

**PERUBAHAN KECERAHAN *GAME* 2 DIMENSI BERDASARKAN
REALITAS NYATA EKSPRESI WAJAH BERBASIS METODE
*CONVOLUTIONAL NEURAL NETWORK***

SKRIPSI



**UIN SUNAN AMPEL
SURABAYA**

Disusun Oleh:

SALMA NAVISA

H76217048

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA
2022**

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini,

Nama : Salma Navisa

NIM : H76217048

Program Studi : Sistem Informasi

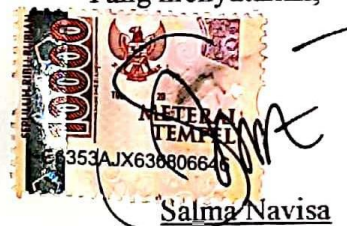
Angkatan : 2017

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul: "PERUBAHAN KECERAHAN GAME 2 DIMENSI BERDASARKAN REALITAS NYATA EKSPRESI WAJAH BERBASIS METODE CONVOLUTIONAL NEURAL NETWORK". Apabila suatu saat nanti terbukti saya melakukan Tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 20 Januari 2022

Yang menyatakan,



1000
METAL
TEMPER
E 353AJX630806646

Salma Navisa

NIM. H76217048

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi Oleh:

NAMA : SALMA NAVISA

NIM : H76217048

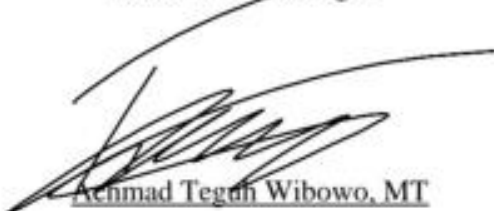
JUDUL : PERUBAHAN KECERAHAN *GAME* 2 DIMENSI BERDASARKAN
REALITAS NYATA EKSPRESI WAJAH BERBASIS METODE
CONVOLUTIONAL NEURAL NETWORK

Ini telah diperiksa dan disetujui untuk diujikan

Surabaya, 20 Januari 2022

Menyetujui

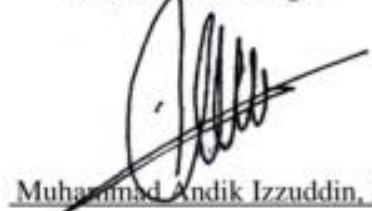
Dosen Pembimbing 1



Achmad Teguh Wibowo, MT

NIP. 198810262014031003

Dosen Pembimbing 2



Muhammad Andik Izzuddin, MT

NIP. 198403072014031001

LEMBAR PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi Salma Navisa ini telah dipertahankan
di depan tim penguji skripsi
Surabaya, 20 Januari 2022

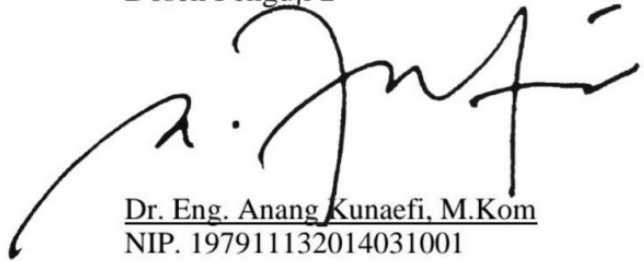
Mengesahkan,
Dewan Penguji

Dosen Penguji 1



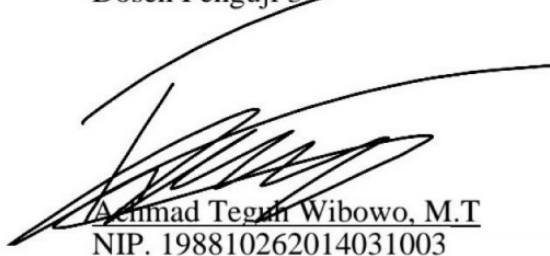
Bayu Adhi Nugroho, M.Kom
NIP. 197905182014031001

Dosen Penguji 2



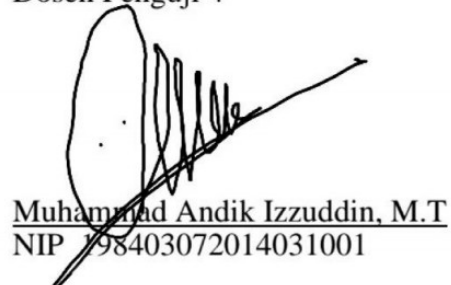
Dr. Eng. Anang Kunaefi, M.Kom
NIP. 197911132014031001

Dosen Penguji 3



Achmad Teguh Wibowo, M.T
NIP. 198810262014031003

Dosen Penguji 4



Muhammad Andik Izzuddin, M.T
NIP. 198403072014031001

Mengetahui,

Dekan Fakultas Sains dan Teknologi

UIN Sunan Ampel Surabaya



Prof. Dr. Hj. Evy Fatmatur Rusydiyah, M.Ag.
NIP. 197312272005012003



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN

Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

**LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : SALMA NAVISA
NIM : H76217048
Fakultas/Jurusan : SAINS DAN TEKNOLOGI/ SISTEM INFORMASI
E-mail address : salmanavisa17@gmail.com

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :

☒ Skripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (.....)
yang berjudul :

PERUBAHAN KECERAHAN GAME 2 DIMENSI BERDASARKAN REALITAS

NYATA EKSPRESI WAJAH BERBASIS METODE CONVOLUTIONAL

NEURAL NETWORK

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara **fulltext** untuk kepentingan akademis tanpa perlu meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 20 Januari 2022

Penulis

(Salma Navisa)

nama terang dan tanda tangan

ABSTRAK

**PERUBAHAN KECERAHAN GAME 2 DIMENSI BERDASARKAN
REALITAS NYATA EKSPRESI WAJAH BERBASIS METODE
CONVOLUTIONAL NEURAL NETWORK**

Oleh:

Salma Navisa

Keberhasilan *game* bergantung pada kondisi pemain. Selama *game* berlangsung, pemain cenderung memunculkan keadaan mereka melalui ekspresi wajah. Keadaan ini dapat digunakan sebagai pengembangan *game* untuk dikombinasikan dengan ekspresi wajah pemain, seperti *face expression recognition* (FER). Dalam penelitian ini membahas tentang *game Flappy Bird* yang menggunakan FER sebagai simulasi pada perubahan *masking layer*. Selama permainan berlangsung, *webcam* menangkap citra wajah secara *real-time* dan di deteksi ke dalam kategori menggunakan metode CNN. Kemudian *output* dari hasil deteksi ekspresi dikirim pada sistem *game* untuk mengubah *masking layer* dengan tingkat kegelapan masing-masing berdasarkan kelas *angry* (100 %), *happy* (5 %), dan *neutral* (50 %). Aplikasi ini berhasil dijalankan pada kedua dataset menghasilkan akurasi sebesar 88% pada *dataset CK+* dengan tingkat *error* 0.39 dan 76 % pada *dataset custom* dengan tingkat *error* 0.48 selama 100 *epoch*. Sehingga mendapatkan nilai dari hasil deteksi FER pada kamera *webcam* dengan aplikasi *flappy bird* menunjukkan keberhasilan 78 %. Algoritma CNN dapat menghasilkan nilai *error* yang rendah dengan tingkat nilai akurasi yang tinggi dalam pengujian *k-fold cross validation* dan ROC.

Kata Kunci : Face Expression Recognition, Detection, Convolutional Neural Network, Game, Flappy Bird, Masking Layer

DAFTAR ISI

vii

DAFTAR GAMBAR

X

BAB I

PENDAHULUAN

1.1. Latar Belakang

Saat ini, teknologi berbasis gadget komputer mampu digunakan untuk bermain *game*, sehingga *game* menjadi salah satu hiburan digital yang populer digunakan. Perusahaan *game* menjadi ekonomis karena saat ini berbagai perangkat operasional *game* mudah ditemukan dari yang murah hingga mahal (Sahni *et al.*, 2018). Karena rancangan multimedianya yang bervariasi meliputi video, animasi, 3 Dimensi, teks, audio dan gambar, *game* dapat dimainkan dari berbagai kalangan usia (Šag and Orehovacki, 2019). Secara historis, perusahaan *game* dapat menulis kode, merancang *sprite* dan ikon, menangani suara dan musik, menghasilkan animasi yang diperlukan. Seiring dengan perkembangan teknologo, industri *game* berinovasi mengembangkan peran dengan mengerjakan sebuah mesin *Artificial Intelligence*. Pengembang ini akan berfokus pada fitur yang memungkinkan *game* dapat beroperasi mengikuti keadaan realitas dunia nyata (Hunt, 2019)

Game dianggap sebagai cara bermain pada platform tertentu dimana pengguna dapat mengoperasikan karakter untuk berhasil ke tahap berikutnya. Berdasarkan cara bermainnya, *game* dibagi dalam berbagai genre. Mulai dari *Real-Time Strategy* (RTS), *Role Playing Game* (RPG), *First Person Shooter* (FPS), *Board Games*, *Racing games*, *Simulation* hingga *Hybrid*. Genre *Hybrid* terbilang unik karena kombinasi dari dua atau lebih genre *game* (Alfred Tanuwijaya, Pragantha and Kristyadi, 2018). Selain itu, berdasarkan bentuknya *game* dibagi menjadi 2 Dimensi (2D) dan 3 Dimensi (3D). Dimana, *game* 2D tidak kalah aktif diminati dari maraknya penggunaan *game* 3D. Prinsip, antarmuka, dan akses *game* 2D berkontribusi pada pengembangan *game* yang telah mewakili era kejayaan *game* 2D dengan kesederhanaan antarmuka dan kemudahan pengoperasian oleh pengguna, diantaranya *Icy Tower*TM, *Angry Birds*TM, dan *Flappy Bird*TM (Šag and Orehovacki, 2019). Hal ini karena *game* 2D terbilang relatif murah dan mudah diimplementasikan (Sahni *et al.*, 2018).

Game 2D cocok untuk dikombinasikan dengan ekspresi pemain, seperti pengenalan ekspresi wajah secara otomatis. Hal ini dapat meningkatkan

keterlibatan emosional pemain (Ninaus, 2019). Dalam penelitian ini, penerapan *Game Flappy Bird* TM memiliki karakter utama seekor unggas, yang dapat dioperasikan pengguna untuk melewati ruang antara pipa, demi memperoleh poin (Ruiz Estrada, 2011). Keberhasilan dari *game* komputer bergantung pada kondisi pemain. Selama *game* berlangsung, pemain memunculkan ekspresi mereka melalui ekspresi wajah. Ketika pemain berhasil melalui *level game*, pemain cenderung menampilkan ekspresi bahagia. Begitu juga ketika pemain mengalami kesulitan mengalahkan musuh, atau memecahkan teka-teki, pemain cenderung memunculkan wajah frustrasi atau sedih (Moniaga *et al.*, 2018). Secara alami manusia dapat menunjukkan ekspresi dengan dua cara yaitu melalui visualisasi dan emosi. Ketika ekspresi ditunjukkan melalui visualisasi, manusia cenderung mengungkapkannya melalui bahasa, suara, dan ekspresi wajah. Berbeda dengan ekspresi yang ditunjukkan melalui emosi yaitu ditampilkan dalam bentuk mimik wajah seperti senang, datar, sedih, dan marah. Namun, sebesar 55% manusia mudah mengungkapkan ekspresi mereka dalam bentuk mimik wajah (Wang, Lee and Lee, 2018).

Kemajuan teknologi terbaru menghasilkan sub-bidang *Natural Language Processing* (NLP) sebagai solusi untuk analisis data linguistik dengan metode komputasi terhadap tugas-tugas semantik, klasifikasi teks, analisis, hingga deteksi gambar. Sebagian besar model yang diusulkan dalam pemecahan masalah NLP didasarkan pada algoritma *Recurrent Neural Network* (RNN) atau *Convolutional Neural Network* (CNN). Dimana, RNN diterapkan pada pengujian teks dan CNN pada karakter vektor gambar. NLP dapat mengidentifikasi individu dari gambar digital atau video. Sistem pengenalan ini bekerja dengan membandingkan fitur wajah yang dipilih dalam database gambar wajah yang diberikan, konsep ini dikenal dengan topik *Facial expression recognition* (FER) (Merdivan *et al.*, 2019).

FER adalah salah satu cara yang paling kuat dan alami bagi manusia untuk secara langsung mengkomunikasikan ekspresi dan niat mereka (Hassouneh, Mutawa and Murugappan, 2020). *Human computer interaction* (HCI) menjelaskan bahwa pengenalan ekspresi wajah dapat menarik perhatian karena berpusat pada manusia. Beberapa penelitian menunjukkan pentingnya mesin dapat menafsirkan ekspresi manusia dengan mempelajari perubahan status emosi seseorang

berdasarkan citra wajahnya (Hussein, Qidwai and Al-Meer, 2020). Pendekatan modern ini merekayasa fitur FER secara otomatis menggunakan satu tahap pembelajaran “*end-to-end*” yaitu berbasis *deep learning* dengan metode CNN.

Dalam hal pengenalan ekspresi wajah, penelitian terbaru berfokus pada pengembangan dan pencapaian model. Pendekatan yang berbeda telah banyak dikembangkan dengan menambahkan *block*, *layer*, dan *multitasking network*. Hal ini bertujuan untuk meningkatkan ciri-ciri pada objek gambar yang berdekatan. Namun pendekatan ini memungkinkan informasi wajah tetap dominan. Inferensi akan berfokus mendeteksi wajah dari pada ekspresi atau ekspresi wajah tersebut. Untuk menangani kelemahan ini, H. Ding mengusulkan model FER dengan mencairkan *layer convolutional network* untuk menangkap ekspresi dan membekukan *layer* untuk wajah (Melinte and Vladareanu, 2020).

Penelitian AI dan *game* yang mencakup sistem pengenalan ekspresi dapat merasakan ekspresi manusia dan bertindak tanpa intervensi manusia lain. Perangkat kamera yang digunakan untuk menangkap ekspresi wajah pemain, dianalisis dan diklasifikasikan ke dalam 3 jenis ekspresi dengan algoritma CNN. Teknik jaringan

melakukan klasifikasi wajah.

- b. Sebagai sarana pengembangan penerapan keilmuan dalam penelitian FER berbasis deep learning dengan melakukan klasifikasi wajah pemain *game*.
2. Secara Aplikatif
 - a. Membantu perusahaan *game* dalam menganalisa dan meningkatkan kinerja *game* dengan mengetahui ekspresi pemain.
 - b. FER dapat dijadikan sebagai pertimbangan dan inovasi untuk pengembangan *game* 2 dimensi dimasa mendatang.

1.6. Sistematika Penulisan Skripsi

Adapun penulisan skripsi Program Studi Sistem Informasi Fakultas Sains dan Teknologi memiliki aturan dan sistematika yang dibagi menjadi beberapa bagian, sebagai berikut:

- a. BAB I PENDAHULUAN

Pendahuluan memaparkan latar belakang permasalahan yang diangkat dalam penelitian yakni permainan 2 dimensi yang dapat mengklasifikasikan wajah pemain melalui kamera *webcam* serta mengatur tingkat kesulitan *game*. Lalu bab ini juga menjelaskan perumusan masalah, tujuan penelitian, dan manfaat yang diperoleh dari penelitian.

- ## b. BAB II TINJAUAN PUSTAKA

Tinjauan pustaka berisi tentang penjelasan penelitian terdahulu yang relevan dengan penelitian saat ini. Bab ini memaparkan secara singkat teori dan konsep yang terkait dengan penelitian seperti pemaparan objek penelitian, metode-metode yang terkait dengan penelitian, dan sebagainya.

