



**KOMPARASI ALGORITMA *CONVOLUTIONAL NEURAL NETWORK*
DAN *TRANSFER LEARNING* (RESTNET 50, MOBILE NET V2, DAN
VGG16) UNTUK IDENTIFIKASI EKSPRESI WAJAH POTENSI
KEJAHATAN**

TUGAS AKHIR

AFRIZAL NAJWA SYAUQI

NPM 20670117

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PGRI SEMARANG**

2024



**KOMPARASI ALGORITMA *CONVOLUTIONAL NEURAL NETWORK*
DAN *TRANSFER LEARNING* (RESTNET 50, MOBILE NET V2, DAN
VGG16) UNTUK IDENTIFIKASI EKSPRESI WAJAH POTENSI
KEJAHATAN**

TUGAS AKHIR

**Diajukan kepada Fakultas Teknik dan Informatika
Universitas PGRI Semarang untuk Penyusunan Skripsi**

AFRIZAL NAJWA SYAUQI

NPM 20670117

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS PGRI SEMARANG**

2024

TUGAS AKHIR

Komparasi Algoritma *Convolutional Neural Network* dan Transfer Learning (ResNet 50, MobileNet V2, dan VGG16) Untuk Identifikasi Ekspresi Wajah Potensi Kejahatan

Disusun dan diajukan oleh

Afrizal Najwa Syauqi

NPM 20670117

Telah dipertahankan di depan Dewan Penguji pada tanggal 16 Agustus 2024 dan dinyatakan telah memenuhi syarat Dewan Penguji

Ketua



Ibnu Toto Husodo, S.T., M.T.
NIP/NPP 136901387

Sekretaris



Bambang Agus H., S.Kom, M.Kom.
NIP/NPP 148201433

Penguji I




Bambang Agus H., S.Kom, M.Kom.
NIP/NPP 148201433

Penguji II



Khoiriya Latifah, S.Kom., M.Kom.
NIP/NPP 147801434

Penguji III



Noora Qotrun Nada, S.T., M.Eng.
NIP/NPP 158201485

MOTTO DAN PERSEMBAHAN

Motto :

“Makna kesuksesan sebenarnya adalah bagaimana kita bisa memberikan banyak manfaat dan berguna bagi orang banyak, meskipun itu hanya selembar roti”

Persembahan :

Kupersembahkan skripsi ini untuk:

1. Kedua orang tuaku tercinta.
2. Teman-temanku yang selalu menyemangatiku mengerjakan skripsi.
3. Almamaterku Universitas PGRI Semarang.

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini :

Nama : Afrizal Najwa Syauqi
NPM : 20670117
Program Studi : Informatika
Fakultas : Teknik dan Informatika
Instansi : Universitas PGRI Semarang

Menyatakan dengan sebenarnya bahwa skripsi yang saya buat ini benar-benar merupakan hasil karya saya sendiri, bukan plagiarisme.

Apabila pada kemudian hari skripsi ini terbukti hasil plagiarisme, saya bersedia menerima sanksi atas perbuatan tersebut.

Semarang, 16 Agustus 2024

Yang membuat pernyataan



Afrizal Najwa Syauqi

NPM 20670117

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan dan menerapkan algoritma *Convolutional Neural Network* (CNN) dalam mendeteksi ekspresi wajah yang mencurigakan guna mencegah tindak kejahatan di fasilitas publik. Sistem keamanan publik saat ini menghadapi tantangan besar dalam mendeteksi tindakan kriminal dengan cepat dan akurat. Oleh karena itu, penelitian ini bertujuan untuk: (1) merancang sistem prediksi ekspresi wajah yang dapat membantu petugas keamanan dalam memantau dan mencegah kejahatan, (2) mengevaluasi efektivitas model deep learning yang diterapkan pada rekaman CCTV, dan (3) menerapkan langkah-langkah keamanan siber untuk melindungi sistem CCTV dari akses tidak sah. Metode yang digunakan adalah pendekatan kuantitatif dengan memanfaatkan algoritma CNN. Data ekspresi wajah diperoleh dari dataset publik dan digunakan untuk melatih model CNN. Model ini kemudian diuji untuk menentukan akurasi dalam mendeteksi ekspresi wajah yang mencurigakan seperti marah, takut, dan terkejut, yang berpotensi mengindikasikan niat jahat. Selain itu, langkah-langkah keamanan siber diterapkan untuk memastikan bahwa sistem CCTV tetap terlindungi dari serangan siber. Hasil penelitian menunjukkan bahwa model CNN yang dikembangkan mampu mendeteksi ekspresi wajah mencurigakan dengan tingkat akurasi yang tinggi, sehingga dapat digunakan untuk mendukung pencegahan kejahatan di fasilitas publik. Implementasi langkah-langkah keamanan siber juga terbukti efektif dalam melindungi sistem dari ancaman eksternal. Penelitian ini memberikan kontribusi signifikan dalam peningkatan keamanan publik melalui teknologi deteksi ekspresi wajah yang canggih dan aman.

Kata Kunci: Convolutional Neural Network, Deteksi Ekspresi Wajah, Keamanan Publik, CCTV, Keamanan Siber.

PRAKATA

Dengan terima kasih kepada Allah SWT, peneliti dapat menyusun dan menyelesaikan skripsi ini sebagai salah satu syarat untuk memperoleh gelar sarjana teknik informatika adalah skripsi dengan judul "Komparasi Algoritma Convolutional Neural Network Dan Transfer Learning (Resnet 50, Mobile Net V2, Dan VGG16) Untuk Identifikasi Ekspresi Wajah Potensi Kejahatan" Proses penyusunan skripsi ini tidak lepas dari hambatan dan rintangan. Namun, semua tantangan, bantuan, nasihat, motivasi, dan saran dari berbagai pihak, khususnya pembimbing, dapat diatasi dengan baik.

Oleh karena itu, penulis dengan tulus mengucapkan terima kasih kepada:

1. Rektor Universitas PGRI Semarang yang telah memberi kesempatan kepada penulis untuk menimba ilmu di universitas PGRI Semarang
2. Bapak Ibnu Toto Husodo, S.T., M.T. Dekan Fakultas Teknik Dan Informatika yang telah memberikan izin penulis untuk melakukan penelitian
3. Bambang Agus Herlambang, S.Kom., M.Kom. selaku Ketua Program Studi Informatika yang telah menyetujui topik skripsi penulis dan selaku pembimbing 1 yang telah mengarahkan penulis dengan penuh ketekunan dan kecermatan
4. Khoiriya Latifah, S.Kom., M.Kom. selaku pembimbing dua yang telah membimbing penulis dengan penuh dedikasi yang tinggi
5. Bapak Ibu Dosen Program Studi Informatika yang telah memberi bekal ilmu kepada penulis selama belajar di universitas PGRI Semarang dan PT titik-titik yang telah mengizinkan peneliti melakukan penelitian di instansi yang dipimpinnya

Akhirnya penulis berharap semoga skripsi ini bermanfaat bagi pendidik khususnya pendidik di dunia Pendidikan dan menengah

DAFTAR ISI

SAMPUL LUAR	i
SAMPUL DALAM	ii
BAB I HALAMAN PENGESAHAN	iii
MOTTO DAN PERSEMBAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
ABSTRAK	vi
PRAKATA	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Identifikasi Masalah	2
C. Pembatasan Masalah	3
D. Perumusan Masalah	3
E. Tujuan Penelitian	3
F. Manfaat Penelitian	4
G. Penegasan Istilah	4
BAB II KAJIAN PUSTAKA	6
A. Tinjauan Pustaka	6
B. Landasan Teori	11
C. Kerangka Berfikir	37
D. Hipotesis Penelitian	38

BAB III METODE PENELITIAN	40
A. Pendekatan Penelitian	40
B. Fokus Penelitian	40
C. Populasi Dan Sampel	41
D. Variabel Penelitian	42
E. Teknik Pengumpulan Data	43
F. Teknik Analisis Data	43
BAB IV HASIL PENGEMBANGAN DAN PEMBAHASAN	46
A. Hasil	46
B. Pembahasan	7
BAB V KESIMPULAN DAN SARAN	11
A. Kesimpulan	11
B. Saran	12
DAFTAR PUSTAKA	14
LAMPIRAN	18

DAFTAR TABEL

Tabel 2. 1 Tinjauan Pustaka	6
Tabel 2. 2 Jumlah Dataset FER-2013	22
Tabel 2. 3 Use Case Diagram	32
Tabel 2. 4 Activity Diagram	33
Tabel 2. 5 Komponen Pertanyaan System Usability Scale	35
Tabel 2. 6 Kategori Penilaian SUS	36
Tabel 4. 1 Tabel Perbandingan Jumlah Dataset	47
Tabel 4. 2 Hasil Uji Coba ResNet50	63
Tabel 4. 3 Hasil Uji Coba MobileNetV2	65
Tabel 4. 4 Hasil Train VGG16	67
Tabel 4. 5 Inisialisasi Library	89
Tabel 4. 6 White Box Testing	100
Tabel 4. 7 Form Black Box Testing	103
Tabel 4. 8 Form System Usability Scale (SUS)	4
Tabel 4. 9 Hasil Evaluasi SUS	6

DAFTAR GAMBAR

Gambar 2. 1 Python	12
Gambar 2. 2 Pycharm	13
Gambar 2. 3 Deep Neural Network	15
Gambar 2. 4 Artificial Neural Network (ANN)	15
Gambar 2. 5 Convolutional Neural Network (CNN)	16
Gambar 2. 6 Recurrent Neural Network (RNN)	17
Gambar 2. 7 Convolutional Neural Network (CNN)	xviii
Gambar 2. 8 Convolutional Layer	xix
Gambar 2. 9 Pooling Layer	xix
Gambar 2. 10 Kaggle	21
Gambar 2. 11 FER-2013 Dataset	21
Gambar 2. 12 OpenCV	24
Gambar 2. 13 Tensorflow	26
Gambar 2. 14 Fernet Symmetric Encryption	29
Gambar 2. 15 Metode Prototype	31
Gambar 2. 16 Kerangka Berfikir	38
Gambar 4. 1 Load Dataset	1
Gambar 4. 2 Path Gambar dan Label	50
Gambar 4. 3 Preprocess Image	51
Gambar 4. 4 Hyperparameter Tuning	54
Gambar 4. 5 Konfigurasi Arsitektur CNN	58
Gambar 4. 6 Fungsi create_model	59
Gambar 4. 7 Learning Rate Setup	61
Gambar 4. 8 Implementasi Model ResNet50	62
Gambar 4. 9 Arsitektur Model ResNet50	63
Gambar 4. 10 Implementasi Model MobileNetV2	64
Gambar 4. 11 Arsitektur MobileNetV2	64
Gambar 4. 12 Implementasi FineTuning pada Model VGG16	66
Gambar 4. 13 Arsitektur VGG16	67
Gambar 4. 14 Hasil Evaluasi Model dari Training Menggunakan VGG16	68

Gambar 4. 15 Grafik Visualisasi Loss dan Accucary dari Training Model dengan VGG16	70
Gambar 4. 16 Arsitektur CNN	71
Gambar 4. 17 Code Training Model Menggunakan Hasil Hyperparameter Tuning	71
Gambar 4. 18 Confussion Matrix	76
Gambar 4. 19 Grafik Loss dan Accuracy Model	76
Gambar 4. 20 ROC	77
Gambar 4. 21 Use Case Diagram	78
Gambar 4. 22 Activity Diagram : Homepage	79
Gambar 4. 23 Activity Diagram : Detection Page	82
Gambar 4. 24 Activity Diagram : Contact	83
Gambar 4. 25 Wireframe Halaman Hone	83
Gambar 4. 26 Wireframe Halaman Detection Sebelum Kamera Aktif	84
Gambar 4. 27 Wireframe Halaman Detection Setelah Kamera Aktif	84
Gambar 4. 28 Wireframe Halaman Kontak	84
Gambar 4. 29 Prototype Halaman Home	85
Gambar 4. 30 Prototype Halaman Detection	86
Gambar 4. 31 Prototype Halaman Kontak	87
Gambar 4. 32 Import Library Development Web	91
Gambar 4. 33 Arsitektur CNN	92
Gambar 4. 34 Load Model	93
Gambar 4. 35 Database Setup	94
Gambar 4. 36 Enkripsi Simetris Menggunakan Fernet	95
Gambar 4. 37 Fungsi predict_emotion dan save_to_database	96
Gambar 4. 38 Detection Function	97
Gambar 4. 39 Hasil Website : Homepage	98
Gambar 4. 40 Hasil Website : Detection Page	99
Gambar 4. 41 Hasil Website : Kontak	100
Gambar 4. 42 Flowgraph	102

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Ekspresi wajah merupakan salah satu cara dari manusia untuk mengungkapkan apa yang sedang mereka pikirkan atau rasakan. Menurut Oliver dan Alcove (2020) [1] satu ekspresi dapat memberikan informasi yang lebih banyak daripada kata-kata. Keunggulan dari ekspresi wajah yaitu kemampuannya untuk membantu dalam memahami emosi seseorang. Menurut Planalp (2015) [1] pada jurnal yang berjudul “How Important Is Emotion in Everyday Interaction?” ketika seseorang mengalami ketidakstabilan emosi, perubahan emosi tersebut dapat terlihat pada wajahnya, seperti dahi yang mengerut, mata yang berkedip, atau perubahan warna pada kulit. Salah satu contohnya, ketika seseorang merasa marah, akan terlihat tanda-tanda seperti mengerutkan kening dan perubahan warna pada wajah yang menjadi merah.

Keinginan seseorang lebih mudah dianalisis melalui ekspresi yang dipancarkan daripada menganalisisnya melalui sikap dan gestur badan. Hal ini dikarenakan kata-kata yang diucapkan kadang tidak sesuai dengan maksud yang diinginkan. Seorang psikolog senior asal Amerika, Paul Ekman [2] mendefinisikan 6 emosi dasar yang sama, yaitu marah, senang, sedih, jijik, terkejut, dan takut. Sampai saat ini enam dasar tersebut masih menjadi acuan yang dipakai oleh para psikolog dan para pengembang sistem pendeteksi wajah untuk mengklasifikasikan maksud seseorang berdasarkan ekspresi wajah hingga saat ini.

Pembacaan ekspresi ini sering digunakan dalam upaya untuk pencegahan kejahatan. Kejahatan merupakan salah satu hal yang tidak bisa dimungkiri lagi keberadaannya. Kejahatan bisa terjadi kapan saja, dimanapun, dan pada siapapun tidak memandang umur, usia, dan jenis kelamin. Banyak upaya yang telah dilakukan oleh kepolisian untuk mencegah dan menangani kejahatan. Namun, masih saja terjadi kejahatan di sekitar kita. Pada tahun 2019, terhitung sebanyak 38.983 total kasus kriminal terjadi di Indonesia berdasarkan

klasifikasi kriminal kejahatan fisik [3]. Salah satu upaya pencegahan yang telah dilakukan sejauh ini adalah dengan memasang CCTV. Dengan adanya CCTV akan mempermudah petugas dalam memonitoring keadaan sekitar secara realtime. Namun, dalam pengawasan CCTV, diperlukan adanya suatu sistem yang memudahkan manusia dalam memantau dan memonitoring hasil CCTV. Apalagi ketika suatu CCTV itu diterapkan pada suatu fasilitas publik, dimana banyak sekali orang yang berlalu lalang, untuk ukuran manusia tentu ada keterbatasan tertentu dalam memonitoring hasil dari CCTV. Penjaga itu bisa saja mengantuk, kurang teliti dalam memantau dan sebagainya, atau biasa kita sebut dengan human error. Salah satu teknologi yang bisa kita terapkan yaitu teknologi *Deep Learning*.

Deep Learning merupakan metode learning yang memanfaatkan *artificial neural network* yang berlapis-lapis(multi layer) [4]. *Artificial Neural Network* ini dibuat mirip otak manusia, dimana neuron-neuron terkoneksi satu sama lain sehingga membentuk sebuah jaringan neuron yang sangat rumit. *Deep Learning* atau *deep structured learning* atau *hirarchial learning* atau *deep neural* merupakan metode learning yang memanfaatkan multiple non-linier transformation, *deep learning* dapat dipandang sebagai gabungan *machine learning* dengan AI (artificial neural network). Selain ANN, ada juga metode Convolutional Neural Network (CNN), CNN adalah yang terbaik untuk pengenalan citra digital karena didasarkan pada sistem pengenalan citra pada visual cortex manusia. CNN juga menggunakan jaringan syaraf tiruan untuk menyelesaikan masalah dengan belajar dari data sebelumnya. Untuk itu, penerapan ini penting untuk dibuat karena dapat mempermudah proses monitoring keadaan sekitar dan membantu petugas keamanan dalam mencegah tindak kejahatan.

B. Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah diuraikan diatas, maka identifikasi masalah yang dijadikan bahan penelitian yaitu sebagai berikut :

- 1) Keterbatasan kemampuan dari manusia dalam mengawasi keadaan secara terus menerus
- 2) Tingkat kriminalitas yang masih cukup tinggi
- 3) Sulitnya mendeteksi ancaman kejahatan dalam keramaian melalui CCTV
- 4) Belum efektifnya pemanfaatan dari CCTV untuk pengawasan kondisi keramaian

C. Pembatasan Masalah

Dari identifikasi masalah yang diterapkan dalam penelitian ini, maka dirasa perlu dilakukan pembatasan masalah agar dalam pengkajian yang dilakukan lebih fokus kepada masalah – masalah yang akan dipecahkan. Penelitian ini menitik beratkan kepada sistem deteksi ekspresi manusia dan prediksi maksud mencurigakan dan tidak mencurigakan pada orang-orang yang ada di fasilitas publik.

D. Perumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka rumusan masalah yang dijadikan penelitian yaitu sebagai berikut :

- 1) Bagaimana rancangan sistem deteksi ekspresi dapat membantu petugas keamanan dalam memonitoring dan mencegah kejahatan?
- 2) Bagaimana model deep learning CNN yang efektif untuk diterapkan dalam kamera keamanan supaya validitas hasilnya terjamin?
- 3) Bagaimana penerapan langkah-langkah keamanan siber pada sistem CCTV untuk mencegah akses oleh pihak yang tidak berwenang?

E. Tujuan Penelitian

Sejalan dengan rumusan masalah yang telah dipaparkan sebelumnya, maka tujuan yang diharapkan dari penelitian ini adalah sebagai berikut :

- 1) Untuk mengetahui rancangan sistem prediksi ekspresi dapat membantu petugas keamanan dalam memonitoring dan mencegah kejahatan
- 2) Untuk mengetahui model deep learning CNN yang efektif untuk diterapkan dalam CCTV supaya validitas hasilnya terjamin

- 3) Menerapkan langkah-langkah keamanan siber pada sistem CCTV untuk menghindari akses tidak sah oleh pihak yang tidak berwenang.

F. Manfaat Penelitian

Adapun manfaat yang ingin dicapai dari hasil penelitian ini adalah sebagai berikut.

1) Manfaat Teoritis

Memberikan kontribusi pada pengembangan teknologi pengenalan ekspresi wajah berbasis deep learning yang dapat meningkatkan efektivitas sistem keamanan publik.

2) Manfaat Praktis

a) Bagi Masyarakat

Meningkatkan kepercayaan masyarakat kepada fasilitas publik dan dapat memanfaatkan fasilitas publik dengan lebih nyaman.

b) Bagi Petugas Keamanan

Membantu dalam melaksanakan pekerjaannya dan dapat mencegah kejahatan dengan lebih baik lagi

c) Bagi Peneliti dan Mahasiswa

Menambah pengalaman dalam pengembangan dan penerapan teknologi deep learning pada sistem keamanan, serta sebagai referensi bagi penelitian selanjutnya.

G. Penegasan Istilah

Untuk menghindari terjadinya pembahasan yang melebar, kesalahpahaman interpretasi serta memudahkan pemahaman tentang judul tersebut di atas, maka penulis merasa perlu untuk memberikan pembahasan istilah yang terdapat pada judul diatas.

1. *Convolutional Neural Network (CNN)*

Convolutional Neural Network dianggap sebagai model terbaik untuk menyelesaikan masalah pengenalan objek dan pengenalan wajah. CNN yang dimaksud pada skripsi ini adalah salah satu jenis algoritma pembelajaran deep

learning yang efektif dalam mengenali ekspresi wajah dengan tingkat keakuratan yang tinggi.

2. Ekspresi Mencurigakan dan Tidak Mencurigakan

Dalam pengembangan sistem prediksi ekspresi, peneliti menggunakan dua parameter dalam menunjukkan tindak kejahatan, yaitu “Mencurigakan” dan “Tidak Mencurigakan”. Dasar penulis dalam menentukan parameter “Mencurigakan” adalah ekspresi marah, jijik, takut, sedih, dan terkejut. Sedangkan, untuk parameter “Tidak Mencurigakan”, penulis berdasar dari ekspresi senang dan netral.

BAB II

KAJIAN PUSTAKA

A. Tinjauan Pustaka

Penelitian mengenai implementasi algoritma *convolutional neural network (CNN)* dengan identifikasi ekspresi wajah untuk deteksi potensi kejahatan ini dibuat berdasarkan tinjauan terhadap penelitian – penelitian sebelumnya, yang dijadikan sumber informasi terhadap penelitian ini.

Tabel 2. 1 Tinjauan Pustaka

No.	Penulis	Judul	Keterangan
1.	Ali Mollahosseini, David Chan, and Mohammad H. Mahoor (2016)	Going Deeper in Facial Expression Recognition using Deep Neural Networks	Pendekatan FER otomatis mengklasifikasikan wajah sebagai salah satu dari enam emosi dasar dalam satu atau lebih gambar. Selain itu, sangat sulit untuk mendapatkan data pelatihan yang akurat, terutama untuk emosi seperti kesedihan atau ketakutan, yang sangat sulit untuk ditiru dengan benar dan tidak sering terjadi dalam kehidupan nyata.
2.	Ciprian A. Corneanu, Marc Oliu, Jeffrey F. Cohn, and Sergio Escalera (2016)	Survey on RGB, 3D, Thermal, and multimodal Approaches for Facial Expression Recognition: History, Trends,	Metode yang disarankan memiliki keuntungan yang jelas dibandingkan dengan metode konvensional CNN, yaitu jaringan yang lebih dangkal atau lebih tipis. Metode ini meningkatkan

		and Affect-Related Applications	akurasi klasifikasi pada skenario evaluasi subjek independen dan lintas basis data sekaligus mengurangi jumlah operasi yang diperlukan untuk melatih jaringan.
3.	Sunardi, Fadhil Abdul, Prayogi Denis (2022)	Sistem Pengenalan Wajah pada Keamanan Ruang Berbasis <i>Convolutional Neural Network</i>	Penelitian ini merancang sistem pengenalan wajah untuk keamanan ruangan menggunakan metode <i>Convolutional Neural Network</i> (CNN). CNN bekerja dengan meniru cara kerja jaringan saraf tiruan pada manusia untuk memproses citra. Dalam penelitian ini, citra wajah diambil menggunakan kamera webcam yang terpasang pada perangkat berbasis Raspberry Pi, dengan pemrograman Python dan pustaka TensorFlow. Dari 875 sampel data yang dibagi menjadi 75% untuk pelatihan dan 25% untuk pengujian, sistem ini mencapai akurasi prediksi 100%, menunjukkan bahwa semua data berhasil dikenali dengan benar.

4.	PA Nugroho, I Fenriana, R Arijanto (2020)	Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia	<ol style="list-style-type: none"> 1. Jumlah dataset tidak sepenuhnya mempengaruhi nilai akurasi, tetapi kedetailan citra untuk digunakan dataset sangat mempengaruhi hasil akurasi. 2. Dari percobaan yang telah dilakukan didapatkan hasil precision, recall, dan akurasi sebesar 65%. 3. Metode Convolutional Neural Network (CNN) sangat cocok digunakan untuk menguji sebuah citra, karena prosesnya yang berlapis-lapis, terbukti dengan 35 citra, 28 citra bisa ditebak dengan benar walaupun ekspresi hanya berbeda tipis-tipis.
5.	A Agustinus, R Kurniawan, HOL Wijaya (2023)	Klasifikasi Emosi Melalui Ekspresi Wajah Menggunakan Algoritma Deep Learning	<ol style="list-style-type: none"> 1. Sistem yang dibuat bekerja sesuai dengan yang diharapkan hal ini dapat dibuktikan dengan hasil mAP sebesar 0.96 2. Hasil training dataset menggunakan algoritma YOLOv5 mendapatkan

			<p>nilai akurasi yang tinggi yaitu 87%.</p> <p>3. Sistem mampu mendeteksi ekspresi wajah senang, sedih dan kaget dengan benar.</p>
6.	YNB Ginting, IK Swardika, DAIC Dewi (2023)	Deteksi Emosi Anak Dari Ekspresi Wajah Dengan Deep Learning Untuk Menilai Kesehatan Mental	<p>Dalam penelitian ini, dilakukan deteksi emosi anak melalui ekspresi wajah dengan mengimplementasikan algoritma Convolutional Neural Network, Tensorflow Lite dan library OpenCV. Tensorflow Lite dijalankan melalui python dan digunakan untuk membangun, melatih, dan menerapkan model machine learning.</p>

Pendekatan tradisional yang menggunakan fitur buatan tangan seperti HOG, LBPH, dan Gabor, sering kali tidak generalisasi dengan baik pada gambar yang tidak dikenal atau diambil dalam kondisi liar. Untuk mengatasi masalah ini, penulis mengusulkan arsitektur jaringan saraf dalam dengan dua lapisan konvolusi yang masing-masing diikuti oleh max pooling dan empat lapisan Inception. Arsitektur ini diuji pada tujuh database ekspresi wajah publik dan menunjukkan hasil yang sebanding atau lebih baik dibandingkan metode state-of-the-art dan CNN tradisional dalam hal akurasi dan waktu pelatihan. Penelitian ini [5] menunjukkan bahwa menggunakan jaringan saraf dalam dapat meningkatkan kemampuan generalisasi dan akurasi dalam mengenali ekspresi wajah di berbagai database.

Pendekatan awal dalam AFER sebagian besar mengandalkan representasi geometris dan penampilan untuk mendeteksi ekspresi wajah. Seiring waktu, penelitian di bidang ini berkembang untuk mencakup pengenalan ekspresi spontan, analisis kondisi kompleks, serta penggunaan modalitas lain seperti citra 3D dan termal. Tren terkini dalam AFER mencakup upaya untuk meningkatkan akurasi deteksi melalui estimasi intensitas ekspresi dengan menggunakan teknik penggabungan fitur geometris dan penampilan. Selain itu, eksplorasi penggunaan modalitas 3D dan RGB telah menunjukkan potensi yang signifikan dalam meningkatkan kemampuan sistem untuk mendeteksi dan mengenali ekspresi wajah dengan lebih akurat [6].

Sistem pengenalan wajah berbasis *Convolutional Neural Network* (CNN) ini dirancang untuk meningkatkan keamanan ruangan dengan memanfaatkan cara kerja jaringan saraf tiruan dalam mengenali pola pada citra wajah. Menggunakan perangkat Raspberry Pi dan kamera webcam, citra wajah diambil dan diproses menggunakan bahasa pemrograman Python dengan pustaka TensorFlow. Data yang digunakan dalam penelitian ini terdiri dari 875 sampel, dengan pembagian 75% untuk pelatihan dan 25% untuk pengujian. Hasil penelitian menunjukkan akurasi yang luar biasa, yaitu 100%, yang berarti semua data pengujian berhasil dikenali dengan tepat, menegaskan efektivitas sistem dalam aplikasi keamanan[7].

Penelitian ini mengimplementasikan metode deep learning menggunakan Convolutional Neural Network (CNN) untuk mengenali ekspresi wajah manusia. CNN, yang dirancang khusus untuk pengenalan dan klasifikasi gambar, terdiri dari beberapa lapisan yang mengekstrak informasi dari gambar dan menghasilkan skor klasifikasi. Penelitian menggunakan dataset ekspresi wajah dengan berbagai emosi seperti senang, sedih, takut, jijik, netral, marah, dan kaget. Proses training dilakukan dengan berbagai ukuran batch dan epoch untuk mendapatkan model terbaik. Hasil penelitian menunjukkan bahwa dengan epoch 100 dan batch size 128, akurasi training mencapai 90% dan

akurasi validasi mencapai 65%, dengan 28 dari 35 ekspresi berhasil dikenali dengan benar, menghasilkan akurasi 80% [8].

Penelitian ini bertujuan untuk mengembangkan sistem pendeteksi ekspresi wajah menggunakan algoritma deep learning, khususnya YOLOv5. Dataset yang digunakan terdiri dari 1190 foto yang dilabeli menjadi tiga kelas: senang, sedih, dan kaget menggunakan Roboflow. Proses pelatihan menggunakan YOLOv5 menghasilkan nilai mAP sebesar 0.96. Pengujian real-time menunjukkan sistem mencapai akurasi 87% dalam mendeteksi ekspresi wajah. Hasil ini menunjukkan bahwa sistem dapat mengenali ekspresi wajah dengan tingkat akurasi yang tinggi, meskipun jumlah dataset tidak sepenuhnya menentukan nilai akurasi, melainkan detail gambar yang digunakan dalam dataset yang lebih berpengaruh[9].

