

**KLASIFIKASI EKSPRESI WAJAH MENGGUNAKAN
METODE *PRINCIPAL COMPONENT ANALYSIS* (PCA)
DAN *CONVOLUTIONAL NEURAL NETWORK* (CNN)**



OLEH:

DWI LYDIA ZUHARAH ASTUTI

09042621721004

**PROGRAM STUDI MAGISTER TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2019**

**KLASIFIKASI EKSPRESI WAJAH MENGGUNAKAN
METODE *PRINCIPAL COMPONENT ANALYSIS* (PCA)
DAN *CONVOLUTIONAL NEURAL NETWORK* (CNN)**

TESIS

**Diajukan untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Magister**



OLEH:

DWI LYDIA ZUHARAH ASTUTI

09042621721004

**PROGRAM STUDI MAGISTER TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2019**

LEMBAR PENGESAHAN
KLASIFIKASI EKSPRESI WAJAH MENGGUNAKAN
METODE *PRINCIPAL COMPONENT ANALYSIS* (PCA)
DAN *CONVOLUTIONAL NEURAL NETWORK* (CNN)

TESIS

Diajukan Untuk Menengkapi Salah Satu Syarat
Memperoleh Gelar Magister

OLEH :
DWI LYDIA ZUHARAH ASTUTI
09042621721004

Pembimbing I,



Samsuryadi, M.Kom., Ph.D.
NIP 197102041997021003

Palembang, Agustus 2019

Pembimbing II,



Dian Palupi Rini, M.Kom., Ph.D.
NIP 197802232006042602

Mengetahui,
Kordinator Program Studi
Magister Teknik Informatika



4 Dr. Ir. Sukemi, M.T.
NIP 196612032006041001

HALAMAN PERSETUJUAN

Pada Hari Kamis tanggal 18 Juli 2019 telah dilaksanakan ujian sidang akhir Tesis II oleh Magister Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.

Nama : Dwi Lydia ZuharahAstuti
 NIM : 09042621721004
 Judul : Pengklasifikasi Ekspresi Wajah Menggunakan Metode Principal Component Analysis (PCA) dan Convolutional Neural Network (CNN)

1. Pembimbing I

Samsurvadi, M.Kom., Ph.D.
 NIP.197102041997021003



2. Pembimbing II

Dian Palupi Rini, M.Kom., Ph.D.
 NIP.197802232006042002



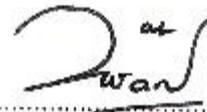
3. Penguji I

Dr. Ermatita, M.Kom.
 NIP. 196709132006042001

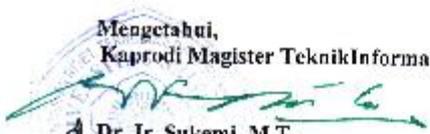


4. Penguji II

Dr. Iwan Pahendra, M.T.
 NIP. 197403222002121002



Mengetahui,
 Kaprodi Magister Teknik Informatika


Dr. Ir. Sukemi, M.T.
 NIP. 196612032006041001

LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Dwi Lyda Zuhara Asati
 NIM : 09045021721004
 Program Studi : Magister Teknik Informatika
 Judul Tesis : Klasifikasi Ekspresi Wajah Menggunakan Metode
Principal Component Analysis (PCA) dan *Convolutional
 Neural Network (CNN)*

Hasil Pengujian Software (Thesis) Tuntas: 14%

Menyatakan bahwa laporan tesis saya merupakan hasil karya sendiri dan bukan hasil pengotakan/bogor. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan tesis ini, maka saya bersedia menerima sanksi akademik dari Universitas Sebelas Maret dengan ketentuan yang berlaku.

Demikian pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapapun.



Probanding, 23 Agustus 2019



Dwi Lyda Zuhara Asati
 NIM 09045021721004

KATA PENGANTAR

Puji syukur Penulis ucapkan kehadirat Allah SWT, yang atas rahmat dan karunia-Nya Penulis dapat menyelesaikan Tesis ini. Shalawat dan salam tercurah kepada Rasulullah Muhammad SAW beserta keluarganya.

Selama melaksanakan Tesis ini, Penulis mendapat bantuan dan dukungan dari berbagai pihak. Untuk itu, Penulis ingin mengucapkan terima kasih kepada:

1. Bapak Samsuryadi, M.Kom., Ph.D. dan Ibu Dian Palupi Rini, M.Kom., Ph.D. selaku pembimbing, yang telah memberikan bimbingan, pengarahan dan semangat selama pengerjaan Tesis ini hingga selesai;
2. Bapak Prof. Drs. Saparudin, M.T., Ph.D., Bapak Deris Stiawan, M.T., Ph.D., Bapak Dr. Iwan Pahendra, M.T. dan Ibu Dr. Ermatita, M.Kom. selaku dosen penguji dari Tesis I, Seminar Hasil hingga Sidang Tesis II yang telah memberikan pengarahan dalam penyelesaian Tesis ini;
3. Bapak Dr. Reza Firsandaya Malik, M.T. yang telah memberikan bimbingan, arahan dan dukungan selama proses penyelesaian Tesis ini;
4. Ayah, Ibu, dan saudara-saudari tercinta, beserta seluruh keluarga yang memberi semangat, do'a beserta dukungannya;
5. Seluruh Staff dan Dosen Fasilkom yang telah membantu;
6. Teman-teman seperjuangan S2 Teknik Informatika yang selalu memberikan motivasi kepada Penulis;
7. Semua pihak yang membantu, yang tidak dapat Penulis sebutkan satu persatu.

Penulis menyadari bahwa Tesis ini bukanlah tanpa kelemahan, untuk itu kritik dan saran sangat diharapkan. Akhir kata, semoga Tesis ini dapat bermanfaat bagi para pembacanya.

Palembang, Agustus 2019

Penulis

DAFTAR ISI

	Halaman
Halaman Judul	i
Lembar Pengesahan	iii
Lembar Persetujuan	iv
Lembar Pernyataan	v
Kata Pengantar	vi
Abstrak	vii
Abstract	vii
i	
Daftar Isi	ix
Daftar Gambar	xi
Daftar Tabel	xii
i	
Daftar Lampiran	xi
v	
Daftar Singkatan	xv
BAB I PENDAHULUAN	
1.1 Latar Belakang	16
1.2 Rumusan Masalah	17
1.3 Tujuan Penelitian	17
1.4 Manfaat Penelitian	18
1.5 Batasan Masalah	18
1.6 Metodologi Penulisan	19
BAB II TINJAUAN PUSTAKA	
2.1 Penelitian Terdahulu	20
2.2 Pengenalan Ekspresi Wajah	21
2.3 Klasifikasi Ekspresi Wajah	26
2.3.1 Deteksi Wajah	26
2.3.2 Ekstraksi Ciri Citra	28
2.3.3 Klasifikasi Citra	29
2.4 Metode <i>Principal Component Analysis</i> (PCA)	30
2.5 Metode <i>Convolutional Neural Network</i> (CNN)	35
2.6 Tipe Layer CNN	40
2.7 Layer <i>Convolutional</i>	41
2.7.1 Kedalaman atau <i>Depth</i>	43
2.7.2 Langkah atau <i>Stride</i>	43
2.7.3 Zero-Padding	45
2.7.4 Layer Aktivasi / <i>Activation Layers</i>	47

2.7.5 <i>Pooling Layer / Layer Kolom</i>	47
2.7.6 <i>Layer Full-connected</i>	49
2.7.7 <i>Batch Normalization / Normalisasi Batch</i>	50
2.7.8 <i>Dropout</i>	50
2.8 Cara Kerja CNN	51
2.8.1 Model <i>Training CNN : Mini Xception</i>	53
BAB III METODOLOGI PENELITIAN	
3.1 Pendahuluan	56
3.2 Kerangka Kerja (<i>Framework</i>)	56
3.3 Pemodelan	57
3.3.1 Tahap Input	57
3.3.2 Tahap Ekstraksi Fitur	58
3.3.3 Tahap Klasifikasi	58
3.4 Tahap Penelitian	59
3.4.1 Menetapkan Kriteria Pengujian	59
3.4.2 Menetapkan Format Data Pengujian	60
3.4.3 Menentukan Alat yang Digunakan dalam Pelaksanaan Penelitian	61
3.4.4 Melakukan Pengujian Penelitian	61
3.4.5 Melakukan Analisa Hasil Pengujian dan Membuat Kesimpulan	62
BAB IV HASIL DAN ANALISIS	
4.1 <i>Dataset</i>	63
4.2 Deteksi Wajah dengan Metode <i>Haar Cascade (Viola-Jones)</i>	64
4.2.1 Skema Proses Deteksi Wajah	64
4.2.2 Praproses Menjadi Citra <i>Grayscale</i>	64
4.3 Ekstraksi Ciri Menggunakan PCA	66
4.4 Hasil	66
4.4.1 Antarmuka Aplikasi Pengklasifikasian Ekspresi Wajah Secara <i>Real Time</i> dengan metode PCA dan CNN	67
4.5 Pengujian Aplikasi	69
4.6 Eksperimen Pengklasifikasian Ekspresi Wajah Secara <i>Real Time</i>	70
4.6.1 Data Pengujian	70
4.6.2 Hasil Pengujian	75
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan	77
5.2 Saran	77
Daftar Pustaka	78

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Alur Proses Metode <i>Viola-Jones</i>	27
Gambar 2.2 Fitur persegi Panjang, A dan B. Dua fitur persegi Panjang, C. Tiga fitur persegi Panjang, D. Empat fitur persegi Panjang	27
Gambar 2.3 Struktur <i>Cascade Classifier</i>	28
Gambar 2.4 Diagram Alir Proses Pengklasifikasian Ekspresi Wajah	29
Gambar 2.5 Contoh konvolusi 2-D tanpa <i>flipping kernel</i>	38
Gambar 2.6 Lapisan Konvolusional di CNN	42
Gambar 2.7 Setelah mendapatkan peta aktivasi K, mereka disusun bersama untuk membentuk <i>volume input</i> ke lapisan berikutnya dalam jaringan	42
Gambar 2.8 Contoh <i>volume input</i> yang melalui aktivasi ReLU	47
Gambar 2.9 Volume Input	49
Gambar 2.10 Lapisan Jaringan Saraf	51
Gambar 2.11 Arsitektur MLP Sederhana	51
Gambar 2.12 Proses konvolusi pada CNN	52
Gambar 2.13 Arsitektur <i>Mini_Xception</i> yang diajukan untuk klasifikasi Ekspresi	54
Gambar 3.1 Tahapan Pengujian Penelitian	59
Gambar 4.1 <i>Sample</i> Data 6 Ekspresi Wajah Sesuai Urutan [1] Marah/ <i>Angry</i> [2] Jijik/ <i>Disgust</i> [3] Takut/ <i>Fear</i> [4] Senang/ <i>Happy</i> [5] Sedih/ <i>Sad</i> [6] Terkejut/ <i>Surprise</i>	64
Gambar 4.2 Alur Proses Deteksi Citra Wajah Menggunakan Metode <i>Viola-Jones</i>	64
Gambar 4.3 Alur Proses Pengenalan Citra Wajah menggunakan Metode <i>Eigenface</i>	65
Gambar 4.4 Hasil Deteksi Wajah pada <i>dataset</i> Citra SFEW 2.0	65
Gambar 4.5 <i>Sample</i> Data Input 6 Ekspresi Wajah Sesuai Urutan [1] Marah/ <i>Angry</i> [2] Jijik/ <i>Disgust</i> [3] Takut/ <i>Fear</i> [4] Senang/ <i>Happy</i> [5] Sedih/ <i>Sad</i> [6] Terkejut/ <i>Surprise</i>	66
Gambar 4.6 Alur Ekstraksi Ciri	66
Gambar 4.7 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara <i>Real Time</i> yang Menunjukkan Ekspresi Senang / <i>Happy</i>	67
Gambar 4.8 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara <i>Real Time</i> yang Menunjukkan Ekspresi Terkejut / <i>Surprised</i>	68
Gambar 4.9 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara <i>Real Time</i> yang Menunjukkan Ekspresi Marah / <i>Angry</i>	68
Gambar 4.10 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara <i>Real Time</i> yang Menunjukkan Ekspresi Sedih / <i>Sad</i>	68
Gambar 4.11 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara <i>Real Time</i> yang Menunjukkan Ekspresi Sedih / <i>Sad</i>	69
Gambar 4.12 Statistik <i>Confusion Matrix</i> dari Ekspresi Marah, Jijik dan Takut	70

Gambar 4.13 Statistik <i>Confusion Matrix</i> dari Ekspresi Senang dan Sedih	72
Gambar 4.14 Statistik <i>Confusion Matrix</i> dari Ekspresi Terkejut	73
Gambar 4.15 Model Akurasi dan Model <i>Loss</i> pada <i>Dataset</i> SFEW 2.0	74
Gambar 4.16 <i>Confusion Matrix</i> Ekspresi Wajah dengan <i>Dataset</i> SFEW 2.0	75
Gambar 4.17 <i>Confusion Matrix</i> Ekspresi Wajah dengan <i>Dataset</i> LFEW	75
Gambar 4.18 Grafik <i>Confusion Matrix</i> Pengklasifikasi Ekspresi Wajah	76
Gambar 4.19 Statistik Keseluruhan dari <i>Confusion Matrix</i> Pengklasifikasi Ekspresi Wajah dengan Metode PCA dan CNN	77

DAFTAR TABEL

	Halaman
Tabel 2.1 Gambar input 5x5 (kiri) yang akan kami gabungkan dengan kernel Laplace (kanan)	44
Tabel 2.2 Output Konvolusi dengan Langkah 1x1	45
Tabel 2.3 Output Penerapan Konvolusi 3x3 ke output 5x5	46
Tabel 3.1 Rancangan Tabel <i>Confusion Matrix</i> untuk Setiap Hasil Pengujian	60
Tabel 3.2 Rancangan Tabel Hasil Pengujian Klasifikasi Ekspresi Wajah	60
Tabel 3.3 Perangkat Keras yang digunakan	61
Tabel 3.4 Perangkat Lunak yang digunakan	61
Tabel 3.5 <i>Confusion Matrix</i>	62
Tabel 4.1 Pengujian Menjalankan Aplikasi Utama	69
Tabel 4.1 Hasil Pengujian Klasifikasi Ekspresi Wajah	75

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Real_time_video.py	L1-L3
Lampiran 2 Crawl_face.py	L4-L5
Lampiran 3 CNN.py	L6-L13
Lampiran 4 Load_and_process.py	L14-L15

DAFTAR SINGKATAN

FER	= Face Expression Recognition
PCA	= Principal Component Analysis
CNN	= Convolutional Neural Network
JAFFE	= Japanese Female Facial Expression
SFEW	= Static Facial Expressions in The Wild
DCNN	= Deep Convolutional Neural Network
STM	= Spatio-Temporal Manifold
UMM	= Universal Manifold Model
CK+ Dataset	= Cohn-Kanade Dataset
ELM	= Extreme Learning Machine

KLASIFIKASI EKSPRESI WAJAH MENGGUNAKAN METODE *PRINCIPAL COMPONENT ANALYSIS* (PCA) DAN *CONVOLUTIONAL NEURAL NETWORK* (CNN)

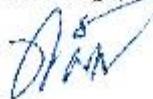
Dwi Lydia Zuharah Astuti

Abstrak

Klasifikasi ekspresi wajah secara nyata menjadi bagian penting dalam dunia komputer dan interaksi antar manusia-komputer, untuk aplikasi seperti hiburan digital, layanan pelanggan, pemantauan kesehatan, robot emosional, dan dalam dunia kedokteran untuk mengukur rasa sakit pada pasien tunawicara. Karena pengklasifikasian ekspresi wajah dipelajari melalui beberapa aspek yang berhubungan dengan wajah itu sendiri. Ketika ekspresi wajah berubah, maka lekukan pada wajah seperti alis, hidung, bibir dan mulut akan otomatis berubah. Mayoritas metode yang diusulkan dievaluasi dengan FER frontal terbatas, dan hasilnya menurun ketika menangani kasus FER *non-frontal* dan *multi-view*. Untuk menghadapi tantangan seperti itu, Penelitian ini melakukan perpaduan pengklasifikasian ekspresi wajah menggunakan metode *Principal Component Analysis* (PCA) dan *Convolutional Neural Network* (CNN). Hasil penelitian ini menunjukkan akurasi yang lebih akurat dari penelitian sebelumnya. Perpaduan antara metode PCA dan CNN pada STEW 2.0 dataset mendapatkan hasil 70,4% dibandingkan menggunakan metode CNN saja dengan hasil 60,8%.

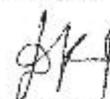
Keywords: Klasifikasi Ekspresi Wajah, Pengenalan Wajah, *Convolutional Neural Network*, *Principal Component Analysis*

Pembimbing I,



Saotunyadi Alkoni, Ph.D.
NIP. 97107941997921962

Pembimbing II,



Dina Palupi Rini, M.Kom., Ph.D.
NIP. 197802242006042002

FACE EXPRESSION CLASSIFICATION USING PRINCIPAL COMPONENT ANALYSIS (PCA) AND CONVOLUTIONAL NEURAL NETWORK (CNN)

Dwi Lydia Zaharah Astuti

Abstract

The classification of facial expressions is fast becoming an essential part of computer systems and human-computer interaction, for applications such as digital entertainment, customer service, driver monitoring, emotional robots, and the classification of facial expressions is studied through several aspects related to the face itself when facial expressions changed based on the point of view or perspective. The curve of the face such as eyebrows, nose, lips, and mouth will automatically change. The majority of the proposed methods with limited frontal FER, and their performance decreases when handling non-frontal and multi-view FER cases. To deal with such challenges, this research combines the classification of facial expressions using the Principal Component Analysis (PCA) and Convolutional Neural Network (CNN) methods. The results of this study indicate a more accurate compared to previous studies. The combination of PCA and CNN methods in the SFLW 2.0 dataset obtained 79.4% compared to using the CNN method alone with 69.9% results.

Keywords: Classification, Face Expression, Face Recognition, Convolutional Neural Network, Principal Component Analysis

Pembimbing I,



Samudiyasli, M.Kom., Ph.D.
NIP. 197162041997021000

Pembimbing II,



Dian Palupi Rini, M.Kom., Ph.D.
NIP. 197802212006042002

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengenalan ekspresi wajah atau *Face Expression Recognition* (FER) menjadi topik penelitian terkini diinteraksi manusia-komputer, untuk aplikasi seperti hiburan digital, layanan pelanggan, pemantauan pengemudi dan robot emosional. Telah ada studi yang luas, dan metode yang dikembangkan. Mayoritas metode yang diusulkan dievaluasi dengan FER frontal terbatas, dan kinerjanya menurun ketika menangani kasus FER non-frontal dan multi-view (Y. Liu et al., 2018). Untuk menghadapi tantangan seperti itu, penelitian ini mengusulkan metode *Principal Component Analysis* (PCA) dan metode *Convolutional Neural Network* (CNN) untuk klasifikasi ekspresi wajah di bawah multi-view, secara *real-time* dengan lingkungan yang tidak dibatasi secara efisien.

Salah satu metode klasifikasi ekspresi wajah yang memiliki akurasi cukup baik berdasarkan beberapa penelitian sebelumnya, *PCA* (*Principle Component Analysis*) atau sering disebut dengan *Eigenfaces* ini merupakan metode yang multifungsi digunakan, karena *Eigenfaces* memiliki banyak fungsi khususnya dalam pengenalan wajah seperti dapat melakukan prediksi, penghapusan redundansi, kompresi data, reduksi dimensi, sampai pada ekstraksi ciri (Putra, Dwidasmara, & Astawa, 2014). Sedangkan pada penelitian sebelumnya, telah dilakukan penelitian menggunakan metode *fisherface* dengan pendekatan jaringan syaraf tiruan *backpropagation*, dimana data uji yang digunakan adalah *dataset JAFFE* (Abidin, 2011; Garg & Choudhary, 2012).

Sistem pengenalan wajah dengan menggunakan *Convolutional Neural Network* telah diimplementasikan terhadap *Data Testing The Extended Yale Face Database B* (Abhirawan, Jondri, & Arifianto, 2017). Sedangkan pada penelitian CNN yang lain, dengan mengimplementasikan *Extended Local Binary Pattern* sebagai klasifikasi tekstur mampu mengatasi pengaruh intensitas cahaya pada gambar. Sehingga gambar yang terkena gangguan berupa intensitas cahaya dapat menghasilkan ekstraksi pola fitur yang hampir sama dengan gambar yang mendapatkan pencahayaan rendah. Konfigurasi inisialisasi parameter bobot

dengan menggunakan persebaran normal standar terdistribusi, dapat mempercepat konvergensi dan kestabilan dibandingkan dengan melakukan inisialisasi secara acak (Zufar, 2016).

Berdasarkan fakta-fakta di atas, penelitian ini memadukan metode *Principal Component Analysis* (PCA) sebagai pengekstraksi fitur dengan *Convolutional Neural Network* (CNN) untuk klasifikasi ekspresi wajah menggunakan *dataset* SFEW 2.0.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan, maka permasalahannya dapat dirumuskan sebagai berikut:

1. Bagaimana cara memadukan metode *Principal Component Analysis* (PCA) dan metode *Convolutional Neural Network* (CNN)? Apakah memungkinkan untuk digunakan pada penelitian klasifikasi ekspresi wajah?
2. Bagaimana cara mengimplementasikan metode *Principal Component Analysis* (PCA) dan metode *Convolutional Neural Network* (CNN)? Apakah penggunaan dua metode ini dapat mengoptimalkan proses klasifikasi ekspresi wajah serta mendapatkan akurasi yang baik?
3. Bagaimana kerangka kerja klasifikasi ekspresi wajah menggunakan 2 metode tersebut? Apakah setelah penggunaan dua metode ini, akurasi dari hasil proses klasifikasi ekspresi wajah meningkat?