- ### c. BAB III METODOLOGI PENELITIAN

Metodologi Penelitian berisi tentang penjelasan alur sistematika penelitian yang terdiri dari beberapa tahap antara lain, identifikasi masalah, perencanaan pengolahan data, pengembangan, implementasi, hingga tahap akhir penulisan laporan.

d. BAB IV HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil-hasil dan pembahasan dari seluruh tahapan penelitian, bagaimana implementasi metode CNN pada *game* 2 dimensi *Flappy Bird*TM berbasis *face recognition*.

e. BAB V PENUTUP

Penutup berisi kesimpulan dan saran yang diperoleh dari serangkaian pembahasan skripsi berdasarkan analisa masalah dan penulisan tugas akhir.

(R Deva Dharshni, 2020). Penelitian tahun 2019 dengan judul “*Real Time Emotion Recognition from Facial Expressions Using CNN Architecture*” bertujuan untuk menganalisis gangguan otak dan psikologis manusia menggunakan *emotion recognition*. Penelitian ini mengklasifikasikan tujuh emosi wajah manusia secara *real-time* dengan perpustakaan *haarcascade* dalam jarak 30 gambar per detik dari kamera komputer. Dimana untuk pengenalan ekspresi wajah menggunakan arsitektur *LeNet* berbasis CNN menghasilkan akurasi yang sangat baik mencapai 96,43%. Sehingga model pengujian *real-time* memiliki fungsionalitas untuk mendeteksi, menangkap, dan membaca wajah dalam hitungan detik (Ozdemir et al., 2019).

Berdasarkan penelitian tahun 2018 dengan judul “*Facial Expression Recognition as Dynamic Game Balancing System*”, mengusulkan untuk menangkap ekspresi pemain dari wajah mereka dan menggunakannya untuk menyesuaikan kesulitan *game* secara dinamis. Sistem dievaluasi dengan mengaktifkan FER pada *dynamic balancing* dan satunya tidak. Ada peningkatan yang signifikan pada fokus pemain, keseimbangan dinamis, dan pengalaman pemain ketika menggunakan FER yakni berdasarkan hasil kuisioner *Kolmogorov-Smirnov Test* dengan nilai awal probabilitas $P < 0,005$, pengalaman pemain meningkat secara statistik $\leq 0,05$ dari pada sebelum menggunakan FER (Moniaga et al., 2018).

Penelitian selanjutnya pada tahun 2021 dengan judul *“Real Time Emotion Recognizer And Classifier For Facial Expressions Based On Machine Learning Approach”* menggunakan sistem yang dapat mengenali ekspresi seseorang setiap saat dengan memeriksa ekspresi dari bingkai ke bingkai dalam media video. Ini terdiri dari beberapa gambar multistage untuk mengekstrak representasi fitur. Representasi wajah menunjukkan algoritma pengenalan dan deteksi memperoleh 90% akurasi. Tangkapan ekspresi wajah dari kamera dibandingkan dengan *database* untuk mendapatkan kategori ekspresi yang paling akurat (Balasaranya et al., 2021).

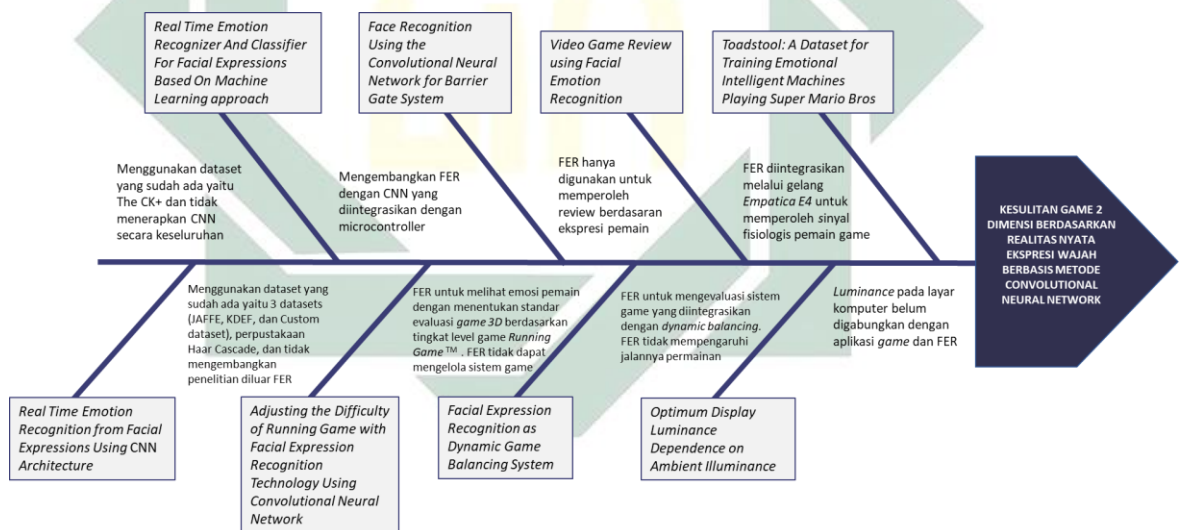
Penelitian tahun 2017 dengan judul “*Optimum Display Luminance Dependence on Ambient Illuminance*”, menjelaskan pencahayaan yang terang dalam kondisi cahaya yang redup menyebabkan silau hingga penyakit mata.

Penelitian akhir dengan judul “*Face Recognition Using the Convolutional Neural Network for Barrier Gate System*” pada tahun 2021, menerapkan teknik pengenalan wajah menggunakan CCTV yang mampu mencegah orang tidak berkepentingan masuk ke pintu gerbang. Metode CNN diimplementasikan untuk membuka dan menutup barrier gate secara *real-time*. Prosesnya terdiri dari *convolutional layer*, *pooling layer*, *max pooling*, *flattening*, dan *fully connected layer*. Hasil eksperimen diperoleh rata-rata tingkat kesalahan 0,320 dan tingkat akurasi sebesar 93,3% yang menunjukkan bahwa teknik pengenalan wajah menggunakan CNN sangat direkomendasikan (Prasetyo et al., 2021).

Penelitian “*Video Game Review using Facial Emotion Recognition*”, “*Real Time Emotion Recognition from Facial Expressions Using CNN Architecture*”, dan “*Real Time Emotion Recognizer And Classifier For Facial Expressions Based On Machine Learning approach*” mengembangkan penelitian dengan mengklasifikasikan wajah kedalam tujuh emosi secara *real-time*. Dimana pengenalan ekspresi wajah CNN menghasilkan akurasi yang sangat baik mencapai

Teknik pengenalan wajah menggunakan CNN sangat direkomendasikan sesuai dengan penelitian di atas dengan judul “*Face Recognition Using the Convolutional Neural Network for Barrier Gate System*” yang menerapkan metode CNN memperoleh hasil akurasi yang sangat baik didukung dengan beberapa proses diantaranya *convolutional layer*, *pooling layer*, *max pooling*, *flattening*, dan *fully connected layer*. Berdasarkan tinjauan pustaka dan beberapa penelitian yang telah dilakukan di atas, dapat dijadikan acuan penelitian ini untuk mengembangkan sistem pengenalan wajah menggunakan algoritma CNN dalam mengontrol kecerahan layar *game* khususnya 2 dimensi

Research Gap penelitian ini dengan penelitian sebelumnya dijelaskan pada Gambar 2.1 di bawah ini



Gambar 2.1 menjelaskan bahwa penelitian “*Perubahan Kecerahan Game 2 Dimensi Berdasarkan Realitas Nyata Ekspresi Wajah Berbasis Metode Convolutional Neural Network*” dikembangkan peneliti berdasarkan pengalaman atau temuan peneliti-peneliti terdahulu. Penelitian ini didasarkan untuk dapat menyelesaikan suatu masalah atau topik terdahulu dengan jawaban baru. 8 jurnal

penelitian terdahulu dipilih sebagai pendukung penelitian ini dengan celah-celah metode atau topik yang belum diintegrasikan.

2.2 Teori-Teori Dasar

2.2.1 Game

Sejarah permainan komputer dimulai sejak awal 1950-an sebagai penelitian sains yaitu dengan merancang *game* dan simulasi sederhana. Video *game* tercapai popularitas sekitar tahun 1970-an dan 1980-an, dengan video *game* arcade, *game* konsol, dan *game* rumah. Sejak video *game* mendapatkan popularitas besar, tampilan *game* berevolusi dengan grafik, cerita, plot, dan telah menjadi bentuk populer dari hiburan budaya modern (Kumar and Kumar, 2016). Seiringnya waktu, *game* terbagi menjadi beberapa jenis diantaranya (Singkoh, Lumenta and Tulenan, 2016).

a. Action Game

Termasuk dalam kategori *gameplay* dengan model pertarungan. Beberapa bentuk *action game* diantaranya *Action Adventure Game*, *Stealth Game*, *Fighting Game*, *Maze Game*, *Platform Game*, dan *Shooter Game* yang terbagi dalam genre *First-Person Shooter Game*, *Massively Multiplayer*, *Third-Person Shooter Game*, dan *Shootem up Game*.

b. Adventure Game

Salah satu kategori *gameplay* yang mengharuskan pemain memecahkan puzzle melalui interaksi dengan objek lain dalam *game* tersebut. *Adventure Game* terbagi dalam beberapa bentuk diantaranya *Text Adventure Game/Interactive Fiction Game*, *Graphical Adventure Game*, *Visual Novel Game*, *Interactive Movie Game*, dan *Dialog Game*.

c. Role-Playing Game (RPG)

RPG mempunyai *gameplay* di mana karakter pengguna akan bertualang dengan *skill combat* sesuai dengan cerita yang sudah dibuat dalam *game*. RPG terbagi dalam beberapa bentuk seperti, *Action Role-Playing Games*, *Massively Multiplayer*, dan *Tactical Role-Playing Game*.

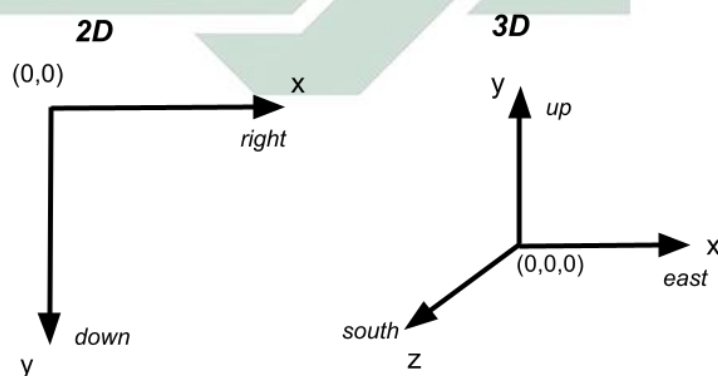
d. Simulation Game

Game ini bertujuan untuk memberi pengalaman pengguna melalui simulasi. *Construction and management simulation game* (CMSs) merupakan *game* simulasi dimana pengguna dapat mendirikan, memperluas, atau mengelola proyek fiksi. *Game* ini dapat diimplementasikan ke dalam jenis *City-Building Game*, *Economic Simulation Game*, *God Game*, dan *Government Simulation Game*.

e. Strategy Game

Strategy game salah satu *gameplay* yang membutuhkan pemikiran yang tepat dan cepat untuk meraih kemenangan. Strategi *game* diimplementasikan ke dalam beberapa jenis *Real-Time Strategy and Turn-Based Strategy Game*, *Tactical Game*, dan *4X Game*

Perkembangan game dalam bentuk multimedia terdiri dari integrasi dan adaptasi video, animasi, 2 dan 3 dimensi, teks, audio dan gambar tidak bergerak. Dimana perkembangan ini berupa pekerjaan yang membutuhkan tim ahli dalam memahami tentang teori dan metode praktis selama proses pengembangan. Sejarah pembuatan game datang dari penelitian dan analisis terperinci seperti, game 2 dimensi dengan sebutan *game* "jadul" pada masa kanak-kanak (Šag and Orehovacki, 2019). *Game* 2D dan 3D dibedakan berdasarkan koordinat yang ditunjukkan pada Gambar 2.1.

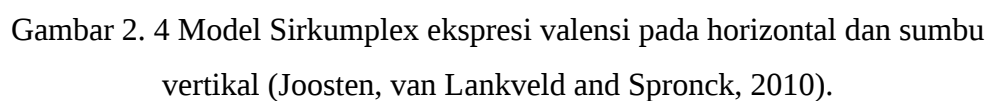


Gambar 2. 2 Perbedaan gambar 2D dan 3D berdasarkan koordinat

Gambar 2.2 menunjukkan *Game 2 Dimensi* terdiri dari 2 sisi yaitu x dan y pada gambar Vektor atau Bitmap. Secara keseluruhan objek dan gerakan karakter *game 2D* terbatas pada satu bidang datar yang dinamakan *StaticView*. *Side scrolling*

2.2.3 Visual Color Brightness Screen

Kontras pencahayaan pada *luminance* digunakan sebagai pembanding cahaya warna hitam dan putih (hitam dan putih) saat muncul di layar. Pada layar LCD, hitam tidak pernah benar-benar bernilai nol. Selalu ada sejumlah cahaya yang bocor melalui matriks kristal cair, bahkan pada gambar yang sepenuhnya hitam. Putih juga dibatasi oleh seberapa terang lampu latar LCD. Usia pengguna juga berpengaruh pada kemampuan membedakan tampilan layar yang secara signifikan menurun pada tingkat pencahayaan dan kontras tertentu. Kontras dan pencahayaan merupakan prediktor kuat keterbacaan atau penglihatan pengguna menggunakan komputer (Schuchhardt et al., 2015).



Tabel 2. 1 Hubungan Emosi dan warna (Plutchik, 2001).

No	Ekspresi	Warna
1	Kaget	Biru Muda
2	Takut	Hijau Tua
3	Penerimaan	Hijau Muda
4	Sukacita	Kuning
5	Waspada	Orange
6	Marah	Merah
7	Jijik	Ungu
8	Sedih	Biru Tua

Ekspresi manusia memiliki beberapa titik sebagai respon sensitive terhadap suatu sistem. Salah satunya adalah mata. Indra penglihatan paling sensitif terhadap perubahan luminance dan intensitas cahaya. Skala warna umumnya memiliki pencahayaan di semua tempat bersamaa dengan warna hitam dan putih sebagai kadar pencahayaannya. Masalah interpretasi adalah bahwa lokasi dengan

Level	Tingkat Kecerahan (%)	Luminance (cd/m ²)
High	100	249.9
Medium	50	52.0
Low	5	0.28
Zore	0	0

Level	Tingkat Kecerahan (%)	Luminance (cd/m ²)
High	100	249.9
Medium	50	52.0
Low	5	0.28
Zore	0	0

2.2.4 Face Recognition

18

Kemudian fitur global yakni output dari *average pooling layer* dan fitur lokal digabungkan untuk meningkatkan informasi berupa struktur yang ada di sekitar *landmark* wajah (Huang *et al.*, 2017).

2.2.5 Facial Expression Recognition (FER)

FER bertujuan untuk memprediksi ekspresi wajah dasar (misalnya, kebahagiaan, kesedihan, kemarahan, terkejut, jijik, dan takut). Metode ini membantu mesin memahami niat atau emosi manusia dengan menganalisis wajah. FER mendapat perhatian yang cukup besar karena aplikasi potensial dalam deteksi perilaku abnormal manusia, pendekatan FER baru-baru ini berdasarkan pembelajaran mendalam mengungguli pendekatan FER tradisional. Namun, hanya sedikit penelitian tentang *deep learning* yang menggunakan gambar kedalaman wajah sebagai *input deep neural network* (Yang *et al.*, 2017).