Penelitian ini [10] mengembangkan sistem deteksi emosi anak melalui analisis ekspresi wajah menggunakan deep learning, khususnya dengan algoritma Convolutional Neural Network (CNN). Dataset yang digunakan terdiri dari gambar ekspresi wajah anak yang telah dikategorikan dalam beberapa jenis emosi. Model CNN dilatih menggunakan TensorFlow Lite dan OpenCV di Google Colaboratory untuk mengenali pola dan mengklasifikasikan ekspresi wajah anak. Hasil penelitian menunjukkan bahwa sistem ini efektif dalam mendeteksi emosi anak dengan tingkat akurasi mencapai 83%, membantu dalam memahami kesehatan mental anak melalui ekspresi wajah mereka.

B. Landasan Teori

Landasan teori merupakan bagian penting dalam sebuah penelitian yang berfungsi sebagai dasar ilmiah untuk memahami fenomena yang diteliti. Bagian ini memaparkan konsep-konsep, teori-teori, dan penelitian-penelitian sebelumnya yang relevan dengan topik yang sedang dibahas. Dengan adanya landasan teori, peneliti dapat mengkontekstualisasikan masalah penelitian, membangun argumen yang kuat, dan menjelaskan hubungan antar variabel.

Selain itu, landasan teori membantu dalam merumuskan hipotesis dan metodologi penelitian yang akan digunakan.

1. Python



Gambar 2. 1 Python

Python adalah bahasa pemrograman yang ditafsirkan, interaktif, dan berorientasi objek. Bahasa ini menyediakan struktur data tingkat tinggi seperti daftar dan larik asosiatif (disebut kamus), pengetikan dinamis dan pengikatan dinamis, modul, kelas, pengecualian, manajemen memori otomatis, dll. Python memiliki sintaks yang sangat sederhana dan elegan dan namun merupakan bahasa pemrograman yang kuat dan memiliki tujuan umum. Bahasa ini dirancang pada tahun 1990 oleh Guido van Rossum. Seperti banyak bahasa skrip lainnya, bahasa ini gratis, bahkan untuk tujuan komersial, dan dapat dijalankan secara praktis komputer modern apa pun. Program python dikompilasi secara otomatis oleh penerjemah ke dalam platform independen kode byte yang kemudian ditafsirkan [11].

Python pada dasarnya bersifat modular. Kernelnya sangat kecil dan dapat diperluas dengan mengimpor mod ekstensi. Distribusi Python mencakup beragam pustaka ekstensi standar (beberapa ditulis dalam Python, yang lain lainnya dalam C atau C++) untuk operasi mulai dari manipulasi string dan ekspresi reguler seperti Perl, hingga generator *Graphical Graphical User Interface* (GUI) dan termasuk utilitas terkait web, layanan sistem operasi, debugging dan alat pembuatan profil, dll.

- 1) Manfaat Bahasa Pemrograman Python
 - a) Ekstensi Numerik Python

Numeric Python adalah modul ekstensi untuk penyimpanan yang efisien dan manipulasi data paralel data numerik. Dengan menggunakan modul ini, banyak operasi sederhana yang terlalu lambat dalam Python dapat dilakukan dengan sangat efisien (pada dasarnya secepat di C) tanpa harus mengimplementasikan kode di C. Modul ini sangat sering memungkinkan kita untuk menulis ekstensi dalam Python (menggunakan Numeric) yang cukup efisien untuk tujuan kita dan tidak perlu dikodekan ulang dalam bahasa C atau C++.

b) Masalah kinerja

Sering kali ada kekhawatiran tentang harga yang dibayarkan dalam hal kinerja ketika menggunakan alat ukur yang ditafsirkan. Penulis telah belajar bahwa, pertama-tama, kami memiliki konsepsi yang salah tentang di mana, kapan, dan berapa banyak kinerja manusia yang kita butuhkan. Selain itu, kode Python berjalan dengan performa yang cukup memadai untuk banyak tugas umum. Terakhir, memiliki ekstensi Numeric membantu menjaga kinerja yang baik bahkan ketika bekerja dengan besar bahkan ketika bekerja dengan array angka[11].

2. Pycharm IDE



Gambar 2. 2 Pycharm

Pycharm menonjol dari segi persaingan karena produktivitas tools yang ada di dalamnya seperti quick fixes (perbaikan cepat)[12]. Tersedia dalam tiga versi, versi Komunitas berlisensi Apache, versi Pendidikan (Edu), dan versi Profesional berpemilik. Dua versi pertama adalah open source dan dapat dikembangkan secara gratis, sedangkan versi Profesional merupakan berbayar. Versi Komunitas dari IDE ini mungkin akan sangat menarik bagi Pengembang karena memiliki fitur

berbeda seperti penyorotan sintaks, auto-completion, dan verifikasi kode secara langsung. Versi berbayar memiliki fitur yang lebih canggih seperti manajemen database penuh dan banyak framework yang lebih penting daripada versi komunitas seperti Django, Flask, Google App, Engine, Pyramid, dan web2py [12].

Keunggulan:

- a) Dukungan komunitas yang aktif.
- b) Verifikasi kode langsung dan penyorotan sintaksis
- c) Menjalankan pengeditan dan debug kode Python tanpa persyaratan eksternal apa pun

Kekurangan:

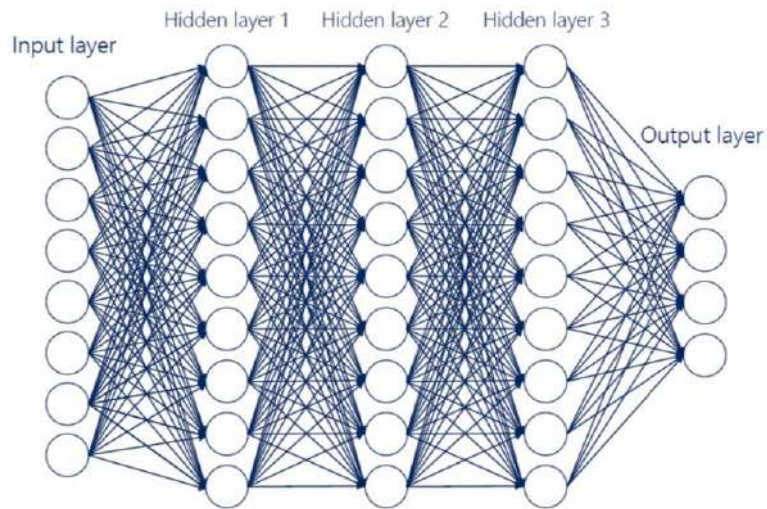
- a) Waktu pemuatan yang cenderung lambat
- b) Pengaturan default mungkin memerlukan penyesuaian sebelum proyek yang ada dapat digunakan.

3. Deep Learning (DL)

Deep Learning (DL) adalah teknik dalam NN yang menggunakan teknik tertentu seperti *Restricted Boltzmann Machine* (RBM) untuk mempercepat proses pembelajaran dalam NN yang menggunakan lapis yang banyak atau lebih dari 7 lapis [13] . Dengan adanya DL, waktu yang dibutuhkan untuk training akan semakin sedikit karena masalah hilangnya gradien pada propagasi balik akan semakin rendah.

Beberapa jenis DL antara lain adalah :

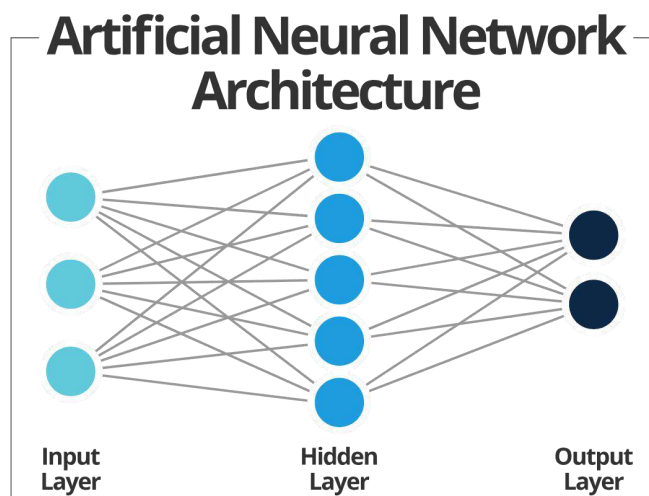
1) *Deep Neural Network (DNN)*



Gambar 2. 3 Deep Neural Network

Algoritma jaringan saraf mendalam yang paling umum digunakan untuk menganalisis fitur atau variabel adalah *Deep Neural Network (DNN)*, yang merupakan perangkat multilapis yang menghubungkan setiap neuron ke neuron di lapisan berikutnya [14]. DNN adalah algoritma proses informasi yang memiliki ciri fungsi tertentu yang dilakukan jaringan saraf biologis secara mendalam.

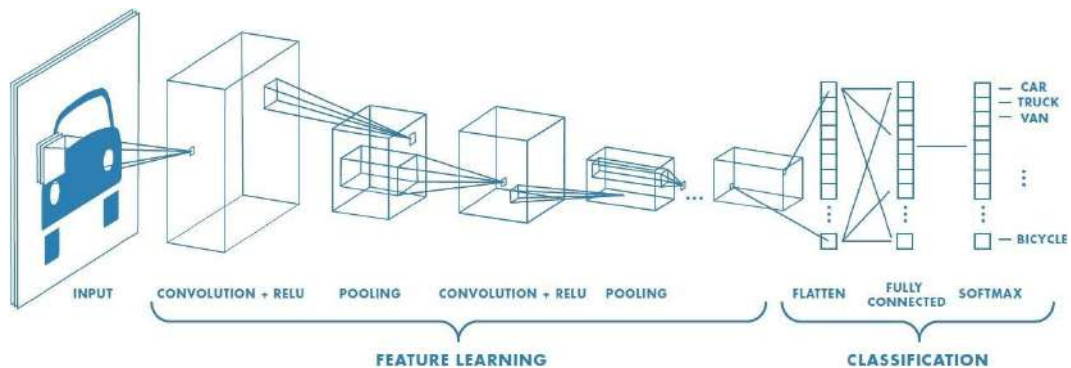
2) *Artificial Neural Network (ANN)*



Gambar 2. 4 Artificial Neural Network (ANN)

Para ilmuwan komputer menggunakan *Artificial Neural Network (ANN)* sebagai komponen penting dari pembelajaran mesin untuk melakukan tugas tugas kompleks seperti mengenali tren, membuat prediksi, dan menyusun strategi. Jaringan saraf belajar dari pengalaman, tidak seperti algoritma pembelajaran mesin lainnya yang dapat mengatur data atau angka crunch [15].

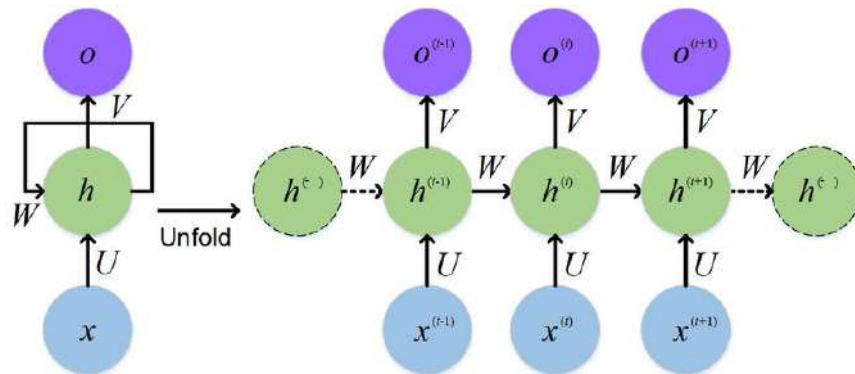
3) *Convolutional Neural Network (CNN)*



Gambar 2. 5 Convolutional Neural Network (CNN)

Salah satu teknik *deep learning*, Convolutional Neural Network (CNN), menggunakan jaringan syaraf tiruan untuk menyelesaikan masalah dengan mempelajari data sebelumnya. Metode Convolutional Neural Network (CNN) adalah yang terbaik untuk pengenalan citra digital karena didasarkan pada sistem pengenalan citra pada visual cortex manusia. Metode Convolutional Neural Network (CNN) adalah yang paling umum digunakan dalam pengolahan citra, dan lapisan konvolusi melakukan operasi konvolusi pada output lapisan sebelumnya. CNN adalah bagian dari Multilayer Perceptron (MLP). Istilah matematis yang disebut "konvolusi" mengacu pada penerapan fungsi pada output fungsi lain berulang kali. Layer penuh terhubung ini, yang biasanya digunakan dalam implementasi MLP, bertujuan untuk mengubah dimensi data sehingga data dapat diklasifikasikan secara linear. Setiap neuron yang berada di lapisan convolution harus diubah menjadi data satu dimensi sebelum dapat dimasukkan ke dalam lapisan penuh terhubung[16].

4) Recurrent Neural Network (RNN)



Gambar 2. 6 Recurrent Neural Network (RNN)

Jenis struktur jaringan saraf tiruan yang disebut RNN menggunakan pemrosesan berulang untuk memproses masukan, biasanya data sequential. Output yang dihasilkan dalam setiap pemrosesan dipengaruhi oleh state internal sampel sebelumnya—atau, lebih tepatnya, oleh bidirectional RNN—dan sampel itu sendiri[17].

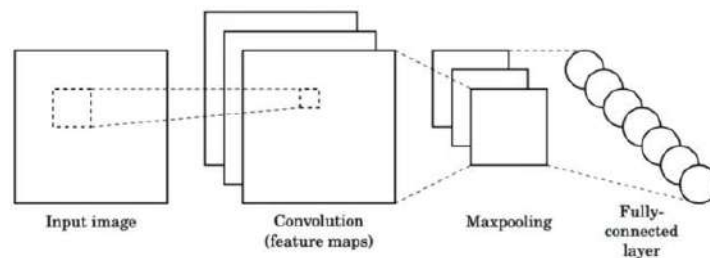
Secara umum, CNN lebih cocok untuk data spasial seperti gambar, ANN digunakan dalam berbagai aplikasi, DNN mengatasi masalah dengan data yang lebih kompleks, sedangkan RNN digunakan untuk data berurutan atau urutan. Pada penelitian kali ini, penulis menggunakan teknik CNN, metode ini menggunakan dataset dan memasukkan setiap komponen ke dalam sebuah gambar. Selanjutnya, proses memecah gambar dan mengambil setiap potongannya. Metode ini dapat digunakan secara real-time karena proses pengolahan gambarnya lebih cepat daripada metode lain.

4. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu jaringan saraf tiruan *deep feedforward* yang luas diterapkan pada visi komputer. Metode ini juga dikenal sebagai *ConvNet*. Mirip dengan jaringan syaraf tiruan, CNN memiliki arsitektur yang terdiri dari *node* atau *neuron* yang saling terhubung [16]. *Neuron* yang saling terhubung satu sama lain dalam sebuah lapisan. CNN juga termasuk dalam kategori *Deep Neural Network* karena

memiliki struktur yang dalam dan sering digunakan dalam pengolahan citra. Metode CNN terdiri dari dua tahap, yaitu klasifikasi yang menggunakan *feedforward* dan proses pembelajaran yang menggunakan algoritma *backpropagation*.

Selanjutnya dalam Metode *Convolutional Neural Network* (CNN) memiliki sebuah arsitektur. Arsitektur CNN adalah sebuah jaringan saraf yang menggunakan sebuah layer *konvolusi* dan *max pooling*. Seperti gambar dibawah ini



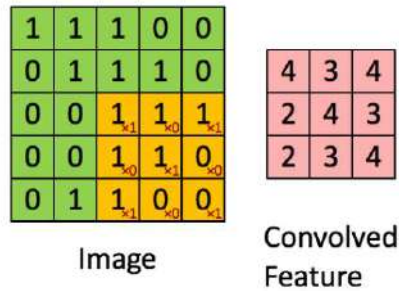
Gambar 2. 7 Convolutional Neural Network (CNN)

CNN umumnya terdiri dari beberapa lapisan, yang dapat dikelompokkan menjadi tiga jenis utama:

a. Lapisan Konvolusi (*Convolutional Layer*)

Convolutional layer merupakan lapisan pertama pada struktur CNN yang merupakan inti dari CNN dan tempat terjadinya sebagian besar proses komputasi. Pada lapisan ini dilakukan proses konvolusi dengan menggeser matriks filter ke seluruh permukaan citra[18].Lapisan ini adalah inti dari arsitektur CNN. Fungsi utamanya adalah untuk mengekstraksi fitur

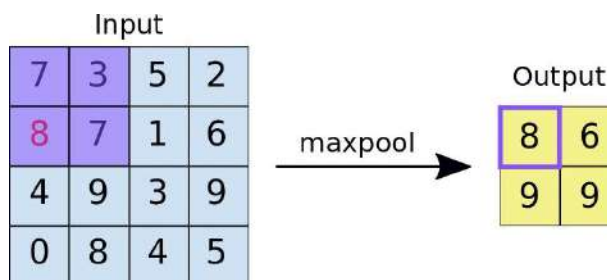
dari input data menggunakan filter (atau *kernel*) yang bergerak melintasi input. Setiap filter mendeteksi berbagai pola seperti tepi, tekstur, atau objek sederhana dalam gambar. Hasilnya adalah peta fitur (*feature map*) yang menunjukkan lokasi dan kekuatan fitur yang terdeteksi.



Gambar 2. 8 Convolutional Layer

b. Lapisan Pooling (*Pooling Layer*)

Setelah konvolusi, lapisan pooling digunakan untuk mengurangi dimensi spasial (panjang dan lebar) dari peta fitur sambil tetap mempertahankan informasi penting. *Pooling* bertujuan untuk memperkecil ukuran matrik agar perhitungan yang dilakukan mesin menjadi lebih cepat [19]. Ini dilakukan dengan cara mengambil nilai maksimum (*max pooling*) atau rata-rata (*average pooling*) dari patch kecil dalam peta fitur. Pooling membantu mengurangi jumlah parameter, mencegah *overfitting*, dan membuat jaringan lebih tahan terhadap perubahan kecil pada input.



Gambar 2. 9 Pooling Layer

c. Lapisan Terhubung Sepenuhnya (*Fully Connected Layer*)

Fully connected layer adalah lapisan dimana semua neuron aktif pada lapisan sebelumnya terhubung dengan neuron pada lapisan berikutnya, mirip dengan jaringan syaraf tiruan [20] . Di tahap akhir, output dari lapisan konvolusi dan pooling dilewatkan ke satu atau lebih lapisan terhubung sepenuhnya (*fully connected layer*). Setiap neuron di lapisan ini terhubung ke semua neuron di lapisan sebelumnya, mirip dengan jaringan saraf tradisional. Lapisan ini bertanggung jawab untuk menggabungkan fitur-fitur yang telah diekstraksi untuk membuat keputusan akhir, seperti klasifikasi objek dalam gambar.

5. Dataset

Dataset adalah sekumpulan data yang dapat digunakan sebagai bahan percobaan penelitian [21] . Dataset yang baik memiliki beberapa ciri penting diantaranya adalah :

- 1) Relevan : data harus sesuai dengan tujuan analisis atau penggunaan yang dimaksud.
- 2) Bersih : bebas dari nilai duplikat, nilai hilang, atau outlier yang dapat mengganggu analisis.
- 3) Akurat: data harus akurat dan tepat dalam konteksnya.
- 4) Komprehensif: dataset harus memiliki informasi yang memadai untuk menjawab pertanyaan.
- 5) Dapat dipercaya: sumber data harus berasal dari sumber yang dapat diandalkan dan didokumentasikan dengan baik.
- 6) Aktual: agar tetap relevan dan akurat, data harus diperbarui secara teratur.
- 7) Format yang sesuai: dataset harus disusun dalam format yang mudah dipahami dan digunakan.
- 8) Anonim: hapus atau enkripsi data pribadi dengan benar untuk melindungi privasi.
- 9) Legal: pastikan pengumpulan, penyimpanan, dan penggunaan data secara sah dan etis sesuai dengan kebijakan dan peraturan privasi yang berlaku.

6. Kaggle



Gambar 2. 10 Kaggle

Kaggle adalah platform online yang memungkinkan praktisi data science dan machine learning untuk menggunakan dataset, kompetisi data science, kursus, dan sumber daya[22].

7. FER-2013



Gambar 2. 11 FER-2013 Dataset

Facial Expression Recognition 2013 (FER-2013) adalah dataset citra wajah yang berukuran 48x48 berupa citra *grayscale* [23]. Dataset ini terbagi menjadi dua bagian, yaitu data untuk testing dan data training. Data testing sejumlah 7.178 dan data yang digunakan untuk training sejumlah 28.709. Dataset ini dibuat berdasarkan penelitian psikolog Amerika Paul Ekman pada tahun 1972. Paul Ekman membagi emosi fundamental manusia ke dalam enam kategori: senang, sedih, terkejut, marah, takut, dan jijik.

Tabel 2. 2 Jumlah Dataset FER-2013

Jenis Ekspresi	Jumlah Data Testing	Jumlah Data Training
Marah (Angry)	958	3995
Jijik (Disgust)	111	436
Takut (Fear)	1024	4097
Senang (Happy)	1774	7215
Netral (Neutral)	1233	4965
Sedih (Sad)	1247	4830
Terkejut (Surprise)	831	3171

8. *Recognition* (Teknik Pengenalan)

Teknik pengenalan pola, yang juga dikenal sebagai pengenalan, adalah cabang ilmu yang mengkhususkan diri dalam identifikasi dan pengelompokan objek berdasarkan fitur-fitur atau karakteristik tertentu yang dapat diukur secara kuantitatif. Proses ini melibatkan ekstraksi fitur dari data mentah, seperti citra, sinyal, atau teks, untuk kemudian dianalisis dan diklasifikasikan ke dalam kategori tertentu [24].

Dengan kemajuan teknologi dan peningkatan kapasitas komputasi, teknik pengenalan pola telah berkembang pesat dan menjadi semakin kompleks. Metode-metode modern seperti deep learning dan convolutional neural networks (CNN) telah membawa revolusi dalam bidang ini, memungkinkan pengenalan pola dengan tingkat akurasi yang tinggi bahkan dalam tugas yang sangat menantang.

9. *Face Recognition* (Pengenalan Wajah)

Secara umum, sistem pengenalan wajah terbagi menjadi dua jenis: sistem berbasis fitur dan sistem berbasis gambar. Pengenalan ini dapat dibagi menjadi

dua bagian, yaitu dikenali atau tidak dikenali, setelah melakukan perbandingan dengan pola yang sebelumnya disimpan di dalam database [24] . Pada bagian pertama menggunakan fitur yang memisahkan bagian wajah, seperti mata, hidung, dan mulut, dan kemudian menggambarkan hubungan antar fitur secara geometris, sistem kedua menggunakan data piksel asli gambar yang kemudian direpresentasikan dengan cara tertentu, yang kemudian digunakan untuk mengklasifikasikan identitas gambar.

10. Penyelarasan Wajah

Proses penyelarasan wajah bertujuan untuk menormalisasi wajah yang dihasilkan dari pendeteksian wajah [25] . Citra wajah yang diperkirakan secara kasar atau memiliki kualitas yang kurang baik, seperti ukuran yang berbeda dari standar, faktor pencahayaan yang kurang atau lebih, atau kejelasan yang kurang akan diproses melalui beberapa tahapan berikut.

1) Grayscale Citra

Pada tahap pertama proses penyelarasan, citra warna warna RGB diubah menjadi citra berwarna abu. Citra warna RGB terdiri dari tiga parameter, yaitu merah, hijau, dan biru. Jika citra warna RGB dimasukkan ke dalam proses ekstraksi, akan sulit untuk melakukannya karena citra RGB terdiri dari tiga parameter, sehingga diperlukan penyamaan parameter.

2) Pemotongan

Pada tahap ini, gambar wajah dipisahkan dari gambar masukan. Tujuan dari pemotongan gambar ini adalah untuk mengambil gambar yang diperlukan untuk proses ekstraksi, yaitu gambar wajah, dan menghilangkan gambar lain yang tidak diperlukan. Disesuaikan dengan dimensi proses pengkotakan atau segmentasi objek wajah yang dilakukan pada pendeteksian wajah, dimensi gambar yang dipotong disesuaikan.

3) Resizing

Tahap *resizing* gambar melibatkan normalisasi dimensi gambar wajah, yang berarti membesarkan atau mengecil gambar wajah menjadi dimensi yang telah ditentukan. Tujuan dari normalisasi ini adalah untuk memastikan bahwa dimensi

masing-masing gambar yang dimasukkan sama, sehingga matriks data gambar wajah tidak berubah dalam proses ekstraksi gambar berikutnya.

4) Equalizing

Tahap terakhir dari proses penyelarasan yang tujuannya adalah untuk mengklarifikasi nilai histogram dari gambar wajah yang dihasilkan dari tahapan sebelumnya.

11. Ekspresi Wajah

Ekstraksi fitur adalah metode untuk mengumpulkan karakteristik unik yang membedakan sampel wajah dari sampel wajah lainnya. Ekstraksi fitur juga dapat digunakan untuk mengumpulkan informasi yang bermanfaat untuk membedakan wajah orang yang berbeda dari gambar wajah yang telah diselaraskan [26]. Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), dan Independent Component Analysis (ICA) adalah beberapa algoritma ekstraksi yang dapat digunakan untuk melakukan proses ini. Vektor fitur, bentuk dasar pencarian gambar berbasis konten, menangkap fitur gambar seperti warna dan tekstur.

12. Pencocokan Wajah

Pencocokan fitur adalah proses penting dalam pengenalan pola dan analisis citra, di mana fitur-fitur yang diekstraksi dari gambar uji dibandingkan dengan fitur-fitur dari gambar-gambar dalam database yang telah melalui proses pelatihan sebelumnya [27]. Proses ini bertujuan untuk mengidentifikasi kesamaan atau perbedaan antara gambar uji dan gambar referensi, memungkinkan sistem untuk mengenali objek atau pola tertentu dalam gambar uji.

13. OpenCV



Gambar 2. 12 OpenCV

OpenCV adalah *library open source* yang dikembangkan oleh Intel yang bertujuan untuk mempermudah programing yang berkaitan dengan gambar digital. OpenCV sudah memiliki banyak fitur, seperti pengenalan wajah, pelacakan wajah, deteksi wajah, kalman filtering, dan berbagai metode AI (*Artificial Intelligence*)[28].

Selain itu, OpenCV adalah kumpulan algoritma terkait penglihatan komputer yang sederhana untuk penggunaan tingkat rendah, dan tersedia untuk bahasa pemrograman C/C++, Python, Java, dan Matlab. OpenCV pertama kali dirilis secara resmi oleh *Inter Research* pada tahun 1999 sebagai bagian dari proyek yang melibatkan aplikasi intensif berbasis CPU, ray tracing real-time, dan tembok penampil 3D. Para kontributor utama proyek ini termasuk mereka yang bekerja di bidang optimasi Intel di Russia, serta tim pusat performa Intel. OpenCV memiliki banyak fitur yang dapat digunakan; berikut adalah beberapa fitur utamanya:

1) Image and video I/O

antar muka ini memungkinkan kita untuk membuat file gambar dan video serta membaca data gambar dari file atau umpan video langsung.

2) Computer vision secara umum

kita dapat melakukan eksperimen uji coba dengan berbagai standar algoritma computer vision. Termasuk juga deteksi garis, tepi, pucuk, proyeksi elips, image pyramid untuk pemrosesan gambar multi skala, pencocokan template, dan berbagai transform (Fourier, cosine diskrit, distance transform) dan lain - lain.

3) Modul *computer vision high level*

4) Metode untuk AI dan *machine learning*

aplikasi *computer vision* sering kali memerlukan machine learning atau metode AI lainnya, beberapa metode tersebut tersedia dalam paket OpenCV *machine learning*.

14. Tensorflow



Gambar 2. 13 Tensorflow

Tensorflow, yang dikembangkan oleh tim Google Brain, adalah library open source yang berfokus pada pengajaran machine learning dan komputasi numerik skala besar yang menggunakan berbagai model, algoritma machine learning, dan deep learning (jaringan syaraf) [29]. Dengan menggunakan Python, Tensorflow dapat membangun API front-end dan menjalankan aplikasi dengan kecepatan tinggi. Fiturnya termasuk melatih dan menjalankan jaringan syaraf tiruan untuk klasifikasi tulisan tangan, pengenalan gambar, penyematan kata, jaringan syaraf ulang, model sequence-to-sequence untuk terjemahan, pemrosesan bahasa alami, dan simulasi berbasis PDE (Partial Defect Detection).

15. Cybersecurity

Cybersecurity adalah sistem yang dirancang untuk melindungi kerahasiaan, integritas, dan ketersediaan informasi dari serangan yang terjadi di Internet, yang sering disebut sebagai *cyberattack* [30]. Beberapa teknik perlindungan yang digunakan untuk melindungi akses kamera diantaranya adalah :

1) Pengaturan dan Perangkat Lunak

- a) Update Perangkat Lunak: Selalu update firmware kamera dan perangkat lunak terkait untuk menutup kerentanan yang diketahui.

