

**PENERAPAN METODE GLCM (*GRAY LEVEL CO-
OCCURRENCE MATRIX*) UNTUK REDUKSI CIRI
PADA PENGENALAN EKSPRESI WAJAH**

Oleh
NURAINI AHMAD
T3116159

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana



PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2020

PENGESAHAN SKRIPSI

PENERAPAN METODE GLCM (*GRAY LEVEL CO-OCCURRENCE MATRIX*) UNTUK REDUKSI CIRI PADA PENGENALAN EKSPRESI WAJAH

Oleh
NURAINI AHMAD
T3116159

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana
dan telah disetujui oleh Tim pembimbing pada tanggal
Gorontalo, 15 Juli 2020

Pembimbing I



Haditsah Annur, M.Kom

NIDN. 0908058403

Pembimbing II



Rofiq Harun, M.Kom

NIDN. 0919048404

PERSETUJUAN SKRIPSI

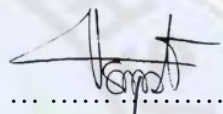
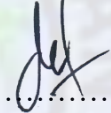
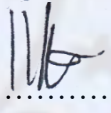
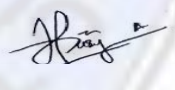
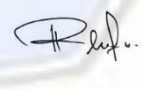
PENERAPAN METODE GLCM (*GRAY LEVEL CO-OCCURRENCE MATRIX*) UNTUK REDUKSI CIRI PADA PENGENALAN EKSPRESI WAJAH

Oleh

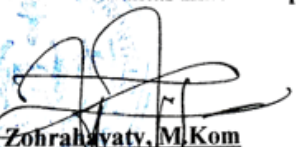
NURAINI AHMAD

T3116159

Diperiksa oleh Panitia Ujian Strata satu (S1)
Universitas Icshan Gorontalo
Gorontalo 06 Agustus 2020

- | | |
|---|--|
| 1. Ketua Penguji
Asmaul Husna N, M.Kom | 
..... |
| 2. Anggota
Sudirman S.Panna, M.Kom | 
..... |
| 3. Anggota
M. Effendi Lasulika, M.Kom | 
..... |
| 4. Anggota
Haditsah Annur, M.Kom | 
..... |
| 5. Anggota
Rofiq Harun, M.kom | 
..... |

Mengetahui :


Dekan Fakultas Ilmu Komputer
Zohrahayaty, M.Kom
NIDN. 0912117702


Ketua Program Studi
Irvan Abraham Salihi, M.Kom
NIDN. 0928028101

□

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa :

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasi orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar, yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma - norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 15 Juli 2020

Nuraini Ahmad

ABSTRACT

Face expression is one of the type of communication nonverbal and it can show the emotion feeling from someone to someone who observe it. Face expression is one of the important way to express social communication in human life. Problem which often happen is interpret someone's expression, it usually hard to hide the feeling or emotion from the face although the are a lot of people would like to do it. Therefore, it has been made the application of face expression by computer and can detect someone's face expression until it can be categorized as the represent of emotion, with perpetuate in an image. Recognition of face expression is doing by use Gray Level Co-Occurrence Matrix Method as a feature extraction to face to get values of parameter that can be the input of classification. Furthermore, in classication, it has been used Algoritma K-nearest Neighbor to recognize face expression. From 32 data of image has been classified 24 data is right, and obtain the 75% accurate.

Kata Kunci : *Face expression, Gray Level Co-occurrence Matrix, K-nearest neighbor, Classification.*

ABSTRAK

Ekspresi wajah merupakan salah bentuk komunikasi nonverbal, dan dapat menyampaikan keadaan emosi dari seseorang kepada orang yang mengamatinya. Ekspresi wajah merupakan salah satu cara penting dalam menyampaikan pesan sosial dalam kehidupan manusia. Masalah yang sering terjadi yaitu salah mengartikan ekspresi seseorang, biasanya amat sulit untuk menyembunyikan perasaan atau emosi tertentu dari wajah walaupun banyak orang yang merasa ingin melakukannya. Oleh karena itu di buat aplikasi pendeteksi ekspresi wajah dengan bantuan komputer dan mendeteksi ekspresi wajah seseorang sehingga didapatkan kategori emosi yang mewakilinya, dengan cara mengabadikannya dalam sebuah citra. Pengenalan Ekspresi Wajah dilakukan dengan menggunakan Metode *Gray Level Co-occurrence Matrix* sebagai Fitur Ekstraksi pada Wajah untuk mendapatkan nilai-nilai parameter yang akan dijadikan inputan pada klasifikasi. Selanjutnya pada tahap klasifikasi digunakan Algoritma *K-nearest neighbor* untuk mengenali ekspresi wajah. Dari 32 Citra Data Uji diklasifikasi 24 data yang benar, dan mendapatkan hasil akurasi sebanyak 75%.

Kata Kunci : Pengenalan Ekspresi Wajah, *Gray Level Co-occurrence Matrix*, *K-nearest neighbor*, Klasifikasi.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT sehingganya penulis dapat menyelesaikan skripsi ini dengan judul: “***Penerapan Metode GLCM (Gray Level Co-Occurrence Matriks) Untuk Reduksi Ciri Pada Pengenalan Ekspresi Wajah***” sebagai salah satu syarat Ujian Akhir guna memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa Usulan Penelitian ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terimakasih dan penghargaan yang setinggi-tingginya kepada:

1. Bapak Mohammad Ichsan Gaffar, S.E, M.Ak, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Ibu Zohrahayati, S.Kom., M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Panna, M.Kom, selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Ibu Irma Surya Kumala Idris, M.Kom, selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Sudirman Melangi, M.Kom, selaku Wakil Dekan III Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Bapak Irvan Abraham Salihi, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
8. Ibu Haditsah Annur, M.Kom, selaku Pembimbing I yang telah banyak membantu penyusunan skripsi ini;

9. Bapak Rofiq Harun, M.Kom, selaku Pembimbing II yang telah banyak membantu penyusunan skripsi ini;
10. Bapak dan ibu dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
11. Kedua orang tua saya yang tercinta (Sukri Ahmad dan Farida Akuna), atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
12. Faruq Natsir Djafar untuk segala dukungan dan usaha membantu penulis dalam pencarian judul dan pencarian data, serta kesabaran dan semangat yang selalu diberikan dari awal sampai skripsi ini sampai selesai;
13. Felia Muliawan dan Saraswati Rojikan untuk segala motivasi agar penulis segera menyelesaikan skripsi ini, selalu memberi semangat pada penulis;
14. Siti Nurkholiza, Yolanda, Serliani, Nurlaila, Sarintan, Fatma, Wahyu, Dana, Sandy, dan Nuriman yang selalu membantu, mendukung, menemani penulis dengan sabar;
15. Kepada semua pihak yang ikut membantu dalam penyelesaian skripsi ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah, SWT melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, 15 July 2020

Penulis

DAFTAR ISI

PENGESAHAN SKRIPSI	ii
PERSETUJUAN SKRIPSI.....	iii
PERNYATAAN SKRIPSI.....	iv
ABSTRACT	v
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah	3
1.3 Rumusan Masalah	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II LANDASAN TEORI	5
2.1 Tinjauan Studi	5
2.2 Tinjauan Pustaka	7
2.2.1. Ekspresi Wajah.....	7
2.2.2. Citra.....	8
2.2.3. Pengolahan Citra	9
2.2.4. GLCM (Gray Level Co-occurrence Matrix).	10
2.2.5. Ekstraksi Tektstur.....	12
2.2.6. KNN (K-Nearest Neighbor).....	13
2.2.7. Confussion Matrix.....	18

2.2	Kerangka Pikir.....	20
BAB III METODE PENELITIAN		21
3.1.	Jenis, Metode, Subjek, Objek, dan Lokasi Penelitian	21
3.1.1.	Jenis.....	21
3.2.	Pengumpulan Data.....	21
3.3.	Pemodelan	22
3.3.1.	Pra Pengolahan Data	22
3.3.2.	Ekstraksi Fitur GLCM.....	23
3.3.3.	Klasifikasi KNN	23
3.3.4.	Evaluasi Model.....	23
BAB IV HASIL PENELITIAN.....		24
4.1.	Hasil Pengumpulan Data	24
4.2.	Hasil Pemodelan.....	31
4.2.1.	Pra Pengolahan Data	31
4.2.2.	Ekstraksi Ciri Gray Level Co-occurrence Matrix	34
4.2.4	Klasifikasi	42
4.3.	Confusion Matrix.....	47
BAB V PEMBAHASAN		51
5.1	Pembahasan Model.....	51
BAB VI PENUTUP		54
6.1.	Kesimpulan.....	54
6.2.	Saran.....	54
DAFTAR PUSTAKA.....		55
Lampiran 1: KODE PROGRAM		57
Lampiran 2 : DATA.....		69
Lampiran 3: RIWAYAT HIDUP		79
JADWAL PENELITIAN.....		81

DAFTAR GAMBAR

Gambar 2. 1 Konversi RGB ke Gray Scale.....	12
Gambar 2. 2 Contoh Klasifikasi pada K-Nearest Neighbor.....	13
Gambar 2. 3 Ilustrasi penggunaan nilai k pada metode KNN.....	14
Gambar 3. 1 Pemodelan	22
Gambar 4. 1 Ekspresi wajah Senyum	26
Gambar 4. 2 Ekspresi Wajah Marah	27
Gambar 4. 3 Ekspresi Wajah Sedih.....	29
Gambar 4. 4 Ekspresi Wajah Takut	31
Gambar 4. 5 Data Testing Dalam Perhitungan GLCM.....	34

DAFTAR TABEL

Tabel 2. 1 State of the art	5
Tabel 2. 2 Klasifikasi Data Training	15
Tabel 2. 3 Perhitungan Kuadrat Jarak Data training dengan data testing	16
Tabel 2. 4 Penentuan 3 tetangga Terdekat dari Data testing.....	16
Tabel 2. 5 Klasifikasi Kelas Tetangga Terdekat	17
Tabel 2. 6 Confusion Matrix menampilkan total positive dan negative.	18
Tabel 4.1 Konversi Citra Asli Ke GreyScale	32
Tabel 4.2 Konversi GreyScale ke Hasil Crop	34
Tabel 4.4 Matriks Normalisasi.....	36
Tabel 4.5 Data Testing	42
Tabel 4.6 Klasifikasi Data Training	44
Tabel 4.7 Perhitungan Euclidean Distance	45
Tabel 4.8 Hasil pengurutan dari nilai terkecil.....	46
Tabel 4.9 Jumlah K=7	46
Tabel 4.10 Kelas Data Uji.....	46
Tabel 4.11 Confusion Matrix (Data Testing).....	50
Tabel 4.12 Tabel Hasil Data Uji	50
Tabel 5.1 Hasil Pemodelan	52

DAFTAR LAMPIRAN

Lampiran 1: KODE PROGRAM	57
Lampiran 2 : DATA	69
Lampiran 3: RIWAYAT HIDUP	79

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ekspresi wajah atau mimik adalah hasil dari salah satu atau lebih dari satu gerakan otot pada wajah. Ekspresi wajah merupakan salah bentuk komunikasi nonverbal, dan dapat menyampaikan keadaan emosi dari seseorang kepada orang yang mengamatinya. Ekspresi wajah merupakan salah satu cara penting dalam menyampaikan pesan sosial dalam kehidupan manusia[1]. Pada sebuah studi didapatkan bahwa ekspresi wajah dapat menyampaikan hingga 55% informasi dalam interaksi, dibandingkan dengan vokal dan verbal keseluruhan hanya dapat memberi informasi hingga sebesar 45%. Dalam jurnal Claudia dkk ekspresi wajah dapat digunakan untuk mengembangkan interaksi yang natural antara manusia dan komputer, menurut A. Mehrabian. [2].

Masalah yang sering terjadi yaitu salah mengartikan ekspresi seseorang, biasanya amat sulit untuk menyembunyikan perasaan atau emosi tertentu dari wajah walaupun banyak orang yang merasa ingin melakukannya.

Membuat aplikasi pendeteksi ekspresi wajah dengan bantuan komputer dan mendeteksi ekspresi wajah seseorang sehingga didapatkan kategori emosi yang mewakilinya, dengan cara mengabadikannya dalam sebuah citra. Menurut Donato dkk, (1999) “Ekspresi wajah seseorang juga dapat mencerminkan kondisi afektif, kegiatan kognitif, tujuan, *personality* dan *phscyopatholgy* dari seseorang”[3].

Dari penelitian sebelumnya yaitu deteksi ekspresi wajah menggunakan fitur gabor dan haar wavelet hasilnya banyak diprediksi benar adalah ekspresi senang dengan tingkat akurasi pada setiap mayoritas fitur sebanyak 100%. Sedangkan untuk ekspresi yang banyak diprediksi salah adalah ekspresi merendahkan dan sedih yang paling banyak diprediksi sebagai ekspresi marah. Dilihat dari jumlah data, ekspresi senang memiliki data yang cukup banyak, 69 citra atau 21% dari keseluruhan data. Sedangkan untuk ekspresi merendahkan memiliki jumlah data

yang relatif sedikit dibandingkan dengan ekspresi lainnya. Data ekspresi merendahkan hanya 5,8% dari total keseluruhan data yaitu 18 citra[2].

Metode GLCM menurut Xie dkk (2010) merupakan suatu metode yang melakukan analisis terhadap suatu piksel pada citra dan mengetahui tingkat keabuan yang sering terjadi. Menurut Pullaperuma & Dharmaratne(2013) Metode ini juga untuk tabulasi tentang frekuensi kombinasi nilai piksel yang muncul pada suatu citra untuk melakukan analisis citra berdasarkan distribusi statistik dari intensitas pikselnya, dapat dilakukan dengan mengekstrak fitur teksturnya[4], Oleh karena itu pada penelitian ini, penulis akan menggunakan metode dalam pengolahan citra digital yaitu *GLCM (Gray Level Co-occurrence Matrix)* karena metode ini adalah metode yang paling sering digunakan untuk analisis tekstur yang didasarkan pada ciri statistik citra dimana dalam perhitungan statistiknya menggunakan distribusi derajat keabuan (histogram) dengan mengukur tingkat kontras, granularitas, dan kekasaran suatu daerah dari hubungan ketetanggaan antar piksel didalam citra.

KNN merupakan salah satu teknik klasifikasi yang paling dasar dan sederhana. Metode ini memiliki kelebihan yakni pelatihan yang sangat cepat, sederhana dan mudah dipelajari, tahan terhadap data yang memiliki derau dan efektif jika data pelatihan besar. Terlihat dari beberapa penelitian yang pernah dilakukan tentang pemanfaatan K-Nearest Neighbor pada pengenalan ekspresi wajah dengan praproses transformasi wavelet. Data yang digunakan berupa gambar atau citra yang didapat dari beberapa objek. Pada percobaan ini terdapat 80 citra dari 10 objek. Sebanyak 64 citra digunakan sebagai database pelatihan dan 16 digunakan untuk uji pengenalan pola. Hasil pemilihan k terbaik adalah 5 dengan jumlah citra pada setiap kelas dalam kelompok pengujian sebanyak 8 buah[5].

Berdasarkan penjelasan tersebut maka akan dikembangkan suatu aplikasi yang dapat mendeteksi ekspresi wajah untuk menunjukkan emosional saat seseorang sedang dalam keadaan Senang, Marah, Sedih dan Takut. Adapun masalah yang akan diangkat adalah Bagaimana membangun aplikasi pengenalan ekspresi wajah dengan bantuan computer, bagaimana mengetahui kondisi emosional seseorang berdasarkan ekspresi wajah, apakah dapat mengenali ekspresi wajah

dengan menggunakan metode GLCM (*Gray Level Co-occurrence Matrix*) dan Apakah dapat mengklasifikasikan setiap ekspresi wajah dengan menggunakan K-NN (*K-Nearest Neighbor*).

Oleh karena itu penulis ingin mengajukan sebuah penelitian yang berjudul **“Penerapan Metode GLCM (*Gray Level Co-Occurence Matriks*) Untuk Reduksi Ciri Pada Pengenalan Ekspresi Wajah”**. Tujuan penelitian agar dapat mengetahui kondisi emosional seseorang berdasarkan ekspresi wajah.

