

IMPLEMENTASI EKSTRAKSI FITUR DAN K-NEAREST NEIGHTBOR UNTUK IDENTIFIKASI WAJAH PERSONAL

Danar Putra Pamungkas¹, Ahmad Bagus Setiawan²
Teknik Informatika, Universitas Nusantara PGRI Kediri
Jl. KH. Achmad Dahlan Gg. 1 Mooroto Kota Kediri
E-mail: danar@unpkediri.ac.id, ahmadbagus@unpkediri.ac.id

Abstraks

Wajah atau muka adalah bagian depan kepala pada manusia meliputi wilayah dari dahi hingga dagu, termasuk rambut, dahi, alis, mata, hidung, pipi, mulut, bibir, gigi, kulit, dan dagu. Sebuah sistem biometrika berdasarkan wajah diharapkan dapat menutup kelemahan sistem presensi konvensional yang berdasarkan tanda tangan. Beberapa kelemahan sistem tersebut antara lain, adanya celah kecurangan dalam proses tanda tangan dan waktu perekapan yang cukup lama. Penulis menggunakan metode ekstraksi fitur Eigenface PCA sedangkan untuk klasifikasi menggunakan K-Nearest Neighbor. Akurasi pengenalan wajah dengan menggunakan metode ekstraksi fitur eigenface dan K-NN mencapai 80%. Nilai rata-rata FAR terendah adalah 20% sedangkan FRR 15%. Semakin banyak data latih yang digunakan akurasi semakin tinggi. Akurasi optimal didapat pada kondisi jarak 50 cm dengan cahaya terang.

Kata Kunci: wajah, ekstraksi, fitur, K-NN

1. PENDAHULUAN

Ciri khas personal yang melekat pada tubuh seseorang antara lain sidik jari, wajah, tangan, kaki, dan retina. Wajah atau muka adalah bagian depan kepala pada manusia meliputi wilayah dari dahi hingga dagu, termasuk rambut, dahi, alis, mata, hidung, pipi, mulut, bibir, gigi, kulit, dan dagu (KBBI, 2008). Elemen-elemen itulah yang membuat wajah manusia memiliki bentuk yang berbeda-beda sehingga wajah sering digunakan sebagai identitas suatu individu dengan individu yang lain. Teknologi pengenalan diri dengan menggunakan bagian tubuh atau perilaku manusia disebut dengan sistem biometrika (Putra, 2009).

Sebuah sistem biometrika berdasarkan wajah diharapkan dapat menutup kelemahan sistem presensi konvensional yang berdasarkan tanda tangan. Beberapa kelemahan sistem tersebut antara lain, adanya celah kecurangan dalam proses tanda tangan dan waktu perekapan yang cukup lama. Penulis akan menggunakan ciri wajah personal untuk digunakan dalam sistem presensi. Sistem biometrika yang akan dibuat membutuhkan data wajah personal. Data wajah tersebut merupakan data masukan pada sistem yang berupa data citra. Citra adalah gambar pada dua dimensi (Aditya, 2013). Sistem yang menggunakan data masukan citra merupakan salah satu pengolahan citra. Pengolahan citra merupakan suatu proses yang dilakukan dengan masukan berupa citra dan hasilnya juga berupa citra (Efendy, 2009).

Sistem pengenalan wajah membutuhkan sebuah ciri dari suatu citra untuk dapat dikenali kemudian ciri tersebut akan dicocokkan dengan ciri citra lain. proses tersebut membutuhkan suatu metode ekstraksi ciri atau fitur dan klasifikasi. Metode ekstraksi fitur dan jumlah neighbor yang melakukan voting mempengaruhi tingkat akurasi klasifikasi gambar (Fuad, 2017). Proses pengenalan wajah dapat

berbeda dari data objek yang digunakan (Wang, 2017). Sistem biometrika untuk pengenalan wajah memiliki tingkat akurasi 87% dengan menggunakan klasifikasi jaringan syaraf tiruan sedangkan dengan klasifikasi *euclidean distance* memiliki akurasi 68.45% (Fitriyah, 2015). Selain metode Klasifikasi Jaringan Saraf Tiruan metode Principal Component Analysis (PCA) juga dapat digunakan untuk pengenalan wajah. *Eigenface* merupakan salah satu metode pengenalan wajah berdasarkan *Principal Component Analysis* (PCA) yang mudah diimplementasikan (Fatta, 2009). Pengenalan wajah menggunakan metode PCA dengan menggunakan data sebanyak 130 gambar memiliki tingkat akurasi 82,27% (Salamun, 2016). Deteksi wajah dengan menggunakan metode *eigenface* dapat memberikan tingkat kemiripan dengan nilai prosentase 80% (Putra, 2013). Metode Eigenface dan City Block Distence dapat mengenali wajah dengan rata-rata akurasi 69,86% (Pamungkas, 2017). Akurasi identifikasi citra dengan metode klasifikasi K-Nearest Neighbor mencapai 88,9% (Wulanningrum, 2018).