1.3 Tujuan Penelitian

Menjawab dari masalah penelitian, adapun tujuan dari penelitian ini adalah memadukan antara metode *Principal Component Analysis* (PCA) dengan metode *Convolutional Neural Network* (CNN), sehingga dapat diimplementasikan untuk klasifikasi ekspresi wajah agar dapat berjalan dengan baik dan efisien meskipun intensitas pencahayaan kurang, *noise* yang tidak stabil maupun pola yang berbeda dari tekstur wajah. Serta dapat mengetahui tingkat akurasi setelah dilakukan proses memadukan metode tersebut.

1.4 Manfaat Penelitian

Manfaat penelitian ini adalah sebagai berikut:

1. Klasifikasi ekspresi wajah secara *real-time* yang bisa dimanfaatkan dalam sebuah sistem keamanan elektronik, hiburan digital, layanan pelanggan, robot emosional, pemantauan pengemudi maupun sistem absensi;
2. Klasifikasi ekspresi wajah sebagai aspek keamanan untuk pencarian identitas individu pada *database* kepolisian yang melibatkan data berupa citra digital wajah manusia;
3. Sistem klasifikasi ekspresi wajah sebagai pendeteksi kebohongan;
4. Dalam dunia kedokteran, untuk mengukur rasa sakit pada pasien tunawicara
5. Ekspresi wajah juga dimanfaatkan sebagai prediksi relasi interpersonal dengan menggunakan metode *Deep Convolutional Neural Network* (DCNN).

1.5 Batasan Masalah

Batasan penelitian agar sesuai dengan yang dimaksudkan dan lebih terarah adalah sebagai berikut:

1. Dataset gambar diambil melalui *Static Facial Expressions in the Wild* (SFEW) 2.0 <https://computervisiononline.com/dataset/1105138659>;
2. [Langkah-langkah memadukan 2 metode, antara metode Principal Component Analysis \(PCA\) dan metode Convolutional Neural Network \(CNN\) untuk mengklasifikasikan ekspresi wajah secara real-time;](#)
3. Bagaimana rancangan klasifikasi ekspresi wajah secara *real-time*, dengan memadukan dua metode pada penelitian ini, yaitu metode *Principal Component Analysis* (PCA) sebagai pengekstraksi ciri pada citra dengan metode *Convolutional Neural Network* (CNN) sebagai pengklasifikasi citra wajah.

1.6 Metodologi Penulisan

Untuk lebih memudahkan dalam menyusun tesis ini dan memperjelas isi dari setiap bab pada laporan ini, maka dibuatlah sistematika penulisan sebagai berikut:

1. BAB I Pendahuluan

Bab ini menjelaskan secara garis besar maksud dan tujuan dari penelitian Penulis, yang berisi tentang latar belakang, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan metodologi penulisan.

2. BAB II Tinjauan Pustaka

Bab ini berisi tentang studi literatur atau studi kepustakaan yang mencirikhaskan sebagai bagian yang mengemukakan teori dan pendapat para ahli terhadap masalah-masalah dalam penelitian Penulis.

3. BAB III Metodologi

Bab ini berisi penjelasan tentang proses atau cara ilmiah untuk mendapatkan data yang akan digunakan untuk keperluan penelitian, dan juga analisis teoretis mengenai suatu cara atau metode yang sistematis dan terorganisasi untuk menyelidiki masalah pada penelitian ini.

4. BAB IV Hasil dan Analisa

Dalam bab ini akan diuraikan hasil analisis data dan interpretasi hasil penelitian yang terdiri atas gambaran subjek penelitian. Data yang didapatkan disajikan pada bagian ini dalam bentuk narasi, tabel maupun gambar.

5. BAB V Kesimpulan dan Saran

Simpulan secara garis besar berupa isi, tujuan, manfaat dan hasil yang didapat pada penelitian ini disampaikan pada bagian ini. Serta berdasarkan hasil penelitian dan manfaat penelitian ini juga disampaikan saran yang perlu dilakukan untuk penelitian selanjutnya.

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian ekspresi wajah dengan menggunakan metode *Principal Component Analysis* (PCA) dan *JAFFE Dataset* yaitu *angry, happy, sad, disgust, fear* dan *surprise*. Serta pendekatan jarak *Euclidean* berdasarkan klasifikasi yang digunakan (Garg & Choudhary, 2012). Penelitian mengenai pengenalan wajah dalam sistem absensi secara *real-time* menggunakan metode *PCA (Principle Component Analysis)* atau sering disebut dengan *Eigenfaces* ini merupakan metode yang multifungsi digunakan, karena *Eigenfaces* memiliki banyak fungsi khususnya dalam pengenalan wajah seperti dapat melakukan prediksi, penghapusan redundansi, kompresi data, reduksi dimensi, sampai pada ekstraksi ciri (Putra et al., 2014).

Penelitian mengenai ekspresi wajah menggunakan 4 *databases* yaitu; CK+, MMI, Oulu-CAISA dan AFEW. Dengan metode *Spatio-Temporal Manifold (STM)* dan *Universal Manifold Model (UMM)* (M. Liu, Shan, Wang, & Chen, 2014). Penelitian mengenai pengenalan wajah secara *realtime* menggunakan *Convolutional Neural Networks* dengan menggunakan konstruksi model CNN sampai kedalaman 7 lapisan dengan input dari hasil ekstraksi *Extended Local Binary Pattern* dengan radius 1 dan *neighbor* 15 menunjukkan kinerja pengenalan wajah meraih rata-rata tingkat akurasi lebih dari 89% dalam ± 2 *frame* per detik (Zufar, 2016).

Penelitian ekspresi wajah berdasarkan *Cross-domain* menggunakan metode *Sparse Coding* (membandingkan antara algoritma TLbSC dengan algoritma PCA dan LPP) dengan SVM *classifier* sebagai pelatihan fitur dan *dataset* CK+, JAFFE serta NVIE (Yang, Zhang, & Huang, 2017). Penelitian mengenai pengenalan wajah menggunakan *Convolutional Neural Network* dengan menggunakan proses *dropout* diperoleh hasil terbaik dengan tingkat akurasi pengenalan setinggi 89,73% (Abhirawan et al., 2017). Penelitian mengenai pengenalan citra ekspresi wajah menggunakan algoritma *Principal Component Analysis* (PCA) dan *Extreme Learning Machine* (ELM) menggunakan dataset dari

JAFFE, berjumlah 210 citra ekspresi wajah, yang terdiri dari 10 orang masing-masing dengan 7 citra ekspresi berbeda, pengambilan foto dilakukan setiap 3 kali dimana perbandingan *Ratio data train* : dataset yang digunakan adalah 4 : 1 (Wiyono, 2018).

Penelitian mengenai pengenalan dan pendeteksian wajah untuk peningkatan kinerja kamera pengaman (Imanuel, 2018). Implementasi *real time* Raspberry Pi II yang ditempatkan pada robot untuk mengenali emosi dalam aplikasi yang interaktif, dengan menggunakan metode deteksi wajah *Viola Jones Haar Cascade*, *Active Shape Model* (ASM) sebagai ekstraksi fitur, serta *AdaBoost* sebagai klasifikasi *real time* (Palaniswamy & Tripathi, 2018).

Penelitian mengenai pengenalan ekspresi wajah menggunakan *Conditional Convolutional Neural Network Enhanced Random Forest*. Dengan menggunakan berbagai *dataset* CK+ publik, JAFFE, BU-3DEF dan *database* LFW (Y. Liu et al., 2018). Metode ini mencapai kinerja yang bagus untuk pengenalan ekspresi wajah dengan pose kepala yang bervariasi dengan membatasi jumlah contoh pelatihan, yang merupakan keunggulan pada *Deep Neural Networks*.

2.2 Pengenalan Ekspresi Wajah

Sistem pengenalan wajah hampir sama dengan sistem pengenalan *biometric* lainnya. Ide dibalik sistem pengenalan wajah (*Face Recognition*) adalah kenyataan bahwa setiap individu mempunyai wajah yang unik atau berbeda-beda. Sama halnya dengan sistem *biometric* lain seperti *Fingerprint* dan *Palm Recognition*. Wajah setiap individu mempunyai beberapa struktur dan fitur yang unik bagi masing-masing individu tersebut. Oleh karena itu sistem pengenalan wajah otomatis juga akan berdasarkan simetris dari wajah masing-masing individu. Tantangan pada sistem pengenalan wajah adalah : (1) Titik skala dan pergeseran pada wajah (2) Perbedaan pada paras wajah, seperti segi wajah, pose wajah, model rambut, *makeup*, kacamata, kumis, jenggot, dan lain-lain (3) Pencahayaan (4) Usia (Çarıkçı & Özen, 2012)

Tahapan pengenalan wajah secara umum terdiri dari dua sistem yaitu mendeteksi wajah yang merupakan tahap awal (*pre-processing*) untuk mengekstraksi ciri dari wajah dan sistem pengenalan wajah (*face recognition*)

(Alexander, 2013). Hampir semua area wajah adalah hal yang penting untuk dideteksi terutama bagian wajah dan bibir, yang berfungsi sebagai area penelitian seperti pendeteksi emosi dan identifikasi wajah (Putra et al., 2014). Metode pendeteksi fitur wajah ini berfungsi sebagai input untuk mendeteksi ekspresi wajah.

Ekspresi atau mimik wajah merupakan cara lain untuk komunikasi secara tidak langsung atau komunikasi nonverbal. Ekspresi wajah terdiri dari beberapa gerakan atau letak otot pada wajah yang dapat mengungkapkan isi hati kepada seseorang yang sedang memperhatikan kita. Sebagai contoh, mengernyitkan alis mata menunjukkan kemarahan atau ketidaksukaan, mengangkat alis mata menunjukkan ekspresi heran dan terkejut. Semua emosi dan berbagai macam isi hati manusia tergambar pada ekspresi wajah yang berbeda-beda (Abidin, 2011).

Istilah biometrik menggambarkan karakteristik terukur yang digunakan dalam pengukuran fitur organisme biologis. Namun, istilah tersebut lebih umum digunakan untuk menggambarkan variasi dalam karakteristik biologis manusia, yang digunakan untuk membedakan identitas antar individu. Beberapa pengukuran seperti itu sekarang menemukan penggunaan dalam keamanan otomatis dan sistem pengawasan, yang menggunakan biometrik untuk memverifikasi identitas seseorang seperti pengenalan wajah (David & Hons, 2005). Definisi pengenalan, dalam konteks sistem biometrik, sebagai kemampuan untuk melakukan verifikasi dan identifikasi. Verifikasi adalah proses membandingkan satu biometrik pola dengan pola biometrik lain, menghasilkan penolakan atau penerimaan keputusan. Sedangkan identifikasi adalah proses membandingkan satu pola biometrik dengan satu set atau lebih pola biometrik untuk menentukan kecocokan yang paling mungkin. Seiring waktu, kebutuhan akan kata sandi, kartu gesek, dan nomor pin perlahan-lahan menjadi digantikan oleh biometrik pengidentifikasi unik. Meskipun penerimaan publik dan pemahaman umum tentang kemampuan teknologi baru ini menghalangi peralihan dari sistem sebelumnya, masih ada insentif besar untuk menggunakan biometrik, yaitu sebagai berikut:

1. **Meningkatkan keamanan.** Kartu gesek dan nomor PIN dapat dengan mudah diperoleh oleh orang yang tidak bertanggung jawab, sedangkan memperoleh

subjek biometrik membutuhkan pengetahuan dan peralatan spesialis, dan dalam banyak kasus tidak mungkin tanpa sepengetahuan subjek.

2. **Mencegah penipuan.** Menjadi sangat sulit bagi seseorang untuk rela menyerahkan data biometriknya sebagai identitas (untuk “*buddy punching*” dalam sistem waktu dan kehadiran). Sebagai tambahan, karena itu perlu untuk mengekspos data biometrik sendiri (mis. wajah Anda sendiri), penipu potensial enggan untuk mencoba verifikasi salah.
3. **Menghemat biaya.** Dengan mengganti kartu gesek plastik, semua biaya terkait dengan memproduksi, mendistribusikan, dan mengganti kartu yang hilang sepenuhnya dihilangkan.

Selain kelebihan yang disebutkan diatas, sekali sistem identifikasi biometrik ada ditempat, keuntungan lain mulai muncul. Misalnya, ada kasus yang diketahui perusahaan besar menemukan beberapa karyawan mereka sebenarnya orang yang sama, telah berhasil memperoleh banyak identitas pada sistem penggajian perusahaan: sesuatu yang mudah diidentifikasi ketika beberapa karyawan tampak memiliki biometrik wajah yang sama. Terlebih lagi, tanpa sistem biometrik pada tempatnya, apa pun disengaja menyesatkan bisa sulit untuk dibuktikan, menempatkan insiden itu ke kesalahan yang biasa, tetapi kemampuan untuk melihat wajah yang sama untuk masuk sebagai banyak orang adalah bukti yang sangat meyakinkan.

Insentif ini telah menyebabkan beberapa opsi biometrik muncul selama beberapa tahun terakhir. Yang paling umum adalah pengenalan sidik jari, wajah dan *iris* tetapi contoh lainnya termasuk retina, suara, tekstur kulit, bentuk telinga, gaya berjalan (berjalan kaki), geometri tangan, pola pembuluh darah, tanda tangan termal dan tanda tangan tulisan tangan. Masing-masing memiliki kelebihan tersendiri dan mungkin sangat cocok untuk aplikasi tertentu. Misalnya *scanner* sidik jari berukuran kecil, ringan dan relatif murah, memungkinkan integrasi ke dalam jangkauan yang luas didalam perangkat seluler. Pola *iris* sangat kompleks dan beragam sehingga kecocokan salah tidak mungkin terjadi bahkan di antara jutaan subjek (meskipun ada laporan yang tinggi pada tingkat kegagalan

pendaftaran), sedangkan tanda tangan termal yang kurang akurat dapat diambil dari kejauhan: ideal untuk operasi rahasia.

Pengenalan wajah, meskipun tidak selalu cocok untuk semua aplikasi, memang memiliki beberapa keunggulan utama dibandingkan biometrik lain yang disebutkan diatas, yang kita sekarang diskusikan secara detail:

1. ***Tidak mengganggu.*** Sedangkan sebagian besar biometrik memerlukan beberapa tingkat interaksi pengguna untuk memperoleh data biometrik, seperti melihat ke pemindai mata atau menempatkan jari pada pembaca sidik jari, pengenalan wajah yang akurat dapat dilakukan hanya dengan melirik ke kamera dari kejauhan. Akuisisi biometrik non-kontak ini sangat praktis. Kemampuan ini dapat diambil selangkah lebih maju, menggunakan penempatan kamera strategis untuk melakukan pengakuan bahkan tanpa sepengetahuan subjek. Contoh pada kamera CCTV yang memonitor area untuk penjahat yang dikenal atau melacak tersangka kejahatan.
2. ***Penerimaan publik.*** Sistem pengenalan wajah pada umumnya menerima tingkat penerimaan publik yang lebih tinggi daripada kebanyakan biometrik lainnya. Ini dikarenakan sifat pengenalan wajah yang tidak mengganggu seperti dijelaskan di atas, tetapi mungkin juga menjadi hasil dari pemahaman dan empati yang lebih besar tentang bagaimana teknologi itu mampu mengenali wajah. Faktor lain adalah hubungan yang dimiliki biometrik lain dengan kejahatan (misalnya sidik jari). Apa pun alasannya, orang sudah terbiasa dengan wajah mereka.
3. ***Database yang ada.*** Satu kunci utama untuk organisasi besar yang mempertimbangkan implementasi sistem biometrik adalah jumlah waktu yang diperlukan dalam pengumpulan *database* biometrik. Pertimbangkan polisi memaksa menggunakan sistem pengenalan *iris*. Ini akan memerlukan beberapa tahun sebelum *database* berukuran besar cukup berguna untuk mengidentifikasi tersangka. Dimana database besar dengan gambar wajah berkualitas tinggi sudah masuk ke sistem, sehingga manfaat memasang sistem pengenalan wajah diperoleh segera setelah instalasi.
4. ***Analogi dengan persepsi manusia.*** Mungkin keuntungan terbesar, (yang juga paling sering diabaikan) adalah data biometrik yang diperlukan untuk

pengenalan wajah dikenali oleh manusia. Ini memungkinkan tingkat cadangan tambahan, seharusnya sistem gagal. Manusia meninjau sumber biometrik yang sama (gambar melalui *data set* dan gambar *live record*) selalu dapat secara manual memeriksa hasil identifikasi atau verifikasi. Sedangkan keputusan apa pun yang dibuat oleh sistem pengenalan biometrik lainnya, seperti *iris* atau sidik jari, akan membutuhkan ahli untuk memberikan konfirmasi yang andal. Sebentar produk dualitas ini dengan metode pengenalan manusia adalah data biometrik dapat didistribusikan ke organisasi lain (dari departemen kepolisian ke badan otoritas misalnya) dan masih berguna bahkan jika organisasi lain tidak memiliki sistem pengenalan wajah dalam operasi.

Sistem pengenalan wajah biometrik yang lengkap mencakup tiga prosedur utama. Itu langkah pendeteksian wajah awal (yang mungkin mencakup beberapa fitur) sering dilakukan jika tidak ada intervensi manual (manusia) yang akan digunakan. Ini melibatkan ekstraksi gambar wajah dari pemandangan yang lebih besar. Banyak metode telah diterapkan untuk masalah ini: teknik berbasis *template*, deteksi gerakan, segmentasi warna kulit, analisis komponen utama, dan klasifikasi oleh jaringan saraf. Semuanya merupakan tugas sulit untuk mengklasifikasikan gambar "non-wajah" dari area tersebut dengan pemandangan kompleks yang mengandung wajah. Prosedur ini sangat membantu jika kondisi di mana akuisisi gambar dilakukan dapat dikontrol. Karena itu tidak mengherankan bahwa banyak algoritma yang tersedia saat ini hanya berlaku untuk spesifik situasi. Asumsi dibuat mengenai orientasi dan ukuran wajah di gambar, kondisi pencahayaan, latar belakang dan kerja sama subjek.

Prosedur selanjutnya adalah pencarian dan pencocokan, sering disebut identifikasi. Ini tahap mengambil gambar yang diekstraksi dari adegan selama tahap deteksi wajah, dan membandingkannya dengan database orang yang dikenal (terdaftar sebelumnya), mencari gambar pencocokan terdekat, sehingga mengidentifikasi orang yang paling cocok. Sebuah poin penting mengenai proses ini adalah tidak menghasilkan 'ya' atau keputusan 'tidak', apakah ada dua gambar

dari orang yang sama atau tidak. Sebaliknya proses hanya menunjukkan gambar mana yang cocok.

Prosedur terakhir adalah verifikasi. Ini menggambarkan proses dimana dua wajah gambar dibandingkan, menghasilkan keputusan 'ya' atau 'tidak', apakah gambar tersebut berasal dari orang yang sama. Proses ini membutuhkan gambar permintaan (biasanya gambar yang diambil langsung) dan satu gambar galeri yang dipilih sebelumnya (juga disebut sebagai gambar target). Pemilihan ini dapat terjadi dalam beberapa cara: kartu gesek atau nomor pin yang menunjukkan gambar galeri yang sesuai; prosedur identifikasi otomatis seperti dijelaskan di atas, memilih kecocokan yang paling mungkin dari kumpulan gambar; gambar yang dipilih secara manual ditawarkan sebagai pertandingan potensial. Kedua gambar tersebut kemudian dibandingkan menghasilkan “klasifikasi orang yang sama” atau “klasifikasi orang yang berbeda”. Keputusan ini sering dibuat oleh aplikasi dari ambang batas untuk skor kesamaan (atau ketidaksamaan), seperti yang dihasilkan dalam proses identifikasi. Dengan menyesuaikan nilai ambang ini, seseorang dapat mengubah tingkat akurasi antara jumlah penerimaan palsu dan penolakan palsu.

2.3 Klasifikasi Ekspresi Wajah

Ekspresi wajah menyediakan informasi mengenai emosi dan memiliki peran yang penting dalam interaksi manusia dan komunikasi nonverbal. Kemampuan untuk mengenali dan memahami ekspresi wajah secara otomatis akan memfasilitasi komunikasi. Proses dalam klasifikasi ekspresi wajah manusia terdiri dari tiga tahapan yaitu deteksi wajah, ekstraksi ciri dan klasifikasi ekspresi wajah.