1.2 Identifikasi Masalah

Dalam uraian latar belakang di atas, maka dapat diidentifikasi beberapa permasalahan sebagai berikut :

1. Tidak dapat mengetahui kondisi emosional senang, marah, sedih ataupun takut seseorang berdasarkan ekspresi.
2. Belum adanya pengenalan ekspresi wajah dengan menggunakan metode GLCM (*Gray Level Co-occurrence Matrix*).
3. Belum dapat mengklasifikasikan setiap ekspresi wajah dengan menggunakan metode KNN (*K-Nearest Neighbor*).

1.3 Rumusan Masalah

Adapun rumusan masalah yang dapat ditarik dari identifikasi masalah, yaitu :

1. Bagaimana hasil uji coba metode GLCM (*Gray Level Co-occurrence Matrix*) pada pengenalan ekspresi wajah?
2. Bagaimana Akurasi metode KNN (*K-Nearest Neighbor*) pada pengenalan ekspresi wajah?

1.4 Tujuan Penelitian

1. Dapat mengenali hasil uji coba metode GLCM (*Gray Level Co-occurrence Matrix*) pada pengenalan ekspresi wajah.
2. Dapat mengklasifikasikan ekspresi wajah dengan menggunakan metode KNN (*K-Nearest Neighbor*).

1.5 Manfaat Penelitian

1. Manfaat Teoritis Sebagai tambahan referensi dalam penelitian analisis Ekspresi Wajah dan dapat juga digunakan sebagai pijakan lanjutan untuk aplikasinya.
2. Manfaat Praktis sebagai tambahan referensi untuk peneliti selanjutnya dalam pengambilan keputusan untuk menganalisis Ekspresi Wajah.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Tabel 2. 1 State of the art

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
1	Claudia Primasiwi, Handayani Tjandrasa, Dini A. Navastara	Deteksi Ekspresi Wajah Menggunakan Fitur Gabor dan Haar Wavelet	2018	Fitur Gabor dan Haar Wavelet	Pada penelitian ini, digunakan kombinasi fitur Gabor dan Haar untuk dibandingkan dan digabungkan dengan fitur Landmark yang telah disediakan di database : * Gabor dan Haar memiliki akurasi sebesar 92%. * Gabor dan Landmark memberikan akurasi tertinggi sebesar 94,8%, dengan kelas terbaik berupa ekspresi senang -
2	Subairi, Rahmadwati, Erni Yudaningtyas	Implementasi Metode k-Nearest Neighbor pada Pengenalan Pola Tekstur Citra Saliva untuk Deteksi Ovulasi	2018	GLCM (<i>Gray Level Co-occurrence Matrix</i>) & KNN (<i>K-Nearest Neighbor</i>)	Data masukan berupa lima parameter dari fitur tekstur GLCM (gray-level co-occurrence matrices), yang meliputi ASM (Angular Second Moment), contrast, IDM (Inverse Different Moment), entropy, dan correlation, yang diambil

					dari rata-rata dari masing-masing sudut 0°, 45°, 90°, dan 135°. Sedangkan data keluaran adalah salah satu dari tiga klasifikasi dari masa ovulasi, yaitu infertile, intermediate, dan fertile dengan akurasi 93.3%
3	Sufiatul Maryana, Lita Karlitasari, Arie Qur'anias	Pemanfaatan K-Nearest Neighbor (KNN) Pada Pengenalan Wajah Dengan Praproses Transformasi Wavelet	2012	Praproses Transformasi Wavelet & Pemanfaatan K-Nearest Neighbor (KNN)	Data yang digunakan berupa gambar atau citra yang diperoleh dari beberapa objek. Pada percobaan ini terdapat 80 citra dari 10 objek. Sebanyak 64 citra digunakan sebagai database pelatihan dan 16 digunakan untuk uji pengenalan pola. Metode yang digunakan adalah transformasi wavelet untuk ekstraksi fitur dan K-NN untuk identifikasi. Hasil pemilihan k terbaik adalah 5 dengan jumlah citra pada setiap kelas dalam kelompok pengujian sebanyak 8 buah.

2.2 Tinjauan Pustaka

2.2.1. Ekspresi Wajah

Ekspresi wajah atau mimik adalah hasil dari salah satu atau lebih dari satu gerakan otot pada wajah. Ekspresi wajah merupakan salah bentuk komunikasi nonverbal, dan dapat menyampaikan keadaan emosi dari seseorang kepada orang yang mengamatinya. Ekspresi wajah merupakan salah satu cara penting dalam menyampaikan pesan sosial dalam kehidupan manusia[1]. Pada sebuah studi didapatkan bahwa ekspresi wajah dapat menyampaikan hingga 55% informasi dalam interaksi, dibandingkan dengan vokal dan verbal keseluruhan hanya dapat memberi informasi hingga sebesar 45%. Ekspresi wajah dapat digunakan untuk mengembangkan interaksi yang natural antara manusia dan komputer[2].

Ekspresi wajah dapat dimiliki oleh manusia baik yang disengaja maupun yang tidak sengaja, tapi umumnya ekspresi wajah dialami secara tidak sengaja akibat perasaan atau emosi manusia tersebut. Biasanya amat sulit untuk menyembunyikan perasaan atau emosi tertentu dari wajah, walaupun banyak orang yang merasa amat ingin melakukannya. Misalnya, orang yang mencoba menyembunyikan perasaan bencinya terhadap seseorang tanpa sengaja akan menunjukkan perasaannya tersebut di wajahnya pada saat tertentu, walaupun ia berusaha menunjukkan ekspresi netral. Hubungan perasaan dan ekspresi wajah juga dapat berjalan sebaliknya. Pengamatan menunjukkan bahwa melakukan ekspresi wajah tertentu dengan sengaja (Misalnya: Tersenyum) dapat mempengaruhi atau menyebabkan perasaan terkait benar benar terjadi[1].

Di dalam kehidupan sehari-hari, khususnya dalam komunikasi interpersonal, wajah sering digunakan untuk berekspresi. Melalui ekspresi wajah, maka dapat dipahami emosi yang sedang bergejolak pada diri individu. Ekspresi wajah merupakan salah satu karakteristik perilaku. Ekspresi wajah bisa menyampaikan apa yang ada di pikiran orang, bahwa kita dapat mengenali emosi mereka. Tapi ketika emosi mencapai intensitas tertentu, seluk beluk ekspresi wajah menjadi hilang[6]

2.2.2. Citra

Citra adalah gambar yang terdapat dalam bidang 2 dimensi. Sedangkan jika ditinjau dari sudut pandang matematis, citra merupakan fungsi yang berkesinambungan (*continue*) dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, kemudian objek memantulkan kembali sebagian dari berkas cahaya. Pantulan cahaya tersebut ditangkap oleh alat-alat optic seperti mata manusia, kamera, pemindai/scanner, dan lain-lain sehingga bayangan objek dalam bentuk citra dapat terekam[6].

Citra sebagai out put dari suatu sistem perekaman data dapat bersifat[6]:

1. Optik berupa foto.
2. Analog berupa sinyal video seperti gambar pada monitor televisi.
3. Digital yang dapat langsung disimpan pada media penyimpanan magnetic.

Citra dapat dikelompokkan menjadi dua yaitu citra diam (*still image*) dan citra bergerak (*moving image*). Citra diam yang ditampilkan secara beruntun (*sekuensial*), sehingga memberi kesan pada mata sebagai gambar yang bergerak. Setiap citra didalam rangkaian tersebut disebut *frame*. Gambar-gambar yang tampak pada film layar lebar atau televisi pada hakekatnya terdiri dari ratusan sampai ribuan *frame* [6].

Citra juga dapat dikelompokkan menjadi dua yaitu citra yang tampak seperti foto, gambar, lukisan, apa yang nampak dilayar monitor/televisi, hologram, dan lain sebagainya. Sedangkan citra tidak tampak seperti data foto/gambar dalam file, citra yang direpresentasikan dalam fungsi matematis[6].

Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa Citra digital yang tersusun dari 3 komponen warna yakni R (*red*), G (*green*), B (*blue*) di dalam matlab direpresentasikan dalam bentuk :

R = citra_rgb.....(:, :,1)

G = citra_rgb..... (:, :,2)

B = citra_rgb..... (:, :,3)

Maksudnya adalah bahwa tiap komponen warna terpisah dalam bentuk koordinat spasial (x,y) dan angka ketiga berfungsi sebagai urutan apakah warna

tersebut R, G atau B. Urutan ini sudah standar, jadi jangan sampai tertukar dan membingungkan. Untuk mengubah warna RGB ke skala warna abu-abu atau *grayscale* dapat digunakan rumus standar berikut:

$$\text{gray_R} = .2989 * \text{citra_rgb} \dots\dots\dots (,:,1)$$

$$\text{gray_G} = .5870 * \text{citra_rgb} \dots\dots\dots (,:,2)$$

$$\text{gray_B} = .1140 * \text{citra_rgb} \dots\dots\dots (,:,3)$$

Citra dalam skala abu-abu memiliki variasi warna 0-255, dan berukuran 1x8 bit. Tentu saja berbeda dengan citra biner yang hanya memiliki warna hitam dan putih berarti hanya memiliki komponen warna 0 dan 1 saja.

2.2.3. Pengolahan Citra

Seringkali citra mengalami penurunan mutu (degradasi), misalnya mengandung cacat (*noise*), warnanya terlalu kontras, kurang tajam, kabur, dan sebagainya. Citra seperti ini lebih sulit untuk diinterpretasi karena informasi yang disampaikan oleh citra tersebut berkurang. Agar lebih mudah diinterpretasi, maka citra tersebut perlu dimanipulasi menjadi citra baru yang memiliki kualitas lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra[7].

Ada beberapa operasi pengolahan citra yang biasanya dilakukan menurut Jaim, 1989, diantaranya [7]:

2.2.3.1.Representasi dan pemodelan citra

Bertujuan untuk memberikan gambaran suatu hal. Contohnya adalah bagaimana pencahayaan benda di suatu lokasi dari hasil foto kamera, karakteristik jaringan tubuh dari citra X-Ray, posisi target dari sebuah radar dari citra radar, suhu suatu daerah dari citra inframerah atau medan gravitasi suatu daerah dari citra geofisika.[7]

2.2.3.2.Perbaikan citra

Bertujuan untuk menonjolkan fitur-fitur tertentu yang ada pada citra untuk kepentingan ekstraksi fitur, analisis citra atau menampilkan informasi citra tersebut. Contohnya adalah perbaikan kontras, perbaikan tepi objek, pemberian warna semu, penipisan derau, penajaman, dan sebagainya.[7]

2.2.3.3.Pemugaran citra

Bertujuan untuk menghilangkan atau mengurangi degradasi citra. Mirip dengan perbaikan citra, hanya saja pada pemugaran citra sudah diketahui penyebab degradasi citra. Contohnya adalah menghilangkan kesamaran (*deblurring*) yang disebabkan oleh keterbatasan sensor, usia citra yang sudah tua atau banyaknya derau (*noise*) pada citra. [7]

2.2.3.4.Analisis citra (*image analys*)

Bertujuan untuk menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsi dari citra tersebut. Teknik analisis citra memerlukan fitur-fitur tertentu untuk membantu dalam identifikasi objek. Proses segmentasi terkadang diperlukan untuk memisahkan objek yang diinginkan dari sekelilingnya sehingga selanjutnya dapat dilakukan pengukuran. Contohnya adalah mendeteksi tepi objek, ekstraksi batas (*boundary*), dan representasi daerah (*region*).[7]

2.2.3.5.Rekonstruksi citra

Bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Contohnya adalah gambar 3D yang dihasilkan dari beberapa foto rontgen dengan sinar X. [7]

2.2.3.6.Kompresi citra

Bertujuan untuk mengurangi jumlah bit yang dibutuhkan untuk menyimpan atau mengirim gambar tanpa kehilangan informasi yang diperlukan. Citra berukuran besar akan mempengaruhi penggunaan memori dan lamanya waktu pengiriman. [7]

2.2.4. GLCM (Gray Level Co-occurrence Matrix).

Metode GLCM menurut Xie dkk (2010) merupakan suatu metode yang melakukan analisis terhadap suatu piksel pada citra dan mengetahui tingkat keabuan yang sering terjadi. Metode ini juga untuk tabulasi tentang frekuensi kombinasi nilai piksel yang muncul pada suatu citra untuk melakukan analisis citra berdasarkan distribusi statistik dari intensitas pikselnya, dapat dilakukan dengan mengekstrak fitur teksturnya[4]

Gray Level Co-occurrence Matrix (GLCM) adalah suatu metode yang digunakan untuk analisis tekstur/ekstraksi ciri. GLCM merupakan suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra [8].

Koordinat pasangan piksel memiliki jarak d dan orientasi sudut Θ . Jarak direpresentasikan dalam piksel dan sudut direpresentasikan dalam derajat. Orientasi sudut terbentuk berdasarkan empat arah sudut yaitu, 0° , 45° , 90° dan 135° , dan jarak antar piksel sebesar 1 piksel menurut [8].

Tahapan yang dilakukan pada perhitungan GLCM adalah sebagai berikut:

1. Pembentukan matriks awal GLCM dari pasangan dua piksel yang berjajar sesuai dengan arah 0° , 45° , 90° atau 135° .
2. Membentuk matriks yang simetris dengan menjumlahkan matriks awal GLCM dengan nilai transposnya.
3. Menormalisasi matriks GLCM dengan membagi setiap elemen matriks dengan jumlah pasangan piksel.
4. Ekstraksi ciri, yaitu:
 - a. **Contrast** menunjukkan ukuran penyebaran (momen inersia) elemen-elemen matriks citra. Jika letaknya jauh dari diagonal utama, nilai kekontrasan besar. Secara visual, nilai kekontrasan adalah ukuran variasi antar derajat keabuan suatu daerah citra.

$$Contrast = \sum_i \sum_j (i - j)^2 P_{ij}$$

b. Correlation

$$\frac{\sum_{i=1}^L \sum_{j=1}^L (i - \mu_i)(j - \mu_j)(GLCM(i, j))}{\sqrt{\sigma_i^2 \sigma_j^2}}$$

- c. **Energy** menyatakan distribusi intensitas piksel terhadap jangkauan aras keabuan.

$$\sum_{i,j} P(i,j)^2$$

- d. **Entropy**

$$\sum_{i,j} P_{ij} \log^2 P_{ij}$$



Gambar 2. 1 Konversi RGB ke Gray Scale

Sumber : Achmad Basuki, Nana Ramadijanti

2.2.5. Ekstraksi Tektstur

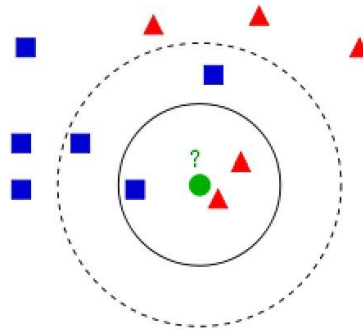
Ciri tekstur merupakan ciri penting dalam sebuah gambar yang merupakan informasi berupa susunan struktur permukaan suatu gambar. Dalam penelitian ini menggunakan *Gray Level oCcurance Matrix* (GLCM) sebagai matrik pengambilan nilai keabuan dari sebuah gambar. Berikut merupakan tahapan yang digunakan dalam pengambilan ciri tekstur dari sebuah gambar.

- 1) Citra warna dirubah menjadi citra grayscale
- 2) Masing-masing nilai dari RGB citra dirubah menjadi abuabu dengan menggunakan rumus sebagai berikut: keabuan = $0.2989 * R + 0.5870 * G + 0.1140 * B$ (1)
- 3) Piksel baru = setPixel(255, nilai keabuan, nilai keabuan, nilai keabuan)
- 4) Segmentasi nilai warna ke dalam 16 bin

- 5) Hitung nilai-nilai co-occurrence matrix dalam empat arah masing-masing 0° , 45° , 90° , dan 135°
- 6) Hitung informasi ciri tekstur yaitu yaitu *contrast*, *correlation*, *energy*, *homogeneity*, dan *entropy* Contrast [9]

2.2.6. KNN (K-Nearest Neighbor)

K-Nearest Neighbor adalah Algoritma *k-nearest neighbor* (K-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Sebuah objek akan diklasifikasikan berdasarkan pilihan terbanyak dari tetangga-tetangganya.