Penulis menggunakan metode ekstraksi fitur Eigenface PCA sedangkan untuk klasifikasi menggunakan K-Nearest Neighbor. Citra yang digunakan berformat *jpg* dan *bmp*. Resolusi citra tes adalah 640x480 pixel. Deteksi wajah menggunakan Webcam 0,3 MP. Data citra sampel diambil dua kali dari 10 orang mahasiswa lingkup UN PGRI Kediri.

1.1 Pengenalan Pola

Pengenalan Pola atau *Pattern Recognition* adalah suatu proses atau rangkaian yang bertujuan mengklasifikasikan data numerik dan *symbol*. Banyak teknik statistik dan sintksis yang telah dikembangkan untuk keperluan klasifikasi pola dan teknik-teknik ini dapat memainkan peran yang penting dalam *system visual* ntuk pengenalan obyek

yang biasanya memerlukan banyak teknik. Bentuk-bentuk obyek tertentu dalam dunia nyata yang sangat kompleks dapat dibandingkan dengan pola-pola dasar di dalam citra sehingga penggolongan obyek yang bersangkutan dapat dilakukan dengan lebih mudah (Ahmad, 2009).

Sistem Pengenalan pola dasar terdiri dari (Fatta, 2009):

- a. Sensor
Sensur digunakan untuk menangkap objek yang cri atau fiturnya akan dieksraksi.
- b. Mekanisme Pre-Processing
Mekanisme pengolahan objek yang ditangkap oleh sensor biasanya digunakan untuk mengurangi kompleksitas ciri yang akan dipakai untuk proses klasifikasi.
- c. Mekanisme Pencari Fitu(manual/otomatis)
Bagian ini digunakan untuk mengekstraksi ciri yang telah melalui tahap preprocessing untuk memisahkannya dari fitur-fitur pada objek yang tidak diperlukan dalam proses klasifikasi.
- d. Algoritma Pemilah
Pada tahapan ini proses klasifikasi dilakukan dengan menggunakan algoritma klasifikasi tertentu. Hasil dari tahapan ini adalah klasifikasi dari objek yang ditangkap ke dalam kriteria-kriteria yang telah ditentukan.

1.2 Pengenalan Wajah

Pengenalan wajah merupakan salah satu pendekatan pengenalan pola untuk keperluan identifikasi *personal* disamping pendekatan biometrik lainnya seperti pengenalan sidik jari, tanda tangan, pengenalan citra wajah berhubungan dengan obyek yang tidak pernah sama, karena adanya bagian-bagian yang dapat berubah (Putra, 2013).

1.3 Eigenface

Salah satu metode untuk ekstraksi fitur adalah *Eigenface*. *Eigenface* salah satu algoritma pengenalan wajah yang didasarkan pada *Principle Component Analysis* (PCA) yang dikembangkan di MIT. Algoritma eigenface pada training image dipresentasikan dalam sebuah *vector flat* (gabungan vector) dan digabung bersama-sama menjadi sebuah matriks tunggal. *Eigenfaces* dari masing-masing citra kemudian diekstraksi dan disimpan dalam file *temporary* atau database. *Test image* yang masuk didefinisikan juga nilai *eigenfaces*-nya dan dibandingkan dengan *eigenfaces* dari image dalam database atau file *temporary*. Adapun algoritma selengkapnya adalah :

1. Buat *Make Flat Vektors(ImageList, N,M)*: *ImageList* adalah kumpulan dari *N training image*, di mana setiap image adalah $W \times H$ piksel, *M* adalah ukuran *vector flat* yang harus dibuat.
2. Gabungkan setiap *image* dalam *WH* elemen *vector* dengan menggabungkanseluruh baris. Buat *ImageMatrix* sebagai matriks $N \times WH$ berisi semua gambar yang digabung.

$$\begin{bmatrix} a & b & c \\ x & y & z \end{bmatrix} \rightarrow [a \ b \ c \ x \ y \ z] \quad (1)$$

3. Jumlahkan semua baris pada *ImageMatrix* dan bagi dengan *N* untuk mendapatkan rata-rata gambar gabungan. Kita namakan *vector* elemen *WH* ini dengan *R*.
4. Kurangi *ImageMatrix* dengan *average image R*. Kita namakan matriks baru ukuran $N \times WH$ sebagai *R'*.
5. Jika pada elemen-elemen dari matriks *R'* ditemukan nilai negative ganti nilainya dengan 0.