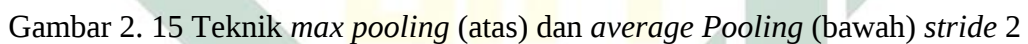
Representasi gambar terdiri dari titik-titik yang dipengaruhi oleh emosi yang berbeda, hal ini ditunjukkan pada Gambar 2.11. Dimana seluruh vektor fitur minimal dari enam emosi dasar (Terkejut, Senang, Jijik, Takut, Marah, dan Sedih) ditunjukkan dengan menggambarkan kelompok, jumlah *Facial Animation Parameters* (FAP), dan deskripsi tekstual emosi. Berdasarkan FAP yang diilustrasikan memberikan informasi mengenai jumlah titik fitur dan nomor kelompok pada keenam emosi dasar. Batasan tiap emosi dimulai dari titik pandang emosi dan puncak. Setiap emosi yang mencapai fase puncak dari fase awal didefinisikan sebagai respon puncak (Kumar and Rajagopal, 2019).

2.2.6 Deep Learning

Deep learning adalah bagian dari pembelajaran mesin dan didasarkan pada algoritma yang dirangsang oleh fungsi otak secara terstruktur. Pertumbuhan eksplosif dan ketersediaan data, beriringan dengan kemajuan teknologi menyebabkan munculnya studi baru dalam pembelajaran terdistribusi dan mendalam. *Deep learning* berakar dari jaringan saraf konvensional, secara signifikan mengungguli pendahulunya. Metode ini menggunakan teknologi grafik dengan transformasi di antara neuron untuk mengembangkan model pembelajaran berlapis banyak. *Deep learning* telah menunjukkan hasil yang menjanjikan di berbagai jenis aplikasi seperti *Natural Language Processing (NLP)*, pemrosesan data visual, pemrosesan ucapan dan audio, dan banyak aplikasi terkenal lainnya (Shyu, Chen and Iyengar, 2020).

Deep learning mendidik sistem komputer untuk menghasilkan hasil dengan melatih mereka pada data sebelumnya. Kemampuan pembelajaran yang mendalam telah memungkinkannya untuk mencapai hasil yang tidak mungkin pada suatu waktu. Banyak teknik pembelajaran mendalam menggunakan arsitektur jaringan saraf, untuk mengklasifikasikan kumpulan data yang diperoleh dan kemungkinan besar disebut sebagai *deep neural network*. Dalam *deep neural network*, kata dalam menunjuk ke jumlah lapisan yang tersembunyi, tidak seperti jaringan saraf tradisional yang dapat menampung dua atau tiga lapisan tersembunyi; jaringan saraf dalam dapat menampung lebih dari seratus lima puluh lapisan tersembunyi. Sehingga dapat mempelajari fitur dari data itu sendiri, menghindari ekstraksi fitur

Pooling layer memiliki fungsi untuk menjaga ukuran data ketika terjadi *down sampling* atau reduksi data. *Pooling* memudahkan representasi data menjadi lebih kecil sehingga mudah dikelola dan mengontrol terjadinya *overfitting*. Banyak metode *pooling* yang bisa digunakan diantaranya yaitu *MaxPooling*, *AveragePooling*, *SumPooling* dan sebagainya. Teknik *pooling* dapat diilustrasikan sesuai pada Gambar 2.16 di bawah ini.



Max pooling mempartisi gambar menjadi sub-wilayah persegi, dan itu hanya mengembalikan nilai maksimum dari nilai sub-wilayah tersebut. Salah satu ukuran yang paling umum yang digunakan dalam *max pooling* adalah 2x2. Seperti dapat dilihat pada Gambar 2.20, ketika *pooling* dilakukan di blok 2x2 kiri atas (area putih), bergerak 2 dan fokus pada bagian kanan atas. Menyatakan operasi *pooling* dilakukan untuk penggabungan. Selain itu, *pooling* dapat digunakan dengan filter dan langkah yang tidak sama untuk meningkatkan efisiensi. Seperti, *max-pooling*

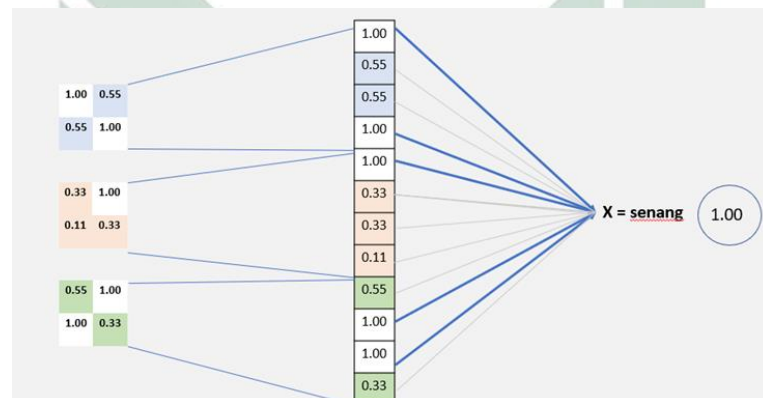
3x3 dengan stride 2 menyimpan nilai tumpang tindih antar sub-wilayah (Albawi, Mohammed and Alzawi, 2017). Perhitungan *max pooling layer* dipaparkan dalam rumus berikut.

$$y_{kij} = (p, q)_{\in R_{ij}}^{max} x_{kpq} \quad (7)$$

Dimana rumus (7) menjelaskan, y_{kij} adalah *output* dari operator *pooling* yang berkaitan dengan *feature map* ke- k . x_{kpq} diartikan sebagai elemen yang ada pada nilai (p,q) sesuai sub-wilayah *pooling*. Dan R_{ij} mewakili nilai yang ada disekitar (i,j). Umumnya, arsitektur CNN menggunakan skema *pooling* konvensional seperti *max pooling* dan *average pooling* untuk mendapatkan nilai maksimum atau rata-rata pada lapisan tersembunyi (Jie and Wanda, 2020).

2.2.7.3 Flatten Layer

Flattening merupakan proses mengubah data menjadi array 1 dimensi satu vektor fitur panjang untuk dimasukkan ke lapisan berikutnya. Setelah memperoleh output hasil dari *feature extraction network*, dilakukan proses *flatten layer* dimana tiap *neuron* dalam *padding* dirubah menjadi satu kolom baris proses ini dikenal dengan sebutan *stacking* (Jeong, 2019).



Gambar 2. 16 Proses perubahan vektor 2D menjadi 1D

Gambar 2.16 menjelaskan proses perubahan vektor 2 dimenjsi menjadi satu dimensi. Dimana beberapa output dari *pooling layer* diratakan ketika penerapan dua atau lebih filter konvolusi. Dropout dilakukan untuk mencegah beberapa variabel diterima berulang kali dalam kombinasi *flatten* karena memungkinkan untuk muncul berulang kali dalam vektor penyatuan yang berbeda. Jika tidak, konversi

flatten hanya digunakan untuk menyalin vektor *pooling-output* ketika menggunakan satu filter konvolusi (Chen *et al.*, 2020).

2.2.7.4 Fully-Connected Layer

Fully-Connected Layer adalah sebuah lapisan dimana seluruh neuron aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya sama seperti halnya dengan *neural network* biasa. Model mampu membedakan antara fitur tingkat rendah yang mendominasi dan mengklasifikasikannya menggunakan teknik Klasifikasi *Softmax*.

Pada dasarnya lapisan ini biasanya digunakan pada *Multi-Layer Perceptron* (MLP) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Perbedaan antara lapisan *Fully connected* dan lapisan konvolusi biasa adalah neuron di lapisan konvolusi terhubung hanya ke daerah tertentu pada *input*, sementara lapisan *Fully-Connected* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk dot, sehingga fungsinya tidak begitu berbeda (Suyanto, Kurniawan Nur Ramadhani, 2019).

Fully connected sepenuhnya terhubung dalam mempelajari kombinasi non-linier dari fitur tingkat tinggi yang diwakili oleh output dari lapisan konvolusi. Gambar *input* dikonversi menjadi bentuk yang sesuai untuk *Perceptron Multi-Level*. Lalu gambar diratakan menjadi vektor kolom. Output tersebut diumpankan ke jaringan saraf forward dan *back propagation* sehingga dapat diterapkan ke setiap iterasi pelatihan. Tahapan *fully connected layer* mendapatkan *input* dari output *flatten layer* untuk tahap *forward propagation* yang menjadi penerimaan pertama. *Forward propagation* dijelaskan dalam persamaan (8) sebagai berikut (Prasetyo *et al.*, 2021).

$$\hat{y} = g(\sum_{i=1}^m x_i w_i + b_i) \quad (8)$$

Dimana:

$$\hat{y} = \text{output}$$

g = fungsi aktivasi

$$\sum_{i=1}^m x_i w_i = \text{kombinasi input linier nilai input dan bobot}$$

b_i = nilai bias ke- i

$$z_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j=1, \dots, K \quad (9)$$
$$\sum_{k=1}^K e^{z_k} = e^{z_k} + e^{z_k} + e^{z_k} + e^{z_k} + \dots n \quad (10)$$

K = vector
 z = bilangan riil ke- j
 e = konstanta
 $\sum_{k=1}^K e^{z_k}$ = hasil penjumlahan e^{z_k}

$$error = \frac{1}{3} \sum_{i=1}^3 (target_i - prediksi_i)^2 \quad (10)$$

$$w_{new} = w_{old} - \alpha \frac{\partial E}{\partial w} \quad (11)$$

$\partial E/\partial w$ = turunan parsial error terhadap weight

[illegible]

nilai klasifikasi yang sesuai. Oleh sebab itu proses ini masuk dalam *fully connected layer*, karena memiliki banyak proses *deep neural network* didalamnya (Prasetyo *et al.*, 2021).

$$b_{new} = b_{new} - \alpha \frac{\partial E}{\partial b} \quad (12)$$

$$\frac{\partial \sigma}{\partial w_{11}} = \frac{\partial E}{\partial y_{11}} \frac{\partial y_1}{\partial z_1} \frac{\partial z_1}{\partial w_{11}} \quad (13)$$

Dimana:

 b = bias
$$\alpha = \text{konstanta}$$

$\partial E/\partial b$ = turunan parsial error terhadap bias

∂ = turunan parsial

$$E = \text{Error}$$
$$y_n = \text{Nilai prediksi}$$

z_1 = Total *input*, weight, dan bias

x_n = matriks *flatten*

$target_i$ = nilai target bilangan biner

$prediksi_i$ = hasil dari *fully connected layer*

2.2.8 Open CV

OpenCV adalah salah satu *library computer vision* yang *open source* di bawah lisensi BSD dan sangat populer digunakan dalam proses Analisa citra gambar. Awalnya OpenCV dikembangkan oleh pusat penelitian Intel di Nizhny Novgorod (Rusia), kemudian sekarang dikelola oleh Itseez (Zufar, 2016). OpenCV mampu memanipulasi, memproses, dan mengedit gambar lebih dari 2500 algoritma. OpenCV mendukung berbagai bahasa pemrograman seperti C, C++, *Python*, Java, dan PHP. Sehingga bisa diterapkan dalam berbagai aplikasi diantaranya adalah *human computer interaction* (HCI), identifikasi dan pengenalan objek, deteksi dan *face recognition*, pengenalan gestur, *tracking* gerakan, proses pada gambar, dan *mobile robotics* (Santoso and Kristianto, 2020). Dalam pengolahan citra ada beberapa teknik yang menggunakan OpenCV diantaranya yaitu (Naveenkumar and Ayyasamy, 2016).

a. *Image Filtering*

Berfungsi untuk memodifikasi dan meningkatkan kualitas citra gambar menggunakan dua jenis proses, yaitu: penyaringan gambar linier dan nonlinier. Dimana penyaringan gambar linier memperoleh nilai output dari kombinasi linear nilai piksel dalam *inputannya*. Sedangkan untuk nonlinier didapatkan nilai output bukan dari fungsi linier *inputnya*.

b. Image Transformation

Teknik ini berfungsi menghasilkan gambar baru lebih baik dari pada gambar *inputan* aslinya, gambar ini dapat dihasilkan dari dua sumber atau lebih. Transformasi gambar menjadi teknik dasar dalam mengimplementasikan operasi aritmatika sederhana ke dalam data gambar.

c. Object Tracking

Object tracking merupakan teknik untuk menemukan suatu atau beberapa objek sesuai dengan urutan gambar. Teknik ini menjadi salah satu komponen yang banyak digunakan dalam berbagai aplikasi *computer vision*, seperti pengawasan, interaksi antara komputer dengan manusia, dan pencitraan gambar.

b. Feature Detection

Teknik ini berfungsi untuk menemukan ciri khusus dari gambar-gambar yang diinputkan secara visual, seperti tepi, garis, atau sudut. *Feauture detection* dapat menemukan struktur algoritma dari gambar yang rumit.

2.2.9 Pygame

Python sebenarnya cukup mumpuni dalam menjalankan *game* dalam waktu kurang dari 30 milidetik. *Game* apa pun yang berjalan secara *real-time* akan memanfaatkan komputer sepenuhnya. Selama beberapa tahun terakhir, ada tren yang menarik dalam pengembangan *game* menuju bahasa tingkat yang lebih tinggi. Biasanya sebuah *game* dibagi menjadi dua bagian besar. Mesin permainan, yang harus secepat mungkin, dan logika permainan, yang membuat mesin benar-benar melakukan sesuatu (Pete Shinnars, 2021).

Python, bukanlah bahasa pertama yang banyak digunakan pemrograman video *game*. Ada beberapa *game* terbesar di luar sana yaitu Battlefield menjadi

2.2.11 Confusion Matrix

	Predicted	
	Negative	Positive
Actual Negative	8, 3, 9	6
Actual Positive	5, 7	5, 5, 5

Callouts: TN (True Negative) at top-left, FP (False Positive) at top-right, FN (False Negative) at bottom-left, TP (True Positive) at bottom-right.

Precision (e.g., 3 out of 4) points to the top row.

Recall (e.g., 3 out of 5) points to the left column.

Gambar 2.18 menjelaskan bentuk matrik dalam pengujian sistem, model ini didukung dalam perhitungan akurasi metode *confusion matrix* sebagai berikut:

Dimana:

ROC adalah grafik dua dimensi di mana TP diplot pada sumbu Y dan laju FP diplot pada sumbu x . Grafik ROC menggambarkan pertukaran relatif antara manfaat (positif benar) dan biaya (positif palsu). Gambar 2.23 menunjukkan grafik ROC dengan lima pengklasifikasi berlabel A melalui E. Pengklasifikasi diskrit adalah pengklasifikasi yang hanya mengeluarkan label kelas. Setiap pengklasifikasi diskrit menghasilkan pasangan (FPR, TPR) sesuai dengan satu titik dalam ruang ROC (Fawcett, 2006).

$$TRP = (TP)/(TP) + (FN) \quad (18)$$

$$FPR = (FP) / (FP) + (TN) \quad (19)$$

$$Accuracy (\%) = (TP) + (TN) / (TP) + (TN) + (FP) + (FN) \quad (20)$$

$$Inaccuracy (\%) = 100 - Accuracy \quad (21)$$

$$Inaccuracy (\%) = 100 - Accuracy \quad (21)$$

TPR = True Positive Rate

FPR = False Positive Rate

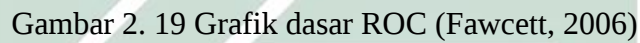
TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

36



Tabel 2. 3 Tingkat nilai akurasi ROC (Rosandy, 2016)

Skala Nilai ROC	Hasil Klasifikasi
0,90 – 1,00	excellent <i>classification</i>
0,80 – 0,90	good <i>classification</i>
0,70 – 0,80	fair <i>classification</i>
0.60 – 0,70	poor <i>classification</i>
0.50 – 0.60	failure

Manusia dapat memperlihatkan perasaan mereka melalui berbagai cara, ada yang melalui intonasi suara, tingkah laku, atau bahkan menunjukkannya melalui mimik wajah. Sistem pengenalan ekspresi wajah dapat digunakan untuk mengetahui secara langsung 'bagaimana perasaan manusia melalui ekspresi yang

Berdasarkan hasil wawancara kedua yang dilakukan dengan pakar bernama Inamul Muttaqien atau yang dikenal sebagai Gus In'am selaku Ketua Pondok Pesantren BPUI Minhajut Thullab, Banyuwangi menjelaskan terkait teknologi dan *games* dalam Islam. Dengan mengatakan bahwa *game* memiliki hukum boleh, yaitu diizinkan untuk dikerjakan (مأذون في فعله). Namun *game* juga dapat dikatakan seperti pisau, jika dipegang seorang jagal kurban maka akan bermanfaat, berbeda artinya jika dipegang oleh pembunuh berbahaya.