2.3.1 Deteksi Wajah

Proses deteksi wajah pada penelitian ini menggunakan metode *Viola-Jones*. Metode *Viola-Jones* merupakan metode deteksi wajah yang memberikan hasil deteksi wajah dengan tingkat akurasi yang tinggi. Penelitian yang menggunakan metode *Viola-Jones* dalam mendeteksi wajah dalam pengenalan

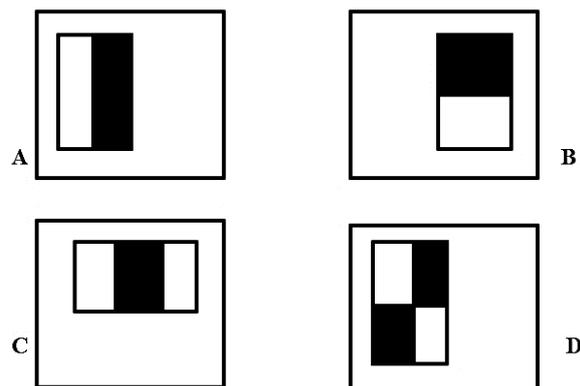
ekspresi wajah (Viola & Jones, 2004). Berikut ini alur proses metode *Viola-Jones* dalam mendeteksi wajah:



Gambar 2.1 Alur Proses Metode *Viola-Jones* (Viola & Jones, 2004)

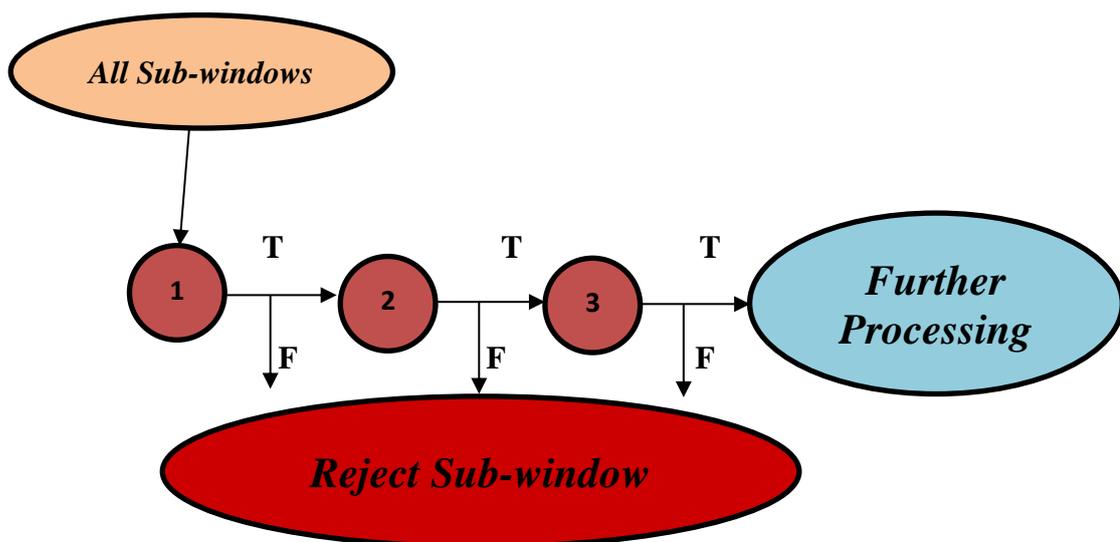
Tahapan-tahapan dalam deteksi wajah dengan metode *Viola-Jones* adalah sebagai berikut:

1. Input berupa citra yang memiliki objek wajah didalamnya dan posisi wajah frontal;
2. *Haar-Features* adalah sebuah fitur dari citra digital yang digunakan untuk mendeteksi sebuah objek. Proses pengambilan ciri menggunakan *Haar-Features* dengan tiga jenis fitur. Nilai fitur pertama terdiri dari dua persegi Panjang yang merupakan nilai selisih antara jumlah piksel dalam dua wilayah persegi Panjang. Nilai fitur kedua terdiri dari tiga persegi Panjang yang menghitung jumlah dalam dua persegi Panjang diluar dikurangi dengan jumlah persegi Panjang pusat. Nilai fitur ketiga terdiri dari empat persegi Panjang yang menghitung perbedaan antara pasangan diagonal persegi Panjang. Berikut ini ilustrasinya:



Gambar 2.2 Fitur persegi Panjang, A dan B. Dua fitur persegi Panjang, C. Tiga fitur persegi Panjang, D. Empat fitur persegi panjang (Viola & Jones, 2004)

3. *Integral Image* digunakan untuk menentukan ada atau tidaknya dari ratusan fitur *Haar* pada sebuah gambar. Fitur persegi Panjang dapat dihitung sangat cepat menggunakan representasi *integral image*;
4. Algoritma *Adaboost* digunakan untuk memilih fitur-fitur penting dan digunakan untuk melatih klasifikasi. Hal ini dilakukan dengan mengevaluasi setiap fitur terhadap data latih dengan menggunakan nilai dari fitur tersebut, fitur yang memiliki batasan terbesar antara objek dan non-objek dianggap sebagai fitur terbaik;
5. *Cascade Classifier* adalah sebuah metode untuk mengkombinasikan *classifier* yang kompleks dalam sebuah struktur bertingkat yang dapat meningkatkan kecepatan pendeteksian objek dengan memfokuskan pada daerah citra yang berpeluang saja. Pada tahap ini diketahui wajah dan bukan wajah pada citra. Ilustrasinya sebagai berikut;



Gambar 2.3 Struktur *Cascade Classifier* (Viola & Jones, 2004)

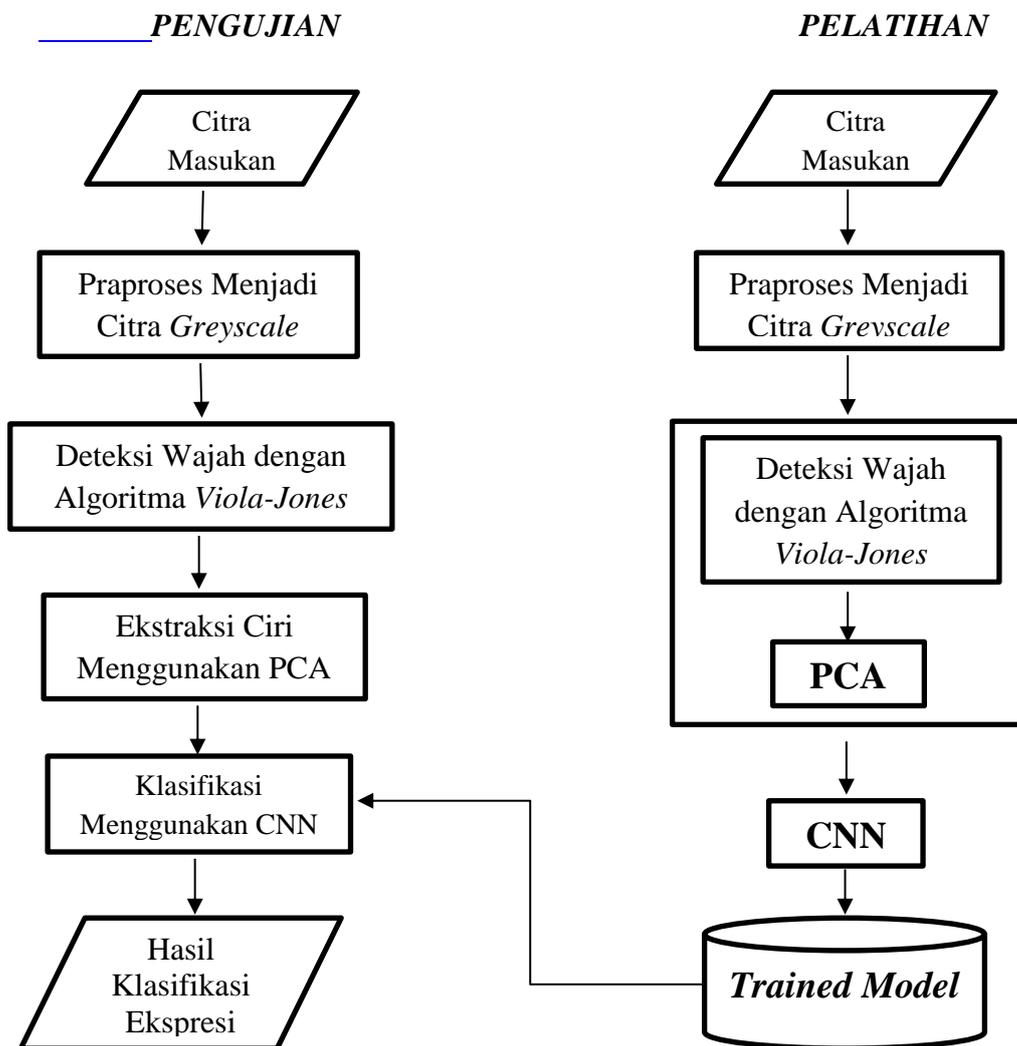
2.3.2 Ekstraksi Ciri Citra

Setelah citra wajah dilakukan praproses citra kemudian citra wajah diekstraksi untuk mendapatkan ciri penting pada citra tersebut. Ekstraksi ciri adalah proses mengambil ciri-ciri yang terdapat pada objek didalam citra. Ekstraksi ciri dilakukan untuk memunculkan pola yang akan digunakan dalam proses pengenalan wajah. Pengekstraksi fitur pada penelitian ini menggunakan

metode *Principal Component Analysis* dan pendekatan baru dalam proses ekstraksi ciri yaitu *feature learning* menggunakan *Convolutional Neural Network*.

2.3.3 Klasifikasi Citra

Proses klasifikasi adalah proses mengelompokkan objek kedalam kelas yang sesuai. Proses klasifikasi merupakan satu kesatuan didalam metode *Convolutional Neural Network*, tepatnya pada *layer* terakhir pada *Convolutional Neural Network* yaitu *fully connected layer*.



Gambar 2.4 Diagram Alir Proses Pengklasifikasian Ekspresi Wajah
(Ian, Yoshua, & Courville, 2014)

2.4 Metode *Principal Component Analysis* (PCA)

PCA adalah cara pengidentifikasian pola pada data, dan selanjutnya data diekstraksi berdasarkan kesamaan dan perbedaannya. Semenjak sulitnya menemukan pola pada data yang mempunyai dimensi yang besar, dimana gambaran grafik yang juga besar tidak mencukupi, PCA adalah metode yang ampuh dalam menganalisa data itu (Zhou, Wang, Zhang, & Wei, 2013). Kelebihan lainnya dari PCA adalah kita dapat menemukan beberapa pola di dalam data, dan kita dapat mengkompres data tersebut. Sebagai contoh, dapat menurunkan jumlah dimensi tanpa menghilangkan informasinya.

Konsep *Principal Component Analysis* (PCA) adalah mengelompokkan variabel-variabel yang berkorelasi linear menjadi 1 komponen utama, sehingga dari p variabel random $(x_1, x_2, x_3, \dots, x_p)$ akan didapat k komponen utama ($k < p$) yang mewakili variabilitas variabel yang ada. Tujuan dilakukannya PCA adalah untuk mereduksi struktur variabel menjadi variabel baru dengan dimensi yang lebih kecil. Variabel baru tersebut mampu menerangkan sebagian besar total varian data dan saling bebas satu sama lain. Selanjutnya variabel baru ini dinamakan *principal component* (PC). Reduksi dimensi data pada PCA dengan cara mentransformasi variabel-variabel asli yang berkorelasi menjadi satu set variabel baru yang tidak berkorelasi, dengan tetap mempertahankan sebesar mungkin varian yang dapat dijelaskan (Johnson & Wichern, 2007).

PC dapat dibentuk dari matriks kovarians maupun matriks korelasi. PC yang dibentuk dari matriks korelasi dilakukan jika variabel-variabel yang diamati mempunyai satuan pengukuran yang berbeda, maka variabel tersebut perlu distandarisasikan terlebih dahulu. Akibat adanya standarisasi data, maka matriks varians-kovarians dari data yang distandarisasi akan sama dengan matriks korelasi data sebelum distandarisasi dan besarnya total varians PC akan sama dengan banyaknya variabel asal.

Secara aljabar linier, komponen utama merupakan kombinasi linier dari p variabel acak $x_1, x_2, x_3, \dots, x_p$. Secara geometris, kombinasi linier ini merupakan sistem koordinat baru yang didapat dari rotasi sistem semula dengan $x_1, x_2, x_3, \dots, x_p$ sebagai sumbu koordinat. Sumbu baru tersebut merupakan arah dengan

variabilitas maksimum dan memberikan kovariansi yang lebih sederhana. Syarat untuk membentuk PC yang merupakan kombinasi linier dari variabel x agar mempunyai keragaman yang besar adalah dengan memilih *eigenvector* $\mathbf{e}_i = (e_{i1}, e_{i2}, \dots, e_{ip})^T$ sedemikian hingga $\text{Var}(e_{iT}x)$ maksimum dan $e_{iT}e_{iT} = 1$ dan $\text{cov}(e_{iT}x, e_{kT}x) = 0$ untuk $k < i$.

PC tergantung kepada matriks varians-kovarians Σ dan matriks korelasi ρ dari x_1, x_2, \dots, x_p . Melalui matriks varians-kovarians diturunkan *eigenvalue* $\tau_1 \geq \tau_2 \geq \tau_3 \geq \dots \geq \tau_p$ dan *eigenvector* $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_p$. Vektor random $\mathbf{x} = [x_1, x_2, \dots, x_p]$ mempunyai matriks varians-kovarians Σ dengan *eigenvalue* $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_p \geq 0$ maka kombinasi linier utama adalah:

$$\begin{aligned} PC_1 &= \mathbf{e}_1^T \mathbf{x} = e_{11}x_1 + e_{21}x_2 + \dots + e_{p1}x_p \\ PC_2 &= \mathbf{e}_2^T \mathbf{x} = e_{12}x_1 + e_{22}x_2 + \dots + e_{p2}x_p \\ &\quad \vdots \\ PC_p &= \mathbf{e}_p^T \mathbf{x} = e_{1p}x_1 + e_{2p}x_2 + \dots + e_{pp}x_p \end{aligned} \quad (2.1)$$

dengan:

PC_1 : PC pertama, yang mempunyai varians terbesar pertama

PC_2 : PC kedua, yang mempunyai varians terbesar kedua

PC_p : PC ke- p , yang mempunyai varians terbesar ke- p

x_1 : Variabel asal pertama

x_2 : Variabel asal kedua

x_p : Variabel asal ke- p

e_p : *Eigenvector* variabel ke- p

Model *Principal Component* ke- i secara umum ditulis dengan:

$$PC_i = e_{iT}x, \text{ dimana } i = 1, 2, \dots, p \quad (2.2)$$

Terdapat beberapa hal yang dapat dipakai sebagai acuan dalam menentukan banyaknya PC, antara lain:

1. Melihat *screen plot*. *Screen plot* menggambarkan besarnya *eigenvalue* λ_i . Dalam menentukan jumlah PC yang sesuai, maka bisa dilihat pada garis yang terbentuk, jika garis yang terbentuk mengalami *range* yang cukup besar maka PC sejumlah garis tersebut.

2. Apabila PC diperoleh dari matriks korelasi, maka banyaknya PC dipilih sesuai dengan banyaknya *eigenvalue* yang lebih besar dari satu.
3. Sebaiknya jumlah PC yang dipilih adalah yang mampu memberikan kumulatif persen varians 80% - 90%.

Pertimbangkan sebuah data set dengan beberapa parameter berikut : data set ditunjukkan oleh $A = (A_1, A_2, \dots, A_N)$, dimana data set mempunyai nilai $N, j = 1, 2, \dots, N$, yang memiliki beberapa kelas, $i = 1, 2, \dots, c$, dan jumlah pixel di dalam gambar bernilai n . Matriks sebaran antar kelas, matriks sebaran dalam kelas, dan total matriks sebaran didefinisikan sebagai berikut :

$$S_b = \sum_{i=1}^c P(\omega_i)(\mu_i - \mu_0)(\mu_i - \mu_0)^T \quad (2.3)$$

Dimana S_b matriks sebaran antar kelas, $P(\omega_i)$ adalah probabilitas apriori dari ω_i , pada umumnya, $P(\omega_i) = (1/c)$, μ_i adalah vektor rata-rata dari kelas ω_i , $\mu_0 = (1/N) \sum_{j=1}^N A_j$ adalah vektor rata-rata dari semua sampel.

$$S_w = \sum_{i=1}^c P(\omega_i) S_i \quad (2.4)$$

Dimana S_w matriks sebaran dalam kelas, $S_i = E \{(A - \mu_i)(A - \mu_i)^T | A \in \omega_i\}$ adalah kovarian matrik dari ω_i .

$$S_t = S_b + S_w = \frac{1}{N} \sum_{j=1}^N (A_j - \mu_0)(A_j - \mu_0)^T \quad (2.5)$$

Matriks sebaran antar kelas menunjukkan kelas sebaran rata-rata μ_i mendekati rata-rata semua μ_0 , dan matriks sebaran dalam kelas adalah sebaran dari sampel yang mendekati rata-rata kelas masing-masing μ_i . Adapun langkah-langkah metode PCA adalah sebagai berikut:

Mencari nilai *Eigenface* (dengan menggunakan metode PCA) yaitu fitur yang signifikan yang merupakan komponen prinsip dari kumpulan wajah dalam *database*. *Eigenface* didapat dari *eigenvector* matrik kovarian dari himpunan citra dalam *database*. *Eigenvector* ini merupakan fitur yang menggambarkan variasi antara citra wajah. Tahapan-tahapan dalam mengambil fitur dengan metode ini adalah:

- Menghitung nilai rata-rata citra

$$\mu = \frac{1}{N} \sum_{k=1}^N x_k \quad (2.6)$$

- Menghitung matriks kovarian citra

$$C = \sum_{k=1}^n (x_k - \mu) (x_k - \mu)^T \quad (2.7)$$

- Menghitung *eigenvalue* dan *eigenvector* PCA

$$Cu_n = \lambda_n u_n \quad (2.8)$$

Keterangan:

- u = *eigenvector*

- λ = *eigenvalue*

- Mengurutkan nilai *eigen* dari terbesar hingga kecil dan mengeliminasi nilai *eigenvalue* yang kecil
- Menentukan nilai *eigenface* yang akan diambil

$$\frac{\sum_{i=1}^M \mu_i}{\sum_{i=1}^M \mu_k} = A \quad (2.9)$$

Digunakan *Eigenvector* PCA yang sudah didapat untuk mengekstraksi fitur yang selanjutnya dipropagasikan kedalam analisis diskriminan linear. Tahapan untuk mengambil fitur yaitu sebagai berikut:

- Menghitung nilai rata-rata citra dalam tiap kelas

$$\mu_k = \frac{1}{N_k} \sum_{m=1}^{N_k} x_k m \quad (2.10)$$

- Menghitung kovarian citra antar kelas

$$S_B = \sum_{k=1}^C N_k (\mu_k - \mu) (\mu_k - \mu)^T \quad (2.11)$$

- Menghitung kovarian citra dalam kelas

$$S_W = \sum_{k=1}^C \sum_{x \in X_k} (x_k - \mu_k) (x_k - \mu_k)^T \quad (2.12)$$

- Menghitung *eigenvalue* dan *eigenvector* LDA

$$S_B u_i = \lambda_i S_W u_i \quad (2.13)$$

- Menghitung *eigenvalue* dan *eigenvector* gabungan

$$u_{opt} = u_{PCA} u_{LDA} \quad (2.14)$$

Pengenalan wajah berdasarkan metode PCA rata-rata menggunakan matriks sebaran atau matriks sebaran antar kelas untuk mengekstraksi fitur wajah. Jika total matriks sebaran digunakan sebagai matriks pembangkit, matriks proyeksi optimal sama dengan menghitung nilai maksimal *eigen* dan sesuai *eigenvector* dari S_t , dan matriks proyeksi optimal (X_1, X_2, \dots, X_d) adalah *eigenvector* yang berhubungan dengan d nilai terbesar yang menyamaratakan nilai *eigen*. Oleh karena itu, (X_1, X_2, \dots, X_d) menggambarkan kontribusi antara masing-masing

eigenface dalam merepresentasikan masukan gambar wajah, maka kita dapat mengekstraksi fitur wajah melalui ini.

Metode yang digunakan dalam penelitian ini adalah metode *Principal Component Analysis* (PCA) sebagai pengekstraksi fitur, karena prinsip dasar Algoritma PCA adalah menentukan dan mempertahankan komponen dari sekumpulan citra yang mempunyai distribusi maksimum. PCA digunakan untuk mendapatkan vektor-vektor atau disebut juga sebagai *principal component* yang dapat memberikan informasi tentang varian maksimum *database* wajah. Setiap *principal component* merupakan representasi dari suatu kombinasi linier dari semua citra-citra wajah *training* yang sudah dikurangi dengan *mean* citra. Gabungan dari citra wajah inilah yang dinamakan *Eigenfaces*. *Eigenfaces* ini adalah merupakan fitur dari suatu citra wajah yang akan dikenali. Maka dari itu metode PCA ini akan dipadukan dengan metode *Convolutional Neural Network* (CNN), merupakan metode *deep learning* yang sudah terbukti sangat efisien dalam klasifikasi citra, yang digunakan untuk melakukan proses pembelajaran pada komputer untuk mencari representasi terbaik. CNN terdiri dari tiga tahapan, yaitu *Input data*, *Feature learning*, dan *Classification*. Setiap data masukan akan melalui ketiga tahapan tersebut dengan proses *filtering*. Sehingga dapat diperoleh sistem pengenalan ekspresi wajah secara *realtime* yang akurat meskipun pencahayaan yang kurang atau lebih dengan melalui pengenalan spesifik pada ekspresi wajah ataupun posisi wajah.

2.5 Metode Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan metode terbaru yang didesain dari pengembangan *Multilayer Perceptron* (MLP) untuk mengolah data dua dimensi. CNN adalah salah satu *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada studi kasus pengklasifikasian citra, metode MLP tidak cocok untuk diimplementasikan karena tidak dapat digunakan untuk menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik (Abhirawan *et al.*, 2017).

CNN dikembangkan pertama kali dengan nama *Neo Cognition* oleh Kunihiko Fukushima, seorang peneliti dari NHK *Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang*. Konsep tersebut dimatangkan oleh Yann Le Chun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Metode CNN yang bernama Le Net sukses diimplementasikan oleh Le Chun pada penelitiannya tentang pengenalan angka dan tulisan tangan. Tahun 2012, Alex Krizhevsky dengan menerapkan metode CNN sukses memenangkan kompetisi *ImageNet Large Scale Visual Recognition Challenge 2012*. Prestasi ini menjadikan ajang pembuktian bahwa metode *Deep Learning*, dalam konteks ini CNN. Metode CNN bisa membuktikan keunggulan metode *Machine Learning* lainnya seperti SVM pada kasus klasifikasi objek citra.

Dalam bentuknya yang paling umum, konvolusi adalah operasi pada dua fungsi argumen bernilai nyata. Untuk memotivasi definisi konvolusi, kita mulai dengan contoh dua fungsi yang mungkin kita gunakan (Ian et al., 2014). Misalkan kita sedang melacak lokasi pesawat ruang angkasa dengan sensor laser. Sensor laser kita menyediakan *output* tunggal $x(t)$, posisi pesawat ruang angkasa pada waktu t . Baik x maupun t bernilai nyata, yaitu kita bisa mendapatkan pembacaan yang berbeda dari sensor laser setiap saat dalam waktu.