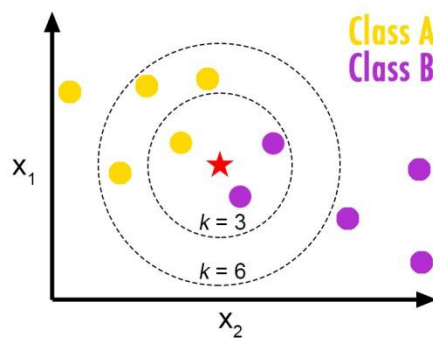
Gambar 2. 2 Contoh Klasifikasi pada K-Nearest Neighbor

Metode ini merupakan salah satu jenis dari pembelajaran berdasarkan perwujudan, di mana fungsi-fungsinya hanyalah merupakan nilai pendekatan secara lokal dan semua perhitungan ditahan hingga proses Klasifikasi. Metode *k-nearest neighbor* merupakan metode yang paling sederhana dari semua metode pembelajaran mesin [11]

Sedangkan menurut peneliti lain K-Nearest Neighbor adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran (*neighbor*) yang jaraknya paling dekat dengan objek tersebut. Dekat atau jauhnya *neighbor* biasanya dihitung berdasarkan jarak Euclidean[12].

Metode K-NN dibagi menjadi dua fase, yaitu pembelajaran (*training*) dan klasifikasi. Pada fase pembelajaran, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi dari data pembelajaran. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk data yang akan diuji coba (yang klasifikasinya tidak diketahui).

Jarak dari vektor yang baru ini terhadap seluruh vektor data pembelajaran dihitung, dan sejumlah k buah *neighbor* yang paling dekat diambil. Sebuah titik akan diprediksi jenisnya berdasarkan pada klasifikasi terbanyak dari *neighbor* di sekitarnya (Gambar 3).



Gambar 2. 3 Ilustrasi penggunaan nilai k pada metode KNN

[12]

Nilai k yang terbaik untuk KNN tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus di mana klasifikasi diprediksikan berdasarkan data pembelajaran yang paling dekat (dengan kata lain, $k = 1$) disebut algoritma *nearest neighbor*.

K-nearest neighbor classification (KNN) merupakan suatu algoritma pembelajaran berbasis contoh yang telah terbukti sangat efektif untuk berbagai domain masalah (Cover & Hart, 1968). Tujuan dari k-nearest

neighbor classification adalah untuk menemukan knearest neighbor classification untuk contoh yang diberikan, kemudian memberikan label kelas untuk contoh yang diberikan sesuai dengan kelas mayoritas knearest neighbor classification. Algoritma ini mengilustrasikan bahwa seluruh contoh sesuai dengan titik-titik pada ruang ndimensi. Unsur kunci dari skema ini ialah ketersediaan ukuran kesamaan yang bisa mengidentifikasi data terdekat (tetangga). Tetangga terdekat dari sebuah contoh didefinisikan dalam hal fungsi jarak seperti jarak Euclidean standar. Lebih tepatnya, misalkan x digambarkan dengan fitur vektor $\langle a_1(x), a_2(x), \dots, a_r(x) \rangle$, dimana $a_i(x)$ merupakan nilai ke i atribut x .

Kemudian jarak antara dua x_i dan x_j didefinisikan sebagai $dist(x_i, x_j)$, dimana:

$$dist(x_i, x_j) = \sqrt{\sum_{q=1}^r (a_q(x_i) - a_q(x_j))^2}$$

[5]

Contoh perhitungan K-Nearest Neighbor.

(sumber: Heru Ajisantosa, "Deteksi Ekspresi Wajah untuk pengguna E-Learning Menggunakan K-Nearest Neighbor Dengan Gray Level Co-occurrence Matrix," April 2019.)

Terdiri dari 2 atribut dengan skala kuantitatif sebagai data training yaitu X_1 dan X_2 serta Y yaitu kelas baik dan buruk seperti pada Tabel 2.2

Tabel 2. 2 Klasifikasi Data Training

X_1	X_2	Y
7	7	Buruk
7	4	Buruk
3	4	Baik
1	4	Baik

Terdapat data testing yaitu $X_1 = 3$ dan $X_2 = 7$, tentukan nilai Y ! Langkah penyelesaian:

1. Tentukan parameter K = jumlah tetangga terdekat, misalkan ditetapkan $K = 3$
2. Hitung jarak antara data baru dengan semua data training seperti pada tabel 2.3.

Tabel 2. 3 Perhitungan Kuadrat Jarak Data training dengan data testing

X1	X2	Kuadrat Jarak dengan Data Baru
7	7	$(7-3)^2 + (7-7)^2 = 16$
7	4	$(7-3)^2 + (4-7)^2 = 25$
3	4	$(3-3)^2 + (4-7)^2 = 9$
1	4	$(1-3)^2 + (4-7)^2 = 13$

3. Urutkan hasil kuadrat jarak tersebut secara ascending dan tetapkan tetangga terdekat nilai K (dimana $K = 3$) seperti pada tabel 2.4 dan tabel 2.5.

Tabel 2. 4 Penentuan 3 tetangga Terdekat dari Data testing

X1	X2	Euclidean	Pertingkat Jarak	Tetangga Terdekat
7	7	$\sqrt{16} = 4$	3	YA
7	4	$\sqrt{25} = 5$	4	TIDAK
3	4	$\sqrt{9} = 3$	1	YA
1	4	$\sqrt{13} = 3,6$	2	YA

Tabel 2. 5 Klasifikasi Kelas Tetangga Terdekat

X 1	X 2	Euclidean Distance	Peringkat Jarak	Tetangga terdekat	Y
7	7	4	3	YA	BURUK
7	4	5	4	TIDAK	-
3	4	3	1	YA	BAIK
1	4	3,6	2	YA	BAIK

4. Berdasarkan hasil dari tabel 5 dimana nilai $K = 3$, mayoritas hasil Y yang diperoleh yaitu “Baik” Jadi hasil klasifikasi data testing dengan nilai $X_1 = 3$ dan $X_2 = 7$ adalah $Y =$ kelas Bai

2.2.7. Confussion Matrix

Confusion matrix menurut dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah classifier tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari TruePositive dan True-Negative memberikan informasi ketika classifier dalam melakukan klasifikasi data bernilai benar, sedangkan FalsePositive dan False-Negative memberikan informasi ketika classifier salah dalam melakukan klasifikasi data.

		Predicted class	
Actual class		Yes	No
Total	Yes	$TP (a=a)$	$FN(b=a)$
P	No	$FP(a=b)$	$TN(b=b)$
N	Total	P'	N'
P + N			

Tabel 2. 6 Confusion Matrix menampilkan total positive dan negative.

(Han dan Kamber (2011))

Nilai akurasi menggambarkan seberapa akurat system dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data Berdasarkan nilai True Negative, True Positive, False Negatve dan False Positive dapat diperoleh nilai akurasi, presisi dan recall.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

$$Presisi = \frac{TP}{TP+FP} * 100\%$$

$$Recall = \frac{TP}{FN + TP} * 100\%$$

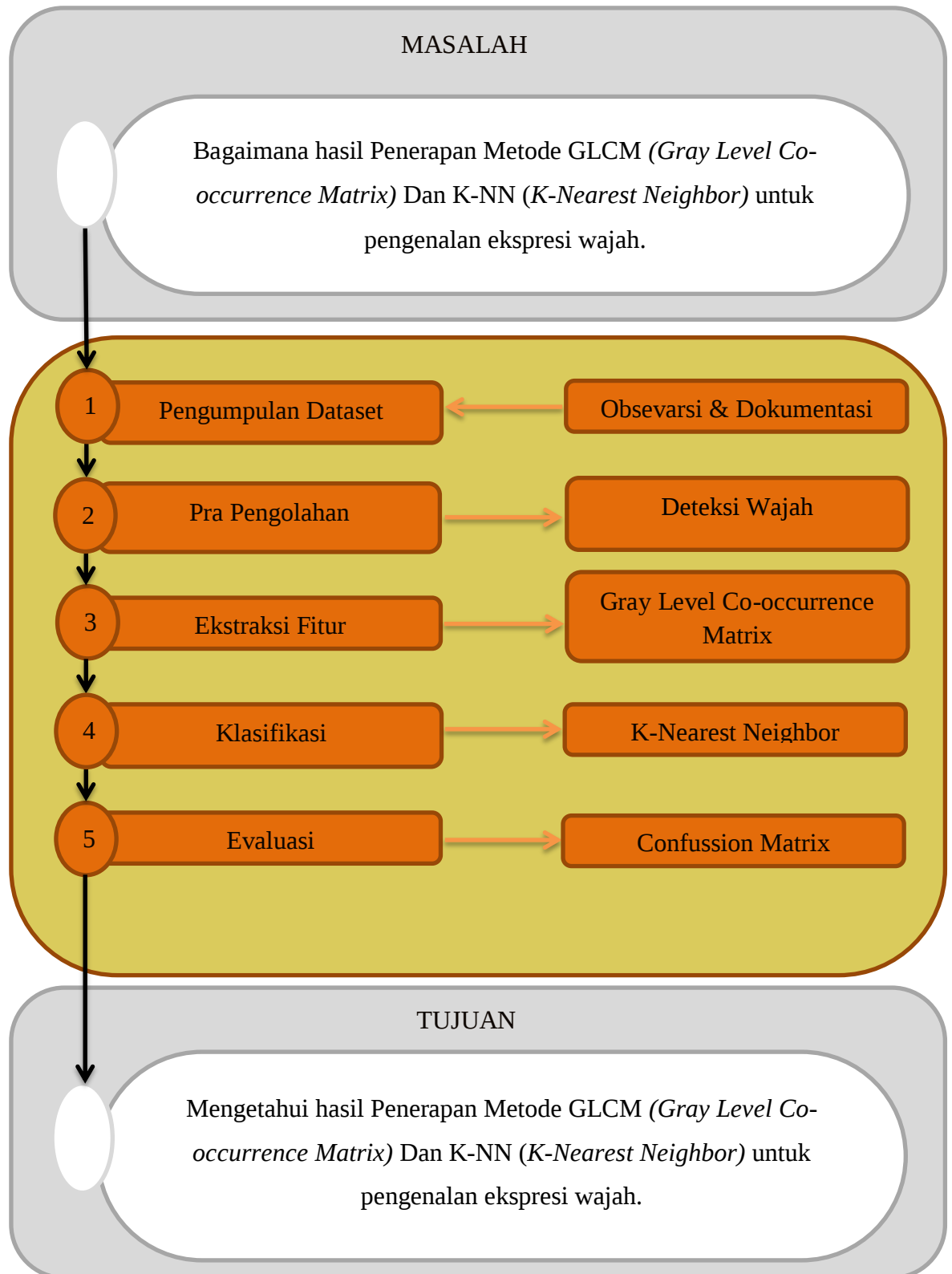
TP (True Positive) → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif

FP (False Positive) → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif

FN (False Negative) → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi negative

TN (True Negative) → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negative [13].

2.3 Kerangka Pikir



BAB III

METODE PENELITIAN

3.1. Jenis, Metode, Subjek, Objek, dan Lokasi Penelitian

3.1.1. Jenis

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan.

3.1.2. Metode

Penelitian ini menggunakan metode penelitian tindakan. Dengan demikian jenis penelitian ini adalah penelitian korelasional.

3.1.3. Subjek

Penelitian ini menggunakan metode penelitian *Eksperimental*. Subjek penelitian ini adalah *Klasifikasi* untuk mendapatkan hasil akhir.

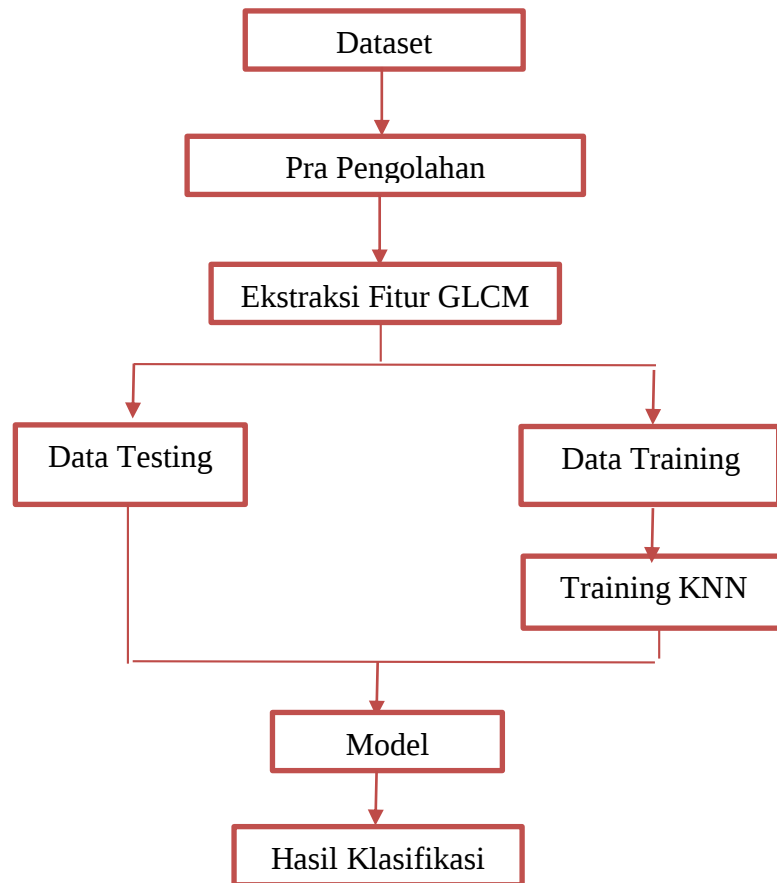
3.1.4. Objek

Objek yang digunakan adalah *Ekspresi Wajah Manusia*.

3.2. Pengumpulan Data

Dataset pada penelitian ini adalah dataset public dengan nama *Collection of Facial Images* Yang diunduh dari halaman <https://cswwww.essex.ac.uk/mv/allfaces/faces94.html>

3.3.Pemodelan



Gambar 3. 1Pemodelan

3.3.1.Pra Pengolahan Data

Sebelum data diolah, terlebih dahulu dilakukan *Histogram Equalization*. *Histogram Equalization* adalah suatu proses untuk meratakan histogram agar derajat keabuan dari yang paling rendah (0) sampai dengan yang paling tinggi (255) mempunyai kemunculan yang rata, Jadi dengan *Histogram Equalization* hasil gambar yang memiliki histogram yang tidak merata akan menjadi gambar yang lebih jelas karena derajat keabuannya tidak dominan gelap atau dominan terang.

3.3.2.Ekstraksi Fitur GLCM

Proses ekstraksi ini dilakukan untuk mengetahui ciri tekstur pada sebuah citra ekspresi wajah yang kemudian citra warna dirubah menjadi citra grayscale dan selanjutnya akan diklasifikasikan untuk hasil akhirnya.

3.3.3.Klasifikasi KNN

Proses klasifikasi dilakukan untuk mengklasifikasikan setiap hasil akhir dari citra ekspresi wajah, akan menampilkan beberapa hasil akhir berupa ekspresi senyum, marah, sedih dan takut dari setiap orang. Kemudian akan dilihat tingkat keakurasi setiap ekspresi.