مِنْ حُسْنِ إِسْلَامِ الْمَرْءِ تَرْكُهُ مَا لَا يَعْنِيهِ

“Di antara kebaikan islam seseorang adalah meninggalkan hal yang tidak bermanfaat (HR. Tirmidzi)”

Dimana sistem pengenalan ekspresi wajah pemain, dapat meningkatkan kesulitan *game* melalui kecerahan layarnya dengan membuat pemain tersenyum

3.1.2. Studi Literatur

Pada tahap studi literatur yang dimaksud adalah dengan mencari referensi yang relevan sesuai kasus atau permasalahan yang sedang dilakukan. Sehingga ditemukan dengan cara menganalisis kebutuhan sistem dari FER. Studi literatur juga didapatkan metode CNN untuk dapat menyelesaikan permasalahan *classification* pada citra gambar. Model dan algoritma yang digunakan dalam metode CNN untuk *face recognition* tidak hanya mampu untuk mendeteksi ekspresi wajah saja, namun juga dapat merubah tingkat kecerahan layar *game* pada transparansi *masking layer*.

3.1.3. Pengumpulan Data

Desain sistem FER, penting untuk memiliki data pelatihan berlabel yang mencakup banyak jenis populasi dan lingkungan. Pada tahap pengumpulan data ini, *database* yang digunakan berisi ekspresi dasar yang banyak digunakan dalam kasus pengenalan ekspresi wajah yaitu *dataset CK+* dan *dataset custom*.

3.1.3.1. Dataset CK+

Dataset Cohn-Kanade (CK) atau dikenal dengan *Dataset CK* merupakan kumpulan data yang dirilis dengan tujuan mempromosikan penelitian secara otomatis dengan dapat mendeteksi ekspresi wajah pada tiap orang pada saat tahun 2000. Sejak saat itu *Dataset CK* telah menjadi salah satu *test-bed* yang paling banyak digunakan untuk pengembangan dan evaluasi algoritma. Akan tetapi *Dataset CK* memiliki keterbatasan yaitu, kurangnya *matrix* kinerja untuk mengevaluasi algoritma baru, dan belum terdapat standar untuk *dataset*. *Dataset CK+* dapat disajikan pada Gambar 3.3 sebagai berikut (Lucey *et al.*, 2010).



Gambar 3. 3 Sampel *Dataset* CK (Lucey *et al.*, 2010).

Berdasarkan Gambar 3.3, *dataset* CK dapat digunakan untuk deteksi emosi. Agar dalam membuat meta-analisis menjadi lebih mudah maka dapat menggunakan *Extended Cohn-Kanade (CK+)*. Dengan menggunakan CK+ jumlah urutan bertambah 22% dan jumlah subjek sebesar 27%. Serta ekspresi target untuk setiap urutan dapat dikodekan dan mendapatkan label serta divalidasi. Jumlah dan *class* yang ada pada *dataset* CK, ditunjukkan pada Tabel 3.1 di bawah ini (Lucey *et al.*, 2010).

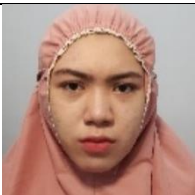
Indeks	Ekspresi	Data <i>Train</i>
0	<i>Angry</i>	153
1	<i>Happy</i>	207
2	<i>Neutral</i>	138
Total Data		498

Tabel di atas menjelaskan total *dataset* CK+ yang sudah dikurang dari 7 jenis ekspresi yang berbeda, dan ditandai dengan 3 label klasifikasi yang berbeda mulai dari label indeks 0 sampai dengan 2. Perubahan *dataset* ini berhubungan dengan *dataset custom* untuk dilakukan perbandingan performa dari kedua data tersebut. Kelas *contempt* dan *sadness* menjadi kelas *neutral*. *Disgust* masuk dalam kelas *angry*, dan beberapa data *surprise* masuk kedalam kelas *happy*. Sehingga dari 981 total data yang digunakan sebanyak 498 data dipilih dalam penelitian. Penulis menerapkan sistem yang dapat mengklasifikasikan 3 ekspresi wajah pada *Dataset* CK+, yaitu *angry*, *happy*, dan *neutral* (Zahara et al., 2020).

3.1.3.2. Data Manual

Pengumpulan data ini berupa citra wajah yang digunakan sebagai *data testing* pengujian model. Pengambilan citra ini menggunakan kamera *webcam* untuk memperoleh total kelas wajah yang dibutuhkan yaitu ekspresi *angry*, *happy*, dan *neutral*. Setiap ekspresi dideteksi berdasarkan titik-titik yang mengukur batas bentuk ekspresi yang ditampilkan. Spesifikasi data yang akan digunakan harus memenuhi kriteria yang terdapat pada tabel berikut:

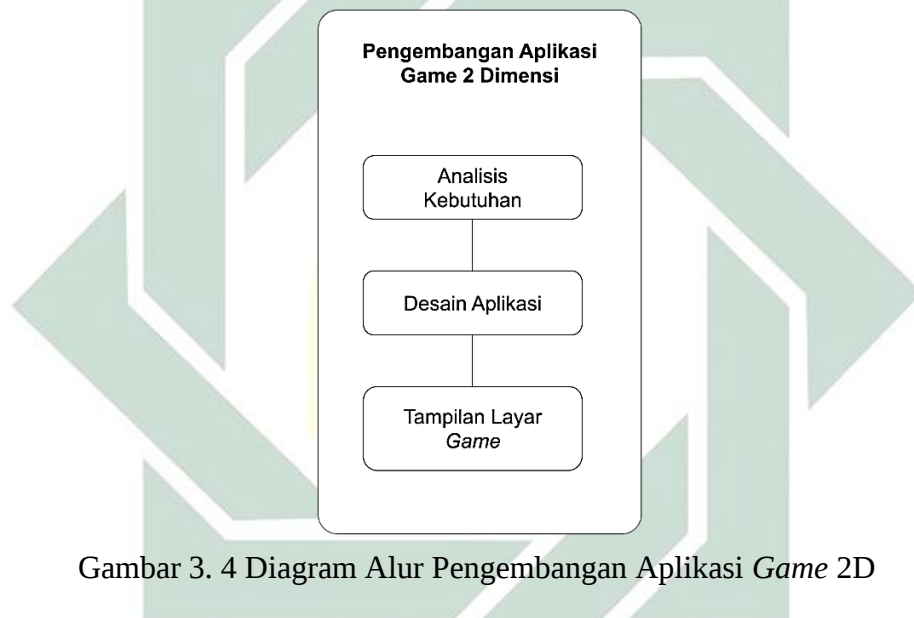
Tabel 3. 2 Jenis Pengumpulan Data

No.	Jenis Citra	Sampel Hasil	Keterangan	Jumlah Data
1.	Angel kamera dari depan	  	Posisi pengambilan foto menghadap kamera dengan lurus dan dengan menampilkan 3 ekspresi	3
2.	Angel kamera dari samping kanan	  	Posisi pengambilan foto menghadap kamera sedikit ke kanan dengan menampilkan 3 ekspresi.	3
3.	Angel kamera dari samping kiri	 	Posisi pengambilan foto menghadap kamera sedikit ke kiri dengan menampilkan 3 ekspresi.	3

mahasiswa yaitu 15, sehingga akan ada $10 \times 15 = 150$ *dataset*. Data *custom* akan di *convert* kedalam ukuran 48×48 *pixel*. Wajah tersebut akan masuk ke dalam sistem dan dilakukan pengujian data pada *data testing*.

3.1.4. Pengembangan Aplikasi *Flappy Bird*™

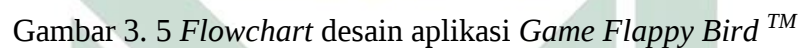
Alur pengembangan aplikasi *game* 2D *Flappy Bird* TM digambarkan ke dalam bentuk diagram alur pada Gambar 3.4. Diagram ini digunakan dengan tujuan untuk mempermudah penyampaian informasi terkait pembuatan *game* yakni dilihat dari analisis kebutuhan, desain aplikasi dan pengaturan tampilan layar *game* yang akan diintegrasikan dengan FER.



Gambar 3. 4 Diagram Alur Pengembangan Aplikasi *Game* 2D

3.1.4.1.Desain Aplikasi

Desain aplikasi dapat dilakukan dengan beberapa konsep dasar dengan tujuan aplikasi ini tidak melebar ke arah yang tidak diperlukan. Terdapat poin-poin dalam pembuatan desain aplikasi *game*, tiga diantaranya *game flow* berisi alur atau *outline* jalannya permainan, *character concept* terkait penjelasan tentang tokoh atau karakter utama, dan *gameplay mechanics* yang menjelaskan mekanisme permainan berlangsung. Penelitian ini menggunakan *Game Flappy Bird™* yang dapat diunduh dari *source code open source github* secara bebas oleh Sourabhv Remove iCCP Chunks pada link berikut, <https://github.com/sourabhv/FlapPyBird>. Desain aplikasi *game* di paparkan pada gambar *flowchart* di bawah ini.



3.1.4.2. Analisis Kebutuhan Aplikasi

49

Alat dan bahan penelitian skripsi adalah sebagai berikut:

Hardware pendukung diantaranya:

- Laptop dengan spesifikasi: *Windows 10 Home 64-bit*, CPU *Intel Core i5-8250U* 2.20 GHz dengan RAM 8 GB DDR3, dan LCD Screen 15.6 inch
- Kamera *webcam* yang digunakan beresolusi 720 x 480 (HD).

Software Pendukung diantaranya :

- Bahasa pemrograman *python* versi 3.7 (64-bit)
- *Notepad*
- *Library PyGame, OpenCV, dan Tensorflow*

3. Tampilan Layar *Game*

Pengaturan kecerahan layar aplikasi *game* diimplementasikan dengan menggunakan *masking layer* warna hitam yang ditentukan berdasarkan skala level kecerahan layar komputer. Parameter kecerahan layar ditentukan dalam satuan persentase. Ketika skor permainan ekspresi pemain. Layar *game* dikatakan cerah ketika mendekati nilai maksimum dalam *game*, layar ditutupi oleh *masking layer* warna hitam. Selanjutnya, ketika skor permainan ekspresi pemain.

a. *Hardware* pendukung diantaranya:

- b. *Software* Pendukung diantaranya :

- #### 3.1.4.3. Tampilan Layar *Game*

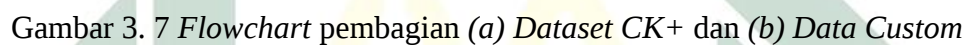
Pemain yang memiliki ekspresi marah dapat menampilkan layar *game* dengan tingkat kecerahan 5% yang berarti sangat gelap yaitu $100\% - 5\% = 95\%$ kepekatan layar hitam. Untuk ekspresi netral dengan tingkat kecerahan 50% menampilkan kecerahan yang sedang. Sedangkan ekspresi pemain dengan wajah tersenyum bahagia menyebabkan *layer game* menjadi sangat cerah dengan tingkat kecerahan 100%.

Setelah aplikasi *Game Flappy Bird* TM dibuat, selanjutnya melakukan pengembangan aplikasi dengan teknologi AI. Sistem ini digambarkan ke dalam bentuk diagram alur pada Gambar 3.5 dengan tujuan untuk mempermudah penyampaian informasi terkait pengembangan aplikasi dan sebagai fitur dalam meningkatkan kesulitan bermain *game*.



3.1.5.1. Pembagian *Dataset*

[illegible]

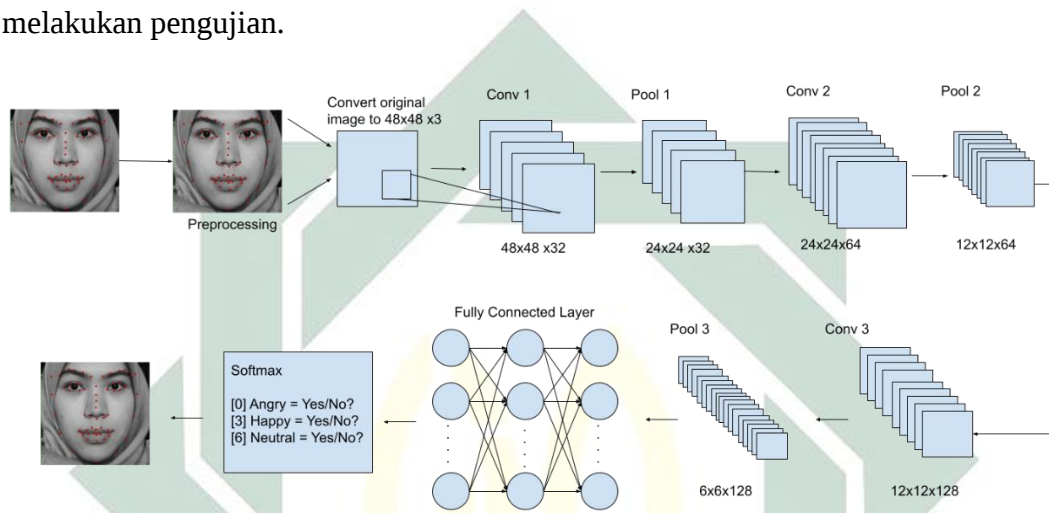


3.1.5.2. Preprocessing Data

52

3.1.5.3. Implementasi Aplikasi

Pada langkah ini sistem sudah dianalisis sesuai pada langkah-langkah sebelumnya. Proses akhir dari pengembangan sistem dilakukan pengkodean program sebelum proses pengujian dilakukan. Aplikasi *Game Flappy Bird™* dan algoritma FER dibuat dengan bahasa pemrograman *python* dengan modul *pygame* untuk aplikasi *game*, dan algoritma CNN untuk mendeteksi ekspresi wajah. Sistem ini didukung dengan program *Tensorflow keras*, sehingga memudahkan dalam melakukan pengujian.



Gambar 3. 8 Implementasi CNN

Gambar 3.8 menjelaskan deteksi ekspresi wajah dilakukan setelah melalui *proses training*. Proses ini dilakukan setelah aplikasi sudah berhasil berjalan. Dimana data awal berupa gambar yang sudah dikonversi pada tahap *preprocessing* dan klasifikasi sebesar 48x48 pixel setiap data. Proses konvolusi dilakukan selama tiga kali, sehingga menghasilkan *input* untuk *fully connected layer*. *Softmax* digunakan pada *layer output* sebagai fungsi aktivasi. *Layer output* memiliki banyak kesamaan dengan *fully connected layer*, yang membedakan adalah penggunaan fungsi aktivasi *softmax* hanya ada pada *layer output*. *Softmax* dipilih karena jumlah kelas *dataset* lebih dari dua (*multi-class*) (Xu et al., 2019). Kemudian, masing-masing gambar diberi label sesuai dengan jenis kelasnya (*neutral*, *happy*, dan *angry*) beserta *bounding box* yang digunakan untuk deteksi.

