

# IDENTIFIKASI CITRA GARIS TELAPAK TANGAN MENGGUNAKAN METODE *LINEAR DISCRIMINANT ANALYSIS* DENGAN PROBABILITAS *NAÏVE BAYESIAN*

Bayu Pradinta<sup>1</sup>, Ernawati<sup>2</sup>, Endina Putri Purwandari<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu.  
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA  
(tel: 0736-341022; fax: 0736-341022)

<sup>1</sup>bayupradinta@gmail.com,

<sup>2</sup>ernawati@unib.ac.id,

<sup>3</sup>endinaputri@unib.ac.id

**Abstrak** : Pencocokan citra merupakan suatu cara untuk mengidentifikasi dan mengukur kesamaan citra. Penelitian ini menggunakan teknologi biometrik yaitu sistem yang melakukan pencocokan citra berdasarkan bagian tubuh manusia dengan cara mencocokkan antara citra uji yang diterima dengan citra latih. Citra yang digunakan adalah telapak tangan. Penggunaan telapak tangan dikarenakan telapak tangan memiliki karakteristik yang unik, sulit dipalsukan dan cenderung stabil. Penelitian ini membangun sebuah aplikasi identifikasi citra garis telapak tangan berbasis tekstur dengan metode *linear discriminant analysis* untuk ekstraksi ciri dan metode *naïve bayesian* untuk menghitung peluang kemiripan citra uji dengan citra latih. Citra yang digunakan sebagai objek penelitian adalah citra telapak tangan manusia bagian kanan dari *CASIA Palmprint Image Database* dan yang diambil langsung dari orang sekitar. Aplikasi ini dibangun dalam bahasa pemrograman Matlab R2016a dan dirancang dengan *Unified Model Language* (UML). Metode pengembangan sistem yang digunakan adalah *prototyping*. Hasil yang diperoleh pada penelitian ini adalah nilai akurasi sebesar 95% untuk pengujian citra uji terhadap citra latih, 93,4% untuk citra uji dengan *noise salt and pepper* 0.005 dan 93,4% untuk pengujian citra uji dengan *Gaussian filter* 2.5 .

**Kata Kunci** : Pencocokan Citra, Citra Garis Telapak tangan, *Linear Discriminant Analysis*, *Naïve Bayesian*.

**Abstract**: *Image matching is a way to identify and measure the similarity image. This research uses biometric technology that system performs image matching based on parts of the human body by matching between the test images received by the trained image. The image use is palm. The use of palm because of the palm has unique characteristics, hard to forge and tend to be stable. This study build an image identification applications palmprint with texture-based linear discriminant analysis method for feature extraction and naïve Bayesian method to calculate the probability similarity of test images with train images. Image that used as an object of research is the image of human hands right part from CASIA Palmprint Image Database and taken directly from the people around. This application is built in programming language Matlab R2016a and is designed with the Unified Model Language (UML). System development*

*method used is prototyping. The results obtained in this research is the accuracy value of 95% for testing the test images to the image of training, 93,4% for the test images with salt and pepper noise 0.005 and 93,4% for testing the test images with the Gaussian filter 2.5.*

**Keywords**: *Image Matching, Palmprint Image, Linear Discriminat Analysis, Naïve Bayesian.*

## I. PENDAHULUAN

Kemampuan sistem pengenalan diri yang mampu mengenali target secara tepat dan akurat dapat difungsikan untuk meningkatkan keamanan sistem. Beberapa metode yang masih banyak diterapkan untuk mengidentifikasi seseorang

adalah metode konvensional, yaitu sesuatu yang diingat oleh user seperti *Personal Identification Number* (PIN) dan *password*. Namun metode ini tidak lagi bisa diandalkan karena PIN dan *password* bisa dilupakan oleh user maupun diketahui oleh orang lain. Sistem biometrika diharapkan mampu meminimalisasi kelemahan-kelemahan yang ada pada sistem identifikasi konvensional. Beberapa sistem biometrika telah dikembangkan dalam beberapa tahun. Namun, beberapa dari sistem tersebut memiliki kekurangan masing-masing.

Telapak tangan memiliki karakteristik yang unik, diantaranya ciri-ciri geometri seperti: panjang, lebar, dan area telapak tangan, garis-garis prinsip seperti garis hati, garis kepala, dan garis kehidupan, garis-garis kust/lemah, titik delta, dan ciri-ciri minusi [1]. Garis-garis prinsip dan kusut memiliki beberapa kelebihan dibandingkan dengan ciri-ciri yang dihasilkan biometrik lainnya seperti dapat diperoleh dari citra resolusi rendah, sulit dipalsukan, dan ciri-ciri garis telapak tangan bersifat stabil karena sedikit mengalami perubahan dalam kurun waktu lama. Dengan adanya karakteristik yang unik ini maka telapak tangan dapat digunakan sebagai alat verifikasi identitas seseorang dengan mencocokkan data yang terdapat dalam database dengan data yang dimasukkan.