3.3.4.Evaluasi Model

Model yang dilakukan kemudian dievaluasi dengan menggunakan Confussion Matrix untuk mengetahui tingkat Akurasi

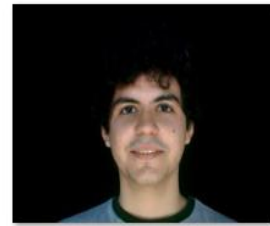
BAB IV

HASIL PENELITIAN

4.1. Hasil Pengumpulan Data

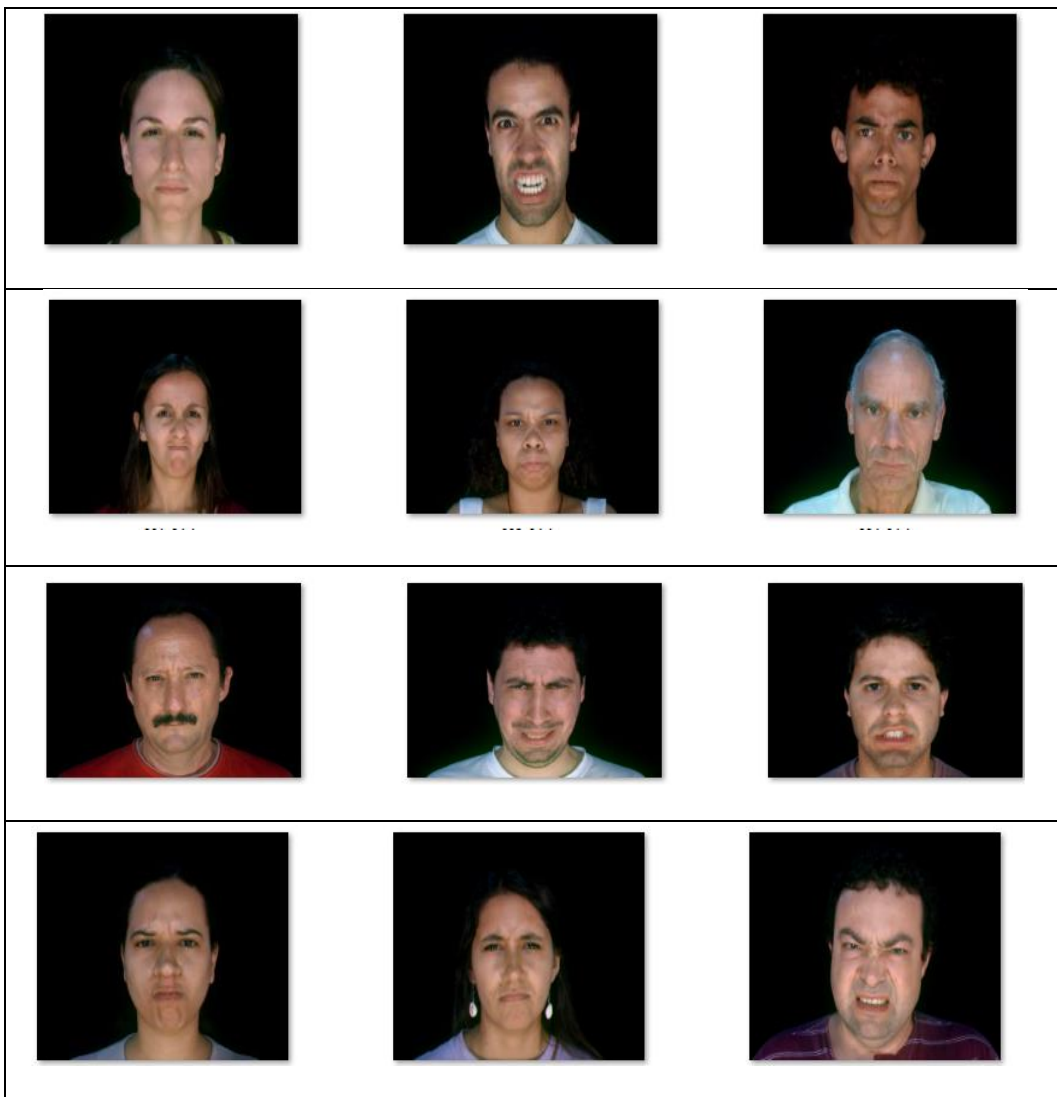
Dataset yang digunakan dalam penelitian ini adalah dataset *public* dengan nama *Collection of Facial Images* yang di unduh dari halaman <https://cswwww.essex.ac.uk/mv/allfaces/faces94.html>. Dataset ini merupakan data yang pada umumnya digunakan dalam eksperimen untuk deteksi wajah. Dengan jumlah data training 108 *image* dan data testing 32 *image*.

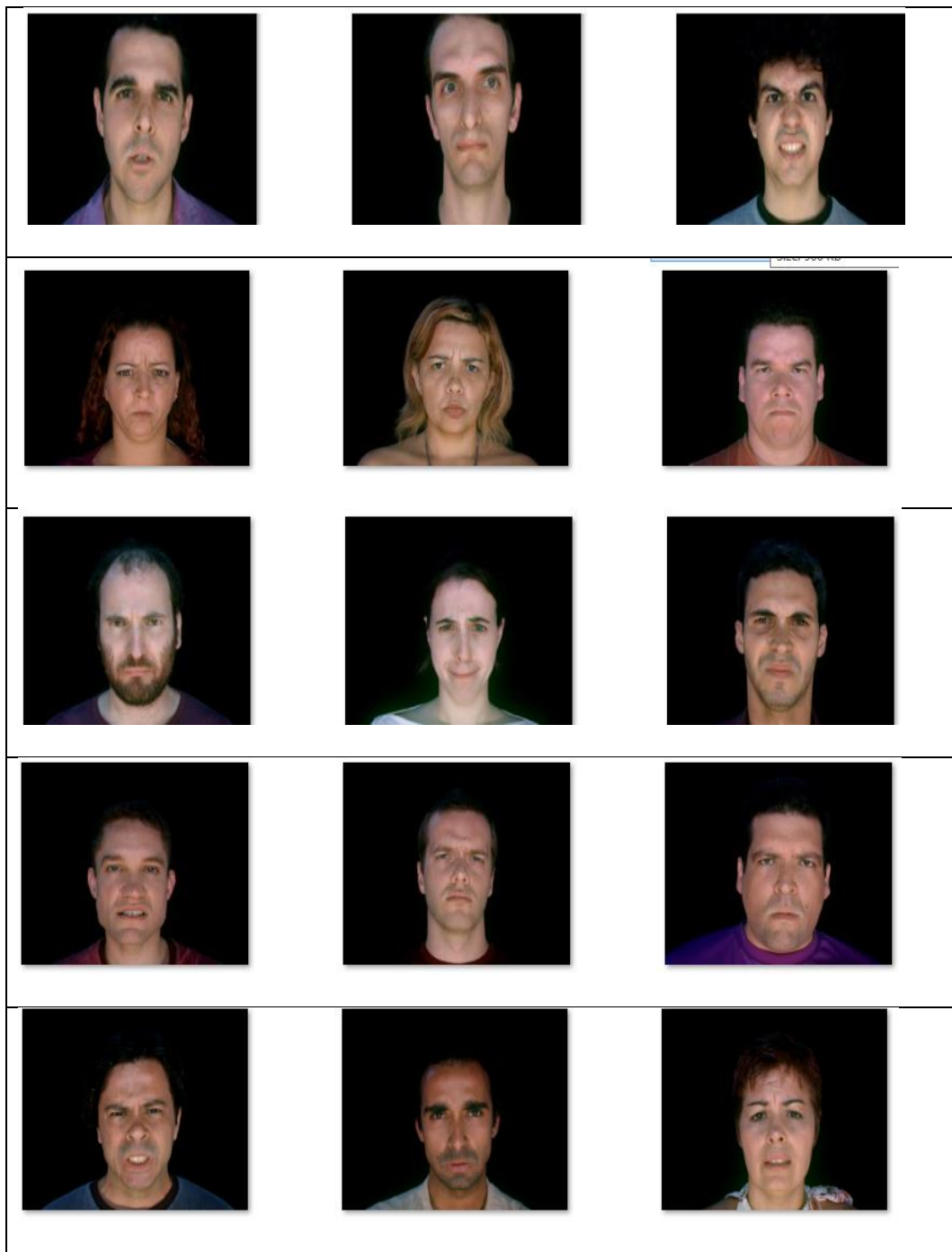




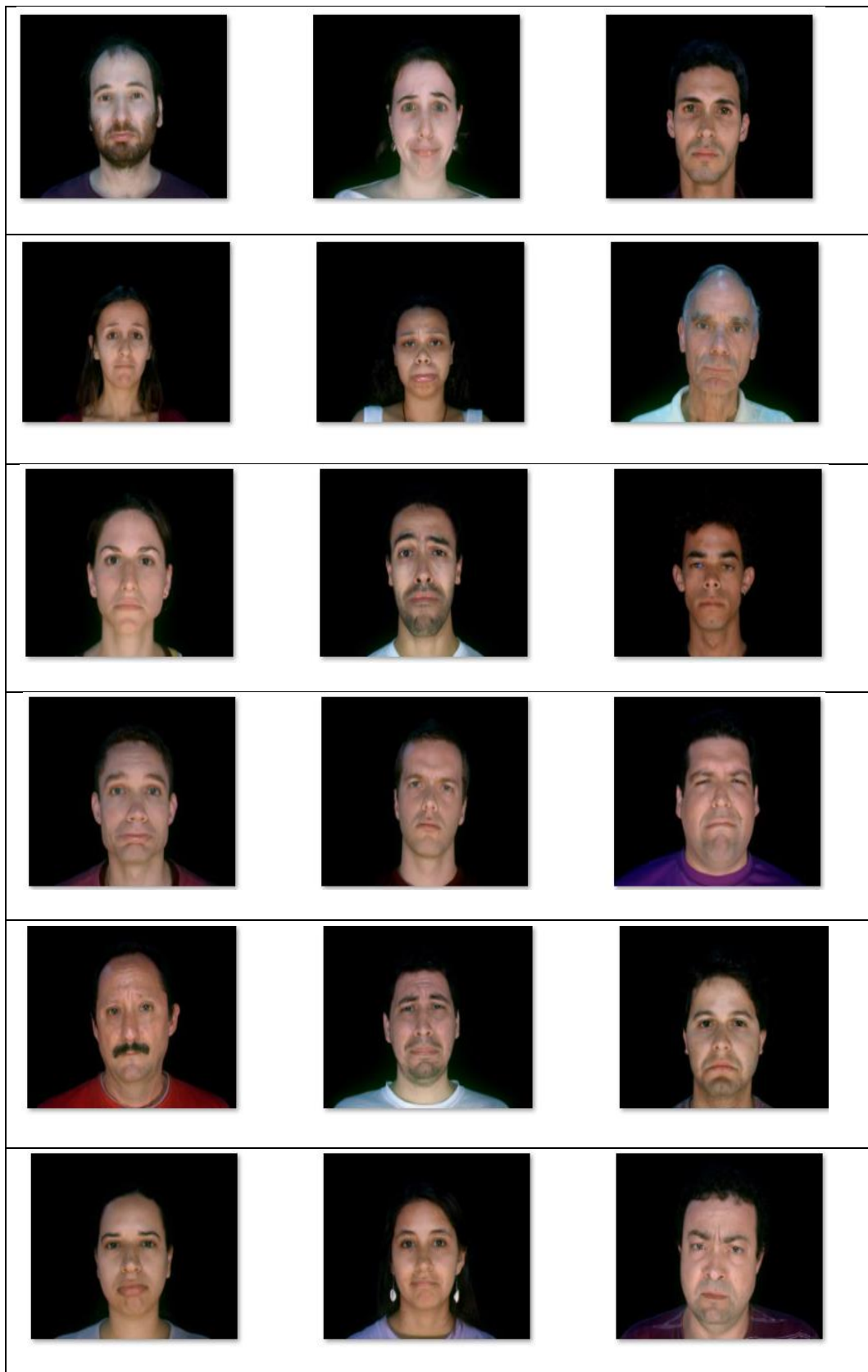


Gambar 4. 1 Ekspresi wajah Senyum





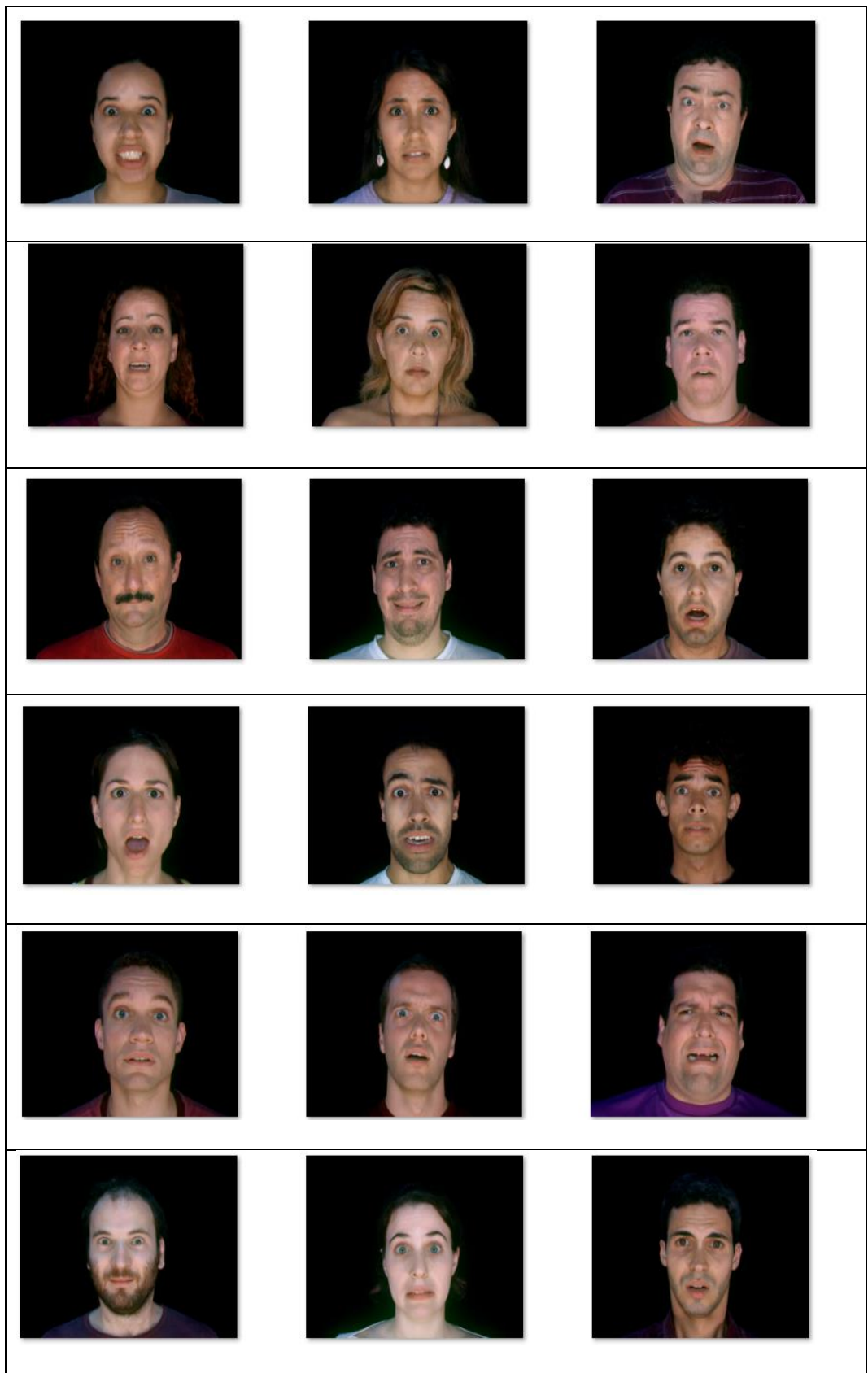
Gambar 4. 2 Ekspresi Wajah Marah

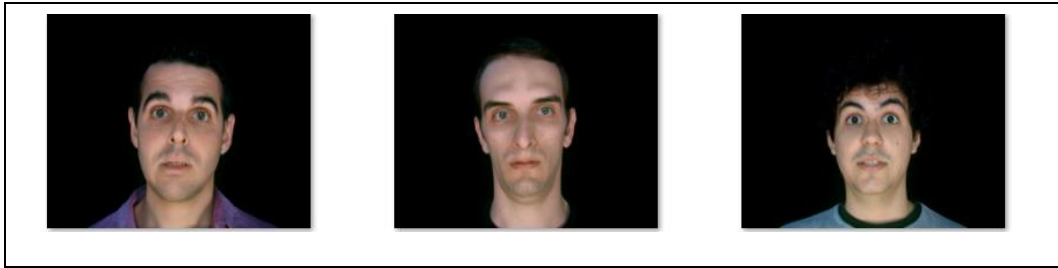




*Gambar4.3*Ekspresi Wajah Sedih







Gambar 4. 4 Ekspresi Wajah Takut

Dari data public yang tersedia 7 ekspresi, tapi pada penelitian ini penulis hanya mengambil 4 ekspresi karena berdasarkan penelitian yang dilakukan oleh Prawitasari dkk pada tahun 1990 ekspresi marah, sedih, senang, dan takut merupakan emosi yang tepat untuk mengenal ekspresi seseorang.