Sekarang anggaplah bahwa sensor laser kita agak bising. Untuk memperoleh perkiraan yang tidak terlalu berisik dari posisi pesawat ruang angkasa, kita ingin pengukuran beberapa rata-rata. Tentu saja, pengukuran yang lebih baru yang lebih relevan, jadi kita ingin ini menjadi bobot rata-rata yang memberikan bobot lebih untuk pengukuran terbaru. Kita dapat melakukan ini dengan fungsi pembobotan $w(a)$, di mana a adalah usia pengukuran. Jika kita menerapkan operasi bobot rata-rata tersebut setiap saat, kita memperoleh fungsi baru s memberikan perkiraan halus posisi kapal ruang angkasa:

$$s(t) = \int x(a)\omega(t - a)da. \quad (2.15)$$

Operasi ini disebut konvolusi (*Convolution*). Operasi konvolusi biasanya dilambangkan dengan tanda bintang:

$$s(t) = (x * \omega)(t) \quad (2.16)$$

Dalam contoh, w harus menjadi fungsi kepadatan probabilitas yang valid, atau *output* tidak akan menjadi bobot rata-rata. Juga, w perlu menjadi 0 untuk

semua argumen negatif, atau itu akan melihat ke masa depan, yang mungkin di luar kemampuan kita. Namun keterbatasan ini khusus untuk contoh kita. Secara umum, konvolusi didefinisikan untuk setiap fungsi *integral* yang didefinisikan di atas dan dapat digunakan untuk tujuan yang lain selain mengambil bobot rata-rata.

Dalam terminologi jaringan konvolusi, argumen pertama (dalam contoh ini, fungsi x) ke konvolusi sering disebut sebagai *input*, dan argumen kedua (dalam contoh ini, fungsi w) sebagai *kernel*. Outputnya kadang-kadang disebut sebagai peta fitur. Sebagai contoh, gagasan sensor laser yang dapat memberikan pengukuran setiap saat tidaklah realistis. Biasanya, ketika kita bekerja dengan data di komputer, waktu akan didiskritisasi, dan sensor kita akan memberikan data secara berkala. Sebagai contoh, mungkin lebih realistis untuk mengasumsikan bahwa laser memberikan pengukuran sekali per detik. Indeks waktu t kemudian hanya dapat mengambil nilai *integer*. Jika sekarang kita mengasumsikan bahwa x dan w didefinisikan hanya pada *integer* t , kita dapat mendefinisikan konvolusi diskrit:

$$s(t) = (x * \omega)(t) = \sum_{\alpha=-\infty}^{\infty} x(\alpha)\omega(t - \alpha). \quad (2.17)$$

Dalam aplikasi pembelajaran mesin (*machine learning*), *input* biasanya berupa *array* data multi dimensi, dan kernel biasanya merupakan *array parameter* multi dimensi yang diadaptasi oleh algoritma pembelajaran. Kami akan menyebut *array* multi dimensi ini sebagai *tensor*. Karena setiap elemen dari *input* dan kernel harus disimpan secara eksplisit dan terpisah, kami biasanya mengasumsikan bahwa fungsi-fungsi ini nol, tetapi dalam himpunan titik hingga kami menyimpan nilai-nilai tersebut. Ini berarti bahwa dalam praktiknya, kita dapat mengimplementasikan penjumlahan tak terbatas sebagai penjumlahan atas sejumlah elemen *array* yang terbatas.

Akhirnya, kita sering menggunakan konvolusi lebih dari satu sumbu pada suatu waktu. Untuk contoh, jika kita menggunakan gambar dua dimensi I sebagai input kita, kita mungkin juga ingin menggunakan kernel dua dimensi K :

$$s(i, j) = (I * K)(i, j) = \sum_m \sum_n (m, n) K(i - m, j - n). \quad (2.18)$$

Konvolusi bersifat komutatif, artinya kita dapat menulis dengan setara

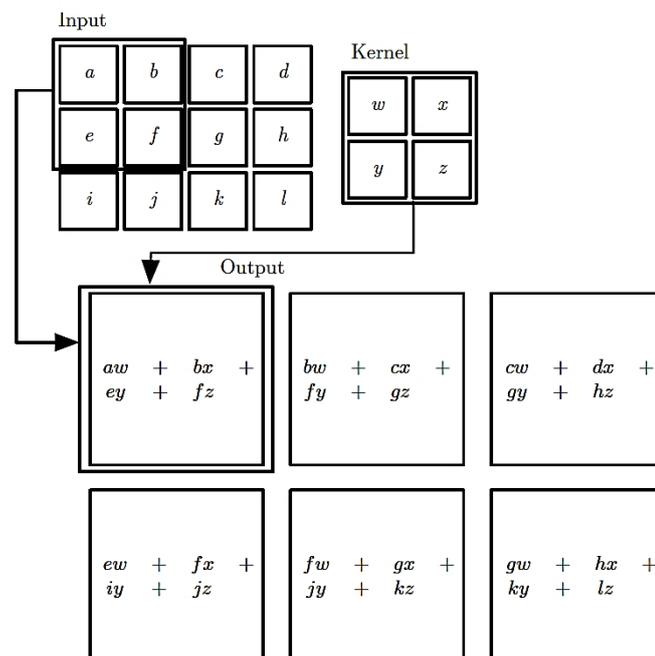
$$s(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n) K(m, n). \quad (2.19)$$

Biasanya rumus yang terakhir ini lebih mudah diterapkan di perpustakaan pembelajaran mesin (*machine learning library*), karena ada sedikit variasi dalam kisaran nilai valid m dan n .

Properti komutatif konvolusi muncul karena kita telah membalik kernel relatif terhadap input, dalam arti bahwa ketika m meningkat, indeks ke dalam input meningkat, tetapi indeks ke dalam kernel menurun. Satu-satunya alasan untuk membalik kernel adalah untuk mendapatkan properti komutatif. Sementara properti komutatif berguna untuk menulis bukti, itu biasanya bukan properti penting dari implementasi jaringan saraf. Sebaliknya, banyak pustaka jaringan saraf menerapkan fungsi terkait yang disebut korelasi silang (*cross-correlation*), yang sama dengan konvolusi tetapi tanpa membalik kernel:

$$s(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n) K(m, n). \quad (2.20)$$

Banyak pustaka pembelajaran mesin menerapkan korelasi silang tetapi menyebutnya konvolusi. Dalam teks ini kami mengikuti konvensi ini yang menyebut kedua operasi konvolusi dan menentukan apakah kami bermaksud membalik kernel atau tidak dalam konteks di mana membalik kernel relevan. Dalam konteks pembelajaran mesin, algoritma pembelajaran akan mempelajari nilai-nilai yang sesuai dari kernel di tempat yang tepat, sehingga suatu algoritma berdasarkan konvolusi dengan membalik kernel akan mempelajari kernel yang dibalik relatif terhadap kernel yang dipelajari oleh suatu algoritma tanpa membalik. Jarang konvolusi digunakan sendirian dalam pembelajaran mesin; alih-alih konvolusi digunakan bersamaan dengan fungsi-fungsi lain, dan kombinasi fungsi-fungsi ini tidak berpindah-pindah terlepas dari apakah operasi konvolusi membalik kernelnya atau tidak. Seperti gambar berikut ini:



Gambar 2.5 Contoh konvolusi 2-D tanpa *flipping kernel*. (Ian *et al.*, 2014).

Kami membatasi *output* hanya pada posisi di mana kernel terletak sepenuhnya di dalam gambar, yang disebut konvolusi “*valid*” dalam beberapa konteks. Kami menggambar kotak dengan panah untuk menunjukkan bagaimana elemen kiri atas dari *tensor output* dibentuk dengan menerapkan kernel ke daerah kiri atas yang sesuai dari *tensor input*. Gambar alami memiliki banyak sifat statistik yang tidak berbeda dengan terjemahan. Misalnya, foto kucing tetap menjadi foto kucing jika diterjemahkan satu piksel ke kanan. CNN mempertimbangkan properti ini dengan berbagi *parameter* di beberapa lokasi gambar. Fitur yang sama (unit tersembunyi dengan bobot yang sama) dihitung di berbagai lokasi dalam *input*. Ini berarti bahwa kita dapat menemukan kucing dengan detektor kucing yang sama apakah kucing muncul di kolom i atau kolom $i + 1$ pada gambar. Berbagi *parameter* telah memungkinkan CNN menurunkan secara dramatis jumlah parameter model yang unik dan untuk secara signifikan meningkatkan ukuran jaringan tanpa memerlukan peningkatan yang sesuai dalam data pelatihan. Itu tetap salah satu contoh terbaik tentang bagaimana secara efektif memasukkan pengetahuan domain ke dalam arsitektur jaringan (Ian *et al.*, 2014).

Bobot pembusukan bertindak dengan menempatkan penalti langsung pada *parameter model*. Strategi lain adalah memberikan penalti pada aktivasi unit dalam

jarangan saraf, mendorong aktivasi mereka menjadi jarang. Ini secara tidak langsung memberikan penalti yang rumit pada parameter model. *Sparsity representasional*, di sisi lain, menggambarkan representasi dimana banyak elemen representasi adalah nol (atau mendekati nol). Pandangan yang disederhanakan dari perbedaan ini dapat diilustrasikan dalam konteks *linear* regresi:

$$\begin{aligned} \begin{bmatrix} 18 \\ 5 \\ 15 \\ -9 \\ -3 \end{bmatrix} &= \begin{bmatrix} 4 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & -1 & 0 & 3 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & -4 \\ 1 & 0 & 0 & 0 & -5 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ -2 \\ -5 \\ 1 \\ 4 \end{bmatrix} \\ \mathbf{y} \in \mathbb{R}^m & \quad \mathbf{A} \in \mathbb{R}^{m \times n} & \quad \mathbf{x} \in \mathbb{R}^n \end{aligned} \quad (2.21)$$

$$\begin{aligned} \begin{bmatrix} -14 \\ 1 \\ 19 \\ 2 \\ 23 \end{bmatrix} &= \begin{bmatrix} 3 & -1 & 2 & -5 & 4 & 1 \\ 4 & 2 & -3 & -1 & 1 & 3 \\ -1 & 5 & 4 & 2 & -3 & -2 \\ 3 & 1 & 2 & -3 & 0 & -3 \\ -5 & 4 & -2 & 2 & -5 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \\ -3 \\ 0 \end{bmatrix} \\ \mathbf{y} \in \mathbb{R}^m & \quad \mathbf{B} \in \mathbb{R}^{m \times n} & \quad \mathbf{h} \in \mathbb{R}^n \end{aligned} \quad (2.22)$$

Dalam ungkapan pertama, kami memiliki contoh model regresi *linier* dengan parameter jarang. Dalam yang kedua, kami memiliki regresi *linier* dengan representasi yang jarang dari data x . Artinya, h adalah fungsi dari x , yang dalam arti tertentu, mewakili informasi dalam x , tetapi begitu juga dengan vektor yang jarang. Representasi regulator dilakukan dengan mekanisme yang sama yang telah kita gunakan dalam regularisasi parameter. Norma hukuman regularisasi dari representasi dilakukan dengan menambah *loss function* J , dimana norma hukuman pada representasi. Hukuman ini dilambangkan $\Omega(h)$. Seperti sebelumnya, kami menunjukkan fungsi kerugian yang diatur oleh:

$$\tilde{J}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{y}) = J(\boldsymbol{\theta}; \mathbf{X}, \mathbf{y}) + \alpha \Omega(\mathbf{h}), \quad (2.23)$$

Dimana $\alpha \in [0, \infty)$ bobot kontribusi relatif dari istilah hukuman norma, dengan nilai yang lebih besar dari α sesuai dengan regularisasi lebih.

2.6 Tipe Layer CNN

Ada banyak jenis lapisan yang digunakan untuk membangun *Convolutional Neural Network* (CNN), tetapi yang paling mungkin ditemui adalah: *Convolutional* (CONV), *Activation* (ACT atau RELU, tempat menggunakan aktivasi yang sama), *Pooling* (POOL), *Fully-connected* (FC), *Batch Normalization* (BN) dan *Dropout* (DO) (Rosebrock, 2017)

Menumpuk serangkaian lapisan ini dengan cara tertentu menghasilkan CNN. Kami sering menggunakan diagram teks sederhana untuk menggambarkan CNN: INPUT => CONV => RELU => FC => SOFTMAX. Disini kita mendefinisikan CNN sederhana yang menerima input, menerapkan lapisan konvolusi, kemudian lapisan aktivasi, kemudian lapisan yang terhubung penuh dan akhirnya sebuah *classifier softmax* untuk mendapatkan *output* probabilitas klasifikasi. Lapisan aktivasi SOFTMAX sering dihilangkan dari diagram jaringan seperti yang diasumsikan langsung mengikuti FC final.

Dari jenis lapisan ini, CONV dan FC (dan pada tingkat lebih rendah, BN) adalah satu-satunya lapisan yang mengandung parameter yang dipelajari selama proses pelatihan. Aktivasi dan *dropout layer* tidak dianggap sebagai “*layer*” itu sendiri, tetapi sering dimasukkan dalam diagram jaringan untuk membuat arsitektur jelas secara eksplisit. *Pooling layers* (POOL), yang sama pentingnya dengan CONV dan FC, juga dimasukkan dalam diagram jaringan karena mereka memiliki dampak substansial pada dimensi spasial suatu gambar saat bergerak melalui CNN. CONV, POOL, RELU dan FC adalah yang paling penting ketika mendefinisikan arsitektur jaringan anda yang sebenarnya. Itu bukan untuk mengatakan bahwa lapisan lain tidak kritis, tetapi mengambil kursi belakang ke set empat kritis ini karena mereka menentukan arsitektur aktual itu sendiri.

2.7 Layer Convolutional

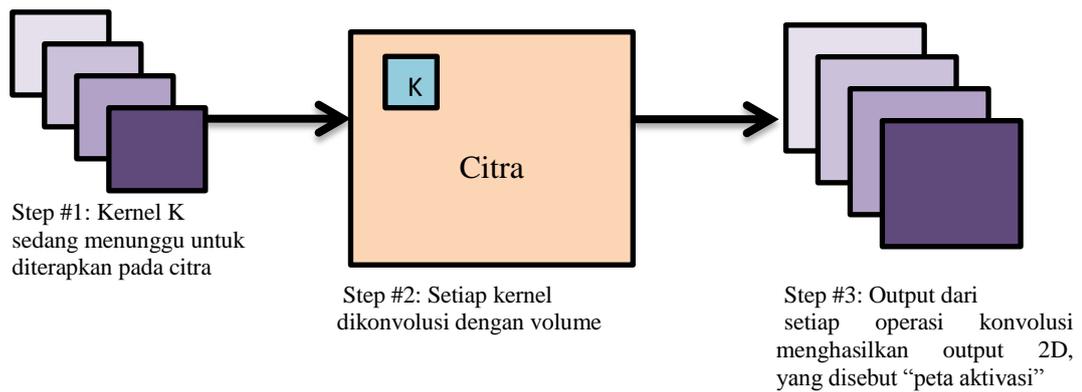
Lapisan CONV adalah blok bangunan inti dari *Convolutional Neural Network* (CNN). Parameter lapisan CONV terdiri dari satu set *filter* yang dapat dipelajari K (mis., “*Kernel*”), dimana setiap *filter* memiliki lebar dan tinggi, dan hamper selalu persegi. Filter-filter ini kecil (dalam hal dimensi spasial mereka) tetapi diperluas sepanjang kedalaman penuh volume.

Untuk *input* ke CNN, kedalamannya adalah jumlah saluran dalam gambar (mis., kedalaman tiga saat bekerja dengan gambar RGB, satu untuk setiap saluran). Untuk volume yang lebih dalam di jaringan, kedalaman akan menjadi jumlah filter yang diterapkan pada lapisan sebelumnya.

Untuk membuat konsep ini lebih jelas, mari kita pertimbangkan umpan maju CNN, di mana kami menggabungkan setiap *filter* K melintasi lebar dan tinggi *volume input*. Lebih sederhana dapat dilihat pada Gambar 2.6, kita dapat memikirkan setiap kernel K kami meluncur melintasi wilayah *input*, menghitung perkalian *element-wise*, menjumlahkan dan kemudian menyimpan nilai *ouput* dalam peta aktivasi 2-dimensi seperti pada Gambar 2.6.

Setelah menerapkan semua *filter* K ke *volume input*, kami sekarang memiliki peta aktivasi 2 Dimensi K . Kami kemudian menumpuk peta aktivasi K kami di sepanjang dimensi kedalaman *array* kami untuk membentuk *volume* keluaran akhir seperti pada Gambar 2.7. Setiap entri dalam *volume output* dengan demikian merupakan *output* dari *neuron* yang “terlihat” hanya pada bagian kecil dari *input*.

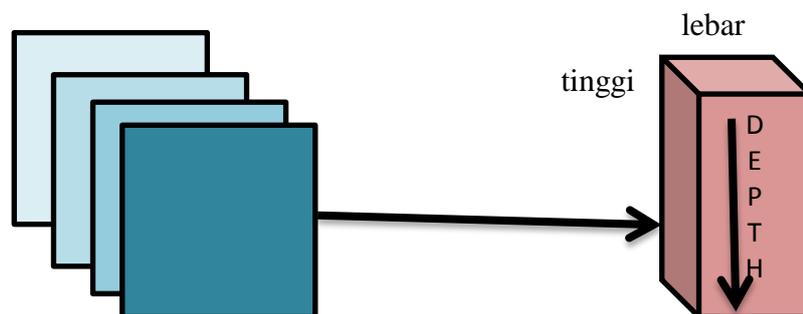
Dengan cara ini, jaringan “mempelajari” filter yang aktif ketika mereka melihat tipe fitur tertentu pada lokasi spasial tertentu dalam *volume input*. Di lapisan bawah jaringan, *filter* mungkin aktif ketika mereka melihat daerah seperti tepi atau sudut. Kemudian, di lapisan jaringan yang lebih dalam, *filter* dapat diaktifkan dihadapan level tinggi fitur, seperti bagian wajah, cakar anjing, kap mobil, dll. Konsep aktivasi ini kembali ke analogi jaringan saraf kita, neuron ini menjadi “bersemangat” dan “mengaktifkan” ketika lihat pola tertentu dalam gambar *input*.



Gambar 2.6 Kiri: Di setiap lapisan konvolusional di CNN, ada kernel K yang diterapkan pada *volume input*. Tengah: Setiap kernel K dilingkari dengan *volume input*. Kanan: Setiap kernel menghasilkan *output 2D*, yang disebut peta aktivasi.

(Rosebrock, 2017)

Konsep konvolusi *filter* kecil dengan *volume input* besar (r) memiliki arti khusus dalam *Convolutional Neural Networks* – khususnya, konektivitas lokal dan bidang penerimaan neuron. Saat bekerja dengan gambar, seringkali tidak praktis untuk menghubungkan neuron dalam volume saat ini ke semua neuron dalam volume sebelumnya. Ada terlalu banyak koneksi dan terlalu banyak bobot, sehingga tidak mungkin untuk melatih jaringan yang dalam pada gambar dengan dimensi spasial yang besar. Ketika menggunakan CNN, kami memilih untuk menyambungkan setiap neuron ke hanya wilayah lokal dari *volume input* – kami menyebut ukuran wilayah lokal ini bidang reseptif (atau sederhananya, variable F) dari neuron.



Gambar 2.7 Setelah mendapatkan peta aktivasi K, mereka disusun bersama untuk membentuk *volume input* ke lapisan berikutnya dalam jaringan. (Rosebrock, 2017)

Untuk memperjelas hal ini, mari kembali ke set data CIFAR-10 kami dimana *volume input* sebagai ukuran input $32 \times 32 \times 3$. Setiap gambar dengan

demikian memiliki lebar 32 piksel, tinggi 32 piksel dan kedalaman 3 (satu untuk setiap saluran RGB). Jika bidang reseptif kami berukuran 3×3 , maka masing-masing neuron masuk lapisan CONV akan terhubung ke wilayah lokal 3×3 dari gambar dengan total $3 \times 3 \times 3 = 27$ bobot (kedalaman *filter* adalah tiga karena memanjang melalui kedalaman penuh dari gambar input, dalam hal ini, tiga saluran).

Sekarang mari kita asumsikan bahwa dimensi spasial dari *volume input* kami telah direduksi menjadi ukuran yang lebih kecil, tetapi kedalaman kami sekarang lebih besar, karena menggunakan lebih banyak *filter* dalam jaringan, sehingga ukuran *volume* sekarang $16 \times 16 \times 94$. Sekali lagi, jika kita mengasumsikan bidang reseptif ukuran 3×3 , maka setiap neuron dilapisan CONV akan memiliki total $3 \times 3 \times 94 = 846$ koneksi ke *volume input*. Sederhananya, bidang reseptif F adalah ukuran *filter*, menghasilkan *kernel* $F \times F$ yang berbelit-belit dengan *volume input*.

Pada titik ini kami telah menjelaskan konektivitas neuron dalam *volume input*, tetapi bukan pengaturan atau ukuran *volume output*. Ada tiga parameter yang mengontrol ukuran *volume output*: kedalaman, langkah dan ukuran *zero-padding*, masing-masing akan kami ulas di bawah ini.

2.7.1 Kedalaman atau *Depth*

Kedalaman *volume output* mengontrol jumlah neuron (mis., *Filter*) di lapisan CONV yang terhubung ke wilayah lokal *volume input*. Setiap *filter* menghasilkan peta aktivasi yang “mengaktifkan” dihadapan tepi berorientasi atau gumpalan atau warna.

Untuk lapisan CONV yang diberikan, kedalaman peta aktivasi adalah K , atau hanya jumlah *filter* yang kita pelajari di lapisan saat ini. Set *filter* yang “melihat” lokasi input $(x;y)$ yang sama disebut kolom kedalaman.

2.7.2 Langkah atau *Stride*

Operasi konvolusi sebagai “mengeser” matriks kecil melintasi matriks besar, berhenti pada setiap koordinat, menghitung perkalian dan jumlah *element-wise*, kemudian menyimpan *output*. Tentang konvolusi di atas, kami masing-

masing hanya mengambil satu langkah piksel teratas. Dalam konteks CNN, prinsip yang sama dapat diterapkan untuk setiap langkah, kami membuat kolom kedalaman baru di sekitar wilayah lokal gambar tempat kami menggabungkan setiap filter K dengan wilayah dan menyimpan *output* dalam *volume* 3D. Saat membuat layer CONV kami, kami biasanya menggunakan ukuran langkah-langkah S dari $S = 1$ atau $S = 2$.