Metode untuk ekstraksi fitur yang paling populer adalah *Principal Component Analysis* (PCA). Namun demikian PCA memiliki kelemahan yaitu pemisah antar kelas yang kurang optimal [2].

*Linear Discriminant Analysis* (LDA) merupakan metode yang digunakan untuk mengatasi kekurangan PCA. Pengurangan dimensi pada LDA dilakukan dengan cara

memaksimalkan nilai *Between-Class Scatter* dan meminimalkan *Within-Class Scatter*. Filter *Gaussian* adalah salah satu filter linear dengan nilai pembobotan untuk setiap anggotanya dipilih berdasarkan bentuk fungsi *Gaussian*. Filter ini sangat baik untuk menghilangkan *noise* yang bersifat sebaran normal [3].

Metode pengukuran berbasis peluang menggunakan *Naive Bayesian* termasuk metode yang sederhana. Beberapa kelebihan dari teori *Naive Bayesian* adalah mudah untuk dipahami, hanya memerlukan pengkodean yang sederhana dan lebih cepat dalam penghitungan [4]. Proses segmentasi citra dilakukan dengan memanfaatkan metode *Region of Interest* (ROI). *Region of Interest* (ROI) memungkinkan pengguna untuk mengakses bagian tertentu dari sebuah citra digital untuk diolah secara berbeda [5].

Untuk itu maka akan dilakukan penelitian deteksi telapak tangan yang memfokuskan pada masalah akurasi dan kecepatan aplikasi melakukan pencocokan citra telapak tangan menggunakan metode *Linear Discriminant Analysis* (LDA) dengan probabilitas *Naive Bayesian*.

## II. LANDASAN TEORI

### A. *Linear Discriminant Analysis* (LDA)

Metode untuk ekstraksi fitur yang paling populer adalah *Principal Component Analysis* (PCA). Namun demikian PCA memiliki kelemahan yaitu pemisah antar kelas yang kurang optimal. *Linear Discriminant Analysis* (LDA) merupakan metode yang digunakan untuk mengatasi kekurangan PCA. Pengurangan dimensi pada LDA dilakukan dengan cara memaksimalkan nilai *Between-Class Scatter* dan meminimalkan

*Within-Class Scatter*. PCA dan LDA mempunyai perbedaan yang sangat jelas, pengklasifikasian terhadap ciri dapat dilakukan oleh PCA, pengklasifikasian terhadap data dapat dilakukan dengan LDA. Pada ekstraksi fitur dengan LDA, data set lokasinya tetap, namun kelas yang dibentuk menjadi lebih terpisah, kondisi ini disebabkan oleh jarak antar data pelatihan dalam satu kelas menjadi lebih kecil [3].

LDA bekerja berdasarkan analisa matrik penyebaran (*scatter matrix analysis*) yang bertujuan menemukan suatu proyeksi optimal yang dapat memproyeksikan data input pada ruang dengan dimensi yang lebih kecil dimana semua pola dapat dipisahkan semaksimal mungkin. Karenanya untuk tujuan pemisahan tersebut maka LDA akan mencoba untuk memaksimalkan penyebaran data-data input diantara kelas-kelas yang berbeda sekaligus juga meminimalkan penyebaran input pada kelas yang sama. Perbedaan antar kelas direpresentasikan oleh matriks  $S_B$  (*scatter between class*) dan perbedaan dalam kelas direpresentasikan oleh matriks  $S_W$  (*scatter within class*).

*Matrix Scatter* dalam kelas  $S_W$ , dan *matrix scatter* antar kelas  $S_B$  didefinisikan masing-masing sebagai berikut:

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \quad (2.1)$$

$$S_B = \sum_{i=1}^c N_i (x_k - \mu_i)(x_k - \mu_i)^T \quad (2.2)$$

Keterangan :

$S_W$ : *matrix within scatter*

$c$  : jumlah kelas

$x_k$  : gambar ke- $k$

$N$  : jumlah gambar pada kelas ke- $i$

$\mu$  : rata-rata total dari keseluruhan gambar

$\mu_i$  : rata-rata dari kelas ke- $i$

Kriteria ini menghasilkan solusi dengan persamaan berikut:

$$S_B = S_W \Psi \Lambda \quad (2.3)$$

Keterangan :

$S_B$ : *matrix between scatter*

$S_W$ : *matrix within scatter*

$\Psi$ : *matrix eigenvector*

$\Lambda$ : adalah *matrix* diagonal nilai *eigen*

Dengan kata lain akan dicari *eigenvector* dan *eigenvalue* dari matix  $C$  yang merupakan kombinasi *within* dan *between scatter matrix* seperti pada persamaan 2.3. Kembali dilakukan pemilihan sebanyak  $l$  kolom *eigenvector* dari  $\Psi$  yang berasosiasi dengan nilai-nilai *eigen* terbesar. Pemilihan  $l$  kolom *eigenvector* ini menghasilkan *matrix* proyeksi  $\Psi_l$  yang selanjutnya digunakan untuk ekstraksi *feature* seperti halnya pada PCA.

