang marah, jijik, takut, bahagia, netral, sedih, dan terkejut yang berarti 7 jenis ekspresi wajah. *Dataset* yang digunakan untuk *data training* sebesar 80% dan sebanyak 20% digunakan untuk *data testing*. Model CNN dibuat dengan arsitektur VGG16 dibuat dengan menggunakan *library keras* dan *tensorflow*. Pada Arsitektur VGG16 ini dipadukan dengan pengenalan wajah menggunakan MTCNN ini akan memberikan hasil yang maksimal, kemudian ketika dilakukan training data arsitektur VGG16 ini merespon dengan baik dan dapat menghasilkan arsitektur yang baik pula.

2.2 Dasar Teori

2.2.1 Citra Digital

Citra adalah representasi, imitasi atau kemiripan dari suatu objek atau benda. Cahaya yang direfleksikan sebuah benda dikenal sebagai citra yang terlihat, secara matematis citra dapat digambarkan sebagai fungsi kontinu yang menggambarkan distribusi intensitas cahaya pada bidang 2D. Gambar kontinu berasal dari sistem optik yang menerima sinyal analog, contohnya kamera analog dan mata, kemudian gambar diskrit berasal dari digitalisasi gambar kontinu. Proses digitalisasi gambar diskrit terdiri dari dua tahap utama yakni sampling dan kuantisasi. Tahap sampling mengumpulkan nilai diskrit secara berkala dari koordinat ruang (x, y) dengan interval sampling T . Proses kuantisasi merupakan langkah dimana nilai tingkat keabuan dari citra kontinu dikelompokkan menjadi beberapa tingkat level. Hal ini mewakili pemisahan skala keabuan $(0, L)$ jadi G level, yang diwakili oleh bilangan bulat positif m dalam notasi $G = 2^m$, di mana G mewakili tingkat keabuan dan m

adalah bilangan bulat positif. Oleh karenanya, citra digital bisa digambarkan sebagai matriks A berukuran M (jumlah baris) \times N (jumlah kolom). Di dalam matriks ini, baris dan kolom menunjukkan titik yang ada di gambar, sedangkan elemen matriks menunjukkan tingkat keabuan pada titik-titik tersebut (Zufar and Setiyono, 2016).

2.2.2 Konvolusi Citra

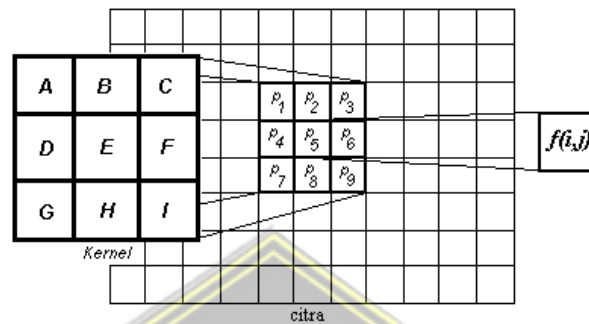
Pada kasus *machine learning* maka *processing data* citra dilakukan konvolusi citra, konvolusi adalah operator matematika yang penting dalam berbagai operasi pada pemrosesan citra. Operator ini memungkinkan penggabungan dua array, yang sering kali pada ukuran yang beda, agar dapat menghasilkan array ketiga dengan dimensinya yang sama. Dalam dimensi lain yang serupa, konvolusi memungkinkan penerapan operator yang menghasilkan nilai piksel keluaran dari kombinasi linear nilai piksel input tertentu. Dalam konteks pemrosesan citra, konvolusi digunakan sebagai operasi menghasilkan nilai piksel *output* berdasarkan kombinasi linier nilai piksel *input*. Teknik konvolusi digunakan sebagai tujuan penghalusan atau penguatan citra dengan cara mengganti nilai piksel asli menggunakan nilai piksel yang berdekatan ataupun sesuai. Namun, melalui proses konvolusi, ukuran citra tetap dipertahankan dan tidak mengalami perubahan (Rambe, 2021).

Pada umumnya, analisis citra dalam domain frekuensi sering menggunakan konvolusi. Keluaran pada suatu sistem linear bisa didapatkan melalui konvolusi antara respons impuls sistem dengan sinyal input. Proses konvolusi dijalankan dengan cara menggeser kernel konvolusi, kernel konvolusi digeser piksel per piksel pada citra, di mana piksel keluaran $f(i,j)$ dihitung dan disimpan menjadi matriks baru. Teknik konvolusi memiliki manfaat besar dalam menjalankan operasi penapisan citra. Dalam pengolahan citra digital, dilakukan konvolusi dalam dua dimensi pada citra, seperti yang diilustrasikan dalam persamaan :

$$h(x, y) = f(x, y) * g(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} f(a, b) \cdot g(x - a, y - b) \quad (1)$$

dimana $h(x,y)$ merupakan citra hasil konvolusi, $f(x,y)$ merupakan citra asal, $g(x,y)$ merupakan matriks konvolusi.

Ilustrasi konvolusi ditunjukkan dalam gambar 2.1



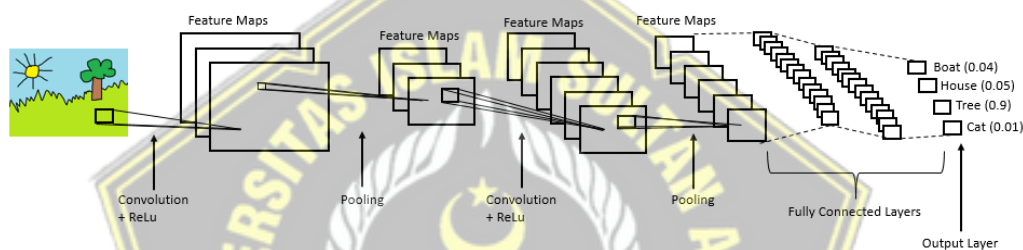
Gambar 2. 1 Ilustrasi operasi konvolusi

Hasil dari nilai $f(i,j)$ merupakan perhitungan dari $f(i,j) = A.P1 + B.P2 + C.P3 + D.P4 + E.P5 + F.P6 + G.P7 + H.P8 + I.P9$ (Wardhani and Delimayanti, 2011).

2.2.3 Convolutional Neural Network

Metode *convolutional neural network* (CNN) yang dipakai pada penelitian kali ini juga mengaplikasikan proses konvolusi citra. *Convolutional neural network* (CNN) merupakan sebuah algoritma dalam jaringan saraf yang paling sering dipakai sebagai analisis citra visual. CNN merupakan *multilayer perceptron* di mana neuronya terhubung dengan semua neuron dalam lapisan berikutnya. Keunggulannya terletak pada kemampuannya untuk menemukan hierarki pola pada data serta merangkai informasi yang kompleks pada informasi piksel yang sederhana dan lebih kecil. CNN memiliki kinerja yang sangat baik dalam mengatasi kompleksitas dan hubungan piksel dalam gambar. Keunggulan ini terlihat dalam kemampuan CNN untuk meminimalkan prapemrosesan yang diperlukan jika dibanding dengan algoritma klasifikasi gambar lainnya. Penggunaan CNN telah luas di berbagai bidang aplikasi, termasuk identifikasi gambar, pengenalan video, klasifikasi gambar, sistem untuk rekomendasi, dan pengolahan *natural language*. Metode ini sendiri pertama diperkenalkan oleh Yann LeCun saat tahun 1988. Keberhasilan dan penerapan luas CNN telah menjadi awal dalam perkembangan dan kesuksesan pada pengembangan *deep learning* (Rasywir dkk., 2020).