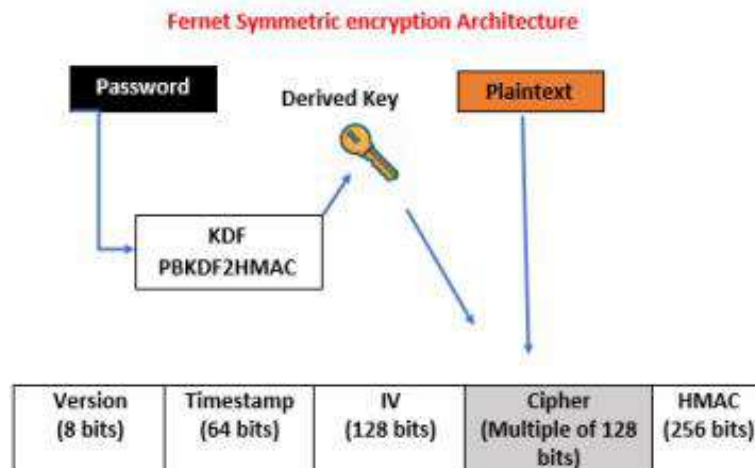
- b) Konfigurasi Keamanan: Pastikan pengaturan keamanan kamera diatur dengan benar, termasuk mengubah kata sandi default.
 - c) Aplikasi Kamera: Gunakan aplikasi kamera resmi dari vendor atau sumber tepercaya.
- 2) Akses dan Autentikasi
- a) Kata Sandi Kuat: Gunakan kata sandi yang kuat dan unik untuk akses kamera. Hindari menggunakan kata sandi default atau yang mudah ditebak.
 - b) Multi-Factor Authentication (MFA): Aktifkan MFA jika tersedia untuk menambah lapisan keamanan.
- 3) Kontrol Akses
- a) Akses Terbatas: Batasi akses ke kamera hanya untuk pengguna yang berwenang.
 - b) Segmentasi Jaringan: Letakkan kamera pada segmen jaringan yang terpisah untuk membatasi akses dari bagian lain jaringan.
- 4) Enkripsi
- a) Enkripsi Data: Pastikan transmisi data dari dan ke kamera dienkripsi untuk mencegah penyadapan.
 - b) VPN: Gunakan VPN untuk mengakses kamera dari jarak jauh untuk memastikan koneksi yang aman.
- 5) Monitoring dan Deteksi
- a) Log Aktivitas: Monitor dan audit log aktivitas untuk mendeteksi akses yang tidak sah atau mencurigakan.
 - b) Intrusion Detection Systems (IDS): Gunakan IDS untuk memonitor jaringan dan mendeteksi aktivitas mencurigakan.
- 6) Perangkat Fisik

- a) Penutup Kamera: Gunakan penutup fisik untuk kamera laptop atau webcam saat tidak digunakan.
 - b) Lokasi Kamera: Letakkan kamera di lokasi yang sulit diakses secara fisik oleh pihak yang tidak berwenang.
- 7) Firewall dan Keamanan Jaringan
- a) Firewall: Konfigurasi firewall untuk membatasi lalu lintas yang dapat mengakses kamera.
 - b) Port Forwarding: Hindari menggunakan port forwarding secara langsung untuk mengakses kamera dari internet. Sebagai gantinya, gunakan layanan cloud yang aman atau VPN.
- 8) Software Keamanan Tambahan
- a) Antivirus dan Anti-Malware: Instal dan update perangkat lunak antivirus dan anti-malware untuk mencegah perangkat dari infeksi yang dapat mengakses kamera.
 - b) Endpoint Protection: Gunakan solusi endpoint protection yang menyediakan fitur khusus untuk mengamankan akses kamera.
- 9) Kebijakan dan Pelatihan
- a) Pelatihan Keamanan: Berikan pelatihan keamanan kepada pengguna mengenai risiko dan praktik terbaik untuk melindungi akses kamera.
 - b) Kebijakan Keamanan: Tetapkan kebijakan keamanan yang jelas mengenai penggunaan dan akses ke kamera.

16. *Fernet Symmetric Encryption*

Enkripsi dengan menggunakan metode Fernet merupakan pendekatan kriptografi simetris yang memanfaatkan kombinasi algoritma yang teruji untuk menjamin kerahasiaan dan integritas data. Fernet mengimplementasikan AES (Advanced Encryption Standard) dengan panjang kunci 128 bit dalam mode Cipher Block Chaining (CBC), di mana setiap blok

plaintext dienkripsi menggunakan kunci yang sama dan hasil enkripsi dari blok sebelumnya, untuk meningkatkan ketahanan terhadap serangan analisis [31].



Gambar 2. 14 Fernet Symmetric Encryption

Dalam konteks enkripsi gambar, metode Fernet diterapkan dengan mengkonversi gambar menjadi aliran byte terlebih dahulu, sehingga dapat diperlakukan sebagai data biner siap enkripsi. Data biner ini kemudian dienkripsi menggunakan kunci simetris dengan algoritma AES-128 dalam mode CBC, dengan penambahan padding PKCS7 untuk memastikan panjang data sesuai ukuran blok yang diperlukan. Selama proses enkripsi, kode autentikasi pesan (MAC) dihasilkan menggunakan HMAC-SHA-256 untuk memastikan integritas data selama transmisi atau penyimpanan. Gambar terenkripsi kemudian dapat disimpan atau ditransmisikan dengan aman, hanya dapat diakses oleh pihak yang memiliki kunci enkripsi yang benar. Untuk mengakses gambar asli, data biner terenkripsi harus didekripsi menggunakan kunci yang sama, melibatkan pembalikan operasi enkripsi dan verifikasi HMAC untuk menjaga keutuhan data.

17. Metode Prototype

Penelitian ini menggunakan metode prototype yang melibatkan beberapa tahapan utama. Metode *Prototype* merupakan metode pengembangan

perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna. Model *prototype* yang digunakan oleh sistem akan mengizinkan pengguna mengetahui seperti apa tahapan sistem yang dibuat sehingga sistem mampu beroperasi secara baik [32]. Metode *prototype* yang diterapkan pada penelitian ini dimaksudkan agar mendapatkan representasi dari pemodelan aplikasi yang akan dibuat, berikut adalah tahapan dalam metode *prototype*:

a. Pengumpulan Kebutuhan:

Pada tahapan ini, informasi dikumpulkan dari pengguna untuk memahami kebutuhan dan tujuan perangkat lunak yang akan dibangun.

b. Membangun Prototype:

Berdasarkan kebutuhan yang telah dikumpulkan, pengembang membuat prototipe awal. Prototipe ini adalah representasi sederhana dari sistem yang diinginkan dan berfokus pada fungsi-fungsi utama. Tujuan utama dari prototipe ini adalah untuk menunjukkan konsep dan mendapatkan umpan balik awal dari pengguna.

c. Evaluasi Prototype

Prototype yang telah dibuat dievaluasi oleh pengguna, untuk menguji dan memberikan umpan balik mengenai fitur, antarmuka, dan seluruh pengalaman penggunaan prototipe tersebut

d. Mengkodekan Sistem:

Setelah mendapatkan umpan balik dari evaluasi prototipe, pengembang mulai menulis kode untuk sistem yang sebenarnya.

e. Menguji Sistem

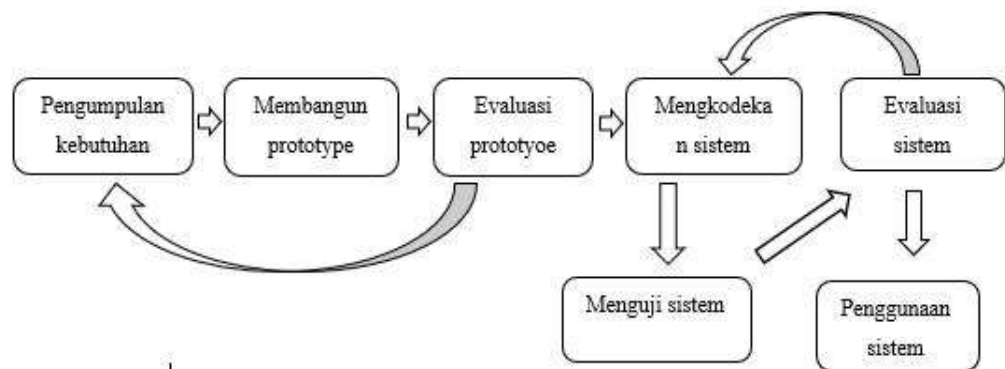
Sistem yang telah dikodekan diuji untuk memastikan bahwa semua fungsionalitas bekerja sebagaimana mestinya dan tidak ada bug atau kesalahan.

f. Evaluasi Sistem

Setelah pengujian, sistem dievaluasi secara keseluruhan untuk memastikan bahwa memenuhi kebutuhan pengguna dan bekerja sesuai dengan spesifikasi yang telah ditetapkan.

g. Penggunaan Sistem:

Setelah sistem lolos evaluasi, sistem siap digunakan oleh pengguna akhir. Tahap ini mencakup pelatihan pengguna, instalasi, dan dukungan teknis awal untuk memastikan bahwa pengguna dapat menggunakan sistem dengan efektif.



Gambar 2. 15 Metode Prototype

Metode prototype memungkinkan penelitian untuk membuat sistem yang sesuai dengan kebutuhan dan keinginan pengguna, mengurangi risiko kesalahan dan penolakan pengguna terhadap sistem baru. Metode ini juga memungkinkan penyesuaian dan perbaikan yang bergantung pada masukan langsung pengguna dilakukan secara lebih cepat dan efisien.

18. UML

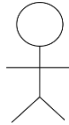

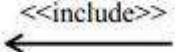
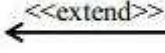

Unified Modeling Language (UML) merupakan satu set standar teknik diagram yang memberikan representasi grafis yang cukup kaya untuk model setiap pengembangan sistem proyek dari analisis melalui implementasi [33]. UML menyediakan notasi grafis untuk menggambarkan berbagai aspek sistem, seperti struktur, perilaku, dan interaksi antar komponen.

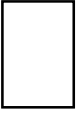


Beberapa komponen yang terdapat pada UML adalah sebagai berikut:

a. Use Case Diagram

Use case diagram merupakan deskripsi fungsi dari sebuah sistem dari perspektif atau sudut pandang para pengguna system. *Use case diagram* menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk actor. Sebuah use case digambarkan sebagai elips horizontal dalam suatu diagram *UML use case*[34].

Tabel 2. 3 Use Case Diagram

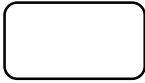


NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Mewakili peran orang, system yang lain ,atau alat berkomunikasi dengan <i>use case</i>
2		<i>Generalization</i>	Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan <i>use case</i> .
3		<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya suatu fungsionalitas dari <i>use case</i> lainnya
4		<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas dari <i>use case</i> lainnya jika suatu kondisi terpenuhi
5		<i>Association</i>	Abstraksi dari penghubung antara aktor dengan use case.

6		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
7		<i>Use Case</i>	Abstraksi dan interaksi antara sistem dan aktor.
8		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.


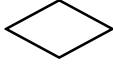
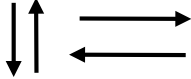
b. Activity Diagram

Activity diagram adalah suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu *system* [34] . *Diagram* ini menunjukkan serangkaian aktivitas atau tindakan yang terjadi dalam suatu proses, serta urutan dan kondisi di mana aktivitas tersebut dilakukan

Tabel 2. 4 Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.

19. Pengujian Black Box

4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan diakhiri.
5		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
6		<i>Line Conector</i>	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

black box menguji perangkat lunak tanpa memperhatikan bagian dalam program. Metode ini terutama berkonsentrasi pada nilai-nilai masukan yang diberikan kepada program, tanpa mempertimbangkan bagaimana program menghasilkan keluaran atau output [35].

Salah satu keuntungan utama dari metode black box adalah bahwa pengujian dapat dilakukan tanpa memerlukan pengetahuan mendalam tentang bagaimana program bekerja. Metode ini menggabungkan pengujian dari sudut pandang pengguna akhir, sehingga pengujian dan programmer bergantung satu sama lain. Metode black box berfokus pada spesifikasi fungsional perangkat lunak, bukan alternatif untuk teknik program sistem informasi yang diuji. Partisi setara, yang membagi nilai masukan menjadi kelompok yang sama untuk pengujian, adalah salah satu metode yang digunakan dalam metode black box.

20. Pengujian White Box

Pengujian White Box, atau pengujian kotak kaca, adalah metode pengujian perangkat lunak yang berfokus pada struktur internal dan kode program [36]. Pengujian ini berfokus pada bagaimana program tersebut bekerja secara internal, dengan menganalisis kode program dan melacak aliran data dan control, Karakteristik dari penggunaan white box adalah sebagai berikut:

- a. Fokus pada struktur internal: Pengujian ini memastikan apakah kode program ditulis dengan benar dan sesuai dengan spesifikasi.
- b. Membutuhkan pengetahuan kode: Penguji perlu memiliki pengetahuan tentang kode program untuk melakukan pengujian *white box*.
- c. Dapat menemukan error yang tidak terdeteksi oleh pengujian *black box*: Pengujian ini dapat menemukan error yang terkait dengan struktur internal program, seperti *loop* tak terbatas dan kondisi yang tidak terduga.
- d. Lebih kompleks dan memakan waktu: Pengujian ini lebih kompleks dan memakan waktu dibandingkan dengan pengujian *black box*.

21. *System Usability Scale (SUS)*

Sistem Usability Scale (SUS) adalah kuesioner yang terdiri dari sepuluh pertanyaan yang dirancang untuk menilai kemudahan penggunaan sistem komputer dari perspektif pengguna (Brooke, 1996). Perhitungan SUS dilakukan menggunakan lima skala Likert. Responden diminta memberikan sepuluh pernyataan SUS berdasarkan penilaian subjektifnya. Menurut Brooke (2013), kuesioner SUS dapat digunakan untuk mengetahui seberapa puas pengguna dengan produk [37]. Setiap item pernyataan memiliki skor kontribusi, yang berkisar antara 0 dan 4. Skor kontribusi untuk item 1,3,5,7 dan 9 berkisar antara 1 dan 4, sedangkan skor kontribusi untuk item 2,4,6,8 dan 10 berkisar antara 5 dan dikurangi posisi skala. Kalikan kemudian total skor SUS dari 0 hingga 100. Komponen pertanyaan SUS disajikan dalam Tabel .

Tabel 2. 5 Komponen Pertanyaan System Usability Scale

No.	Komponen
1.	Saya berpikir akan menggunakan sistem ini lagi.
2.	Saya merasa sistem ini rumit untuk digunakan.
3.	Saya merasa sistem ini mudah digunakan.
4.	Saya membutuhkan bantuan dari orang lain atau teknisi

	dalam menggunakan sistem ini.
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6.	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
8.	Saya merasa sistem ini membingungkan.
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Pada Tabel 2.6, merupakan Skor SUS, hasil dari pengolahan komponen pertanyaan SUS dari responden dapat dihitung untuk nilai rata-rata dari skor SUS berikut

Tabel 2. 6 Kategori Penilaian SUS

>81	A	Excellent
68-81	B	Good
68	C	OK/Fair
51-67	D	Poor
<51	E	Worst

Metode Usability System Usability Scale (SUS) menggunakan pengukuran usability yang "quick and right" (cepat dan tepat) selama pemrosesan

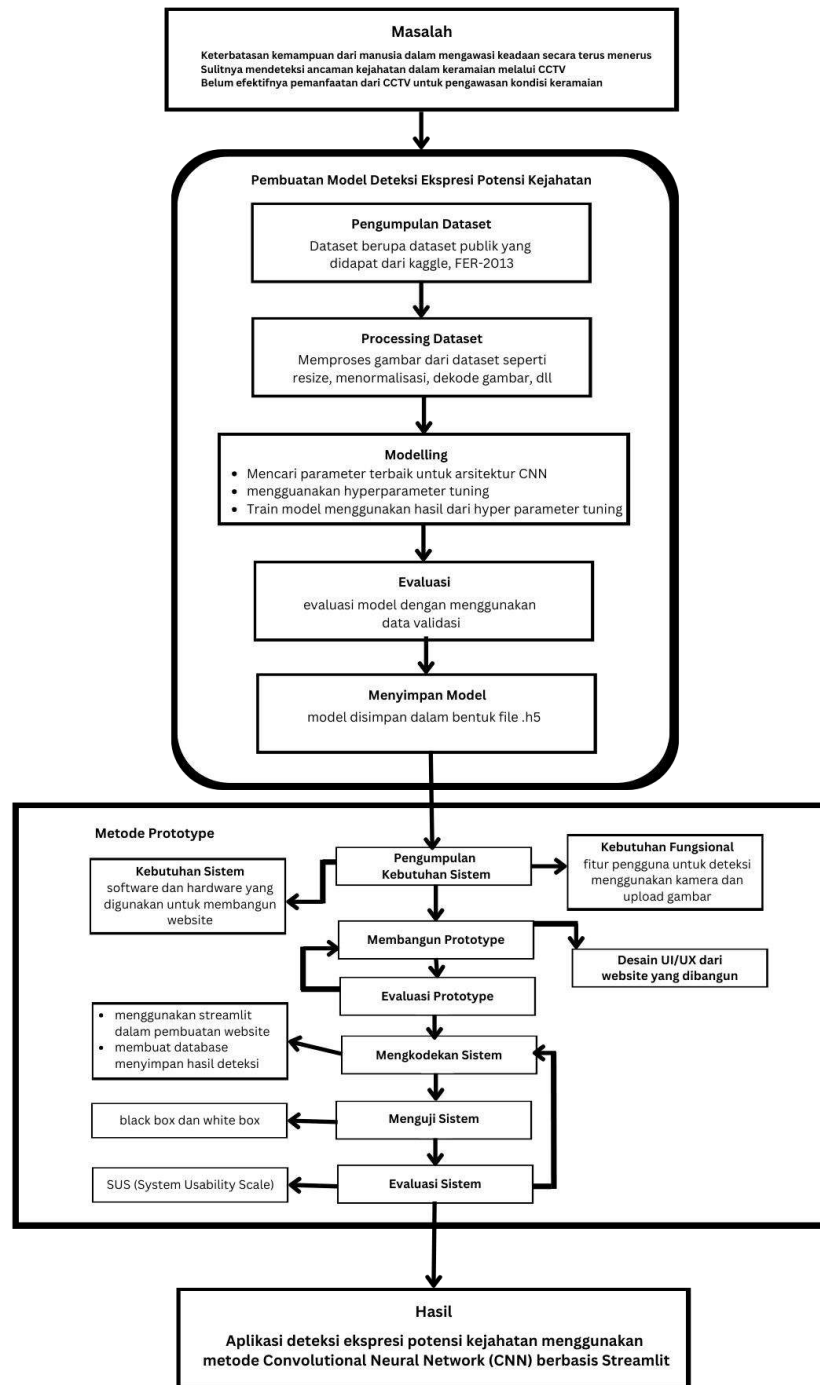
kuesioner SUS, yang menghasilkan skor yang tampak mudah dipahami dari 0 hingga 100. Skor yang lebih tinggi menunjukkan kualitas usability.

C. Kerangka Berfikir

Peningkatan kasus kejahatan dalam masyarakat telah menjadi perhatian utama bagi pihak berwenang dan masyarakat umum. Kejahatan seperti pencurian, perampokan, dan tindak kekerasan seringkali terjadi tanpa terdeteksi secara cepat dan akurat. Hal ini menimbulkan ketidaknyamanan dan mengancam keselamatan masyarakat. Oleh karena itu, diperlukan Solusi yang efektif untuk mendeteksi potensi kejahatan secara dini dan mengambil tindakan pencegahan yang tepat.

Permasalahan utama yang ingin dipecahkan adalah kurangnya sistem yang mampu mendeteksi potensi kejahatan secara cepat dan akurat menjadi salah satu penyebab tidak efektifnya tindak pencegahan yang dilakukan. Kelemahan dalam beberapa sistem yang ada menyebabkan petugas keamanan kesulitan untuk mengidentifikasi tindakan kriminal secara tepat waktu, yang pada akhirnya dapat meningkatkan tingkat kriminalitas dan mengancam keselamatan masyarakat.

Dalam pengolahan citra, metode Convolutional Neural Network (CNN) adalah yang paling populer. CNN adalah bagian dari Multilayer Perceptron (MLP), dan lapisan konvolusi melakukan operasi konvolusi pada output lapisan sebelumnya. Penerapan fungsi pada output fungsi lain berulang kali disebut dengan istilah matematis "konvolusi". Tujuan dari lapisan penuh terhubung ini, yang biasanya digunakan dalam implementasi MLP, adalah untuk mengubah dimensi data sehingga data dapat diklasifikasikan secara linear. Sebelum lapisan convolution dapat terhubung secara keseluruhan, setiap neuron di dalamnya harus diubah menjadi data satu dimensi.



Gambar 2. 16 Kerangka Berfikir

D. Hipotesis Penelitian

Hipotesis adalah pernyataan tentang suatu ide yang harus diuji kebenarannya. Hipotesis deskriptif adalah hipotesis yang tidak membandingkan dan tidak

menghubungkan dengan variabel lain[38]. Hipotesis ini juga dapat digunakan untuk menjelaskan suatu peristiwa atau untuk menjawab masalah taksiran. Hipotesis pada penelitian ini adalah :

Hi : Implementasi CNN dan Enkripsi pada Sistem deteksi ekspresi dapat berfungsi dengan baik dan aman dalam membatu memonitoring kegiatan publik.

Ho : Implementasi CNN dan Enkripsi pada Sistem deteksi ekspresi belum berfungsi dengan baik dan kurang efisien dalam membatu memonitoring kegiatan publik.

BAB III

METODE PENELITIAN

A. Pendekatan Penelitian

Jenis penelitian ini adalah penelitian kuantitatif. Penelitian kuantitatif dalam konteks ini dilakukan dengan mengumpulkan data berupa gambar dan video dari rekaman CCTV. Data yang berupa citra visual tersebut kemudian diolah menggunakan algoritma Convolutional Neural Network (CNN) untuk mendeteksi ekspresi wajah yang mencurigakan. Hasil dari pemrosesan ini dianalisis untuk mendapatkan informasi ilmiah mengenai efektivitas model CNN dalam mendeteksi potensi kejahatan.

Penelitian ini juga akan menggunakan metrik kuantitatif seperti akurasi, presisi, recall, dan F1-score untuk mengevaluasi performa model yang dikembangkan. Dengan demikian, penelitian ini bertujuan untuk menghasilkan bukti empiris yang dapat mendukung peningkatan sistem keamanan publik melalui teknologi pengenalan ekspresi wajah berbasis deep learning.

B. Fokus Penelitian

Penelitian ini difokuskan pada pengembangan dan implementasi algoritma *Convolutional Neural Network (CNN)* untuk mendeteksi ekspresi wajah yang mencurigakan dalam rangka meningkatkan keamanan publik, Fokus utama penelitian ini mencakup beberapa aspek penting :

1. Rancangan Sistem Prediksi Ekspresi :

- a. Mengembangkan model CNN yang mampu mengidentifikasi ekspresi wajah dasar (marah, takut, Bahagia, sedih, terkejut, jijik, dan netral) dari rekaman CCTV.
- b. Mensimulasikan apakah sistem deteksi ekspresi ini dapat membantu petugas keamanan dalam memonitor dan mencegah tindak kejahatan dengan lebih efektif.

2. Evaluasi Model Deep Learning :

- a. Menilai keefektifan model CNN yang digunakan, termasuk validitas dan reliabilitas hasil prediksi ekspresi wajah.
- b. Menggunakan dataset yang representatif untuk melatih dan menguji model, serta menerapkan teknik validasi silang untuk memastikan model tidak overfitting.

3. Penerapan Cybersecurity Pada CCTV

Penelitian ini menerapkan enkripsi dan dekripsi pada dataset yang digunakan selama proses dari sistem deteksi ekspresi berbasis CNN.

C. Populasi Dan Sampel

Populasi dalam sebuah penelitian merujuk pada keseluruhan komponen yang menjadi objek kajian. Ini mencakup seluruh subjek dan objek yang memiliki karakteristik tertentu yang relevan dengan tujuan penelitian. Oleh karena itu, memahami populasi secara mendalam sangat penting untuk memastikan hasil penelitian yang valid dan dapat diterapkan secara lebih luas. Pemilihan sampel dari populasi ini kemudian dilakukan dengan metode tertentu untuk mendapatkan representasi yang akurat dari keseluruhan populasi.

1. Populasi

Populasi dapat didefinisikan sebagai keseluruhan komponen penelitian, termasuk objek dan subjek yang memiliki karakteristik tertentu. Oleh karena itu, populasi dapat mencakup semua anggota kelompok, apakah itu manusia, binatang, atau peristiwa, atau benda yang tinggal bersama dalam suatu tempat secara terencana disebut sebagai terikat hasil penelitian. Populasi bukan hanya individu tetapi juga organisasi, seperti guru, siswa, kurikulum, fasilitas, sekolah, hubungan sekolah dan masyarakat, karyawan perusahaan, varietas tanaman hutan, padi, kegiatan marketing, hasil produksi, dan sebagainya [39].

Populasi pada penelitian ini mencakup populasi ekspresi, Sebagai bagian dari penelitian, sistem CNN dapat mengidentifikasi semua jenis ekspresi wajah. Untuk menentukan apakah seseorang mencurigakan atau

tidak, rekaman webcam dapat menunjukkan ekspresi wajah yang berbeda, seperti senang, sedih, marah, terkejut, takut, atau netral.

2. Sampel

Sampel adalah bagian dari populasi yang berfungsi sebagai sumber data penelitian. Dengan kata lain, sampel adalah sebagian dari populasi untuk menggambarkan seluruh populasi [39]. Dalam penelitian ini, sampel dapat mencakup sampel ekspresi, data testing berupa gambar sejumlah 7.178 dan data gambar yang digunakan untuk training sejumlah 28.709.

D. Variabel Penelitian

1. Variabel Bebas

Variabel bebas, juga disebut sebagai variabel independen, adalah variabel yang berfungsi sebagai penyebab atau memiliki kemungkinan teoritis untuk berdampak pada variabel lain. Variabel bebas biasanya ditunjuk dengan huruf X, dan karena itu, ketika variabel bebas ditinjau, biasanya variabel pertama muncul (ada) dan kemudian diikuti oleh variabel lainnya. Peneliti tidak boleh secara sembarangan menentukan variabel bebas dalam proses ilmiah. Variabel bebas bukanlah kondisi yang tidak terpengaruh oleh variabel terikat [39].

Variabel bebas pada penelitian ini mencakup :

1) Ekspresi Wajah

Jenis ekspresi yang diidentifikasi oleh sistem (senang, sedih, marah, terkejut, takut, netral, jijik).

2) Jumlah Data Training

Jumlah gambar yang digunakan untuk melatih model CNN.

2. Variabel Terikat

Variabel terikat, yang besarnya bergantung pada besaran variabel independen, memungkinkan perubahan variabel dependen (terikat) sebesar koefisien perubahan yang disebabkan oleh variabel independen [39].

Variabel terikat pada penelitian ini mencakup :

1) Akurasi Deteksi Ekspresi Wajah

Persentase keberhasilan model CNN dalam mengidentifikasi ekspresi wajah yang berbeda.

2) Tingkat Keamanan Data

Efektivitas enkripsi data dalam melindungi privasi (misalnya, seberapa sulit data dapat didekripsi tanpa kunci yang benar).

3) Kualitas Enkripsi

Kualitas enkripsi yang diterapkan pada data, seperti panjang kunci yang diterapkan

3. Variabel Kontrol

Menurut Sugiyono, variabel kontrol dikendalikan atau dibuat konstan sehingga pengaruh variabel independen terhadap dependen tidak dipengaruhi oleh faktor luar yang tidak diteliti [39].

Dalam penelitian ini, variabel kontrol mencakup :

1) Resolusi Gambar

Resolusi gambar yang digunakan dalam semua rekaman 720p

2) Jenis Kamera

Menggunakan tipe kamera yang sama, webcam Logitech C310

E. Teknik Pengumpulan Data

Menggunakan dataset citra wajah dari sumber-sumber publik yang sudah tersedia dan terstandarisasi yaitu FER-2013. Dataset ini berisi berbagai ekspresi wajah yang dikategorikan ke dalam enam ekspresi dasar manusia (marah, takut, bahagia, sedih, terkejut, dan jijik).

F. Teknik Analisis Data

Teknik analisis data dalam penelitian ini melibatkan beberapa langkah penting sebagai berikut:

1. Preprocessing Data

- 1) Normalisasi dan Augmentasi: Data citra wajah yang diperoleh melalui dataset publik akan dinormalisasi dan di-augmentasi untuk memastikan bahwa model dapat belajar dari berbagai variasi data. Teknik

augmentasi yang digunakan meliputi rotasi, flipping, zooming, dan perubahan pencahayaan.