### B. Naïve Bayesian

Model pengukuran kemiripan objek berbasis peluang dengan mempertimbangkan peluang parameter yang melekat pada suatu objek seperti ciri citra latih dan ciri citra uji. Salah satu metode pengukuran kemiripan berbasis peluang yang sering digunakan adalah *Naive Bayesian*. Metode pengukuran kemiripan menggunakan *Naive Bayesian* termasuk metode yang sederhana, namun metode itu telah banyak digunakan untuk melakukan klasifikasi kedokteran, biometrika, klasifikasi teks dan masih banyak lagi [6]. *Naive Bayesian* menggunakan distribusi *Gaussian* untuk menangani atribut kontinu. Distribusi *Gaussian* ini mempertimbangkan dua parameter penting yaitu, rata-rata  $\mu$  dan varian  $\sigma$ . *Gaussian* yang digunakan untuk pengukuran kemiripan *Naive Bayesian* adalah menggunakan Persamaan [2].

$$P(X_i=x_i | Y=y_i) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} \exp\left[-\frac{1}{2} \left(\frac{x_i - \mu_{ij}}{\sigma_{ij}}\right)^2\right] \quad (2.4)$$

Keterangan :

$P$  : probabilitas dari atribut  $x_i$

$i$  : indek untuk nilai dari atribut  
 $j$  : indek kelas  
 $Y$  : merepresentasikan kelas yang dicari  
 $x$  : data uji  
 $\mu$  : nilai rata-rata ciri citra  
 $\sigma$  : nilai standar deviasi dari ciri citra

Dengan  $P$  merupakan nilai probabilitas dari citra latih terhadap citra uji. Langkah pertama yaitu mencari nilai hasil bagi 1 dengan  $\sqrt{2\pi}$  dikali standar deviasi dari citra latih kemudian dikali dengan nilai eksponensial dari perkalian antara  $-\frac{1}{2}$  dengan kuadrat dari hasil bagi dari citra uji dikurang ciri dari citra latih dibagi standar deviasi dari citra latih .

Misalkan diketahui 6 buah citra yang telah dikelompokkan menjadi 3 kelas, masing-masing mempunyai 2 data dari 3 ciri yang unik sebagaimana terlihat pada Tabel 2.1.

Tabel 2. 1 Data Ciri 3 Kelas Citra

Citra	Kelas	Ciri 1	Ciri 2	Ciri 3
1	A	20	60	30
2		19	55	35
3	B	10	70	32
4		12	72	28
5	C	10	60	40
6		11	40	43

Jika ada satu citra yang akan diuji dengan ciri ke-1 = 14, ciri ke-2 = 60 dan ciri ke-3 = 38, dengan menggunakan metode *Naïve Bayesian* tentukan masuk kelas mana citra tersebut. Untuk melakukan klasifikasi citra menggunakan *Naïve Bayesian*, maka perlu dihitung rata-rata dan standar deviasi tiap kelas untuk masing-masing ciri. Rata-rata dan standar deviasi dari 3 kelas citra dengan 3 ciri dapat dilihat pada Tabel 2.2 dan Tabel 2.3.

Tabel 2. 2 Rata-rata Ciri Citra 3 Kelas

Kelas	Rata-rata		
	Ciri 1	Ciri 2	Ciri 3
A	19.5	57.5	32.5
B	11	71	30
C	10.5	50	41.5

Tabel 2. 3 Standar Deviasi Ciri Citra 3 Kelas

Kelas	Standar Deviasi		
	Ciri 1	Ciri 2	Ciri 3
A	0.70711	3.5355	3.5355
B	1.4142	1.4142	2.8284
C	0.70711	14.142	2.1213

Peluang kelas A terhadap semua kelas adalah jumlah data kelas a dibagi dengan jumlah data semua kelas, dalam hal ini adalah  $2/6 = 0.33$ . Begitu juga dengan peluang kelas B terhadap semua kelas masing-masing juga 0.33, sehingga

$$P(\text{Kelas} = A) = \frac{2}{6} = 0.33$$

$$P(\text{Kelas} = B) = \frac{2}{6} = 0.33$$

$$P(\text{Kelas} = C) = \frac{2}{6} = 0.33$$

Data rata-rata dan standar deviasi sebagaimana terlihat pada Tabel 2.2 dan 2.3 digunakan untuk menghitung peluang kemiripan tiap-tiap ciri terhadap citra kelas. Berdasarkan persamaan (2.4), maka nilai peluang tiap ciri-ciri data uji terhadap masing-masing karakteristik kelas A dapat dihitung sebagaimana berikut ini