Convolutional neural network (CNN) merupakan suatu metode yang dirancang khusus sebagai pemrosesan data yang memiliki struktur grid contohnya adalah gambar. *Convolutional neural network* (CNN) digunakan pada klasifikasi data terlabel melalui pendekatan *supervised learning*, dimana metode ini melibatkan data latih yang memiliki variabel yang dituju. Dengan demikian, tujuan pada metode adalah mengklasifikasikan data ke dalam kelompok yang ada berdasarkan pola yang dikenali dari data pelatihan. CNN umumnya digunakan untuk melakukan pengenalan benda dalam gambar, serta untuk melakukan deteksi objek. Contoh arsitektur *convolutional neural network* ditunjukkan pada gambar 2.2



Gambar 2. 2 Contoh arsitektur *convolutional neural network* (Gunawan, Irawan and Setianingsih, 2021)

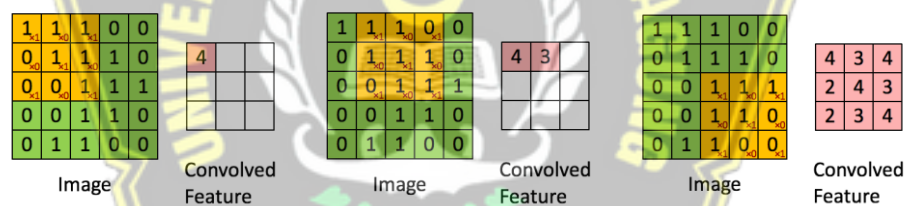
Convolutional neural network mengintegrasikan tiga arsitektur, yaitu *local receptive fields*, *shared weights* yang direpresentasikan sebagai *filter*, dan *spatial subsampling* dalam bentuk *pooling*. Konvolusi mewakili suatu matriks yang fungsinya adalah melakukan proses *filter* (Kusumaningrum, 2018).

CNN terdiri menjadi tiga jenis layer utama, yakni *convolution layer*, *pooling layer*, dan *fully connected layer*. *Convolution layer* sendiri berfungsi untuk memproses gambar *input* serta membuat *feature map* yakni merupakan hasil keluaran sebuah *output* yang di dapat dari sebuah proses konvolusi. *Pooling layer* ditempatkan setelah *convolution layer*, memiliki tujuan agar dimensi *spasial feature map* berkurang serta mengurangi beban dari komputasi. *Max pooling* dan *Average pooling* merupakan cara pengambilan *downsampling nonlinier* yang sering di pakai. *Fully connected layer* biasanya ditempatkan bagian akhir dengan tujuan agar dapat memastikan bahwa *neuron* dalam lapisan dapat dihuungkan sepenuhnya dengan aktivasi di lapisan sebelumnya, serta berfungsi untuk mengubah peta fitur

2D menjadi peta fitur 1D, yang diperlukan untuk representasi fitur dan langkah klasifikasi yang lebih lanjut (Gunawan dkk., 2021).

2.2.3.1 Convolution Layer

Layer pertama pada *convolution neural network* adalah *convolution layer*. *Convolution layer* melakukan konvolusi pada output di lapisan sebelumnya. *Convolution layer* merupakan inti dari proses dalam *Convolutional neural network*. *convolution layer* adalah lapisan yang utama dan paling penting. Konvolusi adalah istilah pada pengolahan citra merujuk pada penerapan kernel (matriks berbentuk kotak warna kuning) ke seluruh area citra di semua offset, seperti terlihat dalam Gambar 2.3. Area yang lebih besar dengan warna hijau merupakan citra yang nantinya dilakukan konvolusi. Kemudian kernel (kotak kuning) bergerak mulai dari bagian sudut kiri atas berlanjut sampai dengan sudut kanan bawah. Hasilnya operasi ini dapat di lihat dalam gambar di sebelah kanannya. Contoh proses pada *convolution layer* ditunjukkan pada gambar 2.3



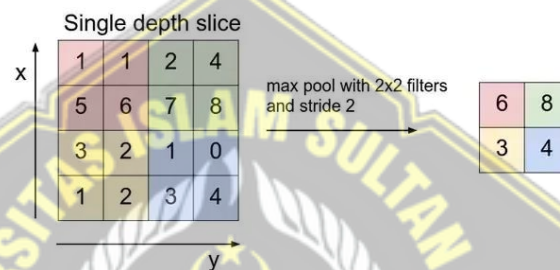
Gambar 2. 3 Proses *convolution layer*

Tujuan konvolusi pada citra adalah mengambil fitur-fitur penting yang ada pada citra masukan. Proses tersebut dapat menciptakan transformasi linear dari data masukan dengan memperhitungkan informasi spasial dalam data tersebut. Bobot dalam layer menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi bisa di latih berdasarkan input di metode CNN (Kusumaningrum, 2018).

2.2.3.2 Pooling Layer

Kemudian *pooling layer* yang merupakan lapisan dalam *Convolutional Neural Network* (CNN) yang mengambil *feature map* sebagai input dan memprosesnya menggunakan operasi statistik tertentu dari nilai pada piksel yang paling dekat. Dalam arsitektur CNN, lapisan ini umumnya ditempatkan dengan berkala setelah beberapa *convolution layer*. Dengan menyelipkan lapisan ini secara berurutan di antara *convolution layer* proses ini dengan progresif dapat mengurangi

dimensi volume hasil keluaran dari *feature map*, sehingga hal ini berdampak pada pengurangan banyaknya parameter dan perhitungan dalam jaringan, dan juga dapat membantu mengendalikan masalah *overfitting*. Metode yang digunakan pada *pooling layer* yakni nilai rata-rata (*average pooling*) ataupun nilai maksimal (*max-pooling*), namun metode pooling yang lebih sering dipakai yakni menggunakan *max-pooling*. Dalam metode *max-pooling* keluaran/output pada lapisan konvolusi dibagi untuk menghasilkan grid kecil, kemudian nilai maksimum pada grid tersebut diambil agar dapat membentuk matriks citra baru yang sudah direduksi, seperti yang diilustrasikan pada gambar 2.4



Gambar 2. 4 Proses *pooling layer* metode *max pooling*

Area kotak grid dengan warna merah, kuning, hijau, dan biru di sebelah kiri adalah kumpulan grid yang nantinya diambil nilai maksimalnya. Dengan demikian, hasilnya bisa di lihat di himpunan grid di sebelah kanan. *Pooling layer* dapat memastikan bahwa fitur di ekstraksi tetap konsisten meski objek dalam citra bergeser atau translasi. *Pooling layer* memiliki tujuan agar dapat mengurangi dimensi citra sehingga dapat di ganti menggunakan sebuah *convolution layer* dengan nilai *stride* sama dengan *pooling layer* yang bersangkutan. *Stride* adalah sebuah parameter yang akan mengatur seberapa banyak *filter* bergeser. Ketika nilai stridenya satu, filter bergeser sejauh satu piksel dalam arah horizontal dan vertikal. Apabila nilai *stride* kecil, maka dampaknya adalah akan banyak detail dari informasi yang akan diambil, tetapi komputasi akan menjadi lebih berat dibandingkan dengan nilai yang *stride* lebih besar (Kusumaningrum, 2018).