- 2) Pembagian Data: Data akan dibagi menjadi data latih (training set) dan data uji (testing set) untuk memastikan model dilatih dan diuji secara efektif.

2. Pelatihan Model (Training)

- 1) Model Convolutional Neural Network (CNN): Model CNN akan dilatih menggunakan data latih yang telah dipreproses. Selama pelatihan, model akan mempelajari pola-pola pada citra wajah untuk mengklasifikasikan ekspresi wajah dengan benar.
- 2) Hyperparameter Tuning: Melakukan tuning hyperparameter untuk menemukan kombinasi parameter yang optimal, seperti jumlah lapisan (layers), ukuran kernel, learning rate, dan epoch.

3. Validasi dan Evaluasi Model

- 1) Validasi Silang (Cross-Validation): Menggunakan teknik validasi silang untuk menguji kinerja model pada data validasi dan memastikan bahwa model tidak mengalami overfitting.
- 2) Metrik Evaluasi: Menggunakan metrik evaluasi seperti akurasi, presisi, recall, F1-score, dan AUC-ROC untuk mengevaluasi performa model CNN pada data uji. Metrik-metrik ini akan memberikan gambaran tentang seberapa baik model dalam mendeteksi ekspresi wajah yang mencurigakan.

4. Analisis Hasil

- 1) Confusion Matrix: Menyusun confusion matrix untuk menganalisis hasil prediksi model dan melihat distribusi prediksi benar dan salah untuk masing-masing kategori ekspresi wajah.
- 2) Visualisasi Hasil: Membuat visualisasi hasil pelatihan dan pengujian model, seperti grafik loss dan akurasi, untuk memahami perkembangan kinerja model selama proses pelatihan.

5. Implementasi Cybersecurity:

- 1) Evaluasi Keamanan: Menggunakan metode analisis risiko untuk mengevaluasi keamanan sistem CCTV terhadap ancaman siber. Menerapkan standar keamanan yang ada dan mengevaluasi efektivitasnya dalam melindungi sistem dari potensi serangan.
- 2) Uji Penetrasi (Penetration Testing): Melakukan uji penetrasi pada sistem untuk mengidentifikasi kelemahan dan memastikan bahwa sistem aman dari ancaman siber.

BAB IV

HASIL PENGEMBANGAN DAN PEMBAHASAN

Pada bab ini akan membahas mengenai hasil dan pembahasan dari penelitian yang telah dilakukan oleh penulis yaitu aplikasi deteksi ekspresi potensi kejahatan

A. Hasil

1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah langkah pertama dalam pengembangan suatu sistem atau proyek.

a. Kebutuhan Perangkat Keras:

Berikut ini adalah spesifikasi perangkat keras minimum yang mendukung untuk pembuatan aplikasi deteksi ekspresi potensi kejahatan:

1. Perangkat Keras Aplikasi Backend:

Processor : Intel® Core™ i7-10750H CPU @ 2.60 GHz (12 CPUs), ~ 2.6GHz
RAM : 16GB
SSD : 256 GB
VGA : NVIDIA GeForce GTX 1650Ti

2. Perangkat Keras Aplikasi Frontend:

OS : Minimum Windows 64 bit
RAM : Minimum 4GB
Hardisk : Minimum 4GB yang tersedia
Resolusi : 1280 x 800

b. Kebutuhan Perangkat lunak:

Berikut ini adalah spesifikasi minimum yang digunakan untuk pembuatan aplikasi deteksi ekspresi potensi kejahatan :

1. Jupyter Notebook
2. Pycharm
3. Figma

4. Kaggle
- c. Kebutuhan Fungsional
 1. Pada halaman detection, aplikasi bisa digunakan untuk mendeteksi ekspresi dengan menggunakan upload gambar
 2. Pada halaman detection, aplikasi bisa digunakan untuk mendeteksi ekspresi dengan menggunakan kamera
 3. Tombol start dan stop bisa berfungsi untuk mengoperasikan kamera

2. Implementasi Model Deteksi Potensi Kejahatan

a. Pengumpulan Dataset

Data yang digunakan untuk membuat model adalah dataset yang berupa gambar dari 7 ekspresi dasar manusia, data ini diperoleh dari *website* Kaggle <https://www.kaggle.com/datasets/msambare/fer2013/data>.

Dataset dibagi menjadi dua folder yaitu training data folder dan testing data folder. Perbandingan antara data train dan data test adalah 3:1.

1. Data Train

Training data folder digunakan untuk menyimpan dataset yang nantinya akan digunakan untuk melatih model. Pada training data ini terdapat 7 class yaitu *angry*, *disgust*, *fear*, *happy*, *neutral*, *sad*, dan *surprise*. Dengan rincian sebagai berikut diterangkan dalam tabel 4.1.

2. Data Test

Data Test digunakan untuk mengevaluasi kinerja model pada saat proses pelatihan. Pada *test* data ini terdapat 7 class yaitu *angry*, *disgust*, *fear*, *happy*, *neutral*, *sad*, dan *surprise*. Dengan rincian sebagai berikut diterangkan dalam tabel 4.1.

Tabel 4. 1 Tabel Perbandingan Jumlah Dataset

Jenis Ekspresi	Jumlah Data Testing	Jumlah Data Training
Marah (Angry)	958	3995
Jijik (Disgust)	111	436

Takut (Fear)	1024	4097
Senang (Happy)	1774	7215
Netral (Neutral)	1233	4965
Sedih (Sad)	1247	4830
Terkejut (Surprise)	831	3171

b. Load Dataset

Proses pemuatan dataset ini melibatkan beberapa langkah kunci yang memastikan data siap digunakan untuk pelatihan model. Langkah-langkah tersebut meliputi membaca data dari sumber yang ditentukan, mengelompokkan data ke dalam format yang sesuai, dan melakukan pra-pemrosesan untuk memastikan bahwa data tersebut berada dalam kondisi optimal untuk pelatihan model.

Dalam penelitian ini, kita akan menggunakan **dua kelas** (2-class classification system) untuk klasifikasi ekspresi wajah. Kelas-kelas tersebut adalah:

1. Kelas Mencurigakan: Termasuk ekspresi seperti marah, jijik, takut, terkejut, dan sedih.
2. Kelas Tidak Mencurigakan: Termasuk ekspresi seperti bahagia dan netral.

```
def load_dataset(image_paths, labels, batch_size, is_training=False):
    dataset = tf.data.Dataset.from_tensor_slices((image_paths, labels))
    dataset = dataset.map(preprocess_image,
                        num_parallel_calls=tf.data.AUTOTUNE)

    if is_training:
        dataset = dataset.shuffle(buffer_size=10000)

    dataset = dataset.batch(batch_size)
    dataset = dataset.prefetch(buffer_size=tf.data.AUTOTUNE)

    return dataset
```

Gambar 4. 1 Load Dataset

Fungsi `load_dataset` memuat dataset dari path gambar dan label yang diberikan, kemudian melakukan pra-pemrosesan dan batching.

1. Input:

- a) `image_paths`: List path gambar.
- b) `labels`: List label yang sesuai dengan gambar.
- c) `batch_size`: Ukuran batch untuk pelatihan.
- d) `is_training`: Boolean yang menunjukkan apakah dataset ini untuk pelatihan (training) atau validasi/pengujian (testing).

2. Proses:

a) Membuat Dataset:

```
tf.data.Dataset.from_tensor_slices((image_paths, labels))
```

membuat objek dataset dari path gambar dan label.

b) Pra-pemrosesan: `dataset.map(preprocess_image,`

```
num_parallel_calls=tf.data.AUTOTUNE)
```

 memanggil fungsi

`preprocess_image` untuk setiap gambar, dengan paralelisasi otomatis.

c) Shuffling: Jika `is_training` adalah True, dataset akan diacak dengan buffer size 10000 untuk meningkatkan generalisasi model.

d) Batching: Dataset di-batch sesuai dengan ukuran batch yang ditentukan.

e) Prefetching: `dataset.prefetch(buffer_size=tf.data.AUTOTUNE)` untuk melakukan prefetch data agar training berjalan lebih lancar.

3. Output: Dataset yang sudah diproses dan siap untuk pelatihan atau validasi.

```

def get_image_paths_and_labels(directory):
    class_mencurigakan = ['angry', 'disgust', 'fear', 'surprise', 'sad']
    class_tidak_mencurigakan = ['happy', 'neutral']

    image_paths = []
    labels = []

    for class_name in class_mencurigakan:
        class_dir = os.path.join(directory, class_name)
        for fname in os.listdir(class_dir):
            file_path = os.path.join(class_dir, fname)
            if os.path.isfile(file_path):
                image_paths.append(file_path)
                labels.append(1) # Label "Mencurigakan"

    for class_name in class_tidak_mencurigakan:
        class_dir = os.path.join(directory, class_name)
        for fname in os.listdir(class_dir):
            file_path = os.path.join(class_dir, fname)
            if os.path.isfile(file_path):
                image_paths.append(file_path)
                labels.append(0) # Label "Tidak Mencurigakan"

    return image_paths, labels

```

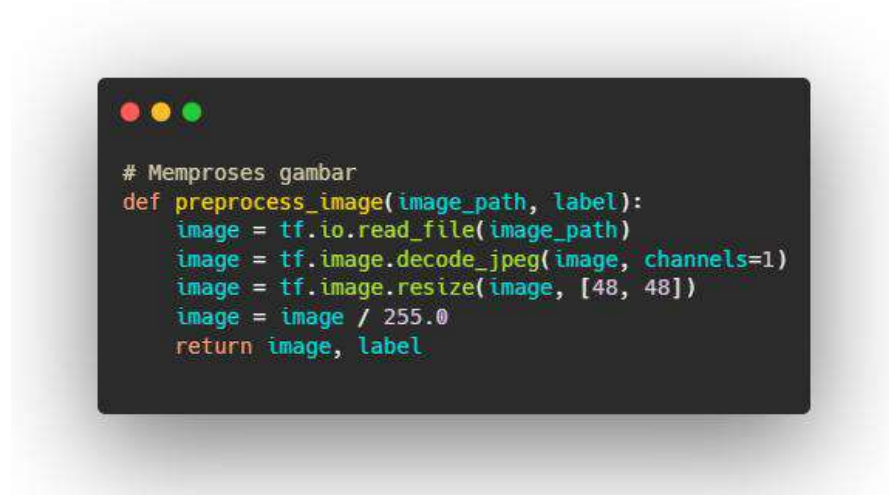
Gambar 4. 2 Path Gambar dan Label

Fungsi *get_image_paths_and_labels* mengambil path gambar dan label dari direktori yang diberikan.

1. Input: *directory* - path ke direktori utama yang berisi subdirektori dengan gambar-gambar.
2. Proses:
 - a) Kelas Mencurigakan: Mengumpulkan path gambar dan memberikan label 1 untuk kelas-kelas yang termasuk dalam *class_mencurigakan*.
 - b) Kelas Tidak Mencurigakan: Mengumpulkan path gambar dan memberikan label 0 untuk kelas-kelas yang termasuk dalam *class_tidak_mencurigakan*.
3. Output: Dua list - *image_paths* berisi path gambar dan *labels* berisi label yang sesuai.

c. Pemrosesan Dataset

Pemrosesan dataset adalah langkah penting untuk menyiapkan data agar siap digunakan dalam pelatihan model Convolutional Neural Networks (CNN) untuk deteksi ekspresi wajah. Berikut adalah langkah-langkah yang dilakukan :



Gambar 4. 3 Preprocess Image

1. Membaca gambar

```
“image = tf.io.read_file(image_path)”
```

Langkah pertama adalah membaca file gambar dari path yang diberikan. Fungsi `tf.io.read_file` membaca konten dari file gambar dalam bentuk byte.

2. Dekode Gambar

```
“image = tf.image.decode_jpeg(image, channels=1)”
```

Gambar yang dibaca dalam bentuk byte kemudian didekode menjadi format gambar yang dapat diproses lebih lanjut. Fungsi `tf.image.decode_jpeg` digunakan untuk mendekode gambar JPEG. Argumen `channels=1` menunjukkan bahwa gambar akan didekode dalam format grayscale (hitam-putih). Ini biasanya dilakukan untuk mengurangi kompleksitas dan ukuran data karena informasi warna tidak selalu diperlukan untuk deteksi ekspresi wajah.

3. Mengubah ukuran gambar

```
“image = tf.image.resize(image, [48, 48])”
```

Setelah gambar didekode, langkah selanjutnya adalah mengubah ukuran gambar menjadi dimensi tertentu. Dalam hal ini, gambar diubah ukurannya menjadi 48x48 piksel. Ini adalah ukuran standar yang sering digunakan dalam dataset ekspresi wajah, seperti dataset FER-2013.

4. Normalisasi gambar

```
“image = image / 255.0”
```

Gambar yang diubah ukurannya kemudian dinormalisasi dengan membagi nilai pikselnya dengan 255.0. Ini mengubah nilai piksel dari rentang [0, 255] menjadi rentang [0, 1]. Normalisasi ini penting untuk mempercepat konvergensi selama pelatihan model dan untuk memastikan bahwa nilai input berada dalam rentang yang sesuai untuk jaringan saraf.

5. Mengembalikan gambar dan label

```
“return image, label”
```

Fungsi ini kemudian mengembalikan gambar yang sudah diproses beserta label yang sesuai. Label digunakan untuk melatih model sehingga dapat belajar mengaitkan gambar dengan kelas ekspresi wajah yang benar.

d. Hyperparameter Tuning (Metode *Grid Search*)

Metode yang penulis terapkan dalam hyperparameter tuning ini adalah *grid search*. Metode ini melakukan pencarian exhaustive pada ruang hyperparameter yang telah ditentukan. Grid search mencoba semua kombinasi yang mungkin dari hyperparameter dan mengevaluasi kinerja model untuk setiap kombinasi.

Dalam konteks penelitian ini, grid search digunakan untuk mencari kombinasi hyperparameter terbaik dengan langkah-langkah sebagai berikut:

1. Definisi Ruang Hyperparameter:

Hyperparameter yang akan diatur termasuk fungsi aktivasi (relu, tanh, sigmoid), ukuran batch (32, 64, 128), dan jumlah epoch (20, 50, 100).

2. Pelatihan dan Evaluasi Model:

- a) Untuk setiap kombinasi hyperparameter, model dilatih pada dataset pelatihan dan dievaluasi pada dataset validasi.
- b) Akurasi validasi digunakan sebagai metrik kinerja untuk menentukan kombinasi hyperparameter terbaik.

3. Penyimpanan Hasil Terbaik:

Kombinasi hyperparameter yang menghasilkan akurasi validasi tertinggi disimpan sebagai kombinasi terbaik.

```

# Mencari kombinasi terbaik
from itertools import product

activations = ['relu', 'tanh', 'sigmoid']
batch_sizes = [32, 64, 128]
epochs_list = [20, 50, 100]

best_accuracy = 0
best_params = {}

for activation, batch_size, epochs in product(activations, batch_sizes,
epochs_list):
    print(f"Training with activation={activation}, batch_size=
{batch_size}, epochs={epochs}")
    train_dataset = load_dataset(train_image_paths, train_labels,
batch_size=batch_size, is_training=True)
    val_dataset = load_dataset(val_image_paths, val_labels,
batch_size=batch_size, is_training=False)
    model = create_model(activation)
    history = model.fit(train_dataset, validation_data=val_dataset,
epochs=epochs)
    val_accuracy = max(history.history['val_accuracy'])

    if val_accuracy > best_accuracy:
        best_accuracy = val_accuracy
        best_params = {
            'activation': activation,
            'batch_size': batch_size,
            'epochs': epochs
        }

print(f"Best parameters: {best_params}")
print(f"Best validation accuracy: {best_accuracy}")

```

Gambar 4. 4 Hyperparameter Tuning

Setelah proses hyperparameter tuning selesai, kombinasi terbaik yang dipilih adalah penggunaan fungsi aktivasi **ReLU (Rectified Linear Unit)**, ukuran batch **32**, dan jumlah epoch **100**. Berikut adalah penjelasan mengenai pemilihan dan hubungan antara hyperparameter tersebut:

1. Fungsi Aktivasi: ReLU (Rectified Linear Unit)

ReLU adalah fungsi aktivasi yang paling umum digunakan dalam jaringan saraf dalam. Fungsi ini mengubah nilai negatif menjadi nol dan mempertahankan nilai positif, dinyatakan sebagai

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

Keunggulan:

- a) Mengatasi Masalah Vanishing Gradient: ReLU membantu mengatasi masalah vanishing gradient, yang sering terjadi pada fungsi aktivasi seperti sigmoid dan tanh. Dengan mempertahankan gradient untuk nilai positif, ReLU memungkinkan gradient descent untuk beroperasi lebih efisien dan cepat.
- b) Sparsity: ReLU menghasilkan output sparsity, yang berarti banyak neuron akan memiliki output nol. Hal ini membantu dalam komputasi yang lebih efisien dan mencegah overfitting.
- c) Konvergensi Cepat: Model dengan fungsi aktivasi ReLU cenderung konvergen lebih cepat dibandingkan dengan fungsi aktivasi lainnya, sehingga mempercepat proses pelatihan.

2. Ukuran Batch: 32

Ukuran batch adalah jumlah sampel yang diproses sebelum model memperbarui parameter. Batch size menentukan seberapa sering model akan diperbarui dalam satu epoch.

Keunggulan:

- a) Efisiensi Komputasi: Ukuran batch sebesar 32 adalah kompromi yang baik antara efisiensi komputasi dan kecepatan konvergensi. Batch yang terlalu kecil bisa menyebabkan model tidak stabil dan batch yang terlalu besar bisa membutuhkan banyak memori dan komputasi.
- b) Generalization: Batch size sebesar 32 sering kali memberikan keseimbangan yang baik antara bias dan variasi, membantu model untuk generalisasi lebih baik pada data yang belum pernah dilihat.
- c) Memori: Ukuran batch 32 biasanya dapat diproses dengan baik oleh GPU modern tanpa memori yang berlebihan, memungkinkan pemrosesan yang lebih cepat dan efisien.

3. Jumlah Epoch: 100

Epoch adalah satu siklus penuh melalui dataset pelatihan. Jumlah epoch menentukan berapa kali model akan melihat seluruh dataset selama pelatihan.

Keunggulan:

- a) Pelatihan Mendalam: Dengan 100 epoch, model memiliki kesempatan yang cukup untuk mempelajari fitur-fitur yang mendalam dan kompleks dari dataset.
- b) *Balancing* : Pemilihan 100 epoch memungkinkan model untuk mempelajari data secara mendalam tanpa terlalu overfitting. Penting untuk memantau kinerja model pada dataset validasi untuk memastikan tidak terjadi overfitting.

Kombinasi hyperparameter yang tepat dapat meningkatkan kinerja model secara signifikan, sedangkan kombinasi yang tidak sesuai dapat menghambat proses pelatihan atau bahkan menyebabkan model tidak dapat belajar dengan baik. Berikut adalah pembahasan mengenai hubungan antara aktivasi, batch size, dan epochs dalam konteks konvergensi dan stabilitas model.

1. Konvergensi dan Stabilitas: Kombinasi fungsi aktivasi ReLU dengan batch size 32 dan 100 epoch mendukung konvergensi yang stabil dan efisien. ReLU membantu menjaga gradient tetap berfungsi, batch size 32 memberikan pembaruan model yang cukup sering untuk stabilitas, dan 100 epoch memberikan cukup waktu untuk pelatihan mendalam.
2. Efisiensi Komputasi: Ukuran batch yang moderat (32) memungkinkan penggunaan memori yang efisien dan komputasi yang cepat, sementara ReLU mempercepat konvergensi. Jumlah epoch yang lebih tinggi memungkinkan model untuk benar-benar mempelajari pola dalam data tanpa terlalu banyak overfitting.

3. Generalization: Kombinasi ini juga memberikan keseimbangan yang baik antara pelatihan yang cukup mendalam (100 epoch) dan pembaruan model yang cukup sering (batch size 32) untuk memastikan model dapat generalisasi dengan baik pada data baru.

d. *Modelling*

Proses pembuatan model untuk sistem deteksi ekspresi identifikasi ekspresi wajah untuk deteksi potensi kejahatan yaitu menggunakan arsitektur CNN

1. Konfigurasi Model CNN

Pada tahap ini, penulis mendefinisikan struktur dari CNN yang akan digunakan untuk memproses gambar. Struktur ini umumnya terdiri dari beberapa layer yang berbeda, seperti pada gambar 4.5

```

# Definisikan model
class EmotionModel(Model):
    def __init__(self, activation='relu', **kwargs):
        super(EmotionModel, self).__init__(**kwargs)
        self.conv1 = layers.Conv2D(32, (3, 3), activation=activation)
        self.conv2 = layers.Conv2D(64, (3, 3), activation=activation)
        self.pool1 = layers.MaxPooling2D(pool_size=(2, 2))
        self.drop1 = layers.Dropout(0.5)

        self.conv3 = layers.Conv2D(128, (3, 3), activation=activation)
        self.pool2 = layers.MaxPooling2D(pool_size=(2, 2))
        self.conv4 = layers.Conv2D(128, (3, 3), activation=activation)
        self.pool3 = layers.MaxPooling2D(pool_size=(2, 2))
        self.drop2 = layers.Dropout(0.5)

        self.flatten = layers.Flatten()
        self.dense1 = layers.Dense(1024, activation=activation)
        self.drop3 = layers.Dropout(0.5)
        self.dense2 = layers.Dense(2, activation='softmax')

    def call(self, inputs):
        x = self.conv1(inputs)
        x = self.conv2(x)
        x = self.pool1(x)
        x = self.drop1(x)

        x = self.conv3(x)
        x = self.pool2(x)
        x = self.conv4(x)
        x = self.pool3(x)
        x = self.drop2(x)

        x = self.flatten(x)
        x = self.dense1(x)
        x = self.drop3(x)
        return self.dense2(x)

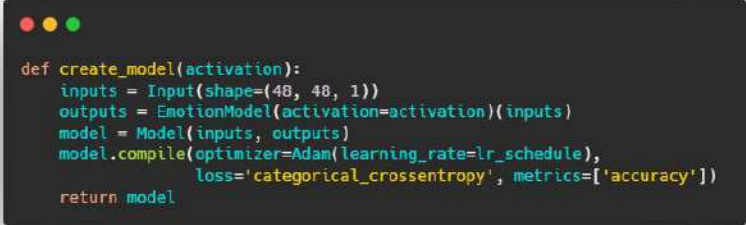
```

Gambar 4. 5 Konfigurasi Arsitektur CNN

- a) Kelas: EmotionModel adalah sebuah kelas yang mewarisi dari kelas Model dalam Keras.
- b) Inisialisasi: Metode `__init__` mendefinisikan struktur jaringan saraf. Parameter `activation` digunakan untuk menentukan fungsi aktivasi yang digunakan di semua layer kecuali layer output.

- c) Superclass Call: `super(EmotionModel,self).__init__(**kwargs)` memanggil inisialisasi dari superclass Model.
- d) Conv2D Layers: Layer konvolusi 2D (Conv2D) dengan filter 32, 64, dan 128 serta ukuran kernel 3x3. Fungsi aktivasi yang digunakan ditentukan oleh parameter `activation`.
- e) MaxPooling2D Layers: Layer pooling (MaxPooling2D) dengan ukuran pool 2x2 digunakan untuk mengurangi dimensi spasial dari fitur peta.
- f) Dropout Layers: Layer dropout (Dropout) dengan rasio dropout 0.5 digunakan untuk mencegah overfitting dengan menonaktifkan setengah neuron secara acak selama pelatihan.
- g) Flatten Layer: Layer Flatten digunakan untuk meratakan output dari layer konvolusi ke dalam vektor satu dimensi sebelum memasukkannya ke dalam layer dense.
- h) Dense Layers:
`Dense(1024, activation=activation)`: Layer dense dengan 1024 neuron dan fungsi aktivasi yang ditentukan.
`Dense(2, activation='softmax')`: Layer output dengan 2 neuron (karena klasifikasi biner) dan fungsi aktivasi softmax untuk mengeluarkan probabilitas kelas.

2. Compile Model



```
def create_model(activation):
    inputs = Input(shape=(48, 48, 1))
    outputs = EmotionModel(activation=activation)(inputs)
    model = Model(inputs, outputs)
    model.compile(optimizer=Adam(learning_rate=lr_schedule),
                  loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

Gambar 4. 6 Fungsi `create_model`

Fungsi `create_model` membuat dan mengkompilasi model Keras menggunakan arsitektur `EmotionModel`.

a) Input: `activation`: Fungsi aktivasi yang akan digunakan dalam layer-layer `EmotionModel`.

b) Proses:

1) Input Layer: `Input(shape=(48, 48, 1))` mendefinisikan layer input dengan bentuk gambar 48x48 piksel dalam skala abu-abu (1 channel).

2) `EmotionModel`: `EmotionModel(activation=activation)(inputs)` memanggil model `EmotionModel` dengan fungsi aktivasi yang ditentukan dan menghubungkan layer input ke model.

3) Model Keras: `Model(inputs, outputs)` membuat model Keras dengan layer input dan output yang telah ditentukan.

4) Compile Model:

- `Optimizer`: `Adam(learning_rate=lr_schedule)` menggunakan optimizer Adam dengan learning rate yang diatur oleh `lr_schedule`.
- `Loss`: `loss='categorical_crossentropy'` menggunakan `categorical_crossentropy` sebagai fungsi kerugian, cocok untuk klasifikasi multi-kelas.
- `Metrics`: `metrics=['accuracy']` menentukan bahwa akurasi akan digunakan sebagai metrik evaluasi model selama pelatihan.

c) Output: Model yang telah dikompilasi siap untuk dilatih.


```
lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(  
    initial_learning_rate=0.0001,  
    decay_steps=100000,  
    decay_rate=0.96,  
    staircase=True  
)
```

Gambar 4. 7 Learning Rate Setup

Pengaturan pembelajaran menggunakan eksponensial decay untuk mengurangi learning rate secara bertahap selama pelatihan.

- a) Initial Learning Rate: `initial_learning_rate=0.0001` mengatur learning rate awal sebesar 0.0001.
- b) Decay Steps: `decay_steps=100000` menentukan jumlah langkah pelatihan sebelum learning rate dikurangi.
- c) Decay Rate: `decay_rate=0.96` menentukan faktor pengurangan learning rate. Setiap 100000 langkah, learning rate dikalikan dengan 0.96.
- d) Staircase: `staircase=True` mengatur decay secara diskrit pada setiap interval `decay_steps`, bukan secara kontinu.

Fungsi `create_model` membuat model klasifikasi ekspresi wajah menggunakan arsitektur *EmotionModel* dengan input gambar 48x48 piksel dalam skala abu-abu. Model ini dikompilasi dengan optimizer Adam yang menggunakan eksponensial decay untuk mengatur learning rate, dan menggunakan categorical crossentropy sebagai fungsi kerugian serta akurasi sebagai metrik evaluasi. Penggunaan learning rate schedule memastikan bahwa learning rate dikurangi secara bertahap selama pelatihan untuk membantu stabilitas dan konvergensi model.

e. *Training Model*

Dalam proses pengembangan model pembelajaran mesin untuk deteksi ekspresi wajah, pemilihan arsitektur model yang tepat sangatlah penting. Berbagai arsitektur telah dicoba untuk menentukan model yang paling optimal. Dalam penelitian ini, beberapa pendekatan telah diujikan, termasuk penggunaan arsitektur jaringan saraf konvolusi (CNN) tradisional, serta model-model pra-trlatih yang terkenal seperti ResNet50, MobileNetV2, dan VGG16. Berikut adalah pengantar mengenai pendekatan yang digunakan dalam pelatihan model:

1. Transfer Learning

a) ResNet50 (Dengan *Fine Tuning*)

ResNet50 (Residual Networks) adalah arsitektur jaringan saraf konvolusi yang terdiri dari 50 layer dengan residual connections yang memungkinkan pelatihan jaringan yang lebih dalam tanpa masalah vanishing gradient.

