



**PENINGKATAN AKURASI PENGENALAN WAJAH
PRINCIPAL COMPONENT ANALYSIS BERBASIS *MULTI-
LEVEL REVERSE BIORTHOGONAL WAVELET
TRANSFORM* DENGAN *K-NEAREST NEIGHBOR***

Skripsi

disusun sebagai salah satu syarat
untuk memperoleh gelar Sarjana Komputer
Program Studi Teknik Informatika

oleh

Iin Dinariyah
4611416029

**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN PENGETAHUAN ALAM
UNIVERSITAS NEGERI SEMARANG
2019**

PERNYATAAN

Saya menyatakan dengan sebenar-benarnya bahwa skripsi saya yang berjudul “Peningkatan Akurasi Pengenalan Wajah *Principal Component Analysis* Berbasis *Multi-level Reverse Biorthogonal Wavelet Transform* dengan *K-Nearest Neighbor*” disusun berdasarkan penelitian saya dengan arahan dosen pembimbing. Sumber informasi atau kutipan yang berasal dari karya yang diterbitkan telah disebutkan dalam teks dan dicantumkan dalam daftar pustaka di bagian akhir skripsi ini, dan saya menyatakan bahwa skripsi ini bebas plagiat, dan apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.

Semarang, Januari 2020



Iin Dinariyah
4611416029

PERSETUJUAN PEMBIMBING

Nama : Iin Dinariyah

NIM : 4611416029

Program Studi : S-1 Teknik Informatika

Judul Skripsi : Peningkatan Akurasi Pengenalan Wajah *Principal Component Analysis* Berbasis *Multi-level Reverse Biorthogonal Wavelet Transform* dengan *K-Nearest Neighbor*

Skripsi ini telah disetujui oleh pembimbing untuk diajukan ke sidang panitia ujian skripsi Program Studi Teknik Informatika FMIPA UNNES.

Semarang, 27 Januari 2020

Pembimbing



Dr. Alamsyah, S.Si., M.Kom.
NIP. 197405172006041001

PENGESAHAN

Skripsi yang berjudul

Peningkatan Akurasi Pengenalan Wajah *Principal Component Analysis*
Berbasis *Multi-level Reverse Biorthogonal Wavelet Transform* dengan *K-Nearest Neighbor*


disusun oleh

Iin Dinariyah

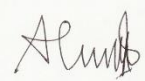
4611416029


Telah dipertahankan dihadapan sidang Panitia Ujian Skripsi FMIPA UNNES pada
tanggal 14 Januari 2020





Riza Arifudin, S.Pd., M.Cs.
NIP.198005252005011001
Anggota Penguji

Sekretaris


Dr. Alamsyah S.Si., M.Kom.
NIP. 197405172006041001
Penguji 2


Zaenal Abidin, S.Si., M.Cs., Ph.D
NIP. 198205042005011001


Dr. Alamsyah S.Si., M.Kom.
NIP.197405172006041001

MOTTO DAN PERSEMBAHAN

MOTTO

- Prestasi tanpa kontribusi adalah nol besar.
- Nelayan tangguh tidak dilahirkan dari lautan yang tenang.
- Mutiara lahir dari lautan yang keruh.

PERSEMBAHAN

Skripsi ini ku persembahkan kepada:

- Kedua orang tua yang saya sayangi dan cintai, Saya ucapkan terimakasih atas do'a, dukungan, semangat dan kasih sayang yang tiada hentinya untuk saya.
- Kakak dan Adik tersayang, Asti Aryanti, Yesi Astiani dan Nungki Pratiwi yang selalu memberikan semangat serta do'anya untuk saya.
- UKM RIPTEK dan UKM PENELITIAN yang telah menjadi pijakan untuk berkembang
- Seluruh punggawa Robotika UNNES-Alim, Tino, Oky, Deni, Rizky, Maxi, Minhajul, Anggit dan teman-teman lain.
- *STD Parents*-Adel, Fahri, Rois, Danang, dan Novita yang selalu mendengarkan keluh kesah
- Teman Jogja-Rahmad Prasetyo, Yudistiro Septian Dwi Saputro, Ridho Prasakti, & Rosemery Fams
- Semua teman-teman seperjuangan di jurusan Ilmu Komputer angkatan 2016.
- Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah membantu hingga terselesaikannya penulisan skripsi ini.
- Almamaterku UNNES.

PRAKATA

Puji syukur Alhamdulillah senantiasa penulis panjatkan kehadiran Tuhan Yang Maha Esa atas segala limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “**Peningkatan Akurasi Pengenalan Wajah *Principal Component Analysis* Berbasis *Multi-level Reverse Biorthogonal Wavelet Transform* dengan *K-Nearest Neighbor*”.**

Penulis menyadari bahwa dalam penulisan skripsi ini tidak akan selesai tanpa adanya dukungan, bantuan, serta motivasi dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada:

1. Prof. Dr. Fathur Rokhman. M.Hum., Rektor Universitas Negeri Semarang.
2. Dr. Sugianto, M. Si., Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam.
3. Dr. Alamsyah, S.Si., M.Kom., Selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Negeri Semarang sekaligus Dosen Pembimbing yang telah meluangkan waktu, membantu, membimbing, mengarahkan dan memberikan saran sehingga penulis dapat menyelesaikan skripsi ini.
4. Riza Arifudin, S.Pd., M.Cs., Selaku Dosen Penguji I yang telah memberikan banyak masukan, arahan, kritik dan saran dalam penyelesaian skripsi ini.
5. Zaenal Abidin S.Si., M.Cs., Ph.D., Selaku Dosen Penguji II yang telah memberikan banyak masukan, arahan, kritik dan saran dalam penyelesaian skripsi ini.

6. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan bekal kepada penulis dalam penyusunan skripsi ini.
7. Bapak dan Ibu tercinta, serta kakak dan adik tersayang, Asti Aryanti, Yesi Astiani, dan Nungki Pratiwi yang selalu memberikan do'a serta memberikan dukungan baik secara moral maupun spiritual dalam menyelesaikan skripsi ini.
8. UKM RIPTEK dan UKM PENELITIAN yang telah menjadi rumah kedua.
9. Seluruh punggawa Robotika UNNES (Alim, Tino, Oky, Deni, Rizky, Minhajul, Anggit dan teman-teman lain)
10. STD *Parents* (Adel, Fahri, Rois, Danang, Novita) yang selalu mendengarkan keluh kesah.
11. Teman Jogja (Rahmad Prasetyo, Yudhistiro Septian Dwi Saputro, Ridho Prasakti, & Rosemery Fams)
12. Teman-teman seperjuangan di Jurusan Ilmu Komputer angkatan 2016.
13. Semua pihak yang telah membantu dalam terselesaikannya skripsi ini yang tidak dapat penulis sebutkan satu persatu, terimakasih atas bantuannya.

Semoga skripsi ini dapat memberikan manfaat bagi pembaca demi kebaikan di masa yang akan datang.

Semarang, Januari 2020



Lin Dinariyah
4611416029

ABSTRAK

Dinariyah, I. 2020. Peningkatan Akurasi Pengenalan Wajah *Principal Component Analysis* Berbasis *Multi-level Reverse Biorthogonal Wavelet Transform* dengan *K-Nearest Neighbor*. Skripsi. Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Dr. Alamsyah, S.Si., M.Kom

Kata kunci: 2DPCA, *Computer Vision*, *Face Recognition*, PCA, KNN, *Wavelet*

Pengenalan wajah merupakan suatu teknik biometrik fisik yang memungkinkan komputer untuk mengenali wajah manusia. Saat ini, pengenalan wajah merupakan metode yang banyak digunakan dalam autentifikasi sistem keamanan. Akurasi merupakan hal yang sangat penting dalam pengenalan wajah. Peningkatan akurasi pengenalan wajah dapat dilakukan dengan melakukan dekomposisi citra menggunakan *multi-level reverse biorthogonal wavelet*. Citra hasil dikomposisi akan diekstraksi cirinya menggunakan 1D-PCA dan 2D-PCA. Klasifikasi citra dilakukan dengan *K-Nearest Neighbor*. Selanjutnya hasil akurasi antara pengenalan citra dengan maupun tanpa dekomposisi *wavelet* akan dibandingkan. Data yang digunakan adalah data sekunder yakni AT&T *face of database* sebanyak 400 citra dan YALE *database* sebanyak 165 citra sehingga total data yang digunakan adalah 565 citra. Hasil akurasi tertinggi pada percobaan menggunakan AT & T adalah 97,5 %, sedangkan pada YALE mencapai 100%.

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN SKRIPSI.....	ii
PERSETUJUAN PEMBIMBING.....	iii
HALAMAN PENGESAHAN.....	iv
MOTTO DAN PERSEMBAHAN	v
PRAKATA	vi
ABSTRAK	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR	xvi
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Batasan Masalah.....	7
1.4 Tujuan Penelitian.....	8
1.5 Manfaat Penelitian.....	8
1.6 Sistematika Penelitian	8
BAB II TINJAUAN PUSTAKA.....	10
2.1 Pengolahan Citra Digital	10
2.2 Pengenalan Wajah.....	14

2.3	<i>Wavelet</i>	16
2.3.1	Transformasi <i>Wavelet</i> Diskret	20
2.3.1.1.	<i>Reverse Biorthogonal Wavelet</i>	23
2.4	<i>Principal Component Analysis (PCA)</i>	24
2.4.1	1D-PCA	25
2.4.2	2D-PCA	26
2.5	<i>K-Nearest Neighbor (KNN)</i>	28
2.6	<i>Confussion Matrix</i>	31
2.7	Penelitian Terkait	32
BAB III METODE PENELITIAN.....		33
3.1	Alat & Bahan.....	33
3.1.1	Alat.....	33
3.1.2	Bahan	33
3.2	Metode yang Diusulkan	35
3.2.1	Pembagian Data	38
3.2.2	Dekomposisi Citra dengan <i>Reverse Wavelet Transform</i>	38
3.2.3	Ekstraksi Fitur dengan PCA	40
3.2.3.1.	1D-PCA	40
3.2.3.1	2D-PCA	40
3.3	Klasifikasi dengan KNN	41
3.4	Perhitungan Akurasi dengan <i>Confussion Matrix</i>	41
3.5	Perancangan Sistem.....	42
3.6	Analisis Hasil	43

3.7	Penarikan Kesimpulan.....	43
BAB IV HASIL DAN PEMBAHASAN		44
4.1	Hasil Penelitian	44
4.1.1	Tahap Implementasi Sistem.....	44
4.1.2	Tahap Pengujian Sistem.....	53
4.1.2.1	Pengujian Menggunakan AT & T.....	58
4.1.2.2	Pengujian Menggunakan YALE.....	71
4.2	Pembahasan.....	91
BAB V PENUTUP.....		92
5.1	Simpulan.....	92
5.2	Saran.....	93
DAFTAR PUSTAKA		95

DAFTAR TABEL

Tabel	Halaman
2.1 <i>Confussion Matrix</i> untuk Klasifikasi Dua Kelas	31
2.2 Penelitian Terkait.....	32
3.1 Pembagian <i>data training</i> dan <i>data testing</i> AT&T	38
3.2 Pembagian <i>data training</i> dan <i>data testing</i> YALE	38
4.1 Hasil Percobaan 10% <i>Training Images</i> dan 90% <i>Testing Images</i> dengan 1D-PCA+KNN	59
4.2 Hasil Perhitungan 10% <i>Training Images</i> dan 90% <i>Testing Images</i> dengan 1D-PCA+KNN	59
4.3 Hasil Percobaan 10% <i>Training Images</i> dan 90% <i>Testing Images</i> dengan <i>Wavelet</i> +1D-PCA+KNN	60
4.4 Hasil Perhitungan 10% <i>Training Images</i> dan 90% <i>Testing Images</i> dengan <i>Wavelet</i> +1D-PCA+KNN.....	60
4.5 Hasil Percobaan 10% <i>Training Images</i> dan 90% <i>Testing Images</i> dengan 2D-PCA+KNN	61
4.6 Hasil Pehitungan 10% <i>Training Images</i> dan 90% <i>Testing Images</i> dengan 2D-PCA+KNN	61
4.7 Hasil Percobaan 10% <i>Training Images</i> dan 90% <i>Testing Images</i> dengan <i>Wavelet</i> +2D-PCA+KNN	62
4.8 Hasil Perhitungan 10% <i>Training Images</i> dan 90% <i>Testing Images</i> dengan <i>Wavelet</i> +2D-PCA+KNN.....	62

4.9 Hasil Percobaan 50% Training Images dan 50% Testing	
Images dengan 1D-PCA+KNN	63
4.10 Hasil Perhitungan 50% <i>Training Images</i> dan 50% <i>Testing</i>	
<i>Images</i> dengan 1D-PCA+KNN	64
4.11 Hasil Percobaan 50% <i>Training Images</i> dan 50% <i>Testing</i>	
<i>Images</i> dengan <i>Wavelet</i> +1D-PCA+KNN	64
4.12 Hasil Perhitungan 50% <i>Training Images</i> dan 50% <i>Testing</i>	
<i>Images</i> dengan <i>Wavelet</i> +1D-PCA+KNN	65
4.13 Hasil Percobaan 50% <i>Training Images</i> dan 50% <i>Testing</i>	
<i>Images</i> dengan 2D-PCA+KNN	65
4.14 Hasil Perhitungan 50% <i>Training Images</i> dan 50% <i>Testing</i>	
<i>Images</i> dengan 2D-PCA+KNN	66
4.15 Hasil Percobaan 50% <i>Training Images</i> dan 50% <i>Testing</i>	
<i>Images</i> dengan <i>Wavelet</i> +2D-PCA+KNN	66
4.16 Hasil Perhitungan 50% <i>Training Images</i> dan 50% <i>Testing</i>	
<i>Images</i> dengan <i>Wavelet</i> +2D-PCA+KNN	67
4.17 Hasil Percobaan 90% <i>Training Images</i> dan 10% <i>Testing</i>	
<i>Images</i> dengan 1D-PCA+KNN	68
4.18 Hasil Percobaan 90% <i>Training Images</i> dan 10% <i>Testing</i>	
<i>Images</i> dengan 1D-PCA+KNN	68
4.19 Hasil Percobaan 90% <i>Training Images</i> dan 10% <i>Testing</i>	
<i>Images</i> dengan DWT-PCA+KNN	69

4.20 Hasil Perhitungan 90% <i>Training Images</i> dan 10% <i>Testing Images</i> dengan <i>Wavelet+1D-PCA+KNN</i>	69
4.21 Hasil Percobaan 90% <i>Training Images</i> dan 10% <i>Testing Images</i> dengan <i>2D-PCA+KNN</i>	70
4.22 Hasil Perhitungan 90% <i>Training Images</i> dan 10% <i>Testing Images</i> dengan <i>2D-PCA+KNN</i>	70
4.23 Hasil Percobaan 90% <i>Training Images</i> dan 10% <i>Testing Images</i> dengan <i>Wavelet-2DPCA+KNN</i>	71
4.24 Hasil Perhitungan 90% <i>Training Images</i> dan 10% <i>Testing Images</i> dengan <i>Wavelet-2DPCA+KNN</i>	71
4.25 Hasil Percobaan 1 <i>Training Images</i> dan 10 <i>Testing Images</i> dengan <i>1D-PCA+KNN</i>	72
4.26 Hasil Perhitungan 1 <i>Training Images</i> dan 10 <i>Testing Images</i> dengan <i>1D-PCA PCA+KNN</i>	72
4.27 Hasil Percobaan 1 <i>Training Images</i> dan 10 <i>Testing Images</i> dengan <i>Wavelet+1D-PCA+KNN</i>	73
4.28 Hasil Perhitungan 1 <i>Training Images</i> dan 10 <i>Testing Images</i> dengan <i>Wavelet+1D-PCA+KNN</i>	73
4.29 Hasil Percobaan 1 <i>Training Images</i> dan 10 <i>Testing Images</i> dengan <i>2D-PCA+KNN</i>	74
4.30 Hasil Perhitungan 1 <i>Training Images</i> dan 10 <i>Testing Images</i> dengan <i>2D-PCA+KNN</i>	74

4.31 Hasil Percobaan 1 <i>Training Images</i> dan 10 <i>Testing Images</i> dengan <i>Wavelet+2DPCA+KNN</i>	75
4.32 Hasil Perhitungan 1 <i>Training Images</i> dan 10 <i>Testing Images</i> dengan <i>Wavelet+2DPCA+KNN</i>	75
4.33 Hasil Percobaan 6 <i>Training Images</i> dan 5 <i>Testing Images</i> dengan 1D-PCA+KNN.....	76
4.34 Hasil Perhitungan 6 <i>Training Images</i> dan 5 <i>Testing Images</i> dengan 1D-PCA+KNN.....	77
4.35 Hasil Percobaan 6 <i>Training Images</i> dan 5 <i>Testing Images</i> dengan <i>Wavelet+1D-PCA+KNN</i>	77
4.36 Hasil Perhitungan 6 <i>Training Images</i> dan 5 <i>Testing Images</i> dengan <i>Wavelet+1D-PCA+KNN</i>	78
4.37 Hasil Percobaan 6 <i>Training Images</i> dan 5 <i>Testing Images</i> dengan 2D-PCA+KNN.....	78
4.38 Hasil Perhitungan 6 <i>Training Images</i> dan 5 <i>Testing Images</i> dengan 2D-PCA+KNN	79
4.39 Hasil Percobaan 6 <i>Training Images</i> dan 5 <i>Testing Images</i> dengan <i>Wavelet+2D-PCA+KNN</i>	79
4.40 Hasil Perhitungan 6 <i>Training Images</i> dan 5 <i>Testing Images</i> dengan <i>Wavelet+2D-PCA+KNN</i>	80
4.41 Hasil Percobaan 90% <i>Training Images</i> dan 10% <i>Testing Images</i> dengan 1D-PCA+KNN	80

4.42 Hasil Perhitungan 90% <i>Training Images</i> dan 10% <i>Testing Images</i> dengan 1D-PCA+KNN	81
4.43 Hasil Percobaan 90% <i>Training Images</i> dan 10% <i>Testing Images</i> dengan <i>Wavelet</i> +1D-PCA+KNN	81
4.44 Hasil Perhitungan 90% <i>Training Images</i> dan 10% <i>Testing Images</i> dengan <i>Wavelet</i> +1D-PCA+KNN	82
4.45 Hasil Percobaan 10 <i>Training Images</i> dan 1 <i>Testing Images</i> dengan 2D-PCA+KNN	82
4.46 Hasil Percobaan 10 <i>Training Images</i> dan 1 <i>Testing Images</i> dengan 2D-PCA+KNN	83
4.47 Hasil Percobaan 10 <i>Training Images</i> dan 1 <i>Testing Images</i> dengan <i>Wavelet</i> +2D-PCA+KNN	83
4.48 Hasil Percobaan 10 <i>Training Images</i> dan 1 <i>Testing Images</i> dengan <i>Wavelet</i> +2D-PCA+KNN	84
4.49 Rekapitulasi Citra Dominan <i>Error</i>	85

4.50 Rekapitulasi Hasil Percobaan

86

DAFTAR GAMBAR

Gambar	Halaman
2.1 Sistem Koordinat Citra Digital	10
2.2 Representasi citra sebagai $f(x,y)$ (kiri); citra kiri yang telah diperbesar (kanan).....	11
2.3 Proses Pengenalan Wajah.....	16
2.4 Transformasi <i>Wavelet</i> 2 Dimensi.....	19
2.5 Dekomposisi <i>Wavelet</i> 3 Tingkat.....	22
2.6 Operasi <i>Reverse Biorthogonal Wavelet</i>	26
3.1 <i>Sample</i> wajah AT&T “ <i>The Database of Face</i> ”	34
3.2 <i>Sample</i> wajah <i>YALE database face</i>	35
3.3 <i>Flowchart Sistem</i> (Menggunakan 1D-PCA).....	36
3.4 <i>Flowchart Sistem</i> (Menggunakan 2D-PCA).....	37
3.5 Operasi <i>Wavelet</i> 2 Level.....	39
4.1 Tampilan Menu Utama.....	45
4.2 Tampilan Menu Metode	46
4.3 <i>Pseudocode</i> Ekspor Citra Asli AT & T	47
4.4 <i>Pseudocode</i> Ekspor Citra Hasil Dekomposisi AT & T	48
4.5 <i>Pseudocode</i> Ekspor Citra Asli YALE	48
4.6 <i>Pseudocode</i> Ekspor Citra Dekomposisi YALE.....	49
4.7 <i>Pseudocode</i> Pemisahan <i>Training Images</i> dan <i>Testing Images</i>	50
4.8 <i>Pseudocode</i> Perhitungan <i>Eigenvectors</i>	50
4.9 <i>Pseudocode</i> Kode Proses Klasifikasi.....	51

4.10	Tampilan Pengujian.....	51
4.11	<i>Pseudocode</i> Perhitungan Akurasi.....	52
4.12	Tampilan Akurasi Sistem	52
4.13	Diagram Alur Kasus 1	54
4.14	Pengaturan Percobaan 10% <i>Training Images</i> dan 90% <i>Testing Images</i>	58
4.15	Pengaturan Percobaan 50% <i>Training Images</i> dan 50% <i>Testing Images</i>	63
4.16	Pengaturan Percobaan 10% <i>Training Images</i> dan 90% <i>Testing Images</i>	67
4.17	Pengaturan Percobaan 1 <i>Training Images</i> dan 10 <i>Testing Images</i>	72
4.18	Pengaturan Percobaan 6 <i>Training Images</i> dan 5 <i>Testing Images</i>	76
4.19	Pengaturan Percobaan 10 <i>Training Images</i> dan 1 <i>Testing Images</i>	81
4.20	Grafik Perbandingan Akurasi Percobaan Menggunakan AT & T.....	87
4.21	Grafik Perbandingan Akurasi Percobaan Menggunakan YALE.....	87

DAFTAR LAMPIRAN

Lampiran	Halaman
1 <i>Source Code</i>	99
2 Hasil Percobaan Menggunakan AT & T	113
3 Hasil Percobaan Menggunakan YALE.....	127

BAB I

PENDAHULUAN

1.1. Latar Belakang

Wajah merupakan bagian tubuh manusia yang menjadi fokus perhatian dalam interaksi sosial. Wajah memiliki peranan vital untuk menunjukkan identitas manusia dan emosi yang sedang ditunjukkan. Setiap manusia memiliki garis lengkung dan karakteristik wajah yang berbeda. Oleh karena itu, wajah dapat digunakan sebagai organ tubuh manusia yang digunakan untuk indikasi pengenalan seseorang (Marti, 2010: 11). Hal tersebut dikenal dengan istilah pengenalan wajah atau *face recognition*.

Face recognition merupakan salah satu metode yang banyak digunakan dalam autentikasi sistem keamanan. Dalam penggunaannya, metode *face recognition* memerlukan identifikasi yang unik yakni identifikasi geometri wajah. Keuntungan yang dimiliki dari sistem pengamanan berbasis pengenalan wajah adalah kemampuan pengamanannya yang relatif sulit untuk ditembus (Soelaiman, 2006: 12). Selain itu *face recognition* juga lebih cepat dari metode *biometrika biologis* seperti tanda tangan DNA (Yanuskevich, 2005: 15)

Pengenalan wajah merupakan teknik biometrik fisik yang memungkinkan komputer untuk mengenali wajah manusia. Pengenalan wajah melalui komputer mengadopsi kemampuan *learning* manusia. Dengan pendekatan konsep *learning*, program komputer dapat mempelajari karakteristik wajah yang berbeda kemudian

menginterpretasikannya dalam bentuk pengetahuan. Pengetahuan ini kemudian digunakan untuk membuat sistem pengenalan wajah yang memiliki kemampuan seperti manusia. Secara umum, pengenalan wajah memiliki tiga tahap yakni deteksi wajah, ekstraksi fitur, dan klasifikasi (Soni & Sahu, 2013: 3899).

Sebelum suatu sistem melakukan pengenalan wajah, sistem harus melakukan pendeteksian wajah. Deteksi wajah adalah suatu langkah dalam *face recognition* yang dilakukan untuk menemukan posisi wajah dari sebuah citra yang akan di ekstraksi selanjutnya. Deteksi wajah bertujuan untuk membuang fitur-fitur yang tidak diperlukan saat pencarian pola wajah dan mampu membuang daerah pada gambar yang tidak memiliki pola gambar wajah. Keberhasilan pengenalan wajah bergantung pada citra wajah yang dijadikan masukan dalam basis data yang diproses dari suatu proses pendeteksian wajah (Firdausy, 2013: 34). Proses deteksi wajah akan dilakukan bersama dalam proses ekstraksi fitur karena dalam proses deteksi wajah diperlukan ekstraksi fitur di dalam citra masukan yang akan mendeteksi apakah citra masukan merupakan citra yang mengandung ciri wajah.

Ekstraksi fitur merupakan tahapan untuk menentukan karakteristik alami dari suatu wajah yang selanjutnya akan diklasifikasi atau dikenali. Ekstraksi fitur dapat diteliti dengan mengambil beberapa bagian citra yang bisa menunjukkan fitur utama wajah. Hal terpenting dalam *face recognition* adalah ekstraksi semua informasi yang relevan pada citra wajah (Budi *et al*, 2016: 167). Ciri hasil ekstraksi kemudian digunakan sebagai parameter/nilai masukan untuk membedakan antara citra satu dengan lainnya pada tahapan klasifikasi.

Terdapat beberapa algoritma ekstraksi fitur, salah satunya algoritma *Principal Component Analysis* (PCA). Algoritma PCA merupakan algoritma yang sangat populer dalam ekstraksi fitur. Algoritma PCA dapat digunakan untuk mengolah citra wajah seseorang sehingga secara otomatis sistem akan mengenali wajah seseorang melalui ciri-ciri utamanya seperti mata, hidung, bibir, alis sebagai identitas. Identitas dari citra wajah seseorang tersebut oleh sistem akan dikenali melalui berbagai pelatihan (*training images*) yang disimpan di *database*. PCA adalah metode yang baik untuk melakukan analisa data. Algoritma PCA juga memiliki kelemahan yakni bersifat skalar dan membutuhkan data training dan testing yang besar agar mendapatkan akurasi yang akurat (Purnama & Salsabila, 2013: 1). Hal ini dikarenakan dalam proses ekstraksi fitur menggunakan PCA terdapat tahapan dimana citra yang semula dua dimensi diubah ke citra berdimensi satu. Adanya transformasi dimensi pada citra menyebabkan ruang vektor dimensi citra menjadi besar, dimana akan sulit untuk dapat mengevaluasi matriks kovarian yang besar dengan akurat apabila hanya dengan menggunakan data sampel yang kecil.

Discrete Wavelet Transform (DWT) merupakan teknik dekomposisi multi level lokalisasi fitur yang bermanfaat untuk mempermudah dalam kompresi, dan analisis citra. DWT dilakukan dengan menerapkan konvolusi *low-pass filter* (H) untuk mendapatkan sinyal global dan *high-pass filter* (G) untuk mendapatkan sinyal detil. Proses dekomposisi ini akan mengurangi ukuran *eigenface* citra tanpa mengurangi informasi-informasi penting yang dimiliki oleh citra.

Menurut penelitian Muhedkar & Powalkar (2015: 765) penerapan dekomposisi *daubechies wavelet transform* sebanyak 3 level pada citra sebelum dilakukan ekstraksi fitur menggunakan PCA dalam meningkatkan akurasi pengenalan wajah rata-rata sebesar 1% menggunakan dekomposisi *wavelet Daubechies2* sebanyak 3 level dan 3% menggunakan dekomposisi *wavelet Daubechies1* sebanyak 3 level pada 4 *dataset* yakni *face94*, *face95*, *face96* dan *face self*. *Daubhecies wavelet transform* merupakan jenis *orthogonal wavelet* yang cocok digunakan untuk kompresi dan *noise removal* pada sinyal suara. Kelemahan *Daubhecies wavelet transform* adalah adanya *overlapping windows* yang menyebabkan perubahan sinyal secara signifikan.

Hasil penelitian Prasad *et al* (2016: 120) menunjukkan bahwa *biorthogonal wavelet* mampu meningkatkan akurasi *edge detection* pada *X-Ray images* dengan jumlah *edge point* sebesar 222.964 pada bior 1.3 dan 229.276 pada bior 2.4. Nilai ini lebih tinggi dibandingkan *edge detection* dengan *orthogonal wavelet* yakni sebesar 16.296 pada Haar, 16.182 pada db3, dan 16.467 pada sym2. Hal ini disebabkan kompresi citra dengan *biorthogonal wavelet* melakukan kompresi citra dengan dua *wavelet* sekaligus, dengan satu *wavelet* untuk dekomposisi dan *wavelet* lainnya untuk rekonstruksi citra ke citra semula (Singh, 2011). Tingginya akurasi *edge detection* dapat meningkatkan akurasi *feature extraction* pada pengenalan citra.

Dalam penelitian Surowka & Ogorzalek (2016: 663) menyebutkan bahwa metode *reverse biorthogonal wavelet* lebih baik daripada *orthogonal wavelet*. Penelitian ini menerapkan berbagai jenis *wavelet* untuk melakukan dekomposisi

citra Melonema yakni rbio3.1, rbio 2.2, rbio1.5, rbio 4.4, bior 1.5, bior 1.1, dan coiflet 1. Hasil dekomposisi citra digunakan sebagai *input* dalam klasifikasi dengan *ensemble learning*. Setelah dilakukan percobaan dengan 3 citra Melonema dengan resolusi yang berbeda, diperoleh kesimpulan bahwa rbio3.1 lebih baik dari *wavelet transform* lainnya. Hal ini disebabkan rekonstruksi citra pada *reverse biorthogonal* lebih baik dari *orthogonal wavelet*, sehingga dapat mengembalikan citra dengan kerusakan minimal.

Pemilihan *pre-processing* citra yang tepat dapat meningkatkan hasil ekstraksi fitur pada citra. Semakin akurat hasil ekstraksi fitur, maka perbedaan antar citra wajah akan semakin besar. Hal ini akan memudahkan sistem pengenalan wajah untuk mengenali citra. Dengan demikian, akurasi pengenalan wajah akan meningkat.

Tahapan terakhir dalam *face recognition* adalah proses klasifikasi. Fitur-fitur penting wajah dihasilkan pada proses ekstraksi akan digunakan sebagai parameter untuk proses klasifikasi. Berdasarkan penelitian yang dilakukan oleh Sravanthi (2012: 6), metode *K-Nearest Neighbor* (KNN) lebih baik dan efisien dalam pengenalan wajah karena memiliki tingkat kesalahan yang rendah. Menurut Mahalle & Nanda (2016: 20195), KNN merupakan algoritma yang sangat mudah (*simple*) untuk dikenali tapi bekerja sangat baik.

Penelitian Sutarti *et al* (2019: 70) menyebutkan bahwa KNN cocok digunakan untuk klasifikasi wajah. Penelitian ini menggunakan dua jenis ekstraksi fitur yakni PCA dan 2D-PCA. Pengenalan wajah dengan ekstraksi fitur PCA dan KNN *classifier* memberikan hasil akurasi sebesar 89.38%, sedangkan pada

pengenalan wajah algoritma 2D-PCA dengan KNN *classifier* menghasilkan akurasi terbesar 96,88 %.

Berdasarkan uraian permasalahan di atas, maka diusulkan penelitian untuk meningkatkan akurasi pada pengenalan wajah menggunakan AT & T “The Database of Face” dan YALE dataset. Pada tahap *pre-processing* citra akan diterapkan dekomposisi citra menggunakan *multi-level reverse biorthogonal wavelet transform*. Setelah dekomposisi selesai, dilakukan ekstraksi fitur menggunakan PCA dengan jenis 1D-PCA dan 2D-PCA. Penelitian ini berjudul “Peningkatan Akurasi Pengenalan Wajah *Principal Component Analysis* Berbasis *Multi-level Reverse Biorthogonal Wavelet Transform* dengan *K-Nearest Neighbor*”.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah

- 1) Bagaimana implementasi pengenalan wajah *PCA* berbasis *multi-level reverse biorthogonal wavelet* dengan KNN?
- 2) Bagaimana hasil pengenalan wajah *PCA* berbasis *multi-level reverse biorthogonal wavelet* dengan KNN?

1.3. Batasan Masalah

Pada penelitian ini diperlukan batasan-batasan agar tujuan penelitian dapat tercapai. Adapun batasan masalah yang dibahas dalam penelitian ini adalah sebagai berikut:

- 1) Algoritma yang digunakan dalam penelitian ini adalah algoritma 1D-PCA dan 2D-PCA, *Reverse Biorthogonal Wavelet* 3.7, dan KNN.
- 2) *Dataset* yang digunakan dalam penelitian ini adalah dataset yang tersedia dari *website* UCI yakni AT&T “*The Database of Face*” (dulunya *ORL database*) dan *YALE dataset*.
- 3) Dekomposisi citra dilakukan dengan *Reverse Biorthogonal Wavelet* 3.7 sebanyak 2 level.
- 4) Ekstraksi ciri dilakukan dengan 1D-PCA dan 2D-PCA.
- 5) Proses klasifikasi dilakukan dengan KNN (berbasis *Euclidean distance*)
- 6) Perhitungan akurasi pengenalan wajah dilakukan dengan *confussion matrix*. Kemudian dilakukan perbandingan hasil akurasi antara 2D-PCA+KNN konvensional & *multi-level Reverse Biorthogonal Wavelet Transform*+PCA+KNN, dan 2D-PCA+KNN konvensional & *multi-level Reverse Biorthogonal Wavelet Transform*+2D-PCA+KNN
- 7) Lingkungan pengembangan perangkat lunak yang digunakan adalah Matlab 2019a.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut.