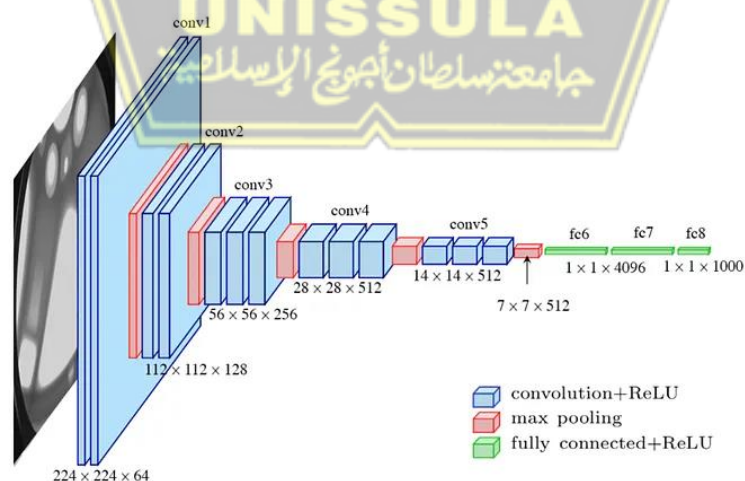
2.2.3.3 Fully Connected Layer

Dan terakhir *Fully connected layer* adalah lapisan di mana setiap neuron aktivasi dari lapisan sebelumnya dihubungkan dengan semua neuron di lapisan berikutnya. Pada penerapan *Multi Layer Perceptron (MLP)*, *fully connected layer*

digunakan untuk mentransformasikan dimensi supaya bisa terklasifikasi secara linier. *Fully connected layer* dan *convolution layer* memiliki perbedaan umum yakni bahwa neuron-neuron dalam *convolution layer* terhubung dengan wilayah tertentu saja pada input, sedangkan pada lapisan *fully connected layer* memiliki neuron terhubung menyeluruh. Meskipun demikian, keduanya masih mengoperasikan produk dot, sehingga fungsinya sendiri relatif serupa (Kusumaningrum, 2018).

2.2.4 VGG16

Salah satu arsitektur pada *convolution neural network* adalah VGG16. VGGNet merupakan hasil pengembangan oleh Tim Visual Geometry Group di Oxford University. Model ini meraih posisi sebagai 1st Runner-up dalam kontes ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) dalam kategori Klasifikasi Gambar dan pemenang Image Localization pada tahun 2014 (Pardede and Hardiansah, 2022). VGG16 adalah sebuah arsitektur *convolutional neural network* (CNN) yang dikembangkan oleh Karen Simonyan dan Andrew Zisserman. VGG16 menunjukkan bahwa *deep networks* adalah komponen yang penting untuk kinerja yang baik. Gambaran contoh arsitektur VGG16 dapat dilihat pada gambar 2.5



Gambar 2. 5 Arsitektur VGG16

Arsitektur VGG16 terdiri dari total 16 lapisan bobot (*weight layer*) yang mana terdiri dari *convolution layer* dan *fully connected layer*. Keunikan dari model

ini adalah memiliki arsitektur yang sangat seragam, dimulai dari awal hingga akhir, dengan menerapkan konvolusi 3x3 dan pooling 2x2 secara konsisten. Kelemahan dari arsitektur VGG16 ini yakni lebih berat saat evaluasi dan memakai lebih banyak memori serta parameter (Gunawan dkk., 2021).

2.2.5 Optimizer

Dalam proses *training data* digunakan sebuah *optimizer*. *Optimizer* merupakan algoritma atau metode yang digunakan sebagai penyesuaian terhadap atribut pada jaringan saraf tiruan. Contoh atribut yang dapat diubah *optimizer* meliputi *learning rate* dan nilai bobot (*weights*) agar dapat mengurangi nilai *loss* saat proses *training*. *Adam (Adaptive Moment Estimation)* merupakan sebuah algoritma optimisasi umum yang digunakan dalam pelatihan jaringan saraf tiruan (*neural networks*) dalam bidang *machine learning*. *Adam* merupakan sebuah algoritma berasal dari pengembangan algoritma *stochastic gradient descent (SGD)*, di mana dalam *Adam*, bobot pada jaringan diperbarui nilainya. *Adam* menggabungkan konsep dari algoritma optimisasi lainnya seperti *stochastic gradient descent (SGD)* dengan mekanisme adaptif untuk mengatur *learning rate* (tingkat pembelajaran) berdasarkan estimasi momen pertama dan kedua dari gradien.

Secara singkat, algoritma *Adam* memperhitungkan perubahan pada gradien dan adaptif dalam memperbarui parameter model. Algoritma ini juga membantu mengatasi beberapa masalah umum dalam optimisasi, seperti *learning rate* yang tidak stabil atau terlalu besar. Dalam penggunaan algoritma ini, langkah awal yang diperlukan adalah mengatur nilai *learning rate* sebelum memulai proses *trainig*. *Learning rate* adalah suatu parameter *training* yang berfungsi mengatur seberapa besar perubahan yang akan diterapkan pada bobot koneksi selama proses pelatihan. Besarnya *learning rate* yang dipilih akan mempengaruhi seberapa efektif dan efisien pelatihan model tersebut (Kurniawan dkk., 2023).

2.2.6 Metriks Evaluasi

Pada penelitian ini akan digunakan *accuracy*, *precision*, *recall*, dan *f1-score* untuk tahap evaluasi. Rumus untuk menghitung *accuracy*, *precision*, *recall*, dan *f1-score* sebagai berikut :

$$accuracy = \frac{TP+TN}{Total} \quad (2)$$

$$precision = \frac{TP}{TP+FP} \quad (3)$$

$$recall = \frac{TP}{TP+FN} \quad (4)$$

$$f1score = \frac{TP}{TP + \frac{1}{2}(FP+FN)} \quad (5)$$

Accuracy, *precision*, *recall*, dan *f1-score* didapatkan dari *confusion matrix*. *Confusion matrix* merupakan sebuah tabel yang menggambarkan jumlah klasifikasi data uji benar serta jumlah klasifikasi data uji salah. Sebagai contohnya, sebuah *confusion matrix* untuk klasifikasi biner dapat dilihat pada tabel 2.1

Tabel 2. 1 Contoh *confusion matrix*

		Kelas Prediksi (Predicted)	
		1	0
Kelas Sebenarnya (Actual)	1	TP	FN
	0	FP	TN

Keterangan:

TP (*True Positive*) = jumlah dari kelas 1 yang benar diklasifikasikan sebagai kelas 1

TN (*True Negative*) = jumlah dari kelas 0 yang benar diklasifikasikan sebagai kelas 0

FP (*False Positive*) = jumlah dari kelas 0 yang salah diklasifikasikan sebagai kelas 1

FN (*False Negative*) = jumlah dari kelas 1 yang salah diklasifikasikan sebagai kelas 0 (Normawati and Prayogi, 2021).

BAB III

METODE PENELITIAN DAN PERANCANGAN SISTEM

Metode penelitian yang dilakukan agar dapat membangun sistem deteksi masker pada wajah menggunakan metode *convolutional neural network* arsitektur *VGG16* ini meliputi studi dan pengumpulan data, pengembangan model deteksi masker, dan evaluasi.

3.1 Studi dan Pengumpulan Data

Studi dan pengumpulan data ini dilakukan agar dapat menganalisa dan mencari data yang tepat untuk digunakan, sehingga dapat memenuhi tujuan yang diharapkan serta dapat menyelesaikan masalah yang ada. Tahapan studi dan pengumpulan data pada penelitian ini adalah :

1. Studi Literatur

Studi literatur yang dilakukan dengan cara mempelajari berbagai sumber referensi yang ada termasuk jurnal ilmiah, buku, laporan penelitian, serta platform digital lainnya yang relevan. Studi literatur ini memiliki tujuan utama untuk mengidentifikasi serta menganalisis permasalahan yang relevan dengan deteksi masker, serta untuk membangun landasan pengetahuan yang kokoh seputar topik tersebut. Melalui pemahaman mendalam terhadap berbagai konsep, teori, dan temuan terkini dalam literatur, upaya ini diarahkan untuk menggali wawasan yang lebih mendalam tentang tantangan dan potensi solusi dalam pengembangan sistem deteksi masker yang efektif.