```
# Model ResNet50
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(48, 48, 3))
base_model.trainable = False # Bekukan lapisan ResNet50

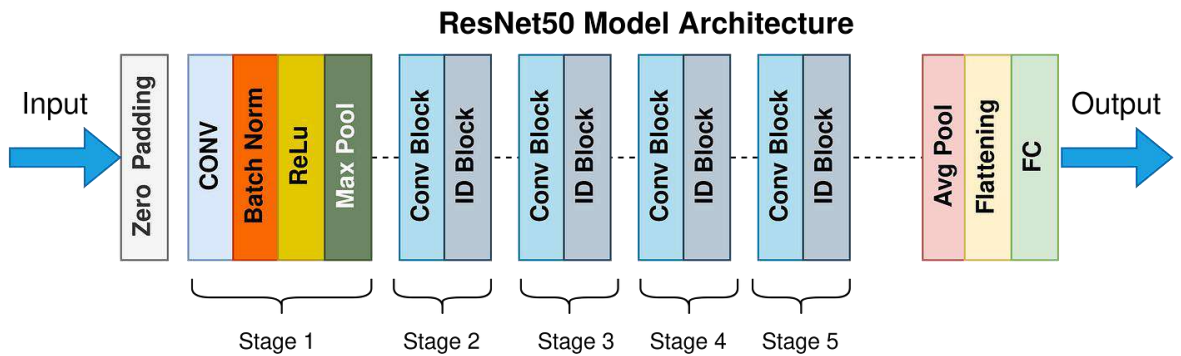
# Menambahkan lapisan baru di atas ResNet50
x = base_model.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(512, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001))(x)
x = layers.Dropout(0.5)(x)
predictions = layers.Dense(7, activation='softmax')(x)

emotion_model = Model(inputs=base_model.input, outputs=predictions)

# Kompilasi model
emotion_model.compile(optimizer=Adam(learning_rate=0.0001),
                    loss='categorical_crossentropy', metrics=['accuracy'])

# Fine-tune beberapa lapisan terakhir
base_model.trainable = True
fine_tune_at = 100 # Membuka 100 lapisan terakhir
for layer in base_model.layers[:fine_tune_at]:
    layer.trainable = False
return go(f, seed, [])
}
```

Gambar 4. 8 Implementasi Model ResNet50



Gambar 4. 9 Arsitektur Model ResNet50

Berikut hasil uji coba menggunakan ResNet50 :

Tabel 4. 2 Hasil Uji Coba ResNet50

No	Epoch	Accuracy (Acc)	Validation Accuracy (Val_acc)	Loss	Validation Loss (Val_loss)
1.0	1.0	0.3967	0.202	2.2649	2.7236
2.0	2.0	0.4125	nan	2.0847	nan

Berikut adalah beberapa kesimpulan yang dapat diambil:

- 1) Peningkatan Akurasi Pelatihan: Akurasi pelatihan meningkat dari 39.67% pada epoch pertama menjadi 41.25% pada epoch kedua, menunjukkan bahwa model mulai memahami pola dalam data pelatihan.
- 2) Penurunan Loss Pelatihan: Penurunan loss pelatihan dari 2.2649 menjadi 2.0847 menunjukkan bahwa model semakin baik dalam memprediksi label yang benar untuk data pelatihan.
- 3) Kinerja Validasi: Pada epoch pertama, akurasi validasi hanya 20.20% dengan loss 2.7236, menunjukkan bahwa model belum menggeneralisasi dengan baik pada data validasi. Tidak adanya data validasi pada epoch kedua membuat sulit untuk menilai perbaikan dalam generalisasi.

b) MobileNet V2 (Dengan *Fine Tuning*)

MobileNetV2 adalah salah satu arsitektur jaringan saraf konvolusi (CNN) yang dirancang khusus untuk aplikasi pada perangkat dengan sumber daya terbatas seperti ponsel dan perangkat edge. Dikembangkan oleh Google, MobileNetV2 adalah penerus dari MobileNetV1 dan menawarkan beberapa perbaikan signifikan dalam hal efisiensi dan kinerja.

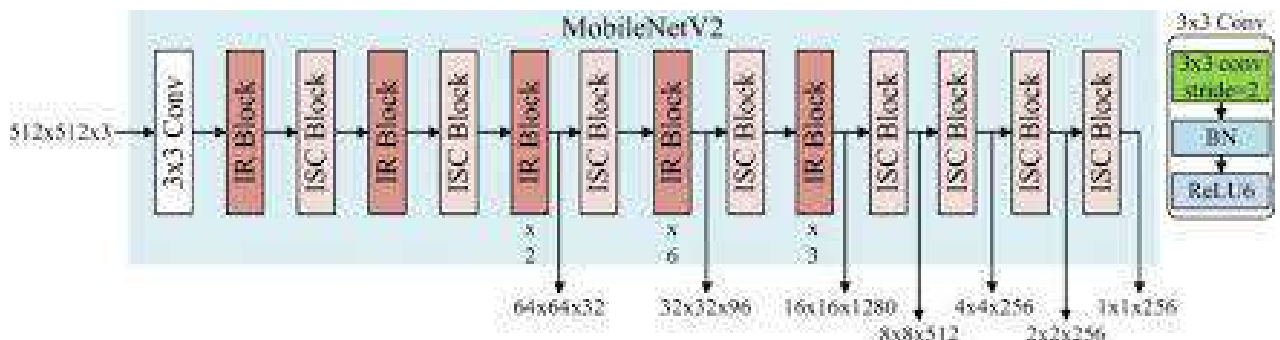
```
# Model MobileNetV2
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
base_model.trainable = False # Bekukan lapisan MobileNetV2
# Menambahkan lapisan baru di atas MobileNetV2
x = base_model.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(512, activation='relu', kernel_regularizer=tf.keras.regularizers.l2(0.001))(x)
x = layers.Dropout(0.5)(x)
predictions = layers.Dense(7, activation='softmax')(x)

emotion_model = Model(inputs=base_model.input, outputs=predictions)
# Kompilasi model

emotion_model.compile(optimizer=Adam(learning_rate=0.00001), loss='categorical_crossentropy',
metrics=['accuracy'])

# Fine-tune beberapa lapisan terakhir
base_model.trainable = True
fine_tune_at = 100 # Membuka 100 lapisan terakhir
for layer in base_model.layers[:fine_tune_at]:
    layer.trainable = False
```

Gambar 4. 10 Implementasi Model MobileNetV2



Gambar 4. 11 Arsitektur MobileNetV2

Berikut hasil uji coba menggunakan MobileNetV2 :

Tabel 4. 3 Hasil Uji Coba MobileNetV2

No	Epoch	Accuracy		Loss	
		Acc	Val_acc	Loss	Val_loss
1	Epoch 1	0.2672	0.2685	2.9112	2.6320
2	Epoch 2	0.2578	0.3211	2.8022	2.5379
3	Epoch 3	0.2685	0.3346	2.6865	2.4709
4	Epoch 4	0.2747	0.3486	2.6094	2.4262
5	Epoch 5	accuracy: 0.2155 Accuracy -- Stop Training			

Meskipun model menunjukkan peningkatan dalam akurasi validasi dan penurunan loss validasi dari epoch 1 hingga epoch 4, masih terdapat beberapa kekurangan yang signifikan:

1) Akurasinya Rendah:

Akurasi pelatihan dan validasi tetap rendah, berkisar sekitar 26.85% hingga 34.86%. Ini menunjukkan bahwa model belum cukup efektif dalam mempelajari fitur-fitur penting dari data.

2) Ketidakstabilan Pelatihan:

Fluktuasi dalam akurasi pelatihan, seperti yang terlihat pada epoch kelima, menunjukkan potensi ketidakstabilan dalam proses pelatihan. Ini dapat mengindikasikan bahwa model belum mencapai stabilitas yang diperlukan untuk pelatihan yang konsisten.

c) VGG16

VGG16 adalah salah satu arsitektur jaringan saraf konvolusi (Convolutional Neural Network, CNN) yang terkenal dan banyak

digunakan dalam tugas-tugas klasifikasi gambar. Dikembangkan oleh Simonyan dan Zisserman dari Visual Geometry Group (VGG) di Universitas Oxford, model ini pertama kali diperkenalkan dalam makalah "Very Deep Convolutional Networks for Large-Scale Image Recognition" pada tahun 2014. VGG16 dikenal karena kesederhanaannya dalam desain namun menunjukkan kinerja yang sangat baik dalam berbagai kompetisi dan aplikasi klasifikasi gambar.

Untuk menyesuaikan model VGG16 dengan tugas klasifikasi spesifik, pendekatan yang efektif adalah membuka beberapa lapisan terakhir dari model. Proses ini memungkinkan lapisan-lapisan tersebut untuk dilatih ulang menggunakan dataset yang baru, sehingga fitur-fitur umum yang telah dipelajari oleh model sebelumnya dapat diadaptasi menjadi lebih spesifik dan relevan untuk tugas klasifikasi yang dihadapi.

```
# Model VGG16
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(48, 48, 3))
base_model.trainable = False # Bekukan lapisan VGG16

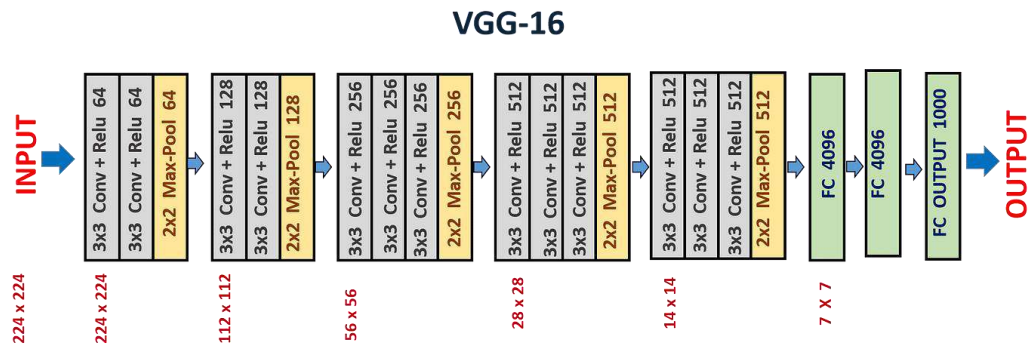
# Menambahkan lapisan baru di atas VGG16
x = base_model.output
x = layers.Flatten()(x)
x = layers.Dense(512, activation='relu')(x)
x = layers.Dropout(0.5)(x)
predictions = layers.Dense(2, activation='softmax')(x)

emotion_model = Model(inputs=base_model.input, outputs=predictions)

# Kompilasi model
emotion_model.compile(optimizer=Adam(learning_rate=0.0001),
                      loss='categorical_crossentropy', metrics=['accuracy'])

# Fine-tune beberapa lapisan terakhir
base_model.trainable = True
fine_tune_at = 15 # Membuka 15 lapisan terakhir
for layer in base_model.layers[:fine_tune_at]:
    layer.trainable = False
```

Gambar 4. 12 Implementasi FineTuning pada Model VGG16



Gambar 4. 13 Arsitektur VGG16

Berikut hasil uji coba menggunakan transfer learning VGG16 :

Tabel 4. 4 Hasil Train VGG16

No	Epoch	Accuracy (Acc)	Validation Accuracy (Val_acc)	Loss	Validation Loss (Val_loss)
1.	1.0	0.8947	0.5177	0.233	0.8534
2.	2.0	0.8862	0.5599	0.2553	0.7811
3.	3.0	0.9024	0.7108	0.2278	0.5705
4.	4.0	0.9178	0.6676	0.1994	0.7051
5.	5.0	0.918	0.6651	0.2079	0.6713
6.	6.0	0.9181	0.6917	0.199	0.6613
7.	7.0	0.9306	0.6984	0.177	0.6554
8.	8.0	0.9371	0.7239	0.1647	0.5746
9.	9.0	0.9434	0.7569	0.1483	0.5711
10.	10.0	0.9567	0.7548	0.1182	0.6442

11.	11.0	0.9638	0.7831	0.1037	0.5573
12.	12.0	0.9678	0.7758	0.0902	0.6246
13.	13.0	0.9732	0.7774	0.0792	0.6913
14.	14.0	0.9786	0.7778	0.063	0.7135
15.	15.0	0.9832	0.7857	0.0535	0.7545
16.	16.0	0.9878	0.7823	0.0381	0.8247
17.	17.0	0.9884	0.7888	0.036	0.8922
18.	18.0	0.9902	0.7947	0.0301	0.8701
19.	19.0	0.9927	0.7955	0.0248	0.9686
20.	20.0	0.9936	0.797	0.0196	0.9841
21.	21.0	0.9953	0.7935	0.0143	1.0422

Berdasarkan hasil pelatihan yang diberikan dari epoch 1 hingga epoch 21, berikut adalah beberapa poin utama yang dapat disimpulkan:

```

precision    recall  f1-score   support

Tidak Mencurigakan    0.72    0.78    0.75    3007
Mencurigakan          0.83    0.78    0.81    4171

 accuracy                    0.78    7178
 macro avg                   0.78    0.78    0.78    7178
 weighted avg                0.79    0.78    0.78    7178

```

Gambar 4. 14 Hasil Evaluasi Model dari Training Menggunakan VGG16

1) Peningkatan Akurasi Pelatihan:

Akurasi pelatihan meningkat secara signifikan dari 89.47% pada epoch pertama menjadi 99.53% pada epoch ke-21. Ini menunjukkan bahwa model semakin mampu mengenali pola dalam data pelatihan dengan sangat baik seiring berjalannya waktu.

2) Peningkatan Akurasi Validasi

Akurasi validasi juga mengalami peningkatan dari 51.77% pada epoch pertama menjadi sekitar 79.70% pada epoch ke-20. Ini menunjukkan bahwa model mampu menggeneralisasi pola yang dipelajari dari data pelatihan ke data validasi dengan cukup baik.

3) Penurunan Loss Pelatihan:

Loss pelatihan menurun dari 0.2330 pada epoch pertama menjadi 0.0143 pada epoch ke-21. Penurunan ini menunjukkan bahwa model semakin baik dalam memprediksi label yang benar untuk data pelatihan, dengan kesalahan yang semakin kecil.

4) Penurunan Loss Validasi:

Loss validasi menurun dari 0.8534 pada epoch pertama menjadi 0.5573 pada epoch ke-11, namun meningkat kembali pada epoch-epoch selanjutnya, mencapai 1.0422 pada epoch ke-21. Ini menunjukkan bahwa meskipun ada peningkatan dalam akurasi validasi, ada fluktuasi yang signifikan dalam loss validasi.

5) Learning Rate:

Learning rate awal yang digunakan adalah $1.0000e-04$, kemudian menurun menjadi $2.0000e-05$ pada epoch ke-9 dan seterusnya, serta $1.0000e-05$ pada epoch ke-17 dan seterusnya. Penurunan learning rate ini mungkin membantu model untuk melakukan penyesuaian lebih halus pada parameter, yang dapat berkontribusi pada peningkatan akurasi.

6) Overfitting Potensial:

Meskipun akurasi pelatihan sangat tinggi, akurasi validasi yang lebih rendah dan peningkatan loss validasi pada epoch-epoch selanjutnya dapat mengindikasikan adanya potensi overfitting, di mana model terlalu

menyesuaikan diri dengan data pelatihan dan kurang mampu menangani data baru.



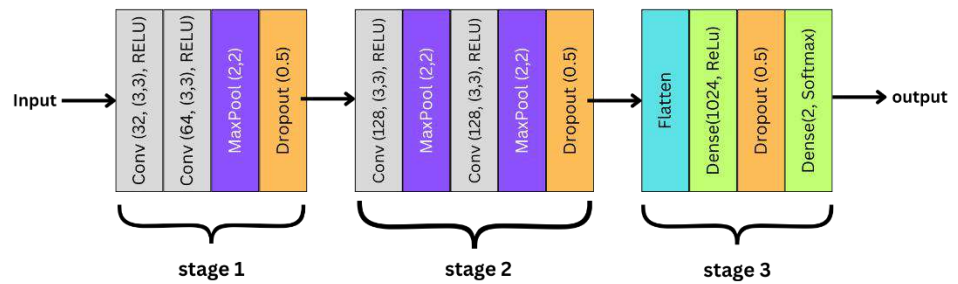
Gambar 4. 15 Grafik Visualisasi Loss dan Accucary dari Training Model dengan VGG16

7) Stabilitas Pelatihan:

Meskipun ada peningkatan yang konsisten dalam akurasi pelatihan dan validasi, fluktuasi dalam loss validasi menunjukkan bahwa model mungkin memerlukan penyesuaian lebih lanjut untuk mencapai kinerja yang lebih stabil dan optimal pada data validasi.

2. CNN

Pada bagian ini, kita akan membahas arsitektur Convolutional Neural Network (CNN) yang digunakan dalam penelitian ini untuk mendeteksi dan mengklasifikasikan ekspresi wajah. Untuk mencapai kinerja optimal, model CNN dalam penelitian ini telah dioptimalkan melalui proses hyperparameter tuning. Berdasarkan hasil dari proses tersebut, kombinasi parameter terbaik yang ditemukan adalah sebagai berikut:



Gambar 4. 16 Arsitektur CNN

```

# Melatih model dengan parameter terbaik
activation = 'relu'
batch_size = 32
epochs = 100

print(f"Training with activation={activation}, batch_size={batch_size}, epochs={epochs}")
train_dataset = load_dataset(train_image_paths, train_labels, batch_size=batch_size,
is_training=True)
val_dataset = load_dataset(val_image_paths, val_labels, batch_size=batch_size,
is_training=False)
model = create_model(activation)
history = model.fit(train_dataset, validation_data=val_dataset, epochs=epochs)