1. Melakukan peningkatan akurasi pengenalan wajah PCA berbasis *multi-level reverse biorthogonal wavelet transform* dengan KNN.
2. Melakukan perbandingan akurasi pengenalan wajah menggunakan PCA dengan KNN dengan pengenalan wajah menggunakan PCA berbasis *multi-level reverse biorthogonal wavelet transform* dengan KNN.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah memperoleh pengenalan wajah yang lebih akurat dengan PCA berbasis *multi-level reverse biorthogonal wavelet transform* dengan KNN.

1.6. Sistematika Penulisan

Sistematika penulisan berguna untuk memudahkan dalam memahami jalan pemikiran secara keseluruhan skripsi. Penulisan skripsi ini secara garis besar dibagi menjadi tiga bagian, yaitu sebagai berikut.

1.6.1. Bagian Awal Skripsi

Bagian awal skripsi terdiri dari halaman judul, halaman pengesahan, halaman pernyataan, halaman motto dan persembahan, abstrak, kata pengantar, daftar isi, daftar gambar, daftar tabel dan daftar lampiran.

1.6.2. Bagian Isi Skripsi

Bagian isi skripsi terdiri dari lima bab, yaitu sebagai berikut.

1) **BAB 1: PENDAHULUAN**

Bab ini berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian serta sistematika penulisan skripsi.

2) **BAB 2: TINJAUAN PUSTAKA**

Bab ini berisi penjelasan mengenai definisi maupun pemikiran-pemikiran yang dijadikan kerangka teoritis yang menyangkut dan mendasari pemecahan masalah dalam skripsi ini.

3) **BAB 3: METODE PENELITIAN**

Bab ini berisi penjelasan mengenai studi pendahuluan, tahap pengumpulan data, dan tahap pengembangan sistem.

4) **BAB 4: HASIL DAN PEMBAHASAN**

Bab ini berisi hasil penelitian beserta pembahasannya.

5) **BAB 5: PENUTUP**

Bab ini berisi simpulan dari penulisan skripsi dan saran yang diberikan penulis untuk mengembangkan skripsi ini.

1.6.3. Bagian Akhir Skripsi

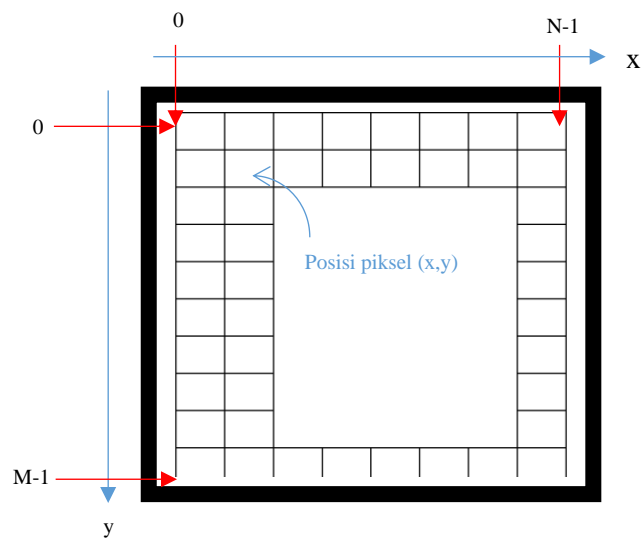
Bagian akhir skripsi ini berisi daftar pustaka yang merupakan informasi mengenai buku-buku, sumber-sumber dan referensi yang digunakan penulis serta lampiran-lampiran yang mendukung dalam penulisan skripsi ini.

BAB II

TINJAUAN PUSTAKA

2.1. Pengolahan Citra Digital

Citra digital merupakan kumpulan piksel yang dinotasikan dalam bentuk bilangan pada sejumlah baris dan kolom (Ardhianto,2010: 12). Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (*pixel* atau “*picture element*”). Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi. Sistem koordinat yang dipakai untuk menyatakan citra digital ditunjukkan pada Gambar 2.1.



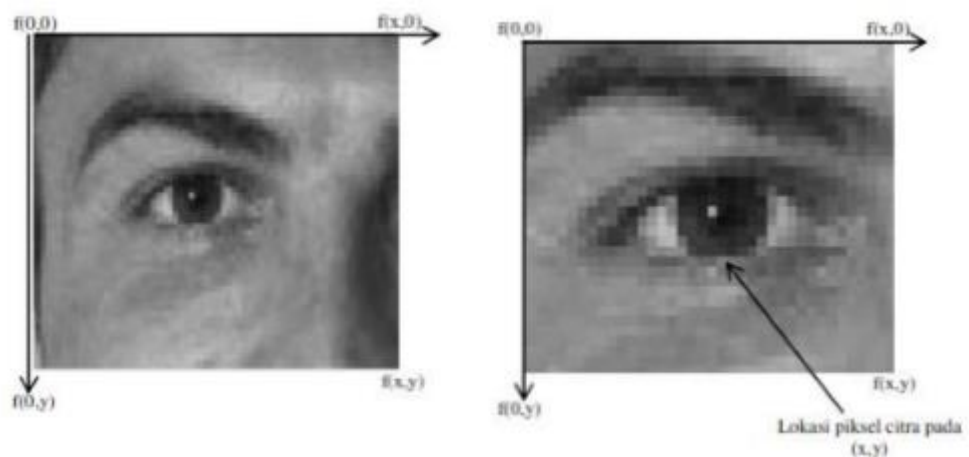
Gambar 2.1 Sistem Koordinat Citra Digital

Dimana:

- x menyatakan posisi kolom;
- y menyatakan posisi baris;

- piksel pojok kiri-atas mempunyai koordinat $(0, 0)$ dan piksel pada pojok kanan-bawah mempunyai koordinat $(N-1, M-1)$.

Menurut Gonzales & Woods (2008: 5), citra digital merupakan fungsi 2 dimensi, dimana $f(x, y)$ dengan x dan y sebagai koordinat spasial amplitudo f pada pasangan koordinat (x, y) yang disebut intensitas atau tingkat keabuan citra titik tersebut. Nilai-nilai intensitas dinyatakan dalam skala-skala tertentu yang telah disepakati berdasarkan jenis citra yang ditampilkan. Nilai intensitas $f(x, y)$ merupakan output dari sensor pada alat di posisi yang tetapi sesuai dengan koordinat spasial. Nilai $f(x, y)$ merupakan konsekuensi dari ketidakmampuan komputer dalam memproses citra, yang seharusnya merupakan fungsi kontinu dimensi 3 sesuai dengan kondisi nyatanya. Maka dari itu citra ditampilkan dalam bentuk titik-titik data diskrit.



Gambar 2.2. Representasi citra sebagai $f(x, y)$ (kiri); citra kiri yang telah diperbesar (kanan)

Citra digital merupakan citra yang didapat oleh komputer, disimpan dalam memori komputer dan berisi angka-angka yang menunjukkan besarnya intensitas

pada masing-masing piksel. Citra digital mengandung sejumlah elemen-elemen dasar. Elemen - elemen dasar tersebut dimanipulasi dalam pengolahan citra, elemen tersebut adalah kecerahan(*brightness*), kontras, kontur, warna, bentuk (*shape*), dan tekstur.

Pengolahan gambar atau pengolahan citra yang sering disebut *image processing*, merupakan suatu proses yang mengubah sebuah gambar menjadi gambar lain yang memiliki kualitas lebih baik untuk tujuan tertentu (Ardhianto, 2010: 12). Menurut Suparman & Marlan (2007: 45), pengolahan citra dikenal sebagai penghalusan (*enhancement*) citra yang merupakan bagian perbaikan kualitas citra. Pengolahan citra digital umumnya dapat dikelompokkan menjadi lima kelas dasar (Harjito, 1998: 4) yaitu:

- a. Meningkatkan kualitas penampakan citra, menonjolkan beberapa aspek informasi yang terkandung dalam citra (*image enhancement*), contoh: perbaikan kontras gelap/terang, perbaikan tepi objek, penajaman, pemberian warna semu, dan sebagainya.
- b. *Image restoration*, yaitu proses menghilangkan atau meminimumkan cacat pada suatu citra, contoh: penghilangan kesamaran (*deblurring*).
- c. *Image segmentation*, yaitu pengelompokan, pencocokan, dan mengukur elemen dalam citra. Operasi ini berkaitan erat dengan pengenalan pola.
- d. Mengetahui ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu mengidentifikasi objek (*object analysis*). Proses segmentasi (*image segmentation*) kadangkala diperlukan untuk merelokasi objek yang diinginkan dari sekelilingnya. Contoh: pendeteksian tepi objek.

- e. *Image compression*, yaitu memperkecil ukuran suatu citra agar mudah diolah atau diproses dalam komputer, contoh: suatu file citra berbentuk JPEG berukuran 258 KB dikompres dengan metode JPEG menjadi berukuran 50 KB.

Pada umumnya, objektivitas dari pengolahan citra adalah mentransformasi atau menganalisis suatu gambar sehingga informasi baru tentang gambar dibuat lebih jelas. Menurut Gonzalez & Woods (2008: 36) terdapat empat klasifikasi dasar dalam pengolahan citra yaitu :

- a. *Point*, memproses nilai *pixel* suatu gambar berdasarkan nilai atau posisi dari *pixel* tersebut. Contoh dari proses *point* adalah *adding*, *subtracting*, *contrast stretching* dan lainnya.
- b. *Area*, memproses nilai *pixel* suatu gambar berdasarkan nilai *pixel* tersebut beserta nilai *pixel* sekelilingnya. Contoh dari proses area adalah *convolution*, dan *blurring*.
- c. *Geometric*, digunakan untuk mengubah posisi dari *pixel*. Contoh dari proses *geometric* adalah *scaling*, *rotation*, dan *mirroring*.

Menurut Hermawati (2013 : 2) terdapat tiga tipe pengolahan citra yaitu:

- a. *Low-level process*

Low-level process meliputi operasi datar seperti *image processing: reduce noise, contrast enhancement, image sharpening*. *Input* dan *output* dalam level berupa gambar.

- b. *Mid-level process*

Mid-level process meliputi segmentasi (membagi sebuah gambar dalam *region* atau *object*), mendeskripsikan objek tersebut untuk direduksi dalam

bentuk yang diinginkan dan klasifikasi (*recognition*) dari objek tersebut. Input dan proses ini berupa gambar, dan output-nya berupa atribut yang diambil dari gambar tersebut (misal: *edge*, *contours* dan identitas dari objek tertentu).

c. *High-level process*

High-level process meliputi pemberian arti dari suatu rangkaian objek-objek yang dikenali dan akhirnya menampilkan fungsi-fungsi kognitif secara normal sehubungan dengan penglihatan

2.2. Pengenalan Wajah

Pengenalan wajah atau *face recognition* merupakan salah satu teknologi biometrik yang banyak diaplikasikan khususnya dalam sistem *security*. Sistem pengenalan wajah banyak diaplikasikan dalam kehidupan sehari-hari seperti dalam sistem presensi, dan pengenalan pelaku tindak kriminal dengan CCTV. Efisiensi dan akurasi menjadi faktor utama mengapa pengenalan wajah banyak diaplikasikan khususnya dalam sistem *security* (Marti, 2010: 11).

Menurut Sinurat (2014: 114-115) terdapat enam komponen dalam pengenalan citra yaitu:

- a. *Acquisition module*, merupakan blok input dari proses pengenalan wajah sumbernya dapat berasal dari kamera ataupun file citra.
- b. *Pre-processing module*, merupakan proses penyesuaian citra input yang meliputi, normalisasi ukuran citra, *histogram equalization* untuk memperbaiki kualitas citra input agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya, *media filtering* untuk menghilangkan *noise* akibat

kamera atau pergeseran *frame*, *high pass filtering* untuk menunjukkan bagian tepi dari citra, *background removal* untuk menghilangkan *background* sehingga hanya bagian wajah saja yang diproses dan normalisasi pencahayaan ketika mengambil citra input. Bagian *pre-processing* ini bertujuan untuk menghilangkan masalah yang akan timbul pada proses pengenalan wajah.

- c. *Feature extraction module*, digunakan untuk mengutip bagian terpenting sebagai suatu vektor yang merepresentasikan wajah dan bersifat unik.
- d. *Classification module*, pada modul ini fitur wajah yang dibandingkan dengan fitur yang telah tersimpan di database dengan bantuan pemisahan pola, sehingga dapat diketahui apakah citra wajah tersebut dikenali.
- e. *Training set*, modul ini digunakan selama proses pembelajaran, dan proses pengenalan. Semakin kompleks dan sering proses pengenalan wajah akan semakin baik.
- f. *Database*, berisi kumpulan citra wajah pengklasifikasian

Face recognition adalah proses mengecek suatu wajah apakah dikenali (*known*) atau tidak dikenali (*unknown*), dimana setelah dibandingkan citra akan disimpan secara tersendiri (Kustian, 2016: 193). Prinsip dasar dari pengenalan wajah adalah mengutip berbagai informasi unik dari citra wajah, kemudian informasi tersebut di-*encode* dan dibandingkan dengan hasil *decode* yang sebelumnya dilakukan (Suprianto *et al*, 2014: 180). Menurut Soni & Sahu (2013: 3899), sistem pengenalan wajah terdiri dari 3 langkah yaitu deteksi wajah, ekstraksi fitur & pengenalan wajah. Deteksi wajah (*face detection*) merupakan proses awal yang harus dilakukan pada citra input untuk mendeteksi adanya citra wajah di dalam

citra masukan. Proses deteksi wajah akan dilakukan bersama dalam proses ekstraksi fitur. Hal ini karena dalam proses deteksi wajah diperlukan ekstraksi fitur di dalam citra masukan yang akan mendeteksi apakah citra masukan merupakan citra yang mengandung ciri wajah. Proses pengenalan wajah ditunjukkan oleh Gambar 2.3.



Gambar 2.3. Proses Pengenalan Wajah

Teknologi pengenalan wajah saat ini sudah sangat canggih dan diterapkan di hampir semua sistem mutakhir yang mengelola citra dan foto (Kurniawan, 2015: 211). Dalam sistem pengenalan wajah, untuk mengidentifikasi wajah yang ada dalam gambar dan video secara otomatis dapat diklasifikasikan menjadi berikut (Patil & Deore, 2013: 31):

1. Verifikasi wajah (*Authentication*)

Dalam verifikasi wajah terdapat pencocokan satu wajah uji dengan satu wajah latih.

2. Identifikasi wajah

Dalam identifikasi wajah terdapat pencocokan satu wajah uji dengan banyak wajah latih

2.3. *Wavelet*

Wavelet adalah suatu metode pengolahan sinyal yang mana sebuah sinyal dipecah menjadi beberapa bagian yang merujuk kepada frekuensi yang berbeda-beda. *Wavelet* digunakan untuk menyusun, menganalisis dan mensintesis data numeris hasil pengukuran/pengamatan suatu fenomena fisis tertentu. Dengan

transformasi *wavelet*, sinyal digital dikalkulasi untuk menentukan domain frekuensi dan waktu secara bersamaan. Transformasi *wavelet* dapat diaplikasikan pada pengenalan objek, penghalusan citra (*smoothing*) dan kompresi (Saragih, 2008 :15).

Sebagai fungsi matematika, *wavelet* digunakan untuk mengekstraksi informasi didalam data yang berbeda-beda seperti sinyal audio dan gambar. Susunan dataset *wavelet* sepenuhnya dibutuhkan untuk menganalisa data. *Wavelet* bersifat komplemen dalam memecah data tanpa menghasilkan rentang atau menimpa set data sehingga data dapat dikembalikan seperti semula (*reversible*). Oleh karena itu, *wavelet* digunakan sebagai algoritma kompresi dan dekompresi dimana data yang udah dipecah dapat dikembalikan lagi dengan tingkat kerusakan yang minimal.

Wavelet dibentuk dari suatu fungsi *mother wavelet* yang dilambangkan dengan $\Psi(x)$ dalam suatu interval berhingga. *Wavelet* merupakan gelombang singkat (*small wave*) yang energinya terkonsentrasi pada suatu selang waktu untuk memberikan kemampuan analisis transien, ketidakstasioneran, atau fenomena berubah terhadap waktu (*time varying*). Karakteristik dari *wavelet* antara lain adalah beresilasi singkat, translasi (pergeseran), dan dilatasi (skala) (Burrus *et al*, 1998: 6-8).

Wavelet memiliki banyak famili, dibedakan berdasarkan pada bank filter yang digunakan. Famili *wavelet* terdiri atas *biorthogonal wavelet*, Meyer *wavelet*, Morlet *wavelet*, Shanon *wavelet*, dan masih banyak lainnya. *Biorthogonal wavelet* merupakan pengembangan dari *orthogonal wavelet*.

Transformasi *wavelet* akan mengolah suatu sinyal digital menjadi domain frekuensi dan domain waktu secara bersamaan. Transformasi *wavelet* pada awalnya digunakan untuk menganalisis sinyal bergerak (*non-stationary signals*). *Wavelet* merupakan suatu fungsi variabel *real t*, diberi notasi Ψt dalam ruang fungsi $L^2(\mathcal{R})$. Fungsi ini dihasilkan oleh parameter penskala (dilatasi) dan penggeseran (translasi) dari sebuah fungsi tunggal (induk) yang dinyatakan dalam Persamaan (2.1) (Burrus *et al*, 1998: 12):

$$\Psi_{a,b}(t) = a^{-\frac{1}{2}}\Psi\left(\frac{t-b}{a}\right), a > 0, b \in \mathcal{R} \quad (2.1)$$

Fungsi Persamaan (2.1) pertama kali diperkenalkan oleh Grossman dan Moret dengan $a, b \in \mathcal{R}$, dan $a \neq 0$ dimana a adalah parameter skala dan b adalah parameter translasi. Persamaan (2.1) diatas kemudian diperbaharui oleh Daubechies seperti Persamaan (2.2) (Burrus *et al*, 1998: 15):

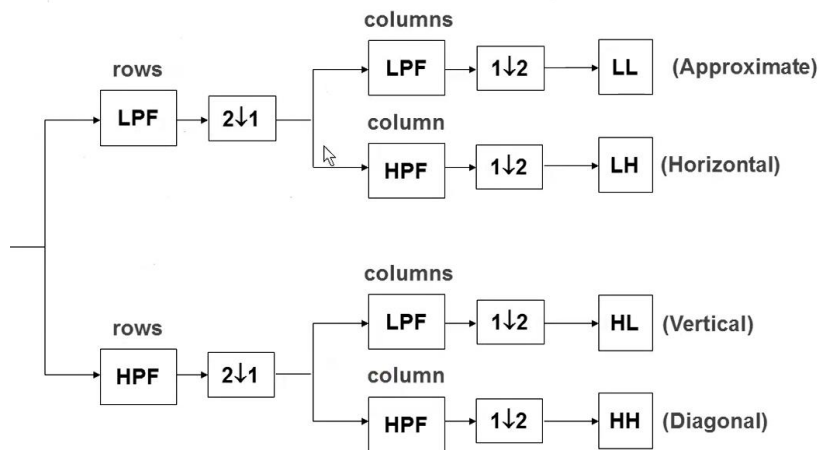
$$\Psi_{j,k}(t) = a^{\frac{j}{2}}\Psi(2^j t - k), j, k \in \mathbb{Z} \quad (2.2)$$

Dimana

- a : parameter penskala
- 2^j : parameter dilatasi
- k : parameter waktu atau lokasi ruang

Wavelet didasarkan pada proses pembangkitan sejumlah tapis (*filter*) yakni dengan cara mengeser dan menskala *mother wavelet* berupa tapis pelewat tengah (*band-pass filter*). Oleh karena itu, diperlukan pembangkit *filter*. Penambahan dan pengurangan skala akan mempengaruhi durasi waktu, lebar bidang (*bandwith*) dan nilai frekuensi (Burrus *et al*, 1998: 38).

Wavelet dibagi menjadi dua berdasarkan ruang dan waktu yaitu *wavelet* 1D (waktu) dan *wavelet* 2D (ruang). Waktu yang dimaksud adalah untuk gelombang 1D, adalah titik awal *point shifting* dari sumber menuju akhir, sedangkan pengertian ruang didalam *wavelet* 2D, *point shifting*nya adalah dua dimensi. Pada transformasi *wavelet* 2D, citra mengalami dekomposisi pada baris dan hasilnya akan dibagi dengan dekomposisi pada kolom. Cara kerja transformasi *wavelet* ditunjukkan oleh Gambar 2.4.



Gambar 2.4. Transformasi *Wavelet* 2 Dimensi

Cara kerja transformasi *wavelet* adalah dengan melakukan filterisasi digital. Sinyal yang diterima akan dianalisis pada *filter* dengan frekuensi dan skala yang berbeda. Sebuah sinyal harus dilewatkan dalam dua filterisasi DWT yaitu *high-pass filter* dan *low-pass filter* agar frekuensi dari sinyal tersebut dapat dianalisis. Analisis sinyal dilakukan terhadap hasil filterisasi *high-pass filter* dan *low-pass filter* dimana *high-pass filter* digunakan untuk menganalisis frekuensi tinggi dan *low-pass filter* digunakan untuk menganalisis frekuensi rendah. Analisis terhadap frekuensi

dilakukan dengan cara menggunakan resolusi yang dihasilkan setelah sinyal melewati filterisasi (Saragih, 2008: 16).

Transformasi *wavelet* dapat dibedakan menjadi dua, yaitu *continuous wavelet transform* (CWT) dan *discrete wavelet transform* (DWT). Fungsi yang digunakan dalam transformasi CWT dan DWT diturunkan dari *mother wavelet* melalui translasi/ pergeseran dan penskalaan/dilatasi. Transformasi *wavelet* kontinu mempunyai dua kelemahan yaitu *redundancy* dan ketidakpastian (*impracticality*) (Mallat, 1999: 15). Masalah tersebut dapat diselesaikan dengan mendiskretkan parameter penskala dan pergeseran.

2.3.1. Transformasi Wavelet Diskret / *Discrete Wavelet Transform* (DWT)

Transformasi *wavelet* merupakan teknik pemrosesan sinyal multiresolusi. Proses transformasi *wavelet* dilakukan dengan mengkonvolusi sinyal dengan data tapis (*filter*) atau dengan proses perata-rataan dan pengurangan secara berulang, yang sering disebut dengan metode filter bank. Transformasi *wavelet* merupakan perbaikan dari transformasi *Fourier*. Pada transformasi *Fourier* hanya dapat menentukan frekuensi yang muncul pada suatu sinyal, namun tidak dapat menentukan dimana frekuensi itu muncul. Dengan kata lain, transformasi *Fourier* tidak memberikan informasi tentang domain waktu (*time domain*). Kelemahan lain dari transformasi *Fourier* adalah perubahan sedikit terhadap sinyal pada posisi tertentu akan berdampak atau mempengaruhi sinyal pada posisi lainnya. Hal ini disebabkan karena transformasi *Fourier* berbasis sin-cos yang bersifat periodik dan kontinu.

Transformasi *wavelet* selain mampu memberikan informasi frekuensi yang muncul, juga dapat memberikan informasi tentang skala atau durasi atau waktu. *Wavelet* dapat digunakan untuk menganalisa suatu bentuk gelombang (sinyal) sebagai kombinasi dari waktu (skala) dan frekuensi. Selain itu perubahan sinyal pada suatu posisi tertentu tidak akan berdampak banyak terhadap sinyal pada posisi-posisi yang lainnya. Dengan *wavelet*, suatu sinyal dapat disimpan lebih efisien dibandingkan dengan *Fourier* dan lebih baik dalam hal melakukan aproksimasi terhadap *real-word signal* (Putra, 2010: 95).

Prinsip dasar dari DWT adalah bagaimana cara mendapatkan representasi waktu dan skala dari sebuah sinyal menggunakan teknik *filter* digital dan operasi *sub-sampling*. Proses DWT bertujuan mengurangi redundansi yang terjadi pada transformasi *wavelet* kontinu. DWT akan menganalisis suatu sinyal dengan skala yang berbeda.

Sebuah sinyal dilewatkan dalam dua filter DWT yaitu *high-pass filter* dan *low-pass filter* agar frekuensi dari sinyal tersebut dapat dianalisis. Pembagian sinyal menjadi frekuensi tinggi dan frekuensi rendah dalam proses *high-pass filter* dan *low-pass filter* disebut sebagai dekomposisi. Proses dekomposisi ini dapat melalui satu atau lebih tingkatan. Dekomposisi satu tingkat ditulis dengan ekspresi matematika pada Persamaan (2.3) dan (2.4).

$$y_{rendah}[k]=\sum_n x[n]g[2k - n] \quad (2.3)$$

$$y_{tinggi}[k]=\sum_n x[n]gh \quad (2.4)$$

Dimana

$y[k]$ tinggi : hasil *high-pass filter*

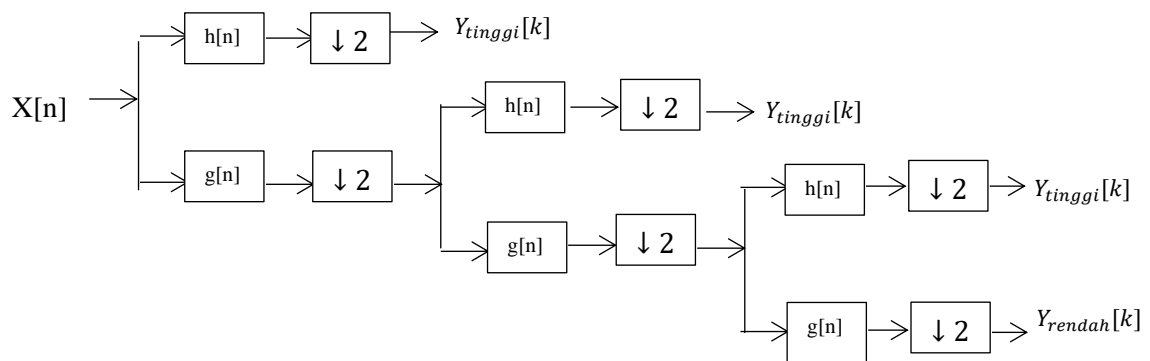
$y[k]$ rendah : hasil *low-pass filter*

$x[n]$: sinyal asal

$h[n]$: *high-pass filter*

$g[n]$: *low-pass filter*

Contoh ilustrasi dekomposisi dipaparkan pada Gambar 2.5 dengan menggunakan dekomposisi tiga tingkat. Pada Gambar 2.5 $y[k]$ tinggi dan $y[k]$ rendah yang merupakan hasil dari *high-pass filter* dan *low-pass filter*, $y[k]$ tinggi disebut sebagai koefisien DWT. $y[k]$ tinggi merupakan detail dari informasi sinyal, sedangkan $y[k]$ rendah merupakan taksiran kasar dari fungsi penskalaan. Dengan menggunakan koefisien DWT ini maka dapat dilakukan proses *inverse discrete wavelet transform* (IDWT) untuk merekonstruksi menjadi sinyal asal.



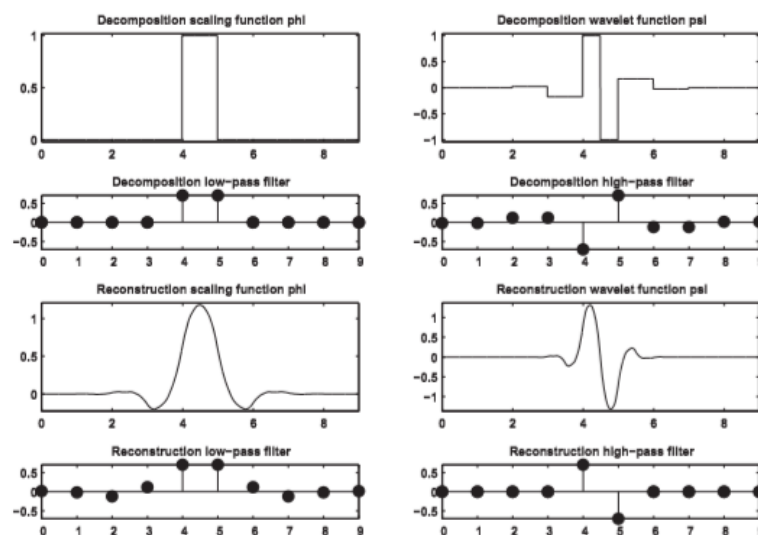
Gambar 2.5 Dekomposisi *Wavelet* 3 Tingkat

Proses rekonstruksi diawali dengan menggabungkan koefisien DWT dari yang berada pada akhir dekomposisi dengan sebelumnya meng *upsample* oleh 2 ($\uparrow 2$) melalui *high-pass filter* dan *low-pass filter*. Proses rekonstruksi ini sepenuhnya merupakan kebalikan dari proses dekomposisi sesuai dengan tingkatan pada proses dekomposisi. Dengan demikian, persamaan rekonstruksi pada masing-masing tingkatan dapat ditulis sebagai Persamaan (2.5)

$$x[n] = \sum_k (y_{tinggi}[k]h[-n + 2k] + y_{rendah}[k]g[-n + 2k]) \quad (2.5)$$

2.3.1.1. Reverse Biorthogonal Wavelet

Dalam kasus-kasus pengolahan sinyal tertentu, kita membutuhkan *symmetrical coefficient* untuk mencapai linearitas. Pengolahan sinyal seperti ini digunakan dalam pengolahan citra dan rekonstruksi sinyal. *Biorthogonal wavelet* merupakan jenis *wavelet* agar dapat mencapai properti simetri dan rekonstruksi *exact* sehingga memungkinkan penggunaan *Finite Impulse Response (FIR)*, yakni suatu *filter* yang tidak bisa dilakukan *orthogonal wavelet*. *Biorthogonal wavelet* juga mendukung *spline*. *Biorthogonal wavelet* menggunakan fungsi penskalaan yang terpisah untuk dapat menganalisis gambar. *Reverse biorthogonal wavelet* merupakan gabungan *biorthogonal wavelet*. Dalam praktiknya, *biorthogonal wavelet* melakukan 2 wavelet dalam 1 citra yakni sebelah kiri untuk dekomposisi dan sebelah kanan untuk rekonstruksi. Operasi *reverse biorthogonal wavelet* ditunjukkan oleh Gambar 2.6.



Gambar 2.6. Operasi *Reverse Biorthogonal Wavelet*

2.4. *Principal Component Analysis (PCA)*

Metode PCA adalah metode yang paling banyak digunakan dalam pengenalan wajah manusia (Affandi, 2011: 6). PCA dapat digunakan untuk menemukan pola atau ciri dalam data yang memiliki dimensi tinggi dengan cara mereduksi citra masukan menjadi citra yang memiliki dimensi lebih rendah. Ada banyak teknik untuk mereduksi dimensi dari *image* yang digunakan pada proses ekstraksi ciri, salah satunya *eigenface algorithm*. *Eigenface* merupakan algoritma yang didasarkan pada PCA (Budi *et al*, 2016: 171).

PCA pertama kali ditemukan pada 1991 oleh Turk dan Pentland. PCA juga dikenal sebagai *Karhunen-Loeve expansion* adalah fitur dasar ekstraksi dan teknik representasi data, dan teknologi ini banyak digunakan dalam bidang pengenalan pola dan *computer vision* (Kirby & Sirovich, 1990: 103) . Tujuan dari PCA adalah untuk mereduksi dimensi yang besar dari ruang data (*observed variables*) menjadi dimensi intrinsik yang lebih kecil dari ruang fitur (*independent variables*), yang dibutuhkan untuk mendeskripsikan data lebih sederhana. Walaupun dimensi citra menjadi kecil, citra tidak akan kehilangan banyak informasi karena variasi data tetap dipertahankan minimal 80% (Johnson & Wichern, 1992: 642).

PCA akan mencari *eigenface* yang merupakan kumpulan dari vektor eigen. *Eigenface* adalah ciri-ciri penting (*principal component*) dari distribusi citra wajah yang didapatkan dari *eigenvector*. Untuk mendapatkan *eigenface*, PCA melakukan perhitungan matriks kovarian dari kumpulan citra *training*. *Eigenface* tersebut akan menjadi dasar perhitungan jarak wajah yang merepresentasikan nilai bobot individu yang mewakili satu atau lebih citra wajah. Nilai bobot hasil *eigenface* digunakan

untuk mengenali citra *testing* dengan mencari jarak nilai bobot citra *testing* dengan nilai bobot citra *training*. Perhitungan jarak nilai bobot dilakukan dengan perhitungan jarak Euclidean (*Euclidean distance*). Dalam penelitian ini terdapat dua jenis PCA yang digunakan yakni 1D-PCA dan 2D-PCA

2.4.1. 1D-PCA

Metode 1D-PCA merupakan pencarian fitur-fitur citra dengan *eigenface* dimana citra yang semula dua dimensi diubah ke citra berdimensi satu. Adanya transformasi dimensi pada citra menyebabkan ruang vektor dimensi citra menjadi besar, sehingga akan sulit untuk dapat mengevaluasi matriks kovarian yang besar dengan akurat apabila hanya dengan menggunakan data sampel yang kecil. Oleh sebab itu, metode 1D-PCA memiliki kelemahan yakni metode ekstraksi ciri ini bersifat skalar dan juga membutuhkan data yang luas/banyak untuk data *training* maupun data *testing* agar mendapatkan akurasi yang lebih baik (Purnama & Salsabila, 2013: 1).