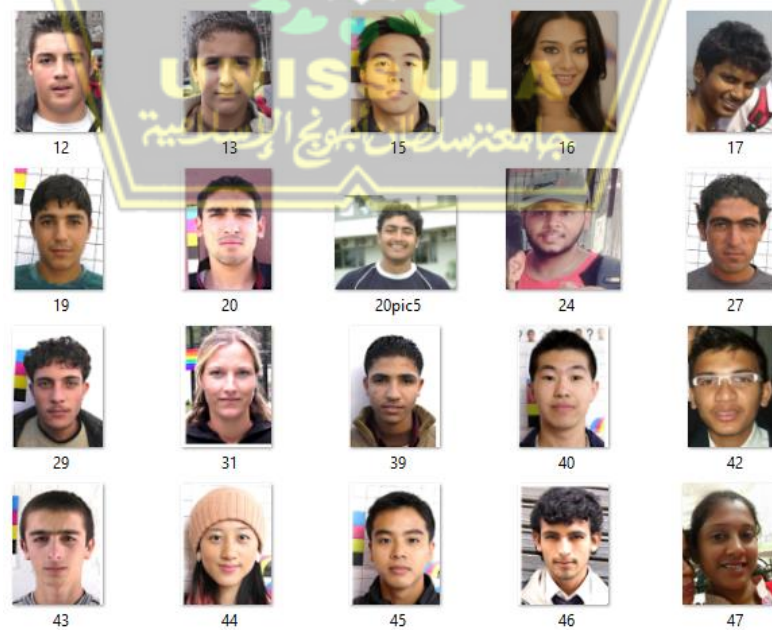
2. Pengumpulan *dataset*

Dataset yang digunakan pada sistem ini berasal dari repositori GitHub (<https://github.com/prajnasb/observations>). *Dataset* ini secara keseluruhan mencakup sebanyak 1376 sampel foto. Lebih spesifik lagi, terdapat total 690 *dataset* wajah menggunakan masker dan 686 *dataset* wajah tidak menggunakan masker. Namun untuk mengurangi beban komputasi dan meminimalkan waktu yang dibutuhkan dalam tahap pelatihan, maka dilakukan pengurangan jumlah *dataset* sehingga menjadi sebanyak 160 foto wajah yang menggunakan masker, serta 160 foto wajah tidak menggunakan

masker. *Dataset* foto tersebut juga memiliki ukuran *pixel* yang beragam. Contoh *sample dataset* yang menggunakan masker dapat dilihat pada gambar 3.1



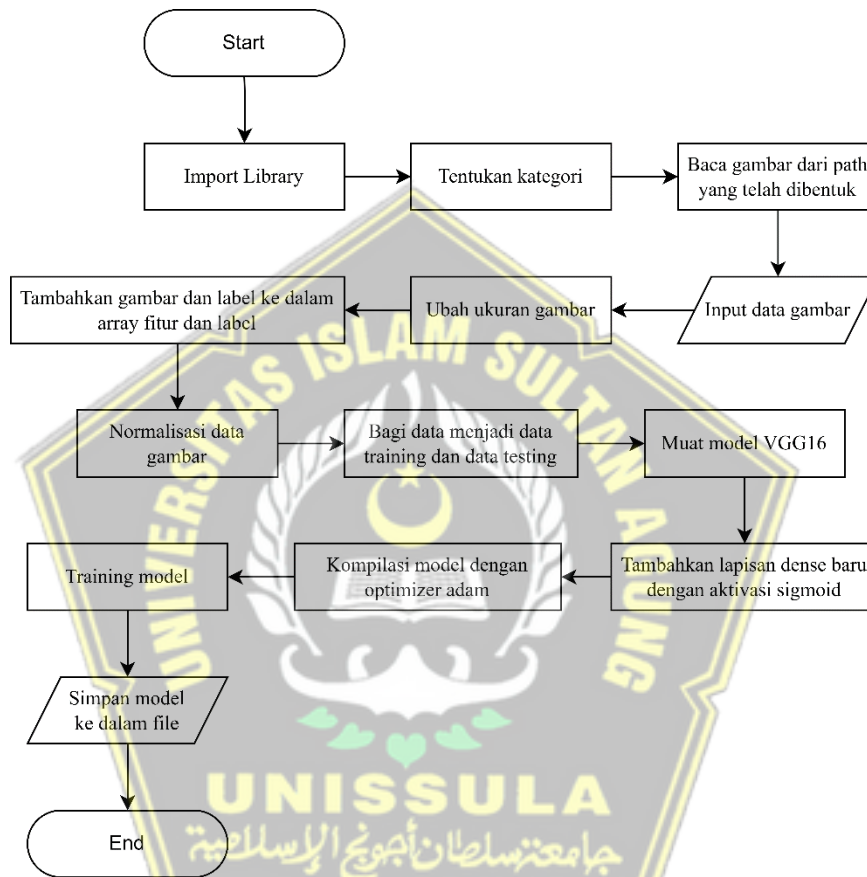
Gambar 3. 1 Contoh *sample dataset* wajah menggunakan masker
 Contoh *sample dataset* wajah tidak menggunakan masker dapat dilihat pada gambar 3.2



Gambar 3. 2 Contoh *sample dataset* wajah tidak menggunakan masker

3.2 Pengembangan Model Sistem Deteksi Masker

Pengembangan model sistem deteksi masker ini menggunakan metode *convolutional neural network* arsitektur *VGG16* dengan bahasa pemrograman *python*. *Flowchart training dataset* untuk membuat model sistem deteksi masker ini dapat dilihat pada gambar 3.3



Gambar 3.3 *Flowchart training dataset* untuk membuat model

Langkah-langkah dalam proses *training* model deteksi masker ini adalah :

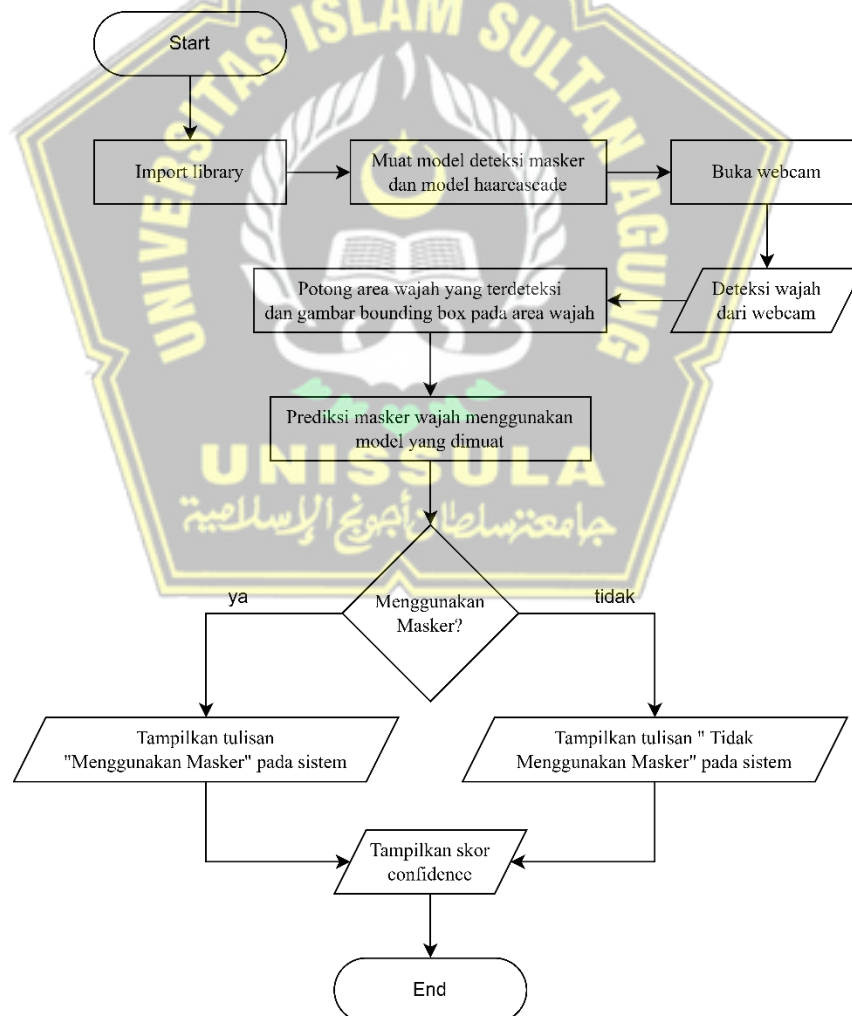
1. Import library yakni *library os, cv2(OpenCV), keras, numpy, train_test_split* dari *sklearn.model_selection, VGG16* dari *keras.applications.vgg16* . *Os* digunakan untuk berinteraksi dengan sistem operasi dan termasuk juga untuk membaca direktori folder maupun file, *cv2 (OpenCV)* yang merupakan *library* untuk memproses gambar dan video serta digunakan untuk membaca gambar dan mengubah ukuran gambar, *keras* yang merupakan *framework deep learning* yang digunakan untuk membangun dan melatih model *neural*