Kemudian identifikasi dilakukan dengan proyeksi menggunakan algoritma seperti berikut :

1. Buat *projectToFaceSpace (test_image)*: image berukuran $W \times H$ piksel.
2. Kita gabung elemen *vector WH* dan kita sebut *img*.
3. Load nilai rata-rata *R* dari database atau file.
4. Kurangi *img* dengan *R* sehingga kita dapatkan *img'*.
5. Jika pada *img'* ditemukan elemen dengan nilai *negative*, ganti dengan nilai 0 untuk mendapatkan *vector* ukuran *img'*.

Proses terakhir adalah identifikasi, yaitu memproyeksikan test image ke *face space* dan menghitung *score*.

1. Load semua wajah yang sudah diproyeksikan dari database.
2. *Proj=projectToFaceSpace(test_image)*.
3. Lakukan operasi pengurangan, *proj* dengan semua wajah yang telah diproyeksikan. Ambil nilai absolutnya dan jumlahkan, hasilnya adalah *score*.

Ambil *score* terkecil sebagai hasil dari wajah yang telah diproyeksikan. Wajah ini menjadi identifikasi (Fatta, 2009)

1.4 K-Nearest Neighbor

K-Nearest Neighbor adalah algoritma supervised learning dimana hasil instance yang baru diklasifikasikan berdasarkan mayoritas dari katogrodi K-tetangga terdekat. Metode Manhattan Distance merupakan salah satu dari metode K-Nearest Neighbor. Manhattan Distance digunakan untuk menentukan kesamaan antara dua buah objek. Hasil pengukuran didasarkan penjumlahan jarak selisih antara dua objek. Hasil dari Manhattan Distance bernilai mutlak (Sinwar,2014). Berikut adalah persamaannya:

$$d_{ij} = \sum_{k=1}^n |x_{ik} - x_{jk}| \quad (2)$$

2. PEMBAHASAN

2.1 Metodologi Penelitian

Langkah-langkah yang ditempuh dalam penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Alur Penelitian

Input Citra

Pada tahap ini citra hasil *capture* dari *webcam* dimasukan untuk dijadikan citra *training* atau pelatihan. Citra pelatihan diambil dua kali dari 10 orang mahasiswa.

Ekstraksi Ciri Menggunakan Eigenface PCA

Pada tahap ini merupakan mencari ciri dari citra yang digunakan. Data nilai ciri tersebut akan dijadikan data latih atau *training*. Data tersebut berupa nilai matrik citra.

Pencocokan Nilai Training dan Testing

Pada tahap ini adalah proses pencocokan antara data latih dan data tes. Proses pencocokan ini menggunakan metode Manhattan Distance.

2.2 Top Down Implementasi Sistem

Tahap implementasi pada deteksi wajah menggunakan metode *eigenface* PCA sesuai dengan alur implementasi *top-down*. Model *top-down* ini merupakan model yang sering digunakan pada pembuatan sistem karena pengerjaan aplikasi secara berurutan linier. Berikut ini urutan dalam pengerjaan program sebagai berikut :

- a. *Buka Webcam*
Webcame harus di aktifkan untuk menjalankan fungsi pendeteksi wajah atau untuk menampilkan gambar yang ditangkap *webcame* kedalam aplikasi.
- b. *Capture image*
Penangkapan citra wajah (*image capturing*) dapat dilakukan secara langsung (*real-time*) menggunakan *webcame*, setelah terdeteksi adanya gambar wajah pada tampilan *window* dari *webcame*.