Pada tahap ini sistem *face expression* dan *game* sudah jadi. Sistem melakukan identifikasi wajah dari masing masing individu yang mempunyai hak akses, dengan mendeteksi garis wajah, mata, hidung, mulut dengan algoritma dan pengembangan yang sudah dilakukan. Pengujian dapat dilakukan secara individu, dengan kamera *webcam* resolusi 720 x 480 *Pixels Resolution Video*, dan menggunakan *Personal Computer* dengan spesifikasi *Operating System Windows* 10 64-bit.

Pengujian ini menggunakan model pengujian *confusion matrix* untuk mengukur kinerja algoritma yang diimplementasikan. Serta menilai peningkatan akurasi wajah dengan pengukuran kinerja algoritma CNN dengan metode *Receive Operating Characteristic (ROC) Curve* dan pengujian performa menggunakan *n-Fold Cross validation*. Pengujian dengan model *ROC Curve* pada aplikasi *Game Flappy Bird™* digambarkan pada tabel dan gambar sebagai berikut.

Testing	Face expression recognition			
	TPR	FPR	FNR	TNR
Average				

54

dalam rumus (12) bab 2.2.17. Sehingga sistem mampu secara cepat dalam mengidentifikasi citra. Kemudian, pengujian performa algoritma CNN dilakukan menggunakan metode *Cross validation*, adalah metode pengambilan sampel ulang data untuk menilai kemampuan generalisasi model prediktif dan untuk mencegah *overfitting*. Contoh metode *Cross validation* yang umumnya digunakan dan digunakan dalam penelitian ini adalah *K-Fold Cross validation*.

K-Fold Cross validation melakukan pengujian pada seluruh gambar dalam kumpulan data dengan semua jaringan selama 100 *epoch* karena terbatasnya ukuran pelatihan *data custom*. Set pembelajaran yang tersedia dipartisi menjadi k himpunan dengan ukuran yang kira-kira sama. *Fold* mengacu pada jumlah himpunan yang dihasilkan untuk pengambilan sampel kasus secara acak dari set pembelajaran tanpa pengembalian. *Fold cross validation* memiliki persamaan sebagai berikut.

$$\hat{\epsilon}_{cv} = \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{f} - k(x_i)) \quad (22)$$

Dimana nilai $\hat{f}$ menunjukkan model yang dilatih kecuali bagian *data training*. $y_i = \hat{f} - k(x_i)$ adalah nilai yang diprediksi atau diperkirakan untuk label kelas sebenarnya, dan n adalah nilai banyaknya fold yang dibutuhkan untuk validasi. Penelitian ini menggunakan 1 hingga 5 k untuk dilatih menggunakan k 1 *subset*, yang bersama-sama mewakili himpunan pelatihan. Kemudian, model tersebut diterapkan pada *subset* yang tersisa, yang dilambangkan sebagai set validasi, dan kinerjanya diukur. Prosedur ini diulang sampai masing-masing himpunan bagian telah berfungsi sebagai himpunan validasi. Rata-rata dari pertunjukan pengukuran pada k set validasi adalah kinerja yang divalidasi silang. Dimana *1-fold*, berfungsi sebagai set validasi pada *data training* 1 dan sembilan *subset* yang tersisa berfungsi sebagai *data test* 1. Pada *2-fold*, set validasi dan *subset* yang tersisa adalah set pelatihan, dan seterusnya (Berrar, 2018).

3.1.7. Evaluasi Hasil

Saat memulai permainan, IDLE Python program *Flappy Bird* dijalankan hingga muncul tampilan layar *game*. Kemudian *player* mulai menjalankan panggilan awal atau “start” untuk memulai *game*. Sistem secara otomatis menghidupkan kamera *webcam* dan melakukan klasifikasi wajah pemain dengan menampilkan label ekspresinya. Proses pengenalan ekspresi wajah dihasilkan dari metode *convolutional neural network (CNN)*. Ekspresi yang ditampilkan, diterima oleh sistem untuk mengatur *masking layer Game Flappy Bird™*. Proses ini dilakukan menggunakan memori lokal komputer.

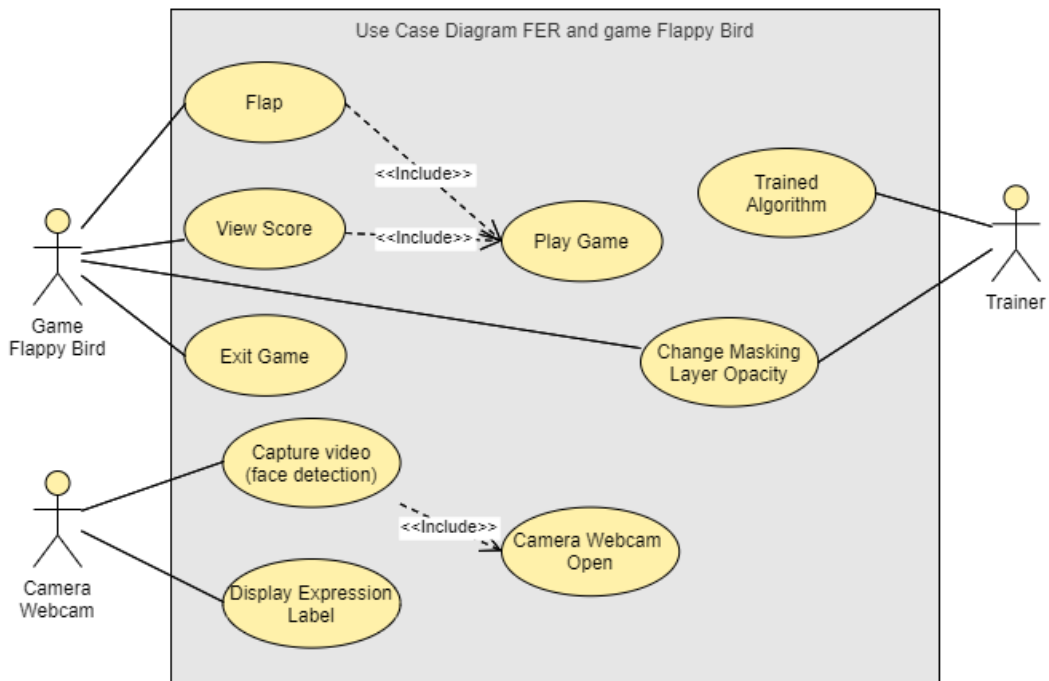
Semakin tersenyum *user*, semakin tipis *masking layer* yang

Use Case Diagram

se diagram merupakan UML yang memaparkan scenario ja
pengguna maupun sistem yang berhubungan satu dengan lainnya
scenario aktor memiliki proses yang berbeda. Namun, per
membuat proses antar sistem saling terhubung. Rancangan u
dari beberapa fitur yang menjelaskan actor ada tiga sisi sepe
da Gambar 4.1. Dimana aplikasi ini menggambarkan tiga acto
BirdTM, *trainer*, dan *camera webcam*.

Use Case Diagram

BirdTM, trainer, dan camera webcam.



Gambar 4. 4 *Use Case Diagram*

activity diagram

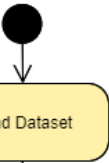
diagram menjelaskan aktifitas yang ada pada s

pada aplikasi ini, sistem mmeiliki tiga arsitektur diag

ki fungsi yang berbeda, diagram tersebut diantaranya

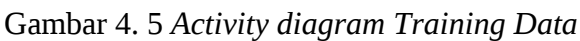
dengan FER, dan *masking layer* pada *game*. Pengg

ng data terlihat pada Gambar 4.5 sebagai berikut.



```
graph TD; Start(( )) --> ReadDataset[Read Dataset];
```

Activity diagram menjelaskan aktifitas yang ada pada sistem dari awal hingga akhir. Pada aplikasi ini, sistem memiliki tiga arsitektur diagram, dimana tiap proses memiliki fungsi yang berbeda, diagram tersebut diantaranya *training* data, pelabelan data dengan FER, dan *masking layer* pada *game*. Penggambaran *Activity diagram training* data terlihat pada Gambar 4.5 sebagai berikut.



default *expression* yaitu *neutral* dan jika tidak maka sistem tidak akan menampilkan apapun pada layar. Jika *expression* label ditampilkan, sistem harus dapat digunakan sebagai variabel pemicu *macro* *Bird*TM. Selanjutnya memasukkan sistem tersebut ke dalam *Bird*TM yang dijelaskan pada Gambar 4.7.

expression label ditampilkan, sistem harus dapat dapat untuk digunakan sebagai variabel pemicu *main BirdTM*. Selanjutnya memasukkan sistem tersebut ke *BirdTM* yang dijelaskan pada Gambar 4.7.



Gambar 4. 7 Activity diagram Masking layer Game

Activity diagram Gambar 4.7 memaparkan sistem dengan tiga program yang saling terhubung. *Player*, *game interface*, dan *camera webcam* semua beroperasi bersamaan. Proses ini diawal dari *trainer* atau *player* yang menjalankan *Game*

game berhenti atau *game over*.

5 UML Sequence diagram

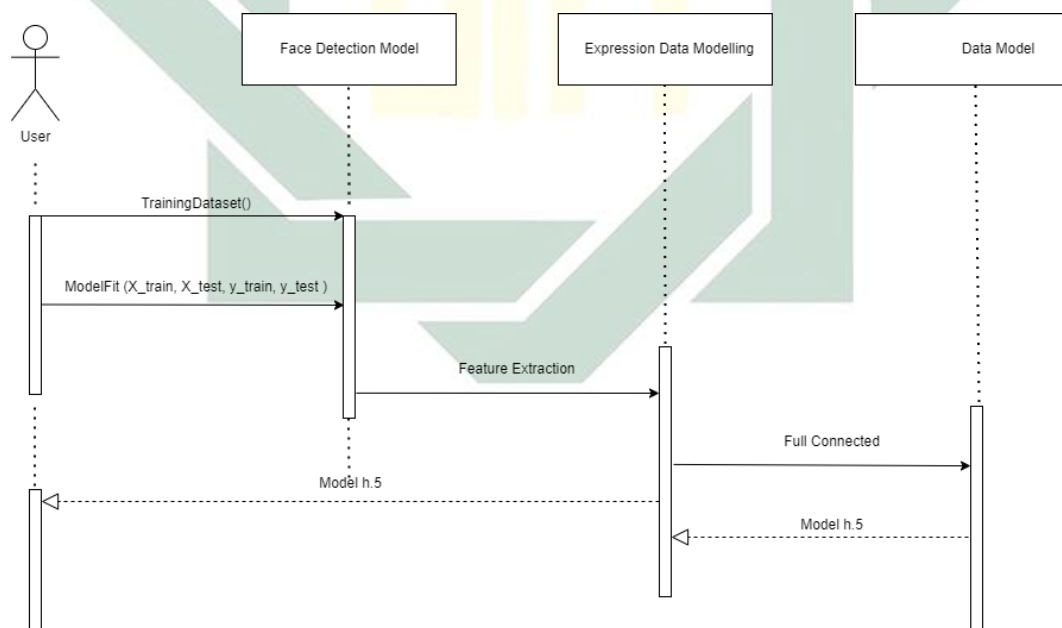
Sequence diagram merupakan salah satu diagram UML yang berfokus untuk menjelaskan interaksi atau jalannya antar proses satu ke proses selanjutnya dalam suatu sistem secara rinci. Pada penelitian ini terdapat 3 model yang digambarkan dengan diagram *sequence training dataset*, *face expression recognition* dan *working layer*. 3 model tersebut di tunjukkan pada gambar dibawah ini.

```
sequenceDiagram
    participant TD as TrainingDataset()
    participant FDM as Face Detection Model
    participant EDM as Expression Data Modelling
    participant D as Data

    TD->>FDM: 
    FDM-->>EDM: 
    EDM-->>D: 
```

4.3.5 UML Sequence diagram

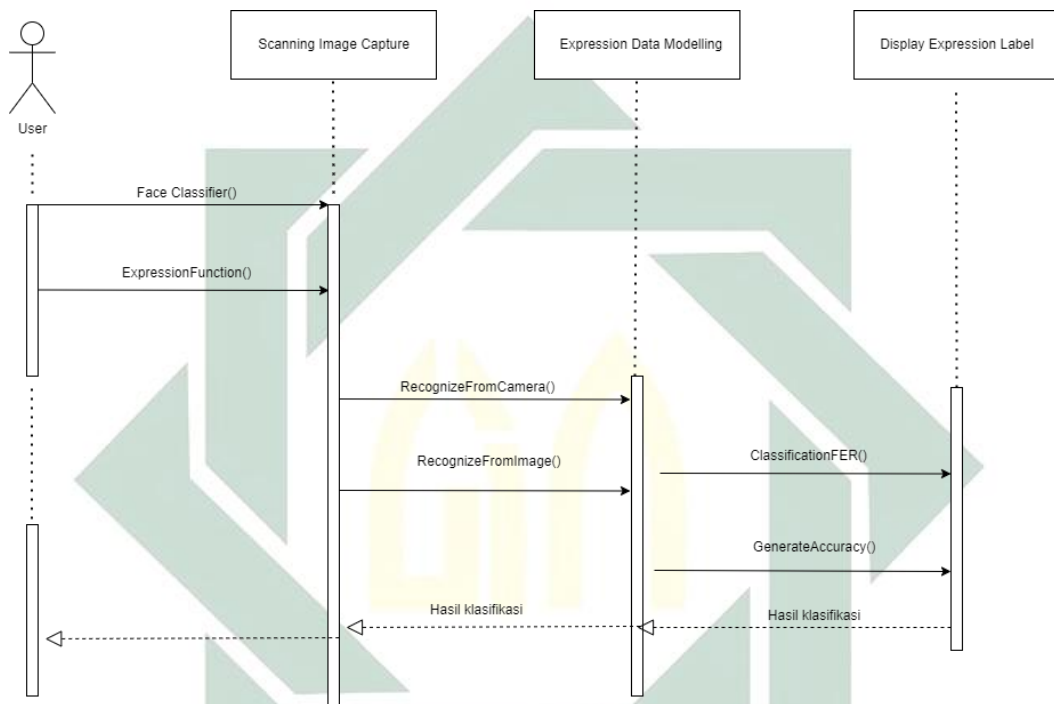
Sequence diagram merupakan salah satu diagram UML yang berfungsi untuk menjelaskan interaksi atau jalannya antar proses satu ke proses selanjutnya sebuah sistem secara rinci. Pada penelitian ini terdapat 3 model yang digambarkan dengan diagram *sequence training dataset*, *face expression recognition*, dan *masking layer*. 3 model tersebut di tunjukkan pada gambar dibawah ini.