$$P(\text{Kelas A/Ciri ke-1}=14) = \frac{1}{\sqrt{2*3.14*0.70711}} *$$

$$e^{-0.5\left(\frac{14-19.5}{0.70711}\right)^2} = 4.11270063676419E-14$$

$$P(\text{Kelas A/Ciri ke-2}=60) = \frac{1}{\sqrt{2*3.14*3.5355}} *$$

$$e^{-0.5\left(\frac{60-57.5}{3.5355}\right)^2} = 0.0879005416323996$$

$$P(\text{Kelas A/Ciri ke-3}=38) = \frac{1}{\sqrt{2*3.14*3.5355}} *$$

$$e^{-0.5\left(\frac{38-32.5}{3.5355}\right)^2} = 0.033656492064405$$

Peluang kemiripan data citra uji terhadap citra kelas A dapat dihitung sebagaimana berikut ini

$$\begin{aligned}
 P(\text{Kelas=A/Data uji}) &= P(\text{Kelas=A/Ciri} \\
 &=14)*P(\text{Kelas = A/Ciri 2=60})*P(\text{Kelas=A/Ciri} \\
 &3=38)*P(\text{Kelas=A}) = \\
 &4.11270063676419E14*0.0879005416323996*0.0 \\
 &33656492064405*0.33= 4.05570392768E-17
 \end{aligned}$$

Dengan cara yang sama, maka nilai peluang tiap ciri-ciri data uji terhadap masing-masing karakteristik pada kelas B dapat dihitung sebagaimana berikut ini

$$\begin{aligned}
 P(\text{Kelas B/Ciri ke-1=14}) &= \frac{1}{\sqrt{2*3.14*1.4142}} * \\
 &e^{-0.5\left(\frac{14-11}{1.4142}\right)^2} \\
 &= 0.029740117461812
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Kelas B/Ciri ke-2=60}) &= \frac{1}{\sqrt{2*3.14*1.412}} * \\
 &e^{-0.5\left(\frac{60-71}{1.4142}\right)^2} \\
 &= 2.05635031838209E-14
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Kelas B/Ciri ke-2=38}) &= \frac{1}{\sqrt{2*3.14*2.8284}} * \\
 &e^{-0.5\left(\frac{38-30}{2.8284}\right)^2} \\
 &= 0.00258028243954
 \end{aligned}$$

Peluang kemiripan data citra uji terhadap citra kelas B dapat dihitung sebagaimana berikut ini

$$\begin{aligned}
 P(\text{Kelas=B/Data uji}) &= P(\text{Kelas=A/Ciri} \\
 &1=14)*P(\text{Kelas = A/Ciri 2=60})*P(\text{Kelas=A/Ciri} \\
 &3=38)*P(\text{Kelas=B}) = \\
 &0.029740117461812*2.05635031838209E- \\
 &14*2.0563 \\
 &5031838209E-14 = 5.26763532066598E-19
 \end{aligned}$$

Dengan cara yang sama dihitung peluang tiap ciri-ciri data uji terhadap masing-masing karakteristik pada kelas C. Peluang kemiripan data uji terhadap kelas C adalah

$$P(\text{Kelas = C/ Data Uji}) = 9.53957340552974E-10$$

Untuk membuat keputusan akhir data uji masuk dalam kelas A, B atau C. Data peluang citra uji terhadap kelas A, B dan C dibandingkan, yang mempunyai peluang paling besar adalah kelas yang paling mirip. Dalam contoh ini yang

mempunyai peluang maksimal adalah kelas C, nilai peluangnya adalah 9.53957340552974E-10.

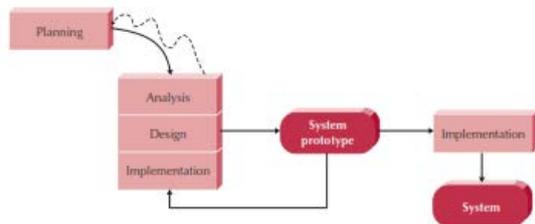
### III. METODE PENELITIAN

#### A. Dataset Penelitian

Data set penelitian dengan sampel 180 citra telapak tangan manusia yang diambil dari 30 objek terdiri dari 120 sampel yang diambil secara acak dari lingkungan sekitar dan 60 diambil dari *CASIA Palmprint Image Database*. Dari setiap objek diambil 6 sampel yang dipengaruhi oleh pencahayaan serta sudut pengambilan gambar yang berbeda.

#### B. Metode Pengembangan Sistem

Metode pengembangan sistem yang akan digunakan dalam penelitian ini adalah metode *prototyping*. *Prototyping* mengerjakan fase analisis, perancangan, dan implementasi secara bersamaan, dan ketiga fase tersebut dikerjakan berulang hingga sistem lengkap. Dengan metodologi ini, dasar dari analisis dan perancangan dikerjakan, dan segera mulai mengerjakan sistem prototipe, suatu program “*quick-and-dirty*” yang menyediakan sejumlah fitur minimum [7]. Metodologi *prototyping* ditunjukkan gambar 3.1.