- c. *Pemrosesan Awal*
Proses pemrosesan awal meliputi normalisasi ukuran citra, RGB ke *grayscale*, *histogram equalization* untuk memperbaiki kualitas citra input agar memudahkan proses pengenalan tanpa menghilangkan informasi utamanya.
- d. *Hitung nilai Eigenface*
Proses menghitung nilai *eigenface* yang menghasilkan nilai *eigenvalue* dan *eigenvector*.
- e. *Identifikasi*
Proses pengenalan wajah dengan menghitung jarak antara fitur wajah yang ada dalam data dan fitur wajah yang baru. Jarak yang didapat dicari yang terkecil untuk identifikasi

2.3 Debuging

Debugging merupakan tahap untuk meningkatkan kinerja sistem dengan mencari dan mengurangi kesalahan atau kerusakan di dalam sistem yang telah diimplementasikan. Dengan proses *debugging*, dapat diketahui apakah perangkat telah bekerja sesuai dengan kebutuhan. Pada proses pembuatan sistem deteksi wajah untuk presensi terdapat proses *debugging*, antara lain adalah *debugging* data dan *debugging coding*.

- a. *Debugging Data*
Debugging atau analisa data digunakan untuk mencari kesalahan pada data sample atau data training. Kesalahan yang mungkin terdapat pada data training adalah orientasi pengambilan data wajah yang tidak sesuai aturan seperti gelap terang ruangan atau posisi wajah yang setengah. Meskipun tidak mempengaruhi peforma dari sistem, namun ketidak konsistenan dalam hal orientasi maupun ukuran citra dapat menyebabkan kesalahan ekstraksi nilai.
- b. *Debugging Coding*
Debugging coding dilakukan untk mencari kesalahan pada *coding* sistem yang menyebabkan sistem tidak dapat berjalan atau sistem berjalan namun memberikan hasil yang tidak tepat. Proses *debuggingcoding* dilakukan seiring dengan proses *coding*. Cara melakukan *debugging coding* adalah dengan menjalankan setiap fungsi *coding* dari sistem. Jika fungsi tidak bisa dijalankan, berarti ada kesalahan *syntax coding* yang harus diperbaiki. Jika fungsi bisa dijalankan, maka tetap harus dilakukan pengecekan antara *output* dari fungsi hasil *coding* dengan data asli.

2.4 Perangkat Uji Coba

Perangkat uji coba yang digunakan meliputi perangkat keras dan perangkat lunak. Spesifikasi perangkat keras dan lunak dapat dilihat pada tabel 1.

Tabel 1. Spesifikasi Perangkat Uji Coba

Perangkat Keras	Perangkat Lunak
1. Processor : Intel Core i3-6100U CPU @ 2.30 GHz 2. Memory : Ram 6 GB 3. webcam : 0,3 MP 4. Harddiks : SSD 240GB	1. Sistem Operasi : Windows 10 pro 64-bit 2. IDE : Visual Studio 2013

2.5 Skenario Uji Coba

Uji coba dilakukan untuk mengetahui nilai akurasi, FAR (*False Acceptance Rate*) dan FRR (*False Rejection Rate*) dalam satuan persen (%). Untuk mengetahui nilai-nilai tersebut ada tiga kondisi pengujian yaitu jumlah data latih, kondisi cahaya, jarak antara webcam dengan wajah. Kondisi cahaya redup menggunakan dua lampu neon sedangkan cahaya terang menggunakan 4 lampu neon. Uji coba dilakukan di salah satu ruang perkuliahan UN PGRI Kediri Ada empat skenario berdasarkan jumlah data latih. Skenario dapat dilihat pada tabel 2, 3,4 dan 5.

Tabel 2. Skenario 1

data training	kondisi cahaya	jarak (cm)
3	Terang	50
3	Terang	100
3	Redup	50
3	Redup	100

Tabel 3. Skenario 2

data training	kondisi cahaya	jarak (cm)
5	Terang	50
5	Terang	100
5	Redup	50
5	Redup	100

Tabel 4. Skenario 3

data training	kondisi cahaya	jarak (cm)
8	Terang	50
8	Terang	100
8	Redup	50
8	Redup	100

Tabel 5. Skenario 4

data training	kondisi cahaya	jarak (cm)
10	Terang	50
10	Terang	100
10	Redup	50
10	Redup	100

2.6 Hasil Uji Coba

Dari empat skenario uji coba, setiap proses uji coba terdapat 5 mahasiswa sebagai data uji.

Hasil uji coba skenario 1 dengan menggunakan data latih 3 setiap citra mahasiswa dapat dilihat pada tabel 6.