networks, *numpy* yang merupakan *library* numerik untuk melakukan operasi numerik pada data *array*, *train_test_split* dari *sklearn.model_selection* yang berfungsi untuk membagi *dataset* menjadi data *training* dan data *testing*, dan terakhir *library VGG16* dari *keras.applications.vgg16* yang merupakan model pre-trained *VGG16* yang telah di *training* sebelumnya yang tersedia di *keras*.

2. Mendefinisikan kategori dan pembentukan data, terdapat dua kategori, yaitu “with_mask” (dengan masker) dan “without_mask” (tanpa masker). Kemudian baca gambar dari *path* yang telah dibentuk. *Input* data gambar untuk proses selanjutnya, data gambar dari setiap kategori dimuat, namun karena ukuran *pixel* setiap gambar berbeda-beda maka gambar akan diubah ukuran menjadi 224x224 *pixel* semua dan disimpan sebagai pasangan data dan label dalam list *data*.
3. Data gambar dinormalisasi dengan membagi setiap nilai piksel dalam gambar dengan 255 untuk memastikan nilai piksel berada dalam rentang [0, 1]. Data dibagi kedalam data *training* dan data *testing* menggunakan fungsi *train_test_split* dengan jumlah *data training* sebanyak 80% serta *data testing* dengan jumlah sebanyak 20%.
4. Memuat model *VGG16*. Lapisan-lapisan *VGG16* ditambahkan ke dalam model baru (*Sequential*) kecuali lapisan *output* terakhir (pengecualian dilakukan dengan *looping* pada *vgg.layers[:-1]*). Lapisan baru dengan satu neuron dan fungsi aktivasi *sigmoid* ditambahkan di akhir model untuk output biner yang menunjukkan apakah gambar memuat masker atau tidak. Model dikompilasi dengan menggunakan *optimizer adam*, *loss function binary_crossentropy*, dan *metrik accuracy* untuk evaluasi.
5. Model difitkan menggunakan data latih dan data uji selama 15 *epochs* (putaran latihan). Akurasi dan loss dihitung selama pelatihan. Model yang telah di *training* akan menghasilkan output dan disimpan dalam file *detection_model.h5*.

Setelah melewati proses *preprocessing data* dan *training* model, langkah selanjutnya yakni memuat model yang sudah di *training* untuk deteksi masker.

Model yang telah di *training* ini telah mempelajari fitur-fitur penting dari gambar wajah dengan dan tanpa masker selama pelatihan. Setelah proses *training* selesai, model disimpan menjadi ke dalam *file detection_model.h5*. Memuat model untuk deteksi masker ini memungkinkan kita untuk *testing* model terhadap data yang baru dan belum pernah dilihat sebelumnya dan mengevaluasi kinerja model dalam mengenali apakah seseorang memakai masker atau tidak. Kemudian dengan *library cv2(OpenCV)* akan digunakan untuk mengakses kamera webcam secara real-time. Ketika program berjalan, kamera akan terhubung dan terbuka, dan model akan digunakan untuk melakukan deteksi masker pada wajah yang terdeteksi dalam bingkai kamera. Proses tersebut dapat dilihat lebih lengkapnya pada flowchart gambar 3.4



Gambar 3. 4 Flowchart deteksi masker

Gambar 3.4 merupakan flowchart implementasi deteksi masker wajah secara real-time menggunakan model yang telah dilatih sebelumnya. Langkah-langkahnya sebagai berikut :

1. Mengimport *library* yang dibutuhkan, yaitu *cv2 (OpenCV)* untuk manipulasi gambar dan video, serta *load_model* dari *keras.models* untuk memuat model yang sudah terlatih. Fungsi *load_face_mask_detection_model()* digunakan untuk memuat model deteksi masker wajah dari file yang disimpan sebelumnya dengan nama *detection_model.h5* dan menyimpannya ke dalam variabel model. Model *Haar Cascade* juga dimuat dan digunakan untuk mendeteksi wajah dalam gambar.
2. Program membuka kamera menggunakan *cv2.VideoCapture(0)*. Kemudian, dalam *loop while*, setiap bingkai gambar yang diterima dari kamera akan diolah. Apabila wajah terdeteksi maka akan menggambar *bounding box* berwarna biru pada wajah yang telah terdeteksi. Kemudian mengubah ukuran gambar menjadi *224x224 pixel*.
3. Setelah mendeteksi wajah, program akan memanggil fungsi *detect_face_mask* untuk mendeteksi apakah wajah dalam gambar memakai masker atau tidak. Hasil dari deteksi ini akan digunakan untuk menampilkan label pada gambar, menunjukkan apakah wajah menggunakan masker atau tidak. Apabila terdeteksi menggunakan masker maka akan menampilkan tulisan “Menggunakan Masker”, apabila terdeteksi tidak menggunakan masker maka akan menampilkan tulisan “Tidak Menggunakan Masker”, serta menampilkan skor dari tingkat kepercayaan dari prediksi tersebut atau *confidence score*.
4. Gambar yang telah diolah tersebut ditampilkan dalam jendela dengan judul "DETEKSI MASKER". Program akan terus melakukan deteksi dan menampilkan hasilnya sampai tombol “x” pada keyboard ditekan untuk menghentikan program. Terakhir, ketika loop berakhir, jendela yang menampilkan gambar akan ditutup dengan memanggil *cv2.destroyAllWindows()*.

Berikut ini merupakan rancangan *user interface* pada sistem deteksi masker pada wajah :



Gambar 3. 5 Rancangan *user interface* saat tidak menggunakan masker

Pada Gambar 3.5, tergambar sebuah rancangan *user interface* yang telah dirancang, menunjukkan tampilan ketika wajah tidak menggunakan masker. Dalam pengoperasian sistem deteksi ini webcam digunakan untuk melakukan pendeteksian secara real-time. Ketika wajah masuk ke dalam bingkai (frame), sistem secara otomatis akan mengidentifikasinya dan diberikan sebuah bingkai (*bounding box*) dengan warna biru yang mengelilingi area wajah yang terdeteksi dan kemudian dilakukan prediksi. Saat wajah tidak menggunakan masker maka akan diberikan tulisan “Tidak Menggunakan Masker” disertai list warna merah dibawah tulisan tersebut, serta terdapat nilai dari *confidence score* semakin mendekati nilai 1, semakin tinggi tingkat keyakinan model bahwa prediksinya benar.



Gambar 3. 6 Rancangan *user interface* saat menggunakan masker

Kemudian pada Gambar 3.6, tergambar sebuah rancangan *user interface* yang telah dirancang, menunjukkan tampilan ketika wajah menggunakan masker. Sama halnya dengan sebelumnya, dalam pengoperasian sistem deteksi ini webcam juga digunakan untuk melakukan pendeteksian secara real-time. Ketika wajah masuk ke dalam bingkai (frame), sistem secara otomatis akan mengidentifikasinya dan diberikan sebuah bingkai (*bounding box*) dengan warna biru yang mengelilingi area wajah yang terdeteksi dan kemudian dilakukan prediksi. Saat wajah menggunakan masker maka akan diberikan tulisan “Menggunakan Masker” disertai list warna hijau dibawah tulisan tersebut, serta terdapat nilai dari *confidence score* semakin mendekati nilai 1, semakin tinggi tingkat keyakinan model bahwa prediksinya benar.