Berikut ini langkah-langkah dalam ekstraksi citra wajah terhadap suatu basis data citra pelatihan metode 1D-PCA (Aris *et al*, 2016: 171):

1. Buatlah citra berukuran sama $N \times N$ pada gambar wajah I .
2. Baca tiap- tiap citra wajah pelatihan, $I = (I_1, I_2, I_3, \dots, I_i)$.
3. Ubah dimensi citra wajah menjadi vektor berukuran $1 \times N_2$ dan lambangkan setiap gambar I_i sebagai vector r_i
4. Hitung rata-rata matriks dengan Persamaan (2.6)

$$\Psi = \frac{1}{M} \sum_{i=1}^M r_i \quad (2.6)$$

5. Kurangkan setiap matriks citra wajah dengan rata-ratanya menggunakan Persamaan (2.7)

$$\Phi = r_i - \Psi \quad (2.7)$$

6. Menghitung matriks kovarian menggunakan Persamaan (2.8)

$$C = \Phi \Phi^T \quad (2.8)$$

$$\text{Dimana } \Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \dots \\ \Phi_4 \end{bmatrix}$$

Dikarenakan ukuran matriks terlalu besar, maka pencarian matriks kovarian menjadi Persamaan (2.9)

$$C = A^T A \quad (2.9)$$

7. Menghitung *eigenvalue* dan *eigenvector*

Bentuk persamaan *eigenvalue* dan *eigenvector* dinyatakan oleh Persamaan (2.10)

$$C_{u_i} = \lambda_{u_i} \quad (2.10)$$

Dimana

$$\text{Nilai eigen } (\lambda) = \text{Det}(C - \lambda I)$$

$$\text{Vektor eigen } (u_i) = (C - \lambda_i I) u_i$$

8. Menghitung *eigenface*(μ) dengan Persamaan (2.11)

$$\mu = \sum_{i=1}^M u_i \Phi_i \quad (2.11)$$

2.4.2. 2D-PCA

Metode 2D-PCA merupakan penyempurnaan 1D-PCA dimana pencarian fitur dilakukan tanpa melakukan transformasi pada citra dua dimensi menjadi citra satu dimensi. Metode 2D-PCA memiliki dua keunggulan penting atas metode PCA.

Pertama, lebih mudah untuk mengevaluasi matriks kovarian secara akurat. Kedua, lebih sedikit waktu yang diperlukan untuk menentukan vektor *eigen* yang sesuai (Oliveira *et al*, 2011: 9).

Definisi metode 2D-PCA menurut Purnama & Salsabila (2013: 2) adalah jika A merupakan matriks citra asli berukuran $m \times n$ dan X merupakan matriks dengan kolom *orthonormal*, maka 2D-PCA merupakan proyeksi A pada X dengan persamaan yang ditunjukkan oleh Persamaan (2.12):

$$Y = AX \quad (2.12)$$

Cara menentukan matriks X optimal untuk proyeksi citra adalah sebagai berikut.

Terdapat persamaan seperti Persamaan (2.13)

$$J(X) = \text{trace}(Sx) \quad (2.13)$$

Sx merupakan matriks kovariansi dari proyeksi vektor fitur dari data training, $\text{trace}(Sx)$ merupakan trace Sx .

Secara matematis Sx didefinisikan seperti Persamaan (2.14)

$$Sx = E[(Y - E(Y))(Y - E(Y))^T] \quad (2.14)$$

Maka dari persamaan di atas dapat diturunkan menjadi Persamaan (2.15), yang dapat diturunkan kembali menjadi Persamaan (2.16), dengan bentuk akhir persamaan adalah Persamaan (2.17).

$$J(X) = \text{trace} \{ E[(Y - E(Y))(Y - E(Y))^T] \} \quad (2.15)$$

$$J(X) = \text{trace} \{ E[(AX - E(AX))(AX - E(AX))^T] \} \quad (2.16)$$

$$J(X) = X^T E[(A - E(A))^T (A - E(A))]X \quad (2.17)$$

Berikutnya dimisalkan G yaitu matriks kovariansi citra definit non negatif berukuran $n \times n$ sebagai Persamaan (2.18)

$$G = E[(A - E(A))^T (A - E(A))] \quad (2.18)$$

Maka persamaan dapat diturunkan menjadi Persamaan (2.19)

$$J(X) = X^T GX \quad (2.19)$$

Jika terdapat sebanyak M citra data latih, dengan \bar{A} merupakan rata-rata semua citra latih maka rata-rata semua citra latih dinyatakan sebagai Persamaan (2.20)

$$\bar{A} = \frac{1}{M} \sum_k A_k \quad (2.20)$$

Maka G dapat dihitung menggunakan Persamaan (2.21)

$$G = \frac{1}{M} \sum_{k=1}^m (A_k - \bar{A})^T (A_k - \bar{A}) \quad (2.21)$$

Pada percobaan, tiap-tiap citra dapat di-*shift* terlebih dahulu dengan pengurangan terhadap *mean* citra itu sendiri, sehingga nilai $\bar{A} = 0_{(m \times n)}$

Dimisalkan $X_{optimal}$ yaitu merupakan himpunan matriks proyeksi optimal.

Berdasarkan persamaan diatas, maka persamaan diturunkan menjadi Persamaan (2.22) dan Persamaan (2.23).

$$X_{optimal} = \arg \max J(X) \quad (2.22)$$

$$X_{optimal} = \arg \max (X^T GX) \quad (2.23)$$

Secara intuitif didapatkan $X_{optimal}$ merupakan himpunan sejumlah d *eigenvector* dari G berkorespondensi dengan sejumlah d *eigenvalue* tertinggi yang diperoleh dengan Persamaan (2.24).

$$X_{optimal} = \{X_1, X_2, \dots, X_d\} \quad (2.24)$$

Maka hasil ekstraksi fitur citra A menggunakan 2D-PCA adalah $Y_{optimal}$ dengan bentuk Persamaan (2.25) dan Persamaan (2.26)

$$Y_{optimal} = AX_{optimal} \quad (2.25)$$

$$Y_{optimal} = \{Y_1, Y_2 \dots Y_d\} \quad (2.26)$$

2.5. K-Nearest Neighbor (KNN)

KNN adalah suatu metode yang menggunakan algoritma *supervised* dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari label class pada KNN. Tujuan dari algoritma KNN adalah mengklasifikasikan objek baru berdasarkan atribut dan *training* data. Algoritma KNN bekerja berdasarkan jarak terpendek dari *query instance* ke training data untuk menentukan KNN-nya. Salah satu cara untuk menghitung jarak dekat atau jauhnya tetangga menggunakan metode *Euclidian distance*.

KNN merupakan algoritma klasifikasi yang termasuk dalam “*lazy learner*” karena algoritma tidak banyak melakukan proses pelatihan selain menyimpan data pelatihan. KNN tidak membangun sebuah model klasifikasi melainkan hanya menyimpan data pelatihan yang terlabeli. Oleh karena itu, algoritma KNN merupakan algoritma yang sangat fleksibel dalam pengenalan wajah penambahan *training images* tidak memerlukan banyak proses pelatihan.

Klasifikasi KNN dilakukan dengan membandingkan jarak antara data *training* dengan data *testing*. Ketika ada masukan data testing, maka KNN mencari jarak terdekat (*Euclidean distance*) data *testing* terhadap data *training* yang diketahui. Matriks *Euclidean distance* digunakan untuk menentukan kedekatan titik-titik data/jarak antar data dalam K-NN (Dhriti & Kaur, 2012: 15). Fungsi yang digunakan untuk mencari *Euclidean distance* ditunjukkan oleh Persamaan (2.27) (Han *et al*, 2012: 460):

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2} \quad (2.27)$$

keterangan:

x_1 = data *training*

x_2 = data *testing*

i = variabel data

d_i = jarak

p = dimensi data

Semakin besar nilai d maka semakin jauh nilai keserupaan citra, dan sebaliknya semakin kecil nilai d maka citra akan semakin serupa. Citra pengujian akan dikelompokkan dengan citra pelatihan yang memiliki jarak d terdekat. Menurut Han *et al* (2012: 421-425), pengklasifikasian KNN didasarkan pada analogi seperti membandingkan data uji dengan data latih yang mirip dengannya. Data latih diberikan sebanyak n atribut. Setiap data merupakan titik dalam ruang n -dimensi. Semua data pelatihan disimpan diruang pola n -dimensi. Ketika ada data yang tidak diketahui, KNN akan mencari ruang pola untuk k data pelatihan yang terdekat dengan data yang tidak diketahui.

Pengklasifikasian KNN, data akan masuk dalam kelas yang paling umum (banyak kesamaan) diantara k tetangga terdekatnya. KNN ketika diberikan “*test*” data maka data akan mencari pola ruang k data latih yang paling dekat dengan “*test*” data tersebut. Pengklasifikasian algoritma KNN tidak hanya didasarkan pada fitur yang memiliki bobot sama antara data *testing* dengan data *training* tetapi juga fitur yang memiliki bobot data *testing* mendekati data *training* (Vedayoko *et al*, 2017: 135)

Keuntungan klasifikasi KNN (Alimjam *et al*, 2017: 2):

- a. Sederhana, mudah dimengerti, mudah diterapkan, tidak perlu memperkirakan parameter, tidak ada pelatihan.
- b. Cocok untuk klasifikasi kejadian langka.
- c. Sangat cocok untuk masalah multi klasifikasi (multi modal, objek memiliki beberapa kategori label), KNN lebih baik daripada kinerja *Support Vector Machine* (SVM).

2.6. *Confussion Matrix*

Evaluasi model klasifikasi didasarkan pada pengujian untuk memprediksi klasifikasi yang benar dan salah. Pengukuran kinerja klasifikasi dapat dilakukan dengan cara matriks konfusi (*confussion matrix*). Dimana kelas yang diprediksi ditampilkan di bagian atas matriks dan yang diamati di sisi kiri matriks. Setiap sel berisi angka yang menunjukkan berapa banyak kasus yang sebenarnya dari kelas yang diamati untuk prediksi (Han *et al.*, 2012: 364). *Confussion matrix* untuk klasifikasi dapat dilihat pada Tabel 2.1.

Tabel 2.1. *Confussion Matrix* untuk Klasifikasi Dua Kelas

<i>Classification</i>		Kelas hasil prediksi	
		YES	NO
Kelas	YES	TP	FN
asli	NO	FP	TN

Confussion matrix melakukan klasifikasi dalam dua kelas, di mana nilai dua kelas tersebut yaitu YES dan NO. *Confussion matrix* dilakukan dengan menghitung jumlah kelas hasil prediksi terhadap kelas asli. Hasilnya dinyatakan dalam *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, dan *False Negative (FN)*. *True Positive (TP)* dan *True Negative (TN)* memberikan informasi ketika *classifier* benar sesuai dengan kelas asli, sedangkan *False Positive (FP)* dan *Negative Negative (FN)* memberikan informasi ketika *classifier* salah atau tidak sesuai dengan kelas asli (Hussain *et al.*, 2018: 449). Kuantitas *confussion matrix* dapat diringkas menjadi dua nilai, yaitu akurasi dan tingkat kesalahan. Dengan mengetahui jumlah data yang diklasifikasikan secara benar, dapat diketahui akurasi hasil prediksi, dan dengan mengetahui jumlah data yang diklasifikasikan secara salah dapat mengetahui tingkat kesalahan dari prediksi yang dilakukan. Oleh karena itu, pengukuran akurasi (*accuracy*) dapat dituliskan dengan Persamaan (2.28), sedangkan ukuran tingkat kesalahan klasifikasi juga dapat dihitung dengan Persamaan (2.29) berikut ini:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (2.28)$$

$$Error Rate = \frac{FP+FN}{TP+TN+FP+FN} \times 100\% \quad (2.29)$$

2.7. Penelitian Terkait

Penelitian ini dikembangkan dari beberapa referensi penelitian terdahulu yang mempunyai keterkaitan dengan metode dan objek penelitian. Penggunaan referensi ini ditujukan untuk memberikan batasan-batasan terhadap metode yang

nantinya akan dikembangkan lebih lanjut. Beberapa penelitian terkait dengan penelitian yang diusulkan ditunjukkan oleh Tabel 2.2.

Tabel 2.2. Penelitian Terkait

Penulis	Judul	Metode	Hasil
Moresh M Mukhedkar & Samarjeet B. Powalkar (2015)	<i>Fast Face Recognition Based on Wavelet Transform on PCA</i>	<i>Daubechies wavelet 3 level (Daubechies 1 & Daubechies 2) dan PCA</i>	Penerapan dekomposisi <i>Daubechies wavelet transform</i> sebanyak 3 level pada citra dapat meningkatkan akurasi pengenalan wajah rata-rata sebesar 1% menggunakan dekomposisi <i>wavelet Daubechies2</i> sebanyak 3 level dan 3% menggunakan dekomposisi <i>wavelet Daubechies1</i> .
Prasad <i>et al</i> (2016)	<i>Perfomance Analysis of Orthogonal an Biorthogonal Wavelet for Edge Detection of X- Ray Images</i>	<i>Orthogonal wavelet & biorthogonal wavelet</i>	<i>Biorthogonal wavelet</i> lebih baik dari <i>orthogonal wavelet</i> dalam <i>edge detection</i> pada <i>X-Ray Images</i>
Grzegorz Sur´owka & Maciej	<i>On Optimal Wavelet Bases for</i>	Jenis <i>wavelet</i> yang digunakan yakni <i>rbio 3.1</i>	Jenis <i>wavelet</i> yang paling cocok untuk klasifikasi citra

Ogorzalek (2016)	<i>Classification of Melanoma Images Through Ensemble Learning</i>	(Reverse biorthogonal wavelet pair: (3,1)), Rbio 2.2, Rbio 1.5, Rbio 4.4, Bior 1.5 (Biorthogonal wavelet pair: (1,5)), Bior 1.1, dan Coif 1 (Coiflet 1	Melonema adalah rbio3.1.
Sutarti et al (2019)	<i>Comparison of PCA and 2DPCA Accuracy with K-Nearest Neighbor Classification in Face Image Recognition</i>	PCA, 2D-PCA, & KNN	Akurasi PCA+KNN adalah 89,83% sedangkan akurasi 2DPCA+KNN mencapai 96,88%.

BAB III

METODE PENELITIAN

3.1 Alat dan Bahan

3.1.1. Alat

Alat yang digunakan dalam pembuatan program peningkatan akurasi pengenalan wajah pengenalan wajah PCA berbasis *multi-level reverse biorthogonal wavelet transform* dengan KNN adalah sebagai berikut:

1. Laptop dengan spesifikasi sebagai berikut
 - a) *Processor* : Intel Core i5-8250U
 - b) *Memory* : 8 GB
 - c) *Hard disk* : 1 TB
 - d) *SSD* : *No*
 - e) *Sistem operasi* : Windows 10 64 bit
2. Matlab R2019a

3.1.2. Bahan

Bahan yang digunakan dalam AT & T “*Face of Database*” dan YALE *face database*. Keduanya merupakan *dataset* UCI yang dapat diunduh di <https://www.kaggle.com/kasikrit/att-database-of-faces> dan <https://www.kaggle.com/olgabelitskaya/yale-face-database>. AT&T “*The Database of Face*” (sebelumnya bernama *ORL database*) adalah *database* yang berisi serangkaian citra wajah yang diambil antara bulan April 1992 dan April 1994 di

sebuah laboratorium. Terdapat 40 subjek atau kelas dimana tiap kelas terdiri dari 10 citra wajah sehingga total keseluruhan data yaitu 400 citra wajah (Sutarti, 2019: 68). Keseluruhan citra wajah ini, akan dibagi menjadi data *training* dan data *testing* sesuai dengan komposisi yang telah ditetapkan dan akan digunakan dalam proses pengolahan data. Citra wajah diambil secara berbeda baik dari segi waktu, *lighting*, dan aksesoris (memakai kacamata/tidak memakai kacamata). Semua gambar diambil dengan latar belakang gelap dengan subjek dalam posisi tegak dan menghadap ke depan (dengan toleransi untuk beberapa perubahan pengambilan sudut sisi wajah). *File-file* tersebut dalam bentuk format BMP dengan ukuran setiap gambar yaitu 92x112 piksel dengan *grey levels* 21 per piksel. Sampel citra wajah AT&T “*The Database of Face*” dapat dilihat pada Gambar 3.1.



Gambar 3.1. *Sample* wajah AT&T “*The Database of Face*”

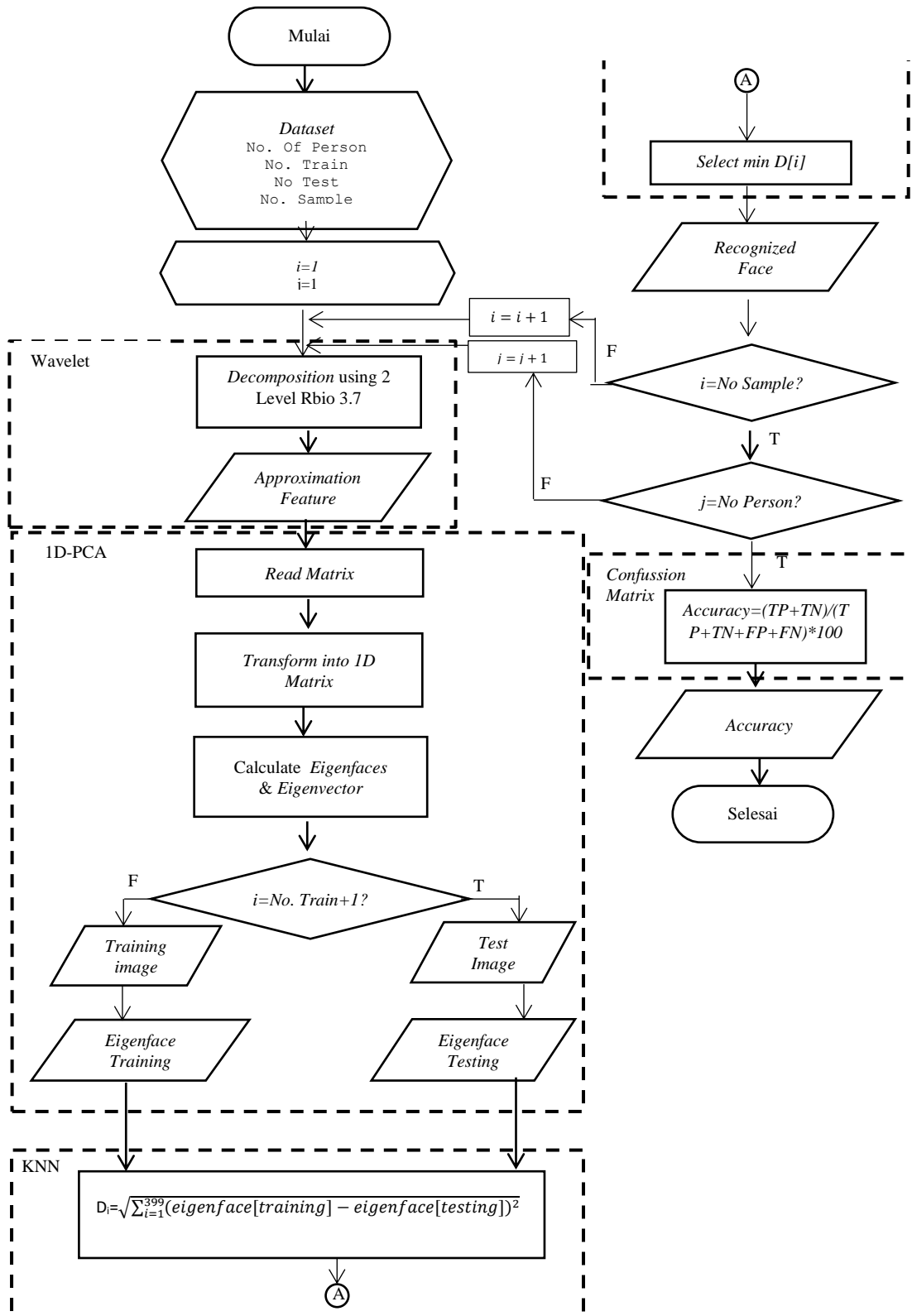
Dataset kedua yang digunakan adalah YALE *face database*. YALE *face database* berisi citra 15 subjek dengan total 165 citra (Linge & Pawar, 2014 : 4012). YALE *face database* disimpan dalam format GIF dengan ukuran 6.4 MB. Aksesoris yang digunakan adalah kacamata. Citra wajah diambil dengan tingkat kecerahan dan aksesoris yang berbeda (memakai kacamata/tidak memakai kacamata). Sampel citra wajah YALE *face database* dapat dilihat pada Gambar 3.2.



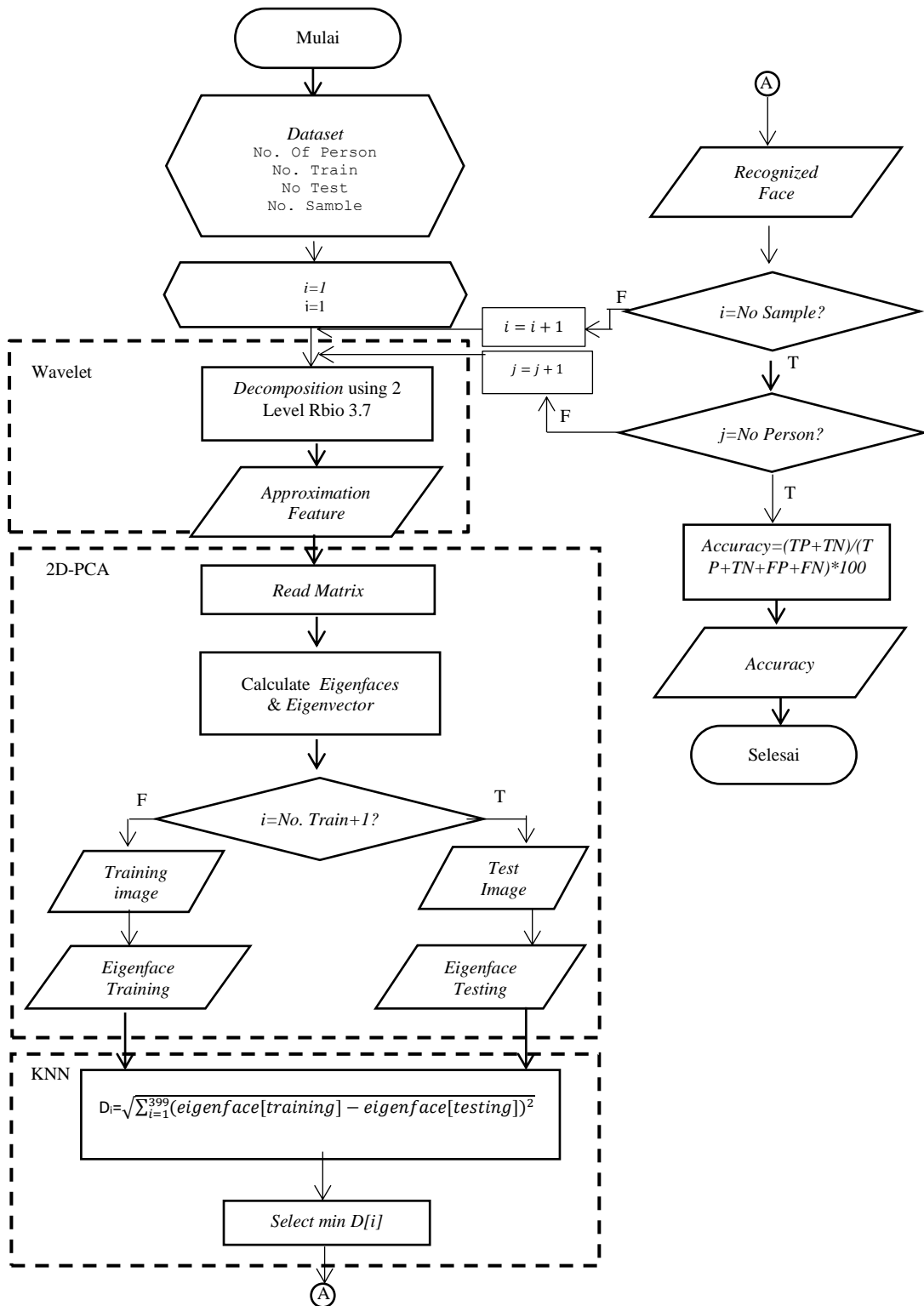
Gambar 3.2. *Sample wajah YALE face database*

3.2 Metode yang Diusulkan

Pada penelitian ini, dilakukan penerapan dilakukan dekomposisi citra menggunakan *multi-level reverse biorthogonal wavelet* pada tahap *preprocessing*. *Multi-level reverse biorthogonal wavelet* dilakukan untuk mereduksi ukuran citra sekaligus meningkatkan hasil ekstraksi fitur menggunakan PCA baik 1D-PCA maupun 2D-PCA. Adapun metode klasifikasi yang digunakan adalah algoritma KNN. Dari hasil klasifikasi akan diuji apakah terjadi peningkatan akurasi antara pengenalan citra sebelum dan sesudah menerapkan dekomposisi citra dengan *multi-level reverse biorthogonal wavelet*. *Flowchart* sistem pengenalan citra pada AT&T *Database* ditunjukkan oleh Gambar 3.3. Pada tahap awal sistem pengenalan wajah akan dilakukan inisiasi jenis *dataset* yang digunakan (AT&T atau YALE), No. Of Sample yang menunjukkan jumlah *subject* penelitian, No. Train menunjukkan jumlah citra yang digunakan sebagai *training images*, No. Test menunjukkan jumlah citra yang digunakan sebagai *testing images*, dan No. Sample menunjukkan jumlah citra tiap subjek.



Gambar 3.3 Flowchart Sistem (Menggunakan 1D-PCA)



Gambar 3.4 Flowchart Sistem (Menggunakan 2D-PCA)

3.2.1. Pembagian Data

Terdapat dua *dataset* dengan total masing-masing citra adalah 400 citra dari AT & T *face of database* dan 165 citra YALE *database face* sehingga total adalah 565 citra. Keseluruhan citra wajah ini akan dibagi menjadi data *training* dan data *testing*. Pembagian data ini dilakukan secara manual dengan mendefinisikan pada awal program. Secara rinci pembagian presentase data yang dijadikan untuk pengujian dalam penelitian ini ditunjukkan pada Tabel 3.1. dan Tabel 3.2 .

Tabel 3.1 Pembagian Data *Training* dan Data *Testing* AT&T

Persentase <i>Data Training</i>	Persentase <i>Data Testing</i>	Jumlah <i>Data Training</i>	Jumlah <i>Data Testing</i>	Total Data
10%	90%	40	360	400
50%	50%	200	200	400
90%	10%	360	40	400

Tabel 3.2 Pembagian Data *Training* dan Data *Testing* YALE

Persentase <i>Data Training</i>	Persentase <i>Data Testing</i>	Jumlah <i>Data Training</i>	Jumlah <i>Data Testing</i>	Total Data
9,09%	90,01%	15	150	165
54,54%	45,45%	90	75	165
90,01%	9,09%	150	15	165

Dari sejumlah *training image* tersebut akan dipilih satu *image* yang akan dibaca sebagai *test image*. *Training* dan *testing image* akan diperlakukan sama hingga diperoleh *euclidean distance* yang akan dibandingkan dalam proses klasifikasi.

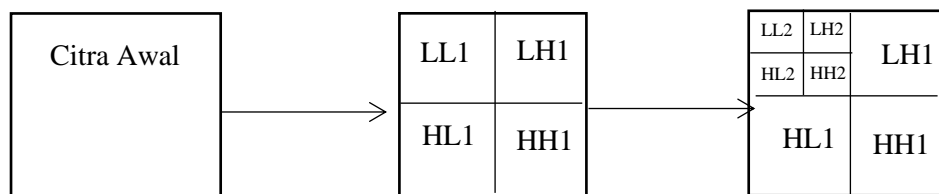
3.2.2. Dekomposisi Citra dengan *Reverse Biorthogonal Wavelet Transform*

Reverse biorthogonal wavelet dilakukan untuk dekomposisi citra. *Reverse biorthogonal wavelet* pada level pertama akan membagi citra menjadi empat *subband* yakni LH, HH, dan HH yang merepresentasikan skala koefisien *wavelet* terbaik,

sedangkan LL yang merepresentasikan skala koefisien *wavelet* kasar (Woods, 2012: 140). Koefisien sinyal hasil dekomposisi *wavelet* pada level pertama meliputi:

- a. LL merupakan koefisien aproksimasi paling kasar. Koefisien LL didapatkan *low-pass filter* pada koordinat baris dan kolom.
- b. HL dan HH merupakan koefisien yang didapatkan dengan menerapkan *low-pass filter* pada salah satu koordinat, dan *high-pass filter* pada koordinat yang lain.
- c. HH merupakan koefisien yang didapatkan dengan menerapkan *high-frequency* secara diagonal. HH diperoleh dengan menerapkan *high-pass filter* pada koordinat baris dan kolom.

Untuk mendapatkan koefisien *wavelet* pada level kedua, operasi *wavelet* dilakukan terhadap koefisien LL. Dari koefisien LL akan didekomposisi hingga diperoleh empat koefisien yakni LL2, LH2, dan HH2. Operasi *wavelet* dua level ditunjukkan oleh Gambar 3.5.



Gambar 3.5 Operasi *Wavelet* Dua Level

3.3.2. Ekstraksi Fitur dengan PCA

PCA digunakan untuk melakukan ekstraksi fitur terhadap citra LL2 hasil operasi *reverse biorthogonal wavelet* sebelumnya. PCA akan menghasilkan komponen-komponen utama dari wajah.

3.3.2.1.1D-PCA

Adapun tahapan dalam 1D-PCA adalah sebagai berikut:

1. Ubah citra wajah kedalam bentuk matriks
2. Ubah menjadi matriks 2 dimensi
3. Hitung rata-rata (*mean*) vektor citra wajah
4. Hitung *deviation from mean*
5. Hitung matriks kovarian
6. Hitung *eigenvalues* dan *eigenvectors*
7. Hitung *eigenfaces*

3.3.2.2.2D-PCA

Adapun tahapan dalam 2D-PCA untuk adalah sebagai berikut:

1. Ubah citra wajah kedalam bentuk matriks
2. Hitung rata-rata (*mean*) vektor citra wajah
3. Hitung *deviation from mean*
4. Hitung matriks kovarian
5. Hitung *eigenvalues* dan *eigenvectors*
6. Hitung *eigenfaces*

3.3.3. Klasifikasi dengan KNN

Tahapan selanjutnya adalah klasifikasi wajah menggunakan KNN. Hal ini dilakukan dengan membandingkan *eigenface* masing-masing citra uji dan citra latih. Wajah citra uji dikelompokkan dengan wajah citra latih yang memiliki jarak

eigenface terdekat. Perhitungan KNN ditunjukkan oleh persamaan (2.27) pada pembahasan sebelumnya.

Eigenface citra uji akan dibandingkan dengan setiap *eigenface* citra latih hingga didapatkan nilai *distance* minimal. Citra dengan nilai *distance* minimal tersebut merupakan *recognized face* yang dianggap sebagai citra yang serupa dengan citra uji.

3.3.4. Perhitungan Akurasi dengan *Confussion Matrix*

Tahapan terakhir adalah perhitungan akurasi dengan *confussion matrix*. Pengujian menggunakan model klasifikasi didasarkan pada pengujian untuk memperkirakan obyek yang benar dan salah, urutan pengujian ditabulasikan dalam *confusion matrix*. Setiap sel berisi angka yang menunjukkan berapa banyak kasus yang sebenarnya dari kelas yang diamati untuk didapatkan tingkat akurasinya. Komponen dalam tabel *confussion matrix* adalah:

- TP: *True Positives*, merupakan jumlah data dengan kelas positif yang diklasifikasikan positif. Hal ini berarti jumlah *train images* serupa dengan *test image* yang berhasil diklasifikasikan serupa dengan *test image*.
- TN : *True negatives*, merupakan jumlah data dengan kelas negatif yang diklasifikasikan negatif. Hal ini berarti jumlah *train images* tidak serupa dengan *test image* yang berhasil diklasifikasikan tidak serupa dengan *test image*.
- FP : *False positives*, merupakan jumlah data dengan kelas positif diklasifikasikan negatif. Hal ini berarti jumlah *train images* serupa dengan *test image* yang diklasifikasikan tidak serupa dengan *test image*.

- *FN* : *False negatives*, merupakan jumlah data dengan kelas negatif diklasifikasikan positif. Hal ini berarti jumlah *train images* tidak serupa dengan *test image* yang diklasifikasikan tidak serupa dengan *test image*.

Dari komponen-komponen diatas, dilakukan perhitungan akurasi dengan persamaan (2.28) pada pembahasan sebelumnya.

Hasil perhitungan diatas adalah akurasi pengenalan wajah dengan metode DWT+1D-PCA dan KNN. Nilai tersebut akan dibandingkan dengan nilai akurasi metode 1D-PCA+KNN konvensional. Begitu pula dengan nilai akurasi DWT+ 2D-PCA dan KNN, akan dibandingkan dengan nilai 2D-PCA dan KNN konvensional.

3.3. Perancangan Sistem

Pada penelitian ini, sistem dibuat sebagai alat uji penerapan *multi-level reverse biorthogonal wavelet* pada pengenalan wajah dengan ekstraksi fitur PCA dan klasifikasi KNN untuk mendapatkan hasil akurasi. Sistem dibuat dengan memakai bahasa pemrograman Matlab. *Matrix Laboratory* (Matlab) merupakan salah satu bahasa pemrograman yang dikembangkan oleh Mathworks (Paulus & Nataliani, 2007: 17). Matlab tidak hanya berfungsi sebagai bahasa pemrograman, tetapi sekaligus sebagai alat visualisasi, yang berhubungan langsung dengan ilmu matematika. Oleh karena itu, Matlab semakin banyak digunakan oleh para programmer yang menghendaki kepraktisan dalam membuat program. Dalam penelitian ini versi Matlab yang digunakan adalah Matlab 2019a.