Langkah kecil akan menyebabkan bidang reseptif yang tumpang tindih dan *volume output* yang lebih besar. Sebaliknya, langkah yang lebih besar akan menghasilkan bidang reseptif yang kurang tumpang tindih dan *volume output* yang lebih kecil. Untuk membuat konsep langkah konvolusional lebih konkret. Pertimbangkan Tabel 2.1 di mana kami memiliki gambar input 5×5 (kiri) Bersama dengan kernel Laplacian 3×3 (kanan).

Dengan menggunakan $S = 1$, kernel kami meluncur dari kiri ke kanan dan dari atas ke bawah, satu piksel pada satu waktu. Namun, jika kami menerapkan operasi yang sama, hanya saja kali ini dengan langkah $S = 2$, kita melewati dua piksel sekaligus (dua piksel sepanjang sumbu x dan dua piksel sepanjang sumbu y), menghasilkan yang lebih kecil *volume* keluaran (kanan).

95	242	186	152	39	
39	14	220	153	180	0
5	247	212	54	46	1
46	77	133	110	74	0
156	35	74	93	116	1

Tabel 2.1 Gambar input 5×5 (kiri) yang akan kami gabungkan dengan kernel Laplace (kanan) (Rosebrock, 2017)

Kali ini dengan langkah $S = 2$, kita melewati dua piksel sekaligus (dua piksel sepanjang sumbu x dan dua piksel sepanjang sumbu y), menghasilkan *volume output* yang lebih kecil (kanan).

Dengan demikian, kita dapat melihat bagaimana lapisan konvolusi dapat digunakan untuk mengurangi dimensi spasial dari *volume input* hanya dengan mengubah langkah kernel. Seperti yang akan kita lihat nanti di bagian ini, lapisan konvolusional dan lapisan penyatuan adalah metode utama untuk mengurangi

ukuran input spasial. Itu bagian *pooling layers* juga akan memberikan contoh yang lebih visual tentang bagaimana ukuran langkah yang bervariasi akan mempengaruhi ukuran *output*.

692	-315	-6
-680	-194	305
153	-59	-85

692	-6
153	-86

Tabel 2.2 (Kiri): *output* konvolusi dengan langkah 1x1. (Kanan): *output* konvolusi dengan langkah 2x2. Perhatikan bagaimana langkah yang lebih besar dapat mengurangi dimensi spasial *input* (Rosebrock, 2017)

2.7.3 Zero-padding

Seperti yang kita ketahui, kita perlu “membalut” batas gambar untuk mempertahankan ukuran gambar asli saat menerapkan konvolusi, hal yang sama berlaku untuk *filter* di dalam CNN. Menggunakan *zero-padding*, kita dapat “mengisi” input kita di sepanjang perbatasan sehingga ukuran *volume output* kita cocok dengan ukuran *volume input* kita. Jumlah *padding* yang kami gunakan dikendalikan oleh parameter P.

Teknik ini sangat penting ketika kita mulai melihat arsitektur CNN mendalam yang menerapkan beberapa *filter* CONV di atas satu sama lain. Untuk memvisualisasikan *zero-padding*, lihat kembali tabel 2.1 di mana kami menerapkan kernel 3x3 Laplacian ke gambar input 5x5 dengan langkah $S = 1$.

Kita dapat melihat pada tabel 2.3 (kiri) bagaimana *volume output* lebih kecil (3x3) daripada *volume input* (5x5) karena sifat operasi konvolusi. Jika sebaliknya kita menetapkan $P = 1$, kita dapat mengisi *volume input* kita dengan nol (tengah) untuk membuat *volume* 7x7 dan kemudian menerapkan operasi konvolusi, yang mengarah ke ukuran *volume output* yang cocok dengan ukuran *volume input* asli yaitu 5x5 (kanan).

Tanpa *zero-padding*, dimensi spasial dari *volume input* akan berkurang terlalu cepat, dan kami tidak akan dapat melatih jaringan yang dalam (karena *volume input* akan terlalu kecil untuk mempelajari pola yang bermanfaat).

Menyatukan semua parameter ini, kita dapat menghitung ukuran *volume output* sebagai fungsi dari ukuran *volume input* (W , dengan asumsi gambar input persegi, yang hampir selalu seperti itu), ukuran bidang reseptif F , langkah S , dan jumlah *zero-padding* P . Untuk membangun lapisan CONV yang valid, kita perlu memastikan persamaan berikut adalah bilangan bulat:

$$\left(\frac{(W - F + 2P)}{S}\right) + 1 \quad (2.24)$$

Jika itu bilangan bulat, maka langkahnya diatur secara tidak benar, dan neuron tidak dapat di-*tile* seperti itu bahwa mereka cocok di *volume input* dengan cara yang simetris.

692	-315	-6
-680	-194	305
153	-59	-86

0	0	0	0	0	0	0
0	95	242	186	152	39	0
0	39	14	220	153	180	0
0	5	247	212	54	46	0
0	46	77	133	110	74	0
0	156	35	74	93	116	0
0	0	0	0	0	0	0

-99	-673	-130	-230	176
-42	692	-315	-6	-482
312	-680	-194	305	124
54	153	-59	-86	-24
-543	167	-35	-72	-297

Tabel 2.3 (Kiri): *Output* penerapan konvolusi 3x3 ke *output* 5x5 (mis., Dimensi spasial berkurang). (Kanan): Menerapkan *zero-padding* ke *input* asli dengan $P = 1$ meningkatkan dimensi spasial menjadi 7x7. (Bawah): Setelah menerapkan konvolusi 3x3 ke *input*, *volume output* cocok dengan ukuran *volume input* asli 5x5, sehingga *zero-padding* membantu kami melestarikan dimensi spasial

(Rosebrock, 2017)

2.7.4 Layer Aktivasi / Activation Layers

Setelah setiap lapisan CONV dalam CNN, kami menerapkan fungsi aktivasi nonlinier, seperti ReLU, ELU atau varian *Leaky ReLU*. Kami biasanya menunjukkan lapisan aktivasi sebagai ReLU dalam diagram jaringan seperti sejak

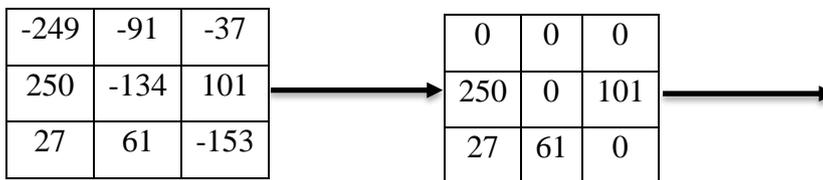
aktivasi ReLU yang paling umum digunakan, kami mungkin juga hanya menyatakan ACT dalam kedua kasus, kami membuatnya jelas bahwa fungsi aktivasi sedang diterapkan di dalam arsitektur jaringan.

Lapisan aktivasi tidak secara teknis “lapisan” (karena fakta bahwa tidak ada parameter/bobot yang dipelajari dalam lapisan aktivasi) dan kadang-kadang dihilangkan dari diagram arsitektur jaringan karena diasumsikan bahwa aktivasi segera mengikuti konvolusi.

Dalam hal ini, akan menyebutkan fungsi aktivasi mana yang mereka gunakan setelah setiap lapisan CONV di suatu tempat di kertas mereka. Sebagai contoh, pertimbangkan arsitektur jaringan berikut: INPUT => CONV => RELU => FC. Untuk membuat diagram ini lebih ringkas, kita cukup menghapus komponen RELU karena diasumsikan bahwa aktivasi selalu mengikuti konvolusi: INPUT => CONV => FC.

2.7.5 Pooling Layer / Layer Kolom

Ada dua metode untuk mengurangi ukuran *volume input*, lapisan dengan >1 (yang telah kita lihat) dan lapisan POOL. Adalah umum untuk menyisipkan lapisan POOL di antara berturut-turut.



Gambar 2.8 Contoh *volume input* yang melalui aktivasi ReLU, maks (0;x).

aktivasi dilakukan di tempat sehingga tidak perlu membuat *volume output* terpisah meskipun mudah untuk memvisualisasikan aliran jaringan dengan cara ini (Rosebrock, 2017)

Lapisan CONV dalam arsitektur CNN: INPUT => CONV => RELU => POOL => CONV => RELU => POOL => FC

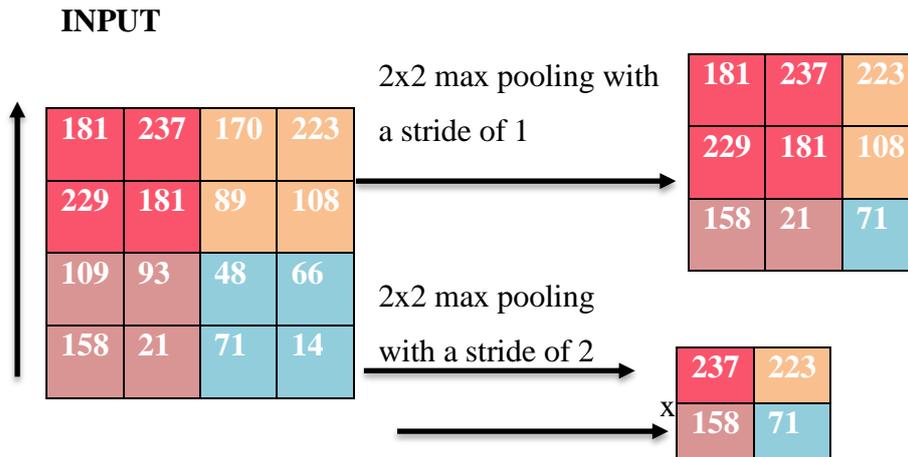
Fungsi utama dari layer POOL adalah untuk secara progresif mengurangi ukuran spasial (mis., lebar dan tinggi) dari *volume input*. Melakukan hal ini

memungkinkan kita untuk mengurangi jumlah parameter dan perhitungan dalam jaringan, pengumpulan juga membantu kita mengontrol *overfitting*.

Lapisan POOL beroperasi pada setiap irisan kedalaman input secara independen menggunakan fungsi maks atau rata-rata. *Max pooling* biasanya dilakukan di tengah arsitektur CNN untuk mengurangi ukuran spasial, sedangkan *pooling* rata-rata biasanya digunakan sebagai lapisan akhir jaringan (mis., GoogleNet, SqueezeNet, ResNet) dimana kami ingin menghindari penggunaan lapisan FC seluruhnya. Jenis paling umum dari POOL layer adalah *Max Pooling*, meskipun tren ini berubah dengan pendahuluan mikro-arsitektur yang lebih eksotis.

Biasanya kami akan menggunakan ukuran kumpulan 2x2, meskipun CNN lebih dalam yang menggunakan gambar *input* yang lebih besar (>200 piksel) dapat menggunakan ukuran kumpulan 3x3 di awal arsitektur jaringan. Kami juga biasanya mengatur langkah ke $S = 1$ atau $S = 2$. Gambar 2.8 (Rosebrock, 2017) mengikuti contoh penerapan *max pooling* dengan ukuran 2x2 *pool* dan langkah $S = 1$, perhatikan untuk setiap blok 2x2, kami hanya menyimpan nilai terbesar, mengambil satu langkah (seperti jendela geser), dan menerapkan operasi lagi sehingga menghasilkan ukuran *volume output* 3x3.

Kami selanjutnya dapat mengurangi ukuran *volume output* kami dengan meningkatkan langkahnya, disini kami menerapkan $S = 2$ untuk *input* yang sama (Gambar 2.8) untuk setiap blok 2x2 di input. Kami hanya menyimpan nilai terbesar, kemudian mengambil langkah dua piksel, dan menerapkan operasi lagi. Penggabungan ini memungkinkan kami untuk mengurangi lebar dan tinggi dengan faktor dua, secara efektif membuang 75% aktivasi lapisan sebelumnya.



Gambar 2.9 (Kiri): *Volume Input* 4x4. (Kanan): Menerapkan penyatuan 2x2 max dengan langkah $S = 1$ (Bawah): Menerapkan penyatuan 2x2 max dengan $S = 2$. Ini secara dramatis mengurangi dimensi spasial dari input kami (Rosebrock, 2017).

2.7.6 Layer *Fully-connected*

Neuron di lapisan FC sepenuhnya terhubung ke semua aktivasi di lapisan sebelumnya, seperti standar untuk jaringan saraf *feedforward* yang telah kita bahas. Lapisan FC selalu ditempatkan di akhir jaringan (yaitu, kami tidak dapat menerapkan lapisan CONV, lalu lapisan FC, diikuti oleh lapisan lain CONV).

Adalah umum untuk menggunakan satu atau dua lapisan FC sebelum menerapkan pengklasifikasi *softmax*, seperti yang ditunjukkan oleh arsitektur berikut (disederhanakan):

INPUT => CONV => RELU => POOL => CONV => RELU => POOL => FC
=>FC

Disini kita menerapkan dua lapisan yang terhubung sepenuhnya sebelum pengklasifikasi *softmax* (tersirat) kami yang akan menghitung probabilitas hasil akhir kami untuk setiap kelas.

2.7.7 *Batch Normalization / Normalisasi Batch*

Pertama kali diperkenalkan oleh Loffe dan Szgedy dalam makalah 2015 mereka, normalisasi *Batch*: mempercepat pelatihan jaringan dalam dengan mengurangi internal *covariate shift*, lapisan normalisasi *batch* (atau singkatnya BN), seperti Namanya, digunakan untuk menormalkan aktivasi sebuah *volume input* sebelum meneruskannya ke lapisan berikutnya dalam jaringan.

Normalisasi *batch* telah terbukti sangat efektif dalam mengurangi jumlah *epoch* yang dibutuhkan untuk melatih jaringan saraf. Normalisasi *batch* juga memiliki manfaat tambahan untuk membantu pelatihan “menstabilkan”, memungkinkan variasi tingkat pembelajaran dan kekuatan regularisasi yang lebih besar. Menggunakan normalisasi *batch* tidak mengurangi kebutuhan untuk menyempurnakan parameter, tetapi itu akan membuat hidup anda lebih mudah dengan membuat tingkat belajar dan regularisasi menjadi lebih tidak stabil dan lebih mudah.

Kelemahan terbesar dari normalisasi *batch* adalah ia dapat benar-benar memperlambat waktu dinding yang diperlukan untuk melatih jaringan anda (meskipun anda akan membutuhkan lebih sedikit zaman untuk mendapatkan akurasi yang masuk akal) sebesar 2-3x karena perhitungan statistic per-*batch* dan normalisasi.

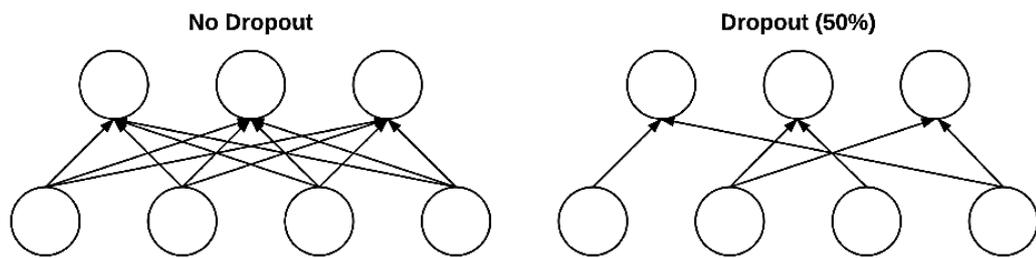
Yang mengatakan, saya sarankan menggunakan normalisasi *batch* di hampir setiap situasi karena itu membuat perbedaan yang signifikan. Seperti yang akan kita lihat nanti dalam buku ini, menerapkan normalisasi *batch* ke arsitektur jaringan kami dapat membantu kami mencegah *overfitting* dan memungkinkan kami untuk mendapatkan akurasi klasifikasi yang jauh lebih tinggi di zaman yang lebih sedikit dibandingkan dengan arsitektur jaringan yang sama tanpa normalisasi *batch*.

2.7.8 *Dropout*

Jenis *layer* terakhir yang akan kita diskusikan adalah *dropout*. *Dropout* sebenarnya adalah bentuk regularisasi yang bertujuan untuk membantu mencegah *overfitting* dengan meningkatkan akurasi pengujian, mungkin dengan mengorbankan akurasi pelatihan. Untuk setiap *mini batch* dalam *set* pelatihan

kami, *layer dropout*, dengan probabilitas p , lepaskan secara acak input dari lapisan sebelumnya ke lapisan berikutnya dalam arsitektur jaringan (Rosebrock, 2017)

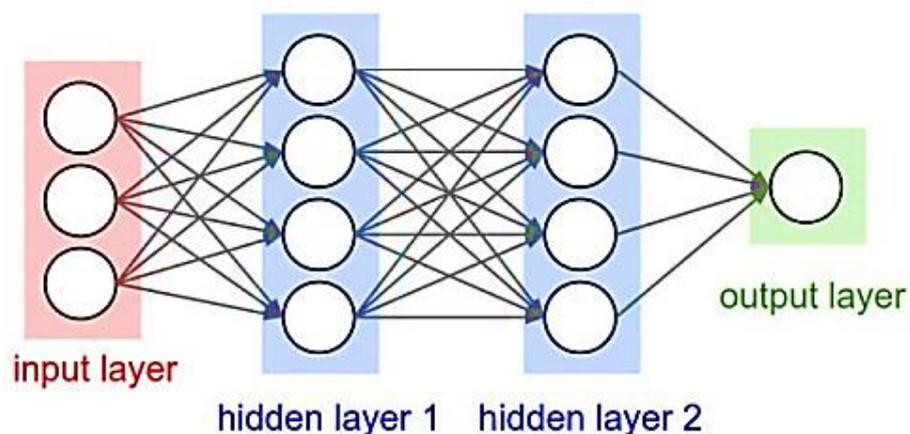
Gambar 2.10 memvisualisasikan konsep ini di mana kami memutuskan secara acak dengan probabilitas $P = 0.5$ koneksi antara dua lapisan FC untuk *batch-mini* yang diberikan. Sekali lagi perhatikan bagaimana setengah dari koneksi terputus untuk *mini batch*, kami menghubungkan kembali koneksi yang terputus, dan kemudian mencoba satu set koneksi lain untuk penurunan.



Gambar 2.10 (Kiri): Dua lapisan jaringan saraf yang sepenuhnya terhubung tanpa putus. (Kanan): Dua lapisan yang sama setelah menjatuhkan 50% koneksi (Rosebrock, 2017).

2.8 Cara Kerja CNN

Konsep kerja CNN mempunyai persamaan pada *Multi-Layer Perceptron* (MLP), akan tetapi dalam CNN setiap neuron diimplementasikan ke dalam tipe dua dimensi, sedangkan pada MLP, tiap neuron hanya mempunyai ukuran satu dimensi.



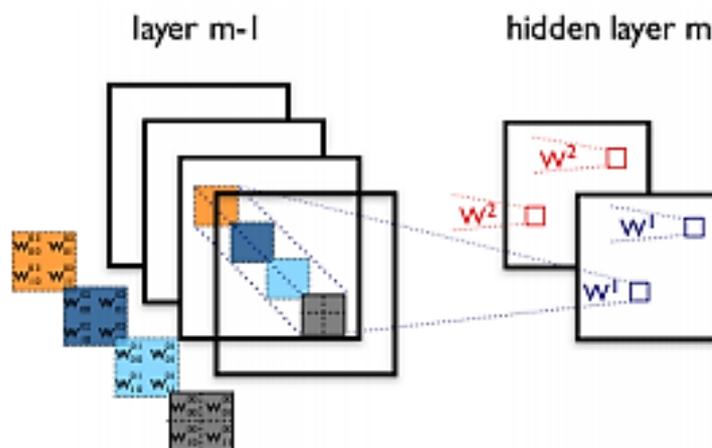
Gambar 2.11 Arsitektur MLP Sederhana (Ian et al., 2014)

Arsitektur MLP di Gambar 2.11 mempunyai *layer i* (warna merah dan biru) pada masing-masing *layer* berisi neuron j_i (lingkaran berwarna putih). MLP mendapatkan *input* berupa data satu dimensi dan mentransmisikan data tersebut kepada jaringan sehingga menciptakan *output*. Pada tiap relasi antara *neuron* dengan *layer* yang bersampingan mempunyai *parameter* bobot satu dimensi untuk menentukan kualitas mode. Pada tiap data *input layer* diberikan operasi linear dengan nilai bobot yang ada, selanjutnya hasil dari komputasi itu akan diganti dengan menggunakan operasi *nonlinear* yaitu fungsi aktivasi.

Pada metode CNN, data ditransmisikan dengan jaringan sehingga menjadi data dua dimensi, yang dapat menghasilkan operasi *linear* dan parameter bobot pada CNN menjadi berbeda. Pada metode operasi *linear* CNN dengan menggunakan operasi konvolusi, walaupun bobot tidak berukuran satu dimensi lagi, namun bertransformasi menjadi empat dimensi yang merupakan sekumpulan *kernel* konvolusi seperti pada Gambar 2.12 Dimensi bobot pada CNN adalah:

$$\text{neuron input} \times \text{neuron output} \times \text{tinggi} \times \text{lebar} \quad (2.24)$$

dikarenakan sifat proses konvolusi, jadi CNN hanya bisa digunakan pada data set yang mempunyai struktur dua dimensi seperti citra dan suara.