BAB IV HASIL DAN EVALUASI

4.1 Hasil

Berikut merupakan hasil dari sistem yang dibuat, cara menjalankannya adalah pertama-tama pastikan telah menyiapkan model yang telah terlatih sebelumnya melalui proses *training dataset* yang telah dilakukan dan pastikan file model tersebut letaknya sama dengan file kode program untuk deteksi masker. Setelah memastikan bahwa model telah berada pada posisi yang sesuai, langkah selanjutnya adalah menjalankan program yang telah di rancang. Proses ini akan menginisiasi antarmuka pengguna yang telah disusun sebelumnya untuk tujuan deteksi masker. Melalui program ini, antarmuka pengguna akan muncul yang terhubung dengan webcam. Berikut ini merupakan hasil dari rancangan yang sebelumnya telah dibuat, yang mana terdapat dua kondisi yakni menggunakan masker dan tidak menggunakan masker :

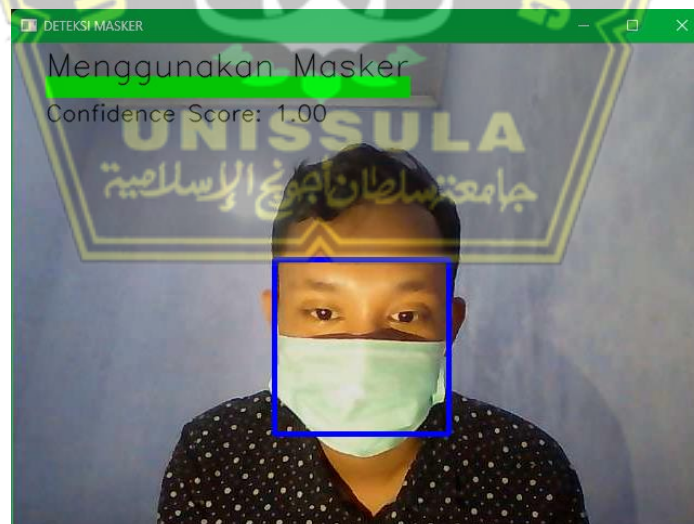


Gambar 4. 1 Tampilan saat tidak menggunakan masker

Dalam gambar 4.1 ini, kita disajikan dengan visualisasi tentang situasi di mana seseorang tidak menggunakan masker. Sebuah *dataset* telah diolah dan dimanfaatkan untuk membuat sebuah model yang memungkinkan deteksi keberadaan masker pada wajah seseorang. Model ini, setelah melewati tahap pelatihan, telah mencapai kemampuan untuk mengenali kondisi ketika seseorang

tidak memakai masker. Pada gambar 4.1 tersebut, wajah yang terdeteksi akan diberikan *bounding box* atau frame berwarna biru di sekitar area wajah yang terdeteksi, selain itu kita dapat melihat sebuah teks yang ditampilkan di bagian atas, yaitu "Tidak Menggunakan Masker". Teks ini berfungsi sebagai informasi yang membantu kita memahami isi dari gambar secara lebih jelas. Selain itu, teks tersebut diberikan garis berwarna merah dibawahnya. Selain teks dan garis berwarna merah, juga diberikan informasi mengenai nilai dari *confidence score*. *Confidence score* menggambarkan sejauh mana model yakin atau percaya terhadap prediksinya terkait tidak menggunakan masker pada gambar tersebut. Semakin mendekati nilai 1 maka semakin baik.

Secara keseluruhan, gambar 4.1 ini memberikan informasi tidak menggunakan masker pada seseorang. Model yang telah dilatih dengan dataset tersebut menunjukkan potensi untuk digunakan dalam berbagai konteks, seperti pengawasan kepatuhan penggunaan masker dalam kehidupan sehari-hari, keamanan, dan juga dalam upaya pencegahan penyebaran penyakit yang dapat mengenali penggunaan masker pada seseorang.



Gambar 4. 2 Tampilan saat menggunakan masker

Pada gambar 4.2 ini, digambarkan sebuah situasi di mana seseorang telah menggunakan masker dengan benar. Model yang telah dilatih menggunakan dataset yang telah diolah berhasil mengenali dan membedakan antara kondisi di mana seseorang memakai masker dan tidak menggunakan masker. Terlihat di gambar

bahwa seseorang telah mengenakan masker. Wajah yang terdeteksi akan diberikan *bounding box* atau frame berwarna biru di sekitar area wajah yang terdeteksi. Terdapat sebuah teks yang menunjukkan bahwa orang tersebut "Menggunakan Masker" dibagian atas frame. Selain itu, terdapat garis berwarna hijau yang dibawah teks "Menggunakan Masker", juga diberikan informasi mengenai nilai dari *confidence score*. Semakin *confidence score* mendekati nilai 1 maka menandakan semakin tinggi keyakinan model bahwa hasil prediksinya benar.



4.2 Evaluasi

Evaluasi adalah tahap dimana sistem atau alat sudah tahap uji coba dan setelah semua bekerja bagaimana semestinya. Maka, dilakukan evaluasi terhadap hasil yang diperoleh. Pada sistem ini di bangun dengan beberapa nilai parameter yang telah ditentukan sebelumnya seperti :

1. *Epochs*, jumlah iterasi yang digunakan saat melatih model yakni ditentukan untuk dilatih selama 15 epoch.
2. *Optimizer*, di sini, digunakan Adam (*Adaptive Moment Estimation*) sebagai optimizer.
3. *Learning rate*, nilai learning rate tidak secara eksplisit diberikan saat melakukan kompilasi model. Oleh karena itu, *learning rate* akan mengikuti nilai default yang digunakan oleh *optimizer* adam pada framework keras. Nilai *learning rate* default untuk *optimizer* adam pada keras adalah 0.001
4. *Batch*, ukuran *batch* tidak secara eksplisit ditentukan. Ukuran *batch* adalah jumlah sampel data yang digunakan dalam satu iterasi proses pelatihan sebelum bobot model diperbarui. Karena itu, ukuran *batch* akan mengikuti default dari *framework* keras, yaitu 32. Artinya, model akan menggunakan ukuran *batch* sebesar 32 saat melatih model secara default.

Dengan menggunakan parameter tersebut maka dihasilkan hasil uji coba model seperti yang ditunjukkan pada tabel 4.1 berikut ini :


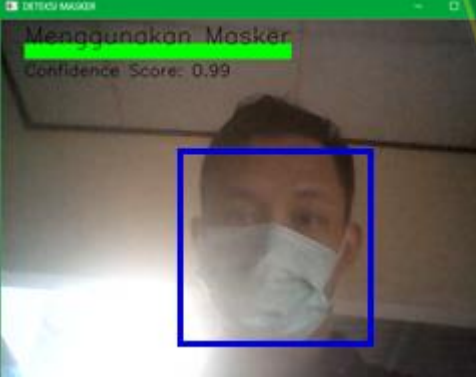
Tabel 4. 1 Hasil uji coba sistem


No	Hasil	Predicted	Actual	Confidence Score	Threshold				
					0.9	0.8	0.7	0.6	0.5
1		Menggunakan Masker	Menggunakan Masker	0.89	FP	TP	TP	TP	TP
2		Menggunakan Masker	Menggunakan Masker	0.53	FP	FP	FP	FP	TP