Tabel 5. Hasil Uji Coba Skenario 1

kondisi cahaya	jarak (cm)	akurasi (%)	FAR (%)	FRR (%)
Terang	50	60	40	0
Terang	100	40	20	40
Redup	50	20	60	20
Redup	100	0	60	40

Dari tabel 5 dapat diketahui bahwa akurasi tertinggi adalah 60% pada kondisi cahaya terang dan jarak 50 cm. Akurasi terendah pada kondisi redup dengan jarak 100 cm yaitu 0%. Perbandingan akurasi dapat dilihat pada gambar 2.



Gambar 2. Grafik Akurasi Skenario 1

Pada gambar 2 akurasi skenario 1 turun drastis dari kondisi terang dengan jarak 50 cm sampai dengan jarak 100 cm dengan keadaan redup.



Gambar 3. Perbandingan FAR dan FRR Skenario 1

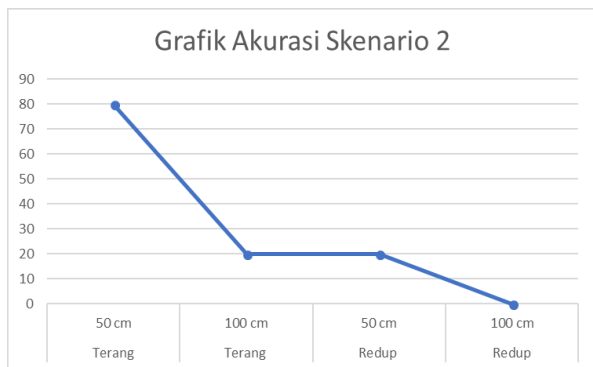
Perbandingan nilai FAR dengan FRR skenario 1 dapat dilihat pada gambar 3. Nilai FAR terendah yaitu 20% pada kondisi terang dengan jarak 100 cm. Nilai FRR 0% pada kondisi terang dengan jarak 100 cm.

Hasil Uji Coba Skenario 2 dapat dilihat pada tabel 6.

Tabel 6. Hasil Uji Coba Skenario 2

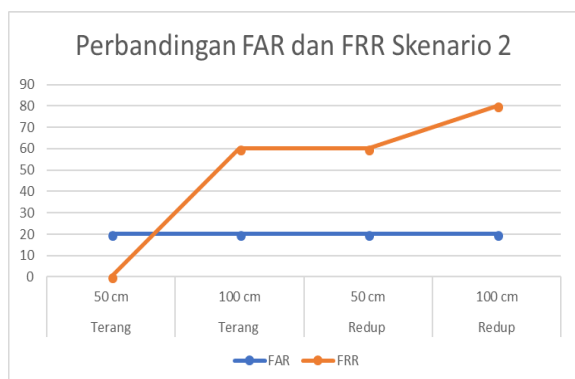
kondisi cahaya	jarak (cm)	akurasi (%)	FAR (%)	FRR (%)
Terang	50	80	20	0
Terang	100	20	20	60
Redup	50	20	20	60
Redup	100	0	20	80

Akurasi tertinggi pada skenario 2 adalah 80% pada kondisi cahaya terang dan jarak 50 cm. Akurasi terendah 0% pada kondisi redup dengan jarak 100 cm. Perbandingan akurasi dapat dilihat pada gambar 4.



Gambar 4. Grafik Akurasi Skenario 2

Pada skenario 2 akurasi maksimal lebih baik dari pada skenario 1. Tren penurunan akurasi dari kondisi jarak dan cahaya hampir sama dengan skenario 1. Namun ada dua akurasi yang sama 20% pada kondisi yang berbeda, yaitu jarak 100 cm cahaya terang dan jarak 50 cm cahaya redup.



Gambar 5. Perbandingan FAR dan FRR Skenario 2

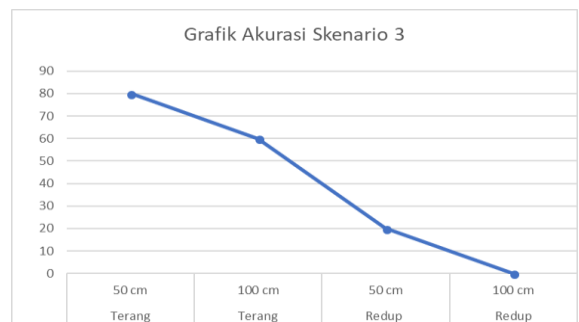
Nilai FAR pada semua kondisi skenario 2 bernilai sama yaitu 20%. Nilai FRR 0% pada kondisi terang dengan jarak 50 cm. perbandingan nilai FAR dengan FRR skenario 2 dapat dilihat pada gambar 5.