3.4 Analisis Hasil

Analisis hasil merupakan tahapan yang paling penting dalam penyelesaian suatu kegiatan penelitian ilmiah. Dalam penelitian ini, dilakukan analisis tentang akurasi dari pengenalan wajah menggunakan algoritma PCA berbasis *multi-level reverse biorthogonal wavelet transform* dengan KNN menggunakan AT&T “*The Database of Face*” dan YALE “*Database Face*”.

3.5 Penarikan Kesimpulan

Penarikan kesimpulan didasarkan pada studi literatur, pengumpulan data dan pengembangan sistem serta hasil analisis dari penelitian. Dari metode penelitian yang digunakan, simpulan yang diperoleh dalam penelitian ini adalah tentang bagaimana hasil akurasi penggunaan algoritma PCA berbasis *multi-level reverse biorthogonal wavelet transform* dengan KNN pada proses pengenalan wajah.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Penelitian

4.1.1. Tahap Implementasi Sistem

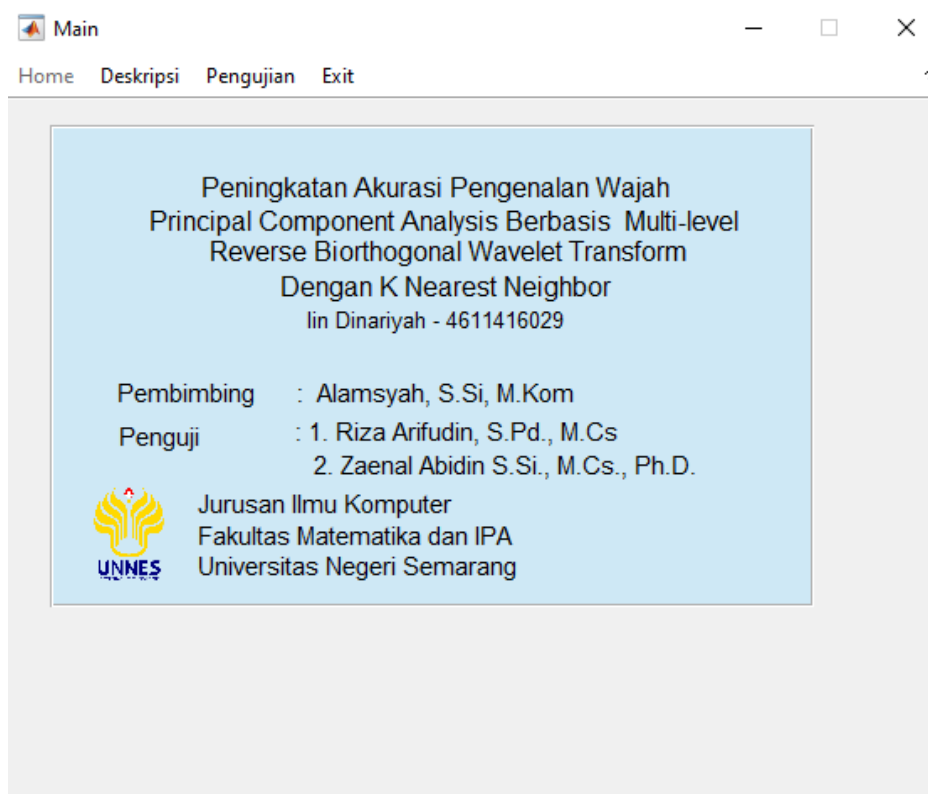
Sistem diimplementasikan dalam bentuk Matlab *Graphical User Interface* (GUI) yang dirancang dengan *library GUIDE (GUI Design)*. Program ditulis dalam bahasa Matlab dengan bantuan *Image Processing Toolbox* dan *Wavelet Toolbox*. *Image Processing Toolbox* merupakan fitur yang disediakan oleh Matlab untuk memudahkan pengolahan, visualisasi, dan pengembangan algoritma untuk pemrosesan citra. Adapun *wavelet toolbox* digunakan untuk proses dekomposisi citra. Kedua *toolbox* ini menyediakan fungsi-fungsi dasar yang dapat dipanggil dalam program utama. Adapun fungsi yang digunakan dalam sistem pengenalan wajah ini adalah

- `imread()`, digunakan untuk membaca file citra.
- `imshow()`, digunakan untuk menampilkan file citra.
- `Wavedec()`, digunakan untuk melakukan dekomposisi pada citra.
- `Appcoef2()`, digunakan untuk menyimpan *approximation coefficient*.

Sistem pengenalan wajah ini dirancang menggunakan Matlab 2019a. Tampilan sistem terdiri dari 3 desain yang terbagi atas 2 desain *interface* dengan GUI Matlab, dan 1 kelas utama yang secara otomatis akan melakukan *generate* tampilan *figure* pada Matlab:

a. Desain Tampilan Utama

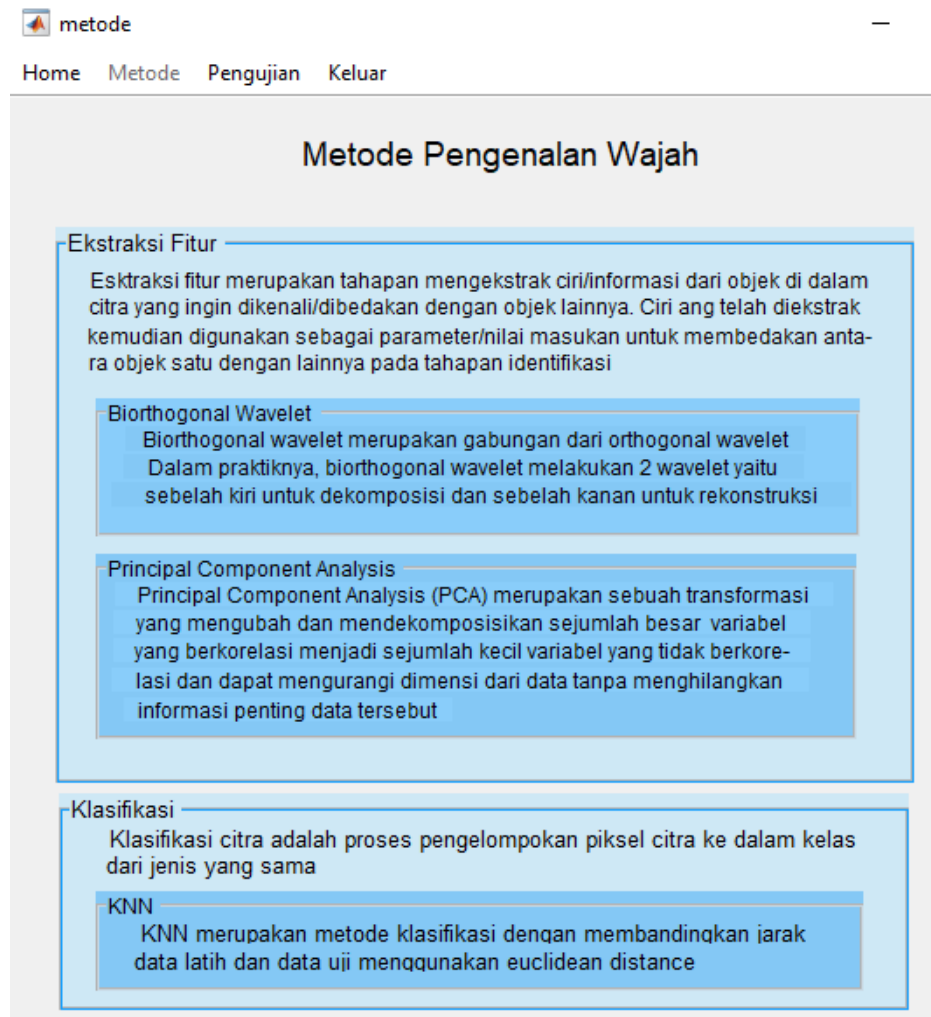
Menu utama didesain untuk menampilkan judul skripsi yang dibuat, data diri penulis dan nama dosen pembimbing menggunakan *static text*. Saat program pengenalan wajah dijalankan halaman menu utama yang pertama kali terlihat. Tampilan menu utama dapat dilihat pada Gambar 4.1.



Gambar 4.1. Tampilan Menu Utama

b. Tampilan Metode

Menu metode didesain untuk menampilkan penjelasan singkat mengenai metode-metode yang digunakan dalam proses pengenalan wajah, yaitu *reverse biorthogonal wavelet transform* dan PCA untuk tahap ekstraksi ciri serta KNN untuk tahap klasifikasi. Tampilan menu metode dapat dilihat pada Gambar 4.2.



Gambar 4.2. Tampilan Menu Metode

c. Pengujian

Pengujian pengenalan wajah ditulis dalam bentuk *script non GUI* yang secara otomatis akan membangkitkan *figure* apabila dijalankan. Proses awal dalam *script* ini adalah *load dataset* dalam bentuk file `.mat`. Terdapat 2 *dataset* yang dapat di-load yakni *AT&T Database of Face* dan *YALE Database Face*. Setiap *dataset* dikonversi dalam bentuk `.mat` yang berisi setiap matriks dari masing-masing citra dimana satu *dataset* terdiri atas satu *file .mat* citra asli dan satu *file .mat* citra hasil dekomposisi *reverse biorthogonal wavelet* sebanyak dua level.

File .mat citra asli diperoleh dengan membaca ukuran panjang dan lebar citra pertama *dataset*. Ukuran panjang dan lebar ini akan menjadi acuan dalam membuat matriks penyimpanan setiap elemen citra. Langkah selanjutnya adalah membuat perulangan sebanyak jumlah citra untuk menyimpan setiap elemen masing-masing citra. Hasil proses ini adalah matriks tiga dimensi dengan ukuran panjang \times lebar \times jumlah citra. *File .mat* citra hasil dekomposisi *reverse biorthogonal wavelet* diperoleh dengan melakukan dekomposisi citra pada gambar pertama dengan *reverse biorthogonal wavelet* sebanyak 2 level. Ukuran *approximation coefficient* citra pertama akan menjadi patokan dalam indeks array penyimpanan keseluruhan citra. Dekomposisi citra kedua sampai terakhir dilakukan dengan melakukan iterasi sejumlah citra. Pada tiap iterasi dilakukan penyimpanan *approximation coefficient*. Hasil proses ini adalah matriks tiga dimensi dengan ukuran panjang \times lebar \times jumlah citra dengan ukuran yang jauh lebih kecil dari *file .mat* citra asli. *Pseudocode* untuk ekspor citra asli AT & T, citra hasil dekomposisi AT & T, citra asli YALE, dan citra hasil dekomposisi YALE ditunjukkan oleh secara berturut-turut oleh Gambar 4.4, Gambar 4.5, Gambar 4.6, dan Gambar 4.7.

```

Program SaveOri
{Program digunakan untuk export AT&T dataset menjadi .mat}
Kamus :
- Img      : uint8
- r,c,h,k  : int
Algoritma
img←imread('1.bmp');
[r,c]←size(img);
h←NaN(r,c,5);
h(:,:,1)←img;
for k ← 2 to 400 do
    h(:,:,k)←imread(sprintf('%d.bmp',k));
end_for
save ORLDWT h

```

Gambar 4.3. *Pseudocode* Ekspor Citra Asli AT & T

```

Program SaveDWT
{Program digunakan untuk export AT&T dataset terdekomposisi
menjadi .mat}
Kamus :
- Img,C,S : uint8
- r,c,h,k : int
Begin
img←imread('1.bmp');
img← approximation feature of 2 level reverse biorthogonal
wavelet on img
[r,c] ←size(img);
h←NaN(r,c,5);
h(:,:,1) ←img;
for k ← 2 to 400 do
gambar=imread(sprintf('%d.bmp',k));
gambar← approximation feature of 2 level reverse
biorthogonal wavelet on gambar
h(:,:,k)=gambar;
end_for
save ORLDWT h

```

Gambar 4.4. *Pseudocode* Ekspor Citra Hasil Dekomposisi AT & T

```

Program SaveOri
{Program digunakan untuk export YALE dataset menjadi .mat}
Kamus :
- Img : uint8
- r,c,h,k : int
Algoritma
img←imread('1.bmp');
[r,c] ←size(img);
h←NaN(r,c,5);
h(:,:,1) ← img;
for k ← 2 to 165 do
h(:,:,k) ←imread(sprintf('%d.bmp',k));
end_for
save YALE h

```

Gambar 4.5. *Pseudocode* Ekspor Citra Asli YALE

```

Program SaveDWT
{Program digunakan untuk export YALE dataset terdekomposisi
menjadi .mat}
Kamus :
- Img,C,S : uint8
- r,c,h,k : int
Begin
img←imread('1.bmp');
img← approximation feature of 2 level reverse biorthogonal
wavelet on img
[r,c] ←size(img);
h←NaN(r,c,5);

```

```

    h(:,:,1) ←img;
for k ← 2 to 400 do
gambar=imread(sprintf('%d.bmp',k));
    gambar← approximation feature of 2 level reverse
    biorthogonal wavelet on gambar
    h(:,:,k)=gambar;
end_for
save YALEDWT h

```

Gambar 4.6. *Pseudocode* Ekspor Citra Dekomposisi YALE

Pengujian akan melakukan *training* dan *testing* pada citra wajah. Proses pemisahan citra *testing* dan *training* dilakukan dengan pendefinisian jumlah *training* dan *testing* pada awal program pengujian. *Pseudocode* fungsi ini ditunjukkan oleh Gambar 4.7. Setiap matriks citra latih akan diberi label 1 sampai total citra latih yang diperoleh dengan mengalikan jumlah *training sample* dengan jumlah subjek. Sedangkan matriks citra uji akan diberi label 1 sampai total citra uji yang diperoleh dengan mengalikan jumlah *testing sample* dengan jumlah subjek

```

function Train_Test(DATA : double ,No_SampleClass:
double,No_TrainSamples: double,No_TestSamples: double)
[m n TotalSamples]=size(DATA)←double
{fungsi ini digunakan untuk membagi sample images menjadi
training images & testing images}
Kamus
- d1,d2,t1,t2 : int
Algoritma
1. %Train Images
d1←1; d2←No_TrainSamples;
t1←1; t2←No_TrainSamples;
for i←1 to No_TrainSamples*(TotalSamples/No_SampleClass) do
Train(:,:,d1:d2)←DATA(:,:,t1:t2);
d1←d1+No_TrainSamples; d2←d2+No_TrainSamples;
t1←t1+No_SampleClass; t2←t2+No_SampleClass;
if (t1 > TotalSamples)
break;
endif
endfor
2. %Train Images
c1←1; c2←No_TestSamples;
e1=e1+No_SampleClass; e2=e2+No_SampleClass;

```

```

e1←No_TrainSamples+1;
e2←No_TestSamples+No_TrainSamples;
for i ←1 to No_TestSamples*(TotalSamples/No_SampleClass) do
  Test(:, :, c1:c2) ←DATA(:, :, e1:e2);
  c1←c1+No_TestSamples; c2←c2+No_TestSamples;
  if (e1 > TotalSamples)
    break;
  endif
endfor

```

Gambar 4.7. *Pseudocode* Pemisahan *Training Images* dan *Testing Images*

Selanjutnya dilakukan ekstraksi fitur pada seluruh citra. Pada 1D-PCA, citra terlebih dahulu mengalami proses reshape, sedangkan pada 2D-PCA tidak. Citra tersebut akan menjadi *input* dalam perhitungan *eigenvectors* dengan fungsi *eig* dengan fungsi seperti pada Gambar 4.8.

```

function
eig_decomp(St1:double)←[eigvec_St1:double,eigval_St1:double]
{fungsi ini digunakan untuk mendapatkan eigenvalue & eigenvector}
Kamus
- eigvec,eigval,eigenvalSt1,eigenvecSt1 : double
[eigvec,eigval]← eigenvalue of images St1
eigval←abs(diag(eigval)');
[eigval,I]←sort(eigval);
eigval_St1←swap column of eigval;
eigvec_St1←swap column of eigenvector

```

Gambar 4.8. *Pseudocode* Perhitungan *Eigenvectors*

Masing-masing *eigenvector* akan dibandingkan untuk mencari keserupaannya. Hal ini merupakan proses klasifikasi citra uji. Citra latihan dengan nilai *euclidian distance* minimum akan dianggap sebagai citra serupa dengan citra uji. Citra ini akan ditampilkan beserta nilai *Euclidian distancenya*. Setiap nilai selisih antara citra uji dan seluruh *training images* akan ditampilkan dalam grafik. *Pseudocode* proses klasifikasi ditampilkan dalam Gambar 4.9.

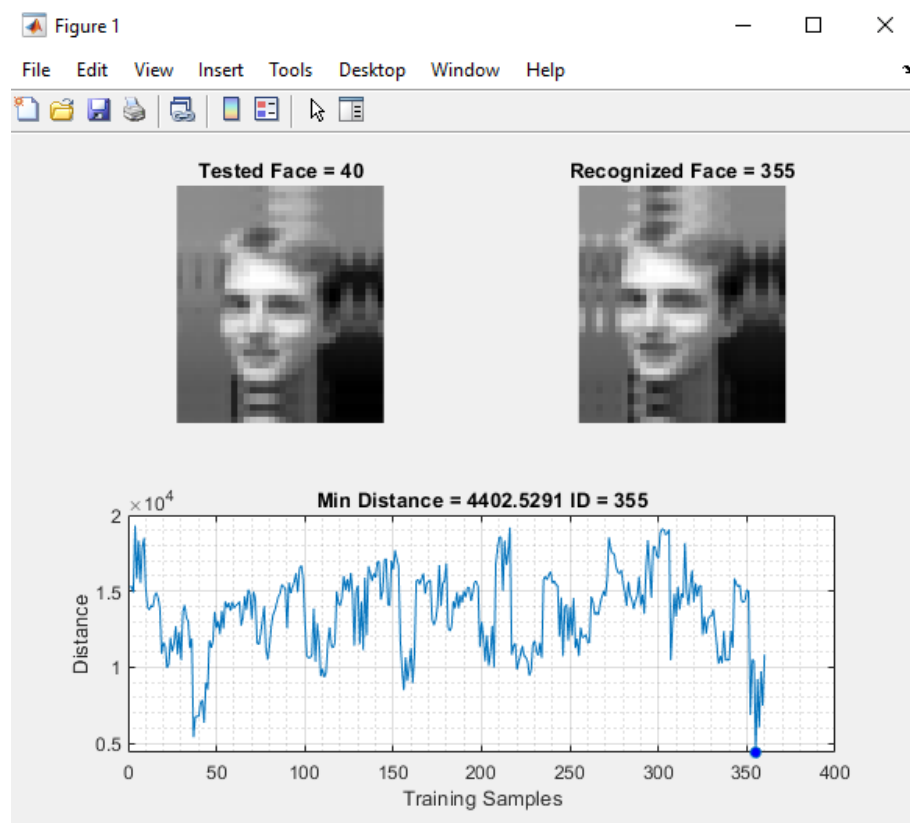

```

DIST←sqrt(sum(DIST1.*DIST1));
[MINDIST ID] ←min(DIST);
Show Testedface
Show Recognized Face
Printf MINDIST
Show Graph
TestResult(i) = TrainLabel(ID);

```

Gambar 4.9. *Pseudocode* Kode Proses Klasifikasi

Hasil tampilan kode di atas adalah citra uji dan citra latih yang diprediksi serupa dengan citra uji beserta nilai *Euclidian distance*. Nilai *distance* ditunjukkan dengan grafik. Semakin rendah titik pada grafik maka akan kedua citra akan dianggap semakin serupa. Tampilan pengujian ditunjukkan oleh Gambar 4.10.



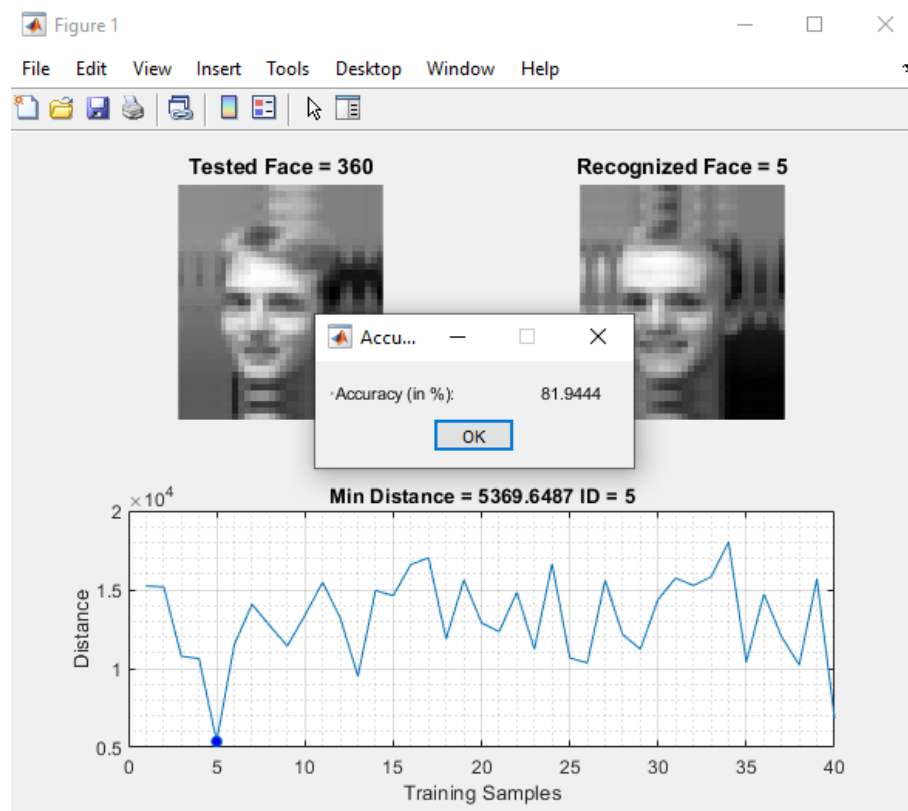
Gambar 4.10. Tampilan Pengujian

Setelah proses pengenalan seluruh citra uji selesai sistem akan menghitung tingkat akurasi pengenalan wajah. Hasil akurasi akan ditampilkan dalam *message box*. Perhitungan akurasi dilakukan dengan *pseudocode* seperti pada Gambar 4.11.

```
Result ← (TestResult == TestLabel);
CorrectRate ← 100*sum(Result/(TotalTestSamples));
msgbox(['·Accuracy (in %):', num2str(CorrectRate)], 'Accuracy')
```

Gambar 4.11. *Pseudocode* Perhitungan Akurasi

Suatu pengenalan dikatakan benar ketika `TestResult` bernilai sama dengan `TestLabel`. Artinya kedua citra berada dalam satu kelas yang sama. Nilai `CorrectRate` dipanggil dalam *message box* dan ditampilkan seperti paa Gambar 4.12.



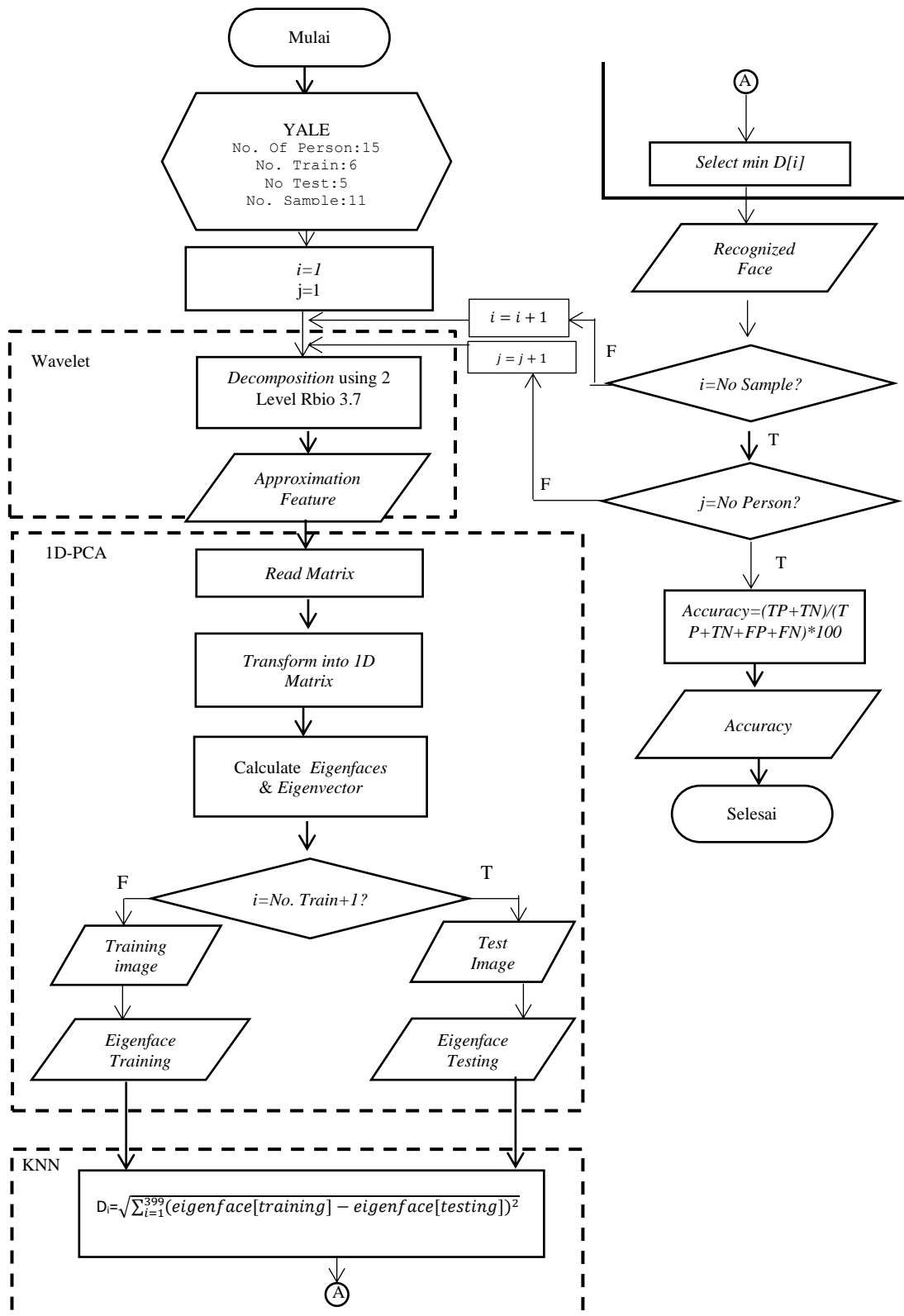
Gambar 4.12. Tampilan Akurasi Sistem

4.1.2. Tahap Pengujian Sistem

Metode yang digunakan untuk pengenalan wajah dalam penelitian ini adalah PCA yang meliputi 1D-PCA maupun 2D-PCA dengan metode klasifikasi KNN yang diterapkan pada *AT&T Face Database* dan *YALE Database*, baik pada citra asli maupun pada citra yang telah didekomposisi dengan *multi-level reverse biorthogonal wavelet*. Pengujian dilakukan sebanyak tiga kali dengan rasio *training images* dan *testing image* yang berbeda. Total pengujian yang dilakukan adalah 24 kali pengujian (12 kali percobaan menggunakan *AT&T Face Database* dan *YALE Database* dengan ekstraksi fitur 1D-PCA serta 12 kali percobaan menggunakan *AT&T Face Database* dan *YALE Database* dengan ekstraksi fitur 2D-PCA).

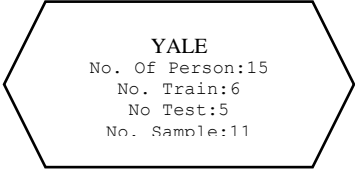
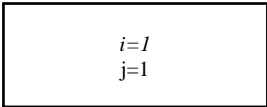
Output pengujian yang dilakukan adalah hasil akurasi pengenalan wajah. Akurasi dihitung menggunakan *confussion matrix*. Hasil akurasi pengenalan wajah dengan dekomposisi *multi-level reverse biorthogonal wavelet* akan dibandingkan dengan hasil akurasi pengenalan wajah tanpa dekomposisi *multi-level reverse biorthogonal wavelet*.

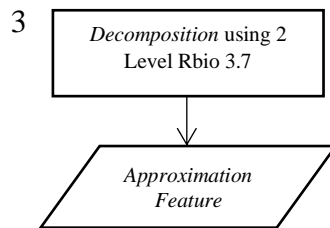
Salah satu contoh pengujian yang dilakukan adalah menggunakan *YALE dataset* dengan 6 citra latih dan 5 citra uji. *Flowchart* kasus pengujian di atas oleh Gambar 4.13



Gambar 4.13. Diagram Alur Pengujian Kasus

Penjelasan *flowchart* diatas adalah sebagai berikut :

1.  : Proses *load dataset* YALE.mat. YALE.mat berisi 165 citra wajah dengan ukuran 243x320x165. *Dataset* ini merupakan data yang akan diproses dalam pengenalan wajah. Penjelasan variabel adalah sebagai berikut
- No Of Person: jumlah subjek dalam *dataset* YALE yaitu 15 subjek
 - No. Train : jumlah citra yang akan digunakan sebagai citra latih tiap subjek.
 - No. Test : jumlah citra yang akan digunakan sebagai citra uji tiap subjek.
 - No. Sample : jumlah citra tiap subjek. Jumlah No. Sample harus sama dengan total No. Train dan No. Test
2.  : - i,j : *counter* perulangan



4. Proses 1D-PCA

: Proses dekomposisi *rbio* dua level. Proses ini akan melakukan kompresi citra dan akan menghasilkan matriks gambar dari ukuran semula 243 x 320 menjadi matriks dengan ukuran 72 x 91 piksel.

: Proses ekstraksi fitur pada citra wajah untuk menghasilkan *eigenface* pada masing-masing citra. Tahapan proses ini adalah

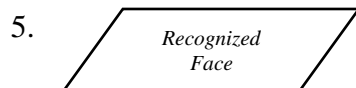
- *Read Matrix*: proses membaca elemen matriks
- *Transform info 1D*: proses *reshaping* matriks dari dua dimensi menjadi satu dimensi. Dari citra berukuran 72 x 91 piksel menjadi 6552 x 1
- *Calculate eigenvalue & eigenvectors*: proses perhitungan *eigenvalue & eigenvector* citra
- Pengecekan apakah citra termasuk *training images* atau *testing images*: proses penentuan apakah citra termasuk *testing* atau *training*. Proses ini menghasilkan 2 kelompok citra

yakni *training images* dan *testing images*

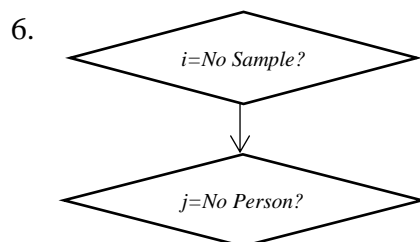
- Perhitungan *eigenface*: menghitung *eigenface* masing-masing citra

4. KNN

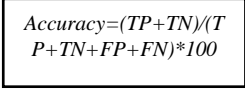
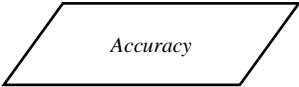
: Proses klasifikasi kelas citra uji. Hal ini dilakukan dengan membandingkan 2 *eigenfaces* citra. Citra uji dikelompokkan satu kelas dengan citra yang paling sedikit *euclidian distance*-nya. Hasil proses KNN adalah kelompok citra latih yang seurpa dengan citra uji dan citra yang paling mirip dengan citra uji



: Proses menampilkan citra latih yang paling mirip dengan citra uji



: Pengecekan kondisi untuk perulangan program. Setiap proses dilakukan sebanyak *No. Sample x No Person*, yakni $6 \times 5 = 165$ kali. Penentuan kelas subjek dilakukan tiap *No. Sample*. Artinya setiap perulangan terakhir dari *no sample*, citra akan dianggap sebagai citra dengan kelas baru.

7.  : Perhitungan akurasi menggunakan *confusion matrix*
8.  : Tampilan akurasi dalam *message box*

Untuk pengujian menggunakan ekstraksi fitur 2D-PCA, proses yang terjadi hampir sama akan tetapi tidak mengalami proses *reshape* menjadi matriks $N \times 1$. Hasil pengujian tiap percobaan dibahas dalam sub-bab selanjutnya.

4.1.2.1. Pengujian Menggunakan AT&T

- a. Pengujian Menggunakan 10% *Training Images* dan 90% *Testing Images*

Pengujian ini menggunakan 10% citra latih yakni sebanyak 40 citra dan 90% citra uji yakni sebanyak 360 citra. Pengaturan awal pada percobaan ini ditunjukkan oleh Gambar 4.14.

```
Num_Class←40;
No_SampleClass←10;
No_TrainSamples←1;
No_TestSamples←9;
```

Gambar 4.14. Pengaturan Percobaan 10% *Training Images* dan 90% *Testing Images*

- Metode 1D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 64,4444%, ini artinya dari 360 *images* terdapat 232 *images* yang dikenali dengan benar.

2. Error Rate

Tingkat kesalahan didapatkan dari percobaan ini adalah 35,5556% , ini artinya dari 360 *images* terdapat 128 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.1

Tabel 4.1. Hasil Percobaan 10% *Training Images* dan 90% *Testing Images* dengan 1D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	18	3228,8030
1_2	1	1	1972,0305
1_3	1	18	2562,1538
1_4	1	18	2485,5085
1_5	1	1	2481,6122
1_6	1	1	1634,0377
1_7	1	1	2959,9710
1_8	1	18	2487,5949
1_9	1	18	2413,1154
2_1	2	2	1324,2476

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.2.