4.2. Hasil Pemodelan

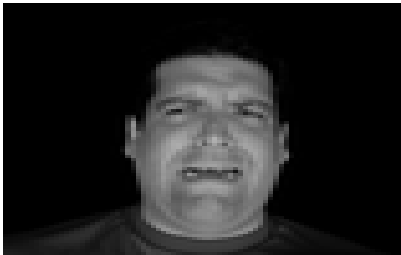
4.2.1. Pra Pengolahan Data

Dalam pemrosesan atau analisa citra secara digital, pra-pengolahan citra diartikan sebagai pemrosesan awal sebelum dilakukan pengolahan citra lebih lanjut. Prosedur ini bertujuan untuk memperbaiki data citra yang mengalami distorsi atau kesalahan kedalam bentuk aslinya.

Pra-pengolahan dilakukan dengan beberapa tahapan, yaitu:

1. Pengubahan citra warna ke grayscale

Pra-pengolahan proses awal yang merubah citra *training* atau citra *testing* yang awalnya citra dari RGB menjadi citra *grayscale*, perubahan ini dilakukan karena citra *grayscale* memiliki persamaan yang sederhana dan mampu mengurangi kebutuhan memory dimana nilai warna putih diwakili dengan angka 255 dan nilai warna hitam diwakili dengan angka 0.

CITRA ASLI	GREYSCALE
	
	
	
	

Tabel 4. 1 Konversi Citra Asli Ke GreyScale

Berikut ini merupakan rumus konversi citra RGB ke Grayscale dan rumus konversi citra grayscale ke citra histogram equalization bersertis perintah yang digunakan dalam bahasa python dan hasil outputnya:

Rumus Konversi Citra RGB ke Grayscale:

$$s = \frac{R + B + G}{3}$$

Dimana: R = Red, B = Blue, G = Green

Contoh sebuah Gambar memiliki nilai R = 142, G = 142, B = 160

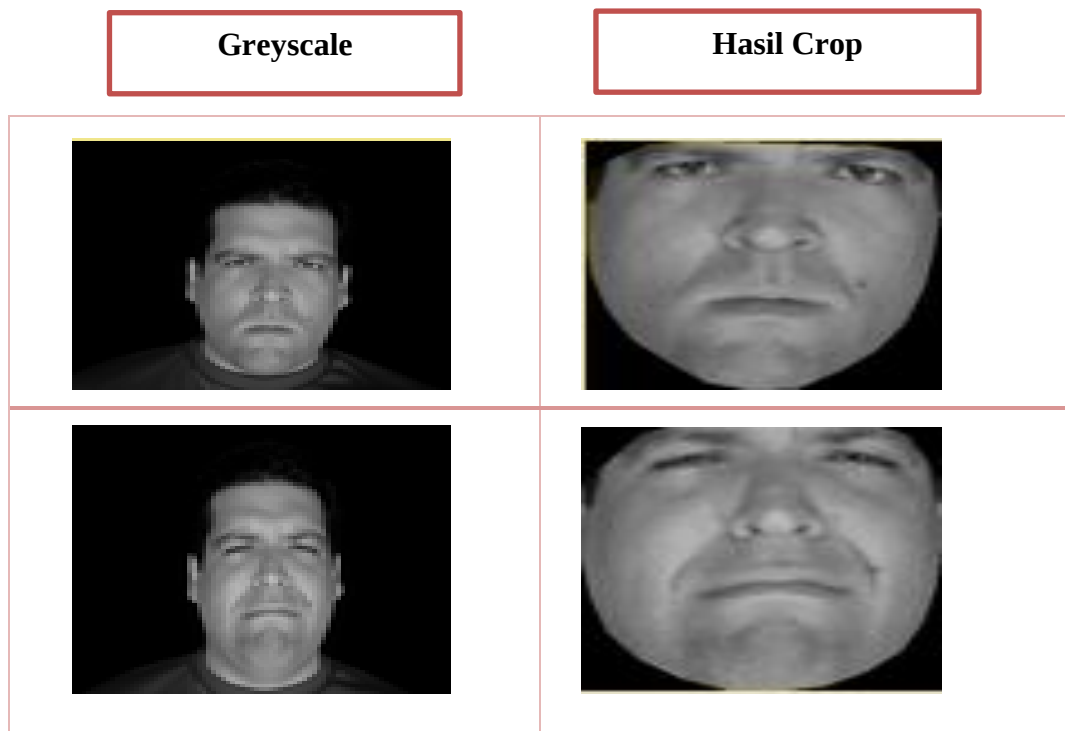
Konversikan Menjadi RGB

$$s = \frac{142 + 142 + 160}{3}$$

$$s = 1,332$$

2. Segmentasi Citra

Tahapan ini bertujuan untuk mempartisi citra menjadi bagian-bagian pokok yang mengandung informasi penting. Contohnya, memisahkan objek dan latar belakang.

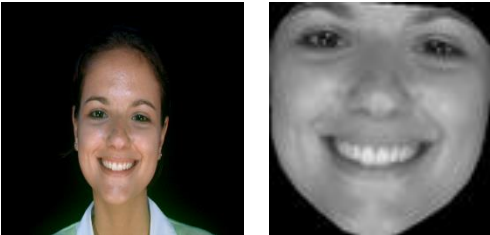




Tabel 4. 2 Konversi GreyScale ke Hasil Crop

4.2.2. Ekstraksi Ciri Gray Level Co-occurrence Matrix

Fitur ekstraksi Grey Level Co-occurrence Matrix (GLCM) merupakan matrix yang menggambarkan frekuensi munculnya pasangan dua pixsel dengan intensitas tertentu dalam jarak d dan orientasi arah sudut 0° , 45° , 90° , 135° . Tertentu dalam citra. Untuk GLCM 4 arah $d = 1$ artinya koordinat (x,y) adalah $(1,0)$. Contoh :



i/j	300	301	302	303	...	639
300	121	118	133	144		0
301	127	126	128	141		0
302	132	133	135	136		0
303	143	142	144	141		0
...						...
479	0	0	0	0	...	0

Gambar 4. 5 Data Testing Dalam Perhitungan GLCM

Setelah menentukan arahnya selanjutnya membentuk matriks kookurens dengan cara menghitung frekuensi kemunculan pasangan nilai keabuan piksel referensi dan piksel tetangga pada jarak dan arah yang ditentukan. Perhitungan

menggunakan nilai orientasi dengan arah sudut 0° . 0° yaitu nilai gambar yang memiliki nilai yang sama dan bersebelahan. Hasilnya sebagai berikut:

Tabel 4. 3 Nilai dari 0

J	0	118	121	126	127	128	132	133	135	136	141	142	143	144
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
118	0	0	1	1	0	0	0	0	0	0	0	1	0	0
121	0	1	0	0	1	0	0	0	0	0	0	1	0	0
126	0	1	0	0	1	1	0	1	0	0	0	1	0	0
127	0	0	1	1	0	0	1	0	0	0	0	1	0	0
128	0	0	0	1	0	0	0	0	1	0	1	1	0	0
132	0	0	0	0	1	0	0	1	0	0	0	1	1	0
133	0	1	0	1	0	1	1	0	1	0	0	0	0	0
135	0	0	0	0	0	1	0	1	0	1	0	1	0	1
136	0	0	0	0	0	0	0	0	1	0	2	1	0	0
141	0	0	0	0	0	1	0	0	0	2	0	1	0	1
142	0	0	0	0	0	0	1	1	0	0	0	1	1	1
143	0	0	0	0	0	0	0	0	0	0	0	0	0	0
144	0	0	0	0	0	0	0	0	1	0	2	0	0	0

Selanjutnya menormalisasikan matriks dengan cara menjumlahkan seluruh nilai yang ada dalam matriks.

Tabel 4. 3 Matriks Normalisasi

i/j	0	118	121	126	127	128	132	133	135	136	141	142	143	144
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
118	0	0	0,022	0,022	0	0	0	0	0	0	0	0,022	0	0
121	0	0,022	0	0	0,022	0	0	0	0	0	0	0,022	0	0
126	0	0,022	0	0	0,022	0,022	0	0,022	0	0	0	0,022	0	0
127	0	0	0,022	0,022	0	0	0,022	0	0	0	0	0,022	0	0
128	0	0	0	0,022	0	0	0	0	0,022	0	0,022	0,022	0	0
132	0	0	0	0	0,022	0	0	0,022	0	0	0	0,022	0,022	0
133	0	0,022	0	0,022	0	0,022	0,022	0	0,022	0	0	0	0	0
135	0	0	0	0	0	0,022	0	0,022	0	0,022	0	0,022	0	0,022
136	0	0	0	0	0	0	0	0	0,022	0	0,333	0,022	0	0
141	0	0	0	0	0	0,022	0	0	0	0,333	0	0,022	0	0,022
142	0	0	0	0	0	0	0,022	0,022	0	0	0	0,022	0,022	0,022
143	0	0	0	0	0	0	0	0	0	0	0	0	0	0
144	0	0	0	0	0	0	0	0	0,022	0	0,333	0	0	0

Langkah terakhir adalah menghitung ciri statistik GLCM yaitu :

a. **Contrast**

$$Contrast = \sum_i \sum_j (i - j)^2 P_{ij}$$

$$\begin{aligned}
 Contrast = & ((118-121)^2 * 0,022) + ((118-126)^2 * 0,022) + ((118-142)^2 * 0,022) \\
 & + ((121-118)^2 * 0,022) + ((121-127)^2 * 0,022) + ((118-121)^2 * 0,022) + ((118- \\
 & 18)^2 * 0,022) + ((126-127)^2 * 0,022) + ((126-133)^2 * 0,022) + ((126-142)^2 * ,022) \\
 & ((127-121)^2 * 0,022) + ((127-126)^2 * 0,022) + ((127-132)^2 * 0,022) + ((127-42)^2 \\
 & ,022) + ((128-126)^2 * 0,022) + ((128-135)^2 * 0,022) + ((128-141)^2 * 0,022) \\
 & ((128-142)^2 * 0,022) + ((132-127)^2 * 0,022) + ((132-133)^2 * 0,022) + ((132-42)^2 \\
 & ,022) + ((132-143)^2 * 0,022) + ((133-118)^2 * 0,022) + ((133-126)^2 * 0,022) \\
 & ((133-128)^2 * 0,022) + ((133-132)^2 * 0,022) + ((133-135)^2 * 0,022) + ((135-28)^2
 \end{aligned}$$

$$,022) + ((135-133)^2 * 0,022) + ((135-136)^2 * 0,022) + ((136-142)^2 * 0,022) + ((135-44)^2 * 0,022) + ((136-135)^2 * 0,022) + ((136-141)^2 * 0,333) + ((141-128)^2 * 0,022) + ((141-136)^2 * 0,022) + ((141-142)^2 * 0,022) + ((141-144)^2 * 0,022) + ((142-132)^2 * 0,022) + ((142-133)^2 * 0,022) + ((142-142)^2 * 0,022) + ((144-143)^2 * 0,022) + ((142-144)^2 * 0,022) + ((144-133)^2 * 0,022) + ((144-141)^2 * 0,333)$$

$$\begin{aligned} Contrast = & 9,022 + 64,022 + 576,022 + 9,022 + 36,022 + 441,022 + \\ & 64,022 + 1,022 + 49,022 + 256,022 + 64,022 + 1,022 + 25,022 + 225,002 \\ & + 4,022 + 49,022 + 169,022 + 196,022 + 25,022 + 1,022 + 100,022 + \\ & 100,022 + 196,022 + 49,022 + 25,022 + 1,022 + 4,022 + 49,022 + 4,022 \\ & + 25,022 + 36,022 + 81,022 + 1,022 + 8,325 + 169,022 + 1,665 + 1,022 \\ & + 9,022 + 100,022 + 81,022 + 0 + 1,022 + 4,022 + 121,022 + 2,997 \\ & + 1,199.154 + 1,823.308 \end{aligned}$$

$$Contrast = 345,3853462$$

b. Correlation

$$\frac{\sum_{i=1}^L \sum_{j=1}^L (i - \mu_i)(j - \mu_j)(GLCM(i, j))}{\sqrt{\sigma_i^2 \sigma_j^2}}$$

$$\begin{aligned} \mu_i = & (118 * 0,022) + (121 * 0,022) + (126 * 0,022) + (127 * 0,022) + \\ & (128 * 0,022) + (132 * 0,022) + (133 * 0,022) + (135 * 0,022) + (136 * \\ & 0,022) + (141 * 0,022) + (142 * 0,022) + (143 * 0,022) + (144 * 0,022) \\ & + \\ \mu_i = & 40.92 \end{aligned}$$

$$\begin{aligned} \mu_j = & (118 * 0,333) + (121 * 0,333) + (126 * 0,333) + (127 * 0,333) + \\ & (128 * 0,333) + (132 * 0,333) + (133 * 0,333) + (135 * 0,333) + (136 * \\ & 0,333) + (141 * 0,333) + (142 * 0,333) + (143 * 0,333) + (144 * 0,333) \\ \mu_j = & 510,822 \end{aligned}$$

$$\sigma i = \sqrt{\begin{aligned} &((0,053 (124 - 67,552)^2) + ((0,053 (124 - 67,552)^2) + \\ &((0,053(127 - 67,552)^2) + ((0,053(127 - 67,552)^2) + \\ &((0,053(131 - 67,552)^2) + ((0,053(132 - 67,552)^2) + \\ &((0,053(252 - 67,552)^2) + ((0,053(252 - 67,552)^2) + \\ &((0,053(253 - 67,552)^2) + ((0,053(253 - 67,552)^2) + + ((0,053(253 - 67,552)^2) + \\ &((0,053(255 - 67,552)^2) + ((0,053(255 - 67,552)^2) + ((0,053(255 - 67,552)^2) \end{aligned}}$$

$$\sigma i = \sqrt{\begin{aligned} &168,878 + 168,878 + 187,3054 + 187,3054 + 213,3594 + 220,1379 + \\ &1,803,116 + 1,803,116 + 1,822,721 + 1,822,721 + 1,822,721 + \\ &1,862,248 + 1,862,248 + 1,862,248 \end{aligned}}$$

$$\sigma i = 15,807$$

$$\sigma j =$$

$$\sqrt{\begin{aligned} &((0,053 (124 - 148,029)^2) + ((0,053 (124 - 148,029)^2) + \\ &((0,053(127 - 148,029)^2) + ((0,053(127 - 148,029)^2) + \\ &((0,053(131 - 148,029)^2) + ((0,053(132 - 148,029)^2) + \\ &((0,053(252 - 148,029)^2) + ((0,053(252 - 148,029)^2) + \\ &((0,053(253 - 148,029)^2) + ((0,053(253 - 148,029)^2) + ((0,053(253 - 148,029)^2) + \\ &((0,053(255 - 148,029)^2) + ((0,053(255 - 148,029)^2) + ((0,053(255 - 148,029)^2) \end{aligned}}$$

$$\sigma i = \sqrt{\begin{aligned} &30.60182 + 30,60182 + 23,4376 + 23,4376 + 15,3693 + \\ &13,61723 + 572,9284 + 572,9284 + \\ &584,0022 + 584,0022 + 584,0022 + \\ &606,4779 + 606,4779 + 606.4779 \end{aligned}}$$

$$\sigma j = 4,854,362$$

$$\begin{aligned} \text{Correlation} = &((127 - 67,552) * (127 - 148,029) * (0,053) / (15,807 * 4,854,362)) + \\ &((131 - 67,552) * (131 - 148,029) * (0,053) / (15,807 * 4,854,362)) + \\ &((124 - 67,552) * (127 - 148,029) * (0,053) / (15,807 * 4,854,362)) + \end{aligned}$$

$$\begin{aligned}
& (0,022*\log 0,022)+ (0,022*\log 0,022)+ (0,333*\log 0,333)+ \\
& (0,022*\log 0,022)+ (0,022*\log 0,022)+ (0,022*\log 0,022)+ \\
& (0,022*\log 0,022)+ (0,022*\log 0,022)+ (0,022*\log 0,022)+ \\
& (0,022*\log 0,022)+ (0,022*\log 0,022)+ (0,333*\log 0,333)+ \\
& = 6,6673200774
\end{aligned}$$