# Save the trained model
model.save('model_deteksi_ekspresi_6.h5')

```

Gambar 4. 17 Code Training Model Menggunakan Hasil Hyperparameter Tuning

- a) Fungsi Aktivasi (Activation Function): ReLU (Rectified Linear Unit)
ReLU adalah fungsi aktivasi yang umum digunakan dalam jaringan saraf karena kemampuannya untuk mengatasi masalah vanishing gradient dan mempercepat konvergensi model.
- b) Ukuran Batch (Batch Size): 32
Ukuran batch menentukan jumlah sampel yang akan diproses sebelum memperbarui parameter model. Ukuran batch 32

memberikan keseimbangan yang baik antara kecepatan pelatihan dan stabilitas gradien.

c) Jumlah Epoch (Epochs): 100

Jumlah epoch menentukan berapa kali seluruh dataset pelatihan akan diproses oleh model. Dengan 100 epoch, model memiliki cukup waktu untuk mempelajari pola-pola dalam data tanpa risiko overfitting yang terlalu tinggi.

Berikut hasil training menggunakan arsitektur CNN dengan menggunakan parameter dari hyperparameter tuning sebelumnya :

Tabel 4. 5 Hasil Train Epoch ke 1-25

No	Epoch	Accuracy (Acc)	Validation Accuracy (Val_acc)	Loss	Validation Loss (Val_loss)
1.	1.0	0.8657	0.4519	0.2679	0.7845
2.	2.0	0.8423	0.5474	0.3009	0.6961
3.	3.0	0.8579	0.5705	0.278	0.6813
4.	4.0	0.8803	0.5652	0.2521	0.6974
5.	5.0	0.8789	0.5623	0.2506	0.7471
6.	6.0	0.8804	0.6117	0.2488	0.6447
7.	7.0	0.8927	0.5815	0.2309	0.6779
8.	8.0	0.8934	0.6279	0.2334	0.6716
9.	9.0	0.9043	0.6066	0.2108	0.6881
10.	10.0	0.9033	0.6573	0.2163	0.61
11.	11.0	0.9079	0.6474	0.2087	0.6381

12.	12.0	0.9079	0.6851	0.2086	0.5832
13.	13.0	0.9127	0.6755	0.1995	0.583
14.	14.0	0.9164	0.6913	0.1977	0.577
15.	15.0	0.9175	0.6459	0.195	0.6333
16.	16.0	0.9182	0.695	0.196	0.5695
17.	17.0	0.921	0.7035	0.188	0.5621
18.	18.0	0.9198	0.6792	0.1922	0.5893
19.	19.0	0.9231	0.7072	0.185	0.5627
20.	20.0	0.9218	0.6895	0.1882	0.6068
21.	21.0	0.9216	0.7104	0.1785	0.5623
22.	22.0	0.9264	0.7527	0.1737	0.4914
23.	23.0	0.9335	0.7056	0.1662	0.5848
24.	24.0	0.9278	0.7325	0.1716	0.5433
25.	25.0	0.9291	0.7193	0.1744	0.562
...

Tabel 4. 6 Hasil Train Epoch 75-100

No	Epoch	Accuracy (Acc)	Validation Accuracy (Val_acc)	Loss	Validation Loss (Val_loss)
75.	75.0	0.9589	0.7995	0.1042	0.4701
76.	76.0	0.9624	0.7976	0.0915	0.4686

77.	77.0	0.9632	0.8037	0.0944	0.4509
78.	78.0	0.965	0.7881	0.0914	0.502
79.	79.0	0.9611	0.7977	0.0978	0.4752
80.	80.0	0.9639	0.8008	0.0914	0.4794
81.	81.0	0.9649	0.7981	0.0902	0.4848
82.	82.0	0.9667	0.8104	0.0868	0.4565
83.	83.0	0.9649	0.8004	0.0875	0.4644
84.	84.0	0.9653	0.7843	0.0882	0.5013
85.	85.0	0.9644	0.8031	0.0922	0.4882
86.	86.0	0.9671	0.7842	0.0842	0.5137
87.	87.0	0.9665	0.8037	0.0859	0.4687
88.	88.0	0.9685	0.8121	0.0822	0.4649
89.	89.0	0.9671	0.8111	0.0861	0.4561
90.	90.0	0.9702	0.7901	0.0807	0.5185
91.	91.0	0.9636	0.788	0.0907	0.4929
92.	92.0	0.9654	0.8062	0.0876	0.4613
93.	93.0	0.9693	0.7892	0.0801	0.5094
94.	94.0	0.9654	0.8018	0.0885	0.4828
95.	95.0	0.9695	0.8079	0.0784	0.4792
96.	96.0	0.9684	0.8026	0.0809	0.4775
97.	97.0	0.9678	0.8082	0.0799	0.4754
98.	98.0	0.9708	0.8048	0.0759	0.4797

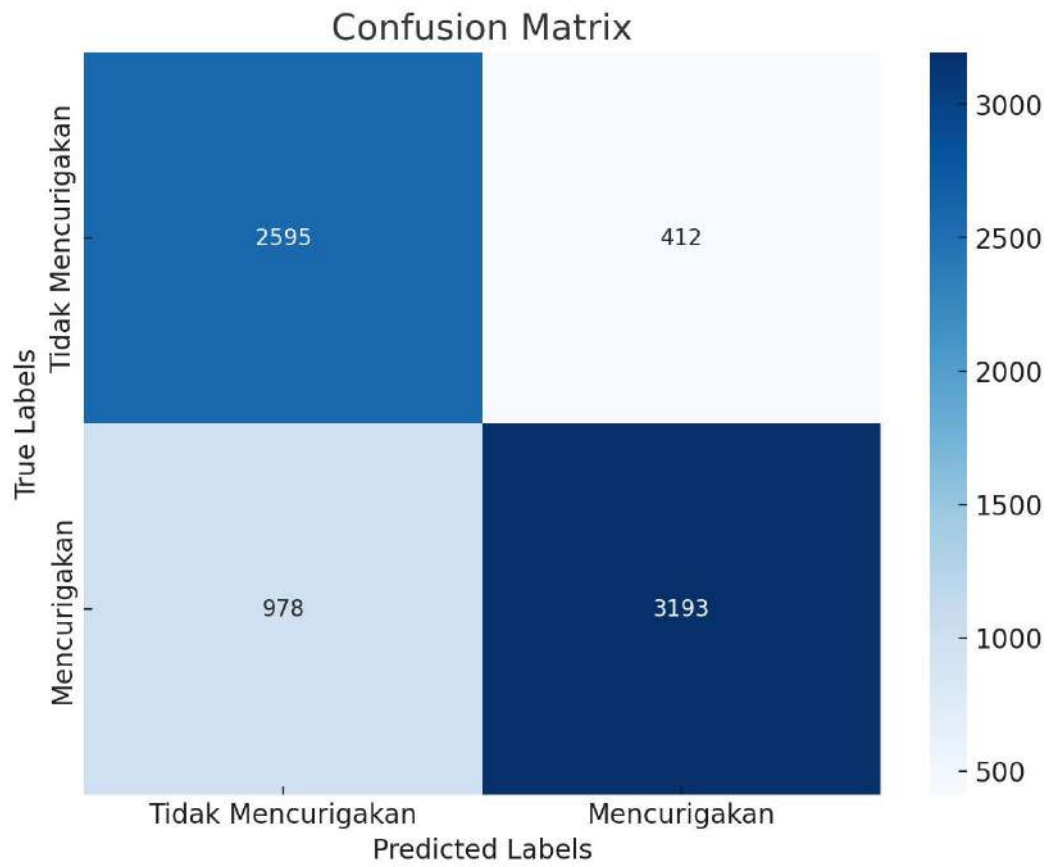
99.	99.0	0.9718	0.8149	0.0716	0.4751
100.	100.0	0.9742	0.8064	0.0698	0.4774

f. Evaluasi

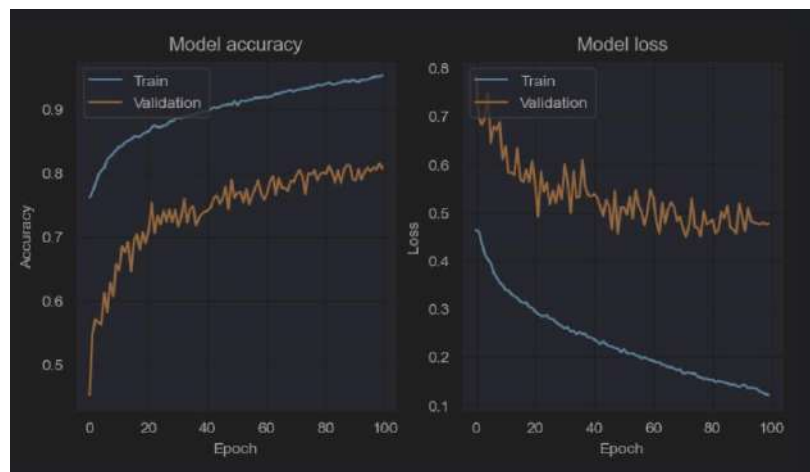
Model yang telah dilatih dievaluasi menggunakan data validasi untuk mengukur kinerjanya. Proses evaluasi ini melibatkan pengukuran akurasi, presisi, recall, dan metrik lainnya untuk memastikan bahwa model bekerja dengan baik dalam mengenali ekspresi potensi kejahatan. *Confusion matrix* digunakan dalam mengevaluasi kinerja model klasifikasi dengan membandingkan hasil prediksi model dengan nilai sebenarnya dari data yang diuji. Untuk melihat hasil evaluasi kinerja model dapat dilihat pada Tabel 4.7, Gambar 4 .12

Tabel 4. 7 Hasil Evaluasi Model

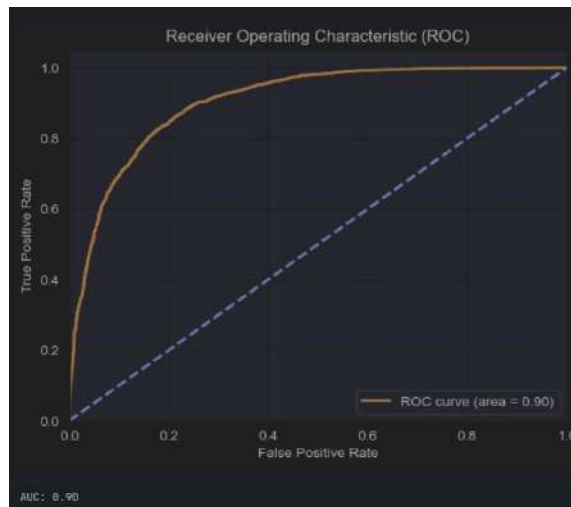
	Precision	Recall	F1-Score	Support
Tidak Mencurigakan	0.7263	0.8629	0.7887	3007
Mencurigakan	0.8857	0.7655	0.8212	4171
Accuracy	0.8064	0.8064	0.8064	7178
Macro Avg	0.8060	0.8143	0.8050	7178
Weighted Avg	0.8189	0.8064	0.8076	7178



Gambar 4. 18 Confussion Matrix



Gambar 4. 19 Grafik Loss dan Accuracy Model



Gambar 4. 20 ROC

g. Menyimpan model

Setelah melakukan semua tahapan selanjutnya model akan di simpan dalam bentuk .H5.

3. Implementasi Sistem

Setelah melatih model dengan menggunakan CNN, tahap berikutnya adalah mendeploy model. Adapun tahapan-tahapan implementasi sistem sebagai berikut:

a. Pengumpulan Kebutuhan Sistem

1. Kebutuhan Fungsional

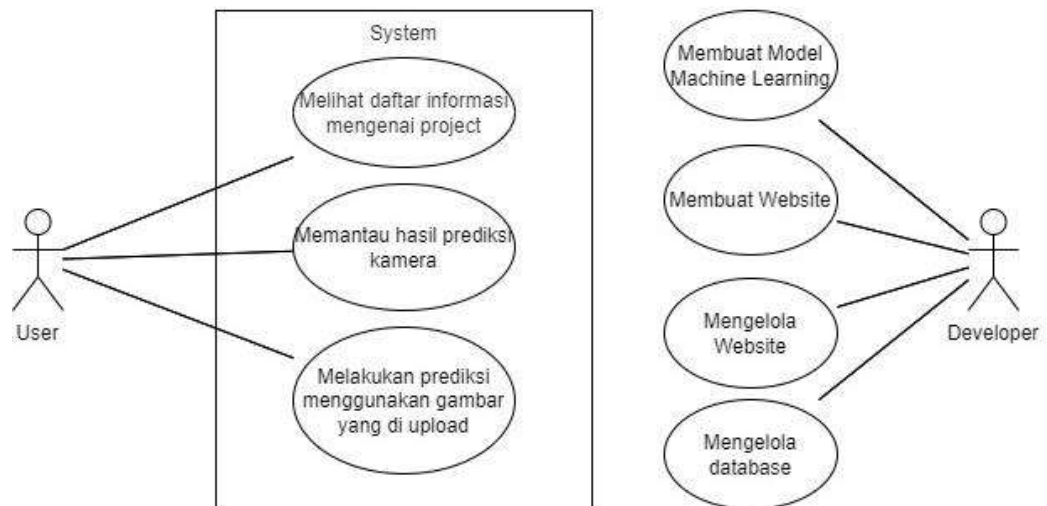
- a) Pada *detection* aplikasi bisa digunakan untuk mendeteksi ekspresi dengan menggunakan upload gambar
- b) Pada *detection* aplikasi bisa digunakan untuk mendeteksi ekspresi dengan menggunakan kamera
- c) Tombol start dan stop bisa berfungsi untuk mengoperasikan kamera

2. Desain *Unified Modelling Language* (UML)

a) *Use Case Diagram*

Use Case Diagram menggambarkan interaksi antara aktor (pengguna) dengan sistem yang akan dikembangkan.

Diagram ini menunjukkan fungsi-fungsi utama yang dapat dilakukan oleh pengguna.



Gambar 4. 21 Use Case Diagram

a) *Activity Diagram*

Activity Diagram menunjukkan alur kerja atau aktivitas dalam sistem. Diagram ini membantu memahami proses yang terjadi dalam aplikasi dari awal hingga akhir.

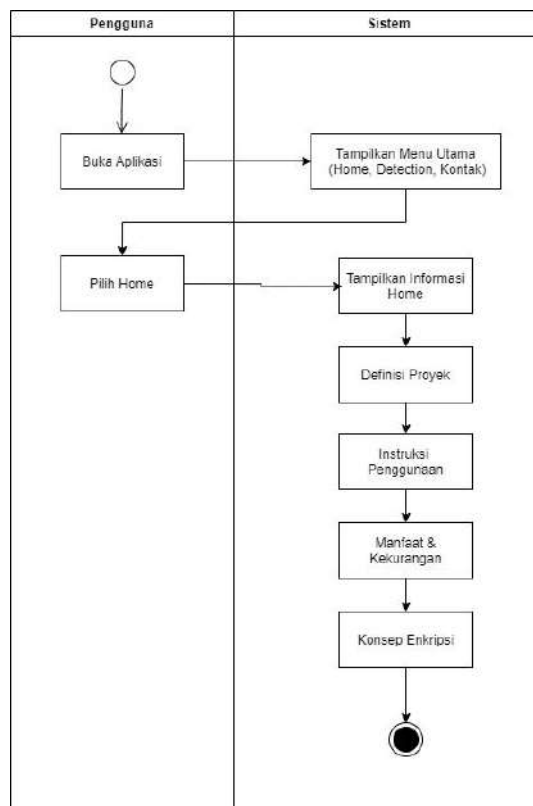
1) *Activity Diagram Home*

Pada *Activity Diagram Home* menjelaskan tentang bagaimana alur pengguna dalam menerima informasi pada halaman *Home*. Pengguna memulai interaksi dengan aplikasi dengan membukanya. Pada tahap ini, aplikasi Streamlit diluncurkan, dan pengguna melihat antarmuka utama aplikasi. Setelah aplikasi dibuka, sistem menampilkan menu utama yang berisi tiga pilihan: Home, Deteksi, dan Kontak. Menu ini memungkinkan pengguna untuk menavigasi ke bagian yang berbeda dari aplikasi. Pengguna memilih opsi "Home" dari menu utama. Ini merupakan langkah awal di mana pengguna ingin mengetahui lebih banyak tentang proyek dan fitur-fitur

aplikasi. Sistem merespons pilihan pengguna dengan menampilkan halaman Home. Halaman ini berisi beberapa informasi penting:

- a. Penjelasan mengenai proyek deteksi ekspresi wajah menggunakan algoritma Convolutional Neural Network (CNN).
- b. Langkah-langkah yang perlu diikuti pengguna untuk menggunakan aplikasi ini dengan efektif.
- c. Kelebihan dan kekurangan dari proyek ini, memberikan pengguna wawasan tentang apa yang bisa diharapkan.
- d. Penjelasan tentang bagaimana gambar wajah dienkripsi sebelum disimpan untuk menjaga privasi dan keamanan data..

Activity Diagram Home dapat dilihat pada Gambar 4. 21



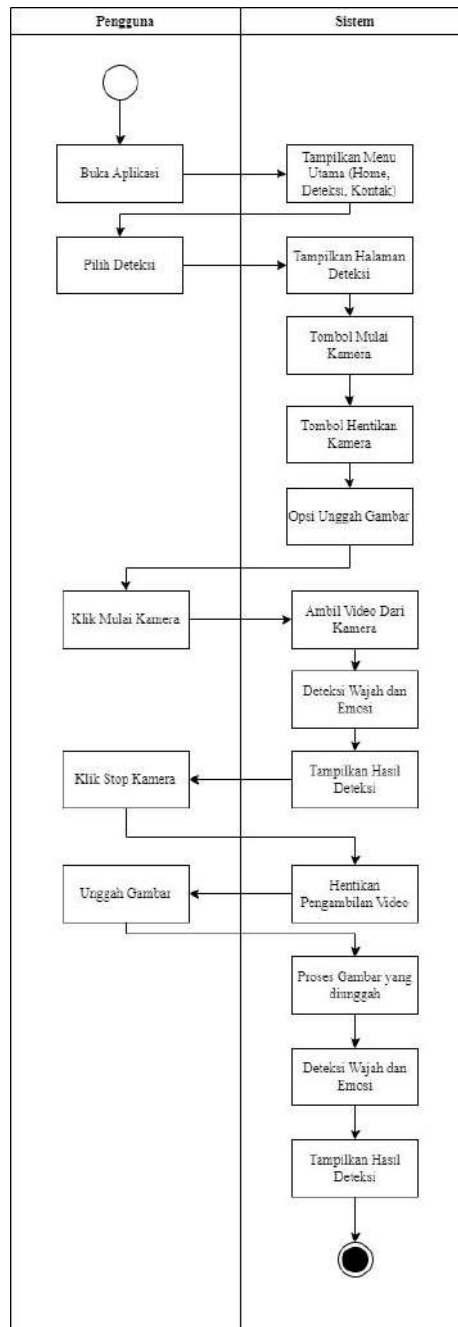
Gambar 4. 22 Activity Diagram : Homepage

2) *Activity Diagram Detection*

Pada *Activity Diagram Detection* memberikan gambaran bagaimana pengguna dalam menggunakan fitur deteksi kamera dan deteksi menggunakan gambar yang diupload. Setelah memahami informasi di halaman Home, pengguna dapat memilih untuk melanjutkan ke halaman Deteksi. Seperti aktivitas Home, pengguna membuka aplikasi Streamlit dan sistem menampilkan menu utama dengan pilihan Home, Deteksi, dan Kontak. Pengguna memilih opsi "Deteksi" dari menu utama untuk memulai proses deteksi ekspresi wajah. Sistem menampilkan halaman Deteksi yang menyediakan beberapa fitur utama:

- a. Tombol Mulai Kamera: Tombol ini memungkinkan pengguna untuk memulai deteksi wajah secara real-time melalui kamera.
- b. Tombol Hentikan Kamera: Tombol ini digunakan untuk menghentikan deteksi wajah yang sedang berjalan.
- c. Opsi Unggah Gambar: Pengguna dapat mengunggah gambar wajah untuk dideteksi ekspresi wajahnya.
- d. Klik Mulai Kamera: Pengguna mengklik tombol "Mulai Kamera" untuk memulai pengambilan video secara real-time dari kamera perangkat.
- e. Ambil Video dari Kamera: Sistem mulai menangkap video dari kamera dan memproses setiap frame untuk mendeteksi wajah dan emosi yang terekspresi.
- f. Deteksi Wajah dan Emosi: Sistem menggunakan model yang sudah dilatih untuk mendeteksi wajah dan menganalisis emosi yang terekspresi pada wajah yang terdeteksi.

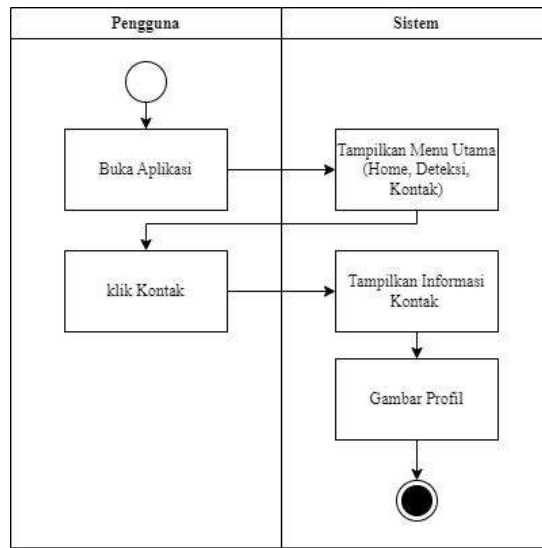
- g. Tampilkan Hasil Deteksi: Sistem menampilkan hasil deteksi dengan menambahkan kotak pembatas (bounding box) dan label emosi pada wajah yang terdeteksi dalam video.
- h. Klik Hentikan Kamera: Pengguna mengklik tombol "Hentikan Kamera" untuk menghentikan pengambilan video.
- i. Hentikan Pengambilan Video: Sistem menghentikan proses pengambilan video dan deteksi wajah.
- j. Unggah Gambar: Pengguna mengunggah gambar melalui opsi yang tersedia.
- k. Proses Gambar yang Diunggah: Sistem memproses gambar yang diunggah oleh pengguna, melakukan deteksi wajah dan analisis emosi.
- l. Tampilkan Hasil Deteksi: Sistem menampilkan hasil deteksi pada gambar yang diunggah dengan menambahkan kotak pembatas dan label emosi pada wajah yang terdeteksi.



Gambar 4. 23 Activity Diagram : Detection Page

3) Activity Diagram Contact

Pengguna membuka aplikasi Streamlit dan sistem menampilkan menu utama dengan pilihan Home, Deteksi, dan Kontak. Pengguna memilih opsi "Kontak" dari menu utama untuk melihat informasi kontak.



Gambar 4. 24 Activity Diagram : Contact

3. Desain Wireframe

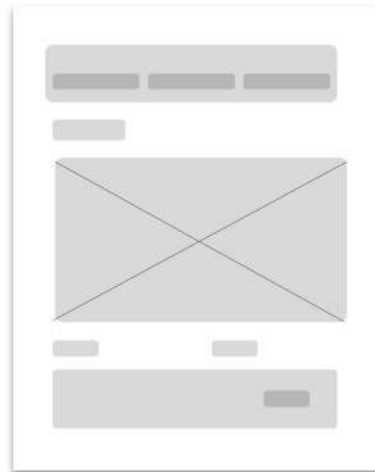
Wireframe merupakan representasi visual sederhana dari sebuah antarmuka pengguna. Adapun perancangan antarmuka yang dibuat dalam aplikasi. Berikut ini desain *wireframe* aplikasi deteksi ekspresi potensi kejahatan.



Gambar 4. 25 Wireframe Halaman Hone



Gambar 4. 26 Wireframe Halaman Detection Sebelum Kamera Aktif



Gambar 4. 27 Wireframe Halaman Detection Setelah Kamera Aktif



Gambar 4. 28 Wireframe Halaman Kontak

b. Membangun Prototype

Prototype dari aplikasi mulai dibangun berdasarkan kebutuhan yang telah dikumpulkan. Desain *UI/UX* dibuat untuk memastikan antarmuka yang mudah digunakan dan menarik bagi pengguna.

1. Halaman Home



Gambar 4. 29 Prototype Halaman Home

Pada halaman Home, pengguna dapat mempelajari informasi proyek, termasuk manfaat dan tujuannya, serta kelebihan dan kekurangannya. Selain itu, pengguna juga dapat memahami bagaimana gambar wajah dienkripsi untuk menjaga privasi dan keamanan data sebelum diunggah ke dalam database.

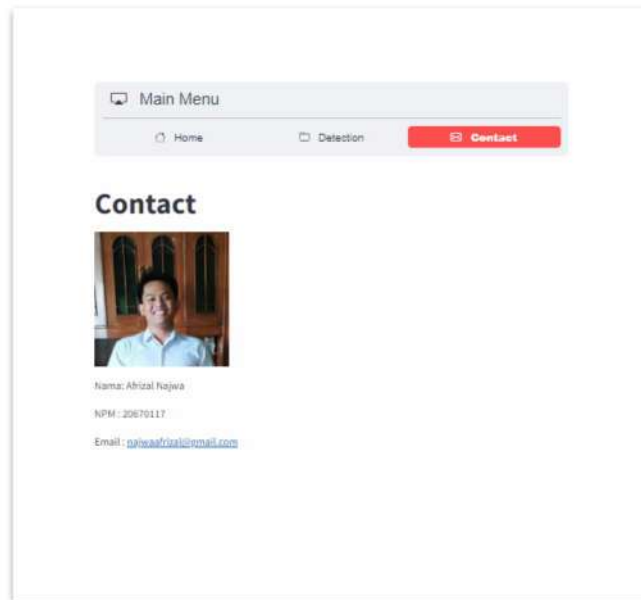
2. Halaman Detection



Gambar 4. 30 Prototype Halaman Detection

Di halaman ini, pengguna dapat menggunakan beberapa fitur utama, termasuk "Start Camera" untuk memulai kamera dan mendeteksi ekspresi wajah secara real-time, serta "Stop Camera" untuk menghentikan kamera. Selain itu, terdapat fitur untuk mengunggah gambar wajah dengan cara menarik dan melepaskan file ke area yang disediakan atau dengan mengklik tombol "Browse files". Area unggahan memiliki batas ukuran file maksimum 200MB per file. Fitur-fitur ini dirancang untuk memudahkan pengguna dalam mendeteksi ekspresi wajah melalui kamera langsung atau melalui gambar yang diunggah.

3. Halaman Kontak



Gambar 4. 31 Prototype Halaman Kontak

Pada halaman ini, pengguna dapat menemukan informasi kontak dari pengembang atau pemilik situs. Halaman ini memudahkan pengguna untuk menghubungi pengembang untuk pertanyaan lebih lanjut, dukungan teknis, atau kolaborasi terkait proyek yang dikembangkan.

c. Evaluasi Prototype

Evaluasi prototype adalah tahap penting dalam pengembangan sistem untuk memastikan hasil akhir sesuai dengan harapan dan kebutuhan klien. Halaman Home memberikan penjelasan umum tentang proyek, termasuk penggunaan CNN untuk deteksi ekspresi wajah dan enkripsi data, dengan informasi yang disajikan secara jelas dan terstruktur, serta meningkatkan kepercayaan pengguna terhadap keamanan data. Halaman Detection memungkinkan pengguna untuk memulai dan menghentikan kamera serta mengunggah gambar untuk deteksi ekspresi wajah, dengan desain

yang sederhana dan intuitif, serta fitur yang mudah ditemukan dan digunakan. Halaman Contact menampilkan informasi kontak pengembang dengan jelas dan mudah diakses, memberikan tampilan yang sederhana dan efektif. Secara keseluruhan, prototype ini sudah cukup baik dan memberikan pengalaman pengguna yang memadai, dengan fitur yang berfungsi sesuai kebutuhan pengguna.

d. Mengkodekan Sistem

Pada bagian ini akan dijelaskan proses pengkodean sistem aplikasi deteksi ekspresi potensi kejahatan menggunakan streamlit dan penyimpanan menggunakan SQLite.

1. Pembuatan Web berbasis Streamlit

Streamlit dipilih karena fitur-fitur seperti visualisasi data, integrasi langsung dengan pustaka Python seperti Pandas dan Matplotlib, serta kemampuan untuk menampilkan model machine learning secara real-time menjadikannya pilihan ideal untuk aplikasi data-driven. Selain itu, Streamlit menawarkan antarmuka pengguna yang ramah dan responsif, serta mendukung deployment cepat ke berbagai platform, sehingga sangat cocok untuk prototyping dan presentasi proyek. Adapun langkah langkah pengembangannya sebagai berikut :

a) *Install Requirements Library*

Untuk membangun aplikasi ini, beberapa pustaka penting perlu diimpor dan diinstal terlebih dahulu. Pustaka-pustaka ini mencakup berbagai fungsi penting yang diperlukan untuk antarmuka pengguna, pengolahan gambar, manajemen database, enkripsi data, serta pengembangan dan pelatihan model machine learning. Berikut adalah daftar pustaka yang digunakan beserta penjelasannya:

Tabel 4. 8 Inisialisasi Library

No	Nama Library	Penjelasan
1	streamlit	Mengimpor Streamlit untuk membangun antarmuka pengguna aplikasi web.
2	streamlit_option_menu	Mengimpor komponen `option_menu` dari pustaka `streamlit_option_menu` untuk membuat menu navigasi.
3	cv2	Mengimpor OpenCV untuk pengolahan gambar, terutama untuk deteksi wajah.
4	numpy	Mengimpor NumPy untuk manipulasi array.
5	sqlite3	Mengimpor SQLite untuk manajemen database.
6	uuid	Mengimpor UUID untuk menghasilkan ID unik.
7	base64	Mengimpor Base64 untuk encoding dan decoding data biner.
8	cryptography.fernet	Mengimpor Fernet dari pustaka Cryptography untuk enkripsi dan dekripsi

		data.
9	tensorflow.keras.models	Mengimpor <code>'load_model'</code> dari Keras untuk memuat model TensorFlow yang telah dilatih.
10	tensorflow.keras	Mengimpor <code>'layers'</code> dan <code>'Model'</code> dari Keras untuk membangun dan melatih model neural network.
11	tensorflow	Mengimpor TensorFlow untuk pengembangan dan pelatihan model machine learning.
12	datetime	Mengimpor modul datetime untuk menangani tanggal dan waktu.

b) Inisialisasi Library

Dalam pengembangan aplikasi deteksi ekspresi wajah ini, berbagai pustaka penting digunakan untuk memastikan aplikasi berfungsi dengan baik dan aman. Pustaka-pustaka ini mencakup alat untuk membangun antarmuka pengguna, pengolahan gambar, manajemen database, enkripsi data, serta pengembangan dan pelatihan model machine learning. Berikut adalah pustaka-pustaka yang diimpor dalam kode beserta penjelasan singkat mengenai fungsinya:

```
import streamlit as st
from streamlit_option_menu import option_menu
import cv2
import numpy as np
import sqlite3
import uuid
import base64
from cryptography.fernet import Fernet
from tensorflow.keras.models import load_model
from tensorflow.keras import layers, Model
import tensorflow as tf
import datetime
```

Gambar 4. 32 Import Library Development Web

Kode di atas mengimpor berbagai pustaka yang diperlukan untuk membangun aplikasi web interaktif ini. streamlit digunakan untuk membangun antarmuka pengguna aplikasi web, sedangkan streamlit_option_menu memungkinkan pembuatan menu navigasi yang interaktif.

c) Permodelan

Pada tahap ini, kita akan menginisialisasi model deteksi ekspresi wajah menggunakan Convolutional Neural Network (CNN). Model ini dirancang khusus untuk mendeteksi berbagai ekspresi wajah dari gambar yang diberikan. Berikut ini adalah kode untuk mendefinisikan model deteksi ekspresi wajah khusus yang dinamakan EmotionModel.

```

# Define the custom EmotionModel
class EmotionModel(Model):
    def __init__(self, activation='relu', **kwargs):
        super(EmotionModel, self).__init__(**kwargs)
        self.conv1 = layers.Conv2D(32, (3, 3), activation=activation)
        self.conv2 = layers.Conv2D(64, (3, 3), activation=activation)
        self.pool1 = layers.MaxPooling2D(pool_size=(2, 2))
        self.drop1 = layers.Dropout(0.5)

        self.conv3 = layers.Conv2D(128, (3, 3), activation=activation)
        self.pool2 = layers.MaxPooling2D(pool_size=(2, 2))
        self.conv4 = layers.Conv2D(128, (3, 3), activation=activation)
        self.pool3 = layers.MaxPooling2D(pool_size=(2, 2))
        self.drop2 = layers.Dropout(0.5)

        self.flatten = layers.Flatten()
        self.dense1 = layers.Dense(1024, activation=activation)
        self.drop3 = layers.Dropout(0.5)
        self.dense2 = layers.Dense(2, activation='softmax')

    def call(self, inputs):
        x = self.conv1(inputs)
        x = self.conv2(x)
        x = self.pool1(x)
        x = self.drop1(x)

        x = self.conv3(x)
        x = self.pool2(x)
        x = self.conv4(x)
        x = self.pool3(x)
        x = self.drop2(x)

        x = self.flatten(x)
        x = self.dense1(x)
        x = self.drop3(x)
        return self.dense2(x)

    def compute_output_shape(self, input_shape):
        return (input_shape[0], 2)

```

Gambar 4. 33 Arsitektur CNN

Kelas `EmotionModel` dirancang dengan lapisan-lapisan konvolusional, pooling, dan padat yang dioptimalkan untuk tugas klasifikasi emosi pada gambar wajah. Struktur jaringan saraf ini memungkinkan model untuk mengekstrak fitur-fitur penting dari gambar dan membuat prediksi yang akurat tentang emosi yang diekspresikan.

d) Load Model

Selanjutnya adalah memastikan bahwa *custom object* 'EmotionModel' telah terdaftar dan dapat diakses oleh Keras dengan mengupdate 'custom_objects' menggunakan 'tf.keras.utils.get_custom_objects().update({'EmotionModel' : EmotionModel'})'. Kemudian, inisialisasi model yang telah di

training sebelumnya dengan menggunakan 'load_model' dengan memberikan file model serta mendefinisikan *custom object* yang akan digunakan selama proses pemuatan model. Lalu, detektor wajah Haar Cascade diinisialisasi dengan memuat classifier Haar Cascade yang telah disediakan oleh OpenCV, khususnya 'haarcascade_frontalface_default.xml', yang digunakan untuk mendeteksi wajah dalam gambar. Kode ini mencakup tahap persiapan yang penting untuk memanfaatkan model deteksi emosi berbasis CNN dan deteksi wajah menggunakan algoritma Haar Cascade.

```
# memastikan custom object sudah terupdate
tf.keras.utils.get_custom_objects().update({'EmotionModel':
EmotionModel})

# Load pre-trained model
emotion_model = load_model('model_deteksi_ekspresi_6.h5',
custom_objects={'EmotionModel': EmotionModel})

# Load Haar cascade untuk face detection
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
```

Gambar 4. 34 Load Model

e) Database

Langkah selanjutnya adalah menghubungkan ke basis data SQLite dan memastikan struktur tabel yang dibutuhkan tersedia. Jika tabel belum ada, kode akan membuatnya dengan kolom-kolom yang diperlukan untuk menyimpan data emosi yang terkait dengan gambar, termasuk ID, tipe emosi, data gambar, dan stempel waktu. Kode ini memastikan bahwa basis data siap digunakan untuk menyimpan informasi yang relevan.

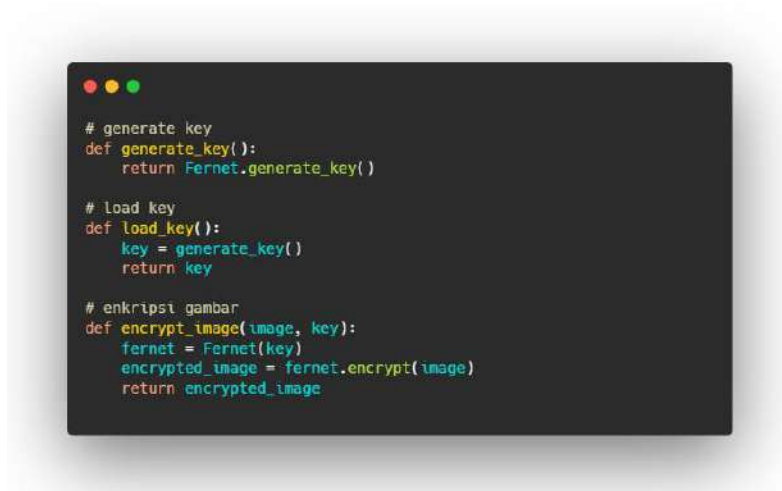
```
# fungsi untuk connect ke database
def connect_database():
    db_path = 'database_emotion.sqlite'
    return sqlite3.connect(db_path)

# memastikan tabel ada di database
def create_table(conn):
    c = conn.cursor()
    c.execute('''CREATE TABLE IF NOT EXISTS emotion_data (
                id TEXT PRIMARY KEY,
                tipe TEXT,
                gambar BLOB,
                timestamp TEXT)''')
    conn.commit()
```

Gambar 4. 35 Database Setup

f) Enkripsi Simetris Menggunakan Fernet

Setelah melakukan *setup database*, selanjutnya membuat skema enkripsi simetris menggunakan Fernet untuk menghasilkan kunci enkripsi, memuat kunci tersebut, dan mengenkripsi gambar menggunakan kunci tersebut. Pertama, fungsi untuk menghasilkan kunci enkripsi menggunakan `Fernet.generate_key()`. Selanjutnya, kunci ini dimuat oleh fungsi yang memanggil fungsi sebelumnya untuk menghasilkan kunci baru setiap kali dipanggil. Terakhir, gambar dienkripsi dengan fungsi yang menggunakan objek Fernet dan kunci yang telah dihasilkan untuk mengenkripsi data gambar. Kode ini memastikan bahwa gambar yang akan disimpan atau ditransmisikan dapat dienkripsi untuk menjaga kerahasiaan dan keamanan data.



Gambar 4. 36 Enkripsi Simetris Menggunakan Fernet

g) Fungsi Deteksi

Dalam fungsi deteksi ini penulis membuatnya menjadi beberapa *function*.

1) Predict_emotion() dan save_to_database()

Pada bagian ini terdiri dari dua fungsi yang berperan penting dalam sistem deteksi emosi. Fungsi pertama, `predict_emotion`, menerima gambar wajah dan model prediktif sebagai input. Gambar wajah diubah ukurannya menjadi 48x48 piksel, dinormalisasi, dan disesuaikan dimensinya agar sesuai dengan format input model. Model kemudian memprediksi emosi berdasarkan gambar yang diproses, dan hasil prediksi yang berupa probabilitas untuk setiap kelas emosi dikonversi menjadi kelas dengan probabilitas tertinggi. Fungsi kedua, `save_to_database`, menerima gambar yang telah diprediksi, label hasil prediksi, koneksi ke basis data, dan kunci enkripsi sebagai input. Gambar dikonversi ke format JPEG dan dienkripsi menggunakan kunci enkripsi. Data yang telah dienkripsi, bersama dengan label, ID unik, dan stempel waktu, kemudian disimpan ke dalam basis data. Label

yang disimpan berupa "Tidak Mencurigakan" untuk label 1 dan "Mencurigakan" untuk label lainnya. Proses ini memastikan bahwa hasil prediksi emosi tersimpan dengan aman dan terstruktur dalam basis data.

```
# fungsi untuk predict ekspresi / emosi
def predict_emotion(face_image, model):
    face_image = cv2.resize(face_image, (48, 48))
    face_image = face_image.astype('float32') / 255.0
    face_image = np.expand_dims(face_image, axis=0)
    face_image = np.expand_dims(face_image, axis=-1)
    prediction = model.predict(face_image)
    return np.argmax(prediction)

# fungsi untuk menyimpan gambar hasil deteksi ke database
def save_to_database(image, label, conn, key):
    c = conn.cursor()
    _, buffer = cv2.imencode('.jpg', image)
    image_blob = buffer.tobytes()
    encrypted_image = encrypt_image(image_blob, key)
    label_text = "Tidak Mencurigakan" if label == 1 else "Mencurigakan"
    record_id = str(uuid.uuid4())
    timestamp = datetime.datetime.now().isoformat()
    c.execute('INSERT INTO emotion_data (id, tipe, gambar, timestamp)
VALUES (?, ?, ?, ?)',
            (record_id, label_text, encrypted_image, timestamp))
    conn.commit()
```

Gambar 4. 37 Fungsi predict_emotion dan save_to_database

2) Detection()

Fungsi detection() ini menggunakan Streamlit untuk membuat antarmuka web yang mendeteksi ekspresi wajah yang berpotensi mencurigakan melalui kamera atau gambar yang diunggah. Pada antarmuka web, terdapat tombol untuk memulai dan menghentikan kamera. Jika kamera dimulai, video dari kamera ditangkap dan wajah-wajah dalam frame dideteksi menggunakan Haar Cascade. Setiap wajah yang terdeteksi diprediksi ekspresinya menggunakan model prediksi emosi. Hasil prediksi kemudian disimpan dalam basis data setelah dienkripsi. Wajah yang dianggap mencurigakan diberi kotak merah, sedangkan yang tidak mencurigakan diberi kotak hijau.

Jumlah wajah mencurigakan dan tidak mencurigakan ditampilkan pada frame video. Pengguna juga dapat mengunggah gambar untuk dianalisis, dan hasil deteksi ditampilkan pada gambar yang diunggah. Hasil deteksi kemudian ditampilkan pada halaman web, menunjukkan apakah wajah yang terdeteksi mencurigakan atau tidak.