Tabel 4.2. Hasil Perhitungan 10% *Training Images* dan 90% *Testing Images* dengan 1D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	360
Jumlah Citra Uji yang Dikenali	232
Jumlah Citra Uji yang Tidak Dikenali	128
Akurasi (%)	64,4444
<i>Error Rate (%)</i>	35,5556

- Metode *Wavelet*+1D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 68,6111% , ini artinya dari 360 *images* terdapat 247 *images* yang dikenali dengan benar.

2. Error Rate

Tingkat kesalahan didapatkan dari percobaan ini adalah 13,3889%, ini artinya dari 360 *images* terdapat 113 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.3.

Tabel 4.3. Hasil Percobaan 10% *Training Images* dan 90% *Testing Images* dengan *Wavelet +1D-PCA+KNN*

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidian Distance</i>
1_1	1	32	5444,3036
1_2	1	1	3412,4733
1_3	1	36	5016,6489
1_4	1	12	3964,0032
1_5	1	36	5376,2886
1_6	1	1	2386,3298
1_7	1	2	3946,2082
1_8	1	36	4053,8955
1_9	1	1	3542,4145
2_1	2	2	2207,9208

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.4.

Tabel 4.4. Hasil Perhitungan 10% *Training Images* dan 90% *Testing Images* dengan *Wavelet+1D-PCA +KNN*

Parameter	Hasil
Jumlah Citra Uji	360
Jumlah Citra Uji yang Dikenali	247
Jumlah Citra Uji yang Tidak Dikenali	113
Akurasi (%)	68,6111
<i>Error Rate</i> (%)	31,3889

- Metode 2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 74,4444%, ini artinya dari 360 *images* terdapat 268 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 25,5556%, ini artinya dari 360 *images* terdapat 92 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.5.

Tabel 4.5. Hasil Percobaan 10% *Training Images* dan 90% *Testing Images* dengan 2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	32	8947,6654
1_2	1	1	6929,8683
1_3	1	18	9596,0867
1_4	1	18	7921,5699
1_5	1	1	8654,5128
1_6	1	1	6442,5243
1_7	1	1	8097,3735
1_8	1	18	9928,7578
1_9	1	17	6092,2518
2_1	2	2	4603,1397

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.6.

Tabel 4.6. Hasil Perhitungan 10% *Training Images* dan 90% *Testing Images* dengan 2D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	360
Jumlah Citra Uji yang Dikenali	268
Jumlah Citra Uji yang Tidak Dikenali	92
Akurasi(%)	74,4444
<i>Error Rate</i> (%)	25,5556

- Metode *Wavelet*+2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 81,9444%, ini artinya dari 360 *images* terdapat 295 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 18,0556%, ini artinya dari 360 *images* terdapat 155 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.7.

Tabel 4.7. Hasil Percobaan 10% *Training Images* dan 90% *Testing Images* dengan

Wavelet+2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	32	13997,0593
1_2	1	1	10130,9374
1_3	1	18	14933,4036
1_4	1	18	12209,7105
1_5	1	1	14536,4262
1_6	1	1	8407,2910
1_7	1	1	19786,1352
1_8	1	18	13900,8484
1_9	1	17	10569,8985
2_1	2	2	6158,2597

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.8.

Tabel 4.8. Hasil Perhitungan 10% *Training Images* dan 90% *Testing Images* dengan *Wavelet*+2D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	360
Jumlah Citra Uji yang Dikenali	295
Jumlah Citra Uji yang Tidak Dikenali	155
Akurasi(%)	81.9444
<i>Error Rate</i> (%)	18.0556

b. Pengujian Menggunakan 50% *Training Images* dan 50% *Testing Images*

Pengujian kedua menggunakan 50% citra latih yakni sebanyak 200 citra dan 50% citra uji yakni sebanyak 200 citra. Pengaturan awal pada percobaan ini ditunjukkan oleh Gambar 4.15.

```
Num_Class←40;
No_SampleClass←10;
No_TrainSamples←5;
No_TestSamples←5;
```

Gambar 4.15. Pengaturan Percobaan 50% *Training Images* dan 50% *Testing Images*

▪ Metode 1D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 84% , ini artinya dari 200 *images* terdapat 168 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 16% , ini artinya dari 200 *images* terdapat 32 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.9.

Tabel 4.9. Hasil Percobaan 50% *Training Images* dan 50% *Testing Images* dengan 1D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	2095,9148
1_2	1	1	1643,1006
1_3	1	16	2061,0761
1_4	1	1	1742,996
1_5	1	1	1949,8265
2_1	2	2	869,6467
2_2	2	2	868,8359
2_3	2	2	283,1424
2_4	2	2	1183,3647
2_5	2	2	751,4609

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.10.

Tabel 4.10. Hasil Perhitungan 50% *Training Images* dan 50% *Testing Images* dengan 1D-PCA +KNN

Parameter	Hasil
Jumlah Citra Uji	200
Jumlah Citra Uji yang Dikenali	168
Jumlah Citra Uji yang Tidak Dikenali	32
Akurasi(%)	84
<i>Error Rate</i> (%)	16

- Metode Wavelet+1D-PCA dan KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 86,5%, ini artinya dari 200 *images* terdapat 173 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 13,5%, ini artinya dari 200 *images* terdapat 17 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.11.

Tabel 4.11. Hasil Percobaan 50% *Training Images* dan 50% *Testing Images* dengan Wavelet+1D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	3325,5867
1_2	1	1	2464,4184
1_3	1	1	2891,8895
1_4	1	1	2865,2179
1_5	1	16	2913,7032
2_1	2	2	1527,7462
2_2	2	2	1875,7669
2_3	2	2	999,1235
2_4	2	2	1660,8826
2_5	2	2	2021,5845

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.12.

Tabel 4.12. Hasil Perhitungan 50% *Training Images* dan 50% *Testing Images* dengan *Wavelet+1D-PCA+KNN*

Parameter	Hasil
Jumlah Citra Uji	200
Jumlah Citra Uji yang Dikenali	173
Jumlah Citra Uji yang Tidak Dikenali	37
Akurasi (%)	86,5
<i>Error Rate</i> (%)	13,5

- Metode 2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 92,5% , ini artinya dari 200 *images* terdapat 185 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 7,5% , ini artinya dari 200 *images* terdapat 15 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.13.

Tabel 4.13. Hasil Percobaan 50% *Training Images* dan 50% *Testing Images* dengan 2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	6617,8149
1_2	1	1	6513,8709
1_3	1	1	7214,0765
1_4	1	1	7271,3365
1_5	1	17	6652,8522
2_1	2	2	3909,7338
2_2	2	2	3654,0128
2_3	2	2	2157,3138
2_4	2	2	4253,1368
2_5	2	2	2983,6531

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.14.

Tabel 4.14. Hasil Perhitungan 50% *Training Images* dan 50% *Testing Images* dengan 2D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	200
Jumlah Citra Uji yang Dikenali	185
Jumlah Citra Uji yang Tidak Dikenali	15
Akurasi (%)	92,5
<i>Error Rate</i> (%)	7,5

- Metode *Wavelet*+2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 94,5%, ini artinya dari 360 *images* terdapat 189 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 5,5%, ini artinya dari 360 *images* terdapat 11 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.15.

Tabel 4.15. Hasil Percobaan 50% *Training Images* dan 50% *Testing Images* dengan *Wavelet*+2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	10051,3684
1_2	1	1	8574,1706
1_3	1	1	9270,5253
1_4	1	1	9369,1698
1_5	1	1	9178,3106
2_1	2	2	5373,0592
2_2	2	2	5836,7524
2_3	2	2	3683,1002
2_4	2	2	7249,0303
2_5	2	2	5509,1923

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.12

Tabel 4.16. Hasil Perhitungan 50% *Training Images* dan 50% *Testing Images* dengan *Wavelet+2D-PCA+KNN*

Parameter	Hasil
Jumlah Citra Uji	200
Jumlah Citra Uji yang Dikenali	189
Jumlah Citra Uji yang Tidak Dikenali	11
Akurasi (%)	94,5
<i>Error Rate</i> (%)	5,5

c. Pengujian Menggunakan 90% *Training Images* dan 10% *Testing Images*

Pengujian ini menggunakan 90% citra latih yakni sebanyak 360 citra dan 90% citra uji yakni sebanyak 40 citra. Pengaturan awal pada percobaan ini ditunjukkan oleh Gambar 4.16.

```
Num_Class←40;
No_SampleClass←10;
No_TrainSamples←9;
No_TestSamples←1;
```

Gambar 4.16. Pengaturan Percobaan 10% *Training Images* dan 90% *Testing Images*

▪ Metode 1D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 92,5% , ini artinya dari 40 *images* terdapat 37 *images* yang dikenali dengan benar.

3. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 7,5% , ini artinya dari 40 *images* terdapat 3 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.17.

Tabel 4.17. Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan 1D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	1748,6593
2_1	2	2	574,6993
3_1	3	3	521,6226
4_1	4	4	748,4565
5_1	5	18	1275,5793
6_1	6	6	373,9567
7_1	7	7	943,0408
8_1	8	8	577,6611
9_1	9	9	839,3134
10_1	10	38	2024,5329

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.18.

Tabel 4.18. Hasil Perhitungan 90% *Training Images* dan 10% *Testing Images* dengan 1D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	40
Jumlah Citra Uji yang Dikenali	37
Jumlah Citra Uji yang Tidak Dikenali	3
Akurasi(%)	92,5
<i>Error Rate</i> (%)	7,5

- Metode *Wavelet*+1D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 95%, ini artinya dari 40 *images* terdapat 38 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 4%, ini artinya dari 40 *images* terdapat 2 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.19.

Tabel 4.19. Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan *Wavelet+1D-PCA +KNN*

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	16	2942,5782
2_1	2	2	1597,3887
3_1	3	3	654,2113
4_1	4	4	819,9438
5_1	5	5	1195,7727
6_1	6	6	589,8915
7_1	7	7	853,1899
8_1	8	8	689,9506
9_1	9	9	545,3984
10_1	10	10	2628,7090

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.20.

Tabel 4.20. Hasil Perhitungan 90% *Training Images* dan 10% *Testing Images* dengan *Wavelet+1D-PCA +KNN*

Parameter	Hasil
Jumlah Citra Uji	40
Jumlah Citra Uji yang Dikenali	38
Jumlah Citra Uji yang Tidak Dikenali	2
Akurasi (%)	95
<i>Error Rate</i> (%)	5

- Metode 2D-PCAdan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 92,5% , ini artinya dari 40 *images* terdapat 37 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 7,5% , ini artinya dari 40 *images* terdapat 3 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.21.

Tabel 4.21. Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan 2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	17	6664,589
2_1	2	2	2536,6421
3_1	3	3	3052,8538
4_1	4	4	3871,1251
5_1	5	40	4525,0406
6_1	6	6	2328,7479
7_1	7	7	4247,4737
8_1	8	8	3645,2352
9_1	9	9	4696,9284
10_1	10	38	7695,9353

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.22.

Tabel 4.22. Hasil Perhitungan 90% *Training Images* dan 10% *Testing Images* dengan 2D-PCA +KNN

Parameter	Hasil
Jumlah Citra Uji	40
Jumlah Citra Uji yang Dikenali	37
Jumlah Citra Uji yang Tidak Dikenali	3
Akurasi (%)	92,5
<i>Error Rate</i> (%)	7,5

- Metode *Wavelet*+2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 97,5%, ini artinya dari 40 *images* terdapat 39 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 5,5%, ini artinya dari 40 *images* terdapat 1 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.23.

Tabel 4.23. Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan *Wavelet+2D-PCA+KNN*

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	9067,2753
2_1	2	2	4296,4452
3_1	3	3	3014,0086
4_1	4	4	4343,4193
5_1	5	5	5225,066
6_1	6	6	2355,9597
7_1	7	7	3721,9896
8_1	8	8	3057,7524
9_1	9	9	4869,7728
10_1	10	8	8867,797

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.24.

Tabel 4.24. Hasil Perhitungan 90% *Training Images* dan 10% *Testing Images* dengan *Wavelet+2D-PCA+KNN*

Parameter	Hasil
Jumlah Citra Uji	40
Jumlah Citra Uji yang Dikenali	39
Jumlah Citra Uji yang Tidak Dikenali	1
Akurasi (%)	97,5
<i>Error Rate</i> (%)	2,5

4.1.2.2. Pengujian Menggunakan YALE

a. Pengujian Menggunakan 1 *Training Images* dan 10 *Testing Images*

Pengujian ini menggunakan 1 citra latih dan 10 citra uji. Pengaturan awal pada percobaan ini ditunjukkan oleh Gambar 4.17.

```
Num_Class←15;
No_SampleClass←11;
No_TrainSamples←1;
No_TestSamples←10;
```

Gambar 4.17. Pengaturan Percobaan 1 *Training Images* dan 10 *Testing Images*

- Metode 1D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 46,667% , ini artinya dari 150 *images* terdapat 73 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 53,333% , ini artinya dari 150 *images* terdapat 77 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.25.

Tabel 4.25. Hasil Percobaan 1 *Training Images* dan 10 *Testing Images* dengan
1D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	9043,1595
1_2	1	1	8723,3480
1_3	1	1	6980,2270
1_4	1	2	7922,4496
1_5	1	1	8451,8240
1_6	1	7	17677,8501
1_7	1	1	8517,9394
1_8	1	1	7484,4676
1_9	1	1	7857,000
1_10	1	1	7069,4236

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 3.

Hasil perhitungan ditunjukkan oleh Tabel 4.26.

Tabel 4.26. Hasil Perhitungan 1 *Training Images* dan 10 *Testing Images* dengan
1D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	150
Jumlah Citra Uji yang Dikenali	73
Jumlah Citra Uji yang Tidak Dikenali	77
Akurasi (%)	46,667
<i>Error Rate (%)</i>	51,333

- Metode *Wavelet*+1D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 50%, ini artinya dari 150 *images* terdapat 75 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 50%, ini artinya dari 150 *images* terdapat 75 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.27.

Tabel 4.27. Hasil Percobaan 1 *Training Images* dan 10 *Testing Images* dengan *Wavelet*+1D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	9077,0738
1_2	1	1	8769,4656
1_3	1	1	9059,5072
1_4	1	2	7638,7788
1_5	1	1	8454,0195
1_6	1	7	17848,8981
1_7	1	1	8541,8922
1_8	1	1	7152,9331
1_9	1	1	7848,6348
1_10	1	1	7086,2483

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 3.

Hasil perhitungan ditunjukkan oleh Tabel 4.28.

Tabel 4.28. Hasil Perhitungan 1 *Training Images* dan 10 *Testing Images* dengan *Wavelet*+1D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	150
Jumlah Citra Uji yang Dikenali	75
Jumlah Citra Uji yang Tidak Dikenali	75
Akurasi (%)	50
<i>Error Rate</i> (%)	50

- Metode 2D-PCA dan klasifikasi KNN

- Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 54,6667% , ini artinya dari 150 *images* terdapat 82 *images* yang dikenali dengan benar.

- Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 46,3337% , ini artinya dari 150 *images* terdapat 38 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.29.

Tabel 4.29. Hasil Percobaan 1 *Training Images* dan 10 *Testing Images* dengan 2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	25133,7165
1_2	1	1	24976,6049
1_3	1	1	29175,8700
1_4	1	12	22058,193
1_5	1	1	25437,6051
1_6	1	8	46807,0823
1_7	1	1	23878,0676
1_8	1	1	22234,5222
1_9	1	1	24056,3268
1_10	1	1	21116,7999

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 3.

Hasil perhitungan ditunjukkan oleh Tabel 4.30.

Tabel 4.30. Hasil Perhitungan 10% *Training Images* dan 90% *Testing Images* dengan 2D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	150
Jumlah Citra Uji yang Dikenali	82
Jumlah Citra Uji yang Tidak Dikenali	58
Akurasi (%)	54,667
<i>Error Rate</i> (%)	45,333

- Metode *Wavelet*+2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 56% , ini artinya dari 150 *images* terdapat 84 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 44% , ini artinya dari 150 *images* terdapat 72 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.31

Tabel 4.31. Hasil Percobaan 1 *Training Images* dan 10 *Testing Images* dengan *Wavelet*+2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	26273,4828
1_2	1	1	25842,9453
1_3	1	1	37504,0281
1_4	1	2	22646,0477
1_5	1	1	26247,9288
1_6	1	8	49122,6850
1_7	1	1	25239,4868
1_8	1	1	23096,0233
1_9	1	1	23918,0642
1_10	1	1	21584,6595

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 3.

Hasil perhitungan ditunjukkan oleh Tabel 4.32.

Tabel 4.32. Hasil Perhitungan 1 *Training Images* dan 10 *Testing Images* dengan *Wavelet*+2D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	150
Jumlah Citra Uji yang Dikenali	84
Jumlah Citra Uji yang Tidak Dikenali	67
Akurasi(%)	56
<i>Error Rate</i> (%)	44

b. Pengujian Menggunakan 6 *Training Images* dan 5 *Testing Images*

Pengujian kedua menggunakan 6 citra latih dan 6 citra uji. Pengaturan awal pada percobaan ini ditunjukkan oleh Gambar 4.18.

```
Num_Class←15;
No_SampleClass←11;
No_TrainSamples←6;
No_TestSamples←5;
```

Gambar 4.18. Pengaturan Percobaan 6 *Training Images* dan 5 *Testing Images*

▪ Metode 1D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 81,3333%, ini artinya dari 75 *images* terdapat 61 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 19,6667%, ini artinya dari 75 *images* terdapat 14 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.33.

Tabel 4.33. Hasil Percobaan 6 *Training Images* dan 5 *Testing Images* dengan 1D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	6	17818,5102
1_2	1	1	1039,6784
1_3	1	1	1729,6801
1_4	1	1	3703,4842
1_5	1	1	3912,4024
2_1	2	4	10471,8616
2_2	2	2	1833,9824
2_3	2	2	332,0957
2_4	2	2	3858,5551
2_5	2	2	4420,3525

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 3.

Hasil perhitungan ditunjukkan oleh Tabel 4.34.

Tabel 4.34. Hasil Perhitungan 6 *Training Images* dan 5 *Testing Images* dengan 1D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	75
Jumlah Citra Uji yang Dikenali	61
Jumlah Citra Uji yang Tidak Dikenali	14
Akurasi(%)	81,3333
<i>Error Rate</i> (%)	19,6667

- Metode Wavelet-1D-PCA dan klasifikasi KNN

- Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 81,3333%, ini artinya dari 75 *images* terdapat 61 *images* yang dikenali dengan benar.

- *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 19,6667%, ini artinya dari 75 *images* terdapat 14 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.35.

Tabel 4.35. Hasil Percobaan 6 *Training Images* dan 5 *Testing Images* dengan Wavelet-1D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	6	27357,0464
1_2	1	1	1017,7336
1_3	1	1	1615,7264
1_4	1	1	3462,8403
1_5	1	1	3625,0878
2_1	2	4	11212,4957
2_2	2	2	1644,3802
2_3	2	2	320,3679
2_4	2	2	3566,2291
2_5	2	2	4118,4979

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.

Hasil perhitungan ditunjukkan oleh Tabel 4.36.

Tabel 4.36. Hasil Perhitungan 6 *Training Images* dan 5 *Testing Images* dengan *Wavelet-1D-PCA+KNN*

Parameter	Hasil
Jumlah Citra Uji	75
Jumlah Citra Uji yang Dikenali	61
Jumlah Citra Uji yang Tidak Dikenali	14
Akurasi	81,3333
<i>Error Rate</i>	19,6667

- Metode 2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 86,6667%, ini artinya dari 75 *images* terdapat 65 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 13,3333%, ini artinya dari 75 *images* terdapat 10 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.37

Tabel 4.37. Hasil Percobaan 6 *Training Images* dan 5 *Testing Images* dengan 2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	6	50153,2634
1_2	1	1	8486,1120
1_3	1	1	8018,0573
1_4	1	1	13200,0715
1_5	1	1	11863,5796
2_1	2	2	33971,1342
2_2	2	2	7461,6693
2_3	2	2	2228,4284
2_4	2	2	15639,2599
2_5	2	2	9362,7048

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 3.

Hasil perhitungan ditunjukkan oleh Tabel 4.38.

Tabel 4.38. Hasil Perhitungan 6 *Training Images* dan 5 *Testing Images* dengan 2D-PCA +KNN

Parameter	Hasil
Jumlah Citra Uji	75
Jumlah Citra Uji yang Dikenali	65
Jumlah Citra Uji yang Tidak Dikenali	10
Akurasi	86,6667
<i>Error Rate</i>	13,3333

▪ Metode *Wavelet*+2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 86,6667% , ini artinya dari 75 *images* terdapat 65 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 13,3333% , ini artinya dari 75 *images* terdapat 10 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.39.

Tabel 4.39. Hasil Percobaan 6 *Training Images* dan 5 *Testing Images* dengan *Wavelet*+2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	6	50661,7686
1_2	1	1	7433,4260
1_3	1	1	7028,7806
1_4	1	1	11944,1853
1_5	1	1	10906,7796
2_1	2	4	36832,0743
2_2	2	2	6749,9479
2_3	2	2	1791,8111
2_4	2	2	15928,4160
2_5	2	2	8834,7410

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 3.

Hasil perhitungan ditunjukkan oleh Tabel 4.40.

Tabel 4.40. Hasil Perhitungan 50% *Training Images* dan 50% *Testing Images* dengan *Wavelet+2D-PCA+KNN*

Parameter	Hasil
Jumlah Citra Uji	75
Jumlah Citra Uji yang Dikenali	65
Jumlah Citra Uji yang Tidak Dikenali	10
Akurasi (%)	86,6667
<i>Error Rate</i> (%)	13,3333

c. Pengujian Menggunakan 10 *Training Images* dan 1 *Testing Images*

Pengujian ini menggunakan 90% citra latih yakni sebanyak 360 citra dan 90% citra uji yakni sebanyak 40 citra. Pengaturan awal ditunjukkan oleh Gambar 4.19.

```

Num_Class←40;
No_SampleClass←11;
No_TrainSamples←10;
No_TestSamples←1;

```

Gambar 4.19. Pengaturan Percobaan 10 *Training Images* dan 1 *Testing Images*

▪ Metode 1D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 93,3333% , ini artinya dari 15 *images* terdapat 14 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 6,6667%, ini artinya dari 15 *images* terdapat 1 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.41.

Tabel 4.41. Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan 1D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidian Distance</i>
1_1	1	1	1872,0887
2_1	2	2	2793,0378
3_1	3	3	1236,2666
4_1	4	4	1026,4637

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidian Distance</i>
5_1	5	5	558,4457
6_1	6	6	966,5066
7_1	7	7	1611,1848
8_1	8	10	4318,8841
9_1	9	9	3139,8519
10_1	10	10	1591,2099

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 23

Hasil perhitungan ditunjukkan oleh Tabel 4.42.

Tabel 4.42. Hasil Perhitungan 10 *Training Images* dan 1 *Testing Images* dengan 1D-PCA +KNN

Parameter	Hasil
Jumlah Citra Uji	15
Jumlah Citra Uji yang Dikenali	14
Jumlah Citra Uji yang Tidak Dikenali	1
Akurasi(%)	93,3333
<i>Error Rate</i> (%)	6,6667

- Metode *Wavelet PCA* dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 93,3333% , ini artinya dari 15 *images* terdapat 14 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 6,6667% , ini artinya dari 15 *images* terdapat 1 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.43.

Tabel 4.43. Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan *Wavelet+1D-PCA+KNN*

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	1812,6937
2_1	2	2	2429,8504
3_1	3	3	1067,4464
4_1	4	4	1208,9305
5_1	5	5	529,4681
6_1	6	6	1126,4688
7_1	7	7	1572,6393

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
8_1	8	10	4562,3550
9_1	9	9	2841,3569
10_1	10	10	1480,6092

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 3.

Hasil perhitungan ditunjukkan oleh Tabel 4.44.

Tabel 4.44. Hasil Perhitungan 10 *Training Images* dan 1 *Testing Images* dengan *Wavelet+1D-PCA+KNN*

Parameter	Hasil
Jumlah Citra Uji	15
Jumlah Citra Uji yang Dikenali	14
Jumlah Citra Uji yang Tidak Dikenali	1
Akurasi	93,3333
<i>Error Rate</i>	6,6667

- Metode 2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 100% , ini artinya dari 15 *images* terdapat 15 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 100% , ini artinya dari 15 *images* terdapat 0 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh tabel 4.45.

Tabel 4.45. Hasil Percobaan 10 *Training Images* dan 1 *Testing Images* dengan 2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	8660,7126
2_1	2	2	8380,7358
3_1	3	3	5714,5878
4_1	4	4	6916,1544
5_1	5	5	3605,4198
6_1	6	6	7034,3670
7_1	7	7	12515,5519
8_1	8	8	16349,8114
9_1	9	9	11384,1015
10_1	10	10	6257,3567

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 3.

Hasil perhitungan ditunjukkan oleh Tabel 4.46.

Tabel 4.46. Hasil Perhitungan 10 *Training Images* dan 1 *Testing Images* dengan 2D-PCA+KNN

Parameter	Hasil
Jumlah Citra Uji	40
Jumlah Citra Uji yang Dikenali	37
Jumlah Citra Uji yang Tidak Dikenali	3
Akurasi	92,5
<i>Error Rate</i>	7,5

- Metode *Wavelet*+2D-PCA dan klasifikasi KNN

1. Tingkat Akurasi

Tingkat akurasi yang didapatkan dari percobaan ini adalah 100%, ini artinya dari 15 *images* terdapat 15 *images* yang dikenali dengan benar.

2. *Error Rate*

Tingkat kesalahan didapatkan dari percobaan ini adalah 100%, ini artinya dari 15 *images* terdapat 0 *images* yang tidak dikenali dengan benar.

Hasil pengenalan wajah ditunjukkan oleh Tabel 4.47.

Tabel 4.47. Hasil Percobaan 10 *Training Images* dan 1 *Testing Images* dengan *Wavelet*+2D-PCA+KNN

<i>Test Image</i>	Kelas Uji	Kelas Latih	<i>Euclidian Distance</i>
1_1	1	1	9067,2753
2_1	2	2	4296,4452
3_1	3	3	3014,0086
4_1	4	4	4343,4193
5_1	5	5	5225,066
6_1	6	6	2355,9597
7_1	7	7	3721,9896
8_1	8	8	3057,7524
9_1	9	9	4869,7728
10_1	10	8	8867,797

Hasil perhitungan selengkapnya dapat dilihat pada Lampiran 2.




Hasil perhitungan ditunjukkan oleh Tabel 4.48.

Tabel 4.48. Hasil Perhitungan 10 *Training Images* dan 1 *Testing Images* dengan *Wavelet+2D-PCA +KNN*

Parameter	Hasil
Jumlah Citra Uji	15
Jumlah Citra Uji yang Dikenali	15
Jumlah Citra Uji yang Tidak Dikenali	0
Akurasi (%)	100
<i>Error Rate (%)</i>	0

Selanjutnya dilakukan rekapitulasi citra yang dominan mengalami kesalahan dalam proses pengenalan. Hasil rekapitulasi dapat dilihat dalam Tabel 4.49.

Tabel 4.49. Rekapitulasi Citra Dominan *Error*

<i>Dataset</i>	Gambar	Penyebab & Penanganan
AT & T		Fitur alis tidak dapat dikenali dengan maksimal Penanganan : <i>contrast enhancement</i>
	<i>Subject : 1 ; ID : 10</i> 	Satu fitur alis terhalang kacamata dan pengambilan citra yang terlalu miring Penanganan : kemiringan pengambilan citra dikurangi
	<i>Subject : 10 ; ID : 10</i> 	Terdapat kerutan wajah yang berbeda dengan citra di kelas yang sama Penanganan : <i>smoothing</i>
	<i>Subject : 10; ID : 10</i>	

YALE

*Subject : 2; ID : 7**Subject : 8; ID : 10*

Bayangan sebelah kiri
membuat ekstraksi fitur mata
kiri sulit
Penanganan : *brightness
enhancement*

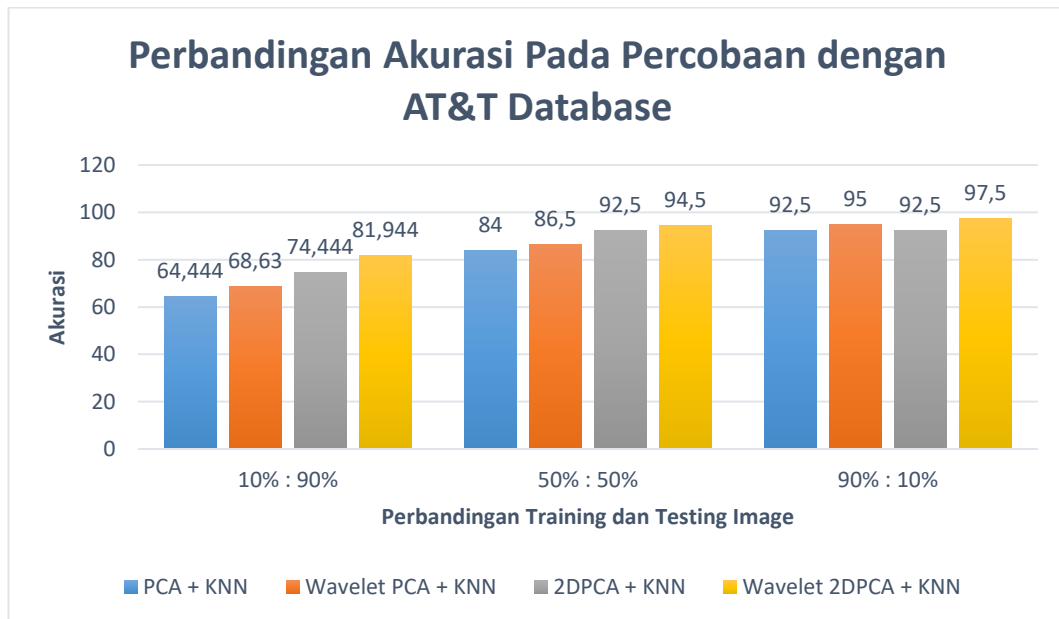
Pantulan cahaya di kacamata
Penanganan : menghindari
penggunaan *flash* / cahaya
berlebih dalam pengambilan
gambar

Dari percobaan diatas, didapatkan rekapitulasi hasil pengenalan wajah yang dapat dilihat dalam Tabel 4.49.

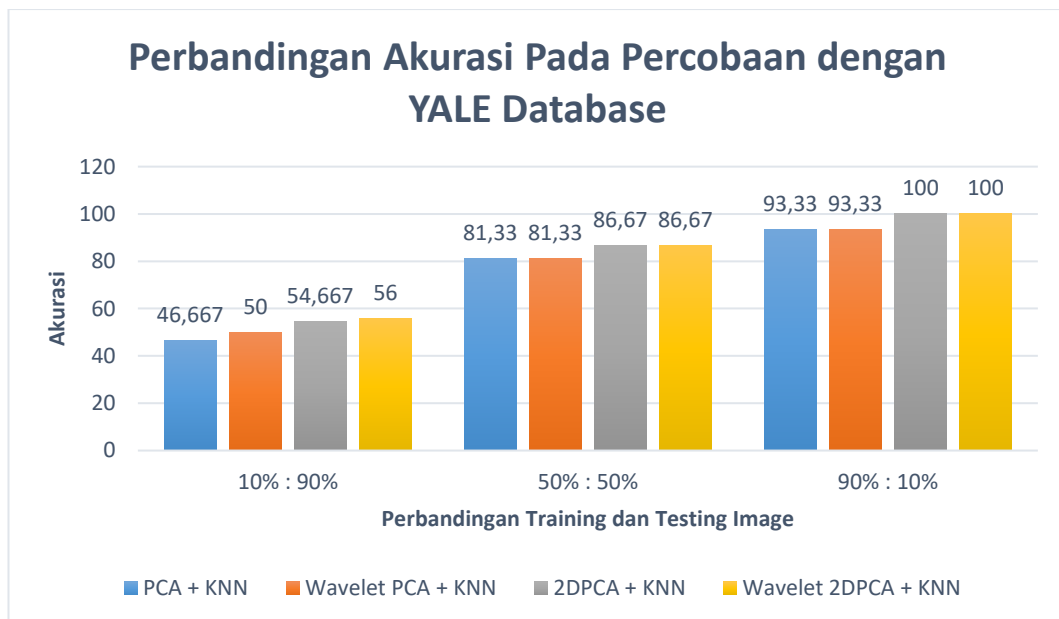
Tabel 4.49. Rekapitulasi Hasil Percobaan

Jenis Percobaan		Hasil Akurasi 1D-PCA + KNN	Hasil Akurasi <i>Wavelet+1D-</i> <i>PCA + KNN</i>	Hasil Akurasi 2D-PCA + KNN	Hasil Akurasi <i>Wavelet+2D-</i> <i>PCA + KNN</i>
AT&T	10% <i>Training</i> <i>Images</i> dan 90% <i>Testing Images.</i>	64,444%	68,63%	74,444%	81,944%
	50% <i>Training</i> <i>Images</i> dan 50% <i>Testing Images.</i>	84%	86,5%	92,5%	94,5%
	90% <i>Training</i> <i>Images</i> Dan 10% <i>Testing Images.</i>	92,5%	95%	92,5%	97,5%
YALE	10% <i>Training</i> <i>Images</i> dan 90% <i>Testing Images.</i>	46,667%	50%	54,667%	56%
	50% <i>Training</i> <i>Images</i> dan 50% <i>Testing Images.</i>	81,33%	81,33%	86,67%	86,67%
	90% <i>Training</i> <i>Images</i> Dan 10% <i>Testing Images.</i>	93,33%	93,33%	100%	100%

Grafik hasil akurasi ditunjukkan oleh Gambar 4.20. dan Gambar 4.21



Gambar 4.20. Grafik Perbandingan Akurasi Percobaan Menggunakan AT & T



Gambar 4.21. Grafik Perbandingan Akurasi Percobaan Menggunakan YALE

4.2. Pembahasan

Peningkatan akurasi pengenalan wajah dapat dilakukan dengan menerapkan dekomposisi *wavelet reverse biorthogonal* sebanyak dua level terhadap citra sebelum citra diproses. Melalui proses tersebut, *range* citra menjadi semakin kecil sehingga piksel-piksel non wajah dapat berkurang tanpa mengurangi fitur-fitur penting bagi citra. Hal ini akan memudahkan proses ekstraksi ciri pada tahap selanjutnya.