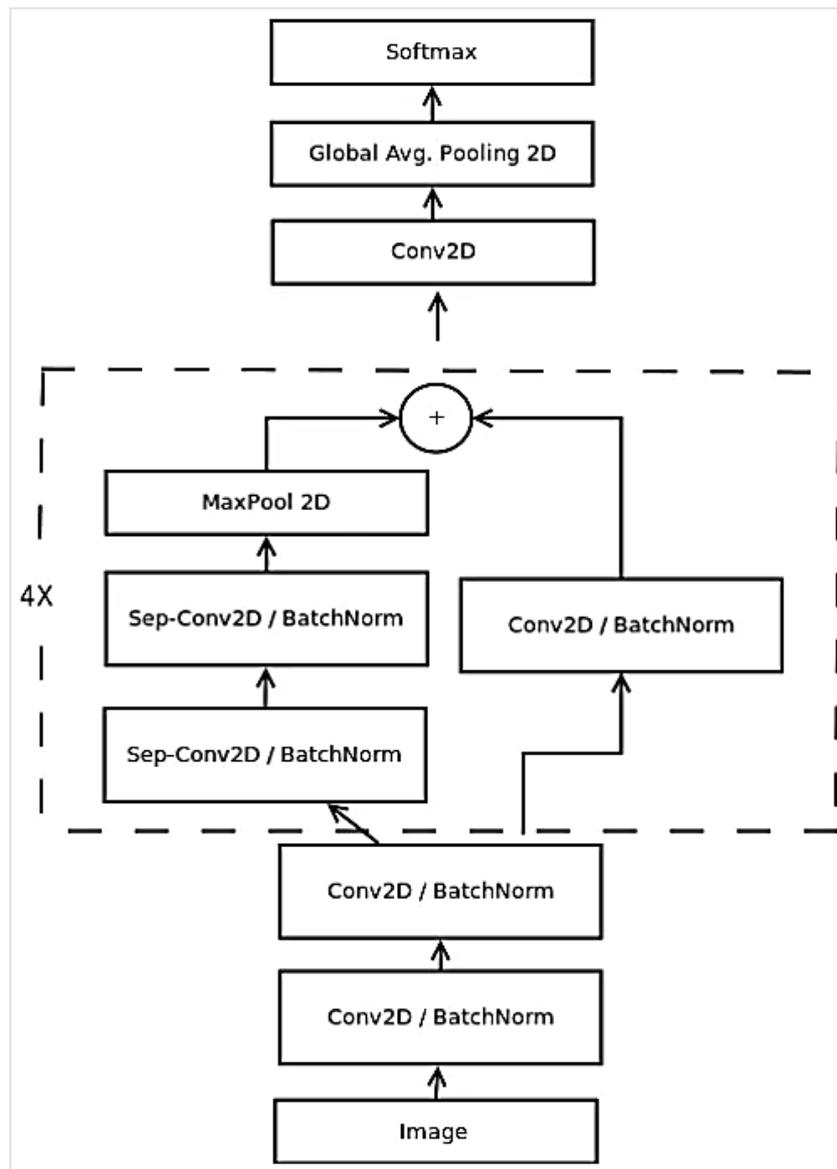
Gambar 2.12 Proses konvolusi pada CNN (Ian et al., 2014)

2.8.1 Model Training CNN: Mini Xception

Berikut ini arsitektur yang menarik yang relatif kecil dan mencapai hampir semua kinerja seni pengklasifikasi emosi diilustrasikan pada Gambar 2.13. Orang dapat memperhatikan bahwa blok tengah diulang sebanyak 4 kali dalam desain.

Arsitektur umum menggunakan lapisan yang terhubung sepenuhnya di ujung tempat sebagian besar parameter berada. Juga, mereka menggunakan konvolusi standar. Arsitektur CNN modern seperti *leverage Xception* dari kombinasi dua asumsi eksperimental paling sukses di CNN: penggunaan modul residual dan konvolusi yang dapat dipisahkan secara mendalam. Ada berbagai Teknik yang dapat diingat saat membangun jaringan saraf yang dalam dan berlaku di sebagian besar masalah penglihatan komputer. Beberapa Teknik yang digunakan saat melatih model CNN seperti pada Gambar 2.13:

1. **Augmentasi Data:** Lebih banyak data dihasilkan menggunakan set pelatihan dengan menerapkan transformasi. Diperlukan jika set pelatihan tidak cukup memadai untuk belajar representasi. Data gambar dihasilkan dengan mengubah gambar pelatihan actual dengan rotasi, memotong, menggeser, geser, *zoom*, *flip*, refleksi, normalisasi dll.
2. **Kernel_Regularizer:** memungkinkan untuk menerapkan penalti pada parameter lapisan selama optimasi. Penalti ini dimasukkan dalam fungsi kerugian yang dioptimalkan jaringan. Argument dalam lapisan konvolusi tidak lain adalah regularisasi bobot L2. penalti bobok puncak ini dan memastikan bahwa semua input dipertimbangkan.



Gambar 2.13 Arsitektur *Mini_Xception* yang diajukan untuk klasifikasi ekspresi (Arriaga, Valdenegro-Toro, & Plöger, 2017)

3. **Batch Normalisasi:** ini menormalkan aktivasi lapisan sebelumnya di setiap *batch*, yaitu menerapkan transformasi yang mempertahankan aktivasi rata-rata mendekati 0 dan deviasi standar aktivasi mendekati 1. Ini mengatasi masalah pergeseran kovariat internal. Itu juga bertindak sebagai regulator, dalam beberapa kasus menghilangkan kebutuhan untuk putus sekolah. Ini membantu mempercepat proses pelatihan.

4. ***Global Average Pooling***: ini mengurangi setiap peta fitur menjadi nilai scalar dengan mengambil rata-rata semua elemen dalam peta fitur. Operasi rata-rata memaksa jaringan untuk mengekstrak fitur global dari gambar input.
5. **Konvolusi dipisahkan secara *Depthwise***: Konvolusi ini terdiri dari dua lapisan berbeda: konvolusi *depth-wise* dan konvolusi *point-wise*. Konvolusi yang dapat dipisahkan dengan kedalaman mengurangi komputasi sehubungan dengan konvolusi standar dengan mengurangi jumlah parameter.

BAB III METODOLOGI PENELITIAN

3.1 Pendahuluan

Metodologi penelitian berkaitan dengan langkah-langkah kerja dalam melakukan penelitian. Bab ini akan membahas tahapan-tahapan dalam melakukan penelitian. Penelitian yang dilakukan berkaitan dengan sistem pengklasifikasian ekspresi wajah yang dilakukan melalui beberapa tahap sebelum mendapatkan *output database* yang diinginkan. Secara umum sistem ini terdiri dari beberapa tahap diantaranya pengambilan *input* data citra, pengolahan citra, ekstraksi fitur, klasifikasi dan evaluasi sistem. Maka dari itu pada metodologi penelitian ini adalah tahapan sistematis dan langkah-langkah yang akan dilakukan pada penelitian.

3.2 Kerangka Kerja (*Framework*)

Kerangka kerja yang digunakan pada penelitian ini diperlihatkan pada Gambar 3.1. Penelitian ini menggunakan *dataset* citra yang diambil melalui *Static Facial Expression in The Wild* (SFEW) 2.0 berisi 1073 citra dengan 6 ekspresi berbeda yaitu marah, jijik, takut, bahagia, sedih, dan terkejut.

Actions facial expressions in the wild (AFEW) adalah database ekspresi wajah temporal yang dinamis yang terdiri dari lingkungan dunia nyata yang dekat dari film. Itu dikumpulkan berdasarkan *subtitle* untuk *Deaf and Hearing Impaired* (SDH) dan *Closed Caption* (CC) untuk tujuan mencari konten terkait ekspresi dan mengekstraksi *time stamps* yang terkait dengan klip video yang mewakili beberapa gerakan wajah yang bermakna. Basis data berisi rentang usia subjek yang besar mulai 1-70 tahun. Informasi tentang klip telah disimpan dalam skema XML yang dapat diperpanjang dan subjek dalam klip tersebut telah dijelaskan dengan atribut seperti Nama, Usia Aktor, Usia Karakter, Pose, Jenis Kelamin, Ekspresi Orang dan Ekspresi Klip Keseluruhan.

Static Facial Expressions in the Wild (SFEW) telah dikembangkan dengan memilih bingkai dari AFEW. Basis data mencakup ekspresi wajah tanpa kendala, pose kepala bervariasi, rentang usia besar, oklusi, beragam fokus, resolusi wajah

berbeda dan dekat dengan iluminasi dunia nyata. Bingkai diekstraksi dari urutan AFEW dan diberi label berdasarkan label urutan. Secara total, SFEW berisi 700 gambar dan yang telah diberi label untuk enam ekspresi dasar marah, jijik, takut, bahagia, sedih, terkejut dan netral dan diberi label oleh dua tenaga kerja independen.

3.3 Pemodelan

Pada sistem ini terbagi menjadi 3 tahapan pengklasifikasian ekspresi wajah, yaitu sebagai berikut:

3.3.1 Tahap Input

Pada tahap ini, proses dilakukan di ruang terbuka. Perekaman wajah dilakukan ketika manusia menghadap kamera ke depan.

Sistem dimulai dengan akuisisi frame yang kemudian dilanjutkan dengan *preprocessing grayscale*. Setelah *preprocessing* dilanjutkan dengan deteksi bentuk manusia. Pendeteksian manusia dilakukan dengan menggunakan *frame difference* yang mendeteksi adanya gerakan. Jika tidak terdeteksi adanya gerakan maka sistem kembali ke proses *akuisisi frame video*. Jika kemudian terdeteksi adanya gerakan maka sistem melanjutkan proses untuk mendeteksi wajah dengan menggunakan algoritma *Viola-Jones*.

Setelah mendapatkan bentuk manusia, maka langkah selanjutnya adalah pendeteksian wajah. Objek wajah dicari dengan menggunakan metode *Viola-Jones* yang sering disebut sebagai algoritma *Haar Cascade Classifier*.

Saat dideteksi adanya wajah, maka sistem akan membuat *Bonding Box* berwarna merah yang kemudian citra wajah tersebut akan disimpan untuk masuk proses ekstraksi fitur.

3.3.2 Tahap Ekstraksi Fitur

Pada penelitian ini menggunakan ekstraksi fitur dengan metode *Principal Component Analysis (PCA)*. Hasil dari tahap input, akan dilakukan proses ciri yang disimpan sebagai model. Pada tahapan proses ekstraksi ciri, sebelum dilakukan pembelajaran ciri (*feature learning*) menggunakan CNN citra wajah

terlebih dahulu melalui proses pra-pengolahan menjadi citra *grayscale*, deteksi wajah, ekstraksi ciri menggunakan PCA dan filter pada *pre-trained model* yang didapat dari proses pelatihan dan proses klasifikasi ekspresi wajah. Berikut ini penjelasan tentang tahapan pelatihan dan pengujian dalam proses mengenali ekspresi wajah:

- a. **Praproses Citra**, sebelum dilakukan proses deteksi wajah, proses pertama yang dilakukan adalah melakukan praproses citra menjadi citra *greyscale*.
- b. **Deteksi Wajah**, pendeteksian wajah dalam penelitian ini menggunakan metode *Viola-Jones* yang telah dijelaskan pada Bab II.
- c. **Ekstraksi dan Pembelajaran Ciri**, Setelah citra wajah telah didapatkan pada proses deteksi wajah, citra kemudian melalui proses ekstraksi ciri menggunakan PCA. Proses ekstraksi ciri menggunakan PCA telah dijelaskan pada Bab II. Setelah ciri PCA didapatkan maka proses selanjutnya adalah proses pembelajaran ciri menggunakan CNN, penjelasan mengenai metode CNN telah dijelaskan pada Bab II. Setelah dilakukan ekstraksi ciri dan pembelajaran ciri maka *filter* yang didapat dalam proses pembelajaran ciri disimpan sebagai model untuk digunakan pada tahap pengujian.

3.3.3 Tahap Klasifikasi

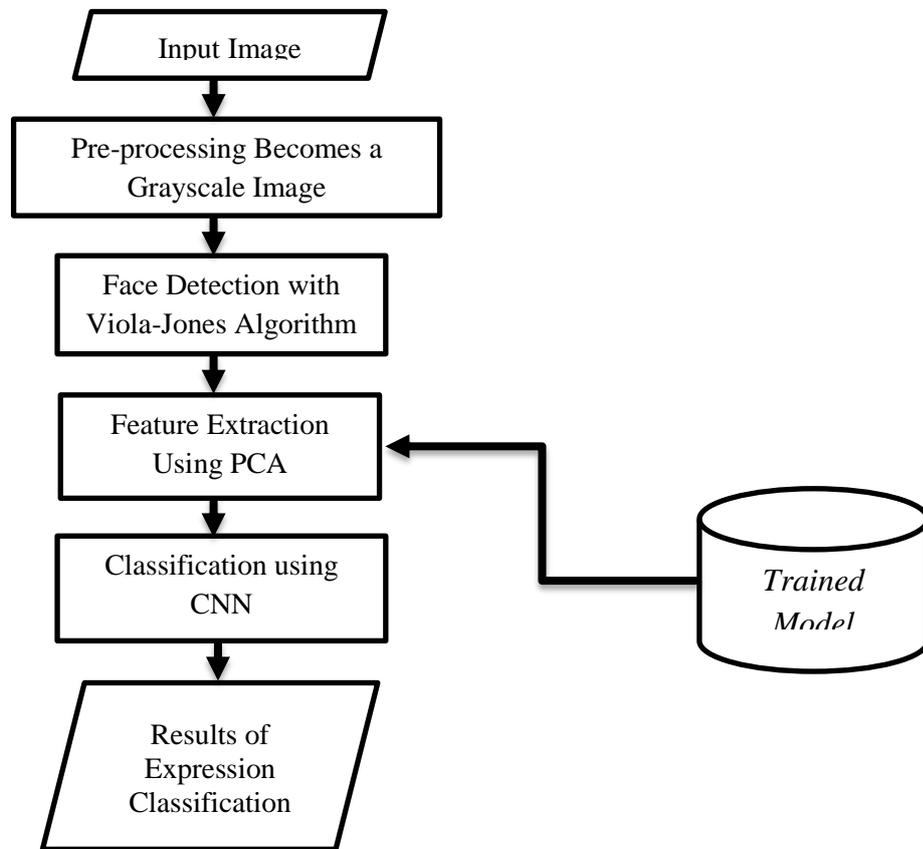
Setelah mendapatkan fitur uji dan fitur latih, maka akan diklasifikasikan untuk mendapatkan kelas pengenalan yang sesuai. Pada kajian ini akan menggunakan metode CNN sebagai klasifikasi ekspresi wajah.

3.4 Tahap Penelitian

Dalam melakukan penelitian mengenai pengklasifikasi ekspresi wajah secara *real time* menggunakan kombinasi metode *Principal Component Analysis* dan *Convolutional Neural Network*, penelitian akan dilakukan dengan tahapan-tahapan yang akan dijelaskan pada subbab 3.4.1 sampai 3.4.6.

3.4.1 Menetapkan Kriteria Pengujian

Pembahasan mengenai tahapan ini akan dijelaskan pada bab IV. Pada proses pelatihan model akan dilatih menggunakan satu bagian data yang telah dibagi dengan teknik validasi *k-fold cross validation*. Kemudian model tersebut diuji menggunakan data uji sebanyak satu bagian dan masing-masing pengujian diukur nilai akurasi. Jika *dataset* yang digunakan bersifat tidak seimbang, maka tiap pengujian juga akan mengukur nilai *precision*, *recall* dan *f-measure*. Alur pengujian sebagai berikut:



Gambar 3.1 Tahapan Pengujian Penelitian

3.4.2 Menetapkan Format Data Pengujian

Hasil pengujian klasifikasi dengan tabel *confusion matrix* pada tiap bagian data akan digambarkan dalam tabel 3.1

Tabel 3.1 Rancangan Tabel *Confusion Matrix* untuk Setiap Hasil Pengujian

Cross Validation	Terprediksi						
	Label	Marah	Jijik	Takut	Bahagia	Sedih	Kaget
	Marah						
	Jijik						
	Takut						
	Bahagia						
	Sedih						
	Kaget						
	Netral						

Hasil pengujian klasifikasi ekspresi wajah menggunakan metode *Principal Component Analysis* dan *Convolutional Neural Network* akan digambarkan dalam tabel 3.2.

Tabel 3.2 Rancangan Tabel Hasil Pengujian Klasifikasi Ekspresi Wajah

Cross Validation	Label	Precision	Recall	F-Measure	Akurasi
1	Marah				
	Jijik				
	Takut				
	Bahagia				
	Sedih				
	Kaget				

Cross Validation	Label	Precision	Recall	F-Measure	Akurasi
2	Marah				
	Jijik				
	Takut				
	Bahagia				
	Sedih				
	Kaget				

Cross Validation	Label	Precision	Recall	F-Measure	Akurasi
n	Marah				
	Jijik				
	Takut				
	Bahagia				
	Sedih				
	Kaget				

3.4.3 Menentukan Alat yang Digunakan dalam Pelaksanaan Penelitian

Untuk melaksanakan penelitian mengenai pengklasifikasi ekspresi wajah secara *real time* menggunakan metode *Principal Component Analysis* dan *Convolutional Neural Network* dibutuhkan alat penelitian. Oleh karena itu, penulis akan mengembangkan sebuah perangkat lunak yang dapat melakukan proses ekstraksi ciri menggunakan metode *Principal Component Analysis* dan *Convolutional Neural Network* serta dapat melakukan proses klasifikasi pada ciri yang didapat dengan baik.

Tabel 3.3 Perangkat Keras yang digunakan

No.	Jenis Perangkat Keras	Spesifikasi Hardware
1	Processor	Intel Core i5-4200U, up to 2.3 GHz
2	RAM	4 GB
3	Harddisk	1 TB

Tabel 3.4 Perangkat Lunak yang digunakan

No.	Jenis Perangkat lunak	Nama perangkat lunak
1	Sistem Operasi	Windows 10
2	Bahasa Pemrograman	Python

3.4.4 Melakukan Pengujian dan Analisis

Pada penelitian ini yang dianalisa berupa akurasi atau ketepatan sistem dalam mengklasifikasi data. Maka dari itu contoh penggunaan *confusion matrix* untuk menguji ketepatan hasil pengklasifikasian. *Confusion matrix* membandingkan hasil perhitungan manual dengan hasil yang diproses oleh sistem.

Tabel 3.5 *Confusion Matrix*

Actual Class		Predicted Class	
		C1	C2
	C1	True Positive	False Negative
	C2	False Positive	True Negatives

$$\text{Sensitivitas} = \frac{TP}{P} * 100\% \quad (2.25)$$

$$\text{Spesifisitas} = \frac{TN}{N} * 100\% \quad (2.26)$$

$$\text{Akurasi} = \frac{TP+TN}{(TP+FP+TN+FN)} * 100\% \quad (2.27)$$

3.4.5 Melakukan Analisa Hasil Pengujian dan Membuat Kesimpulan

Hasil akurasi klasifikasi ekspresi wajah dengan ekstraksi ciri menggunakan metode *Principal Component Analysis* dan *Convolutional Neural Network*, didapatkan dengan analisa hasil pengujian yang dapat dilihat pada tabel 3.2. Jika dataset yang digunakan bersifat tidak seimbang, maka perlu juga melakukan perbandingan pada nilai *precision*, *recall* dan *f-measure*. Setelah hasil analisa pengujian penelitian didapatkan, maka selanjutnya adalah membuat kesimpulan penelitian yang akan dijelaskan pada bab V.

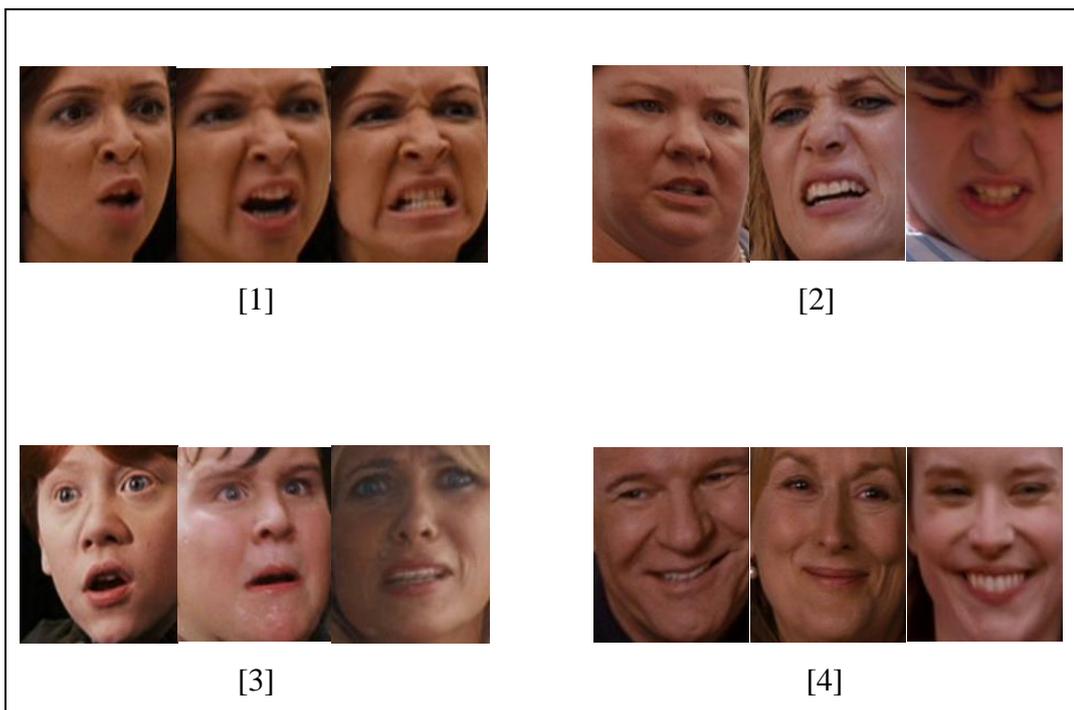
BAB IV

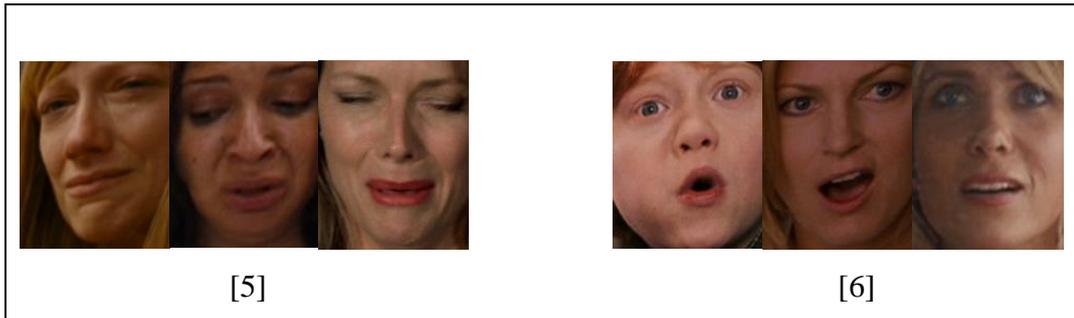
HASIL DAN ANALISIS

Pengklasifikasian ekspresi wajah dilakukan dengan menghitung tingkat akurasi dan waktu pemrosesan sistem. Model klasifikasi hasil pembelajaran dan hasil pengujian menggunakan metode *Principal Component Analysis* (PCA) dan *Convolutional Neural Network* (CNN). Hasil klasifikasi akan dibandingkan dengan klasifikasi ekspresi wajah yang hanya menggunakan metode CNN.