Hasil Uji Coba Skenario 3 dapat dilihat pada tabel 7

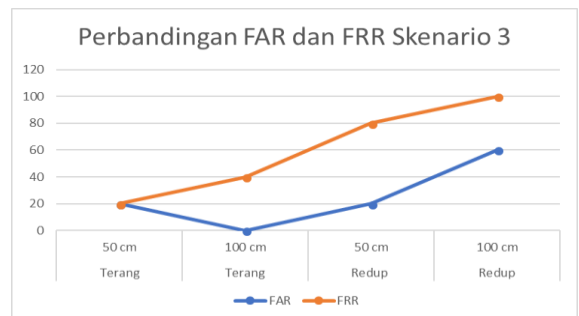
Tabel 7. Hasil Uji Coba Skenario 3

kondisi cahaya	jarak (cm)	akurasi (%)	FAR (%)	FRR (%)
Terang	50	80	20	0
Terang	100	60	0	40
Redup	50	20	20	60
Redup	100	0	60	40

Dari tabel 7 dapat diketahui bahwa akurasi tertinggi adalah 80% pada kondisi cahaya terang dan jarak 50 cm. Perbandingan akurasi dapat dilihat pada gambar 6. Dari gambar tren akurasi turun hampir sama dengan dua skenario sebelumnya yaitu skenario 1 dan 2. Pada kondisi jarak 100 cm keadaan redup mempunyai akurasi 0%, kondisi ini sama dengan skenario 1 dan 2. Dari hasil skenario 1 sampai 3 dapat diketahui bahwa pencahayaan berpengaruh sangat besar terhadap akurasi.



Gambar 6. Grafik Akurasi Skenario 3



Gambar 7. Perbandingan FAR dan FRR Skenario 3

Nilai FAR terendah yaitu 0% pada kondisi terang dengan jarak 100 cm. Nilai FRR 0% pada kondisi terang dengan jarak 50 cm. Perbandingan nilai FAR dengan FRR skenario 3 dapat dilihat pada gambar 7.

Hasil Uji Coba Skenario 3 dapat dilihat pada tabel 8

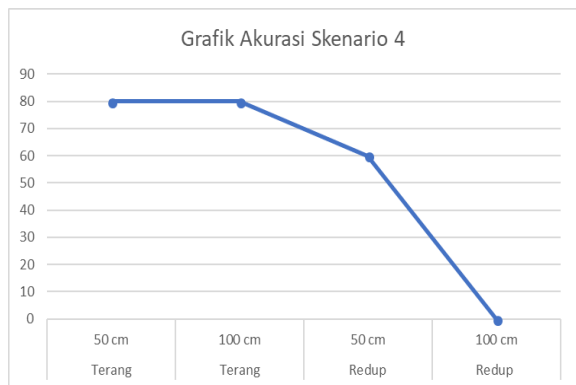
Tabel 8. Hasil Uji Coba Skenario 4

kondisi cahaya	jarak (cm)	akurasi (%)	FAR (%)	FRR (%)
Terang	50	80	20	0
Terang	100	80	20	0
Redup	50	60	20	20
Redup	100	0	60	40

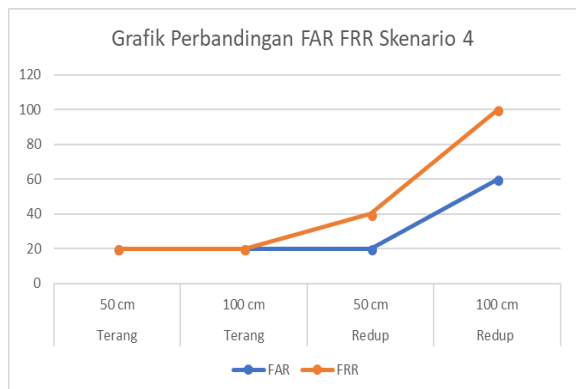
Dari tabel 8 dapat diketahui bahwa akurasi tertinggi adalah 80% pada kondisi cahaya terang dan

jarak 50 cm. Akurasi terendah pada kondisi redup dengan jarak 100 cm yaitu 0%. Perbandingan akurasi dapat dilihat pada gambar 8. Penurunan akurasi yang sangat drastis terjadi dari kondisi jarak 50 cm cahaya sedup ke kondisi jarak 100 cm cahaya redup yaitu dari 60% ke 0%. Dari skenario 1 sampai dengan 4 pada kondisi jarak 100 cm dengan cahaya redup akurasinya 0%. Sedangkan pada kondisi jarak 50 cm cahaya terang dari skenario 1 sampai 4 ada tren naik.