```
def detection():
    st.title("Deteksi Ekspresi Potensi Kejahatan")
    # Generate and load the encryption key
    key = load_key()

    # Align Start and Stop buttons horizontally
    col1, col2 = st.columns(2)
    start_button = col1.button("Start Camera")
    stop_button = col2.button("Stop Camera")

    if start_button:
        cap = cv2.VideoCapture(0)
        conn = connect_database()
        create_table(conn)
        stframe = st.empty()

        while True:
            ret, frame = cap.read()
            if not ret:
                break

            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray, 1.3, 5)

            suspicious_count = 0
            non_suspicious_count = 0

            for (x, y, w, h) in faces:
                face = gray[y:y+h, x:x+w]
                label = predict_emotion(face, emotion_model)
                save_to_database(face, label, conn, key)

                if label == 0: # "Tidak Mencurigakan" category
                    color = (0, 255, 0) # green
                    non_suspicious_count += 1
                else: # "Mencurigakan" category
                    color = (0, 0, 255) # red
                    suspicious_count += 1

                cv2.rectangle(frame, (x, y), (x+w, y+h), color, 2)

            cv2.putText(frame, f'Mencurigakan: {suspicious_count}', (10, 30),
            cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
            cv2.putText(frame, f'Tidak Mencurigakan: {non_suspicious_count}', (10, 70),
            cv2.FONT_HERSHEY_SIMPLEX, 1,
            (0, 255, 0), 2)

            stframe.image(frame, channel="BGR")

            if stop_button:
                break

        cap.release()
        conn.close()

    upload_image = st.file_uploader("Upload Gambar untuk Deteksi Dixin")

    if upload_image:
        file_bytes = np.asarray(bytearray(upload_image.read()), dtype=np.uint8)
        image = cv2.imdecode(file_bytes, 1)
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)

        suspicious_count = 0
        non_suspicious_count = 0

        for (x, y, w, h) in faces:
            face = gray[y:y+h, x:x+w]
            label = predict_emotion(face, emotion_model)

            if label == 0:
                color = (0, 255, 0)
                non_suspicious_count += 1
            else:
                color = (0, 0, 255)
                suspicious_count += 1

            cv2.rectangle(image, (x, y), (x+w, y+h), color, 2)

        st.image(image, channel="BGR")