Ekstraksi fitur dilakukan untuk mengenali informasi-informasi penting yang terdapat dalam wajah. Semakin banyak fitur yang didapatkan, maka hasil pengenalan wajah akan semakin akurat. Dalam percobaan sebelumnya dilakukan ekstraksi fitur dengan dua jenis PCA yakni 1D-PCA dan 2D-PCA. Keduanya diterapkan pada citra pada *dataset* asli maupun yang telah melalui proses dekomposisi *wavelet*. Hasil ekstraksi fitur adalah *eigenface* pada masing-masing citra, baik citra latih maupun citra uji.

Eigenface citra uji akan dibandingkan dengan citra latih untuk mendapatkan kelas citra uji. Perhitungan ini dilakukan melalui KNN dengan perhitungan jarak *Euclidean face*. Semakin kecil jarak *Euclidean* maka citra akan semakin serupa. Citra latih dengan jarak yang paling kecil akan menjadi kelas citra uji.

Hasil percobaan menggunakan AT&T menunjukkan bahwa dekomposisi *wavelet* meningkatkan hasil akurasi wajah pada ekstraksi fitur PCA. Pada 1D-PCA, dekomposisi *wavelet* terbukti mampu meningkatkan akurasi sebesar 4,19% pada rasio 10%:90%, 2% pada rasio 50%:50%, dan 2,5% pada rasio 90%:10%.

Sedangkan pada 2D-PCA, dekomposisi *wavelet* terbukti mampu meningkatkan akurasi sebesar 7,5% pada rasio 10%:90%, 2% pada rasio 50%:50%, dan 5% pada rasio 90%:10%. Peningkatan akurasi akan semakin kecil pada pengenalan wajah dengan jumlah *training images* yang lebih besar, karena pengetahuan sebagai dasar pemrosesan akan semakin seragam. Selain itu, jumlah *testing image* yang semakin sedikit akan mengurangi kemungkinan *error* yang terjadi.

Berbeda dengan percobaan pada AT&T *faces*, pada percobaan dengan YALE *faces* dekomposisi *wavelet* hanya berhasil meningkatkan akurasi pada percobaan dengan rasio 10%:90% yakni pada 1D-PCA sebesar 3,3333%, dan 2% pada 2D-PCA. Hal ini karena jumlah citra pada YALE *faces* lebih beragam sehingga referensi kelompok citra uji dapat lebih baik dari AT&T. Namun, dari percobaan YALE *faces* dapat dibuktikan bahwa *wavelet* mampu meningkatkan performa akurasi pengenalan wajah bahkan pada citra dengan aksesoris kaca mata hitam. Artinya, pengenalan wajah dapat tetap berjalan tanpa objek mata.

Pada pengenalan wajah YALE dengan rasio 50%:50% dan 90% dan 90% *wavelet* tidak meningkatkan akurasi wajah. Akurasi yang didapatkan dari pengenalan wajah dengan dan tanpa dekomposisi *wavelet* tetap sama yakni pada angka 81,33% pada 1D PCA, & 86,67% pada 2D-PCA untuk rasio 50%:50%, serta 93,33% pada 1D PCA, dan 100% pada 2D-PCA. Meskipun demikian, dekomposisi *wavelet* memberikan keuntungan pada sudut pandang yang lain yakni memori penyimpanan. Dekomposisi *wavelet* dapat mengurangi ukuran matriks citra dari semula 243x320x165 menjadi 72x91x165. Hal ini bermanfaat pada menurunnya memori penyimpanan yang diperlukan. Perbandingan ukuran sebelum dan setelah

dilakukan dekomposisi dengan *reverse biorthogonal wavelet* dapat dilihat pada Tabel 4.50.

Tabel 4.50 Perbandingan Ukuran Citra

<i>Dataset</i>	Ukuran Asli	Ukuran Setelah Dekomposisi
AT&T	92×112×400	39×34×400
YALE	243×320×165	72×91×165

Dari 24 percobaan yang dilakukan, dekomposisi *wavelet* terbukti meningkatkan akurasi pengenalan sebanyak 16 percobaan. Pada 8 percobaan yang lain *wavelet* tidak mampu meningkatkan maupun menurunkan akurasi pengenalan wajah sehingga walaupun tidak memberikan keuntungan dari segi akurasi, dekomposisi *wavelet* mampu memberikan manfaat pada besar memori penyimpanan.

Hasil akurasi pengenalan wajah dengan metode yang diusulkan dibandingkan dengan hasil penelitian-penelitian terdahulu dengan *ratio* yang telah ditentukan pada *dataset*. Hasil perbandingan akurasi pengenalan wajah dengan penelitian terdahulu dapat dilihat pada Tabel 4.51.

Tabel 4.51. Perbandingan Hasil Akurasi Metode yang Diusulkan dengan Penelitian Terdahulu

<i>Dataset</i>	<i>Ratio</i>	Metode	<i>Accuracy</i>
AT&T	8:2	IKLDA + PNN (Ouyang Et al, 2019: 5)	97,22%
		Metode yang diusulkan	97,5%
	5:5	(MF, GF, HE)-PCA- multiSVMs (Maw <i>et al</i> , 2019: 124)	91.6%
		Metode yang diusulkan	94,5%
	6:4	RSA+SHA256+LBP (Ahmed <i>et al</i> , 2019 : 299)	97,5%
		2D-PCA + KNN (Sutarti <i>et al</i> , 2019)	96,88%

<i>Dataset</i>	<i>Ratio</i>	<i>Metode</i>	<i>Accuracy</i>
		Weight Sum Rule (Rahmi <i>et al</i> , 2019:48)	98%
		Metode yang diusulkan	98,75%
YALE	8:3	IKLDA + PNN (Ouyang Et al, 2019)	83,80%
		Metode yang Diusulkan	97,77%

Kelemahan metode pengenalan wajah ini adalah tidak dapat berjalan untuk citra yang berukuran kecil setelah proses *crop* dari *face detection*. *Cropping* akan menyebabkan jangkauan piksel semakin sedikit sehingga proses dekomposisi hanya akan memberikan hasil citra berwarna putih saja. Selain itu, semakin besar *training images* yang diproses akan menyebabkan peningkatan akurasi yang semakin sedikit. Hal ini karena semakin banyaknya pengetahuan yang telah terdefinisi sebagai rujukan hasil pengenalan wajah.

BAB V

PENUTUP

5.1. Kesimpulan

Dari hasil penelitian, pengolahan citra, perancangan, pembuatan, pengujian sistem hingga membandingkan hasil pengenalan wajah menggunakan metode *principal component analysis* (PCA) dengan dekomposisi *multi-level reverse biorthogonal wavelet* dengan klasifikasi KNN, dapat ditarik kesimpulan sebagai berikut:

1. *Multi-level reverse biorthogonal wavelet* dapat melakukan kompresi citra tanpa kehilangan fitur-fitur penting citra. Dekomposisi ini akan membagi citra asal menjadi 4 bagian yakni koefisien LL, LH, HL dan HH. Reduksi ukuran pada citra akan mengurangi jumlah piksel *non* wajah sehingga memudahkan ekstraksi fitur PCA dalam melakukan penentuan *eigenface* pada citra wajah. Hal ini akan meningkatkan jumlah fitur yang didapatkan dan meningkatkan akurasi pengenalan wajah.
2. Implementasi dekomposisi *wavelet* pada citra AT&T terbukti dapat meningkatkan hasil akurasi pengenalan wajah dengan ekstraksi fitur PCA baik 1D-PCA maupun 2D-PCA dan klasifikasi KNN. Pada 1D-PCA, pengenalan wajah dengan rasio 10%:90% menghasilkan akurasi sebesar 64,444% pada citra tanpa dekomposisi *wavelet* dan 68,63% pada citra dengan dekomposisi *wavelet*, pada rasio 50:50% menghasilkan akurasi sebesar 84% pada citra tanpa

dekomposisi *wavelet* dan 86,5% pada citra dengan dekomposisi *wavelet*, dan pada rasion 90%:10% menghasilkan akurasi sebesar 92,5% pada citra tanpa dekomposisi *wavlet* dan 95% pada citra dengan dekomposisi *wavelet*. Sedangkan pada 2D-PCA, pengenalan wajah dengan rasio 10%:90% menghasilkan akurasi sebesar 74,444% pada citra tanpa dekomposisi *wavelet* dan 81,944% pada citra dengan dekomposisi *wavelet*, pada rasio 50:50% menghasilkan akurasi sebesar 92,5% pada citra tanpa dekomposisi *wavlet* dan 94,5% pada citra dengan dekomposisi *wavelet*, dan pada rasion 90%:10% menghasilkan akurasi sebesar 92,5% pada citra tanpa dekomposisi *wavelet* dan 97,5% pada citra dengan dekomposisi *wavelet*.

3. Pada implementasi pada citra YALE, implementasi *wavelet* mampu meningkatkan pengenalan wajah pada rasio 10% : 90% yakni dari 54,6667% pada citra tanpa dekomposisi *wavelet* dan 56% pada citra dengan dekomposisi *wavelet*. Pada rasio 50%:50% tidak terjadi peningkatan akurasi pengenalan wajah. Namun, implementasi *wavelet* tetap memberikan keuntungan dari aspek kebutuhan memori penyimpanan yang lebih kecil.

5.2. Saran

Saran dalam penelitian selanjutnya adalah sebagai berikut.

1. Melakukan percobaan ada *dataset* yang lain sehingga hasil penelitian dapat semakin akurat.

2. Melakukan percobaan dengan rasio yang lebih beragam, terutama untuk YALE *dataset* agar diperoleh informasi mengenai titik jenuh kapan dekomposisi *wavelet* berhenti meningkatkan akurasi pengenalan wajah.
3. Penerapan *adaboost* pada KNN sehingga diperoleh hasil yang lebih baik.

DAFTAR PUSTAKA

- Affandi, S. (2011). Rancang Bangun Aplikasi Pengenalan Wajah Menggunakan Metode Eigenface Yang Berorientasi Pada Principal Component Analysis (PCA). Skripsi Universitas Islam Negeri Sultan Syarif Kasim.
- Ahmed, A., Hasan, T., Abdullatif, F. A., Mustafa, S. T., & Rahim, M. S. M. (2019). A Digital Signature System Based on Real Time Face Recognition. *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*,1,298-302
- Alimjan, G., Sun, T., Liang, Y., Jumahun, H., & Guan, Y. (2018). A New Technique For Remote Sensing Image Classification Based On Combinatorial Algorithm Of SVM And KNN. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(07),1-10.
- Ardhianto, E. (2010). Pengolah Citra Digital Untuk Identifikasi Ciri Sidikjari Berbasis Minutiae. *Jurnal Dinamika Informatika*, 2(1),11-18.
- Aris, B. S., Inna, S., & Maulana, H. (2016). Pengenalan Citra Wajah Sebagai Identifier Menggunakan Principal Component Analysis (PCA). *Jurnal Teknik Informatika*,9(2),166-175
- Budi, A., Suma'inna, & Maulana, H. (2016). Pengenalan Citra Wajah Sebagai Identifier Menggunakan Metode Principal Component Analysis (PCA). *Jurnal Teknik Informatika*, 9(2), 166-175
- Burrus, C. S., Gopinath, R.A & Gue, H. (1998). *Introduction to Wavelets and Wavelet Transforms: A primer*. New Jersey : Prentice Hall
- Dhriti & Kaur, M. (2012). K-Nearest Neighbor Classification Approach for Face and Fingerprint at Feature Level Fusion. *International Journal of Computer Applications*, 60(14),13-17.
- Firdausy, K. (2013). Penjejak Pose Wajah Otomatis pada Sistem Pengenalan Wajah. *Forum Teknik*, 35(1),33-41.
- Gonzalez, R.C., & Woods, R.E., (2008), *Digital Image Processing 3rd Edition*, New York: Pearson Prentice Hall.
- Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques (3rd ed)*. Waltham: Morgan Kaufmann.
- Harjito, B. (1998). *Watermarking of images reconstructed using Information dispersal Algorithm*. Tesis James Cook University of North Queensland
- Hermawati, F. A. (2013). *Pengolahan Citra Digital*. Yogyakarta: Andi
- Hussain, S., Neaman, A. D., Fadl, M. B., & Najuo, R. (2018). Educational Data Mining and Analysis of Student's Academic Performance Using WEKA. *Indonesian Journal of Electrical Engineering and Computer Science*,9(2),447-459

- Johnson, R.A., & Wichern, D.W. (1992). *Applied Multivariate Statistical Analysis, Third Edition*. New Jersey : Prentice Hall
- Kirby, M., & Sirovich, L. (1990). Application of the KL Procedure for the Characterization of Human Faces, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12,103-108
- Kurniawan, L. (2015). Metode Face Recognition untuk Identifikasi Personil Berdasar Citra Wajah bagi Kebutuhan Presensi Online Universitas Negeri Semarang. *Scientific Journal of Informatics*, 1(2), 210-220
- Kustian, N. (2016). Principal Component Analysis untuk Sistem Pengenalan Wajah dengan Menggunakan Metode Eigenface. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 1(2),193-202.
- Linge, G., & Pawar, M. (2014). Neural Network Based Face Recognition Using PCA. *Internation Journal of Computer Science and Informatics Technologies*,5(2), 4011-4014
- Mahalle, R.P., & Nanda, S.K. (2016). A Survey on Analysis of ANN and KNN Classifier for Image Classification with Discrete Wavelet Transform. *International Journal of Innovative Research in Computer and Communication Engineering*. 4(11), 20193-20197.
- Mallat,S. (1999). *A Wavelet Tour Of Signal Processing:The Sparse Way*. Florida:Academic Press
- Marti, N. W. (2010). Pemanfaatan GUI Dalam Pengembangan Perangkat Lunak Pengenalan Citra Wajah Manusia Menggunakan Metode Eigenfaces. *Prosiding Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010)*,4(1),11-16.
- Maw, H. M., Thu, S. M., & Mon, M. T. (2019). Face Recognition based on Illumination Invariant Techniques Model. *2019 International Conference on Advanced Information Technologies (ICAIT)*,7(5),120-125
- Mukhedkar, M. M., & Powalkar, S. B. (2015). Fast Face Recognition Based on Wavelet Transform on PCA. *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET)*, 1(4), 761-764
- Oliveira, L.S., Mansano, M., & Koerich, A.L. (2011). 2D Principal Component Analysis for Face and Facial-Expression Recognition. *Computing in Science & Engineering*, 13(3), 9-13.
- Ouyang, A., Liu, Y., Pei, S., Peng, X., He, M., & Wang, Q. (2019). A Hybrid Improved Kernel LDA And PNN Algorithm for Efficient Face Recognition. *Neurocomputing*,378,1-8
- Patil, S. A., & Deore,P J. (2013). Face Recognition: A Survey. *Informatics Engineering, an International Journal (IEIJ)*,1(1),31-41
- Paulus, E., & Nataliani, Y. (2007).*GUI Matlab*. Yogyakarta: Andi

- Prasad, P. M. K., Prasad, D. Y. V., & Rao, G. S. (2016). Performance Analysis of Orthogonal and Biorthogonal Wavelets for Edge Detection of X-Ray Images. *Procedia Computer Science*, 87, 116–121
- Purnama, B., & Salsabila, D. (2013). Identifikasi Telapak Tangan Menggunakan 2DPCA plus PCA dan K-Nearest Neighbor. *Prosiding Seminar Nasional Aplikasi Teknologi Informasi 2013 (SNATI 2013)*,1,1-6
- Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: Andi Offset
- Rahmi, Z., Ahmad, M. I., Isa, M. N. M., & Khalib, Z. I. A. (2019). Matching Score Level Fusion for Face and Palmprint Recognition System on Spatial Domain. In *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*,1,45-49
- Saragih,E. (2008). Penerapan Daubechies Wavelet dan Hamming Code Dalam Watermarking Citra Digital. Skripsi Universitas Sumatera Utara
- Sinurat, S. (2014). Analisa Sistem Pengenalan Wajah Berbentuk Citra Digital dengan Algoritma Principal Components Analysis. *Jurnal Informasi dan Teknologi Ilmiah (INTI)*, 3(1),112-144
- Soelaiman, R. (2006). Sistem Pengenalan Wajah dengan Penerapan Algoritma Genetika pada Optimasi Basis Eigenface dan Proyeksi Fisherface. Disertasi Universitas Indonesia
- Soni, S. & Sahu, R.S. (2013). Face Recognition Based on Two Dimensional Principal Component Analysis (2DPCA) and Result Comparasion with Diffetent Classifiers. *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, 2(10),3899-3904.
- Sravanthi, A. (2012). A Color Face Recognition Using PCA and KNN Classifier. *International Journal of Research in Advanced Computer Science Engineering*. 3(11): 1-7.
- Suparman, & Marlan. (2007). *Komputer dari Masa Depan (Pengenalan Artificial Intelligence)*. Yogyakarta : Andi
- Suprianto, D., & Hasanah, R. N. (2014) . Sistem Pengenalan Wajah Secara Real-Time dengan Adaboost, Eigenface PCA & MySQL. *Jurnal EECCIS*, 7(2),179-184.
- Surówka, G., & Ogorzałek, M. (2016).On Optimal Wavelet Bases for Classification of Melanoma Images Through Ensemble Learning. *Proceeding International Conference Artificial Intelligence and Soft Computing 2016*, 9692, 655-666
- Sutarti, S., Putra, A., & Sugiharti, E. (2019). Comparison of PCA and 2DPCA Accuracy with K-Nearest Neighbor Classification in Face Image Recognition. *Scientific Journal on Informatics*, 6(1), 64-72
- T. Ayyavoo and Jayasudha J.S. (2013). Face Recognition Using Enhanced Energy of Discrete Wavelet Transform, *2013 International Conference on Control Communication and Computing (ICCC)*,1,415-419.

- Vedayoko, L., Sugiharti, E., & Muslim, M. (2017). Expert System Diagnosis of Bowel Disease Using Case Based Reasoning with Nearest Neighbor Algorithm. *Scientific Journal of Informatics*, 4(2), 134-142
- Woods, J. W. (2012). *Multidimensional Signal, Image, and Video Processing and Coding 2nd Edition*. Oxford: Academic Press
- Yanushkevich, S.N. (2005). *Biometric Inverse Problems*, Canada: Taylor and Francis Group.

LAMPIRAN

Lampiran 1. Source Code

Pembagian Data Training Testing

```
function [Train
Test]=Train_Test(DATA,No_SampleClass,No_TrainSamples,No
_TestSamples)
[m n TotalSamples]=size(DATA);
%----- Test Images -----
d1=1; d2=No_TrainSamples;
t1=1; t2=No_TrainSamples;
for i=1: No_TrainSamples*(TotalSamples/No_SampleClass);
Train(:, :, d1:d2)=DATA(:, :, t1:t2);
d1=d1+No_TrainSamples; d2=d2+No_TrainSamples;
t1=t1+No_SampleClass; t2=t2+No_SampleClass;
if (t1 > TotalSamples)
    break;
end
end
%----- Train Images -----
c1=1; c2=No_TestSamples;
e1=No_TrainSamples+1;
e2=No_TestSamples+No_TrainSamples;
for i=1:No_TestSamples*(TotalSamples/No_SampleClass);
    Test(:, :, c1:c2)=DATA(:, :, e1:e2);
    c1=c1+No_TestSamples; c2=c2+No_TestSamples;
    e1=e1+No_SampleClass; e2=e2+No_SampleClass;
    if (e1 > TotalSamples)
        break;
    end
end
end
```

Eigen decomposition

```
function [eigvec_St1,eigval_St1]=eig_decomp(St1);
[eigvec,eigval]=eig(St1);
eigval=abs(diag(eigval)');
[eigval,I]=sort(eigval);
eigval_St1=fliplr(eigval);
eigvec_St1=fliplr(eigvec(:,I));
```

Label Sample

```
function [TrainLabel TestLabel]=LabelSamples(Num_Class,
No_TrainSamples, No_TestSamples)
TrainLabel1=[]; TestLabel1=[];
for i=1: Num_Class
    Ltrain= repmat(i, No_TrainSamples);
Ltest= repmat(i, No_TestSamples);
    Ltrain1=Ltrain(1,:); Ltest1=Ltest(1,:);
```

```

    TrainLabel1=[TrainLabel1 Ltrain1];
TestLabel1=[TestLabel1 Ltest1];
end
TrainLable=TrainLabel1';
TestLable=TestLabel1';

```

Pengenalan Wajah 1D-PCA + KNN AT & T *Dataset*

```

clc
clear all;
load ORL_FaceDataSet; % Dataset Diganti Sesuai Data
Yang Ingin Diolah
A=double(ORL_FaceDataSet); % Kolom di Dataset

Num_Class=40;
No_SampleClass=10;
No_TrainSamples=9; % Ubah jumlah citra latih
No_TestSamples=1; % Ubah jumlah citra uji
DIM=10;
[TrainData,
TestData]=Train_Test(A,No_SampleClass,No_TrainSamples,No
o_TestSamples);
[m,n,TotalTrainSamples] = size(TrainData);
[m1,n1,TotalTestSamples] = size(TestData);
[TrainLabel,TestLabel]=LebelSamples(Num_Class,
No_TrainSamples, No_TestSamples);
%Training
TrainDataV = reshape(TrainData, [m*n
TotalTrainSamples]);
MeanTrainDataV=(mean(TrainDataV'))';
Diff=bsxfun(@minus,TrainDataV,MeanTrainDataV);
[EigVect1 EigVal]=eig_decomp(Diff'*Diff);
EigVect=EigVect1(:,1:DIM);
EigImages=Diff*EigVect;
EigImages=bsxfun(@rdivide,EigImages,sqrt(sum(EigImages.
^2)));
TrainFeatureM=EigImages'*Diff;
% Testing and Recognition
TestResult = zeros(TotalTestSamples,1);
for i=1:TotalTestSamples
    TestImageC=reshape(TestData(:, :, i),m*n,1)-
MeanTrainDataV;
    TestFeatureV=EigImages'*TestImageC;
    DIST1=bsxfun(@minus,TrainFeatureM,TestFeatureV);
    DIST=sqrt(sum(DIST1.*DIST1));

```

```

        subplot 221;
imshow(TestData(:,:,i),[]);title(['Tested Face = '
num2str(i)]);
        subplot 222;
imshow(TrainData(:,:,ID),[]);title(['Recognized Face =
' num2str(ID)]);
        subplot(2,2,[3 4]); plot(DIST,'-
o','MarkerIndices',[ID
ID],'MarkerFaceColor','blue','MarkerSize',5);title(['Mi
n Distance = ' num2str(MINDIST),' ID = ' num2str(ID)]);
        xlabel('Training Samples')
        ylabel('Distance')
        grid on
        grid minor
        TestResult(i) = TrainLabel(ID);
        pause(0.01)
end
Result = (TestResult == TestLabel);
CorrectRate = 100*sum(Result)/(TotalTestSamples));
msgbox(['Accuracy (in %):
',num2str(CorrectRate)],'Accuracy')

```

Pengenalan Wajah *Wavelet* +1D-PCA+ KNN AT & T Dataset

```

clc
clear all;
load ORLDWT;
A=double(h);

Num_Class=40;
No_SampleClass=10;
No_TrainSamples=9;
No_TestSamples=1;
DIM=10;
[TrainData,
TestData]=Train_Test(A,No_SampleClass,No_TrainSamples,No
o_TestSamples);
[m,n,TotalTrainSamples] = size(TrainData);
[m1,n1,TotalTestSamples] = size(TestData);
[TrainLabel,TestLabel]=LebelSamples(Num_Class,
No_TrainSamples, No_TestSamples);
%Training
TrainDataV = reshape(TrainData, [m*n
TotalTrainSamples]);
MeanTrainDataV=(mean(TrainDataV'))';
Diff=bsxfun(@minus,TrainDataV,MeanTrainDataV);
[EigVect1 EigVal]=eig_decomp(Diff'*Diff);
EigVect=EigVect1(:,1:DIM);

```

```

EigImages=Diff*EigVect;
EigImages=bsxfun(@rdivide,EigImages,sqrt(sum(EigImages.
^2)));
TrainFeatureM=EigImages'*Diff;

% Testing and Recognition
TestResult = zeros(TotalTestSamples,1);
for i=1:TotalTestSamples
    TestImageC=reshape(TestData(:,:,i),m*n,1)-
MeanTrainDataV;
    TestFeatureV=EigImages'*TestImageC;
    DIST1=bsxfun(@minus,TrainFeatureM,TestFeatureV);
    DIST=sqrt(sum(DIST1.*DIST1));
    subplot 221;
    imshow(TestData(:,:,i),[]);title(['Tested Face = '
num2str(i)]);
    subplot 222;
    imshow(TrainData(:,:,ID),[]);title(['Recognized Face =
' num2str(ID)]);
    subplot(2,2,[3 4]); plot(DIST,'-
o','MarkerIndices',[ID
ID],'MarkerFaceColor','blue','MarkerSize',5);title(['Mi
n Distance = ' num2str(MINDIST),' ID = ' num2str(ID)]);
    xlabel('Training Samples')
    ylabel('Distance')
    grid on
    grid minor
    TestResult(i) = TrainLabel(ID);
    pause(0.01)
end

Result = (TestResult == TestLabel);
CorrectRate = 100*sum(Result/(TotalTestSamples));
msgbox(['Accuracy (in %):
',num2str(CorrectRate)],'Accuracy')

```

Pengenalan Wajah 2D-PCA + KNN AT & T Dataset

```

clc
clear all;
load ORL_FaceDataSet; % Dataset Diganti Sesuai Data
Yang Ingin Diolah
A=double(ORL_FaceDataSet); % Kolom di Dataset

Num_Class=40;
No_SampleClass=10;
No_TrainSamples=9; % Ubah jumlah citra latih
No_TestSamples=1; % Ubah jumlah citra uji
DIM=6;

```

```

[TrainData,
TestData]=Train_Test(A,No_SampleClass,No_TrainSamples,No
o_TestSamples);
[m,n,TotalTrainSamples] = size(TrainData);
[m1,n1,TotalTestSamples] = size(TestData);
[TrainLabel,TestLabel]=LabelSamples(Num_Class,
No_TrainSamples, No_TestSamples);
TrainMean = mean(TrainData,3); % Menghitung Matriks
Kovarian
Gt=zeros([ n n]);
for i=1:TotalTrainSamples
    Temp = TrainData(:, :, i) - TrainMean;
    Gt = Gt + Temp'*Temp;
end
Gt=Gt/TotalTrainSamples;
[EigVect1,EigVal1]=eig_decomp(Gt);
EigVect=EigVect1(:,1:DIM);
%Training
for i=1:TotalTrainSamples
    Ytrain(:, :, i)=TrainData(:, :, i)*EigVect;
end

% Testing and Classification
TestResult = zeros(TotalTestSamples,1);
for i=1:TotalTestSamples
    Distance = zeros(TotalTrainSamples,1);
    Ytest = TestData(:, :, i)* EigVect;
    for j=1:TotalTrainSamples
        for k=1:DIM
            Distance(j) = Distance(j) +
norm(Ytest(:,k)-Ytrain(:,k,j));
        end
    end

    [MINDIST ID] = min(Distance);
    TestResult(i) = TrainLabel(ID);
    subplot 221;
    imshow(TestData(:, :, i), []);title(['Tested Face = '
num2str(i)]);
    subplot 222;
    imshow(TrainData(:, :, ID), []);title(['Recognized Face =
' num2str(ID)]);
    subplot(2,2,[3 4]); plot(Distance, '-
o', 'MarkerIndices', [ID
ID], 'MarkerFaceColor', 'blue', 'MarkerSize', 5);title(['Mi
n Distance = ' num2str(MINDIST), ' ID = ' num2str(ID)]);
    xlabel('Training Samples')
    ylabel('Distance')
    grid on

```

```

        grid minor
        pause(0.01)
    end
    Result = (TestResult == TestLabel);
    msgbox([' Accuracy (in %):
    ', num2str(CorrectRate)], 'Accuracy')

```

Pengenalan Wajah *Wavelet*+2D-PCA+KNN AT & T Dataset

```

clc
clear all;
load ORL_DWT; % Dataset Diganti Sesuai Data Yang Ingin
Diolah
A=double(ORL_FaceDataSet); % Kolom di Dataset

Num_Class=40;
No_SampleClass=10;
No_TrainSamples=9; % Ubah jumlah citra latih
No_TestSamples=1; % Ubah jumlah citra uji
DIM=6;
[TrainData,
TestData]=Train_Test(A,No_SampleClass,No_TrainSamples,No_
o_TestSamples);
[m,n,TotalTrainSamples] = size(TrainData);
[m1,n1,TotalTestSamples] = size(TestData);
[TrainLabel,TestLabel]=LebelSamples(Num_Class,
No_TrainSamples, No_TestSamples);
TrainMean = mean(TrainData,3); % Menghitung Matriks
Kovarian
Gt=zeros([ n n]);
for i=1:TotalTrainSamples
    Temp = TrainData(:, :, i) - TrainMean;
    Gt = Gt + Temp'*Temp;
end
Gt=Gt/TotalTrainSamples;
[EigVect1,EigVal1]=eig_decomp(Gt);
EigVect=EigVect1(:,1:DIM);
%Training
for i=1:TotalTrainSamples
    Ytrain(:, :, i)=TrainData(:, :, i)*EigVect;
end

% Testing and Classification
TestResult = zeros(TotalTestSamples,1);
for i=1:TotalTestSamples
    Distance = zeros(TotalTrainSamples,1);
    Ytest = TestData(:, :, i)* EigVect;

```

```

    for j=1:TotalTrainSamples
        for k=1:DIM
            Distance(j) = Distance(j) +
norm(Ytest(:,k)-Ytrain(:,k,j));
        end
    end

    [MINDIST ID] = min(Distance);
    TestResult(i) = TrainLabel(ID);
    subplot 221;
imshow(TestData(:,:,i),[]);title(['Tested Face = '
num2str(i)]);
    subplot 222;
imshow(TrainData(:,:,ID),[]);title(['Recognized Face =
' num2str(ID)]);
    subplot(2,2,[3 4]); plot(Distance,'-
o','MarkerIndices',[ID
ID],'MarkerFaceColor','blue','MarkerSize',5);title(['Mi
n Distance = ' num2str(MINDIST),' ID = ' num2str(ID)]);
    xlabel('Training Samples')
    ylabel('Distance')
    grid on
    grid minor
    pause(0.01)
end
Result = (TestResult == TestLabel);
msgbox(['·Accuracy (in %):
',num2str(CorrectRate)],'Accuracy')

```

Pengenalan Wajah 1D-PCA + KNN YALE

```

clc
clear all;
load YALE; % Dataset Diganti Sesuai Data Yang Ingin
Diolah
A=double(ORL_FaceDataSet); % Kolom di Dataset

Num_Class=40;
No_SampleClass=10;
No_TrainSamples=9; % Ubah jumlah citra latih
No_TestSamples=1; % Ubah jumlah citra uji
DIM=10;
[TrainData,
TestData]=Train_Test(A,No_SampleClass,No_TrainSamples,N
o_TestSamples);
[m,n,TotalTrainSamples] = size(TrainData);
[m1,n1,TotalTestSamples] = size(TestData);

```

```

[TrainLabel,TestLabel]=LabelSamples(Num_Class,
No_TrainSamples, No_TestSamples);
%Training
TrainDataV = reshape(TrainData, [m*n
TotalTrainSamples]);
MeanTrainDataV=(mean(TrainDataV'))';
Diff=bsxfun(@minus,TrainDataV,MeanTrainDataV);
[EigVect1 EigVal]=eig_decomp(Diff'*Diff);
EigVect=EigVect1(:,1:DIM);
EigImages=Diff*EigVect;
EigImages=bsxfun(@rdivide,EigImages,sqrt(sum(EigImages.
^2)));
TrainFeatureM=EigImages'*Diff;
% Testing and Recognition
TestResult = zeros(TotalTestSamples,1);
for i=1:TotalTestSamples
    TestImageC=reshape(TestData(:,:,i),m*n,1)-
MeanTrainDataV;
    TestFeatureV=EigImages'*TestImageC;
    DIST1=bsxfun(@minus,TrainFeatureM,TestFeatureV);
    DIST=sqrt(sum(DIST1.*DIST1));
    subplot 221;
    imshow(TestData(:,:,i),[]);title(['Tested Face = '
num2str(i)]);
    subplot 222;
    imshow(TrainData(:,:,ID),[]);title(['Recognized Face =
' num2str(ID)]);
    subplot(2,2,[3 4]); plot(DIST,'-
o','MarkerIndices',[ID
ID],'MarkerFaceColor','blue','MarkerSize',5);title(['Mi
n Distance = ' num2str(MINDIST),' ID = ' num2str(ID)]);
    xlabel('Training Samples')
    ylabel('Distance')
    grid on
    grid minor
    TestResult(i) = TrainLabel(ID);
    pause(0.01)
end
Result = (TestResult == TestLabel);
CorrectRate = 100*sum(Result/(TotalTestSamples));
msgbox(['Accuracy (in %):
',num2str(CorrectRate)],'Accuracy')