Nilai FAR terendah yaitu 20% terjadi pada tiga kondisi terang dengan jarak 50 dan 100 cm dan kondisi redup dengan jarak 50 cm. Nilai FRR 0% pada kondisi terang dengan jarak 50 dan 100 cm. Perbandingan nilai FAR dengan FRR skenario 4 dapat dilihat pada gambar 9.



Gambar 8. Grafik Akurasi Skenario 4



Gambar 9. Perbandingan FAR dan FRR Skenario 4

Rata-rata akurasi dari empat skenario uji coba dapat dilihat pada tabel 9.

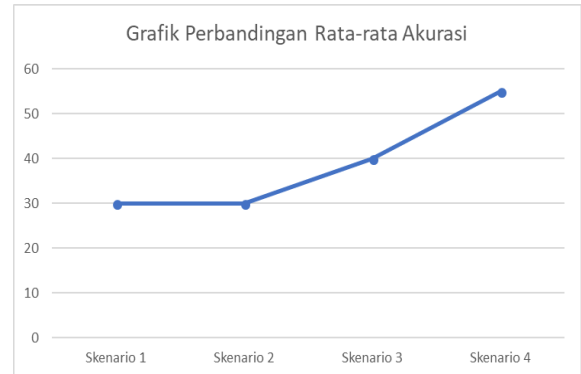
Tabel 9. Rata-rata Hasil Uji Coba

Skenario	Akurasi (%)	FAR (%)	FRR (%)
1	30	45	25
2	30	20	50
3	40	25	35
4	55	30	15

Dari tabel 9 dapat diketahui bahwa rata-rata akurasi tertinggi adalah 55% yaitu pada skenario 4.

Sedangkan rata-rata akurasi terendah adalah 30% pada skenario 1 dan 3.

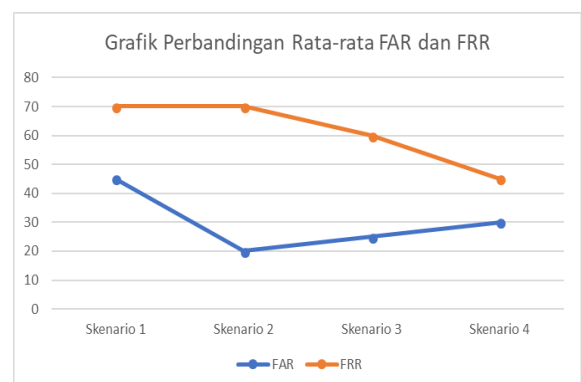
Untuk mempermudah analisa hasil skenario 1 sampai 4 dapat dilihat grafik pada gambar 10 dan 11. Perbandingan akurasi dapat dilihat pada gambar 10 sedangkan perbandingan FAR dan FRR pada gambar 11. Nilai FAR terendah yaitu 20% terjadi pada tiga kondisi terang dengan jarak 50 dan 100 cm dan kondisi redup dengan jarak 50 cm. Nilai FRR 0% pada kondisi terang dengan jarak 50 dan 100 cm.



Gambar 10. Grafik Rata-rata Akurasi Skenario

Dari gambar 10 dapat dilihat bahwa dari skenario 1 sampai dengan 4 tren rata-rata akurasi meningkat. Peningkatan akurasi ini pengaruh dari jumlah data latih yang digunakan. Semakin banyak data latih yang digunakan maka semakin baik hasil akurasinya. Hal itu terjadi karena dengan banyaknya data latih yang digunakan maka sistem mempunyai data ciri dari suatu data citra wajah semakin detail.

Dari hasil skenario 1 sampai 4 akurasi tertingginya terjadi pada kondisi jarak 50 cm dengan cahaya terang. Sehingga kondisi yang ideal untuk mencapai akurasi yang maksimal yaitu pada jarak 50 cm dengan cahaya terang. Hal ini disebabkan karena dengan jarak 50 cm dengan kondisi terang data citra wajah dapat ditangkap dengan kualitas yang baik sehingga data citra memiliki ciri wajah yang detail. Dengan adanya ciri wajah yang detail maka sistem dapat mengenali citra wajah dengan baik.