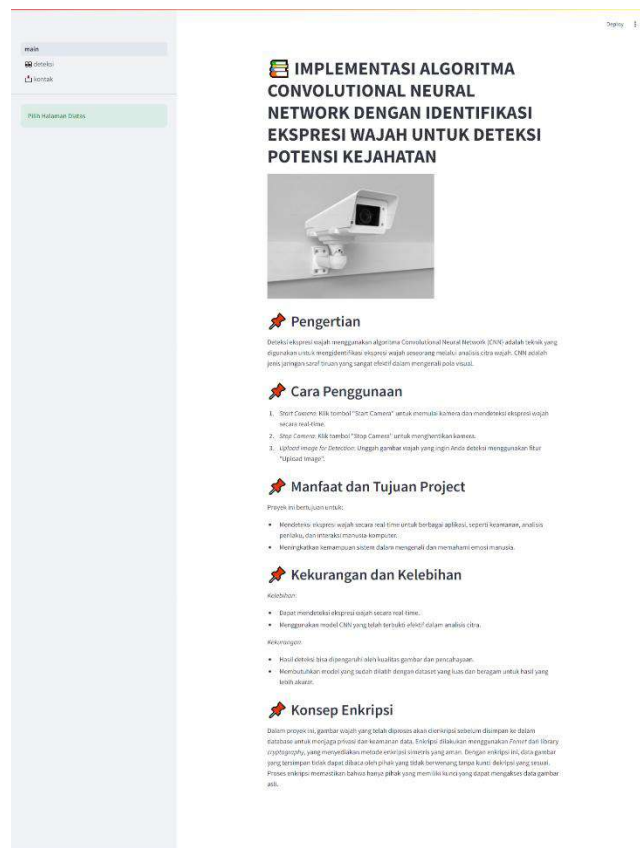
        # Display the detection result
        result_text = "Tidak Mencurigakan" if non_suspicious_count > 0 else "Mencurigakan"
        st.write(f'Detected expression: {result_text}')
```

Gambar 4. 38 Detection Function

e. Hasil website

1. Halaman Home

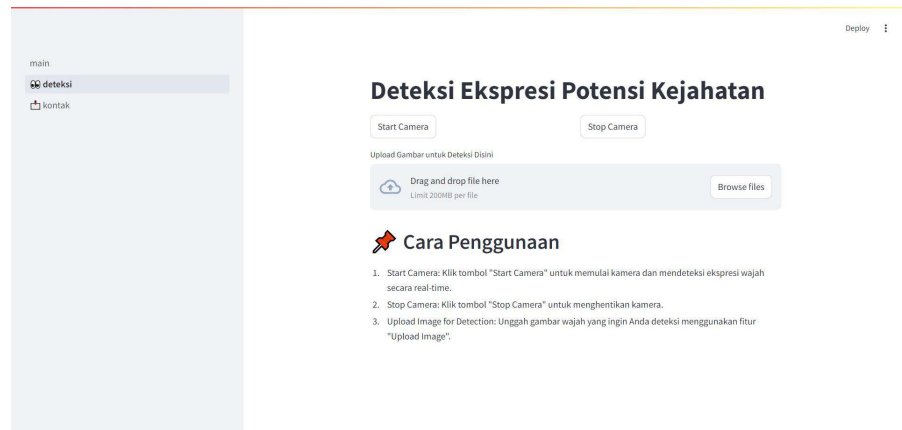
Halaman ini menyediakan informasi tentang pengertian proyek, cara penggunaan, manfaat dan tujuan proyek, serta kekurangan dan kelebihan. Pengguna dapat memulai kamera untuk mendeteksi ekspresi wajah secara real-time, menghentikan kamera, atau mengunggah gambar untuk analisis. Selain itu, halaman ini juga menjelaskan konsep enkripsi yang digunakan untuk menjaga privasi dan keamanan data gambar yang diproses. Dengan antarmuka yang intuitif dan informatif, halaman ini memudahkan pengguna memahami dan memanfaatkan sistem deteksi ekspresi wajah yang telah dikembangkan.



Gambar 4. 39 Hasil Website : Homepage

2. Halaman Deteksi

Halaman ini adalah bagian dari antarmuka web untuk proyek deteksi ekspresi wajah yang berpotensi mencurigakan. Pada halaman ini, pengguna dapat mengakses fitur untuk memulai dan menghentikan kamera, yang memungkinkan deteksi ekspresi wajah secara real-time. Pengguna juga memiliki opsi untuk mengunggah gambar melalui fitur "Upload Image" untuk analisis deteksi ekspresi. Instruksi penggunaan dijelaskan dengan jelas di bawah bagian "Cara Penggunaan", mencakup langkah-langkah untuk memulai kamera, menghentikan kamera, dan mengunggah gambar untuk deteksi. Halaman ini dirancang untuk memberikan pengalaman pengguna yang intuitif dan memudahkan dalam menjalankan dan memahami fungsi deteksi ekspresi wajah dalam proyek ini.



Gambar 4. 40 Hasil Website : Detection Page

3. Halaman Kontak

Halaman ini berfungsi sebagai halaman kontak dalam antarmuka web proyek deteksi ekspresi wajah. Halaman ini menampilkan informasi kontak dari pengembang atau penanggung jawab proyek, termasuk nama, Nomor Pokok Mahasiswa (NPM), dan alamat email. Terdapat juga foto

penanggung jawab untuk memberikan identifikasi visual. Informasi ini memudahkan pengguna yang ingin menghubungi pengembang terkait dengan pertanyaan, saran, atau masukan mengenai proyek deteksi ekspresi wajah. Halaman ini dirancang untuk memberikan akses mudah dan cepat kepada pengguna yang membutuhkan dukungan atau informasi lebih lanjut.



Gambar 4. 41 Hasil Website : Kontak

f. Menguji System

1. *White Box*

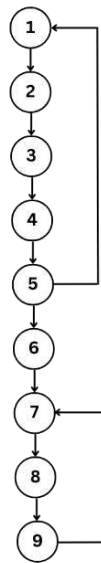
White Box Testing adalah metode pengujian perangkat lunak di mana penguji meneliti apakah kode yang dibuat terdapat kesalahan atau tidak dan struktur internal dari aplikasi yang diuji. Berikut hasil pengujian white box dapat dilihat pada Tabel 4.6

Tabel 4. 9 White Box Testing

Node	Code	Pengertian
1.	<code>conn = connect_database()</code>	Menghubungkan ke basis data SQLite
2.	<code>key = generate_key()</code>	Menghasilkan kunci enkripsi menggunakan Fernet
3.	<code>encrypted_image = encrypt_image(image_blob, key)</code>	Mengenkripsi gambar menggunakan kunci yang dihasilkan
4.	<code>label = predict_emotion(face_image,</code>	Memprediksi

	<code>emotion_model)</code>	ekspresi wajah menggunakan model CNN
5.	<code>save_to_database(face, label, conn, key)</code>	Menyimpan gambar hasil deteksi yang telah dienkripsi ke dalam basis data
6.	<code>cap = cv2.VideoCapture(0)</code>	Memulai kamera untuk mendeteksi ekspresi wajah secara real-time
7.	<code>st.file_uploader("Upload Gambar untuk Deteksi Disini")</code>	Form untuk mengunggah gambar untuk deteksi ekspresi wajah
8.	<code>st.image(image, channels="BGR")</code>	Menampilkan gambar yang diunggah pada halaman Streamlit
9.	<code>st.write("Detected expression: {result_text}")</code>	Menampilkan hasil deteksi ekspresi wajah pada halaman Streamlit

- a) Membuat *Flowgraph* dari program berikut adalah gambaran *Flowgraph* dari script penilaian diatas dapat dilihat pada Gambar



Gambar 4. 42 Flowgraph

b) Menghitung Kompleksitas Siklomatik

Kompleksitas siklomatik adalah metrik yang digunakan untuk mengukur kompleksitas struktural dari sebuah program.

Rumus untuk menghitung kompleksitas siklomatik:

$$V(G) = E - N + 2$$

Dimana:

- E adalah jumlah edge (garis alur dalam flow graph)
- N adalah jumlah node (titik keputusan dalam flow graph)

Berikut adalah perhitungan Kompleksitas Siklomatik diperoleh dengan Perhitungan berikut:

Nodes: 9

Edges: 10

Menggunakan rumus kompleksitas siklomatik:

$$V(G) = E - N + 2 = 10 - 9 + 2 = 3$$

Dari hasil perhitungan diatas dapat ditentukan independent path sebagai berikut Jalur Independen. Terdapat 2 independent path

yang diperoleh. Berikut dibawah ini :

Path 1: 1 → 2 → 3 → 4 → 5 → 1

Path 2: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 7

Path 3: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 1

2. *Black Box*

Black box testing digunakan untuk mengevaluasi kinerja aplikasi tanpa memeriksa struktur internalnya, dengan tujuan memastikan aplikasi berfungsi sesuai harapan pengguna. Proses pengujian ini terdiri dari tiga tahap utama: perencanaan, pelaksanaan, dan evaluasi. Tiga dosen melakukan pengujian ini, dengan fokus utama untuk menilai sejauh mana aplikasi memenuhi fungsionalitas yang diinginkan oleh pengguna akhir.

a) *Form Black Box Testing*

Tabel 4. 10 Form Black Box Testing

Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan					
				Diterima	Perbaikan				
Halaman Home	User dapat melakukan gulir atau scroll ke bawah	User dapat melihat semua fitur yang ada pada halaman home (page prediksi, form upload)	Website akan menampilkan semua fitur yang ada pada halaman home	√	√	√			
Koneksi ke Basis Data	Aplikasi dapat terhubung ke basis data	Koneksi berhasil tanpa error	Aplikasi berhasil terhubung ke basis data	√	√	√			
Membuat	Tabel	Tabel	Tabel	√	√	√			

Tabel di Basis Data	`emotion_data` dibuat di basis data	`emotion_data` ada di basis data	berhasil dibuat di basis data						
Menghasilkan Kunci Enkripsi	Kunci enkripsi dihasilkan	Kunci enkripsi dihasilkan tanpa error	Kunci enkripsi berhasil dihasilkan	√	√	√			
Mengenkripsi Gambar	Gambar dienkripsi dengan benar	Gambar berhasil dienkripsi	Gambar berhasil dienkripsi tanpa error	√	√	√			
Prediksi Ekspresi Wajah	Sistem dapat memprediksi ekspresi wajah	Ekspresi wajah berhasil diprediksi dan ditampilkan	Ekspresi wajah berhasil diprediksi	√	√	√			
Menyimpan Hasil Deteksi ke Basis Data	Hasil deteksi disimpan di basis data	Hasil deteksi disimpan tanpa error	Hasil deteksi berhasil disimpan di basis data	√	√	√			
Menampilkan Gambar yang Diunggah	Gambar yang diunggah ditampilkan di halaman web	Gambar ditampilkan tanpa error	Gambar berhasil ditampilkan di halaman web	√	√	√			
Memulai Kamera untuk Deteksi Real-time	Kamera mulai dan menampilkan video secara	Video real-time ditampilkan tanpa error	Kamera berhasil menampilkan video real-	√	√	√			

	real-time		time						
Mengunggah Gambar untuk Deteksi	Gambar diterima dan diproses untuk deteksi	Gambar diterima dan diproses tanpa error	Gambar berhasil diproses untuk deteksi	√	√	√			
Menampilkan Hasil Deteksi	Hasil deteksi ditampilkan di halaman web	Hasil deteksi ditampilkan tanpa error	Hasil deteksi berhasil ditampilkan	√	√	√			

b) *Hasil Black Box Testing*

Berdasarkan hasil pengujian *black box* yang sudah dilakukan dari 11 *Test Case website* yang sudah dibuat di ujikan kepada 3 dosen informatika berikut adalah hasilnya:

1) Pengujian pertama

$$\text{Tercapai} : \frac{11}{11} \times 100 \% = 100 \%$$

$$\text{Gagal} : \frac{0}{11} \times 100 \% = 0 \%$$

2) Pengujian kedua

$$\text{Tercapai} : \frac{11}{11} \times 100 \% = 100 \%$$

$$\text{Gagal} : \frac{0}{11} \times 100 \% = 0 \%$$

3) Pengujian ketiga

$$\text{Tercapai} : \frac{11}{11} \times 100 \% = 100 \%$$

$$\text{Gagal} : \frac{0}{11} \times 100 \% = 0 \%$$

$$\text{Jumlah presentase rata-rata tercapai} = \frac{300\%}{3} = 100\%$$

$$\text{Jumlah presentase rata-rata gagal} = \frac{0\%}{3} = 0\%$$

g. Evaluasi *System Usability Scale (SUS)*

System Usability Scale (SUS) adalah metode yang digunakan untuk mengevaluasi kegunaan dari suatu sistem atau aplikasi. SUS memberikan cara sederhana dan cepat untuk menilai sejauh mana pengguna merasa aplikasi mudah digunakan. Evaluasi SUS dilakukan dengan memberikan sejumlah pertanyaan kepada pengguna terkait pengalaman mereka dalam menggunakan aplikasi. Hasil dari evaluasi ini akan membantu pengembang memahami aspek-aspek mana yang perlu diperbaiki untuk meningkatkan kegunaan dan kepuasan pengguna secara keseluruhan.

Tabel 4. 11 Form *System Usability Scale (SUS)*

No	Pernyataan	Sangat Tidak Setuju (1)	Tidak Setuju (2)	Netral (3)	Setuju (4)	Sangat Setuju (5)
1	Saya merasa bahwa saya akan sering menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..]
2	Saya merasa sistem ini tidak perlu rumit.	[..]	[..]	[..]	[..]	[..]
3	Saya merasa bahwa sistem ini mudah	[..]	[..]	[..]	[..]	[..]

	digunakan.					
4	Saya pikir saya akan membutuhkan bantuan dari seseorang yang ahli untuk dapat menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..]
5	Saya merasa bahwa berbagai fungsi dalam sistem ini terintegrasi dengan baik.	[..]	[..]	[..]	[..]	[..]
6	Saya merasa bahwa sistem ini memiliki terlalu banyak inkonsistensi.	[..]	[..]	[..]	[..]	[..]
7	Saya merasa bahwa kebanyakan orang akan cepat mempelajari cara menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..]

8	Saya merasa bahwa sistem ini sangat membingungkan ketika digunakan.	[..]	[..]	[..]	[..]	[..]
9	Saya merasa percaya diri saat menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..]
10	Saya harus mempelajari banyak hal sebelum bisa mulai menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..]

Berikut ini merupakan hasil kuesioner pengujian System Usability Scale yang telah disebarakan kepada 5 responden. Hasil evaluasi dari *System Usability Scale* dapat dilihat pada Tabel 4.12.

Tabel 4. 12 Hasil Evaluasi SUS

pertanyaan	Hasil Pengujian				
	Responden 1	Responden 2	Responden 3	Responden 4	Responden 5
1	4	5	5	5	5
2	2	2	1	2	2
3	5	5	5	5	5

4	2	1	1	1	3
5	5	5	4	5	5
6	1	1	1	1	1
7	5	4	4	5	5
8	1	3	1	1	1
9	5	5	4	5	5
10	1	3	1	4	2
Jumlah Skor SUS	92.5	85	92.5	90	90
Nilai Rata Rata	90				

Dari perhitungan tersebut, diperoleh nilai rata-rata sebesar 90. Dengan demikian, dapat disimpulkan bahwa pengujian SUS pada aplikasi ini mendapat kategori "Excellent".

B. Pembahasan

Dalam bagian ini, peneliti menguraikan temuan-temuan utama, menghubungkannya dengan hipotesis awal, serta membandingkannya dengan hasil penelitian sebelumnya. Pembahasan juga mencakup penjelasan tentang implikasi temuan penelitian, baik dalam konteks teori maupun praktik. Selain itu, bagian ini dapat mengidentifikasi keterbatasan penelitian dan memberikan saran untuk penelitian lebih lanjut. Dengan demikian, pembahasan membantu memperjelas signifikansi hasil penelitian dan memberikan pandangan yang lebih mendalam tentang topik yang diteliti.

1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah langkah pertama dalam pengembangan suatu sistem atau proyek. Dalam pengembangan aplikasi deteksi ekspresi potensi kejahatan, diperlukan perangkat keras dengan spesifikasi minimum yang mendukung. Perangkat keras aplikasi backend memerlukan prosesor Intel® Core™ i7-10750H CPU @ 2.60 GHz (12

CPUs), RAM 16GB, SSD 256 GB, dan VGA NVIDIA GeForce GTX 1650Ti. Sementara itu, perangkat keras aplikasi frontend memerlukan sistem operasi Windows 64 bit, RAM minimal 4GB, ruang hardisk minimal 4GB yang tersedia, dan resolusi layar 1280 x 800. Spesifikasi ini memastikan bahwa perangkat keras dapat mendukung proses pengembangan dan operasional aplikasi secara optimal.

Selain perangkat keras, perangkat lunak yang digunakan juga harus memenuhi kebutuhan pengembangan aplikasi deteksi ekspresi potensi kejahatan. Perangkat lunak minimum yang diperlukan mencakup Jupyter Notebook untuk pengembangan dan eksperimen machine learning, PyCharm sebagai Integrated Development Environment (IDE) untuk pengkodean dan debugging, Figma untuk desain UI/UX, serta Kaggle untuk dataset dan eksperimen machine learning. Kombinasi perangkat keras dan perangkat lunak ini memastikan bahwa pengembangan aplikasi dapat berjalan dengan lancar dan efisien, serta memastikan setiap aspek sistem berfungsi optimal dalam mendukung aplikasi deteksi ekspresi potensi kejahatan.

2. Implementasi Model Deteksi Ekspresi Potensi Kejahatan

Pengembangan aplikasi deteksi ekspresi wajah menggunakan Convolutional Neural Network (CNN) memerlukan tahapan yang terstruktur, mulai dari pemrosesan gambar hingga evaluasi model. Dalam penelitian ini, dataset yang digunakan adalah FER-2013, yang terdiri dari 28.709 data latih dan 7.178 data uji. Gambar-gambar dalam dataset ini diubah menjadi format grayscale dan diresize menjadi ukuran 48x48 piksel untuk memastikan konsistensi input pada model. Dataset kemudian dimuat dan dibagi menjadi dua kelas utama, yaitu 'Mencurigakan' (yang mencakup ekspresi marah, jijik, takut, terkejut, dan sedih) dan 'Tidak Mencurigakan' (yang mencakup ekspresi bahagia dan netral). Jumlah data latih dan uji untuk masing-masing kelas adalah sebagai berikut: Marah (3995 data latih, 958 data uji), Jijik (436 data latih, 111 data uji), Takut (4097 data latih, 1024 data uji), Terkejut (3171 data latih, 831 data uji),

Sedih (4830 data latih, 1247 data uji), Bahagia (7215 data latih, 1774 data uji), dan Netral (4965 data latih, 1233 data uji). Model CNN yang digunakan terdiri dari beberapa lapisan konvolusi dan pooling, dengan fungsi aktivasi ReLU dan dropout untuk mengurangi overfitting. Model ini dilatih menggunakan optimizer Adam dengan learning rate yang diatur oleh ExponentialDecay, dan metrik yang digunakan adalah categorical_crossentropy dan akurasi.

Setelah pelatihan model, evaluasi dilakukan untuk menilai kinerja model pada data uji. Prediksi model dibandingkan dengan label sebenarnya untuk menghitung matriks kebingungan, yang kemudian divisualisasikan menggunakan heatmap. Selain itu, laporan klasifikasi yang mencakup precision, recall, f1-score, dan support dihitung dan divisualisasikan dalam bentuk tabel. Untuk menilai kemampuan model dalam membedakan antara kelas, kurva ROC dan nilai AUC dihitung dan diplot. Hasil evaluasi menunjukkan kinerja model yang cukup baik dengan nilai AUC yang tinggi, mengindikasikan bahwa model memiliki kemampuan yang baik dalam mendeteksi ekspresi wajah yang mencurigakan dan tidak mencurigakan. Dengan jumlah data latih dan uji yang besar serta distribusi data yang memadai dalam dataset FER-2013, model ini diharapkan mampu generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

3. Implementasi Sistem

Setelah melatih model deteksi ekspresi wajah menggunakan Convolutional Neural Network (CNN), tahap berikutnya adalah implementasi sistem. Implementasi ini melibatkan pengumpulan kebutuhan sistem yang mencakup kebutuhan fungsional dan desain Unified Modelling Language (UML). Diagram Use Case menggambarkan interaksi antara pengguna dan sistem, sementara Diagram Aktivitas (Activity Diagram) menunjukkan alur kerja dalam aplikasi dari awal hingga akhir. Aplikasi ini memungkinkan pengguna untuk mendeteksi ekspresi wajah melalui gambar yang diunggah atau kamera real-time

dengan tombol start dan stop untuk mengoperasikan kamera. Halaman Home memberikan informasi tentang proyek, langkah-langkah penggunaan, serta kelebihan dan kekurangan proyek, sementara halaman Deteksi menyediakan fitur utama untuk deteksi wajah melalui kamera atau gambar yang diunggah.

Prototipe aplikasi dibangun berdasarkan kebutuhan yang telah dikumpulkan, dengan desain UI/UX yang memastikan antarmuka yang mudah digunakan dan menarik. Evaluasi prototipe dilakukan untuk memastikan hasil akhir sesuai dengan harapan dan kebutuhan klien. Halaman Home menjelaskan proyek, manfaat, dan keamanan data, halaman Deteksi memungkinkan deteksi real-time dan unggahan gambar, serta halaman Kontak memberikan informasi kontak pengembang. Sistem diuji menggunakan metode White Box dan Black Box Testing untuk memastikan fungsionalitas dan struktur internal aplikasi. White Box Testing mengevaluasi kode dan alur logika, sementara Black Box Testing menilai apakah aplikasi berfungsi sesuai dengan kebutuhan pengguna akhir dengan hasil pengujian mencapai 100% kelulusan dalam semua test case.

Evaluasi System Usability Scale (SUS) dilakukan untuk menilai kegunaan aplikasi, dengan hasil rata-rata skor SUS sebesar 90 yang termasuk dalam kategori "Excellent". SUS memberikan cara sederhana dan cepat untuk menilai sejauh mana pengguna merasa aplikasi mudah digunakan. Hasil ini menunjukkan bahwa aplikasi deteksi ekspresi wajah yang dikembangkan memiliki tingkat kegunaan yang sangat baik dan memenuhi harapan pengguna. Aplikasi ini memungkinkan pengguna untuk mendeteksi ekspresi wajah dengan mudah melalui antarmuka yang intuitif dan fitur yang berfungsi dengan baik, menjadikannya alat yang efektif dalam analisis ekspresi wajah.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Dari hasil proses pelatihan model untuk deteksi ekspresi wajah menggunakan metode Convolutional Neural Network (CNN) menghasilkan akurasi sebesar 97.42% dan validation accuracy sebesar 80.64%. Pelatihan model ini menggunakan dataset FER-2013 dengan 28.709 data pelatihan dan 7.178 data pengujian. Model CNN yang digunakan mampu mendeteksi ekspresi mencurigakan dan tidak Mencurigakan berdasarkan tujuh jenis ekspresi wajah: marah, jijik, takut, bahagia, netral, sedih, dan terkejut.
2. Penelitian ini menggunakan tiga tahapan pengujian pada pengembangan aplikasi, yaitu pengujian black box, pengujian white box, dan *system usability scale* yang menghasilkan hasil yang baik. Pada pengujian black box, didapatkan presentase keberhasilan sebesar 100% dengan tidak ada pengujian yang gagal dari 10 skenario pengujian. Pengujian white box juga mencapai keberhasilan 100%, memastikan bahwa semua alur logika dan fungsionalitas kode bekerja dengan baik. Pengujian *system usability scale* mendapatkan hasil baik dengan nilai rata rata sebesar 90 dari 5 responden dengan 10 pertanyaan, menunjukkan bahwa pengguna akhir puas dengan kinerja dan fungsionalitas aplikasi.
3. Penerapan metode Convolutional Neural Network terbukti berhasil dalam mendeteksi ekspresi wajah untuk deteksi potensi kejahatan. Uji coba menunjukkan bahwa model CNN memiliki tingkat akurasi yang tinggi dalam mengidentifikasi ekspresi mencurigakan dan tidak mencurigakan.
4. Setelah dilakukan komparasi antara algoritma *Convolutional Neural Network* (CNN) dengan metode transfer learning menggunakan model

RestNet50, *MobileNetV2*, dan *VGG16* dalam identifikasi ekspresi wajah yang berpotensi mendeteksi kejahatan, hasil menunjukkan bahwa CNN tradisional menghasilkan akurasi yang tinggi pada data pelatihan. Namun, metode transfer learning seperti *RestNet50*, *MobileNetV2*, dan *VGG16* dinilai tidak efektif dalam melatih model pada konteks ini. Oleh karena itu, penelitian ini pada akhirnya menggunakan arsitektur CNN tradisional sebagai pendekatan utama, karena terbukti memberikan hasil yang lebih baik dan stabil dalam mendeteksi ekspresi wajah yang mencurigakan. Kesimpulan ini mendukung penggunaan CNN tradisional sebagai solusi yang lebih handal untuk aplikasi pengenalan wajah dalam konteks keamanan.

B. Saran

1. Meningkatkan usability: disarankan untuk meningkatkan kegunaan aplikasi dengan menambahkan fitur display database yang memungkinkan pengguna untuk melihat dan mengelola data yang telah dikumpulkan. Selain itu, implementasi sistem peran (role) yang lebih terstruktur menjadi tiga kategori, yaitu user, admin, dan developer, dapat membantu dalam membagi tanggung jawab dan aksesibilitas berdasarkan kebutuhan dan tingkat kepentingan masing-masing peran. User dapat menggunakan aplikasi untuk tujuan dasar, admin dapat mengelola dan memelihara data, serta developer dapat mengembangkan dan memperbaiki sistem.
2. Penerapan scheduler: untuk memastikan bahwa model selalu up-to-date dengan data terbaru, disarankan untuk menerapkan scheduler yang memungkinkan pelatihan model secara berkala. Dengan menggunakan data terbaru, model dapat terus belajar dan menyesuaikan diri dengan pola ekspresi wajah yang mungkin berubah dari waktu ke waktu. Scheduler ini dapat diatur untuk melakukan pelatihan ulang model pada interval waktu tertentu, seperti mingguan atau bulanan, tergantung pada kebutuhan dan ketersediaan data.
3. Pengembangan lebih lanjut: diperlukan penelitian lanjutan untuk meningkatkan akurasi model dengan menambahkan lebih banyak data

pelatihan dan menggunakan teknik augmentasi data. Selain itu, integrasi model dengan sistem keamanan yang lebih kompleks, seperti analisis perilaku dan pengenalan pola, dapat meningkatkan efektivitas deteksi.

4. Keamanan dan privasi: implementasi langkah-langkah keamanan siber yang lebih ketat diperlukan untuk melindungi sistem dari ancaman eksternal. Penggunaan enkripsi data pada semua tahap pengolahan dan penyimpanan data sangat disarankan untuk memastikan privasi dan keamanan informasi yang dikumpulkan.
5. Pengujian di lingkungan nyata: disarankan untuk melakukan uji coba di lingkungan nyata dengan kamera CCTV yang digunakan di tempat-tempat umum. Hal ini akan membantu dalam menilai kinerja sistem dalam kondisi sebenarnya dan mengidentifikasi area yang memerlukan perbaikan lebih lanjut.

DAFTAR PUSTAKA

- [1] A. K. Muttaqin, H. Yuana, and M. T. Chulkamdi, "Implementasi Algoritma Convolutional Neural Network Untuk Pengenalan Ekspresi Wajah," *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, vol. 8, no. 2, pp. 772–792, 2023, [Online]. Available: <https://tunasbangsa.ac.id/ejurnal/index.php/jurasik>
- [2] N. Shah, N. Bhagat, and M. Shah, "Crime forecasting: a machine learning and computer vision approach to crime prediction and prevention," Dec. 01, 2021, *Springer*. doi: 10.1186/s42492-021-00075-z.
- [3] K. Ahir, K. Govani, R. Gajera, and M. Shah, "Application on Virtual Reality for Enhanced Education Learning, Military Training and Sports," *Augmented Human Research*, vol. 5, no. 1, Dec. 2020, doi: 10.1007/s41133-019-0025-2.
- [4] E. D. Madyatmadja, M. N. Ridho, A. R. Pratama, M. Fajri, and L. Novianto, "Penerapan Visualisasi Data Terhadap Klasifikasi Tindak Kriminal Di Indonesia," *Infotech: Journal of Technology Information*, vol. 8, no. 1, pp. 61–68, Jun. 2022, doi: 10.37365/jti.v8i1.127.
- [5] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going Deeper in Facial Expression Recognition using Deep Neural Networks," 2016.
- [6] C. A. Corneanu, M. O. Simón, J. F. Cohn, and S. E. Guerrero, "Survey on RGB, 3D, Thermal, and Multimodal Approaches for Facial Expression Recognition: History, Trends, and Affect-Related Applications," *IEEE Trans Pattern Anal Mach Intell*, vol. 38, no. 8, pp. 1548–1568, Aug. 2016, doi: 10.1109/TPAMI.2016.2515606.
- [7] A. Fadlil, D. Prayogi, A. Dahlan, and Y. Penulis Korespondensi, "Sistem Pengenalan Wajah pada Keamanan Ruang Berbasis Convolutional Neural Network," 2022.
- [8] P. Adi Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia,"

Jurnal Algor, vol. 2, no. 1, 2020, [Online]. Available:
<https://jurnal.buddhidharma.ac.id/index.php/algor/index>

- [9] Andi Agustinus, Rudi Kurniawan, and Harma Oktafia Lingga Wijaya, *Klasifikasi Emosi Melalui Ekspresi Wajah Menggunakan Algoritma Deep Learning*, vol. 2. Built Environment Research Unit, University of Wolverhampton, 2023.
- [10] Y. Nathasya and B. Ginting, “Deteksi Emosi Anak Dari Ekspresi Wajah Dengan Deep Learning Untuk Menilai Kesehatan Mental,” 2023.
- [11] M. F. Sanner, “Python: A Programming Language For Software Integration AndDevelopment,”1999.[Online].Available:
<http://www.python.org/doc/Comparisons.html>
- [12] Q. Hu, L. Ma, and J. Zhao, “DeepGraph: A PyCharm Tool for Visualizing and Understanding Deep Learning Models,” in *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, IEEE Computer Society, Jul. 2018, pp. 628–632. doi: 10.1109/APSEC.2018.00079.
- [13] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” May 27, 2015, *Nature Publishing Group*. doi: 10.1038/nature14539.
- [14] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, “Efficient Processing of Deep Neural Networks: A Tutorial and Survey,” Dec. 01, 2017, *Institute of Electrical and Electronics Engineers Inc*. doi: 10.1109/JPROC.2017.2761740.
- [15] P. H. . Sydenham and Richard. Thorn, *Artificial Neural Networks*. Wiley, 2005.
- [16] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognit*, vol. 77, pp. 354–377,May2018,doi: 10.1016/j.patcog.2017.10.013.
- [17] A. Fany Achmalia, S. Walid, I. Artikel, and S. Artikel, “Peramalan Penjualan Semen Menggunakan Backpropagation Neural Network Dan Recurrent NeuralNetwork,”2020.[Online].Available:
<http://journal.unnes.ac.id/sju/index.php/ujm>

- [18] E. Hermawan, “Klasifikasi Pengenalan Wajah Menggunakan Masker atau Tidak Dengan Mengimplementasikan Metode CNN (Convolutional Neural Network),” 2021.
- [19] A. Berliano Handoko, I. K. Timotius, D. Utomo, P. Studi Teknik Komputer, and U. Kristen Satya Wacana, “Klasifikasi Citra X-Ray COVID-19 Menggunakan Three-layered CNN Model,” Salatiga, Mar. 2022.
- [20] F. Fitra Maulana and N. Rochmawati, “Klasifikasi Citra Buah Menggunakan Convolutional Neural Network,” *JINACS*, vol. 1, no. 2, pp. 104–108, 2019.
- [21] H. Nuraliza, O. N. Pratiwi, and F. Hamami, “Analisis Sentimen IMDb Film Review Dataset Menggunakan Support Vector Machine (SVM) dan Seleksi Feature Importance,” 2022.
- [22] A. Rahmat, M. Syafiih, and M. Faid, “Implementasi Klasifikasi Potensi Penyakit Jantung Dengan Menggunakan Metode C4.5 Berbasis Website (Studi Kasus Kaggle.Com),” *INFOTECH journal*, vol. 9, no. 2, pp. 393–400, Jul. 2023, doi: 10.31949/infotech.v9i2.6295.
- [23] G. E. P. Purba, S. Hadi Wijoyo, and N. Y. Setiawan, “Pengaruh Transfer Learning Resnet Dan Densenet Terhadap Performa Klasifikasi Ekspresi Wajah Menggunakan Dataset Fer-2013,” 2017. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [24] M. Arsal, B. Agus Wardijono, and D. Anggraini, “Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN,” *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 6, no. 1, pp. 55–63, Jun. 2020, doi: 10.25077/teknosi.v6i1.2020.55-63.
- [25] A. Zein, “Sistem Absensi Cerdas Menggunakan Open CV Berbasis Pengenalan Wajah,” *Sainstech: Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi*, vol. 33, no. 3, Sep. 2023, doi: 10.37277/stch.v33i3.1733.
- [26] A. Jamhari, F. Mukti Wibowo, and W. Andi Saputra, “Perancangan Sistem Pengenalan Wajah Secara Real-Time pada CCTV dengan Metode Eigenface,” vol. 2, no. 2, pp. 20–032, 2020, doi: 10.20895/INISTA.V2I2.

- [27] T. Susim, C. Darujati, and I. Artikel, “Pengolahan Citra Untuk Pengenalan Wajah (Face Recognition) Menggunakan Opencv,” *Jurnal Syntax Admiration*, vol. 2, no. 3, 2021.
- [28] F. R. Setiawan dan Dewi Agushinta, “Sistem Pengenalan Wajah Dengan Metode Local Binary Pattern Histogram Pada Firebase Berbasis Opencv,” *Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK)*, vol. 4, no. 1, p. 16424, 2020.
- [29] R. Nurfitra Darma, “Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari Publikasi Ilmiah,” 2018.
- [30] H. Ardiyanti, “Cyber-Security Dan Tantangan Pengembangannya Di Indonesia,” 2016. [Online]. Available: <http://kominfo.go.id/index.php/content/detail/3980/>
- [31] D. Musabbihah, “Perencanaan Sistem Keamanan Pada Jaringan Komunikasi ITS (Intelligent Transport System) Antara OBU dan TMC Server,” Surabaya, 2018.
- [32] S. Siswidiyanto, A. Munif, D. Wijayanti, and E. Haryadi, “Sistem Informasi Penyewaan Rumah Kontrakan Berbasis Web Dengan Menggunakan Metode Prototype,” *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 15, no. 1, pp. 18–25, Apr. 2020, doi: 10.35969/interkom.v15i1.64.
- [33] A. Voutama, “Sistem Antrian Cucian Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML,” *Komputika : Jurnal Sistem Komputer*, vol. 11, no. 1, pp. 102–111, Feb. 2022, doi: 10.34010/komputika.v11i1.4677.
- [34] F.- Sonata, “Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer,” *Jurnal Komunika : Jurnal Komunikasi, Media dan Informatika*, vol. 8, no. 1, p. 22, Jun. 2019, doi: 10.31504/komunika.v8i1.1832.

- [35] R. Parlika, T. Ardhan Nisaa', S. M. Ningrum, and B. A. Haque, "Studi Literatur Kekurangan dan Kelebihan Pengujian Black Box," *TEKNOMATIKA*, vol. 10, no. 02, pp. 1–5, Sep. 2020.
- [36] C. T. Pratala, E. M. Asyer, I. Prayudi, and A. Saifudin, "Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 2, p. 111, Jun. 2020, doi: 10.32493/informatika.v5i2.4713.
- [37] F. Galuh Sembodo, G. Fadila Fitriana, and N. A. Prasetyo, "Evaluasi Usability Website Shopee Menggunakan System Usability Scale (SUS)," 2021. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [38] J. H. Yam and R. Taufik, "Hipotesis Penelitian Kuantitatif," vol. 3, no. 2, 2021.
- [39] N. Amin Fadilah, S. Garancang, and K. Abunawas, "Konsep Umum Populasi dan Sampel Dalam Penelitian," *Jurnal Pilar : Jurnal Kajian Islam Kontemporer*, vol. 14, no. 1, pp. 15–31, 2023.

LAMPIRAN

Pengujian Black Box

Implementasi Algoritma Convolutional Neural Network Dengan Identifikasi Ekspresi Wajah Untuk Deteksi Potensi Kejahatan

A. Identitas Penguji

Nama : Nugroho Dwi Saputra, S.kom., M.kom.
 Program Studi : Informatika
 Fakultas : Fakultas Teknik dan Informatika
 Tanggal Pengujian : 5 Agustus 2024

B. Petunjuk pengujian

1. Lakukan pengujian pada aplikasi sesuai dengan skenario yang ada
2. Bandingkan antara hasil pengujian dengan hasil yang diharapkan untuk mengetahui kesesuaiannya
3. Berikan tanda (√) di salah satu kolom kesimpulan, diterima apabila hasil pengujian sudah sesuai dengan yang diharapkan, perbaikan apabila disarankan untuk diperbaiki fitur yang bersangkutan

C. Tabel Pengujian

Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan	
				Diterima	Perbaikan
Halaman Home	User dapat melakukan gulir atau scroll ke bawah	User dapat melihat semua fitur yang ada pada halaman home (page prediksi, form upload)	Website akan menampilkan semua fitur yang ada pada halaman home	√	
Koneksi ke Basis Data	Aplikasi dapat terhubung ke basis data	Koneksi berhasil tanpa error	Aplikasi berhasil terhubung ke basis data	√	
Membuat Tabel di Basis Data	Tabel 'emotion_data' dibuat di basis	Tabel 'emotion_data' ada di basis	Tabel berhasil dibuat di	√	

	data	data	basis data		
Menghasilkan Kunci Enkripsi	Kunci enkripsi dihasilkan	Kunci enkripsi dihasilkan tanpa error	Kunci enkripsi berhasil dihasilkan	✓	
Mengenkripsi Gambar	Gambar dienkripsi dengan benar	Gambar berhasil dienkripsi	Gambar berhasil dienkripsi tanpa error	✓	
Prediksi Ekspresi Wajah	Sistem dapat memprediksi ekspresi wajah	Ekspresi wajah berhasil diprediksi dan ditampilkan	Ekspresi wajah berhasil diprediksi	✓	
Menyimpan Hasil Deteksi ke Basis Data	Hasil deteksi disimpan di basis data	Hasil deteksi disimpan tanpa error	Hasil deteksi berhasil disimpan di basis data	✓	
Menampilkan Gambar yang Diunggah	Gambar yang diunggah ditampilkan di halaman web	Gambar ditampilkan tanpa error	Gambar berhasil ditampilkan di halaman web	✓	
Memulai Kamera untuk Deteksi Real-time	Kamera mulai dan menampilkan video secara real-time	Video real-time ditampilkan tanpa error	Kamera berhasil menampilkan video real-time	✓	
Mengunggah Gambar untuk Deteksi	Gambar diterima dan diproses untuk deteksi	Gambar diterima dan diproses tanpa error	Gambar berhasil diproses untuk deteksi	✓	

Menampilkan Hasil Deteksi	Hasil deteksi ditampilkan di halaman web	Hasil deteksi ditampilkan tanpa error	Hasil deteksi berhasil ditampilkan	✓	
---------------------------	--	---------------------------------------	------------------------------------	---	--

D. Saran

- kemanfaatan ran atau penerapan lebih bagus untuk integrasi petugas!

TTD Penguji



Nugroho Dwi S.

Pengujian Black Box

Implementasi Algoritma Convolutional Neural Network Dengan Identifikasi Ekspresi Wajah Untuk Deteksi Potensi Kejahatan

A. Identitas Penguji

Nama : Febrian MD
Program Studi : Informatika
Fakultas : FTI
Tanggal Pengujian : 5/8/2024.

B. Petunjuk pengujian

1. Lakukan pengujian pada aplikasi sesuai dengan skenario yang ada
2. Bandingkan antara hasil pengujian dengan hasil yang diharapkan untuk mengetahui kesesuaiannya
3. Berikan tanda (✓) di salah satu kolom kesimpulan, diterima apabila hasil pengujian sudah sesuai dengan yang diharapkan, perbaikan apabila disarankan untuk diperbaiki fitur yang bersangkutan

C. Tabel Pengujian

Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan	
				Diterima	Perbaikan
Halaman Home	User dapat melakukan gulir atau scroll ke bawah	User dapat melihat semua fitur yang ada pada halaman home (page prediksi, form upload)	Website akan menampilkan semua fitur yang ada pada halaman home	✓	
Koneksi ke Basis Data	Aplikasi dapat terhubung ke basis data	Koneksi berhasil tanpa error	Aplikasi berhasil terhubung ke basis data	✓	
Membuat Tabel di Basis Data	Tabel 'emotion_data' dibuat di basis	Tabel 'emotion_data' ada di basis	Tabel berhasil dibuat di	✓	

	data	data	basis data		
Menghasilkan Kunci Enkripsi	Kunci enkripsi dihasilkan	Kunci enkripsi dihasilkan tanpa error	Kunci enkripsi berhasil dihasilkan	✓	
Mengenkripsi Gambar	Gambar dienkripsi dengan benar	Gambar berhasil dienkripsi	Gambar berhasil dienkripsi tanpa error	✓	
Prediksi Ekspresi Wajah	Sistem dapat memprediksi ekspresi wajah	Ekspresi wajah berhasil diprediksi dan ditampilkan	Ekspresi wajah berhasil diprediksi	✓	
Menyimpan Hasil Deteksi ke Basis Data	Hasil deteksi disimpan di basis data	Hasil deteksi disimpan tanpa error	Hasil deteksi berhasil disimpan di basis data	✓	
Menampilkan Gambar yang Diunggah	Gambar yang diunggah ditampilkan di halaman web	Gambar ditampilkan tanpa error	Gambar berhasil ditampilkan di halaman web	✓	
Memulai Kamera untuk Deteksi Real-time	Kamera mulai dan menampilkan video secara real-time	Video real-time ditampilkan tanpa error	Kamera berhasil menampilkan video real-time	✓	
Mengunggah Gambar untuk Deteksi	Gambar diterima dan diproses untuk deteksi	Gambar diterima dan diproses tanpa error	Gambar berhasil diproses untuk deteksi	✓	

Menampilkan Hasil Deteksi	Hasil deteksi ditampilkan di halaman web	Hasil deteksi ditampilkan tanpa error	Hasil deteksi berhasil ditampilkan	✓	
---------------------------	--	---------------------------------------	------------------------------------	---	--

D. Saran

- OK -

TTD Penguji

Febrina M D

Pengujian Black Box

Implementasi Algoritma Convolutional Neural Network Dengan Identifikasi Ekspresi Wajah Untuk Deteksi Potensi Kejahatan

A. Identitas Penguji

Nama : Ramadhan Renaldy, S.kom, m.lcum
 Program Studi : Informatika
 Fakultas : Teknik dan Informatika
 Tanggal Pengujian : 5 Agustus 2024

B. Petunjuk pengujian

1. Lakukan pengujian pada aplikasi sesuai dengan skenario yang ada
2. Bandingkan antara hasil pengujian dengan hasil yang diharapkan untuk mengetahui kesesuaiannya
3. Berikan tanda (√) di salah satu kolom kesimpulan, diterima apabila hasil pengujian sudah sesuai dengan yang diharapkan, perbaikan apabila disarankan untuk diperbaiki fitur yang bersangkutan

C. Tabel Pengujian

Nama Pengujian	Test Case	Hasil yang Diharapkan	Hasil yang Didapatkan	Kesimpulan	
				Diterima	Perbaikan
Halaman Home	User dapat melakukan gulir atau scroll ke bawah	User dapat melihat semua fitur yang ada pada halaman home (page prediksi, form upload)	Website akan menampilkan semua fitur yang ada pada halaman home	✓	
Koneksi ke Basis Data	Aplikasi dapat terhubung ke basis data	Koneksi berhasil tanpa error	Aplikasi berhasil terhubung ke basis data	✓	
Membuat Tabel di Basis Data	Tabel 'emotion_data' dibuat di basis	Tabel 'emotion_data' ada di basis	Tabel berhasil dibuat di	✓	

	data	data	basis data		
Menghasilkan Kunci Enkripsi	Kunci enkripsi dihasilkan	Kunci enkripsi dihasilkan tanpa error	Kunci enkripsi berhasil dihasilkan	✓	
Mengenkripsi Gambar	Gambar dienkripsi dengan benar	Gambar berhasil dienkripsi	Gambar berhasil dienkripsi tanpa error	✓	
Prediksi Ekspresi Wajah	Sistem dapat memprediksi ekspresi wajah	Ekspresi wajah berhasil diprediksi dan ditampilkan	Ekspresi wajah berhasil diprediksi	✓	
Menyimpan Hasil Deteksi ke Basis Data	Hasil deteksi disimpan di basis data	Hasil deteksi disimpan tanpa error	Hasil deteksi berhasil disimpan di basis data	✓	
Menampilkan Gambar yang Diunggah	Gambar yang diunggah ditampilkan di halaman web	Gambar ditampilkan tanpa error	Gambar berhasil ditampilkan di halaman web	✓	
Memulai Kamera untuk Deteksi Real-time	Kamera mulai dan menampilkan video secara real-time	Video real-time ditampilkan tanpa error	Kamera berhasil menampilkan video real-time	✓	
Mengunggah Gambar untuk Deteksi	Gambar diterima dan diproses untuk deteksi	Gambar diterima dan diproses tanpa error	Gambar berhasil diproses untuk deteksi	✓	

Menampilkan Hasil Deteksi	Hasil deteksi ditampilkan di halaman web	Hasil deteksi ditampilkan tanpa error	Hasil deteksi berhasil ditampilkan	✓	
---------------------------	--	---------------------------------------	------------------------------------	---	--

D. Saran

- Coba lakukan testing menggunakan data lain diluar dataset untuk mendapatkan hasil akurasi yang dapat dibandingkan.

TTD Penguji



Ramadhan Renoldy

Responden 1

System Usability Scale (SUS) Questionnaire

Silakan berikan penilaian Anda terhadap pernyataan berikut dengan menandai salah satu kotak yang sesuai dengan pengalaman Anda menggunakan sistem ini:

No	Pernyataan	Sangat Tidak Setuju (1)	Tidak Setuju (2)	Netral (3)	Setuju (4)	Sangat Setuju (5)
1	Saya merasa bahwa saya akan sering menggunakan sistem ini.	[..]	[..]	[..]	[..] ✓	[..]
2	Saya merasa sistem ini tidak perlu rumit.	[..]	[..] ✓	[..]	[..]	[..]
3	Saya merasa bahwa sistem ini mudah digunakan.	[..]	[..]	[..]	[..]	[..] ✓
4	Saya pikir saya akan membutuhkan bantuan dari seseorang yang ahli untuk dapat menggunakan sistem ini.	[..]	[..] ✓	[..]	[..]	[..]
5	Saya merasa bahwa berbagai fungsi dalam sistem ini terintegrasi dengan baik.	[..]	[..]	[..]	[..]	[..] ✓
6	Saya merasa bahwa sistem ini	[..] ✓	[..]	[..]	[..]	[..]

	memiliki terlalu banyak inkonsistensi.					
7	Saya merasa bahwa kebanyakan orang akan cepat mempelajari cara menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..] ✓
8	Saya merasa bahwa sistem ini sangat membingungkan ketika digunakan.	[..] ✓	[..]	[..]	[..]	[..]
9	Saya merasa percaya diri saat menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..] ✓
10	Saya harus mempelajari banyak hal sebelum bisa mulai menggunakan sistem ini.	[..] ✓	[..]	[..]	[..]	[..]

Tanda
Ahmad Murohan

Responden 2

System Usability Scale (SUS) Questionnaire

Silakan berikan penilaian Anda terhadap pernyataan berikut dengan menandai salah satu kotak yang sesuai dengan pengalaman Anda menggunakan sistem ini:

No	Pernyataan	Sangat Tidak Setuju (1)	Tidak Setuju (2)	Netral (3)	Setuju (4)	Sangat Setuju (5)
1	Saya merasa bahwa saya akan sering menggunakan sistem ini.	[..]	[..]	[..]	[..]	[✓]
2	Saya merasa sistem ini tidak perlu rumit.	[..]	[✓]	[..]	[..]	[✓]
3	Saya merasa bahwa sistem ini mudah digunakan.	[..]	[..]	[..]	[..]	[✓]
4	Saya pikir saya akan membutuhkan bantuan dari seseorang yang ahli untuk dapat menggunakan sistem ini.	[✓]	[..]	[..]	[..]	[✓]
5	Saya merasa bahwa berbagai fungsi dalam sistem ini terintegrasi dengan baik.	[..]	[..]	[..]	[..]	[✓]
6	Saya merasa bahwa sistem ini	[✓]	[..]	[..]	[..]	[..]

	memiliki terlalu banyak inkonsistensi.	.				
7	Saya merasa bahwa kebanyakan orang akan cepat mempelajari cara menggunakan sistem ini.	[..]	[..]	[..]	[✓]	[..]
8	Saya merasa bahwa sistem ini sangat membingungkan ketika digunakan.	[..]	[..]	[✓]	[..]	[..]
9	Saya merasa percaya diri saat menggunakan sistem ini.	[..]	[..]	[..]	[..]	[✓]
10	Saya harus mempelajari banyak hal sebelum bisa mulai menggunakan sistem ini.	[..]	[..]	[✓]	[..]	[..]

M. PRADA H

Responden 3

System Usability Scale (SUS) Questionnaire

Silakan berikan penilaian Anda terhadap pernyataan berikut dengan menandai salah satu kotak yang sesuai dengan pengalaman Anda menggunakan sistem ini:

No	Pernyataan	Sangat Tidak Setuju (1)	Tidak Setuju (2)	Netral (3)	Setuju (4)	Sangat Setuju (5)
1	Saya merasa bahwa saya akan sering menggunakan sistem ini.	[..]	[..]	[..]	[..]	<input checked="" type="checkbox"/>
2	Saya merasa sistem ini tidak perlu rumit.	<input checked="" type="checkbox"/>	[..]	[..]	[..]	[..]
3	Saya merasa bahwa sistem ini mudah digunakan.	[..]	[..]	[..]	[..]	<input checked="" type="checkbox"/>
4	Saya pikir saya akan membutuhkan bantuan dari seseorang yang ahli untuk dapat menggunakan sistem ini.	<input checked="" type="checkbox"/>	[..]	[..]	[..]	[..]
5	Saya merasa bahwa berbagai fungsi dalam sistem ini terintegrasi dengan baik.	[..]	[..]	[..]	<input checked="" type="checkbox"/>	[..]
6	Saya merasa bahwa sistem ini	<input checked="" type="checkbox"/>	[..]	[..]	[..]	[..]

	memiliki terlalu banyak inkonsistensi.					
7	Saya merasa bahwa kebanyakan orang akan cepat mempelajari cara menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..]
8	Saya merasa bahwa sistem ini sangat membingungkan ketika digunakan.	[..]	[..]	[..]	[..]	[..]
9	Saya merasa percaya diri saat menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..]
10	Saya harus mempelajari banyak hal sebelum bisa mulai menggunakan sistem ini.	[..]	[..]	[..]	[..]	[..]


 ACIL WISOWD

Responden 4

System Usability Scale (SUS) Questionnaire

Silakan berikan penilaian Anda terhadap pernyataan berikut dengan menandai salah satu kotak yang sesuai dengan pengalaman Anda menggunakan sistem ini:

No	Pernyataan	Sangat Tidak Setuju (1)	Tidak Setuju (2)	Netral (3)	Setuju (4)	Sangat Setuju (5)
1	Saya merasa bahwa saya akan sering menggunakan sistem ini.	[..]	[..]	[..]	[..]	[✓]
2	Saya merasa sistem ini tidak perlu rumit.	[..]	[✓]	[..]	[..]	[..]
3	Saya merasa bahwa sistem ini mudah digunakan.	[..]	[..]	[..]	[..]	[✓]
4	Saya pikir saya akan membutuhkan bantuan dari seseorang yang ahli untuk dapat menggunakan sistem ini.	[✓]	[..]	[..]	[..]	[..]
5	Saya merasa bahwa berbagai fungsi dalam sistem ini terintegrasi dengan baik.	[..]	[..]	[..]	[..]	[✓]
6	Saya merasa bahwa sistem ini	[✓]	[..]	[..]	[..]	[..]

	memiliki terlalu banyak inkonsistensi.					
7	Saya merasa bahwa kebanyakan orang akan cepat mempelajari cara menggunakan sistem ini.	[..]	[..]	[..]	[..]	[✓]
8	Saya merasa bahwa sistem ini sangat membingungkan ketika digunakan.	[✓]	[..]	[..]	[..]	[..]
9	Saya merasa percaya diri saat menggunakan sistem ini.	[..]	[..]	[..]	[..]	[✓]
10	Saya harus mempelajari banyak hal sebelum bisa mulai menggunakan sistem ini.	[..]	[..]	[..]	[✓]	[..]

Andika Blaur 09.

Responden 5

System Usability Scale (SUS) Questionnaire

Silakan berikan penilaian Anda terhadap pernyataan berikut dengan menandai salah satu kotak yang sesuai dengan pengalaman Anda menggunakan sistem ini:

No	Pernyataan	Sangat Tidak Setuju (1)	Tidak Setuju (2)	Netral (3)	Setuju (4)	Sangat Setuju (5)
1	Saya merasa bahwa saya akan sering menggunakan sistem ini.	[..]	[..]	[..]	[..]	<input checked="" type="checkbox"/>
2	Saya merasa sistem ini tidak perlu rumit.	[..]	<input checked="" type="checkbox"/>	[..]	[..]	[..]
3	Saya merasa bahwa sistem ini mudah digunakan.	[..]	[..]	[..]	[..]	<input checked="" type="checkbox"/>
4	Saya pikir saya akan membutuhkan bantuan dari seseorang yang ahli untuk dapat menggunakan sistem ini.	[..]	[..]	<input checked="" type="checkbox"/>	[..]	[..]
5	Saya merasa bahwa berbagai fungsi dalam sistem ini terintegrasi dengan baik.	[..]	[..]	[..]	[..]	<input checked="" type="checkbox"/>
6	Saya merasa bahwa sistem ini	<input checked="" type="checkbox"/>	[..]	[..]	[..]	[..]

	memiliki terlalu banyak inkonsistensi.					
7	Saya merasa bahwa kebanyakan orang akan cepat mempelajari cara menggunakan sistem ini.	[..]	[..]	[..]	[..]	[✓]
8	Saya merasa bahwa sistem ini sangat membimngungkan ketika digunakan.	[✓]	[..]	[..]	[..]	[..]
9	Saya merasa percaya diri saat menggunakan sistem ini.	[..]	[..]	[..]	[..]	[✓]
10	Saya harus mempelajari banyak hal sebelum bisa mulai menggunakan sistem ini.	[..]	[✓]	[..]	[..]	[..]

Titin Agustina

Amf