Gambar 4. 8 *Sequence diagram Training Dataset*

Sequence diagram pertama ditunjukkan pada Gambar 4.8 yang menjelaskan bahwa proses *training data* ini memiliki tiga *object* yaitu, *face detection model* (deteksi terhadap variabel gambar x dan y), *expression data modelling* (proses

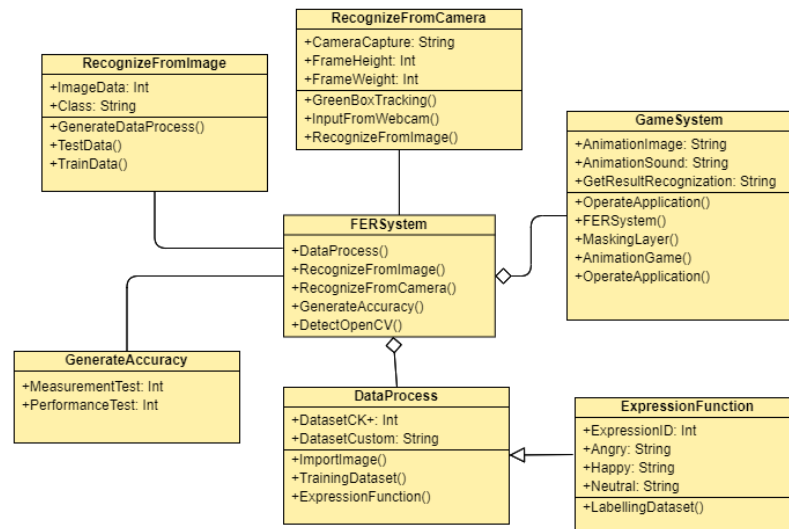
Setelah proses *training dataset* dilakukan, *user* dapat melakukan proses deteksi. Proses ini di gambarkan pada *sequence diagram face expression recognition* Gambar 4.9 sebagai berikut.



ar 4.9 menunjukkan terdapat tiga *object* dan satu *user* dalam sistem
 es ini menjadi proses yang paling utama, karena ekspresi wajah
 dapat di tangkap oleh kamera untuk dideteksi. Selain itu proses ini
 kan bagaimana proses *face recognition* dalam melakukan
 data menggunakan metode CNN. Pada *object* pertama yaitu *scanning*
re menunjukkan bahwa *object* menerima data dari *user* berupa
nction (ekspresi wajah *user*) untuk dilakukan *faceClassifier()*. Setelah
 il melakukan *scanning* wajah, *expression data modelling* menerima
 ri kamera dan gambar *data train*. Sistem secara otomatis melakukan
 spresi dan melakukan proses *generateaccuracy()*.

4.3.6 UML Class diagram

Class diagram adalah salah satu diagram UML yang berfungsi untuk merancang *class* dan *method* yang digunakan dalam aplikasi. Struktur *class* divisualisasikan ke dalam Gambar 4.11 di bawah ini.



Gambar 4. 11 *Class Diagram*

Gambar 4.11 menjelaskan rancangan *class diagram* aplikasi secara menyeluruh dengan total 7 *class*. Beberapa *class* dibagi kedalam bentuk *front end* dan *back end*. Didalam *front end* terdiri dari kelas *Game System* dan *RecognizeFromCamera*, sedangkan pada bagian *back end* terdiri dari *class ExpressionFunction*, *DataProcess*, *GenerateAccuracy*, *RecognizeFromImage*, dan *FERSystem*. Setiap bagian *class* memiliki fungsi yang berbeda, berikut pemaparan deskripsi fungsi dari setiap *class*.

a. *ExpressionFunction*

Pada bagian *expressionfunction* terdapat *attribute expression* dengan tipe data integer yang berisi pixels data gambar. *Attribute angry*, *happy*, dan *neutral* diperoleh dari *labeling expression* pada *dataset* yang masing-masing memiliki tipe data string dengan relasi *one to many*.

b. *DataProcess*

Class ini berisikan dua proses utama yaitu `importimage()` dan `TrainingDataset()`, serta satu proses yaitu `ExpressionFunction()` yang tidak mampu berjalan tanpa *class* tersebut. *Class* `DataProcess()` memiliki *attribute*

dari beberapa jenis variabel yang digunakan yaitu X_{train} , y_{train} , dan X_{test} , y_{test} dengan tipe data float.

c. *RecognizeFromImage*

Class ini memiliki beberapa *attribute* yaitu *Class* dengan tipe data *string*, *Emotion* dengan tipe data *integer* dan membutuhkan masukan proses dari *class* lain untuk melakukan *trainData()* dan *testData()*.

d. *FERSystem*

Class ini memiliki fungsi untuk menjalankan perintah yang sudah dilakukan *class* sebelumnya. *Class* ini terdiri dari banyak proses diantaranya *DataProcess()*, *RecognizeFromCamera()*, *RecognizeFromImage()*, *GenerateAccuracy()*, dan *DetectOpenCV()*. Proses dalam *class* ini dilakukan untuk mendeteksi data yang sudah di olah pada proses sebelumnya. *DetectOpenCV()* salahsatu proses yang berfungsi untuk memanggil model opencv.

e. *RecognizeFromCamera*

Class tersebut berisikan proses `boundingboxTracking()`, `InputFromWebcam()`, dan `RecognizeFromImage()`. Dimana tiap proses menggunakan *attribute* `cameracapture`, `frameHeight`, dan `frameWeight` untuk mengolah data yang diambil dari kamera *webcam*. Tipe data yang digunakan dalam keseluruhan proses ini adalah integer. Namun untuk `cameracapture` dibutuhkan tipe data string untuk mengoperasikan label data.

f. *Game System*

Pada *class* ini terdiri dari *attribute* berupa *animationImage*, *animationSound*, dan *getResultRecognition* yang memiliki sifat *public* dengan model tipe data *string*. Masing-masing *attribute* dioperasikan dalam bagian proses frontend aplikasi dengan method *operateApplication()*, *FERSystem()*, *MaskingLayer()* dan *AnimationGame()*.

g. *GenerateAccuracy*

Class ini berisikan dua proses yaitu *MeasurementTest()* dan *PerformanceTest()*. Dimana tiap *test* yang digunakan bekerja untuk melakukan pengujian pada data yang sudah diproses dalam sistem. Kedua proses ini berjalan dengan tipe data integer.

P_n) merupakan total *feature* map ke- n , dimana perkalian data dengan 3×3 filter dan b adalah nilai bias data. Persamaan ini di hitung pada gambar 3.1. Hasil dari persamaan tersebut sudah dikonversi kedalam *binary matrix* yang ditunjukkan pada Gambar 3.2.

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	2	5	7	3	4	3	6	4	10	8	15	2	3	8	4	11	1	1	1
1	1	1	11	14	5	3	4	17	16	14	23	24	17	3	5	1	8	1	1	1
1	1	1	5	5	4	5	5	22	22	20	14	19	18	17	8	8	1	1	1	1
1	1	1	3	6	10	14	25	22	24	19	21	23	20	20	19	3	1	1	1	1
1	1	7	4	8	23	18	23	19	18	24	22	19	25	22	19	3	1	1	1	1
1	1	1	14	12	15	17	22	20	20	20	23	20	21	23	20	7	6	1	1	1
1	1	1	3	7	14	17	16	25	19	17	20	19	17	24	23	7	2	5	1	1
1	1	4	2	22	23	14	13	12	23	20	16	23	18	23	25	5	3	8	1	1
1	2	2	5	8	25	15	24	15	25	17	19	22	23	25	24	6	3	3	12	7
1	1	1	4	5	24	11	14	14	24	13	20	23	17	23	23	3	2	2	1	1
1	1	1	7	3	23	25	23	18	23	18	23	19	14	19	15	10	3	2	1	1
1	1	1	1	4	15	11	13	19	15	24	22	19	25	22	20	3	2	5	2	1
1	1	1	14	12	20	17	21	23	20	23	18	23	19	18	24	4	1	1	1	1

Gambar 4. 14 *Feature Matrix 3 Filter*

$$\begin{aligned}
 FP1 &= [(0.(-1)) + (0.(-1)) + (0.1) + (0.0) + (0.1) + (0.(-1)) + (0.0) + (0.1) + (0.1)] + \\
 &\quad [(0.1) + (0.0) + (0.0) + (0.1) + (0.(-1)), (0.(-1)) + (0.1) + (0.0) + (0.(-1))] + \\
 &\quad [(0.0) + (0.1) + (0.1) + (0.0) + (0.1) + (0.0) + (0.1) + (0.(-1)) + (0.1)] = [0] \\
 &\quad + [0] + [0] + 1 = 1
 \end{aligned}$$

73

4.4.3. Pooling layer dan Flattening

Pooling layer merupakan proses lanjutan dari tahap convolution. Pooling layer memiliki dua jenis diantaranya average pooling dan max pooling. Dulunya average pooling sangat populer digunakan, namun saat ini penggunaan max pooling lebih banyak digunakan saat ini karena max pooling dapat berkinerja dengan baik. Average pooling dikatakan tidak berkinerja dengan baik karena layer ini menghitung padding berdasarkan nilai rata-rata. Dimana perhitungan ini dapat menghilangkan lebih sedikit informasi dari pada max pooling.

Berbeda dari average, pooling layer jenis max dapat mempertahankan nilai yang lebih kuat dan menghilangkan nilai yang tidak berarti. Sehingga lapisan berikutnya mendapatkan nilai yang lebih bersih. Selain itu, max pooling dapat memilih nilai invariant yang lebih kuat dari pada average pooling (Aurélien Géron, 2019). Oleh sebab itu, penelitian ini menggunakan *pooling* dengan jenis max yaitu mengambil nilai tertinggi sesuai besar ukuran padding yang ditentukan.

Maps di proses *max pooling* menggunakan padding ukuran matriks 2x2. Hasil pada tahap ini ditunjukkan pada Gambar 4.15 sebagai berikut.

1	5	7	4	6	10	15	8	1	1	1
1	22	14	5	22	23	19	18	8	1	1
1	7	23	25	24	24	25	22	3	1	1
1	14	15	22	25	23	21	24	7	5	1
2	5	25	24	25	20	23	25	6	12	7
1	7	24	25	24	23	23	23	10	2	1
1	14	20	21	23	24	25	24	4	5	1
1	14	24	22	25	24	23	25	19	1	1
1	4	25	24	25	24	25	25	29	24	1
1	6	25	23	25	18	21	22	19	1	1
1	1	23	24	25	24	23	18	20	17	1
1	13	25	18	23	24	22	25	19	8	5
1	11	23	5	29	24	25	19	24	25	25

Max Pooling

1
5
7
4
6
10
15
8
1
1
~
25

Flattening

Gambar 4. 15 Hasil *Max Pooling* dan *Flattening*

Gambar 4.15 menampilkan nilai atau bobot yang memiliki nilai paling tinggi berdasarkan ukuran padding. Hasil *max pooling* di *input* ke tahap selanjutnya yaitu *flattening*. Dimana pada tahap *flattening*, ukuran matriks terakhir dirubah menjadi matriks 1 dimensi. Perubahan matriks berfungsi agar saat melakukan

$$\begin{aligned} e^{a_i} &= 2.7^{9.8} + 2.7^{10.4} + 2.7^9 \\ &= 16.879,706414635 + 30.632,546731394 + 7.625,597484987 \\ &= 55.137,85363002 \end{aligned}$$

$$\begin{aligned} softmax &= \left[\frac{2.7^{9,8}}{55.137,853630029} \frac{2.7^{10,4}}{55.137,853630029} \frac{2.7^9}{55.137,853630029} \right] \\ &= [0.3, 0.5, 0.1] \end{aligned}$$

Selanjutnya melakukan perhitungan *error* dengan nilai target $t_1 = 0, t_2 = 0.25$, dan $t_3 = 1$ untuk mengetahui hasil prediksi dari persamaan *mean square error*. Berdasarkan persamaan *error* dapat dilakukan perhitungan dengan proses perhitungan sebagai berikut.

$$E = \frac{1}{3} ((0 - 0.3)^2 + (0.25 - 0.5)^2 + (1 - 0.1)^2)$$
$$= \frac{1}{3} 0.09 + 0.0625 + 0.81 = \frac{0.9625}{3} = 0.32$$

Proses perhitungan nilai *error E* menghasilkan nilai sebesar 0.32. Perhitungan selanjutnya, yaitu melakukan *back propagation* untuk mengoptimalkan proses *neural network* pada tahap *fully connected layer*. Perhitungan menentukan nilai bobot *w* dan nilai bias *b* berdasarkan nilai *error* dipaparkan pada proses perhitungan sebagai berikut.

$$\begin{aligned}\frac{\partial \sigma}{\partial w_{11}} &= -\frac{2}{3}(t_1 - y_1) \cdot y_1(1 - y_1) \cdot x_1 \\ \frac{\partial \sigma}{\partial w_{11}} &= -\frac{2}{3}(0 - 0.3) \cdot 0.3(1 - 0.3) \cdot 1 = 0.042 \\ \frac{\partial \sigma}{\partial w_{12}} &= -\frac{2}{3}(0.25 - 0.5) \cdot 0.5(1 - 0.5) \cdot 5 = 0.208 \\ \frac{\partial \sigma}{\partial w_{13}} &= -\frac{2}{3}(1 - 0.1) \cdot 0.1(1 - 0.1) \cdot 7 = 0.378\end{aligned}$$

Selanjutnya memperbarui nilai bobot dengan konstanta 0.5 dalam perhitungan sebagai berikut.

$$\begin{aligned} b_{11} &= 0.1 - (0.5) \cdot (0.042) = 0.079 \\ b_{11} &= 0.2 - (0.5) \cdot (0.208) = 0.096 \\ b_{11} &= 0.3 - (0.5) \cdot (0.378) = 0.489 \end{aligned}$$

4.4.5. Hasil FER *Training Dataset*

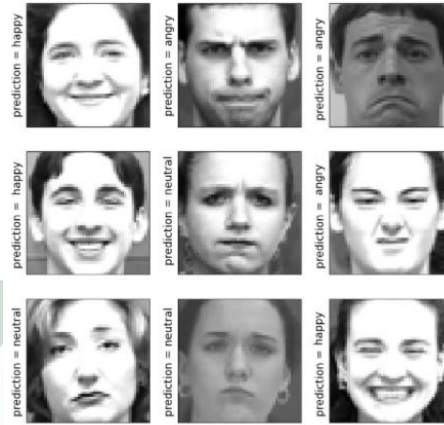
14/14	[=====]	- 0s 18ms/step	- loss: 0.0488	- accuracy: 0.9764	- val_loss: 2.6645	- val_accuracy: 0.7333
Epoch 98/100						
14/14	[=====]	- 0s 19ms/step	- loss: 0.0451	- accuracy: 0.9811	- val_loss: 2.3923	- val_accuracy: 0.7200
Epoch 99/100						
14/14	[=====]	- 0s 18ms/step	- loss: 0.0459	- accuracy: 0.9740	- val_loss: 2.6123	- val_accuracy: 0.7200
Epoch 100/100						
14/14	[=====]	- 0s 18ms/step	- loss: 0.0593	- accuracy: 0.9716	- val_loss: 2.4874	- val_accuracy: 0.7200



78

validasi sangat rendah. Hal ini menyebabkan terjadinya *overfitting* data, maka diperlukan *optimizer Adam*. *Dataset CK+* ditampilkan pada gambar di bawah ini

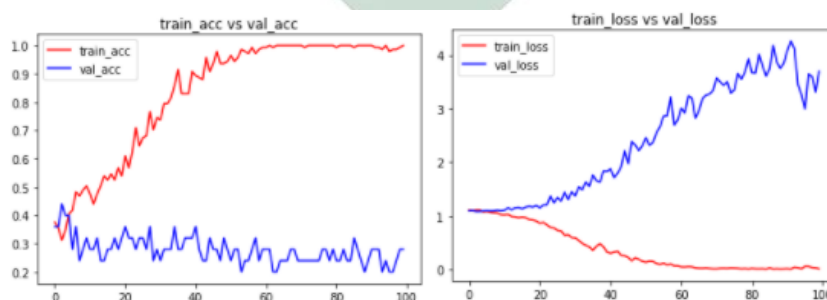
```
Test Loss: 2.4874308109283447
Test accuracy: 0.7200000286102295
(1, 48, 48, 3)
[[6.3598016e-10 1.0000000e+00 9.1228927e-11]]
[[0 1 0]]
[[0. 0. 1.]]
```