Gambar 3. 1 Metodologi berbasis *prototyping*

#### C. Metode Pengujian Sistem

##### 1. White Box Testing

Dalam pengujian *white box*, penulis akan meneliti kode-kode program yang ada dan akan

menganalisis apakah ada kesalahan atau tidak. Pada penelitian ini akan menggunakan pengujian *white box* jenis *basis path testing*. *Basis path testing* adalah salah satu teknik pengujian yang didasarkan pada aliran logika yang diambil dari program atau sistem.

## 2. **Black Box Testing**

*Black box testing* adalah sebuah pengujian kondisi yang dibangun berdasarkan fungsional dari sistem. Pada penelitian ini akan menggunakan pengujian *black box* jenis *Equivalence Partitioning*, yaitu metode yang membagi domain masukan dari suatu program ke dalam kelas-kelas data berdasarkan pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas menurut spesifikasi dari komponen tersebut, yang akan diperlukan sama oleh komponen tersebut.

## D. Metode Uji Kelayakan Sistem

Unjuk kerja suatu sistem biometrika dinyatakan dengan *Decision Error Rate* (rasio kesalahan), yaitu *False Matching Rate/FMR* (Rasio Kesalahan Pencocokan) dan *False Non Matching Rate/FNMR* (Rasio Kesalahan Ketidakcocokan) [8].

Kesalahan dihitung dengan menggunakan persamaan [12], berikut :

$$\text{FMR/Rasio Kesalahan Pencocokan} = \frac{\text{(Jumlah Kesalahan)}}{\text{(Jumlah Keseluruhan Proses)}} \times 100\% \quad (3.1)$$

Selain itu untuk menyatakan tingkat kesuksesan pengenalan suatu sistem biometrika (bukan tingkat kesalahan) disebut sebagai *Genuine Acceptance Rate* (GAR) dan dihitung menggunakan persamaan berikut :

$$\text{GAR} = 1 - \text{FMR} \text{ atau } \text{GAR} = 1 - \text{FNMR} \quad (3.2)$$

Kinerja biometrika ditentukan berdasarkan kedua parameter tersebut. Jika tingkat

kesalahannya tinggi, maka sistem biometrika harus ditinjau apakah memang ada kesalahan algoritma atau kesalahan pembacaan sensor atau memang tipe biometrika itu sendiri tidak bisa memenuhi kekhasan [9]. Umumnya semakin besar ukuran database atau jumlah pengguna dalam database, maka bagian tumpang tindih akan semakin besar yang artinya FMR/FNMR meningkat dan demikian pula sebaliknya [9].

## IV. ANALISIS DAN PERANCANGAN

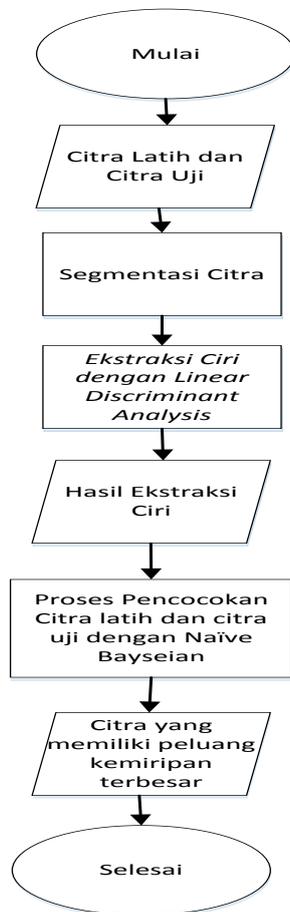
### A. Identifikasi Masalah

Suatu aplikasi identifikasi citra garis telapak tangan biasa menggunakan metode seperti *Principal Component Analysis* (PCA), Jaringan Syaraf Tiruan, Transformasi *Wavelet*, *Hidden Markov Model* (HMM), dan *Linear Discriminant Analysis* (LDA). Namun untuk meningkatkan akurasi pencocokan garis telapak tangan, metode tersebut bisa digabungkan dengan metode-metode lain. Metode lain yang dapat ditambahkan tersebut misalnya saja metode untuk melakukan segmentasi terhadap citra yang akan dideteksi dan metode untuk meningkatkan kualitas dari citra tersebut. Penelitian ini menggunakan metode *Region Of Interest* (ROI) untuk melakukan segmentasi dan metode *Gaussian Filter* untuk meningkatkan kualitas citra. Dengan adanya tambahan metode-metode tersebut diharapkan dapat meningkatkan tingkat akurasi pada saat pencocokan citra.

### B. Analisis Alur Kerja Sistem

Alur sistem merupakan hasil analisis perancangan tahapan kerja sistem yang akan dibangun. Alur ini dimulai dari user memasukkan (*input*) data sampai menghasilkan keluaran (*output*). Dalam sistem ini, input-nya berupa citra telapak tangan manusia dan informasi biodata citra

tersebut. Sedangkan untuk *output* adalah citra hasil pencocokan telapak tangan yang memiliki peluang terbesar.