Gambar 11. Perbandingan Rata-rata FAR dan FRR

Dari gambar 11 tren nilai FAR dan FRR menurun, hal itu berarti penggunaan data latih yang banyak kualitas pengenalan wajah semakin baik.

Namun ada perbedaan tren nilai FAR dan FR, jika nilai FRR dari skenario 1 sampai 4 trennya terus turun, maka nilai FAR dari skenario 1 ke dua turun dan setelah itu dari skenario 2 sampai 4 naik. Dari hasil ini untuk mendapatkan nilai FAR yang optimal data latih yang ideal digunakan adalah 2.

3. KESIMPULAN

Dari analisa hasil uji coba yang telah dilakukan dapat disimpulkan sebagai berikut :

1. Akurasi pengenalan wajah dengan menggunakan metode ekstrasi fitur eigenface dan K-NN mencapai 80%.
2. Nilai rata-rata FAR terendah adalah 20% sedangkan FRR 15%
3. Semakin banyak data latih yang digunakan akurasi semakin tinggi.
4. Akurasi optimal didapat pada kondisi jarak 50 cm dengan cahaya terang.

PUSTAKA

- Adiyat, Iqbal. 2013. *Aplikasi Pengolahan Citra Digital Berbasis Flash pada Perangkat Mobile*. Pontianak: Universitas Tanjungpura.
- Ahmad, U. 2009. 10 langkah membuat program pengolahan citra menggunakan visual C#. Yogyakarta: Graha Ilmu.
- Efendy, Bobby. 2009. Autentifikasi Telapak Tangan pada Citra Digital Menggunakan Metode Support Vector Machine (SVM). Madura: Skripsi Universitas Trunojoyo
- Fatta, H. A. 2009. *Rekayasa Sistem Pengenalan Wajah*. Yogyakarta: ANDI.
- Fitriyah, N., dkk. 2015. Analisis dan Simulasi Sistem Pengenalan Wajah dengan Metode Fisherface Berbasis Outdoorvideo. Seminar Nasional Universitas PGRI Yogyakarta. 432-438.
- Fuad, N. 2017. Pemanfaatan K-Nearest Neighbor Pada Klasifikasi Image Berdasarkan Pola Fitur dan Tekstur. JUOTICLA, Vol. 2, No. 1.
- Kurniawati A. T. dan A. R. Dwi Rama. 2015. Aplikasi Pengenalan Wajah Menggunakan Metode Eigenface dengan Bahasa Pemrograman Java. Seminar Nasional Sains dan Teknologi Terapan III. 315-326.
- Pamungkas, D.P., Fajar, R.H. 2017. Implementasi Metode PCA dan Citu Block Distance Untuk Presensi Mahasiswa Berbasis Wajah. Malang: LPPM STIKI Malang
- Pratiwi, D. E. dan Agus Harjoko. 2013. Implementasi Pengenalan Wajah Menggunakan PCA (Principal Component Analysis). IJEIS. 3. 175-181.
- Putra, Darma, 2009. *Sistem Biometrika Konsep Dasar, Teknik Analisis Citra, dan Tahapan membangun Aplikasi Sistem Biometrika*. Yogyakarta: ANDI.
- Putra, R. S. 2013. Perancangan Aplikasi Absensi dengan Deteksi Wajah Menggunakan Metode Eigenface. Pelita Informatika Budi Darma. 4. 130-137.
- Salamun. 2016. Rancang Bangun Sistem Pengenalan Wajah Dengan Metode Principal Component Analysis. Jurnal Teknologi dan Sistem Informasi UNIVRAB VOL. 1 No. 2, Juli 2016, 1.
- Sinwar, D., Kaushik, R. 2014. Study of Euclidean And Manhattan Distance matrices using Simple K-Means Clustering. International Journal For Research in Applied Science And Engineering Technology (IJRASET). Vol. 2, No. 5.
- Wang, M. Y., Hideaki. T. 2017. Individual Recognition and the Face Inversion Effect in Medaka Fish (*Oryzias Latipes*). eLife Research Article. (<https://search.proquest.com/444260dd-05ba-4edf-a5b7-abcbd0129fd2>, diakses 8 Juni 2018).
- Wulanningrum, R. 2018. Penggunaan Algoritma K-Nearest Neighbor untuk Identifikasi Citra Kamboja. Generation Journal, Vol. 2, No.2. 2008. Kamus Besar Bahasa Indonesia Edisi Keempat. Jakarta: Gramedia Pustaka Utama