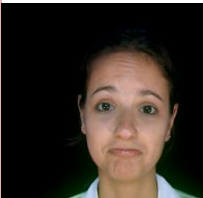
d. Energy

Energy menyatakan distribusi intensitas piksel terhadap jangkauan aras keabuan

$$\sum_{i,j} P(i,j)^2$$

Energy=

$$\begin{aligned}
& (0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+ \\
& (0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+ \\
& (0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+ \\
& (0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,333)^2+(0,022)^2+ \\
& (0,022)^2+(0,333)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+ \\
& (0,022)^2+(0,022)^2+(0,022)^2+(0,022)^2+(0,333)^2. \\
& = 1.872667
\end{aligned}$$

DATA TESTING	CON(X1)	ENE(X2)	COR(X3)	ENT(X4)
	96.44617435 712021	0.2092333658 5047858	0.987416643 8081994	6.335507391 24945
	146.9048456 260722	0.2129452683 3265698	0.982406411 0565544	6.165107343 271998
	117.1619214 8760334	0.1965809365 4541609	0.984033147 7229204	6.369007911 450185
	145.6929762 243063	0.1961060286 2854337	0.980584810 2467806	6.173735201 034688
	345.3853356 4654354	1.8726457676 9088987	0.975763403 8942277	6.607355865 639759
	154.1764270 6131082	0.2081111468 7488494	0.980954983 2847808	6.272556009 4438615

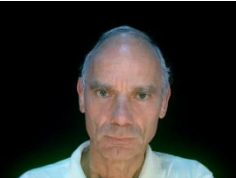
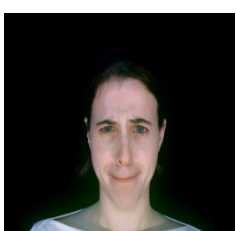
	122.4280378 7374079	0.2007381955 3353242	0.983595278 4983761	6.502747719 611277
	153.3706790 1234573	0.2158785452 7736489	0.980930042 7983503	6.224432400 559386

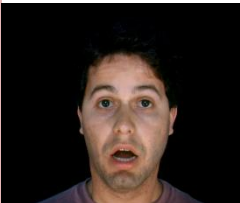
Tabel 4. 4 Data Testing

4.2.4 Klasifikasi

K-Nearest Neighbor adalah Algoritma *k-nearest neighbor* (K-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. Sebuah objek akan diklasifikasikan berdasarkan pilihan terbanyak dari tetangga-tetangganya

Proses klasifikasi dilakukan untuk mengklasifikasikan setiap hasil akhir dari citra ekspresi wajah, akan menampilkan beberapa hasil akhir berupa ekspresi senyum, marah, sedih dan takut dari setiap orang. Kemudian akan dilihat tingkat keakurasi setiap ekspresi.

DATA	CON(X1)	ENE(X2)	COR(X3)	ENT(X4)	Y
 Gambar 0	134.88772 42021896	0.1966757 18390064 8	0.97931875 83824489	6.40865366 8198046	Marah
 Gambar 1	112.30334 21542683 8	0.1786437 97278146 58	0.97930632 4698162	6.51300775 2868001	Marah
 Gambar 2	130.00595 86325152 5	0.2171545 12708942 43	0.98016693 7580202	6.13488962 13696555	Marah
 Gambar 3	59.351815 61208959	0.1699173 92471727 27	0.99192419 51786224	7.26034386 38922775	Marah
 Gambar 4	163.38225 02685284 2	0.2159871 13822672 77	0.98224394 24508631	6.24900809 1633212	Marah
....					
 Gambar 103	107.55226 84552289 8	0.1880647 45122010 2	0.98161885 45890232	6.74393382 7326107	Takut

	86.341938 67096016	0.1982068 74120612 28	0.98698068 61390942	6.71939080 8676	Takut
Gambar 104					
	103.75586 8544601	0.1921049 86612390 3	0.98476986 92542416	6.67877133 70521296	Takut
Gambar 105					
	100.93040 81099895 3	0.2231972 18213863	0.98541818 0789423	6.32973677 8516908	Takut
Gambar 106					
	125.01978 74493927 4	0.1721464 81256607 3	0.98338417 35939888	6.77456945 9006519	Takut
Gambar 107					
	96.446174 35712021	0.2092333 65850478 58	0.98741664 38081994	6.33550739 124945	?
Gambar Test					

Tabel 4.5 Klasifikasi Data Training

Euclidean Distance

$$\begin{aligned}
 d &= \sqrt{(134,8877 - 96,4461)^2 + (0,1966 - 0,2092)^2 + (0,9793 - 0,9874)^2 + (6,4086 - 6,3355)^2} \\
 &= \sqrt{1,477,7566 + 0,0001 + 0,00006 + 0,0053} \\
 &= \sqrt{1,477,7620} \\
 &= 38,4416
 \end{aligned}$$

Data Training	Perhitungan jarak <i>Euclidean Distance</i>	Nilai
0	$d = \sqrt{1,477,7566 + 0,0001 + 0,00006 + 0,0053}$	38,4416
1	$d = \sqrt{251,4507 + 0,0009 + 0,00006 + 0,0315}$	15,8582
2	$d = \sqrt{1,126,2601 + 0,00006 + 0,00005 + 0,0402}$	33,5604
3	$d = \sqrt{1,375,9870 + 0,0015 + 0,00002 + 0,8552}$	37,1058
4	$d = \sqrt{4,480,4414 + 0,00004 + 0,00002 + 0,00748}$	66,9361
....		
103	$d = \sqrt{123,3454 + 0,0004 + 0,00003 + 0,1667}$	11,1136
104	$d = \sqrt{102,0948 + 0,0001 + 0,00003 + 0,1473}$	10,1114
105	$d = \sqrt{53,4317 + 0,0002 + 0,000007 + 0,1177}$	7,3177
106	$d = \sqrt{20,1089 + 0,0139 + 0,000004 + 0,00003}$	4,4858
107	$d = \sqrt{816,4506 + 0,0013 + 0,00001 + 0,1927}$	28,5769

Tabel 4.6 Perhitungan Euclidean Distance

Data	Euclidean	Y
106	4,4858	Takut
22	6,8954	Sedih
105	7,3177	Takut
104	10,1114	Takut
103	11,1136	Takut
1	15,8582	Marah
21	16,4219	Sedih
....		
45	24,1125	Senyum
42	28,4873	Senyum
107	28,5769	Takut
2	33,5604	Marah
3	37,1058	Marah
0	38,4416	Marah
4	66,9361	Marah

Tabel 4.7 Hasil pengurutan dari nilai terkecil

a	Euclidean	Y
106	4,4858	Takut
22	6,8954	Sedih
105	7,3177	Takut
104	10,1114	Takut
103	11,1136	Takut
1	15,8582	Marah
21	16,4219	Sedih

Tabel 4. 8 Jumlah K=7

Y	Jumlah Data
Takut	4
Sedih	2
Marah	1

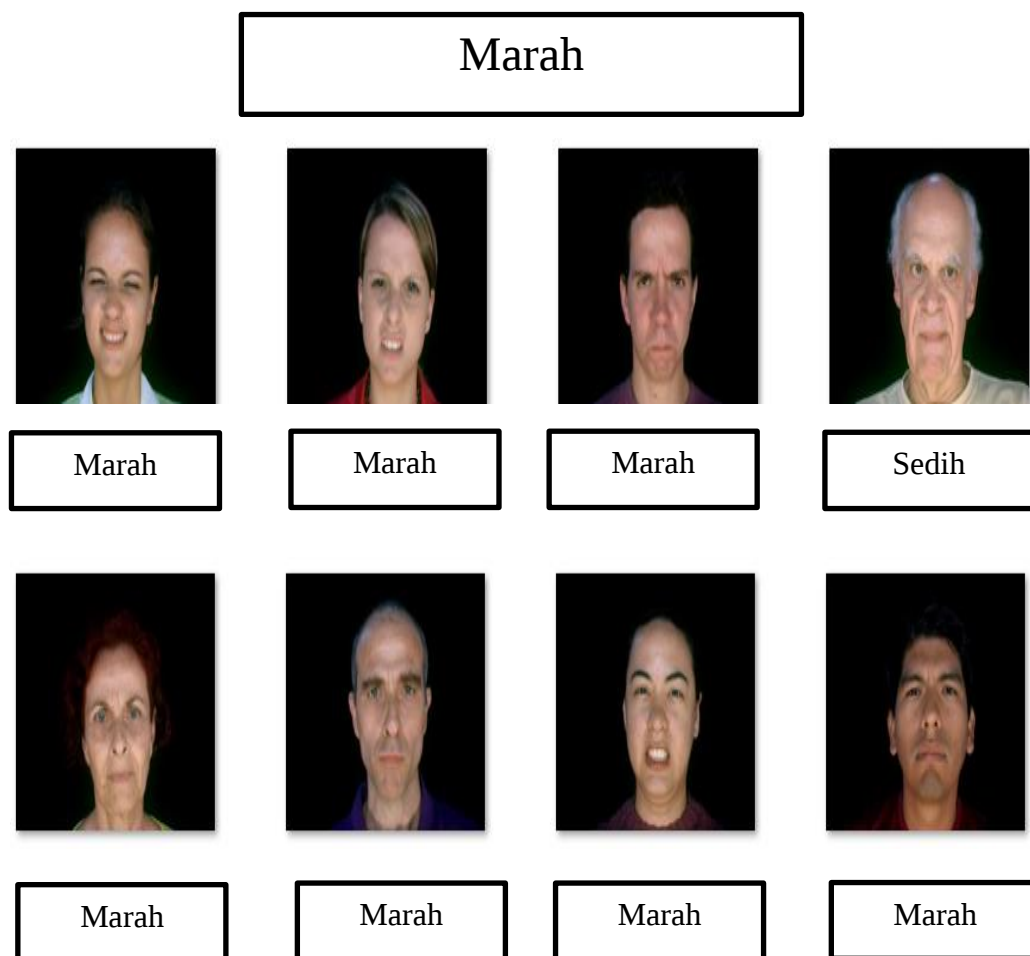
Jumlah data terbanyak menentukan kelas dari data uji

Tabel 4. 9 Kelas Data Uji

Hasil yang didapatkan dari perhitungan jarak euclidean distance untuk data uji adalah “Takut”.

4.3. Confusion Matrix

Confusion matrix menurut dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah classifier tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari TruePositive dan True-Negative memberikan informasi ketika classifier dalam melakukan klasifikasi data bernilai benar, sedangkan FalsePositive dan False-Negative memberikan informasi ketika classifier salah dalam melakukan klasifikasi data.



Senyum



Senyum

Senyum

Sedih

Sedih



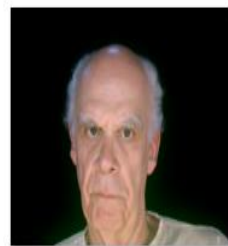
Senyum

Sedih

Senyum

Senyu

Sedih



Sedih

Sedih

Sedih

Marah



Sedih



Sedih



Sedih



Sedih

Takut



Takut



Takut



Takut



takut



Sedih



Sedih



Sedih



Takut

Tabel 4. 10 Confusion Matrix (Data Testing)

	Marah	Sedih	Senyum	Takut
Marah	7	1	0	0
Sedih	1	7	0	0
Senyum	0	3	5	0
Takut	0	3	0	5

Tabel 4. 11 Tabel Hasil Data Uji

Jumlah data yang diklasifikasi dengan benar = 24

Jumlah data yang diklasifikasi salah = 8

$$\text{Akurasi} = \frac{\text{Data Uji Dikenali}}{\text{Data uji}} \times 100 \% \quad \text{Akurasi} = \frac{24}{32} \times 100 \% = 75 \%$$

BAB V

PEMBAHASAN

5.1 Pembahasan Model

Data yang digunakan pada penelitian ini terdiri dari data *public* dengan nama *Collection of Facial Images* yang di unduh dari halaman <https://cswwww.essex.ac.uk/mv/allfaces/faces94.html>. Masing-masing 104 Data Training dan 32 Data Testing.

Berdasarkan hasil penelitian data yang diambil terdiri dari 4 kategori yaitu Senang, Sedih, Marah dan Takut. Selanjutnya citra wajah diekstraksi ciri menggunakan *Gray Level Co-occurence Matrix* dan *Principal Component Analysis*. Setelah didapatkan ciri dari citra wajah selanjutnya diklasifikasikan menggunakan *K-Nearest Neighbor*. Pada tahap ini ciri dari citra wajah dijadikan sebagai inputan untuk mencari nilai Akurasi Terbaik Pada percobaan model K-NN, sebagai berikut :

Data Training	Data Testing	Arah GLCM	Jarak	Jumlah K	Akurasi	Ekspresi
108	32	0°	1	1	87%	Marah
108	32	0°	1	1	62%	Sedih
108	32	135°	5	1	87%	Senyum
108	32	135°	5	1	62%	Takut

Data Training	Data Testing	Arah GLCM	Jarak	Jumlah K	Akurasi	Ekspresi
108	32	0°	1	3	87%	Marah
108	32	0°	1	3	87%	Sedih
108	32	135°	5	3	75%	Senyum
108	32	135°	5	3	62%	Takut

Data Training	Data Testing	Arah GLCM	Jarak	Jumlah K	Akurasi	Ekspresi
108	32	0°	1	5	87%	Marah
108	32	0°	1	5	100%	Sedih
108	32	135°	5	5	87%	Senyum
108	32	135°	5	5	75%	Takut

Data Training	Data Testing	Arah GLCM	Jarak	Jumlah K	Akurasi	Ekspresi
108	32	0°	1	7	87%	Marah
108	32	0°	1	7	100%	Sedih
108	32	135°	5	7	100%	Senyum
108	32	135°	5	7	87%	Takut

Tabel 5.1 Hasil Pemodelan

Dalam tahap perhitungan berdasarkan Jumlah K, Jumlah K 7 adalah hasil terbaik dalam tingkat akurasi sebesar 100% pada ekspresi sedih dan senyum sedangkan untuk akurasi terendah jumlah K 1 dengan tingkat akurasi 62% pada ekspresi sedih dan takut.

Pada tahap Confusion Matrix terdapat kendala pengenalan ekspresi, Beberapa ekspresi keluar dengan hasil yang tidak sesuai. Arah, Jarak dan Jumlah K berpengaruh untuk mendapatkan hasil yang sesuai. Hasil yang didapatkan menggunakan metode GLCM dan K-NN dengan arah 0°, 135°, Jarak 1, 5 dan Jumlah K 1 lebih banyak mengenali ekspresi Marah dan Senyum dengan akurasi 87%. Sedangkan untuk ekspresi Sedih dan Takut tingkat akurasinya hanya senilai 62%. Dengan arah 0°, 135°, Jarak 1, 5 dan Jumlah K 3 lebih banyak mengenali ekspresi Marah dan Sedih dengan akurasi 87%. Sedangkan untuk ekspresi Senyum tingkat akurasi 75% dan Takut tingkat akurasinya paling rendah hanya senilai 62%. Dengan arah 0°, 135°, Jarak 1, 5 dan Jumlah K 7 lebih banyak mengenali ekspresi Sedih dan senyum dengan tingkat akurasi 100%. Sedangkan untuk ekspresi Marah

dan Takut tingkat akurasi hanya 87%. Pada penelitian ini penggunaan metode K-NN kurang baik untuk tahap klasifikasi.

Agar aplikasi berjalan dengan baik maka Spesifikasi hardware dan Software yang direkomendasikan adalah:

1. *Processor* : *Core i3-7020U CPU @2.30GHz*
2. *RAM* : 4GB DDR4
3. *VGA* : 16 Bit
4. *Hardisk* : 500 GB
5. *Operating System* : *Windows 10*

BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan pada ekspresi wajah dan pembahasan yang telah diuraikan sebelumnya maka dapat ditarik suatu kesimpulan bahwa:

1. Hasil uji coba metode *Grey Level Co-onccurrence Matriks* pada pengenalan ekspresi wajah mendapatkan hasil yang kurang baik, hal ini ditunjukan karena ada beberapa hasil yang keluar tidak sesuai itu terjadi karena Arah, Jarak dan Jumlah K yang berbeda.
2. Dapat menggunakan metode *K-nearest neighbor* untuk Klasifikasi, mendapatkan tingkat akurasi sebesar 75% .