Gambar 4. 1 Diagram Alir Sistem

### C. Analisis Fungsional

Analisis fungsional berarti melakukan analisis dari fungsi-fungsi pada sistem yang akan dibangun. Fungsi-fungsi pada sistem yang dimaksud adalah fitur-fitur yang terdapat didalam sistem. Berikut adalah beberapa fitur yang terdapat pada sistem :

- Sistem yang dibangun mampu melakukan segmentasi dengan metode *Region Of Interest (ROI)* dari citra yang diinputkan oleh *user*.
- Sistem yang dibangun dapat melakukan pencocokan citra telapak tangan dengan menggunakan metode ekstraksi *Linear Discriminant Analysis* dan menggunakan

metode *Naïve Bayesian* untuk mengukur peluang kemiripan antara citra uji dengan citra latih.

- Sistem yang dibangun dapat menampilkan *output* berupa identitas citra. Identitas citra tersebut sebelumnya telah disimpan kedalam *database* oleh *user*.

## V. PEMBAHASAN

### A. Implementasi Interface

#### 1. Menu Utama

Halaman utama pada aplikasi ini muncul pertama kali saat aplikasi ini dijalankan. Tampilan dari menu utama aplikasi yang dapat dilihat pada Gambar 5.1.

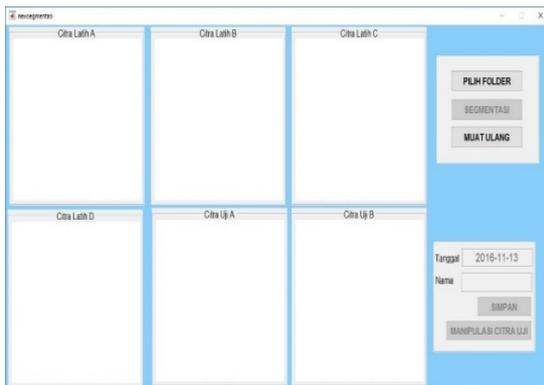


Gambar 5. 1 Menu Utama

Dapat dilihat pada Gambar 5.1 merupakan tampilan menu utama. Pada menu utama aplikasi menggunakan beberapa tombol yaitu tombol Segmentasi, tombol *Database*, tombol Pencocokan Citra, tombol Bantuan dan tombol Keluar. Latar belakang gambar menu utama menggunakan gambar *background.jpg*.

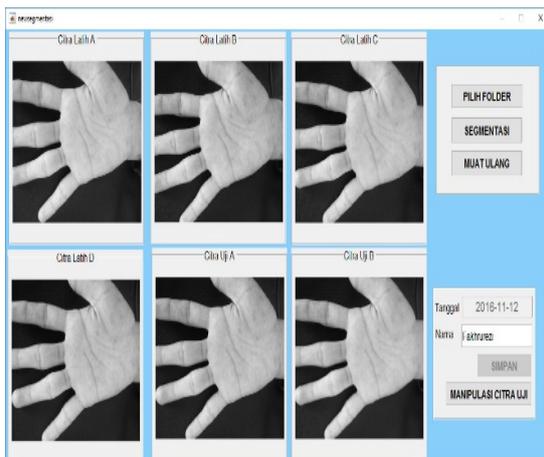
#### 2. Menu Segmentasi

Menu segmentasi akan muncul setelah memilih tombol Segmentasi pada menu utama. Tampilan dari Menu Segmentasi yang dapat dilihat pada Gambar 5.2.



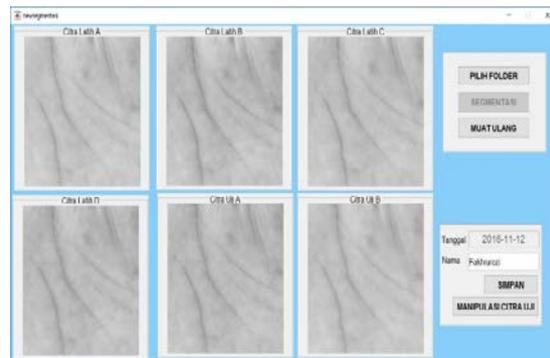
Gambar 5. 2 Menu Segmentasi

Pada menu Segmentasi terdapat tombol Pilih Folder yang digunakan untuk memilih folder citra telapak tangan yang akan disegmentasi, setelah folder citra telah terpilih maka citra yang ada pada folder tersebut akan ditampilkan pada 6 axes yang tersedia pada interface, seperti pada gambar berikut ini.



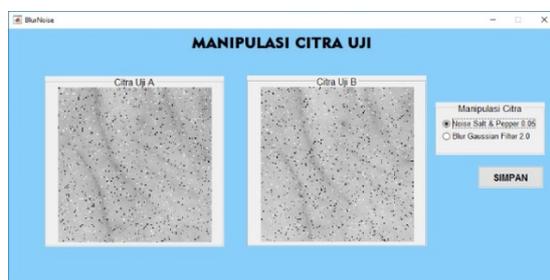
Gambar 5. 3 Tampilan interface setelah memilih folder citra

Setelah memilih folder citra, tekan tombol segmentasi untuk melakukan segmentasi citra. Hasil dari segmentasi citra akan menggantikan posisi citra telapak tangan sebelum segmentasi, seperti pada Gambar 5.4 berikut ini.