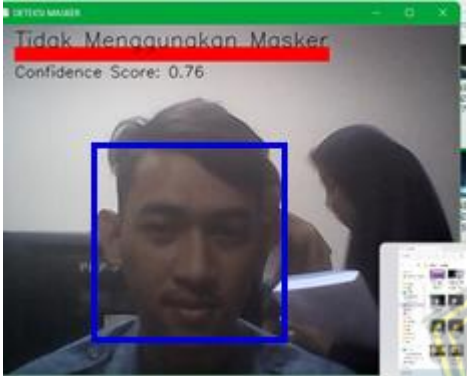

3		Menggunakan Masker	Menggunakan Masker	1.00	TP	TP	TP	TP	TP
4		Menggunakan Masker	Menggunakan Masker	0.99	TP	TP	TP	TP	TP

5		Menggunakan Masker	Menggunakan Masker	1.00	TP	TP	TP	TP	TP
6		Menggunakan Masker	Menggunakan Masker	1.00	TP	TP	TP	TP	TP

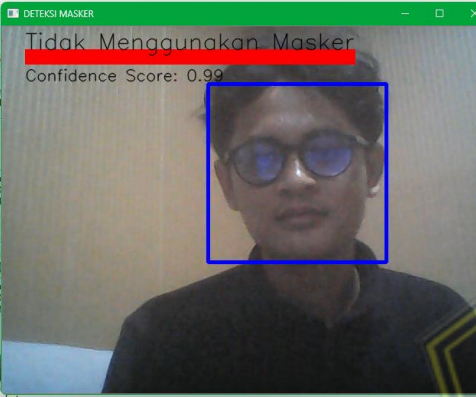
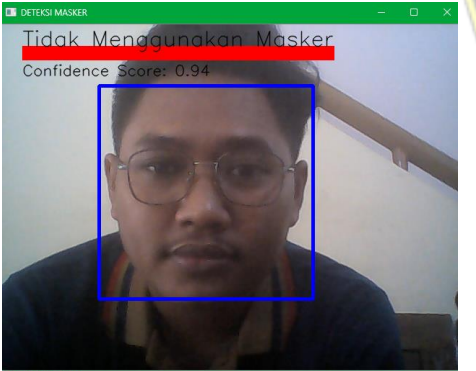
7		Menggunakan Masker	Menggunakan Masker	1.00	TP	TP	TP	TP	TP
8		Menggunakan Masker	Menggunakan Masker	0.69	FP	FP	FP	TP	TP

9		Menggunakan Masker	Menggunakan Masker	1.00	TP	TP	TP	TP	TP
10		Menggunakan Masker	Menggunakan Masker	0.99	TP	TP	TP	TP	TP

11		Tidak Menggunakan Masker	Tidak Menggunakan Masker	0.97	TN	TN	TN	TN	TN
12		Tidak Menggunakan Masker	Tidak Menggunakan Masker	0.99	TN	TN	TN	TN	TN

13		Tidak Menggunakan Masker	Tidak Menggunakan Masker	0.76	FN	FN	TN	TN	TN
14		Tidak Menggunakan Masker	Tidak Menggunakan Masker	0.55	FN	FN	FN	FN	TN

15		Tidak Menggunakan Masker	Tidak Menggunakan Masker	0.55	FN	FN	FN	FN	TN
16		Menggunakan Masker	Tidak Menggunakan Masker	0.69	FP	FP	FP	FP	FP

17		Tidak Menggunakan Masker	Tidak Menggunakan Masker	0.99	TN	TN	TN	TN	TN
18		Tidak Menggunakan Masker	Tidak Menggunakan Masker	0.94	TN	TN	TN	TN	TN

19		Tidak Menggunakan Masker	Tidak Menggunakan Masker	0.89	FN	TN	TN	TN	TN
20		Menggunakan Masker	Tidak Menggunakan Masker	0.90	FP	FP	FP	FP	FP

Evaluasi sistem ini menggunakan nilai akurasi menggunakan rumus nomor 2, presisi menggunakan rumus nomor 3, *recall* menggunakan rumus nomor 4, *f1-score* menggunakan rumus nomor 5 yang melibatkan perhitungan dan interpretasi nilai-nilai tersebut untuk mengukur kinerja sistem deteksi atau klasifikasi. Semakin tingginya nilai akurasi, maka akan semakin baik pula kinerja sistem secara keseluruhan. Semakin tingginya nilai presisi, maka semakin sedikitnya prediksi positif yang salah dilakukan oleh sistem. Semakin tingginya nilai *recall*, maka akan semakin sedikitnya data positif terlewatkan oleh sistem. *F1-score* memberikan keseimbangan antara nilai presisi dan *recall*. Semakin tingginya nilai *f1-score*, maka akan semakin baik juga kinerja sistem secara keseluruhan saat memprediksi kedua kelas dengan baik. Dari hasil uji coba tersebut maka akan dilakukan perhitungan *accuracy*, *precision*, *recall*, dan *f1-score* menggunakan rumus yang ada dan menghasilkan hasil sebagai berikut :

1. Pada nilai *threshold* 0.9 menghasilkan nilai 7 *true positive*, 4 *true negative*, 5 *false positive*, dan 4 *false negative*. Sehingga menghasilkan nilai *accuracy* sebesar 55%, *precision* sebesar 0.583, *recall* sebesar 0.636, dan *f1-score* sebesar 0.608
2. Pada nilai *threshold* 0.8 menghasilkan nilai 8 *true positive*, 5 *true negative*, 4 *false positive*, dan 3 *false negative*. Sehingga menghasilkan nilai *accuracy* sebesar 65%, *precision* sebesar 0.666, *recall* sebesar 0.727, dan *f1-score* sebesar 0.695
3. Pada nilai *threshold* 0.7 menghasilkan nilai 8 *true positive*, 6 *true negative*, 4 *false positive*, dan 2 *false negative*. Sehingga menghasilkan nilai *accuracy* sebesar 70%, *precision* sebesar 0.666, *recall* sebesar 0.8, dan *f1-score* sebesar 0.727
4. Pada nilai *threshold* 0.6 menghasilkan nilai 9 *true positive*, 6 *true negative*, 3 *false positive*, dan 2 *false negative*. Sehingga menghasilkan nilai *accuracy* sebesar 75%, *precision* sebesar 0.75, *recall* sebesar 0.818, dan *f1-score* sebesar 0.782
5. Pada nilai *threshold* 0.5 menghasilkan nilai 10 *true positive*, 8 *true negative*, 2 *false positive*, dan 0 *false negative*. Sehingga menghasilkan nilai *accuracy*

sebesar 90%, *precision* sebesar 0.833, *recall* sebesar 1, dan *f1-score* sebesar 0.9099

Hasil dari evaluasi pada uji coba dapat dilihat pada tabel 4.2

Tabel 4. 2 Hasil evaluasi

	<i>Threshold</i>				
	0.9	0.8	0.7	0.6	0.5
<i>True Positive</i>	7	8	8	9	10
<i>True Negative</i>	4	5	6	6	8
<i>False Positive</i>	5	4	4	3	2
<i>False Negative</i>	4	3	2	2	0
<i>Accuracy</i>	55%	65%	70%	75%	90%
<i>Precision</i>	0,583	0,666	0,666	0,75	0,833
<i>Recall</i>	0,636	0,727	0,8	0,818	1
<i>f1-score</i>	0,608	0,695	0,727	0,782	0,909

Penelitian ini menunjukkan bahwa menggunakan *threshold* rendah sebesar 0.5 pada sistem deteksi masker menggunakan *convolutional neural network* arsitektur *VGG16* memberikan hasil prediksi yang akan lebih akurat. Dengan *threshold* ini, sistem mampu mencapai tingkat akurasi sebesar 90% dan nilai *f1-score* sebesar 0,909. Hasil evaluasi menunjukkan bahwa pengaturan *threshold* pada level 0.5 menghasilkan keseimbangan yang optimal antara tingkat klasifikasi benar dan tingkat klasifikasi salah. Penggunaan *threshold* rendah ini memungkinkan sistem untuk lebih sensitif dalam mengenali apakah seseorang menggunakan masker atau tidak, sehingga potensi kesalahan sistem saat klasifikasi individu yang sebenarnya menggunakan masker maupun tidak menggunakan masker dapat ditekan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dalam tugas akhir dengan judul "Sistem Deteksi Masker Pada Wajah Menggunakan *Convolutional Neural Network* arsitektur *VGG16*", kesimpulan yang dapat diambil adalah akurasi deteksi masker yang dikembangkan menggunakan *Convolutional Neural Network* arsitektur *VGG16* telah mencapai tingkat akurasi sebesar 90% dan *f1-score* sebesar 0,909 dengan tingkat pencahayaan yang baik. Hasil ini menunjukkan kemampuan yang baik dalam mengenali apakah seseorang menggunakan masker atau tidak. Namun, perlu diingat bahwa masih ada ruang untuk peningkatan akurasi melalui penyesuaian *training* model seperti *dataset*, maupun parameter-parameter yang digunakan pada proses *training*.