4.1 Dataset

Citra yang digunakan pada penelitian ini berasal dari *dataset Static Facial Expression in The Wild* (SFEW) 2.0 sebanyak 1073 citra dengan 6 ekspresi berbeda, yaitu marah, jijik, takut, bahagia, sedih dan terkejut. Contoh *dataset*-nya ditampilkan pada Gambar 4.1:





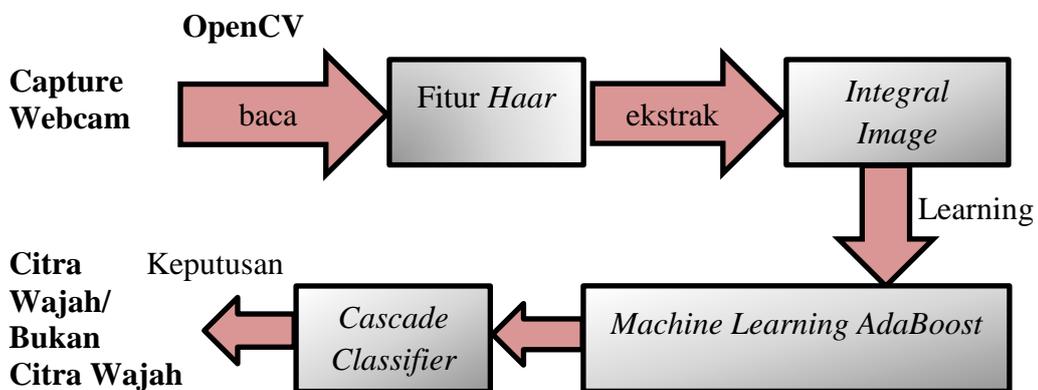
Gambar 4.1 *Sample Data 6 Ekspresi Wajah Sesuai Urutan [1] Marah/Angry [2] Jijik/Disgust [3] Takut/Fear [4] Senang/Happy [5] Sedih/Sad [6] Terkejut/Surprise*

4.2 Deteksi Wajah dengan Metode *Haar-Cascade (Viola-Jones)*

Pada penelitian ini digunakan gabungan dua metode, yaitu metode untuk mendeteksi citra menggunakan metode *Viola-Jones* dan metode untuk pengenalan wajah menggunakan metode *Eigenface*.

4.2.1 Skema Proses Deteksi Wajah

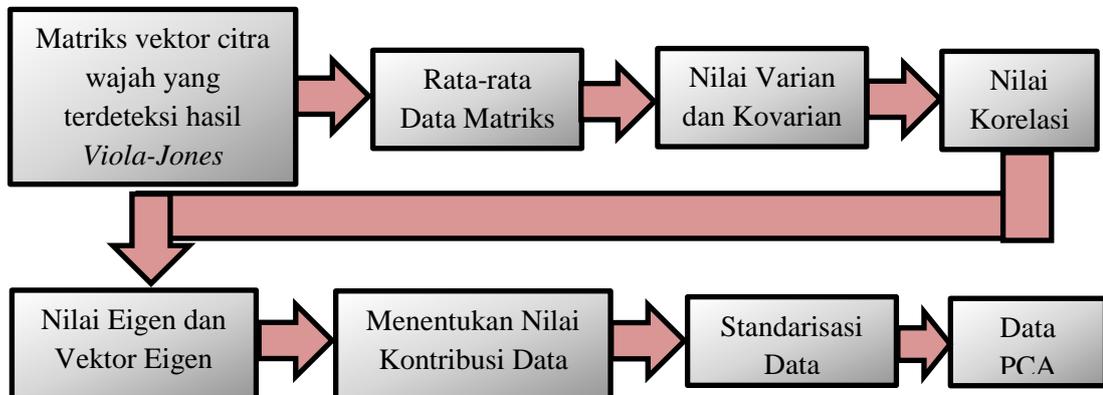
Berikut ini merupakan skema proses deteksi wajah menggunakan metode *Viola-Jones*:



Gambar 4.2 Alur Proses Deteksi Citra Wajah Menggunakan Metode *Viola-Jones*

Hasil deteksi wajah dari metode *Viola-Jones* selanjutnya dijadikan bahan untuk pengenalan wajah pada metode *Eigenface*. Data yang dikumpulkan berupa *dataset* citra wajah yang diekstrak menjadi matriks vektor hasil komputasi.

Analisis data dilakukan melalui serangkaian uji klasifikasi untuk pendeteksian wajah pada metode *Viola-Jones* dan uji korelasi citra wajah data latih.



Gambar 4.3 Alur Proses Pengenalan Citra Wajah menggunakan Metode *Eigenface*



Gambar 4.4 Hasil Deteksi Wajah pada *dataset* Citra SFEW 2.0

4.2.2 Praproses Menjadi Citra *Grayscale*

Pra-pengolahan dari Gambar 4.1 akan diperoleh citra seperti pada Gambar 4.5. Berikut hasil *grayscale sample* citra:

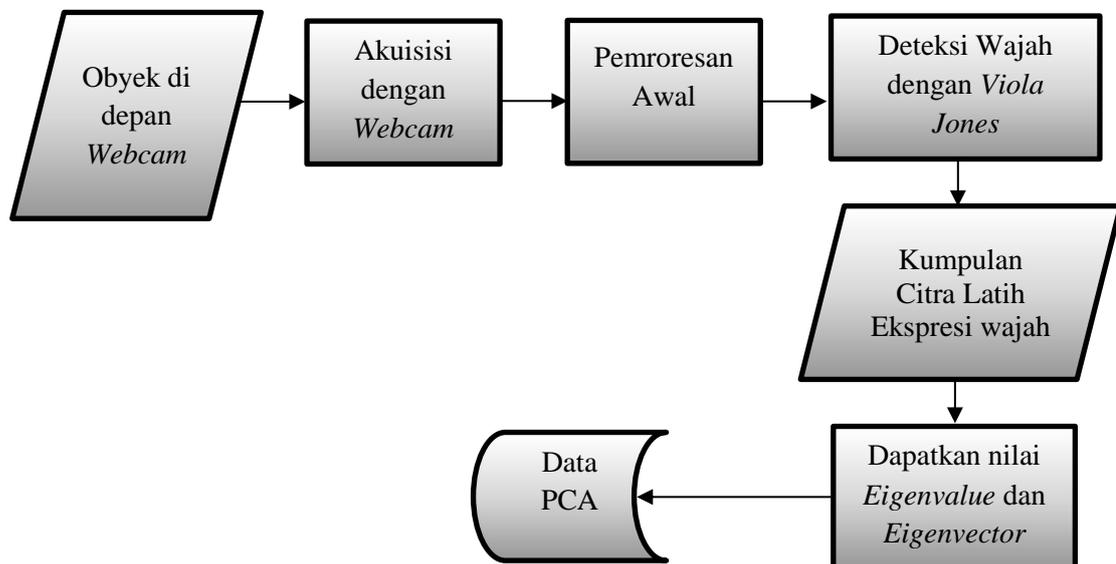




Gambar 4.5 *Sample Data Input 6 Ekspresi Wajah Sesuai Urutan [1] Marah/Angry [2] Jijik/Disgust [3] Takut/Fear [4] Senang/Happy [5] Sedih/Sad [6] Terkejut/Surprise*

4.3 Ekstraksi Ciri Menggunakan PCA

Setelah citra wajah didapatkan pada proses deteksi wajah, citra kemudian melalui proses ekstraksi ciri menggunakan PCA sebagai pengestraksi fitur dengan alur seperti pada Gambar 4.6 berikut:



Gambar 4.6 Alur Ekstraksi Ciri

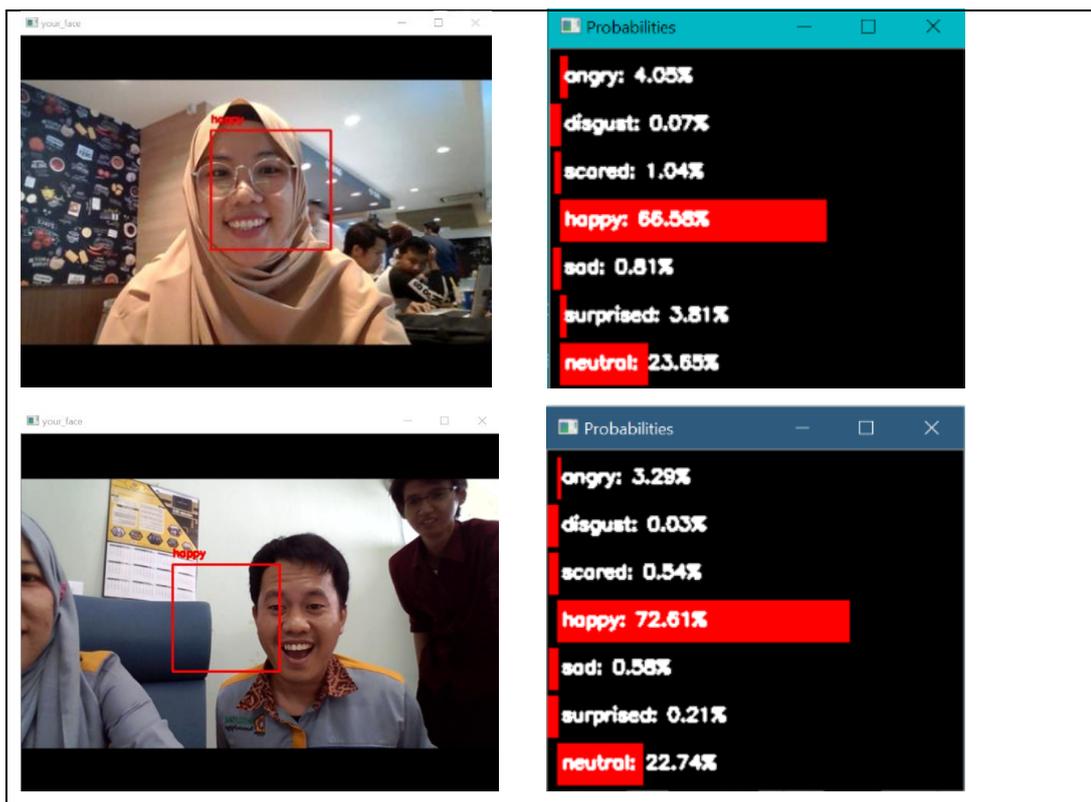
4.4 Hasil

Unjuk kerja dalam penelitian ini diukur melalui *dataset* citra wajah dengan banyaknya pengujian yang dilakukan. Pertama citra yang direkam

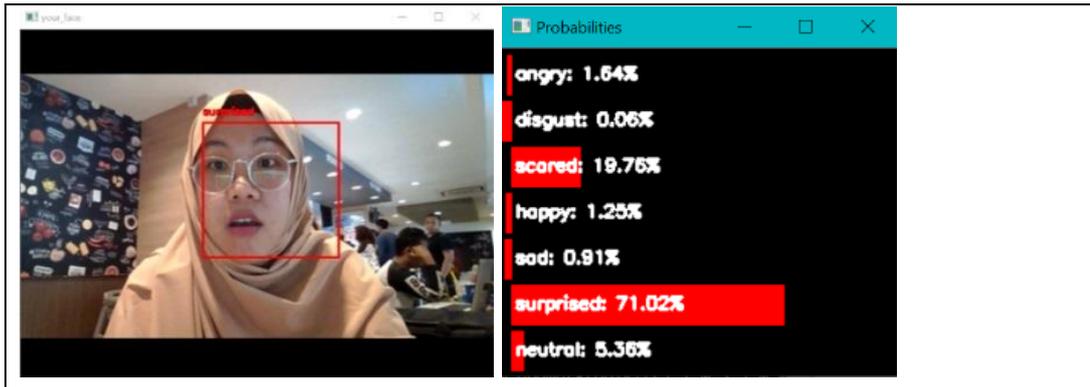
langsung dari antarmuka aplikasi akan dideteksi, proses klasifikasi akan menentukan apakah ekspresi senang atau marah. Akan tetapi terjadi *false detection* pada aplikasi antarmuka dengan menggunakan metode *Haar Cascade*, karena kelemahan menggunakan metode ini membuat objek selain wajah bisa terdeteksi. Akan tetapi karena implementasi penelitian ini untuk *real-time*, metode *haar cascade* sangat efektif digunakan.

4.4.1 Antarmuka Aplikasi Pengklasifikasian Ekspresi Wajah Secara *Real-Time* dengan Metode PCA dan CNN

Tampilan antarmuka aplikasi pengklasifikasian ekspresi wajah secara *real-time* dengan metode PCA dan CNN. Dimana antarmuka ini akan mendeteksi objek wajah secara *real-time* dan akan mengklasifikasikannya seperti gambar berikut:

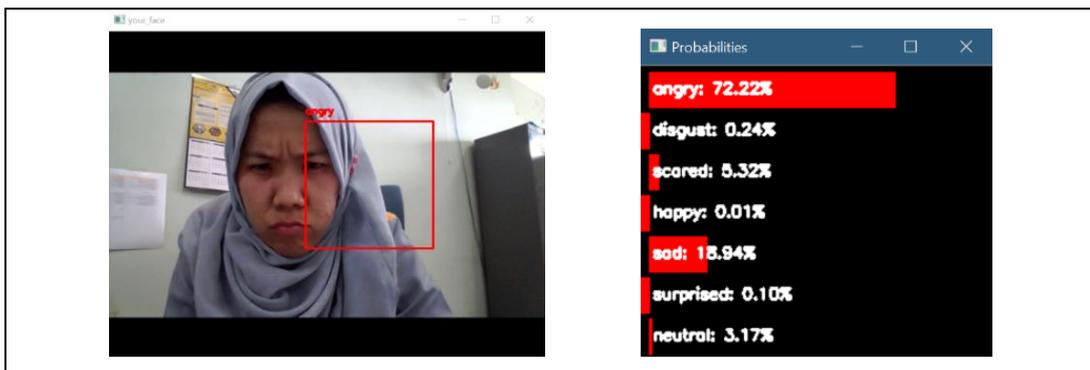


Gambar 4.7 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara *Real Time* yang Menunjukkan Ekspresi Senang / *Happy*

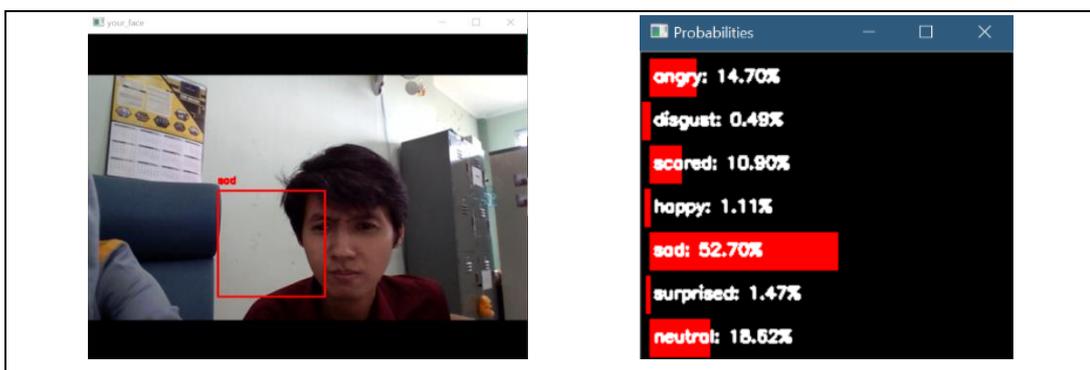


Gambar 4.8 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara *Real Time* yang Menunjukkan Ekspresi Terkejut / *Surprised*

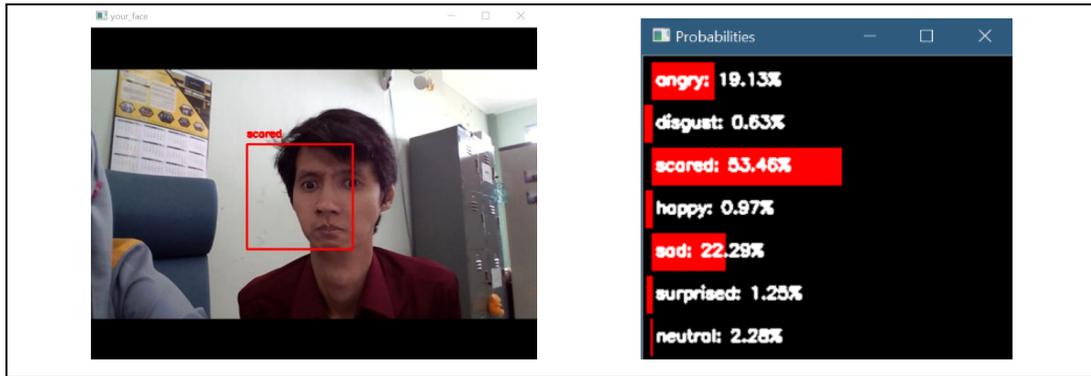
Seperti terlihat pada tabel *probabilities* yang menunjukkan hasil presentase pengklasifikasian ekspresi wajah yang paling tinggi adalah hasil yang tepat.



Gambar 4.9 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara *Real Time* yang Menunjukkan Ekspresi Marah / *Angry*



Gambar 4.10 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara *Real Time* yang Menunjukkan Ekspresi Sedih / *Sad*



Gambar 4.11 Antarmuka Utama Aplikasi Pengklasifikasian Ekspresi Wajah Secara *Real Time* yang Menunjukkan Ekspresi Sedih / *Sad*

4.5 Pengujian Aplikasi

Sebelum aplikasi digunakan untuk proses pengujian penulis melakukan pengujian terhadap aplikasi terlebih dahulu. Pengujian menggunakan metode kotak hitam (*Black-Box Testing*) untuk memastikan apakah fungsi, masukan dan keluaran dari aplikasi sesuai dengan spesifikasi yang dibutuhkan. Berikut tabel pengujian aplikasi yang meliputi proses awal menjalankan aplikasi, memilih pelatihan untuk *training* dan identifikasi untuk pengklasifikasian ekspresi wajah secara *real time* menggunakan metode PCA dan CNN.

Tabel 4.1 Pengujian Menjalankan Aplikasi Utama

No.	Metode	Prosedur	Hasil	Hasil Pengujian
1	Membuka aplikasi Pengklasifikasian Ekspresi Wajah secara <i>Real Time</i> Menggunakan Metode PCA dan CNN	Pengguna mengarahkan wajah ke kamera laptop, dengan memasang berbagai ekspresi wajah	Aplikasi akan mendeteksi objek wajah dan mengklasifikasikannya secara <i>real time</i> , dengan menampilkan hasil deteksi ekspresi wajah pada kotak <i>probabilities</i>	Berhasil. Aplikasi menampilkan <i>Main Activity</i>

4.6 Eksperimen Pengklasifikasian Ekspresi Wajah Secara *Real-Time*

Untuk mengetahui performansi aplikasi yang telah dirancang, maka perlu dilakukan pengujian terhadap aplikasi yang telah dikembangkan. Dalam pengujian akan diukur seberapa besar tingkat keberhasilan aplikasi yang dirancang dalam melakukan analisis pengklasifikasian ekspresi wajah secara *real-time*.

4.6.1 Data Pengujian

Pada penelitian ini dilakukan pengujian dan perhitungan akurasi dengan menggunakan *confusion matrix* seperti berikut:

Class Statistics:

Classes	Angry	Disgust	Fear \
Population	193	193	193
P: Condition positive	25	23	21
N: Condition negative	168	170	172
Test outcome positive	27	0	13
Test outcome negative	166	193	180
TP: True Positive	18	0	12
TN: True Negative	159	170	171
FP: False Positive	9	0	1
FN: False Negative	7	23	9
TPR: (Sensitivity, hit rate, recall)	0.72	0	0.571429
TNR=SPC: (Specificity)	0.946429	1	0.994186
PPV: Pos Pred Value (Precision)	0.666667	NaN	0.923077
NPV: Neg Pred Value	0.957831	0.880829	0.95
FPR: False-out	0.0535714	0	0.00581395
FDR: False Discovery Rate	0.333333	NaN	0.0769231
FNR: Miss Rate	0.28	1	0.428571
ACC: Accuracy	0.917098	0.880829	0.948187
F1 score	0.692308	0	0.705882
MCC: Matthews correlation coefficient	0.645123	NaN	0.702727
Informedness	0.666429	0	0.565615
Markedness	0.624498	NaN	0.873077
Prevalence	0.129534	0.119171	0.108808
LR+: Positive likelihood ratio	13.44	NaN	98.2857
LR-: Negative likelihood ratio	0.295849	1	0.431078
DOR: Diagnostic odds ratio	45.4286	NaN	228
FOR: False omission rate	0.0421687	0.119171	0.05

Gambar 4.12 Statistik *Confusion Matrix* dari Ekspresi Marah, Jijik dan Takut

Pada Gambar 4.12 terdapat hasil statistik *confusion matrix* dari ekspresi marah, jijik dan takut. Dimana *precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. pada ekspresi takut *precision* bernilai paling besar yaitu 92%, sedangkan pada ekspresi marah hanya mendapatkan nilai 66%. Akan tetapi, ekspresi jijik mendapatkan nilai NaN atau *Not a Number*. Ini dikarenakan ekspresi jijik mendapatkan nilai tipe data numerik yang mewakili nilai yang tidak ditentukan atau tidak terwakili, mengingat dari *dataset* yang dipakai, ekspresi jijik hampir sama dengan ekspresi marah.