Gambar 4. 17 Accuracy dan Loss Data Test CK+

Hasil output dari *training dataset* CK+ yaitu untuk memperoleh model data baru yang akan digunakan pada pemrograman FER. Gambar 4.17 menjelaskan citra wajah hasil training data. Proses ini menghasilkan nilai test dengan loss sebesar 24,8% dan akurasi test sebesar 72%. Selain itu, training data juga dilakukan pada *dataset custom*, dijelaskan sebagai berikut.

```
Epoch 58/100
5/5 [=====] - 0s 23ms/step - loss: 0.1065 - accuracy: 0.9716 - val_loss: 3.2211 - val_accuracy: 0.3200
Epoch 59/100
5/5 [=====] - 0s 24ms/step - loss: 0.0728 - accuracy: 0.9858 - val_loss: 2.6980 - val_accuracy: 0.2400
Epoch 60/100
5/5 [=====] - 0s 25ms/step - loss: 0.0704 - accuracy: 0.9929 - val_loss: 2.7910 - val_accuracy: 0.2800
Epoch 61/100
5/5 [=====] - 0s 23ms/step - loss: 0.0523 - accuracy: 0.9929 - val_loss: 3.0105 - val_accuracy: 0.2800
Epoch 62/100
5/5 [=====] - 0s 23ms/step - loss: 0.0508 - accuracy: 1.0000 - val_loss: 2.9227 - val_accuracy: 0.2800
```



Gambar 4. 18 *Accuracy dan Loss Data Training Custom*

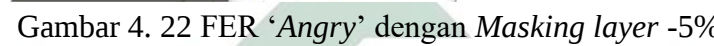
Gambar 4.18 menjelaskan bahwa proses training data custom mampu memperoleh akurasi sebesar 100% dengan loss sebesar 5%. Dalam grafik diatas juga dijelaskan bahwa loss pada data validasi sangat tinggi, sedangkan akurasi pada

```
Test Loss: 3.7026467323303223
Test accuracy: 0.2800000011920929
(1, 48, 48, 3)
[[6.3528911e-05 9.9990404e-01 3.2399992e-05]]
[[0 1 0]]
[[0. 1. 0.]]
```



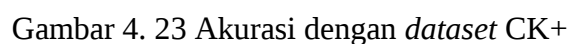
Hasil output dari *training dataset* custom yaitu untuk memperoleh model data baru yang akan digunakan pada pemrograman FER. Gambar 4.19 menjelaskan citra wajah hasil *training data* menghasilkan nilai test dengan loss sebesar 37% dan akurasi test sebesar 28%. Dimana nilai validasi sangat kecil jika dibandingkan dengan besar akurasi dan loss data train yakni sebesar 99% dan 5%. Sehingga dibutuhkan optimasi adam pada tahap pengujian berikutnya untuk menanggulangi data yang tidak cocok.

Masking layer adalah inti dari penelitian ini. Dimana, perubahan nilai *opacity* dari *masking layer* hitam menentukan FER sudah berjalan dengan baik. Percobaan sistem ini menggunakan data wajah *dataset* CK+ sebagai proses *training* ekspresi wajah dan didapatkan hasil sebagai berikut.



4.5 Pengujian *Face expression recognition*

Pengujian *face expression recognition* dalam sistem ini menggunakan dua model, yaitu ROC dan *k-fold cross validation*. Dimana pengujian pertama bertujuan untuk mengetahui perbandingan performa berdasarkan perbedaan *dataset* yang digunakan.

[illegible]

$$Accuracy = \frac{(TP+TN)}{(TN+TP+FP+FN)} = \frac{(19+11)}{(19+11+0+1)} = 0,96$$

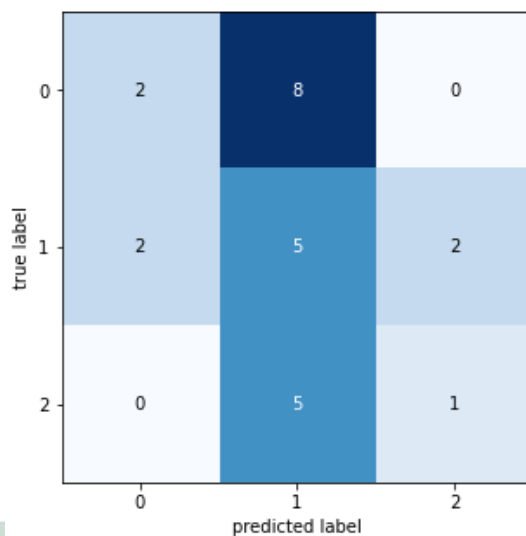
Hasil akurasi *dataset* CK+ pada data latih memperoleh akurasi sebesar 0.96 secara keseluruhan dan hasil tingkat *error* aplikasi memperoleh hasil yang sangat kecil sebesar 0,0001. Selain itu, berdasarkan program yang dibuat, sistem memperoleh nilai *f1 score*, *precision*, dan *recall* yang cukup tinggi dengan nilai rata-rata terbesar sebesar 74%. Ekspresi angry menduduki seluruh akurasi terbaik dalam pengujian ini. Hasil ini dipaparkan pada tabel di bawah ini.

Tabel 4. 1 Hasil Precision, Recall, F1 Score *Dataset* CK+

Ekspressi	Precision	Recall	F1-Score	Support
<i>Angry</i>	0.95	1.00	0.97	19
<i>Happy</i>	0.75	0.65	0.70	37
<i>Neutral</i>	0.48	0.58	0.52	19
Micro Avg	0.72	0.72	0.72	75
Macro Avg	0.73	0.74	0.73	75
Weighted Avg	0.73	0.72	0.72	75
Samples Avg	0.72	0.72	0.72	75

b. Dataset Custom

Pengujian kedua dilakukan pada *dataset custom*. Bertujuan untuk mengetahui perbandingan performa berdasarkan perbedaan *dataset* yang digunakan. Hasil akurasi dijelaskan pada Gambar 4.24 dan persamaan sebagai berikut.

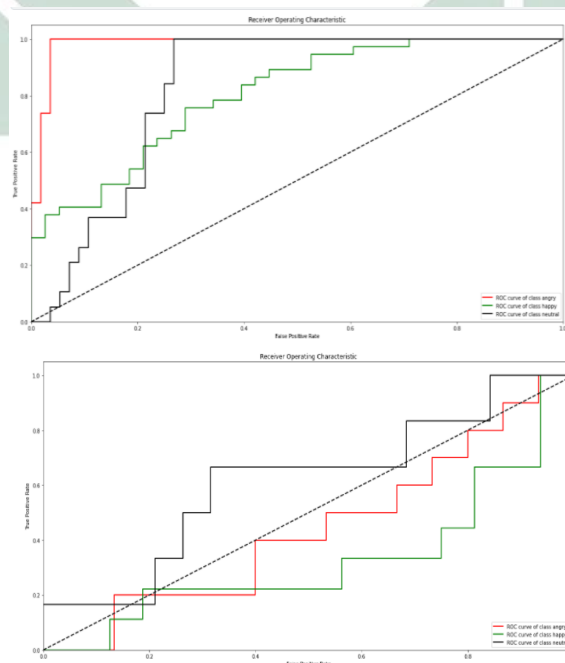


Tabel 4. 2 Hasil Precision, Recall, F1 Score *Dataset* Custom

Ekspressi	Precision	Recall	F1-Score	Support
<i>Angry</i>	0.50	0.20	0.29	10
<i>Happy</i>	0.25	0.44	0.32	9
<i>Neutral</i>	0.50	0.17	0.25	6
Micro Avg	0.32	0.28	0.30	25
Macro Avg	0.42	0.27	0.29	25
Weighted Avg	0.41	0.28	0.29	25
Samples Avg	0.28	0.28	0.28	25

4.5.1. Pengujian ROC

Pengujian selanjutnya adalah menggunakan teknik ROC. Dimana teknik ini berfokus pada keakuratan sistem dalam mengenali citra wajah selama aplikasi *game* berlangsung. Pengujian ini dilakukan pada *dataset* CK+ dan *dataset custom* dengan total masing - masing 498 dan 151 data. ROC membagi proses pengujian ke dalam 4 model yaitu *TPR (True Positive Rate)*, *FPR (False Positive Rate)*, *FNR (False Negative Rate)*, dan *TNR (True Negative Rate)*. Yang berarti tiap model mampu melakukan prediksi sama dengan false atau true. Pengujian ROC dipisah antara *dataset* CK+ dan data *custom* dengan hasil pengujian pada gambar dan tabel dibawah ini.



Gambar 4. 25 Hasil ROC *Dataset CK+* (atas) dan data *Custom* (bawah)

Gambar 4.25 menjelaskan pengujian ROC dari kedua data memperoleh hasil yang berbeda cukup jauh. Garis merah di lambangkan dengan *angry*, hijau dengan *happy*, dan hitam dengan *neutral*. *Dataset CK+* mengungguli *performance* dan keakuratan data dengan hasil prediksi sama dengan true sebesar 1.00 positive rate. Sedangkan *dataset custom* memperoleh nilai rata-rata rendah karena kurva pada *class neutral* dan *happy* menghasilkan nilai *false positive rate* sebesar -0.80%. Hasil pengujian kedua *dataset* dipaparkan pada Tabel 4.3 sebagai berikut.

Testing	Dataset CK+				Dataset custom			
	TPR	FPR	FNR	TNR	TPR	FPR	FNR	TNR
Angry	1.00	0.12	0	0	0.76	0.22	0	0
Happy	0.75	0.17	0	0	0.40	0.63	0	0
Neutral	0.84	0.21	0	0	0.20	0.75	0	0
Average	0.86	0.16	0	0	0.45	0.53	0	0

4.5.2. Pengujian *K-Fold Cross validation*

Tabel 4. 4 Hasil Pengujian K-Fold *Cross validation*

K-Fold	<i>Dataset CK+</i>			<i>Data Custom</i>		
	<i>Accuracy</i>	<i>Loss</i>	<i>RMSE</i>	<i>Accuracy</i>	<i>Loss</i>	<i>RMSE</i>
Fold- 1	92	1.2	-0.34	94	0.62	-0.19
Fold- 2	92	1.5	-0.32	88	0.42	-0.30
Fold- 3	89	1.4	-0.36	82	1.17	-0.21
Fold- 4	80	3.5	-0.25	88	0.20	-0.29
Fold- 5	86	2.2	-0.22	94	0.54	-0.12
Average	87.8	9.8	0.39	76	0.59	0.48

Pada Tabel 4.4 menjelaskan hasil pengujian dari sistem aplikasi FER dengan CNN berhasil memperoleh nilai 5 - *fold cross validation* memperoleh rata-rata nilai akurasi performa sebesar 80.4% selama 100 epoch karena terbatasnya ukuran pelatihan kedua data.

4.6 Evaluasi Hasil

Evaluasi hasil merupakan proses akhir dalam penelitian ini yaitu dengan melakukan uji coba di lapangan menggunakan sistem aplikasi *Game Flappy Bird™*. Pengujian dilakukan dengan mengoperasikan aplikasi dan perekaman data citra harus sudah terekam. Kemudian melakukan identifikasi dalam mengenali setiap citra wajah. Hasil pengujian tersebut dapat dilihat pada tabel sebagai berikut.

Tabel 4. 5 Hasil Evaluasi Pengujian Kamera *Webcam*

No	<i>Testing Data</i>		<i>Result</i>	<i>Masking Layer</i>	<i>Keterangan</i>	<i>Accuracy</i>
	<i>Ekspresi</i>	<i>Actual Condition</i>				
1	<i>Angry</i>	<i>User menghadap lurus kedepan</i>	<i>Angry</i>	100%	Terdeteksi	0.81
2	<i>Angry</i>	<i>User menghadap serong ke kanan</i>	<i>Angry</i>	100%	Terdeteksi	0.73
3	<i>Angry</i>	<i>User menghadap serong ke kiri</i>	<i>Angry</i>	100%	Terdeteksi	0.66
4	<i>Angry</i>	<i>User menghadap ke atas</i>	<i>Neutral</i>	50%	Salah Class	0.57
5	<i>Angry</i>	<i>User menghadap ke bawah</i>	<i>Neutral</i>	50%	Salah Class	0.54

Dari hasil beberapa pengujian di atas yang telah dilakukan pada *dataset* CK+ dan *dataset custom* dalam aplikasi FER. Didapatkan perbandingan nilai dari kedua data tersebut berdasarkan nilai *confusion matrix*, *ROC*, dan *k-fold cross validation*. Perbandingan kedua dataset dijelaskan ke dalam Tabel 4.6 di bawah ini.

Tabel 4. 6 Perbandingan Kinerja *Dataset* CK+ dan *Dataset custom*

Berdasarkan hasil tabel diatas, menjelaskan *dataset* CK+ bekerja lebih baik dari pada *dataset custom*. *Dataset* CK+ mampu mendeteksi ekspresi wajah dan memberi label ekspresi dengan tingkat kesalahan kecil *error* sebesar 0.39. *Dataset* Custom juga mampu melakukan klasifikasi ekspresi wajah, namun masih terdapat ketidak seimbangan yang disebabkan *overfitting* pada data tersebut. Pengujian dengan adam optimizer sudah dilakukan yang menampilkan hasil tidak jauh beda dengan sebelumnya. Sehingga dapat dikatakan bahwa sistem FER berjalan dengan baik oleh dataset CK+ untuk melakukan klasifikasi keakuratan ekspresi wajah dalam menentukan *predicted result* sesuai dengan actual result. Namun, FER tidak berjalan dengan sempurna ketika menggunakan dataset custom. Karena memperoleh nilai akurasi sangat rendah pada validation data yaitu sebesar 28%.

k-Fold memperoleh nilai yang tinggi. Dimana sistem dapat dengan mudah mendeteksi 3 ekspresi wajah.

5.2 Saran

Saran yang di lakukan penulis dalam penggunaan aplikasi dan menjelaskan pengembangan sistem yang akan datang anatra lain.

1. Melakukan pengembangan sistem menggunakan fitur-fitur ataupun metode baru dalam deep learning, sehingga tidak hanya FER saja.
2. Dataset.costum menunjukkan validasi akurasi yang rendah untuk dilakukan testing data. Sehingga dibutuhkan data yang lebih banyak dengan ekspresi yang lebih variative. Pengembangan aplikasi ini, dapat ditambahkan fitur secara luas baik dari segi score, input, menu, hingga kebutuhan hardwarenya.
3. Mengembangkan sistem deteksi ekspresi wajah dengan game karena aplikasi ini menggunakan bahasa pemrograman python yang cukup umum digunakan. Ekspresi wajah yang terdeteksi akan mengirim sinyal untuk merubah *masking layer* saja. Dan akurasi pada sistem FER dapat memperoleh nilai yang lebih baik jika dilakukan pengujian epoch lebih dari 100 kali.

Chen, Huazhou, An Chen, Lili Xu, Hai Xie, Hanli Qiao, Qinyong Lin, and Ken Cai. 2020. "A Deep Learning CNN Architecture Applied in Smart Near-Infrared Analysis of Water Pollution for Agricultural Irrigation Resources." *Agricultural Water Management* 240 (May): 106303. <https://doi.org/10.1016/j.agwat.2020.106303>.