5.2 Saran

Berikut adalah saran untuk pengembangan lebih lanjut dari sistem deteksi masker menggunakan *convolutional neural network* arsitektur *VGG16* :

1. Sistem deteksi masker ini masih cukup bergantung pada pencahayaan yang baik dan tepat. Oleh karena itu, perlu diperhatikan untuk memastikan pencahayaan yang memadai saat pengujian dan penggunaan agar sistem dapat berfungsi dengan baik serta menghasilkan hasil yang konsisten.
2. Model yang dikembangkan pada sistem ini, belum mampu mendeteksi masker berwarna selain putih, penelitian lebih lanjut diharapkan dapat menghasilkan model yang mampu mendeteksi masker yang berwarna lain.
3. Untuk meningkatkan akurasi dan ketepatan sistem deteksi masker ini, disarankan untuk mengumpulkan dataset yang beragam. Namun perlu diperhatikan bahwa seiring dengan peningkatan jumlah dataset, beban dan waktu komputasi juga akan meningkat.
4. Model ini dapat diimplementasikan dan dikembangkan pada berbagai platform lain seperti web, android, dan lain sebagainya.

Dengan mengimplementasikan saran-saran tersebut, diharapkan sistem deteksi masker ini dapat terus ditingkatkan performanya, menjadi lebih baik, dan dapat memberikan manfaat secara luas bagi masyarakat dalam menghadapi situasi dan kondisi tertentu sehari-hari di masa mendatang.



DAFTAR PUSTAKA

- Doga, R.A. (2019) 'Sistem Identifikasi Nominal Uang Logam Menggunakan Tensorflow Dan Convolutional Neural Network Berbasis Raspberry Pi', (2018), pp. 123–126.
- Gunawan, R.J., Irawan, B. and Setianingsih, C. (2021) Pengenalan Ekspresi Wajah Berbasis Convolutional Neural Network Dengan Model Arsitektur Vgg16 Facial Expression Recognition Based On Convolutional Neural Network With Vgg16 Architecture Model.
- Kurniawan, R., Wintoro, P.B., Mulyani, Y. and Komarudin, M. (2023) 'Implementasi Arsitektur Xception Pada Model Machine Learning Klasifikasi Sampah Anorganik', *Jurnal Informatika dan Teknik Elektro Terapan*, 11(2). Available at: <https://doi.org/10.23960/jitet.v11i2.3034>.
- Kustriyani, M., Susanti, A.K. and Arifianto (2018) Hubungan Antara Motivasi Perawat Dengan Kepatuhan Penggunaan Alat Pelindung Diri (Handsoon Dan Masker) Di Instalasi Rawat Inap Rsud Dr. Loekmono Hadi Kudus.
- Kusumaningrum, T.F. (2018) 'Implementasi Convolution Neural Network (Cnn) Untuk Klasifikasi Jamur Konsumsi Di Indonesia Menggunakan Keras'. Available at: <https://doi.org/10.1002/j.1460-2075.1995.tb00098.x>.
- Maharil, A. (2022) 'Perbandingan Arsitektur Vgg16 Dan Resnet50 Untuk Rekognisi Tulisan Tangan Aksara Lampung', *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, 3(2), pp. 236–243. Available at: <http://jim.teknokrat.ac.id/index.php/informatika>.
- Maulana, M.A. (2020) 'Literatur Review : Pengetahuan Penderita Tuberkulosis Dalam Penggunaan Masker Untuk Mencegah Penularan Tuberkulosis'.
- Normawati, D. and Prayogi, S.A. (2021) Implementasi Naive Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter, *Jurnal Sains Komputer & Informatika (J-SAKTI)*.
- Nurona Cahya, F., Hardi, N., Riana, D. and Hadianti, S. (2021) Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN). Available at: <http://sistemasi.ftik.unisi.ac.id>.
- Pardede, J. and Hardiansah, H. (2022) 'Deteksi Objek Kereta Api menggunakan Metode Faster R-CNN dengan Arsitektur VGG 16', *MIND Journal*, 7(1), pp. 21–36. Available at: <https://doi.org/10.26760/mindjournal.v7i1.21-36>.
- Rambe, R. (2021) 'Perbaikan Kualitas Citra Digital Menggunakan Metode Kervel Konvolusi', *Terapan Informatika Nusantara*, 1(11), pp. 557–562. Available at: <https://ejurnal.seminar-id.com/index.php/tin>.
- Rasywir, E., Sinaga, R. and Pratama, Y. (2020) 'Analisis dan Implementasi Diagnosis Penyakit Sawit dengan Metode Convolutional Neural Network (CNN)', 22(2). Available at: <https://doi.org/10.31294/p.v21i2>.

- Setiawan, D., Widodo, S., Ridwan, T. and Ambari, R. (2022) Perancangan Deteksi Emosi Manusia Berdasarkan Ekspresi Wajah Menggunakan Algoritma VGG16, *Syntax: Jurnal Informatika*.
- Simanjuntak, M.R., Rahmayanti, L. and Ginting, R. (2022) 'Analysis Of Factors That Affect Compliance In Using Masks To Prevent The Spread Of Covid-19'.
- Sutari, S., Idris, H. and Misnaniarti, M. (2022) 'Implementasi Kebijakan Vaksinasi Covid-19 Di Indonesia: Narrative Review', *Riset Informasi Kesehatan*, 11(1), p. 71. Available at: <https://doi.org/10.30644/rik.v11i1.637>.
- Taufiq, A.J., Tamam, M.T. and Susiyadi, S. (2022) 'Perancangan Sistem Buka Tutup Pintu Area Terbatas Berdasarkan Deteksi Masker', *JRST (Jurnal Riset Sains dan Teknologi)*, 6(2), p. 221. Available at: <https://doi.org/10.30595/jrst.v6i2.15449>.
- Tuntun, M. (2022) Pola Bakteri Kontaminan Serta Resistensinya di ICU dan Ruang Operasi Pada Rumah Sakit di Bandar Lampung Pola Bakteri Kontaminan Serta Resistensinya di ICU dan Ruang Operasi Pada Rumah Sakit di Bandar Lampung.
- Wardhani, R.N. and Delimayanti, M.K. (2011) Analisis Penerapan Metode Konvolusi Untuk Untuk Reduksi Derau Pada Citra Digital.
- Wulan, A. (2020) 'Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (Cnn) Dengan Tensorflow', 28(2), pp. 1-43. Available at: <http://repositorio.unan.edu.ni/2986/1/5624.pdf>.
- Zufar, M. and Setiyono, B. (2016) 'Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time', *Jurnal Sains dan Seni ITS*, 5(2), p. 128862. Available at: <https://doi.org/10.12962/j23373520.v5i2.18854>.