Nilai *recall* yaitu tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Pada ekspresi marah mendapatkan nilai 72%, lebih besar dari ekspresi takut yang hanya mendapatkan nilai 57%. Sedangkan ekspresi jijik bernilai 0%, ini menandakan bahwa ekspresi jijik sama sekali tidak berhasil dalam menemukan kembali sebuah informasi pada dataset maupun citra input.

F1 *Score* atau F *measure* merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan. Pada ekspresi takut mendapatkan nilai terbesar yaitu 70% dibandingkan ekspresi marah yang hanya sedikit lebih kecil yaitu 69%. Sedangkan pada ekspresi jijik hanya mendapatkan nilai 0%, dalam hal ini sudah jelas bahwa ekspresi jijik tidak mendapatkan nilai perbandingan rata-rata *precision* dan *recall* yang mana pada hasil sebelumnya ekspresi jijik sama sekali tidak mendapatkan nilai.

Selanjutnya nilai akurasi yang didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. Pada ekspresi takut mendapatkan nilai terbesar yaitu 94%, sedangkan pada ekspresi marah sedikit lebih kecil yaitu 91%. Akan tetapi pada ekspresi jijik bernilai 88%. Hal ini menunjukkan bahwa, target dari nilai prediksi tidak tercapai pada ekspresi jijik, dimana nilai aktual tidak dapat mencapai nilai prediksi.

Selanjutnya ekspresi senang dan sedih pada Gambar 4.13, disini ekspresi netral tidak kami gunakan dikarenakan sebagai perbandingan dengan penelitian sebelumnya yang hanya menggunakan 6 ekspresi kecuali ekspresi netral.

Nilai *precision* pada ekspresi senang yaitu 87% lebih besar dari nilai ekspresi sedih yang hanya mendapatkan nilai 49%. Ini menandakan tingkat

ketepatan antara input citra dengan dataset pada ekspresi senang. Nilai *recall* pada ekspresi senang yaitu 87% lebih besar dibandingkan ekspresi sedih yang hanya mendapatkan nilai 9%, ini menandakan bahwa ekspresi senang berhasil dalam menemukan kembali sebuah informasi.

Classes	Happy	Neutral	Sad
Population	193	193	193
P: Condition positive	49	38	30
N: Condition negative	144	155	163
Test outcome positive	49	45	55
Test outcome negative	144	148	138
TP: True Positive	43	32	27
TN: True Negative	138	142	135
FP: False Positive	6	13	28
FN: False Negative	6	6	3
TPR: (Sensitivity, hit rate, recall)	0.877551	0.842105	0.9
TNR=SPC: (Specificity)	0.958333	0.916129	0.828221
PPV: Pos Pred Value (Precision)	0.877551	0.711111	0.490909
NPV: Neg Pred Value	0.958333	0.959459	0.978261
FPR: False-out	0.0416667	0.083871	0.171779
FDR: False Discovery Rate	0.122449	0.288889	0.509091
FNR: Miss Rate	0.122449	0.157895	0.1
ACC: Accuracy	0.937824	0.901554	0.839378
F1 score	0.877551	0.771084	0.635294
MCC: Matthews correlation coefficient	0.835884	0.713057	0.584516
Informedness	0.835884	0.758234	0.728221
Markedness	0.835884	0.670571	0.46917
Prevalence	0.253886	0.196891	0.15544
LR+: Positive likelihood ratio	21.0612	10.0405	5.23929
LR-: Negative likelihood ratio	0.127773	0.17235	0.120741
DOR: Diagnostic odds ratio	164.833	58.2564	43.3929
FOR: False omission rate	0.0416667	0.0405405	0.0217391

Gambar 4.13 Statistik *Confusion Matrix* dari Ekspresi Senang dan Sedih

F1 *Score* atau *F measure* pada ekspresi senang bernilai 87%, sedangkan ekspresi sedih hanya bernilai 63% jauh lebih kecil dibanding ekspresi senang. Hal ini jelas membuktikan bahwa nilai *F measure* pada ekspresi senang, mencapai nilai rata-rata perbandingan nilai *recall* dan presisi yang dibobotkan.

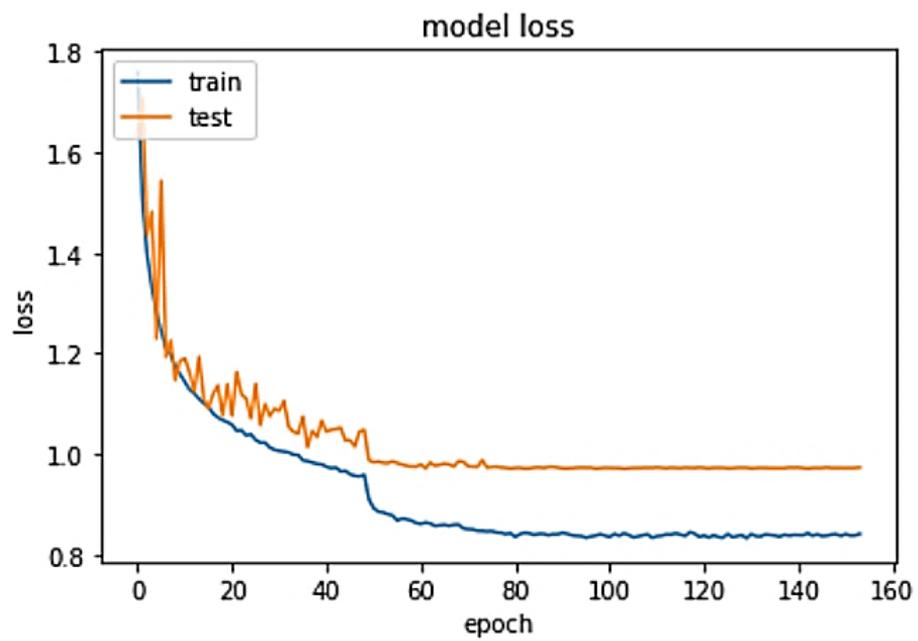
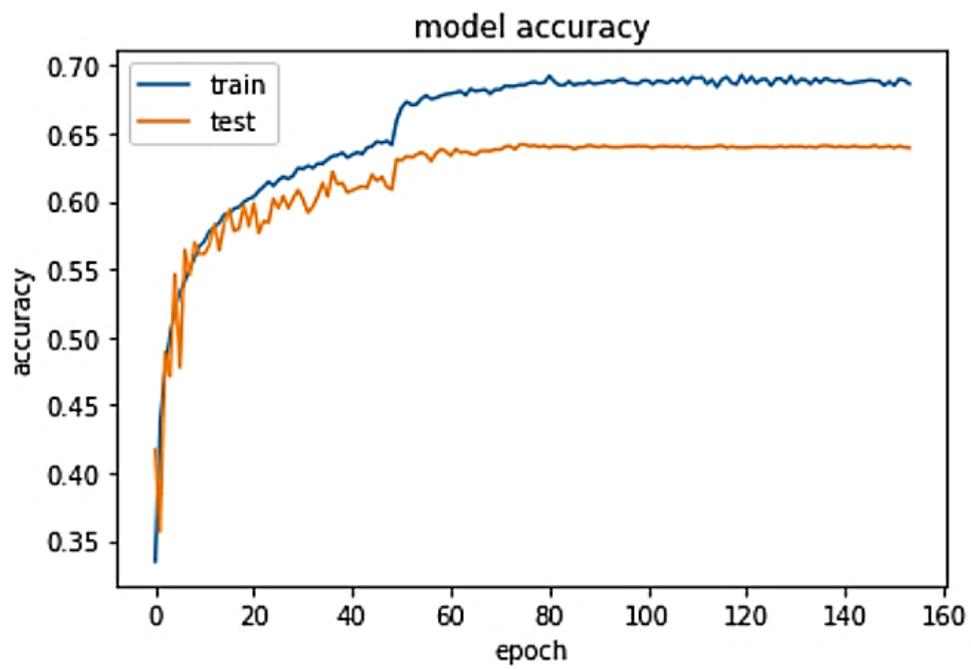
Nilai akurasi pada ekspresi senang mencapai 93%, sedangkan pada ekspresi sedih bernilai 83%. Hal ini menunjukkan bahwa tingkat kedekatan antara nilai prediksi dengan nilai aktual tercapai.

Classes	Surprise
Population	193
P: Condition positive	7
N: Condition negative	186
Test outcome positive	4
Test outcome negative	189
TP: True Positive	4
TN: True Negative	186
FP: False Positive	0
FN: False Negative	3
TPR: (Sensitivity, hit rate, recall)	0.571429
TNR=SPC: (Specificity)	1
PPV: Pos Pred Value (Precision)	1
NPV: Neg Pred Value	0.984127
FPR: False-out	0
FDR: False Discovery Rate	0
FNR: Miss Rate	0.428571
ACC: Accuracy	0.984456
F1 score	0.727273
MCC: Matthews correlation coefficient	0.749906
Informedness	0.571429
Markedness	0.984127
Prevalence	0.0362694
LR+: Positive likelihood ratio	inf
LR-: Negative likelihood ratio	0.428571
DOR: Diagnostic odds ratio	inf
FOR: False omission rate	0.015873

Gambar 4.14 Statistik *Confusion Matrix* dari Ekspresi Terkejut

Selanjutnya nilai *precision* pada ekspresi terkejut hanya mendapatkan nilai 1%, hal ini membuktikan bahwa tingkat ketepatan antara input citra dengan dataset tidak tercapai. Selanjutnya nilai *recall* pada ekspresi terkejut yaitu 57%, ini termasuk nilai rata-rata untuk menemukan kembali sebuah informasi.

Untuk nilai *F1 score* atau *F measure* pada ekspresi terkejut mencapai 72%, ini bermakna bahwa nilai perbandingan antara nilai *recall* dengan presisi yang dibobotkan. Dan nilai akurasi pada ekspresi terkejut adalah 98%, dimana nilai akurasi tertinggi dibandingkan dengan ekspresi lainnya.



Gambar 4.15 Model Akurasi dan Model *Loss* pada *Dataset* SFEW 2.0

Pada Gambar 4.15 menunjukkan grafik peningkatan akurasi pada proses *training* dan *validation/test dataset* SFEW 2.0 dengan tingkat akurasi sampai 75% pada *epoch* ke 160. Proses penurunan *loss* pada proses *training* dan *validation/test dataset* SFEW 2.0 ditunjukkan pada grafik model *loss*, dimana pada setiap *epoch* *loss* mengalami penurunan sampai *epoch* ke 160 didapatkan nilai *loss* (*error*) terendah yaitu 0.8 yang ditunjukkan pada pergerakan grafik yang menurun. Ini membuktikan bahwa dengan metode yang dipakai pada penelitian ini, dapat mengurangi tingkat *loss* pada data *training*.

4.6.2 Hasil Pengujian

Setelah beberapa kali dilakukan pengujian dan eksperimen pada aplikasi pengklasifikasian. Hasil berupa data-data, grafik dan persentase kemudian dicatat dan dibandingkan dengan penelitian yang hanya menggunakan metode CNN, dengan hasil yang dapat dilihat di Gambar 4.16 sebagai berikut:

Confusion Matrix:

Predicted Actual	Angry	Disgust	Fear	Happy	Sadness	Surprise
Angry	0.72	0	0	0.04	0.16	0
Disgust	0.13	0	0	0.174	0.565	0
Fear	0.143	0	0.571	0	0.286	0
Happy	0	0	0	0.878	0.041	0
Sadness	0	0	0	0.033	0.9	0
Surprise	0	0	0	0	0.143	0.571

Gambar 4.16 *Confusion Matrix* Ekspresi Wajah dengan *Dataset* SFEW 2.0

Confusion Matrix:

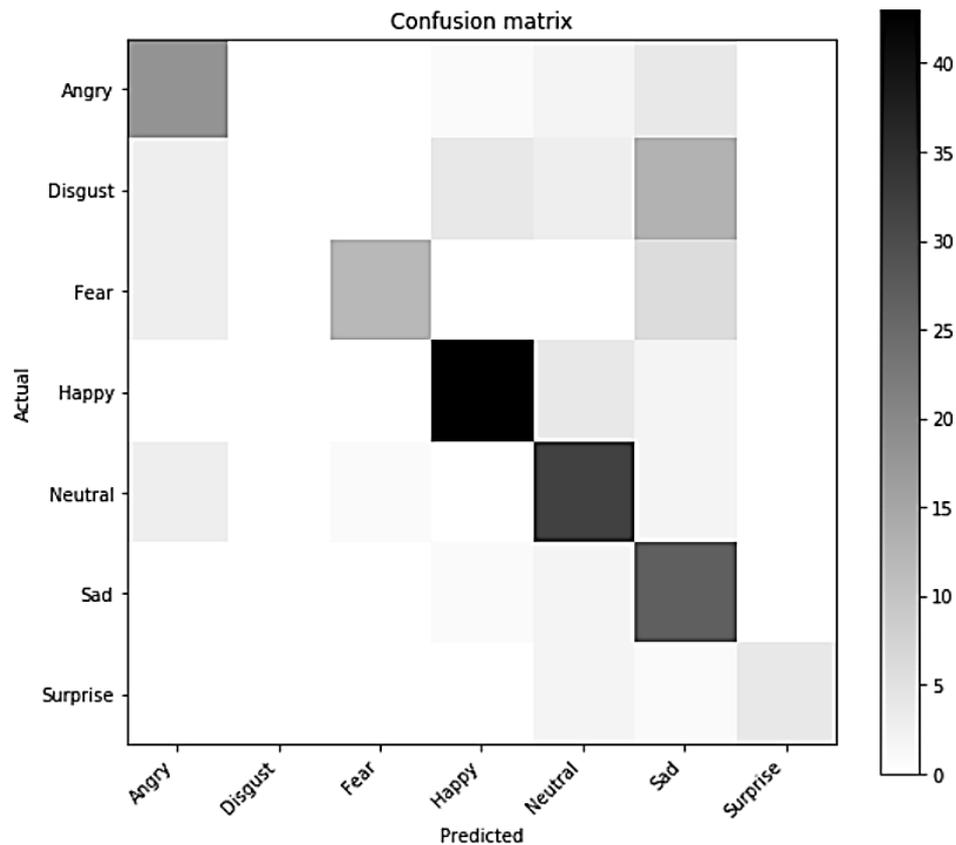
Predicted Actual	Angry	Disgust	Fear	Happy	Sadness	Surprise
Angry	0.559	0.202	0.166	0	0.009	0.006
Disgust	0.137	0.505	0.087	0.02	0.111	0.139
Fear	0.055	0.106	0.595	0.016	0.184	0.004
Happy	0.002	0.16	0.01	0.852	0.004	0.012
Sadness	0.146	0.142	0.115	0	0.572	0.025
Surprise	0.019	0.121	0.156	0.013	0.036	0.655

Gambar 4.17 *Confusion Matrix* Ekspresi Wajah dengan *Dataset LFEW*

(Y. Liu et al., 2018)

Cross Validation	Label	Precision	Recall	F-Measure	Akurasi
1	Marah	0.667	0.72	0.692	0.917
	Jijik	NaN	0	0	0.881
	Takut	0.923	0.571	0.706	0.948
	Bahagia	0.878	0.878	0.878	0.938
	Sedih	0.491	0.9	0.635	0.839
	Kaget	1	0.571	0.727	0.984

Tabel 4.2 Hasil Pengujian Klasifikasi Ekspresi Wajah



Gambar 4.18 Grafik *Confusion Matrix* Pengklasifikasi Ekspresi Wajah

Gambar 4.16 menunjukkan perhitungan akurasi *confusion matrix* pada 6 ekspresi wajah SFEW 2.0 dataset, dimana ekspresi disgust diperoleh akurasi 0, dikarenakan ekspresi yang ditimbulkan dan variasi dalam ekspresi. Dan akurasi yang tertinggi didapat pada ekspresi happy yaitu 0,878, ditunjukkan pada Gambar 4.17 ditunjukkan dengan kotak berwarna paling hitam.

Berikut laporan keseluruhan statistik dari *confusion matrix* pada Gambar 4.16:

Overall Statistics:

Accuracy: 0.7046632124352331

95% CI: (0.6349048278770365, 0.7680120062650024)

No Information Rate: ToDo

P-Value [Acc > NIR]: 1.9627594725872617e-33

Kappa: 0.639453329837441

McNemar's Test P-Value: ToDo

Gambar 4.18 Statistik Keseluruhan dari *Confusion Matrix* Pengklasifikasi Ekspresi Wajah dengan Metode PCA dan CNN

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Dalam penelitian ini, didapatkan hasil bahwa cara memadukan dua metode antara *Principal Component Analysis* (PCA) dan *Convolutional Neural Network* (CNN) dengan memfungsikan PCA sebagai pengekstraksi fitur, sedangkan CNN sebagai pengklasifikasi ekspresi wajah. Jadi dua metode ini dapat diterapkan untuk klasifikasi ekspresi wajah secara *real-time*.
2. Implementasi metode PCA dan CNN dalam klasifikasi ekspresi wajah dilakukan secara bertahap sesuai dengan diagram blok pada Bab IV. Sedangkan performansi perpaduan *Principal Component Analysis* (PCA) dan *Convolutional Neural Network* (CNN) menghasilkan akurasi yang lebih tinggi 9,5% dari penelitian sebelumnya.
3. PCA berfungsi sebagai *feature extraction* dan *feature selection*, yang dapat meningkatkan performa CNN untuk mengklasifikasi ekspresi wajah dibanding hanya menggunakan CNN saja.

5.2 Saran

Saran pengembangan penelitian selanjutnya agar ditambahkan beberapa metode praproses citra, dan lebih fokus pada algoritma deteksi wajah yang akan mengambil bagian penting pada citra. Pada penelitian ini metode deteksi wajah menggunakan *haar-cascade* yang bagus untuk aplikasi *real-time* tetapi kurang mumpuni untuk fokus pada deteksi citra wajah.

Pengklasifikasi ekspresi marah dan jijik hampir sulit dibedakan karena *dataset* yang digunakan menunjukkan ekspresi citra yang hampir sama. Karena performa metode *haar-cascade* sebagai pendeteksi wajah masih kurang dalam penelitian ini dikarenakan metode ini dapat mendeteksi apapun selain citra wajah, sehingga perlu untuk menggunakan metode deteksi wajah yang lainnya.

DAFTAR PUSTAKA

- Abhirawan, H., Jondri, & Arifianto, A. (2017). Pengenalan Wajah Menggunakan Convolutional Neural Networks (CNN). *Universitas Telkom*, 4(3), 4907–4916.
- Abidin, Z. (2011). Pengembangan Sistem Pengenalan Ekspresi Wajah menggunakan Jaringan Syaraf Tiruan Backpropagation (Studi Kasus pada Database MUG). *Jurnal Matematika Murni Dan Terapan*, 5(1), 21–30.
- Alexander, S. (2013). Program Aplikasi Pengenalan Ekspresi Wajah Secara Real Time Dengan Metode Back Propagation Dan Wavelet Haar.
- Çankçı, M. üg., & Özen, F. (2012). A Face Recognition System Based on Eigenfaces Method. *Procedia Technology*, 1, 118–123. <https://doi.org/10.1016/j.protcy.2012.02.023>
- David, T., & Hons, H. (2005). Face Recognition : Two-Dimensional and Three-Dimensional Techniques, (September).
- Fitriyah, N. (2014). Analisis dan simulasi sistem pengenalan wajah dengan metode fisherface berbasis outdoorvideo.
- Ian, G., Yoshua, B., & Courville, C. (2014). *Deep Learning. Linear Algebra*. <https://doi.org/10.1016/B978-0-12-391420-0.09987-X>
- Imanuel, T. (2018). Pendeteksian dan Pengenalan Wajah Manusia untuk Peningkatan Kinerja Kamera Pengaman.
- Liu, Y., Yuan, X., Gong, X., Xie, Z., Fang, F., & Luo, Z. (2018). Conditional convolution neural network enhanced random forest for facial expression recognition. *Pattern Recognition*, 84, 251–261. <https://doi.org/10.1016/j.patcog.2018.07.016>
- Putra, I. N. T. A., Dwidasmara, I. B. G., & Astawa, I. G. S. (2014). Perancangan Dan Pengembangan Sistem Absensi Realtime Melalui Metode Pengenalan

Wajah. *Jurnal Sains Dan Teknologi*, 3(2), 450.

Santoso, A., Ariyanto, G., Learning, F., Network, C. N., & Pendahuluan, I. (n.d.). IMPLEMENTASI DEEP LEARNING BERBASIS KERAS UNTUK, 18(01), 15–21.

Syafira, A. R., & Viola, P. (n.d.). Sistem Deteksi Wajah Dengan Modifikasi Metode Viola Jones. *Jurnal Emitor*, 17(01).

Wahyudi, E., Kusuma, H., & Wirawan. (2011). Perbandingan Unjuk Kerja Pengenalan Wajah Berbasis Fitur Local Binary Pattern dengan Algoritma PCA dan Chi Square, 3–8.

Wiyono, A. R. (2018). MATH unesa COMPONENT ANALYSIS (PCA) DAN EXTREME LEARNING MACHINE (ELM), 6(2), 2–6.

Zhou, C., Wang, L., Zhang, Q., & Wei, X. (2013). Face recognition based on PCA image reconstruction and LDA. *Optik*, 124(22), 5599–5603. <https://doi.org/10.1016/j.ijleo.2013.04.108>

Zufar, M. & B. S. (2016). Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time. *Jurnal Sains Dan Seni*, 5(3), 1–6. <https://doi.org/10.1177/0892705705044556>