Chen, Jiasi, and Xukan Ran. 2019. "Deep Learning With Edge Computing: A Review." *Proceedings of the IEEE*, 1–19. <https://doi.org/10.1109/JPROC.2019.2921977>.

Cross, S. S., R. F. Harrison, and R. L. Kennedy. 1995. *Introduction to Neural Networks. The Lancet*. Vol. 346. [https://doi.org/10.1016/S0140-6736\(95\)91746-2](https://doi.org/10.1016/S0140-6736(95)91746-2).

Diaz-Tula, Antonio, Carlos H. Morimoto, and Ronald D. Ranvaud. 2015. "A Mathematical Model of Saccadic Reaction Time as a Function of the Fixation Point Brightness Gain." *Attention, Perception, and Psychophysics* 77 (6): 2153–65. <https://doi.org/10.3758/s13414-015-0902-9>.

Fawcett, Tom. 2006. "An Introduction to ROC Analysis." *Pattern Recognition Letters* 27 (8): 861–74. <https://doi.org/10.1016/j.patrec.2005.10.010>.

Gupta, Shivam. 2018. "Facial Emotion Recognition in Real-Time and Static Images." *Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018*, no. Icisc: 553–60. <https://doi.org/10.1109/ICISC.2018.8398861>.

Hashemi, Seyed Raein, Sanjay P. Prabhu, Simon K. Warfield, and Ali Gholipour. 2018. "Exclusive Independent Probability Estimation Using Deep 3D Fully Convolutional DenseNets: Application to IsoIntense Infant Brain MRI Segmentation," 260–74. <http://arxiv.org/abs/1809.08168>.

Hassouneh, Aya, A. M. Mutawa, and M. Murugappan. 2020. "Development of a Real-Time Emotion Recognition System Using Facial Expressions and EEG Based on Machine Learning and Deep Neural Network Methods." *Informatics in Medicine Unlocked* 20: 100372. <https://doi.org/10.1016/j.imu.2020.100372>.

- Huang, Rui, Xiaohua Xie, Zhanxiang Feng, and Jianhuang Lai. 2017. "FACE RECOGNITION BY LANDMARK POOLING-BASED CNN WITH CONCENTRATE LOSS School of Electronics and Information Technology , Sun Yat-Sen University , China School of Data and Computer Science , Sun Yat-Sen University , China Guandong Key Laboratory of Informat," 1582–86.
- Hunt, John. 2019. *Multiprocessing*. https://doi.org/10.1007/978-3-030-25943-3_31.
- Hussein, Ealaf S., Uvais Qidwai, and Mohamed Al-Meer. 2020. "Emotional Stability Detection Using Convolutional Neural Networks." *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT 2020*, 136–40. <https://doi.org/10.1109/ICIoT48696.2020.9089440>.
- Hussein, Marwah Kamil, Alyaa J. Jalil, and Asaad Alhijaj. 2020. "Face Recognition Using the Basic Components Analysis Algorithm." *IOP Conference Series: Materials Science and Engineering* 928 (3). <https://doi.org/10.1088/1757-899X/928/3/032010>.
- Huttenlocher, Daniel. 2004. "Computer Vision." *Computer Science Handbook, Second Edition*, 43-1-43–23. <https://doi.org/10.22214/ijraset.2021.35926>.
- Iacopo Masi, Yue Wu, Tal Hassner, And Prem Natarajan. 2018. "Deep Face Recognition: A Survey." *Masi, Iacopo et Al. "Deep Face Recognition: A Survey." 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (2018);*, 471–78. <https://doi.org/10.1109/SIBGRAPI.2018.00067>.
- IBM. 2021. "IBM - Computer Vision." IBM. 2021. <https://www.ibm.com/topics/computer-vision>.
- Jaiswal, Akriti, A. Krishnama Raju, and Suman Deb. 2020. "Facial Emotion Detection Using Deep Learning." *2020 International Conference for Emerging Technology, INCET 2020*, 1–5. <https://doi.org/10.1109/INCET49848.2020.9154121>.
- Jaiswal, Shruti, and G. C. Nandi. 2020. "Robust Real-Time Emotion Detection System Using CNN Architecture." *Neural Computing and Applications* 32 (15): 11253–62. <https://doi.org/10.1007/s00521-019-04564-4>.

- Jeong, Jiwon. 2019. "The Most Intuitive and Easiest Guide for Convolutional Neural Network." 2019. <https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480>.
- Jie, Huang Jin, and Putra Wanda. 2020. "Runpool: A Dynamic Pooling Layer for Convolution Neural Network." *International Journal of Computational Intelligence Systems* 13 (1): 66–76. <https://doi.org/10.2991/ijcis.d.200120.002>.
- Joosten, Evi, Giel van Lankveld, and Pieter Spronck. 2010. "Colors and Emotions in Videogames." *11th International Conference on Intelligent Games and Simulation, GAME-ON 2010*, no. Figure 1: 61–65.
- Karray, Fakhreddine, Milad Alemzadeh, Jamil Abou, and Mo Nours. 2008. "Interaksi Manusia-Komputer : Tinjauan Tentang State of the Art" 1: 137–59.
- Khairani, Nurlaili, Juniardi Nur Fadila, and Fressy Nugroho. 2021. "Perancangan Game 2 Dimensi Petualangan Anak Menyelamatkan Orangtua Sebagai Media Edukatif Bagi Anak Dengan Metode Waterfall" 5 (1): 19–23.
- Kim, Phil. 2017. "MATLAB Deep Learning." *MATLAB Deep Learning*, no. November 2013: 121–47. <https://doi.org/10.1007/978-1-4842-2845-6>.
- Kim, Seung-Ryeol, Seung-Hyuck Lee, Dong-Hwan Jeon, Jeong-Sik Kim, and Seung-Woo Lee. 2017. "Optimum Display Luminance Dependence on Ambient Illuminance." *Optical Engineering* 56 (1): 017110. <https://doi.org/10.1117/1.oe.56.1.017110>.
- Kumar, Manoj Prabhakaran, and Manoj Kumar Rajagopal. 2019. "Detecting Facial Emotions Using Normalized Minimal Feature Vectors and Semi-Supervised Twin Support Vector Machines Classifier." *Applied Intelligence* 49 (12): 4150–74. <https://doi.org/10.1007/s10489-019-01500-w>.
- Kumar, Shreshth, and Rishab Kumar. 2016. "A Research Paper on Endless Fun," no. 11: 1–8.
- Lahariya, Abhinav, Varsha Singh, and Uma Shanker Tiwary. 2021. "Real-Time Emotion and Gender Classification Using Ensemble CNN." <http://arxiv.org/abs/2111.07746>.
- lantern, colin ware. 2021. "Color Mapping." 2021. http://lantern.ncsa.illinois.edu/CI_Intern/mvm/mapping.html.

- Lecun, Yann, Le'on Bottou, Yoshua Bengio, and Parick Haffner. 1998. "Gradient Based Learning Applied to Document Recognition." *Proc. OF THE IEEE*. <http://ieeexplore.ieee.org/document/726791/#full-text-section>.
- Li, Lixiang, Xiaohui Mu, Siying Li, and Haipeng Peng. 2020. "A Review of Face Recognition Technology." *IEEE Access* 8: 139110–20. <https://doi.org/10.1109/ACCESS.2020.3011028>.
- Li, Shan, and Weihong Deng. 2020. "Deep Facial Expression Recognition: A Survey." *IEEE Transactions on Affective Computing*, 1–25. <https://doi.org/10.1109/TAFFC.2020.2981446>.
- Liakos, Konstantinos G., Patrizia Busato, Dimitrios Moshou, Simon Pearson, and Dionysis Bochtis. 2018. "Machine Learning in Agriculture: A Review." *Sensors (Switzerland)* 18 (8): 1–29. <https://doi.org/10.3390/s18082674>.
- Lucey, Patrick, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. 2010. "The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-Specified Expression." *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, no. July: 94–101. <https://doi.org/10.1109/CVPRW.2010.5543262>.
- Lucey, Patrick, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. 2010. "The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-Specified Expression." *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, no. July: 94–101. <https://doi.org/10.1109/CVPRW.2010.5543262>.
- Masithoh, R. E., Raharjo, B., Sutiarso, L., Hardjoko, A. 2011. "Pengembangan Computer Vision System Sederhana" 31 (2): 116–23.
- Melinte, Daniel Octavian, and Luige Vladareanu. 2020. "Facial Expressions Recognition for Human–Robot Interaction Using Deep Convolutional Neural Networks with Rectified Adam Optimizer." *Sensors (Switzerland)* 20 (8). <https://doi.org/10.3390/s20082393>.
- Merdivan, Erinc, Anastasios Vafeiadis, Dimitrios Kalatzis, Sten Hanke, Joahannes Kroph, Konstantinos Votis, Dimitrios Giakoumis, et al. 2019. "Image-Based

- Text Classification Using 2d Convolutional Neural Networks.” *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI* 2019, 144–49. <https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00066>.
- Mishra, Sagar, Basanta Joshi, Rajendra Paudyal, Duryodhan Chaulagain, and Subarna Shakya. 2022. “Deep Residual Learning for Facial Emotion Recognition.” *Lecture Notes on Data Engineering and Communications Technologies* 68 (July): 301–13. https://doi.org/10.1007/978-981-16-1866-6_22.
- Moniaga, Jurike V., Andry Chowanda, Agus Prima, Oscar, and M. Dimas Tri Rizqi. 2018. “Facial Expression Recognition as Dynamic Game Balancing System.” *Procedia Computer Science* 135: 361–68. <https://doi.org/10.1016/j.procs.2018.08.185>.
- Naveenkumar, M, and V Ayyasamy. 2016. “OpenCV for Computer Vision Applications.” *Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15)*, no. March 2015: 52–56. https://www.researchgate.net/publication/301590571_OpenCV_for_Computer_Vision_Applications.
- Neha, R., and S. Nithin. 2018. “Comparative Analysis of Image Processing Algorithms for Face Recognition.” *Proceedings of the International Conference on Inventive Research in Computing Applications, ICIRCA 2018*, no. Icirca: 683–88. <https://doi.org/10.1109/ICIRCA.2018.8597309>.
- Ninaus, Manuel, Simon Greipl, Kristian Kiili, Antero Lindstedt, Stefan Huber, Elise Klein, Hans Otto Karnath, and Korbinian Moeller. 2019. “Increased Emotional Engagement in Game-Based Learning – A Machine Learning Approach on Facial Emotion Detection Data.” *Computers and Education* 142 (June): 103641. <https://doi.org/10.1016/j.compedu.2019.103641>.
- Ozdemir, Mehmet Akif, Berkay Elagoz, Aysegul Alaybeyoglu, Reza Sadighzadeh, and Aydin Akan. 2019. “Real Time Emotion Recognition from Facial

- Santos, Banu, and Ryan Putranda Kristianto. 2020. "Implementasi Penggunaan Opencv Pada Face Recognition Untuk Sistem Presensi Perkuliahan Mahasiswa." *Sistemasi* 9 (2): 352. <https://doi.org/10.32520/stmsi.v9i2.822>.
- Schuchhardt, Matthew, Susmit Jha, Raid Ayoub, Michael Kishinevsky, and Gokhan Memik. 2015. "Optimizing Mobile Display Brightness by Leveraging Human Visual Perception." *2015 International Conference on Compilers, Architecture and Synthesis for Embedded Systems, CASES 2015*, 11–20. <https://doi.org/10.1109/CASES.2015.7324538>.
- Shao, Jie, and Yongsheng Qian. 2019. "Three Convolutional Neural Network Models for Facial Expression Recognition in the Wild." *Neurocomputing* 355: 82–92. <https://doi.org/10.1016/j.neucom.2019.05.005>.
- Shu, Yufeng, Changwei Xiong, and Sili Fan. 2020. "Interactive Design of Intelligent Machine Vision Based on Human–Computer Interaction Mode." *Microprocessors and Microsystems* 75: 103059. <https://doi.org/10.1016/j.micpro.2020.103059>.
- Shyu, Mei-ling, Shu-ching Chen, and S S Iyengar. 2020. "A Survey on Deep Learning Techniques." *Strad Research* 7 (8). <https://doi.org/10.37896/sr7.8/037>.
- Singh, Arjun, Arun Pratap Srivastav, Pushpa Choudhary, and Sandeep Raj. 2021. "Facial Emotion Recognition Using Convolutional Neural Network." *Proceedings of 2021 2nd International Conference on Intelligent Engineering and Management, ICIEM 2021*, 486–90. <https://doi.org/10.1109/ICIEM51511.2021.9445346>.
- Singkoh, Robert Theophani, Arie S M Lumenta, and Virginia Tulenan. 2016. "Perancangan Game Fps (First Person Shooter) Police Personal Training." *Jurnal Teknik Elektro Dan Komputer* 5 (1): 28–34. <https://doi.org/10.35793/jtek.5.1.2016.11490>.
- Suyanto, Kurniawan Nur Ramadhani, Satria Mandala. 2019. *Deep Learning Modernisasi Machine Learning Untuk Big Data*. Bandung: Informatika Bandung.
- Svoren, Henrik, Vajira Thambawita, Pål Halvorsen, Petter Jakobsen, Enrique Garcia Ceja, Farzan Majeed Noori, Hugo Hammer, Mathias Lux, Michael

- Riegler, and Steven Hicks. 2020. "Toadstool: A Dataset for Training Emotional Intelligent Machines Playing Super Mario Bros," no. March. <https://doi.org/10.31219/osf.io/4v9mp>.
- Technology, Information, and Universitas Gadjah Mada. 2020. "Introduction of Deep Learning for Computer Vision : A Brief Update on How AI Can Be Used to Fight against Copyright (c) 2020 - Dr . Sunu Wibirama," no. c.
- Tensorflow. 2021. "Tensorflow Overview." 2021. <https://www.tensorflow.org/overview>.
- Tyagi, Vipin. 2018. "Understanding Digital Image Processing." *Understanding Digital Image Processing*, no. November. <https://doi.org/10.1201/9781315123905>.
- Victor Zhou. 2019. "A Simple Explanation of the Softmax Function." 2019. <https://victorzhou.com/blog/softmax/>.
- Vigneswari, D., N. Komal Kumar, V. Ganesh Raj, A. Gudan, and S. R. Vikash. 2019. "Machine Learning Tree Classifiers in Predicting Diabetes Mellitus." *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 84–87. <https://doi.org/10.1109/ICACCS.2019.8728388>.
- Wang, Mei, and Weihong Deng. 2021. "Deep Face Recognition: A Survey." *Neurocomputing* 429: 215–44. <https://doi.org/10.1016/j.neucom.2020.10.081>.
- Wang, Yingying, Yibin Li, Yong Song, and Xuewen Rong. 2019. "Facial Expression Recognition Based on Auxiliary Models." *Algorithms* 12 (11): 1–15. <https://doi.org/10.3390/a12110227>.
- Wang, Zhi-xiao, Hye-moon Lee, and Won-hyung Lee. 2018. "Adjusting the Difficulty of Running Game with Facial Expression Recognition Technology Using Convolutional Neural Network." *Journal of The Korean Society for Computer Game* 31 (2): 39–46. <https://doi.org/10.22819/kscg.2018.31.2.006>.
- Ware, Colin. 1988. "Color/Maps Color Sequences for Univariate Maps: Theory, Experiments, and Principles." *IEEE Computer Graphics and Applications* 8 (5): 41–49. <https://doi.org/10.1109/38.7760>.
- WorkWithColor.com. 2013. "Color Luminance." 2013. <http://www.workwithcolor.com/color-luminance-2233.htm>.