```

Pengenalan Wajah *Wavelet* +1D-PCA+ KNN

```

clc
clear all;
load YALDWT;

```



```

A=double(h);

Num_Class=40;
No_SampleClass=10;
No_TrainSamples=9;
No_TestSamples=1;
DIM=10;
[TrainData,
TestData]=Train_Test(A,No_SampleClass,No_TrainSamples,No_
o_TestSamples);
[m,n,TotalTrainSamples] = size(TrainData);
[m1,n1,TotalTestSamples] = size(TestData);
[TrainLabel,TestLabel]=LabelSamples(Num_Class,
No_TrainSamples, No_TestSamples);
%Training
TrainDataV = reshape(TrainData, [m*n
TotalTrainSamples]);
MeanTrainDataV=(mean(TrainDataV'))';
Diff=bsxfun(@minus,TrainDataV,MeanTrainDataV);
[EigVect1 EigVal]=eig_decomp(Diff'*Diff);
EigVect=EigVect1(:,1:DIM);
EigImages=Diff*EigVect;
EigImages=bsxfun(@rdivide,EigImages,sqrt(sum(EigImages.
^2)));
TrainFeatureM=EigImages'*Diff;

% Testing and Recognition
TestResult = zeros(TotalTestSamples,1);
for i=1:TotalTestSamples
    TestImageC=reshape(TestData(:, :, i),m*n,1)-
MeanTrainDataV;
    TestFeatureV=EigImages'*TestImageC;
    DIST1=bsxfun(@minus,TrainFeatureM,TestFeatureV);
    DIST=sqrt(sum(DIST1.*DIST1));
    subplot 221;
    imshow(TestData(:, :, i), []);title(['Tested Face = '
num2str(i)]);
    subplot 222;
    imshow(TrainData(:, :, ID), []);title(['Recognized Face =
' num2str(ID)]);
    subplot(2,2,[3 4]); plot(DIST,'-
o','MarkerIndices',[ID
ID],'MarkerFaceColor','blue','MarkerSize',5);title(['Mi
n Distance = ' num2str(MINDIST),' ID = ' num2str(ID)]);
    xlabel('Training Samples')
    ylabel('Distance')
    grid on
    grid minor
    TestResult(i) = TrainLabel(ID);

```

```

        pause(0.01)
    end

    Result = (TestResult == TestLabel);
    CorrectRate = 100*sum(Result/(TotalTestSamples));
    msgbox([' Accuracy (in %):', num2str(CorrectRate)], 'Accuracy')

```

Pengenalan Wajah 2D-PCA + KNN

```

clc
clear all;
load YALE; % Dataset Diganti Sesuai Data Yang Ingin Diolah
A=double(ORL_FaceDataSet); % Kolom di Dataset

Num_Class=40;
No_SampleClass=10;
No_TrainSamples=9; % Ubah jumlah citra latih
No_TestSamples=1; % Ubah jumlah citra uji
DIM=6;
[TrainData,
TestData]=Train_Test(A,No_SampleClass,No_TrainSamples,No_TestSamples);
[m,n,TotalTrainSamples] = size(TrainData);
[m1,n1,TotalTestSamples] = size(TestData);
[TrainLabel,TestLabel]=LabelSamples(Num_Class,
No_TrainSamples, No_TestSamples);
TrainMean = mean(TrainData,3); % Menghitung Matriks Kovarian
Gt=zeros([ n n]);
for i=1:TotalTrainSamples
    Temp = TrainData(:, :, i) - TrainMean;
    Gt = Gt + Temp'*Temp;
end
Gt=Gt/TotalTrainSamples;
[EigVect1,EigVal1]=eig_decomp(Gt);
EigVect=EigVect1(:, 1:DIM);
%Training
for i=1:TotalTrainSamples
    Ytrain(:, :, i)=TrainData(:, :, i)*EigVect;
end

% Testing and Classification
TestResult = zeros(TotalTestSamples,1);
for i=1:TotalTestSamples
    Distance = zeros(TotalTrainSamples,1);
    Ytest = TestData(:, :, i)* EigVect;
    for j=1:TotalTrainSamples

```

```

        for k=1:DIM
            Distance(j) = Distance(j) +
norm(Ytest(:,k)-Ytrain(:,k,j));
        end
    end

    [MINDIST ID] = min(Distance);
    TestResult(i) = TrainLabel(ID);
    subplot 221;
    imshow(TestData(:,:,i),[]);title(['Tested Face = '
num2str(i)]);
    subplot 222;
    imshow(TrainData(:,:,ID),[]);title(['Recognized Face =
' num2str(ID)]);
    subplot(2,2,[3 4]); plot(Distance,'-
o','MarkerIndices',[ID
ID],'MarkerFaceColor','blue','MarkerSize',5);title(['Mi
n Distance = ' num2str(MINDIST),' ID = ' num2str(ID)]);
    xlabel('Training Samples')
    ylabel('Distance')
    grid on
    grid minor
    pause(0.01)
end
Result = (TestResult == TestLabel);
msgbox(['Accuracy (in %):
',num2str(CorrectRate)],'Accuracy')

```

Pengenalan Wajah *Wavelet*+2D-PCA+KNN

```

clc
clear all;
load YALDWT; % Dataset Diganti Sesuai Data Yang Ingin
Diolah
A=double(ORL_FaceDataSet); % Kolom di Dataset

Num_Class=40;
No_SampleClass=10;
No_TrainSamples=9; % Ubah jumlah citra latih
No_TestSamples=1; % Ubah jumlah citra uji
DIM=6;
[TrainData,
TestData]=Train_Test(A,No_SampleClass,No_TrainSamples,No
o_TestSamples);
[m,n,TotalTrainSamples] = size(TrainData);
[m1,n1,TotalTestSamples] = size(TestData);

```

```

[TrainLabel,TestLabel]=LabelSamples (Num_Class,
No_TrainSamples, No_TestSamples);
TrainMean = mean (TrainData,3); % Menghitung Matriks
Kovarian
Gt=zeros ([ n n]);
for i=1:TotalTrainSamples
    Temp = TrainData (:,:,i) - TrainMean;
    Gt = Gt + Temp'*Temp;
end
Gt=Gt/TotalTrainSamples;
[EigVect1,EigVal1]=eig_decomp (Gt);
EigVect=EigVect1 (:,1:DIM);
%Training
for i=1:TotalTrainSamples
    Ytrain (:,:,i)=TrainData (:,:,i)*EigVect;
end

% Testing and Classification
TestResult = zeros (TotalTestSamples,1);
for i=1:TotalTestSamples
    Distance = zeros (TotalTrainSamples,1);
    Ytest = TestData (:,:,i)* EigVect;
    for j=1:TotalTrainSamples
        for k=1:DIM
            Distance (j) = Distance (j) +
norm (Ytest (:,k)-Ytrain (:,k,j));
        end
    end

    [MINDIST ID] = min (Distance);
    TestResult (i) = TrainLabel (ID);
    subplot 221;
    imshow (TestData (:,:,i), []); title (['Tested Face = '
num2str (i)]);
    subplot 222;
    imshow (TrainData (:,:,ID), []); title (['Recognized Face =
' num2str (ID)]);
    subplot (2,2,[3 4]); plot (Distance, '-
o', 'MarkerIndices', [ID
ID], 'MarkerFaceColor', 'blue', 'MarkerSize', 5); title (['Mi
n Distance = ' num2str (MINDIST), ' ID = ' num2str (ID)]);
    xlabel ('Training Samples')
    ylabel ('Distance')
    grid on
    grid minor
    pause (0.01)
end
Result = (TestResult == TestLabel);

```

```
msgbox(['Accuracy (in %):  
,num2str(CorrectRate)], 'Accuracy')
```

Lampiran 2. Hasil Percobaan Menggunakan AT & T

Hasil Percobaan 10% *Training Images* dan 90% *Testing Images* dengan

PCA+KNN

Citra Uji	Kelas Citra Uji	Kelas Citra Latih	<i>Euclidian Distance</i>	Citra Uji	Kelas Citra Uji	Kelas Citra Latih	<i>Euclidian Distance</i>
1_1	1	18	3228.8030	6_4	6	6	1561.2967
1_2	1	1	1972.0305	6_5	6	6	1506.0773
1_3	1	18	2562.1538			...	
1_4	1	18	2485.5085	34_6	34	34	1454.9469
1_5	1	1	2481.6122	34_7	34	34	1282.6586
1_6	1	1	1634.0377	34_8	34	34	1712.7559
1_7	1	1	2959.9710	34_9	34	34	1770.2505
1_8	1	18	2487.5949	35_1	35	25	1721.1872
1_9	1	18	2413.1154	35_2	35	5	2430.9964
2_1	2	2	1324.2476	35_3	35	4	1916.4466
2_2	2	2	77.2495	35_4	35	26	2284.7257
2_3	2	2	1293.3894	35_5	35	25	1651.3434
2_4	2	2	1032.2455	35_6	35	25	2118.1201
2_5	2	15	1575.6812	35_7	35	25	1292.2507
2_6	2	2	1144.3070	35_8	35	26	2614.3779
2_7	2	2	856.5024	35_9	35	40	2084.8523
2_8	2	15	1700.5151	36_1	36	30	1970.1763
2_9	2	2	1208.2981	36_2	36	7	1777.6347
3_1	3	3	8374.4273	36_3	36	7	1762.9537
3_2	3	3	854.3307	36_4	36	36	1101.4384
3_3	3	4	1699.2209	36_5	36	24	1835.8898
3_4	3	26	2077.2452	36_6	36	24	1827.6029
3_5	3	3	1533.0944	36_7	36	7	1893.4941
3_6	3	4	1468.3013	36_8	36	7	2310.5158
3_7	3	3	1412.8906	36_9	36	17	1820.8557
3_8	3	3	1232.8261	37_1	37	14	2744.7301
3_9	3	3	1270.0973	37_2	37	37	460.3108
4_1	4	3	1239.7803	37_3	37	19	2705.4833
4_2	4	4	719.8893	37_4	37	14	2727.3650
4_3	4	3	1436.2097	37_5	37	14	2551.2297
4_4	4	3	1875.1615	37_6	37	37	1021.3626
4_5	4	4	1963.425	37_7	37	26	2725.2424
4_6	4	3	1320.4002	37_8	37	37	517.2657
4_7	4	4	828.5579	37_9	37	37	776.7848
4_8	4	3	1267.1417	38_1	38	38	1014.1298
4_9	4	3	1873.478	38_2	38	38	1224.0284
5_1	5	5	674.8093	38_3	38	23	1595.7289
5_2	5	5	1088.0768	38_4	38	38	931.3524
5_3	5	5	313.445	38_5	38	38	583.2514
5_4	5	5	2013.949	38_6	38	38	1251.0975
5_5	5	5	1776.0071	38_7	38	38	1425.2354
5_6	5	5	467.127	38_8	38	38	1013.6374
5_7	5	25	2114.4217	38_9	38	38	1094.8402
5_8	5	40	1986.0429	39_1	39	39	1051.1286
5_9	5	12	2155.0107	39_2	39	39	1411.0381
6_1	6	6	552.3814	39_3	39	39	1227.7968
6_2	6	6	637.8352	39_4	39	39	1567.8771
6_3	6	6	1226.3429	39_5	39	39	1618.8855

Citra Uji	Kelas Citra Uji	Kelas Citra Latih	<i>Euclidian Distance</i>
39_6	39	29	1458.7652
39_7	39	29	1608.5706
39_8	39	29	1493.0437
39_9	39	29	1633.5874
40_1	40	40	1808.9291
40_2	40	40	1473.8006
40_3	40	5	1143.7199
40_4	40	25	1860.7598
40_5	40	40	1406.0345
40_6	40	25	1848.8318
40_7	40	40	785.8993
40_8	40	12	2216.3456
40_9	40	5	1393.8227

Hasil Percobaan 10% *Training Images* dan 90% *Testing Images* dengan Wavelet -
1D-PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	32	5444,3036	4_7	4	4	861,7594
1_2	1	1	3412,4733	4_8	4	4	1108,1053
1_3	1	36	5016,6489	4_9	4	25	2559,7779
1_4	1	12	3964,0032	5_1	5	5	915,1177
1_5	1	36	5376,2886	5_2	5	5	1038,3979
1_6	1	1	2386,3298	5_3	5	5	1019,2902
1_7	1	2	3946,2082	5_4	5	3	2507,1696
1_8	1	36	4053,8955	5_5	5	3	2285,359
1_9	1	1	3542,4145	5_6	5	5	651,7821
2_1	2	2	2207,9208	5_7	5	3	2387,8902
2_2	2	2	1344,4938	5_8	5	3	2335,6632
2_3	2	2	2162,1117	5_9	5	3	2512,3132
2_4	2	2	1871,9989	6_1	6	6	894,8467
2_5	2	19	2624,0273	6_2	6	6	515,4758
2_6	2	2	1859,2247	6_3	6	21	2776,0242
2_7	2	2	914,9131	6_4	6	21	2571,4121
2_8	2	19	2868,9366			...	
2_9	2	2	2319,2294	34_6	34	34	1918,2045
3_1	3	3	895,7565	34_7	34	34	1917,3952
3_2	3	3	1030,42002	34_8	34	34	2367,914
3_3	3	3	1706,118	34_9	34	34	2446,4682
3_4	3	3	2079,6819	35_1	35	25	2118,4778
3_5	3	3	1668,8355	35_2	35	21	2949,7496
3_6	3	3	2000,9132	35_3	35	25	2577,5097
3_7	3	3	1890,5236	35_4	35	18	3918,7576
3_8	3	3	1783,693	35_5	35	25	2089,1948
3_9	3	3	1707,4087	35_6	35	25	2518,8357
4_1	4	4	1404,9517	35_7	35	25	2500,3965
4_2	4	4	671,7791	35_8	35	37	4655,0190
4_3	4	3	2286,0746	35_9	35	21	4208,0883
4_4	4	3	2671,7936	36_1	36	7	2496,0953
4_5	4	25	2985,2297	36_2	36	7	1246,7996
4_6	4	3	2209,8969	36_3	36	7	1408,2779

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
35_2	35	21	2949,7496
35_3	35	25	2577,5097
35_4	35	18	3918,7576
35_5	35	25	2089,1948
35_6	35	25	2518,8357
35_7	35	25	2500,3965
35_8	35	37	4655,0190
35_9	35	21	4208,0883
36_1	36	7	2496,0953
36_2	36	7	1246,7996
36_3	36	7	1408,2779
36_4	36	36	834,6816
36_5	36	36	2901,1186
36_8	36	7	1990,7965
36_9	36	36	2536,3711
37_1	37	19	3285,4769
37_2	37	37	497,4194
37_3	37	19	3273,2596
37_4	37	16	3342,2460
37_5	37	16	3390,1662
37_6	37	37	1697,7256
37_7	37	19	3051,1621
37_8	37	37	731,0240
37_9	37	37	1330,7803
38_1	38	38	1727,8229
38_2	38	38	2189,787
38_3	38	38	1595,8678
38_4	38	38	1580,23032
38_5	38	38	857,3569
38_6	38	4	1746,7522
38_7	38	9	1673,6677
38_8	38	38	1210,3674
38_9	38	38	1509,3635
39_1	39	39	1516,323
39_2	39	39	1355,494
39_3	39	39	1869,7213
39_4	39	39	1997,9428
39_5	39	33	2605,465
39_6	39	33	1788,9642
39_7	39	33	1618,1061
39_8	39	33	2038,8503
39_9	39	22	2509,7578
40_1	40	3	2533,6709
40_2	40	40	2975,5182
40_3	40	5	1857,274
40_4	40	3	2023,8161
40_5	40	40	1392,7249
40_6	40	3	2311,4348
40_7	40	40	920,2139
40_8	40	12	3097,7119
40_9	40	5	1918,4748

Hasil Percobaan 10% *Training Images* dan 90% *Testing Images* dengan 2D-
PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	32	8947.6654	34_6	34	34	5823.9393
1_2	1	1	6929.8683	34_7	34	34	4013.6195
1_3	1	18	9596.0867	34_8	34	34	4834.9815
1_4	1	18	7921.5699	34_9	34	34	5043.4733
1_5	1	1	8654.5128	35_1	35	25	7245.5595
1_6	1	1	6442.5243	35_2	35	5	8423.0312
1_7	1	1	8097.3735	35_3	35	25	7348.8041
1_8	1	18	9928.7578	35_4	35	5	9068.3274
1_9	1	17	6092.2518	35_5	35	25	6725.6983
2_1	2	2	4603.1397	35_6	35	25	7622.9137
2_2	2	2	3792.002	35_7	35	25	6390.4636
2_3	2	2	4504.6945	35_8	35	37	1013.018
2_4	2	2	4323.1879	35_9	35	25	8535.3536
2_5	2	2	6221.0874	36_1	36	7	7741.5887
2_6	2	2	4769.2975	36_2	36	36	6429.4888
2_7	2	2	4058.4897	36_3	36	36	6512.3407
2_8	2	2	6085.1714	36_4	36	36	3365.2291
2_9	2	2	4872.5846	36_5	36	24	7143.121
3_1	3	3	4468.0587	36_6	36	24	7080.6389
3_2	3	3	4413.1703	36_7	36	7	6360.3667
3_3	3	3	5803.1212	36_8	36	36	7264.8048
3_4	3	5	7105.2454	36_9	36	17	7550.5683
3_5	3	3	4821.0250	37_1	37	14	7490.5975
3_6	3	3	5233.8287	37_2	37	37	2299.1748
3_7	3	3	5181.7887	37_3	37	14	7797.1689
3_8	3	3	4954.4552	37_4	37	14	7457.1326
3_9	3	3	5363.9191	37_5	37	14	7331.9019
4_1	4	4	5918.2865	37_6	37	37	4817.9645
4_2	4	4	3677.7232	37_7	37	22	7810.3388
4_3	4	4	5872.9378	37_8	37	37	2507.3286
4_4	4	18	7443.9737	37_9	37	37	3687.7538
4_5	4	4	7295.6499	38_1	38	38	4987.3954
4_6	4	4	5400.5802	38_2	38	38	4655.6011
4_7	4	4	3442.1376	38_3	38	38	6367.0513
4_8	4	4	5011.0669				
4_9	4	18	7562.7654				
5_1	5	5	3910.5971				
5_2	5	5	4430.0940				
5_3	5	5	1795.5725				
5_4	5	5	5226.5545				
5_5	5	5	4736.5763				
5_6	5	5	2422.1782				
5_7	5	5	7080.5469				
5_8	5	5	6568.7333				
5_9	5	5	6812.9210				
6_1	6	6	2621.4770				
6_2	6	6	2759.3955				
6_3	6	6	6135.6412				
6_4	6	6	5500.9750				
		...					

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
38_4	38	38	4844.3215
38_5	38	38	3365.8898
38_6	38	38	4986.1798
38_7	38	38	5485.0359
38_8	38	38	4374.1842
38_9	38	38	5442.7596
39_1	39	39	4372.3178
39_2	39	39	6507.9656
39_3	39	39	5629.7733
39_4	39	39	6296.0979
39_5	39	29	7411.4839
39_6	39	21	6276.2470
39_7	39	39	6236.6888
39_8	39	39	6773.5123
39_9	39	29	6495.9723
40_1	40	40	6417.4553
40_2	40	40	5118.7648
40_3	40	5	4941.0358
40_4	40	25	6801.4774
40_5	40	40	5496.0641
40_6	40	5	7108.9492
40_7	40	40	3897.5138
40_8	40	5	7362.4054
40_9	40	5	4908.1749

Hasil Percobaan 10% *Training Images* dan 90% *Testing Images* dengan *Wavelet-2D-PCA+KNN*

Citra Uji	Kelas Uji	Kelas Latih	Euclidean Distance	Citra Uji	Kelas Uji	Kelas Latih	Euclidean Distance
1_1	1	32	13997,0593	6_3	6	6	8361,6751
1_2	1	1	10130,9374	6_4	6	6	7500,2294
1_3	1	18	14933,4036			...	
1_4	1	18	12209,7105	36_4	36	36	3618,1441
1_5	1	1	14536,4262	36_5	36	24	8696,6291
1_6	1	1	8407,291	36_6	36	24	8615,1539
1_7	1	1	19786,1352	36_7	36	7	6124,5587
1_8	1	18	13900,8484	36_8	36	36	7962,0531
1_9	1	17	10569,8985	36_9	36	17	9631,8864
2_1	2	2	6158,2597	37_1	37	14	9867,4426
2_2	2	2	4691,6759	37_2	37	37	2503,7903
2_3	2	2	6336,519	37_3	37	14	9429,913
2_4	2	2	6068,2294	37_4	37	14	9922,0054
2_5	2	2	8717,1879	37_5	37	14	10123,0805
2_6	2	2	5776,4366	37_6	37	37	6047,3721
2_7	2	2	4006,5257	37_7	37	22	9503,9745
2_8	2	2	8990,726	37_8	37	37	2778,1146
2_9	2	2	6525,5392	37_9	37	37	4611,3432
3_1	3	3	4394,5061	38_1	38	38	6414,7138
3_2	3	3	4048,5813	38_2	38	38	6528,6551
3_3	3	3	6014,3353	38_3	38	38	5812,4238
3_4	3	5	7204,7653	38_4	38	38	6172,6789
3_5	3	3	5186,3656	38_5	38	38	4059,1135
3_6	3	3	5660,7681	38_6	38	38	5932,0477
3_7	3	3	5806,7769	38_7	38	38	5987,5573
3_8	3	3	5754,0164	38_8	38	38	4981,5485
3_9	3	3	5637,0367	38_9	38	38	6075,6322
4_1	4	4	4961,668	39_1	39	39	4822,7394
4_2	4	4	3631,2785	39_2	39	39	6674,8702
4_3	4	4	6631,6116	39_3	39	39	6360,5697
4_4	4	18	8544,9277	39_4	39	39	6802,2909
4_5	4	4	9278,8748	39_5	39	29	9132,9124
4_6	4	4	7006,5558	39_6	39	21	8046,5915
4_7	4	4	3822,7157	39_7	39	39	7678,8775
4_8	4	4	4423,8272	39_8	39	39	8102,7458
4_9	4	18	8695,3921	39_9	39	29	8550,8212
5_1	5	5	3791,1989	40_1	40	40	8198,6654
5_2	5	5	4390,028	40_2	40	40	7333,1405
5_3	5	5	2408,8331	40_3	40	5	5475,9507
5_4	5	5	7217,8362	40_4	40	25	7317,8836
5_5	5	5	6117,7739	40_5	40	40	5365,1217
5_6	5	5	2569,6732	40_6	40	5	8497,2818
5_7	5	5	7944,7088	40_7	40	40	4639,9449
5_8	5	5	7771,4683	40_8	40	5	10230,1708
5_9	5	5	8153,697	40_9	40	5	5369,6487
6_1	6	6	2675,3613				
6_2	6	6	1853,0143				

Hasil Percobaan 50% *Training Images* dan 50% *Testing Images* dengan
PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	2095,9148	9_2	9	38	1511,3838
1_2	1	1	1643,1006	9_3	9	9	1098,5784
1_3	1	16	2061,0761	9_4	9	9	957,6064
1_4	1	1	1742,996	9_5	9	9	989,6439
1_5	1	1	1949,8265	10_1	10	10	1655,4648
2_1	2	2	869,6467	10_2	10	10	1491,6706
2_2	2	2	868,8359	10_3	10	10	1098,8776
2_3	2	2	283,1424	10_4	10	38	1778,6334
2_4	2	2	1183,3647	10_5	10	38	2057,5887
2_5	2	2	751,4609			...	
3_1	3	17	1577,6297	35_1	35	25	1304,9639
3_2	3	4	1314,4409	35_2	35	25	1671,4977
3_3	3	4	1506,4	35_3	35	25	1114,6779
3_4	3	3	1208,3398	35_4	35	35	1013,8693
3_5	3	3	1083,6882	35_5	35	35	1280,8562
4_1	4	4	1759,8547	36_1	36	24	1893,831
4_2	4	4	523,6569	36_2	36	36	1414,4792
4_3	4	4	493,4133	36_3	36	36	713,4244
4_4	4	4	602,7668	36_4	36	36	1319,4314
4_5	4	4	690,321	36_5	36	17	1776,6256
5_1	5	5	356,1056	37_1	37	37	542,0452
5_2	5	5	338,7903	37_2	37	37	981,7879
5_3	5	5	1068,0094	37_3	37	37	652,1959
5_4	5	5	869,8165	37_4	37	37	481,9192
5_5	5	5	1046,6815	37_5	37	37	839,0862
6_1	6	6	530,4418	38_1	38	38	731,7127
6_2	6	6	555,0469	38_2	38	38	609,6512
6_3	6	6	1033,0925	38_3	38	38	1053,1811
6_4	6	6	780,999	38_4	38	38	1006,4222
6_5	6	6	369,3522	38_5	38	38	531,0791
7_1	7	7	985,7427	39_1	39	39	1167,4911
7_2	7	7	592,4017	39_2	39	39	1265,8528
7_3	7	7	1277,7093	39_3	39	33	1447,2147
7_4	7	7	702,1682	39_4	39	39	683,0487
7_5	7	7	1020,5471	40_1	40	40	1346,5922
8_1	8	8	840,8863	40_2	40	40	568,5868
8_2	8	8	870,4582	40_3	40	40	659,9945
8_3	8	8	752,2878	40_4	40	40	1135,9147
8_4	8	8	1313,8716	40_5	40	40	970,5847
8_5	8	8	649,0795				
9_1	9	38	1461,5593				

Hasil Percobaan 50% *Training Images* dan 50% *Testing Images* dengan *Wavelet-1D-PCA+KNN*

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	3325,5867	9_2	9	25	1810,4527
1_2	1	1	2464,4184	9_3	9	9	1483,567
1_3	1	1	2891,8895	9_4	9	9	104,5546
1_4	1	1	2865,2179	9_5	9	9	675,4623
1_5	1	16	2913,7032	10_1	10	10	2414,0561
2_1	2	2	1527,7462	10_2	10	10	1701,8424
2_2	2	2	1875,7669	10_3	10	10	1174,9077
2_3	2	2	999,1235	10_4	10	10	1835,8305
2_4	2	2	1660,8826	10_5	10	10	2491,5562
2_5	2	2	2021,5845			...	
3_1	3	3	1777,9377	35_1	35	35	2031,1841
3_2	3	3	1948,989	35_2	35	35	2404,9419
3_3	3	3	1866,2991	35_3	35	35	2324,2921
3_4	3	3	1659,2245	35_4	35	35	1480,5456
3_5	3	3	1441,7233	35_5	35	35	1096,5268
4_1	4	4	2166,5136	36_1	36	36	2084,5501
4_2	4	4	686,4927	36_2	36	36	1410,2264
4_3	4	4	474,9977	36_3	36	7	1127,408
4_4	4	4	397,2221	36_4	36	36	1505,1358
4_5	4	4	831,8373	36_5	36	36	1569,3769
5_1	5	5	1146,3128	37_1	37	37	520,9185
5_2	5	5	619,7704	37_2	37	37	1736,6849
5_3	5	5	1135,4151	37_3	37	37	827,8709
5_4	5	5	1387,2009	37_4	37	37	790,8571
5_5	5	40	1945,8881	37_5	37	37	1409,7951
6_1	6	6	461,8344	38_1	38	38	865,0748
6_2	6	6	1098,4313	38_2	38	38	574,6002
6_3	6	6	842,2525	38_3	38	38	1540,6543
6_4	6	6	1704,5656	38_4	38	38	1286,9173
6_5	6	6	633,8105	38_5	38	38	903,9057
7_1	7	7	780,6014	39_1	39	39	1751,8851
7_2	7	7	519,1219	39_2	39	33	1496,4040
7_3	7	7	1080,4123	39_3	39	33	1421,1596
7_4	7	7	699,5363	39_4	39	39	1327,2822
7_5	7	7	852,6167	39_5	39	33	1860,3006
8_1	8	8	801,3026	40_1	40	40	1195,9018
8_2	8	8	792,6550	40_2	40	40	932,2247
8_3	8	8	1015,9387	40_3	40	40	779,6299
8_4	8	8	1707,2262	40_4	40	40	1520,7492
8_5	8	8	710,7356	40_5	40	40	831,0000
9_1	9	9	1553,0614				

Hasil Percobaan 50% *Training Images* dan 50% *Testing Images* dengan 2D-
PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	6617,8149	9_3	9	9	4462,613
1_2	1	1	6513,8709	9_4	9	9	3920,3878
1_3	1	1	7214,0765	9_5	9	9	4710,0185
1_4	1	1	7271,3365	10_1	10	10	6221,2158
1_5	1	17	6652,8522	10_2	10	10	5387,3375
2_1	2	2	3909,7338	10_3	10	10	4177,4428
2_2	2	2	3654,0128	10_4	10	10	7705,6337
2_3	2	2	2157,3138	10_5	10	38	7684,4082
2_4	2	2	4253,1368			...	
2_5	2	2	2983,6531	35_1	35	35	6169,5963
3_1	3	3	4745,5566	35_2	35	35	7469,1196
3_2	3	3	5297,2305	35_3	35	35	5763,4468
3_3	3	3	5280,2741	35_4	35	35	5900,8529
3_4	3	3	4390,9253	35_5	35	35	5929,0234
3_5	3	3	4072,2241	36_1	36	36	7063,0663
4_1	4	4	5857,2259	36_2	36	36	5311,4483
4_2	4	4	2204,3789	36_3	36	36	2499,0234
4_3	4	4	1916,9182	36_4	36	36	4930,6444
4_4	4	4	3578,5548	36_5	36	17	7483,8806
4_5	4	4	3874,9705	37_1	37	37	3256,6064
5_1	5	5	1953,7812	37_2	37	37	4742,6468
5_2	5	5	1838,4966	37_3	37	37	4289,8878
5_3	5	5	5735,2483	37_4	37	37	2490,8254
5_4	5	5	4853,7911	37_5	37	37	3644,4403
5_5	5	40	4523,0410	38_1	38	38	3382,3570
6_1	6	6	2642,6010	38_2	38	38	2947,8462
6_2	6	6	3053,2809	38_3	38	38	5041,5950
6_3	6	6	4299,0142	38_4	38	38	4329,1877
6_4	6	6	5149,2945	38_5	38	38	2625,2302
6_5	6	6	2329,4036	39_1	39	39	6366,4693
7_1	7	7	3712,8551	39_2	39	39	5124,9973
7_2	7	7	2570,2633	39_3	39	39	5639,1651
7_3	7	7	4397,2109	39_4	39	39	3348,6385
7_4	7	7	3264,8425	39_5	39	39	5933,5748
7_5	7	7	4247,1208	40_1	40	40	5602,6697
8_1	8	8	3646,2969	40_2	40	40	4641,7857
8_2	8	8	3855,1547	40_3	40	40	4024,2642
8_3	8	8	2921,7487	40_4	40	40	6254,3350
8_4	8	8	5494,0248	40_5	40	40	4651,7599
8_5	8	8	3617,2635				
9_1	9	9	5324,9423				
9_2	9	9	6325,7708				

Hasil Percobaan 50% *Training Images* dan 50% *Testing Images* dengan *Wavelet-2D-PCA+KNN*

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	10051,3684	9_2	9	9	6518,4634
1_2	1	1	8574,1706	9_3	9	9	4546,622
1_3	1	1	9270,5253	9_4	9	9	4867,185
1_4	1	1	9369,1698	9_5	9	9	4861,7893
1_5	1	1	9178,3106	10_1	10	10	7726,7645
2_1	2	2	5373,0592	10_2	10	10	5760,022
2_2	2	2	5836,7524	10_3	10	10	4031,2633
2_3	2	2	3683,1002	10_4	10	10	8508,3532
2_4	2	2	7249,0303	10_5	10	8	8898,7936
2_5	2	2	5509,1923			...	
3_1	3	3	5054,6800	35_1	35	35	7256,5651
3_2	3	3	5593,6529	35_2	35	35	8882,6481
3_3	3	3	5749,0857	35_3	35	35	8009,9241
3_4	3	3	5482,1589	35_4	35	35	7626,3566
3_5	3	3	4756,5966	35_5	35	35	6803,266
4_1	4	4	7037,1857	36_1	36	36	7766,2661
4_2	4	4	3371,301	36_2	36	36	5148,6055
4_3	4	4	1938,1971	36_3	36	36	3631,9632
4_4	4	4	2910,4354	36_4	36	36	5356,7728
4_5	4	4	4260,3405	36_5	36	36	7860,9069
5_1	5	5	3353,6015	37_1	37	37	2250,3791
5_2	5	5	2476,4505	37_2	37	37	5881,5651
5_3	5	5	5447,8102	37_3	37	37	3323,0299
5_4	5	5	5261,7751	37_4	37	37	2772,0576
5_5	5	40	5347,9912	37_5	37	37	4706,9664
6_1	6	6	2124,4302	38_1	38	38	4068,8056
6_2	6	6	3483,3983	38_2	38	38	2880,7815
6_3	6	6	3636,1829	38_3	38	38	5897,0317
6_4	6	6	6210,9833	38_4	38	38	4932,9287
6_5	6	6	2368,1313	38_5	38	38	2866,8393
7_1	7	7	3423,8123	39_1	39	39	7697,0754
7_2	7	7	2341,1619	39_2	39	39	5733,5559
7_3	7	7	3966,0086	39_3	39	39	5563,0041
7_4	7	7	2806,0935	39_4	39	39	5571,48
7_5	7	7	3662,1352	39_5	39	30	8093,6759
8_1	8	8	2921,9947	40_1	40	40	5390,1799
8_2	8	8	3648,0368	40_2	40	40	4752,8988
8_3	8	8	3017,309	40_3	40	40	4684,9471
8_4	8	8	5511,7331	40_4	40	40	6777,262
8_5	8	8	3076,2772	40_5	40	40	4368,6147
9_1	9	9	5030,9189				

Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan 1D-PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	Euclidean Distance
1_1	1	1	1748,6593
2_1	2	2	574,6993
3_1	3	3	521,6226
4_1	4	4	748,4565
5_1	5	18	1275,5793
6_1	6	6	373,9567
7_1	7	7	943,0408
8_1	8	8	577,6611
9_1	9	9	839,3134
10_1	10	38	2024,5329
11_1	11	11	831,9210
12_1	12	12	574,7508
13_1	13	13	175,4414
14_1	14	14	1402,0586
15_1	15	15	391,1416
16_1	16	16	1387,4930
17_1	17	17	591,6547
18_1	18	18	117,3512
19_1	19	19	1029,8658
20_1	20	20	486,2215
21_1	21	21	650,1464
22_1	22	22	598,4350
23_1	23	23	795,8003
24_1	24	24	701,2940
25_1	25	25	512,3707
26_1	26	26	321,1515
27_1	27	27	1128,4606
28_1	28	28	1177,673
29_1	29	29	929,1154
30_1	30	30	639,4130
31_1	31	31	337,5793
32_1	32	32	516,1120
33_1	33	33	169,1524
34_1	34	34	197,1152
35_1	35	40	1268,543
36_1	36	36	1240,4484
37_1	37	37	503,9721
38_1	38	38	629,6368
39_1	39	39	1090,2620
40_1	40	40	910,7740

Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan *Wavelet-1D-PCA+KNN*

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	16	2942,5782
2_1	2	2	1597,3887
3_1	3	3	654,2113
4_1	4	4	819,9438
5_1	5	5	1195,7727
6_1	6	6	589,8915
7_1	7	7	853,1899
8_1	8	8	689,9506
9_1	9	9	545,3984
10_1	10	10	2628,7090
11_1	11	11	1135,0160
12_1	12	12	724,4790
13_1	13	13	269,7509
14_1	14	14	2190,9183
15_1	15	15	410,4446
16_1	16	16	1238,1475
17_1	17	17	838,7781
18_1	18	18	983,699
19_1	19	19	1537,4216
20_1	20	20	573,3834
21_1	21	21	517,1048
22_1	22	22	1410,5812
23_1	23	23	713,6554
24_1	24	24	1463,4582
25_1	25	25	897,0898
26_1	26	26	514,6545
27_1	27	27	1293,3527
28_1	28	28	1653,8716
29_1	29	29	871,0462
30_1	30	30	770,3558
31_1	31	31	936,9655
32_1	32	32	633,4564
33_1	33	33	342,7008
34_1	34	34	285,0840
35_1	35	35	1098,908
36_1	36	36	1410,4751
37_1	37	37	1071,6469
38_1	38	38	897,5520
39_1	39	33	1755,9117
40_1	40	40	725,0087

Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan 2D-
PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	17	6664,589
2_1	2	2	2536,6421
3_1	3	3	3052,8538
4_1	4	4	3871,1251
5_1	5	40	4525,0406
6_1	6	6	2328,7479
7_1	7	7	4247,4737
8_1	8	8	3645,2352
9_1	9	9	4696,9284
10_1	10	38	7695,9353
11_1	11	11	3750,5218
12_1	12	12	3058,7196
13_1	13	13	1898,7580
14_1	14	14	5781,9024
15_1	15	15	2366,7538
16_1	16	16	4654,7976
17_1	17	17	2953,3222
18_1	18	18	4293,2166
19_1	19	19	2965,8219
20_1	20	20	3094,8971
21_1	21	21	2492,9442
22_1	22	22	3616,0959
23_1	23	23	3314,1571
24_1	24	24	3001,0489
25_1	25	25	2792,4526
26_1	26	26	2859,1993
27_1	27	27	4063,7032
28_1	28	28	5133,6025
29_1	29	29	3685,2611
30_1	30	30	2409,599
31_1	31	31	3076,536
32_1	32	32	3063,5041
33_1	33	33	1575,0897
34_1	34	34	1549,4323
35_1	35	35	5915,9721
36_1	36	36	5631,5919
37_1	37	37	3585,8409
38_1	38	38	2628,2858
39_1	39	39	4806,0307
40_1	40	40	4658,2202

Hasil Percobaan 90% *Training Images* dan 10% *Testing Images* dengan *Wavelet-2D-PCA+KNN*

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	9067,2753
2_1	2	2	4296,4452
3_1	3	3	3014,0086
4_1	4	4	4343,4193
5_1	5	5	5225,0660
6_1	6	6	2355,9597
7_1	7	7	3721,9896
8_1	8	8	3057,7524
9_1	9	9	4869,7728
10_1	10	8	8867,7970
11_1	11	11	3769,8556
12_1	12	12	3522,2864
13_1	13	13	1606,9584
14_1	14	14	7928,0058
15_1	15	15	2660,7463
16_1	16	16	4583,6723
17_1	17	17	3125,6987
18_1	18	18	4239,9646
19_1	19	19	4027,7042
20_1	20	20	3400,9776
21_1	21	21	2741,4259
22_1	22	22	4138,8881
23_1	23	23	3507,6340
24_1	24	24	4331,8034
25_1	25	25	3259,7800
26_1	26	26	2394,9057
27_1	27	27	4555,2143
28_1	28	28	6113,8279
29_1	29	29	4122,3113
30_1	30	30	2292,6212
31_1	31	31	4779,0615
32_1	32	32	3895,0645
33_1	33	33	1374,9465
34_1	34	34	1422,7447
35_1	35	35	6700,7931
36_1	36	36	5828,6862
37_1	37	37	3882,4224
38_1	38	38	2871,5847
39_1	39	39	7311,2743
40_1	40	40	4402,5291

Lampiran 3. Hasil Percobaan Menggunakan YALE

Hasil Percobaan 1 *Training Images* dan 10 *Testing Images* dengan 1D-PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	9043,1595	4_10	4	4	5183,4511
1_2	1	1	8723,348	5_1	5	5	3858,1710
1_3	1	1	6980,227	5_2	5	5	4052,2013
1_4	1	2	7922,4496	5_3	5	5	6693,9055
1_5	1	1	8451,824	5_4	5	5	3210,1861
1_6	1	7	17677,8501	5_5	5	5	4193,3288
1_7	1	1	8517,9394	5_6	5	15	16962,1785
1_8	1	1	7484,4676	5_7	5	5	4602,4459
1_9	1	1	7857,0000	5_8	5	5	4577,4927
1_10	1	1	7069,4236	5_9	5	5	2974,7716
2_1	2	12	8744,1837	5_10	5	5	3908,9595
2_2	2	2	5667,2318			...	
2_3	2	2	9128,3458	11_1	11	11	6229,6391
2_4	2	2	6314,8247	11_2	11	11	7015,4858
2_5	2	2	6314,8247	11_3	11	11	4469,2105
2_6	2	1	10638,2823	11_4	11	11	7473,8459
2_7	2	2	7493,3860	11_5	11	11	6701,0878
2_8	2	2	6437,0252	11_6	11	11	5325,0003
2_9	2	2	8661,0580	11_7	11	11	8601,0937
2_10	2	1	8943,9195	11_8	11	11	6280,6892
3_1	3	3	6575,9229	11_9	11	11	7934,6793
3_2	3	15	6126,5400	11_10	11	11	7090,8401
3_3	3	3	11808,7559	12_1	12	12	5987,6489
3_4	3	15	6750,1593	12_2	12	5	7716,1255
3_5	3	15	6750,1593	12_3	12	12	9496,8235
3_6	3	15	16655,2834	12_4	12	5	8438,0629
3_7	3	15	6654,7561	12_5	12	5	8438,0629
3_8	3	15	6421,2801	12_6	12	12	5836,9354
3_9	3	8	7538,2711	12_7	12	10	8884,1732
3_10	3	15	6085,914	12_8	12	10	9570,3609
4_1	4	4	6779,7397	12_9	12	5	8468,5518
4_2	4	4	5321,9309	12_10	12	12	8154,2996
4_3	4	1	9827,0096	13_1	13	5	10264,125
4_4	4	4	3811,7002	13_2	13	5	9386,1801
4_5	4	4	7420,4379	13_3	13	5	12680,1749
4_6	4	7	16533,888	13_4	13	5	9869,8222
4_7	4	4	7420,4379	13_5	13	5	9128,8561
4_8	4	4	6325,1024	13_6	13	13	11923,2976
4_9	4	4	5421,6080	13_7	13	13	9976,7874

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
13_8	13	13	9465,1630
13_9	13	13	8497,7881
13_10	13	5	9515,0022
14_1	14	14	3182,3257
14_2	14	14	1767,9757
14_3	14	3	15810,041
14_4	14	14	6282,0014
14_5	14	14	6282,0014
14_6	14	14	4881,2910
14_7	14	14	2734,3177
14_8	14	14	7710,8677
14_9	14	14	3480,6567
14_10	14	14	4167,3958
15_1	15	7	8526,0413
15_2	15	15	8081,7813
15_3	15	11	10738,2594
15_4	15	8	6662,2951
15_5	15	8	5589,5992
15_6	15	15	12367,0000
15_7	15	15	9039,0497
15_8	15	8	6368,3374
15_9	15	8	7230,3102
15_10	15	3	7819,0000

Hasil Percobaan 1 *Training Images* dan 10 *Testing Images* dengan *Wavelet+1D-PCA+KNN*

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	9077,0738	2_2	2	2	6147,4745
1_2	1	1	8769,4656	2_3	2	2	11196,3824
1_3	1	1	9059,5072	2_4	2	2	6623,3202
1_4	1	2	7638,7788	2_5	2	2	6623,3202
1_5	1	1	8454,0195	2_6	2	1	11493,7503
1_6	1	7	17848,8981	2_7	2	2	7529,9674
1_7	1	1	8541,8922	2_8	2	2	6718,9436
1_8	1	1	7152,9331	2_9	2	2	8284,0708
1_9	1	1	7848,6348	2_10	2	2	9127,7493
1_10	1	1	7086,2483	3_1	3	3	6530,6167
2_1	2	5	8368,6388	3_2	3	15	6635,7447

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
3_3	3	3	12826,8487	11_9	11	11	7774,4284
3_4	3	15	7346,1448	11_10	11	11	6621,4666
3_5	3	15	7346,1448	12_1	12	12	6529,7081
3_6	3	7	17792,4745	12_2	12	5	7709,6837
3_7	3	15	7773,1528	12_3	12	12	10274,7796
3_8	3	15	6976,4194	12_4	12	5	6447,9233
3_9	3	8	7517,6559	12_5	12	5	6447,9233
3_10	3	15	6470,2122	12_6	12	12	6132,0629
4_1	4	4	6320,0668	12_7	12	3	9397,4583
4_2	4	4	4954,1840	12_8	12	3	9272,5242
4_3	4	1	11467,6406	12_9	12	5	8474,4457
4_4	4	4	4332,4313	12_10	12	12	8244,1476
4_5	4	4	6700,1059	13_1	13	5	10239,3466
4_6	4	7	16578,4200	13_2	13	5	9326,5303
4_7	4	4	6700,1059	13_3	13	3	12627,1486
4_8	4	4	6117,2826	13_4	13	5	9829,7186
4_9	4	4	5829,6714	13_5	13	5	9167,2816
4_10	4	4	5486,2158	13_6	13	13	13467,8278
5_1	5	5	3682,7053	13_7	13	13	9875,2904
5_2	5	5	3759,5999	13_8	13	13	9484,9120
5_3	5	5	7924,8573	13_9	13	13	8437,8348
5_4	5	5	3177,0748	13_10	13	5	9679,7110
5_5	5	5	3942,8545	14_1	14	14	3352,2739
5_6	5	7	17469,2907	14_2	14	14	1904,4799
5_7	5	5	4241,6012	14_3	14	3	15126,2708
5_8	5	5	4219,0099	14_4	14	14	5751,2523
5_9	5	5	2828,1322	14_5	14	14	5751,2523
5_10	5	5	3597,4282	14_6	14	14	5516,8996
...				14_7	14	14	3200,2902
11_1	11	11	5993,3955	14_8	14	14	7169,4044
11_2	11	11	6653,3279	14_9	14	14	3567,0255
11_3	11	11	5834,5134	14_10	14	14	3924,8354
11_4	11	11	7008,3665	15_1	15	7	8422,3325
11_5	11	11	6343,596	15_2	15	15	8127,4649
11_6	11	11	5832,6183	15_3	15	11	11345,3883
11_7	11	11	8238,0684	15_4	15	8	6389,4151
11_8	11	11	5861,6389	15_5	15	15	6148,6080

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
15_6	15	15	12934,3182
15_7	15	15	9118,4142
15_8	15	8	6201,3059
15_9	15	8	7520,9755
15_10	15	3	8041,0902

Hasil Percobaan 1 *Training Images* dan 10 *Testing Images* dengan 2D-PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	25133,7165	3_10	3	15	26695,2883
1_2	1	1	24976,6049	4_1	4	4	19400,7755
1_3	1	1	29175,8700	4_2	4	4	15512,5246
1_4	1	12	22058,1930	4_3	4	1	31113,0791
1_5	1	1	25437,6051	4_4	4	4	12350,1462
1_6	1	8	46807,0823	4_5	4	4	19050,0456
1_7	1	1	23878,0676	4_6	4	15	42567,0029
1_8	1	1	22234,5222	4_7	4	4	19050,0456
1_9	1	1	24056,3268	4_8	4	4	17219,07
1_10	1	1	21116,7999	4_9	4	4	15812,9108
2_1	2	1	22318,2286	4_10	4	4	15077,0857
2_2	2	2	195999,1857	5_1	5	5	13035,2438
2_3	2	2	2870,5697	5_2	5	5	13917,523
2_4	2	2	18370,7974	5_3	5	5	25333,7941
2_5	2	2	18370,7974	5_4	5	5	12742,9324
2_6	2	4	32236,2505	5_5	5	5	13962,2053
2_7	2	2	20647,2469	5_6	5	15	44847,2004
2_8	2	2	18400,8305	5_7	5	5	14827,8944
2_9	2	2	24802,1121	5_8	5	5	14833,9979
2_10	2	1	23120,5470	5_9	5	5	12532,1222
3_1	3	3	18043,4953	5_10	5	5	13188,3127
3_2	3	15	25671,1526			...	
3_3	3	3	37754,0930	11_1	11	11	18484,465
3_4	3	9	25365,2751	11_2	11	11	22966,0667
3_5	3	9	25365,2751	11_3	11	11	25300,6743
3_6	3	15	42461,2013	11_4	11	11	23581,7649
3_7	3	9	2681,5084	11_5	11	11	22597,4803
3_8	3	15	26366,4211	11_6	11	11	18324,1129
3_9	3	8	2308,0596	11_7	11	11	27593,1845

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
11_8	11	11	19709,5103	13_10	13	5	25588,6704
11_9	11	11	26095,0922	14_1	14	14	10705,6934
11_10	11	11	21056,6435	14_2	14	14	10109,1502
12_1	12	12	1897,0285	14_3	14	3	44853,2928
12_2	12	5	23512,4074	14_4	14	14	15664,2989
12_3	12	12	30113,0437	14_5	14	14	15664,2989
12_4	12	5	26092,8667	14_6	14	14	18999,1309
12_5	12	5	26092,8667	14_7	14	14	12899,4570
12_6	12	12	17849,0268	14_8	14	14	18231,4920
12_7	12	10	27569,2136	14_9	14	14	14147,7947
12_8	12	10	28799,0432	14_10	14	14	13698,3018
12_9	12	12	22363,5611	15_1	15	7	27088,0629
12_10	12	12	22239,6955	15_2	15	15	22248,5388
13_1	13	5	25845,8207	15_3	15	11	40345,0799
13_2	13	5	24732,6061	15_4	15	8	28231,7601
13_3	13	3	37035,6957	15_5	15	15	19188,4429
13_4	13	5	25597,2207	15_6	15	15	34971,5168
13_5	13	5	24629,4897	15_7	15	15	23600,7376
13_6	13	13	33279,4592	15_8	15	8	28322,8674
13_7	13	13	25505,9759	15_9	15	7	27964,4892
13_8	13	13	24073,5256	15_10	15	3	23009,4743
13_9	13	13	21950,1313				

Hasil Percobaan 1 *Training Images* dan 10 *Testing Images* dengan *Wavelet+2D-PCA+KNN*

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	1	26273,4828	2_4	2	2	19204,7722
1_2	1	1	25842,9453	2_5	2	2	19204,7722
1_3	1	1	37504,0281	2_6	2	4	33795,2631
1_4	1	2	22646,0477	2_7	2	2	21386,6547
1_5	1	1	26247,9288	2_8	2	2	19254,2321
1_6	1	8	49122,6850	2_9	2	2	25247,2338
1_7	1	1	25239,4868	2_10	2	2	22884,6094
1_8	1	1	23096,0233	3_1	3	3	18242,1497
1_9	1	1	23918,0642	3_2	3	15	26380,3085
1_10	1	1	21584,6595	3_3	3	3	4463,7283
2_1	2	4	24545,8238	3_4	3	15	26481,2618
2_2	2	2	20746,1231	3_5	3	15	26481,2618
2_3	2	12	32906,564				

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
3_6	3	15	48606,1726	12_1	12	12	6529,7081
3_7	3	15	28995,4943	12_2	12	5	7709,6837
3_8	3	15	26854,8942	12_3	12	12	10274,7796
3_9	3	8	25840,9982	12_4	12	5	6447,9233
3_10	3	15	26967,4843	12_5	12	5	6447,9233
4_1	4	4	19528,0786	12_6	12	12	6132,0629
4_2	4	4	15050,9277	12_7	12	3	9397,4583
4_3	4	1	36704,5021	12_8	12	3	9272,5242
4_4	4	4	13652,3371	12_9	12	5	8474,4457
4_5	4	4	1826,8148	12_10	12	12	8244,1476
4_6	4	7	45757,4329	13_1	13	5	10239,3466
4_7	4	4	18260,8148	13_2	13	5	9326,5303
4_8	4	4	17686,1106	13_3	13	3	12627,1486
4_9	4	4	17104,1775	13_4	13	5	9829,7186
4_10	4	4	15759,3094	13_5	13	5	9167,2816
5_1	5	5	14075,2348	13_6	13	13	13467,8278
5_2	5	5	14850,8018	13_7	13	13	9875,2904
5_3	5	5	14850,8018	13_8	13	13	9484,9120
5_4	5	5	30742,7633	13_9	13	13	8437,8348
5_5	5	5	13735,4246	13_10	13	5	9679,7110
5_6	5	15	15174,9197	14_1	14	14	3352,2739
5_7	5	5	48971,2485	14_2	14	14	1904,4799
5_8	5	5	15826,6678	14_3	14	3	15126,2708
5_9	5	5	15827,3807	14_4	14	14	5751,2523
5_10	5	5	13751,6715	14_5	14	14	5751,2523
		...		14_6	14	14	5516,8996
11_1	11	11	5993,3955	14_7	14	14	3200,2902
11_2	11	11	6653,3279	14_8	14	14	7169,4044
11_3	11	11	5834,5134	14_9	14	14	3567,0255
11_4	11	11	7008,3665	14_10	14	14	3924,8354
11_5	11	11	6343,596	15_1	15	7	8422,3325
11_6	11	11	5832,6183	15_2	15	15	8127,4649
11_7	11	11	8238,0684	15_3	15	11	11345,3883
11_8	11	11	5861,6389	15_4	15	8	6389,4151
11_9	11	11	7774,4284	15_5	15	15	6148,608
11_10	11	11	6621,4666	15_6	15	15	12934,3182
				15_7	15	15	9118,4142
				15_8	15	8	6201,3059
				15_9	15	8	7520,9755
				15_10	15	3	8041,0902

Hasil Percobaan 6 *Training Images* dan 5 *Testing Images* dengan PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	6	17818,5102	8_3	8	8	4549,0755
1_2	1	1	1039,6784	8_4	8	9	6329,2751
1_3	1	1	1729,6801	8_5	8	10	4914,1638
1_4	1	1	3703,4842	9_1	9	7	14661,6426
1_5	1	1	3912,4024	9_2	9	9	4344,0559
2_1	2	4	10471,8616	9_3	9	9	3134,5362
2_2	2	2	1833,9824	9_4	9	9	3142,8800
2_3	2	2	332,0957	9_5	9	9	5045,2733
2_4	2	2	3858,5551	10_1	10	10	7590,7512
2_5	2	2	4420,3525	10_2	10	10	635,0125
3_1	3	6	17670,7392	10_3	10	10	689,8969
3_2	3	3	1925,4588	10_4	10	10	1041,4349
3_3	3	3	1233,4157	10_5	10	10	1977,9001
3_4	3	8	4418,1097	11_1	11	11	6470,2839
3_5	3	3	1602,8037	11_2	11	11	5109,9859
4_1	4	6	19894,4793	11_3	11	11	1728,6854
4_2	4	4	1,1756	11_4	11	11	2831,3044
4_3	4	4	1254,9404	11_5	11	11	2450,4102
4_4	4	4	1729,6692	12_1	12	12	5690,1179
4_5	4	4	1833,3857	12_2	12	12	4682,4314
5_1	5	6	16058,7159	12_3	12	12	5496,3273
5_2	5	5	538,7022	12_4	12	12	3314,9434
5_3	5	5	570,4888	12_5	12	12	4145,8446
5_4	5	5	1405,6672	13_1	13	13	15206,3955
5_5	5	5	519,348	13_2	13	13	944,8876
6_1	6	6	9623,0494	13_3	13	13	1531,5099
6_2	6	6	958,5458	13_4	13	13	2473,0651
6_3	6	6	2190,7024	13_5	13	13	1336,9662
6_4	6	6	760,9925	14_1	14	6	9955,0951
6_5	6	6	1667,6375	14_2	14	14	1552,4974
7_1	7	15	15818,0898	14_3	14	14	1920,6102
7_2	7	7	3587,0198	14_4	14	14	1962,7291
7_3	7	7	634,4073	14_5	14	14	1925,4918
7_4	7	7	3986,8498	15_1	15	6	14849,4687
7_5	7	7	3910,3721	15_2	15	15	1107,9144
8_1	8	7	13497,7471	15_3	15	15	851,8163
8_2	8	9	3726,9662	15_4	15	15	1584,3185
				15_5	15	15	4007,7522

Hasil Percobaan 6 *Training Images* dan 5 *Testing Images* dengan *Wavelet+ 1D-PCA+KNN*

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	6	27357,0464	8_4	8	9	6098,2703
1_2	1	1	1017,7336	8_5	8	15	4969,9414
1_3	1	1	1615,7264	9_1	9	6	1483,5771
1_4	1	1	3462,8403	9_2	9	9	4123,3388
1_5	1	1	3625,0878	9_3	9	9	2841,1854
2_1	2	4	11212,4957	9_4	9	9	2972,6463
2_2	2	2	1644,3802	9_5	9	9	4544,3473
2_3	2	2	320,3679	10_1	10	10	8620,3786
2_4	2	2	3566,2291	10_2	10	10	564,8838
2_5	2	2	4118,4979	10_3	10	10	687,0020
3_1	3	6	18061,7608	10_4	10	10	1014,3393
3_2	3	3	2187,0599	10_5	10	10	1886,6471
3_3	3	3	1178,4357	11_1	11	11	8212,5338
3_4	3	8	4773,031	11_2	11	11	4704,3084
3_5	3	3	1404,4772	11_3	11	11	1741,0857
4_1	4	6	19984,3407	11_4	11	11	2834,7291
4_2	4	4	4,7748	11_5	11	11	2346,3570
4_3	4	4	1277,917	12_1	12	12	5756,0333
4_4	4	4	1839,8515	12_2	12	12	4218,3813
4_5	4	4	1786,2158	12_3	12	12	5054,7653
5_1	5	6	16558,0983	12_4	12	12	3217,1676
5_2	5	5	489,9853	12_5	12	12	3,999,3849
5_3	5	5	523,6685	13_1	13	7	17350,0415
5_4	5	5	1437,9923	13_2	13	13	865,8874
5_5	5	5	532,3644	13_3	13	13	1515,7825
6_1	6	6	10328,3775	13_4	13	13	2262,3905
6_2	6	6	1203,1861	13_5	13	13	1305,4953
6_3	6	6	2012,8216	14_1	14	6	9251,9965
6_4	6	6	702,5929	14_2	14	14	1349,8931
6_5	6	6	2019,9172	14_3	14	14	1732,6782
7_1	7	7	17083,7654	14_4	14	14	1984,902
7_2	7	7	3879,1836	14_5	14	14	1895,9325
7_3	7	7	644,6038	15_1	15	6	14296,2576
7_4	7	7	4229,4647	15_2	15	15	1121,6627
7_5	7	7	4430,6243	15_3	15	15	862,9214
8_1	8	10	15001,2972	15_4	15	15	1830,8546
8_2	8	9	4414,6001	15_5	15	15	3944,3036
8_3	8	8	4662,4967				

Hasil Percobaan 6 *Training Images* dan 5 *Testing Images* dengan 2D-PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	6	50153,2634	8_4	8	13	20117,2686
1_2	1	1	8486,1120	8_5	8	8	16039,7316
1_3	1	1	8018,0573	9_1	9	14	31099,5167
1_4	1	1	13200,0715	9_2	9	9	1498,1559
1_5	1	1	11863,5796	9_3	9	9	11270,3696
2_1	2	2	33971,1342	9_4	9	9	11452,5691
2_2	2	2	7461,6693	9_5	9	9	13885,112
2_3	2	2	2228,4284	10_1	10	10	20853,8455
2_4	2	2	15639,2599	10_2	10	10	5172,4242
2_5	2	2	9362,7048	10_3	10	10	5022,3250
3_1	3	14	49427,1629	10_4	10	10	11244,3555
3_2	3	3	7665,5396	10_5	10	10	7212,6303
3_3	3	3	5252,5296	11_1	11	11	20501,5059
3_4	3	3	17434,9958	11_2	11	11	22923,4184
3_5	3	3	7251,6789	11_3	11	11	10621,2578
4_1	4	14	49264,2618	11_4	11	11	17102,9943
4_2	4	4	1,2540	11_5	11	11	13019,6795
4_3	4	4	8788,6331	12_1	12	12	17621,9029
4_4	4	4	8322,54444	12_2	12	12	13492,5327
4_5	4	4	7013,5795	12_3	12	12	14293,4181
5_1	5	6	45207,4337	12_4	12	12	12115,3806
5_2	5	5	4498,7866	12_5	12	12	14003,6513
5_3	5	5	4008,9356	13_1	13	14	39027,0888
5_4	5	5	7117,251	13_2	13	13	3559,0411
5_5	5	5	3708,6071	13_3	13	13	6876,6973
6_1	6	6	24272,4542	13_4	13	13	8559,2399
6_2	6	6	7938,2443	13_5	13	13	7418,0426
6_3	6	6	9463,6144	14_1	14	14	24815,2845
6_4	6	6	9863,2027	14_2	14	14	11126,7681
6_5	6	6	8488,0245	14_3	14	14	5642,1566
7_1	7	14	38763,8738	14_4	14	14	9599,0316
7_2	7	7	13724,2735	14_5	14	14	7996,1251
7_3	7	7	3097,6194	15_1	15	14	35741,5583
7_4	7	7	15279,3043	15_2	15	15	6672,963
7_5	7	7	17188,9154	15_3	15	15	3274,8005
8_1	8	8	31023,0965	15_4	15	15	7721,4827
8_2	8	9	18382,2820	15_5	15	15	12382,4764
8_3	8	8	13409,3045				

Hasil Percobaan 6 *Training Images* dan 5 *Testing Images* dengan DWT- 2D-
PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>	Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidean Distance</i>
1_1	1	6	50661,7686	8_4	8	10	20745,8086
1_2	1	1	7433,4260	8_5	8	8	15612,3614
1_3	1	1	7028,7806	9_1	9	14	3296,6116
1_4	1	1	11944,1853	9_2	9	9	12986,7173
1_5	1	1	10906,7796	9_3	9	9	10402,3883
2_1	2	4	36832,0743	9_4	9	9	10889,0194
2_2	2	2	6749,9479	9_5	9	9	13337,1868
2_3	2	2	1791,8111	10_1	10	10	21999,5141
2_4	2	2	15928,416	10_2	10	10	4403,0889
2_5	2	2	8834,7410	10_3	10	10	4519,3530
3_1	3	14	49058,0054	10_4	10	10	9597,7966
3_2	3	3	9063,0144	10_5	10	10	6311,1463
3_3	3	3	4573,3918	11_1	11	11	24887,3264
3_4	3	3	17230,0547	11_2	11	11	22383,4941
3_5	3	3	6452,9143	11_3	11	11	9116,9317
4_1	4	14	49590,4445	11_4	11	11	16123,9620
4_2	4	4	1,1420	11_5	11	11	11157,3489
4_3	4	4	7874,2387	12_1	12	12	18504,8995
4_4	4	4	8350,9267	12_2	12	12	11976,9764
4_5	4	4	6568,8269	12_3	12	12	13047,0037
5_1	5	6	46812,9769	12_4	12	12	1134,9801
5_2	5	5	4020,5848	12_5	12	12	12866,9737
5_3	5	5	3284,9232	13_1	13	13	40514,2113
5_4	5	5	6740,9527	13_2	13	13	3159,0528
5_5	5	5	3237,5865	13_3	13	13	5721,7557
6_1	6	6	24267,8089	13_4	13	13	7614,2410
6_2	6	6	7201,2874	13_5	13	13	6947,0172
6_3	6	6	9053,5413	14_1	14	14	24984,4142
6_4	6	6	8912,6034	14_2	14	14	10676,9392
6_5	6	6	8376,4253	14_3	14	14	5070,3387
7_1	7	14	41741,9179	14_4	14	14	9345,488
7_2	7	7	14400,0405	14_5	14	14	7537,1554
7_3	7	7	2605,9979	15_1	15	14	37079,5648
7_4	7	7	16637,0641	15_2	15	15	5814,8303
7_5	7	7	18341,3002	15_3	15	15	2945,3994
8_1	8	8	33329,3343	15_4	15	15	7151,0917
8_2	8	9	18488,6819	15_5	15	15	11358,2213
8_3	8	8	13293,6985				

Hasil Percobaan 10 *Training Images* dan 1 *Testing Images* dengan 1D-PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidian Distance</i>
1_1	1	1	1872,0887
2_1	2	2	2793,0378
3_1	3	3	1236,2666
4_1	4	4	1026,4637
5_1	5	5	558,4457
6_1	6	6	966,5066
7_1	7	7	1611,1848
8_1	8	10	4318,8841
9_1	9	9	3139,8519
10_1	10	10	1591,2099
11_1	11	11	1382,777
12_1	12	12	2229,0147
13_1	13	13	1353,9322
14_1	14	14	1088,0463
15_1	15	15	3325,7946

Hasil Percobaan 1 *Training Images* dan 10 *Testing Images* dengan *Wavelet*+1D-PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidian Distance</i>
1_1	1	1	1812,6937
2_1	2	2	2429,8504
3_1	3	3	1067,4464
4_1	4	4	1208,9305
5_1	5	5	529,4681
6_1	6	6	1126,4688
7_1	7	7	1572,6393
8_1	8	10	4562,3550
9_1	9	9	2841,3569
10_1	10	10	1480,6092
11_1	11	11	1278,9046
12_1	12	12	2137,3866
13_1	13	13	1322,6485
14_1	14	14	1161,8795
15_1	15	15	3241,8579

Hasil Percobaan 10 *Training Images* dan 1 *Testing Images* dengan 2D-PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidian Distance</i>
1_1	1	1	8660,7126
2_1	2	2	8380,7358
3_1	3	3	5714,5878
4_1	4	4	6916,1544
5_1	5	5	3605,4198
6_1	6	6	7034,3670
7_1	7	7	12515,5519
8_1	8	8	16349,8114
9_1	9	9	11384,1015
10_1	10	10	6257,3567
11_1	11	11	5888,1203
12_1	12	12	8247,3907
13_1	13	13	7263,6570
14_1	14	14	7632,0000
15_1	15	15	11372,8418

Hasil Percobaan 10 *Training Images* dan 1 *Testing Images* dengan *Wavelet*+2D-PCA+KNN

Citra Uji	Kelas Uji	Kelas Latih	<i>Euclidian Distance</i>
1_1	1	1	7607,5258
2_1	2	2	7592,8819
3_1	3	3	5091,0839
4_1	4	4	6296,3840
5_1	5	5	3200,4050
6_1	6	6	6758,4075
7_1	7	7	11685,3633
8_1	8	8	15602,2509
9_1	9	9	10972,2769
10_1	10	10	5621,3128
11_1	11	11	4998,5984
12_1	12	12	7515,8012
13_1	13	13	6843,8814
14_1	14	14	7368,3067
15_1	15	15	10345,3079