6.2. Saran

Setelah melakukan Penelitian Reduksi Fitur pada pengenalan ekspresi wajah, ada beberapa saran yang perlu di perhatikan untuk mencapai tujuan yang di harapkan, yaitu sebagai berikut:

1. Sebaiknya data atau citra wajah yang digunakan sebagai data *Training* dan *Testing* benar-benar menunjukan posisi yang menampilkan semua komponen wajah (mata, hidung, mulut, alis)
2. Penulis berharap untuk penelitian selanjutnya agar mendapatkan hasil akurasi yang lebih baik dengan metode selain metode *Grey Level Co-onccurrence Matriks* dan *K-nearest neighbor*

DAFTAR PUSTAKA

- [1] Lingga Sekar Utami, “Ekspresi Wajah Anak - Anak Dalam Penciptaan Karya Lukis”, 2014.
- [2] Claudia Primasiwi, Handayani Tjandrasa, Dini A. Navastara, “Deteksi Ekspresi Wajah Menggunakan Fitur Gabor dan Haar Wavelet”, Jurnal Teknik ITS Vol. 7, No. 1 2337-3520, November 2018.
- [3] Lutfy, Abory, “Sistem Pengenalan Ekspresi Wajah Menggunakan Metode Back Propagation dan Wavelet Haar”, Jurnal Pelayanan Kesehatan, 2010.
- [4] Daniel Goleman, Richard Boyatzis, Annie Mckee “Metode GLCM (Gray Level Co-Occurrence Matrix), Journal of Chemical Information and Modeling Vol 53, No 9, 2019.
- [5] Sufiatul Maryana, Lita Karlitasari, Arie Qur’ania Email “Pemanfaatan K-Nearest Neighbor(K-NN) Pada Pengenalan Wajah Dengan Praproses Transformasi Wavelet”, Komputasi, Vol. 9, No. 1, Januari 2012.
- [6] Rionaldi Ali, “Detektor Ekspresi Wajah Manusia”, Jurnal Informatika, Vol.16, No. 1, juni 2016.
- [7] Tamrin Imanuel Panggabean, “Pendeteksian Dan Pengenalan Wajah Manusia Untuk Peningkatan Kinerja Kamera Pengaman”, Skripsi Informatika, Februari 2018.
- [8] Restu Widodo, Agus Wahyu Widodo, Arry Supriyanto, “Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (Citrus reticulata Blanco) untuk Klasifikasi Mutu”, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol. 2, No. 11, November 2018.
- [9] I Gusti Rai Agung Sugiarta, Made Sudarma, I Made Oka Widyantara, “Ekstraksi Fitur Warna, Tekstur dan Bentuk untuk Clustered-Based Retrieval of Images (CLUE)”, Teknologi Elektro, Vol. 16, No1, Januari-April 2017.

- [10] Herralta Bus Umar, “Principal Component Analysis (PCA) dan Aplikasinya Dengan SPSS”, Jurnal Kesehatan Masyarakat, Maret - September 2009, Vol. 03, No. 2.
- [11] Subairi, Rahmadwati, Erni Yudaningsy, “Implementasi Metode K-Nearest Neighbor pada Pengenalan Pola Tekstur Citra Saliva untuk Deteksi Ovulasi”, Jurnal EECCIS Vol. 12, No. 1, April 2018.
- [12] Ida Ayu A. Angreni, Sakti Adji Adisasmita, M. Isran Ramli dan SumarniHamid, “Pengaruh Nilai K Pada Metode K-Nearest Neighbor Terhadap Tingkat Akurasi Identifikasi Kerusakan Jalan,” Rekayasa Sipil, Vol. 7 No. 2 . September 2018
- [13] Mercury Fluorida Fibrianda, Adhitya Bhawiyuga, “Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM)”, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol. 2, No. 9, September 2018.

Lampiran 1: KODE PROGRAM

```
import cv2

import numpy as np

from skimage import feature
from scipy.stats import entropy
from imutils import paths

import os

from scipy import stats

from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA

import dlib

from scipy.spatial import ConvexHull
from skimage import draw

class analisis_citra:

    #=====
    #Proses Pra-pengolahan gambar(Grayscale dan Histtogram ekualisasi)
    #=====

    def getFace(self, img):

        detector = dlib.get_frontal_face_detector()

        predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        face = detector(img)[0]

        sp = predictor(gray, face)

        landmarks = np.array([[p.x,p.y] for p in sp.parts()])

        vertices = ConvexHull(landmarks).vertices

        Y,X = draw.polygon(landmarks[vertices,1],
landmarks[vertices, 0])
```

```

        cropped_img = np.zeros(gray.shape, dtype=np.uint8)

        cropped_img[Y,X] = gray[Y,X]

        rect = cv2.boundingRect(cropped_img)

        x,y,w,h = rect

        result = cropped_img[y:y+h, x:x+w].copy()

        return result

=====
#Proses ekstraksi fitur GLCM
=====

    def fitur_glcm(self, img, Arah, Jarak):

#        imggray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        matrix_glcm = feature.greycomatrix(img, [Jarak], [Arah],
symmetric=False, normed=True)

        contrast          =          feature.greycoprops(matrix_glcm,
'contrast')[0,0]

        energy = feature.greycoprops(matrix_glcm, 'energy')[0,0]

        correlation          =          feature.greycoprops(matrix_glcm,
'correlation')[0,0]

        ent = stats.entropy(matrix_glcm.flatten())

        hasil_fitur = [contrast, energy, correlation, ent]

        return hasil_fitur

=====
#Proses membaca folder data training dan data testing
=====

    def get_Img_Path(self, folderPath):

        imPath = []

        for path in paths.list_images(folderPath):

            imPath.append(path)

        return imPath

    def get_label(self, folderPath):

        label = []

```

```

        imPath = self.get_Img_Path(folderPath)

        for item in imPath:

            label.append(os.path.split(os.path.dirname(item))[-1])

        return label

def training_data(self, folderPath, Arah, Jarak):

    pathList = self.get_Img_Path(folderPath)

    images = []

    for imPath in pathList:

        img = cv2.imread(imPath)

        images.append(img)

    data_train = []

    for img in images :

        hasil_crop = self.getFace(img)

        ekstrasi = self.fitur_glcml(hasil_crop, Arah, Jarak)

        data_train.append(ekstrasi)

    label_train = self.get_label(folderPath)

    return data_train, label_train

def testing_data(self, img, Arah, Jarak):

    data_test = []

    img = cv2.imread(img)

    imggray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    hasil_crop = self.getFace(img)

    hasil_ekstrak_glcml = self.fitur_glcml(hasil_crop, Arah,
Jarak)

    data_test.append(hasil_ekstrak_glcml)

    return data_test, hasil_crop, imggray

def proses_rekognisi(self, folder_train, image_test, Arah,
Jarak, K):

```

```

        x_train, y_train = self.training_data(folder_train, Arah,
Jarak)

        x_test, hasil_crop, gray = self.testing_data(image_test,
Arah, Jarak)

        allData = self.encode_data(x_train + x_test)

        pca = PCA(n_components=2)

        pca.fit(allData)

        hasil_pca = pca.transform(allData)

        data_train = hasil_pca[0:-int(len(x_test)):]

        data_test = hasil_pca[-int(len(x_test)):]

        clf = KNeighborsClassifier(n_neighbors=5,
weights='uniform',

                                algorithm='auto',
metric='euclidean')

        clf.fit(data_train, y_train)

        final_hasil = clf.predict(data_test)

        return final_hasil[0], hasil_crop, gray

=====
#Proses Klasifikasi
=====

    def encode_data(self, data):

        scaler = MinMaxScaler(feature_range=(0,1))

        result = scaler.fit_transform(data)

        return result

#def main():

#    a = analisis_citra()

#    folder_train = "E:/Nuraini/Ain/DATAAIN/data_training"

#    img_test = 'test.jpg'

#    result = a.proses_rekognisi(folder_train, img_test)

#    print(result)

#    cv2.waitKey(0)

#    cv2.destroyAllWindows()

```

```

#if __name__=='__main__':

    #    main()

#=====
#Pembuatan form
#=====

import tkinter as tk

import tkinter.filedialog as fdialog

import tkinter.ttk as ttk

from PIL import ImageTk, Image

#from sklearn.preprocessing import ordinalEncoder

import datapublic as ft

import cv2

class aplikasiku():

    def __init__(self, parent):

        self.parent = parent

        self.frames()

        self.widget()

    def frames(self):

        self.frameAtas = tk.Frame(self.parent, width=998,
height=100,
borderwidth=0,
background='antiquewhite', relief='raised')

        self.frameAtas.grid(row=0, column=0, columnspan=2,
sticky="NSEW")

        self.frameAtas.grid_propagate(0)

        self.frameKiri = tk.Frame(self.parent, width=874,
height=500,

```

```

borderwidth=0,
background='silver', relief='raised')

self.frameKiri.grid(row=1, column=0, columnspan=2,
sticky="NSEW")

self.frameKiri.grid_propagate(0)


self.frameKanan = tk.Frame(self.parent, width=150,
height=500,

borderwidth=0,
background='lightgray', relief='raised')

self.frameKanan.grid(row=1, column=1, sticky="NSEW")

self.frameKanan.grid_propagate(0)


def widget(self):

self.PathTestingImg = 'E:\gambarain\smile2.png'

self.imT = Image.open(self.PathTestingImg)

self.imT = ImageTk.PhotoImage(self.imT)

self.label_dataTrain = tk.Label(self.frameAtas, width=988,
height=100, bg="aliceblue", image=self.imT)

self.label_dataTrain.pack(side="left")


self.MainFrame1 = tk.LabelFrame(self.frameKiri, height=20,
background="lightgray", borderwidth=3, relief='groove',
text="Input Gambar", fg="black")

self.MainFrame1.grid(row=1, column=0, padx=10,
sticky="NSEW")

self.MainFrame1.grid_propagate(0)


self.label_dataTrain = tk.Label(self.frameKiri,
text='Image Training', font='Perpetua',

relief='ridge', width=20,
background='khaki')

self.label_dataTrain.grid(row=2, column=0, padx=3, pady=3,
sticky='E')

```

```

        self.label_dataTesting = tk.Label(self.frameKiri,
text='Image Testing', font='Perpetua',
relief='ridge', width=20,
background='khaki')

        self.label_dataTesting.grid(row=3, column=0, padx=3,
pady=3, sticky='E')


        self.label_Arah = tk.Label(self.frameKiri, text='Arah
GLCM', font='Perpetua',
relief='ridge', width=20,
background='khaki')

        self.label_Arah.grid(row=5, column=0, padx=3, pady=3,
sticky='E')


        self.label_Jarak = tk.Label(self.frameKiri, text='Jarak',
font='Perpetua',
relief='ridge', width=20,
background='khaki')

        self.label_Jarak.grid(row=6, column=0, padx=3, pady=3,
sticky='E')


        self.label_nilaiK = tk.Label(self.frameKiri, text='Jumlah
K', font='Perpetua',
relief='ridge', width=20,
background='khaki')

        self.label_nilaiK.grid(row=8, column=0, padx=3, pady=3,
sticky='E')


        self.MainFrame2 = tk.LabelFrame(self.frameKiri, height=20,
background="lightgray", text="GLCM", fg="black")

        self.MainFrame2.grid(row=4, column=0, padx=10, pady=10,
sticky="NSEW")

        self.MainFrame2.grid_propagate(0)

```



```
self.text_nilaiK.grid(row=8, column=1, padx=3, pady=3,
sticky='W')
```

```
self.button_dataTrain = tk.Button(self.frameKiri,
text='Select', font='Perpetua', relief='sunken', width='10',
background='lightgray',
```

```
command=lambda:value_dataTrain.set(fdialog.askdirectory()))
```

```
self.button_dataTrain.grid(row=2, column=2, padx=3,
pady=3, sticky="NSEW")
```

```
self.button_dataTesting = tk.Button(self.frameKiri,
text='Select', font='Perpetua', relief='sunken', width='10',
background='lightgray',
```

```
command=lambda:value_dataTesting.set(fdialog.askopenfilename()))
```

```
self.button_dataTesting.grid(row=3, column=2, padx=3,
pady=3, sticky="NSEW")
```

```
self.button_proses = tk.Button(self.frameKiri,
text="Proses", font="Perpetua", width=20, background="lightgray",
```

```
command=self.proses)
```

```
self.button_proses.grid(row=10, column=0, padx=3, pady=40,
sticky="NSEW")
```

```
self.valdist = tk.StringVar(self.frameKiri, value="")
```

```
self.cb_Arah = tkk.Combobox(self.frameKiri,
textvariable=self.valdist,
```

```
width=20, state='readonly')
```

```
self.cb_Arah['values']=("0","45","90","135")
```

```
self.cb_Arah.grid(row=5, column=1, padx=3, pady=3,
sticky='W')
```

```
self.labell = tk.Label(self.frameKanan, text="Image
Testing", background="lightgray", font="Perpetua")
```

```
self.label1.grid(row=1, column=0, sticky="NSEW", padx=8,
pady=10)
```

```
self.noimage = 'E:\gambarain\pro-icon.png'
self.noimageTest = Image.open(self.noimage)
self.noimageTest = ImageTk.PhotoImage(self.noimageTest)
self.labelTest = tk.Label(self.frameKanan, bg = "khaki",
image=self.noimageTest)
self.labelTest.grid(row=5,column = 0, padx=50)
```

```
self.label2 = tk.Label(self.frameKanan, text="GreyScale",
background="lightgray", font="Perpetua")
self.label2.grid(row=1, column=4, sticky="NSEW", padx=8,
pady=10)
```

```
self.noimage1 = 'E:\gambarain\pro-icon.png'
self.noimageGrey = Image.open(self.noimage1)
self.noimageGrey = ImageTk.PhotoImage(self.noimageGrey)
self.labe2Grey = tk.Label(self.frameKanan, bg = "khaki",
image=self.noimageGrey)
self.labe2Grey.grid(row=5,column =4, padx=50)
```

```
self.label3 = tk.Label(self.frameKanan, text="HasilCrop",
background="lightgray", font="Perpetua")
self.label3.grid(row=1, column=5, sticky="NSEW", padx=8,
pady=10)
```

```
self.noimage2 = 'E:\gambarain\pro-icon.png'
self.noimagehasil_crop = Image.open(self.noimage2)
self.noimagehasil_crop = ImageTk.PhotoImage(self.noimagehasil_crop)
self.labe3hasil_crop = tk.Label(self.frameKanan, bg =
"khaki", image=self.noimagehasil_crop)
```

```

self.labe3hasil_crop.grid(row=5, column =5, padx=50)

self.label4 = tk.Label(self.frameKanan, text="OUTPUT",
width=15,height=5, font="Perpetua")

self.label4.grid(row=15,          column=0,          sticky="W",
columnspan=10, padx=50, pady=40)

def proses(self):
    foldertraining = self.text_dataTrain.get()
    imagetesting = self.text_dataTesting.get()
    arah = self.cb_Arah.get()
    jarak = self.text_jarak.get()
    K = self.text_nilaiK.get()

    a = ft.analisis_citra()

    hasil,          hasil_crop,          gray          =
a.proses_rekognisi(foldertraining, imagetesting, arah, jarak, K)

    self.label4.configure(text=hasil)

    imgAsli = Image.open(imagetesting)
    imgAsli = imgAsli.resize((80,84), Image.ANTIALIAS)
    imgAsli = ImageTk.PhotoImage(imgAsli)
    self.labelTest.configure(image=imgAsli)
    self.labelTest.image_names = imgAsli

    imgGray = Image.fromarray(gray)
    imgGray = imgGray.resize((80,84), Image.ANTIALIAS)