Gambar 5. 4 Hasil Segmentasi

Pada Gambar 5.4 terlihat bahwa keenam citra telapak tangan telah tersegmentasi secara serentak, setelah proses segmentasi selesai maka proses selanjutnya adalah menyimpan citra hasil segmentasi. Tulis nama citra pada kolom nama, kemudian tekan tombol Simpan. Citra dengan nama Citra Latih A, Citra Latih B, Citra Latih C, dan Citra Latih D akan tersimpan pada database sesuai dengan nama yang telah diinputkan dan tersimpan pada folder citra latihan dengan nama file sesuai dengan nomor urut citra terakhir yang ada di folder citra latihan sedangkan citra dengan nama Citra Uji A dan Citra uji B akan tersimpan di folder Citra Uji dengan nama file yang telah diinputkan pada kolom nama sebelumnya. Tombol manipulasi citra berfungsi untuk memanipulasi citra dengan menambahkan noise ataupun blur pada citra uji seperti Gambar 5.5 dan Gambar 5.6 .



Gambar 5. 5 Citra Uji dengan Noise Salt and Pepper

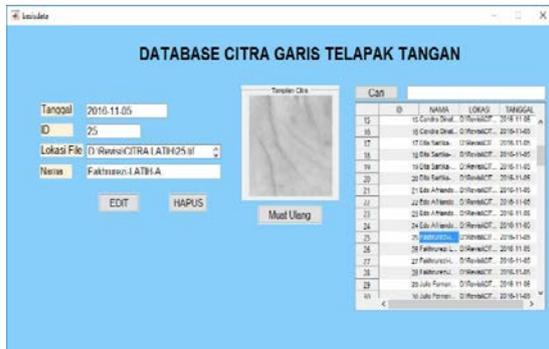


Gambar 5. 6 Citra Uji dengan *Blur Gaussian Filter*

Citra Uji A dan Citra Uji B yang ada pada menu segmentasi akan ditransfer ke menu manipulasi citra uji. Pada menu ini *user* dapat memilih jenis manipulasi yang diinginkan *pada radio button*. Setelah itu *user* dapat menyimpan hasil manipulasi citra dengan menekan tombol simpan.

### 3. Menu Database

Menu *database* akan tampil setelah menekan tombol *Database* pada menu utama. Berikut adalah tampilan dari Menu *database* yang dapat dilihat pada Gambar 5.7.

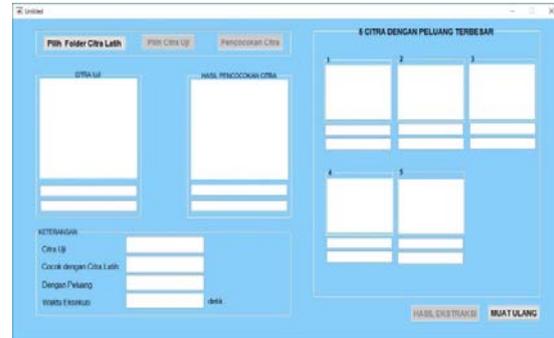


Gambar 5. 7 Menu Database

Pada Gambar 5.7 terlihat daftar citra latih yang tersimpan di *database* pada tabel. Ketika memilih salah satu dari citra yang ada pada tabel maka data citra yang terpilih akan muncul. Tombol *edit* berfungsi untuk mengubah data nama citra, sedangkan tombol *hapus* berfungsi untuk menghapus data citra.

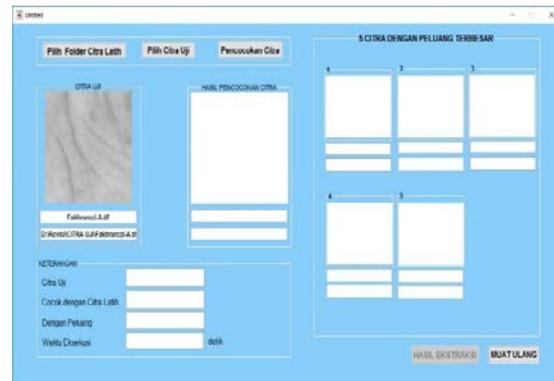
### 4. Menu Pencocokan Citra

Menu pencocokan citra akan tampil setelah menekan tombol pencocokan citra pada menu utama. Tampilan dari Menu pencocokan citra yang dapat dilihat pada Gambar 5.8.