```

```

        imgGray = ImageTk.PhotoImage(imgGray)
        self.labe2Grey.configure(image=imgGray)
        self.labe2Grey.image_names = imgGray

        imcrop = cv2.cvtColor(hasil_crop, cv2.COLOR_BGR2RGBA)
        imghasil_crop = Image.fromarray(imcrop)
        imghasil_crop = imghasil_crop.resize((80,84),
Image.ANTIALIAS)
        imghasil_crop = ImageTk.PhotoImage(imghasil_crop)
        self.labe3hasil_crop.configure(image=imghasil_crop)
        self.labe3hasil_crop.image_names = imghasil_crop

def main():
    root = tk.Tk()
    root.geometry('1024x500+50+50')
    root.configure(background= 'white smoke')
    app = aplikasiku(root)
    app.parent.title("Pengenaln Ekspresi Wajah")
    root.mainloop()

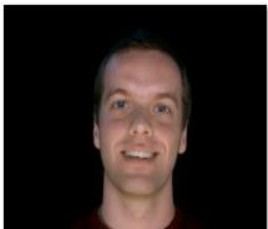
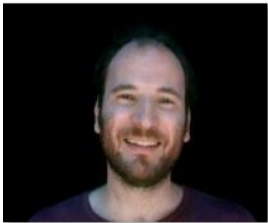
if __name__=='__main__':
    main()

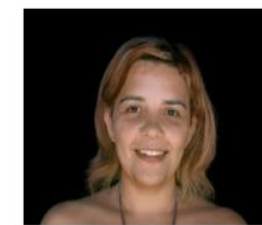
```

Lampiran 2 : DATA

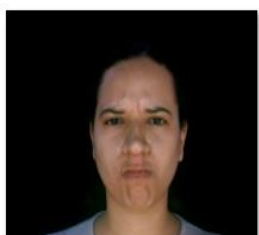
108 Data Training :

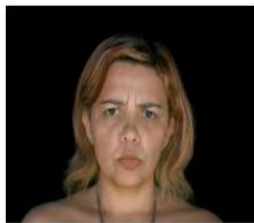
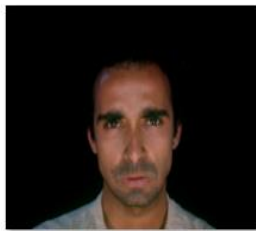
SENYUM



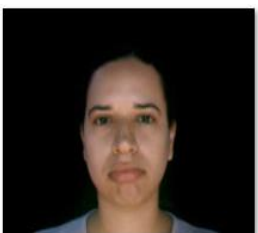
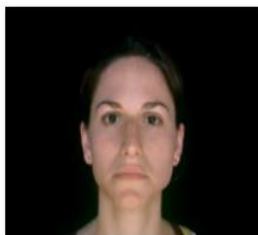
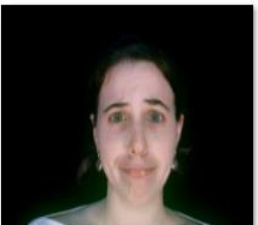
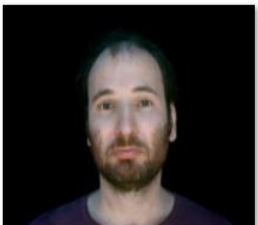


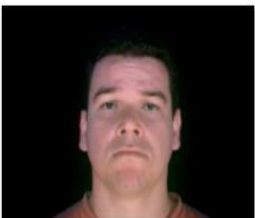
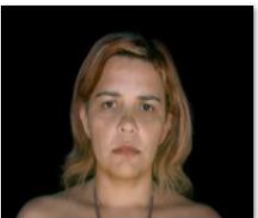
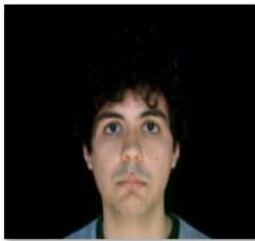
MARAH



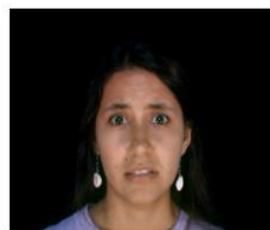
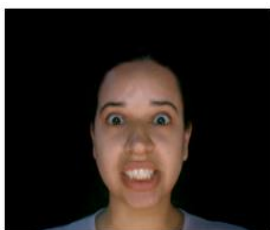
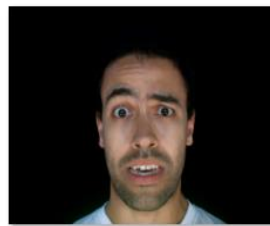
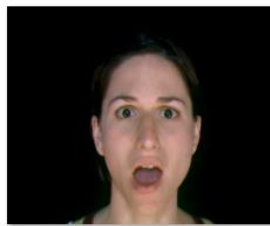
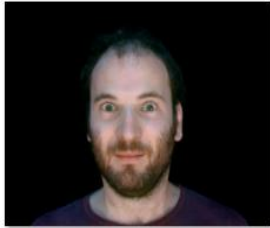


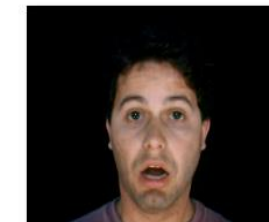
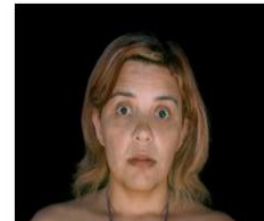
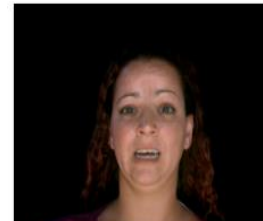
SEDIH





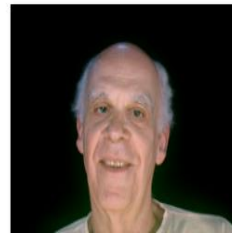
TAKUT



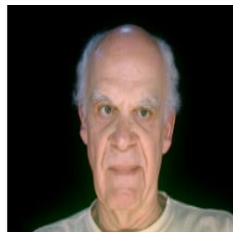
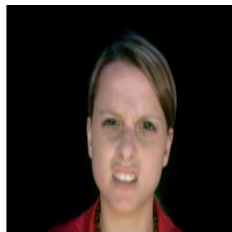


DATA TESTING

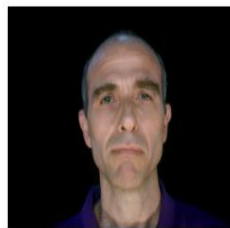
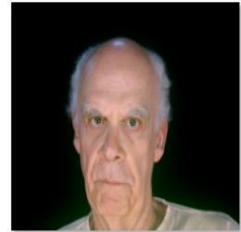
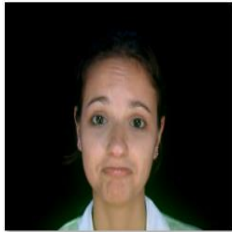
SENYUM



MARAH



SEDIH



TAKUT



Lampiran 3: RIWAYAT HIDUP



Nama : Nuraini Ahmad
Nim : T3116159
Tempat, Tanggal Lahir : Gorontalo, 16 July 1996
Email : nurain181018@gmail.com

Riwayat pendidikan :

1. Tahun 2008, menyelesaikan Pendidikan di Sekolah Dasar Negeri 32 Kota Selatan, Gorontalo
2. Tahun 2011, menyelesaikan Pendidikan di Sekolah Menengah Pertama Negeri 1 Gorontalo
3. Tahun 2014, menyelesaikan Pendidikan di Sekolah Menengah Atas Negeri 1 Gorontalo.
4. Tahun 2016, telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo.



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
LEMBAGA PENELITIAN (LEMLIT)
UNIVERSITAS ICHSAN GORONTALO

Jl. Raden Saleh No. 17 Kota Gorontalo
Telp: (0435) 8724466, 829975, Fax: (0435) 82997,
E-mail: lembagapenelitian@unicon.ac.id

Nomor : 1938/PIP/LEMLIT-UNISAN/GTO/XII/2019

Lampiran : -

Hal : Permohonan Izin Penelitian

Kepada Yth,

Dekan Fakultas Ilmu komputer Universitas Ichsan Gorontalo
di,
Gorontalo

Yang bertanda tangan di bawah ini :

Nama : Dr. Rahmisyari, ST., SE
NIDN : 0929117202
Jabatan : Ketua Lembaga Penelitian

Meminta kesediannya untuk memberikan izin pengambilan data dalam rangka penyusunan **Proposal / Skripsi**, kepada :

Nama Mahasiswa : Nuraini Ahmad
NIM : T3116159
Fakultas : Fakultas Ilmu Komputer
Program Studi : Teknik Informatika
Lokasi Penelitian : FAKULTAS ILMU KOMPUTER UNIVERSITAS ICHSAN GORONTALO
Judul Penelitian : PENERAPAN METODE GLCM (GRAY LEVEL CO-OCCURENCE MATRIX) DAN KNN (K-NEAREST NEIGHBOR) UNTUK PENGENALAN EKSPRESI WAJAH BERDASARKAN BENTUK BIBIR

Atas kebijakan dan kerja samanya diucapkan banyak terima kasih.

Gorontalo, 03 Desember 2019

Ketua


Dr. Rahmisyari, ST., SE
NIDN 0929117202

JADWAL PENELITIAN

[illegible]



Pustikom
Universitas Ichsan Gorontalo

BUKTI PENERIMAAN SOFTCOPY SKRIPSI
PENGECEKAN SIMILARITY TURNITIN

Nama Mahasiswa : NURAINI AHMAD
NIM : T3116159
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Penerapan Metode GLCM (GRAY LEVEL CO-OCCURRENCE MATRIX) Untuk Reduksi Fitur Pada Pengenalan Ekspresi Wajah

Nama File (Pdf) : SKRIPSI - T3116159 - Nuraini Ahmad - Penerapan - 2020

No. HP/WA : 081241407605

e-Mail : Nuraini8101@gmail.com

Tgl. Terima :

Hasil Pengecekan :

2	2	0	6	2	0
35%					

Diterima/Diperiksa Oleh,


Sudirman S. Panna, M.Kom
085340910769



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN
(UNISAN) GORONTALO

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

- | | |
|---------|-------------------------|
| 1. Nama | : Haditsah Annur, M.Kom |
| Sebagai | : Pembimbing I |
| 2. Nama | : Rofiq Harun, M.Kom |
| Sebagai | : Pembimbing II |

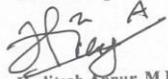
Dengan ini Menyatakan bahwa :

Nama Mahasiswa	: NURAINI AHMAD
NIM	: T3116159
Program Studi	: Teknik Informatika (S1)
Fakultas	: Fakultas Ilmu Komputer
Judul Skripsi	: Penerapan Metode GLCM (GRAY LEVEL CO-OCCURRENCE MATRIX) Untuk Reduksi Fitur Pada Pengenalan Ekspresi Wajah

Setelah kami melakukan pengecekan kembali antara softcopy skripsi dari hasil pemeriksaan aplikasi Turnitin dengan hasil Similarity sebesar 35% oleh Tim Verifikasi Plagiasi di Pustikom dengan Skripsi Aslinya, isinya SAMA dan format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi.

Demikian surat pernyataan ini dibuat untuk mendapatkan Surat Rekomendasi Bebas Plagiasi.

Pembimbing I

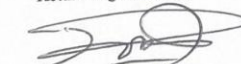

Haditsah Annur, M.Kom
NIDN. 0908058403

Gorontalo, Juni 2020

Pembimbing II


Rofiq Harun, M.Kom
NIDN. 0919048404

Mengetahui
Ketua Program Studi,


Irvan A. Salibi, M.Kom
NIDN. 0928028101

Catatan Perbaikan :

- ☐ Penggunaan tanda petik dua tidak wajar
- ☐ Penulisan Rumus masih berbentuk gambar
- ☐ Beberapa Paragraf berbentuk gambar
- ☐ Beberapa kata tidak lengkap hurufnya / beberapa kata digabung tanpa spasi
- ☐ _____



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN
(UNISAN) GORONTALO

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI

No. 0258/UNISAN-G/S-BP/IV/2020

Yang bertanda tangan di bawah ini:

Nama : Sunarto Taliki, M.Kom
NIDN : 0906058301
Unit Kerja : Pustikom, Universitas Ichsan Gorontalo

Dengan ini Menyatakan bahwa :

Nama Mahasisw : NURAINI AHMAD
NIM : T3116159
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Penerapan Metode GLCM (GRAY LEVEL CO-OCCURRENCE MATRIX) Untuk Reduksi Fitur Pada Pengenalan Ekspresi Wajah

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 35%, berdasarkan SK Rektor No. 237/UNISAN-G/SK/IX/2019 tentang Panduan Pencegahan dan Penanggulangan Plagiarisme, bahwa batas kemiripan skripsi maksimal 35% dan sesuai dengan Surat Pernyataan dari kedua Pembimbing yang bersangkutan menyatakan bahwa isi softcopy skripsi yang diolah di Turnitin SAMA ISINYA dengan Skripsi Aslinya serta format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi, untuk itu skripsi tersebut di atas dinyatakan BEBAS PLAGIASI dan layak untuk diujikan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 23 Juni 2020

Tim Verifikasi,



Sunarto Taliki, M.Kom
NIDN. 0906058301

Tembusan :

1. Dekan
2. Ketua Program Studi
3. Pembimbing I dan Pembimbing II
4. Yang bersangkutan
5. Arsip



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS ICHSAN GORONTALO**

UPT. PERPUSTAKAAN PUSAT

SURAT KEPUTUSAN MENDIKS RI NO. 84/D/0/2001

Jln. Achmad Nadjamuddin No.17 Telp. (0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA

No.001/perpus_fikom/VII/2020

PerpustakaanRabu, 01 Juli 2020 Fakultas Ilmu komputer (FIKOM) Universitas Ichsan
Gorontalo dengan ini menerangkan bahwa :

Nama : Nuraini Ahmad

Nim : T3116159

No anggota : M201963

Terhitung sejak tanggal 01 Juli 2020, dinyatakan telah bebas dari pinjaman buku dan koleksi
lainnya dipergustakaan Fakultas Ilmu komputer.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.

Gorontalo, 01 Juli 2020
Kepala Perpustakaan
Fakultas Ilmu Komputer

Apriyanto Alhamad, M.Kom
NIDN 09240486

PENERAPAN METODE GLCM (GRAY LEVEL CO-OCCURRENCE MATRIX) UNTUK REDUKSI FITUR PADA PENGENALAN EKSPRESI WAJAH

ORIGINALITY REPORT

35%

SIMILARITY INDEX

33%

INTERNET SOURCES

6%

PUBLICATIONS

25%

STUDENT PAPERS

PRIMARY SOURCES



Submitted to Universitas Brawijaya

Student Paper

4%



media.neliti.com

Internet Source

4%



www.unpak.ac.id

Internet Source

3%



docplayer.info

Internet Source

3%



repository.upi.edu

Internet Source

2%



j-ptiik.ub.ac.id

Internet Source

2%



es.scribd.com

Internet Source

2%



docobook.com

Internet Source

2%



jurnal.darmajaya.ac.id
Internet Source

2%



repository.uinjkt.ac.id
Internet Source

1%



e-journal.uajy.ac.id
Internet Source

1%



jurnaleeccis.ub.ac.id
Internet Source

1%



voi.stmik-tasikmalaya.ac.id
Internet Source

1%



lecturer.eepis-its.edu
Internet Source

1%



pelita-informatika.com
Internet Source

1%



www.ejurnal.stmik-budidarma.ac.id
Internet Source

1%



rizkaprnmrsr.blogspot.com
Internet Source

1%



Submitted to Universitas Diponegoro
Student Paper

1%



ejournal.catursakti.ac.id
Internet Source

<1%



digilib.unila.ac.id
Internet Source

		<1 %
22	id.123dok.com Internet Source	<1 %
22	text-id.123dok.com Internet Source	<1 %
23	arsip.e-jmii.org Internet Source	<1 %
24	repository.usu.ac.id Internet Source	<1 %
25	ejurnal.its.ac.id Internet Source	<1 %
26	repository.unhas.ac.id Internet Source	<1 %
27	uir.unisa.ac.za Internet Source	<1 %
28	repository.its.ac.id Internet Source	<1 %
29	eprints.akakom.ac.id Internet Source	<1 %
Exclude quotes On Exclude matches < 25 words		