Gambar 5. 8 Menu Pencocokan Citra

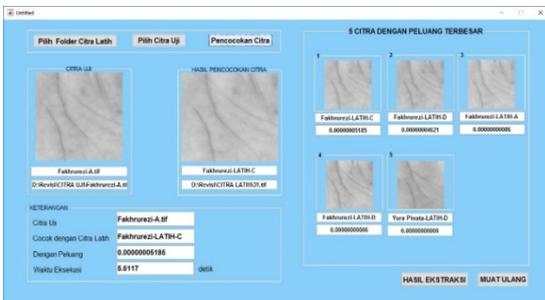
Pada menu pencocokan citra terdapat tombol *Pilih Folder Citra Latih* yang digunakan untuk memilih *folder* citra latih, setelah itu tekan tombol *Pilih Citra Uji* untuk memilih citra yang akan diujikan dengan citra yang ada di *folder* citra latih, ketika citra uji terpilih, maka citra yang terpilih akan tampil pada *axes* citra uji pada *interface* seperti Gambar 5.9 berikut.



Gambar 5. 9 Pilih Citra Uji

Pada Gambar 5.9 terlihat citra uji tampil pada *interface* beserta dengan nama *file* dan lokasi *file* citra uji. Setelah memilih citra uji, tekan tombol *Pencocokan Citra* untuk melakukan proses pencocokan citra uji dengan citra latih yang ada di *folder* citra latih. Jika proses pencocokan citra

telah selesai maka citra latih yang cocok dengan citra uji akan tampil pada *interface* beserta dengan nama *file* dalam *database* dan lokasi *file*-nya. Pada kolom keterangan ditampilkan nilai peluang kemiripan yang diperoleh dari citra latih yang cocok dan lama waktu yang dibutuhkan untuk melakukan proses pencocokan citra. Pada bagian kanan *interface* juga ditampilkan 5 citra latih yang memiliki peluang terbesar cocok dengan citra uji beserta dengan nama *file*-nya dalam *database* dan nilai peluang masing-masing citra tersebut, seperti Gambar 5.10 berikut.



Gambar 5. 10 Hasil Pencocokan Citra

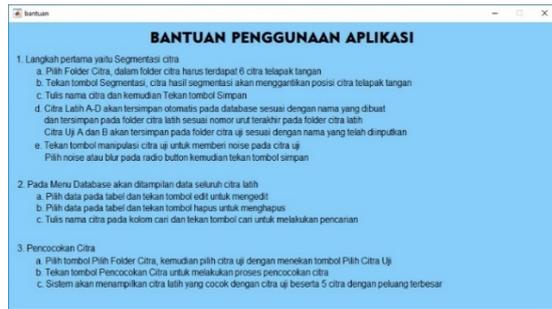
Untuk melihat hasil ekstraksi citra dapat dilakukan dengan menekan tombol hasil ekstraksi. Sistem akan menampilkan hasil ekstraksi semua citra latih, citra uji, dan citra latih yang cocok dengan citra uji seperti pada Gambar 5.11:



Gambar 5. 11 Hasil Ekstraksi

## 5. Menu Bantuan

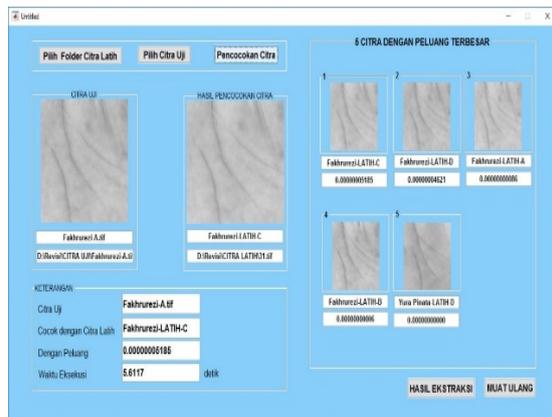
Menu bantuan akan tampil setelah menekan tombol Bantuan pada menu utama. Berikut adalah tampilan dari Menu Bantuan yang dapat dilihat pada Gambar 5.12.



Gambar 5. 12 Menu Bantuan

## B. HASIL PENGUJIAN

Pengujian yang pertama adalah pengujian terhadap metode *Linear Discriminant Analysis (LDA)* dan *Naïve Bayesian* dengan menggunakan 180 citra, dimana 180 citra tersebut terdiri dari 120 citra latih dan 60 citra uji. Pengujian ini ditujukan untuk melakukan proses pencocokan citra telapak tangan yang telah dipilih sebelumnya pada menu segmentasi yang kemudian diinputkan kedalam sistem. Inputan yang diuji merupakan citra uji dengan berbagai variasi keadaan. Gambar 5.13 berikut merupakan hasil dari citra uji yang berhasil dikenali dengan tepat terhadap citra latih nya :



Gambar 5. 13 Citra uji berhasil dikenali dengan tepat

Dari Gambar 5.13 dapat dilihat bahwa citra uji yang berhasil dikenali dengan tepat, ditandai dengan munculnya citra latih dan nama citra yang sama dengan citra uji. Selain itu sistem juga



ini menyebabkan ekstraksi dari file citra uji tersebut menjadi kurang optimal.

## VI. PENUTUP

### 1. Kesimpulan

Berdasarkan analisa perancangan sistem, implementasi dan pengujian sistem maka dapat disimpulkan bahwa :

1. Berdasarkan hasil pengujian dari seluruh citra uji dengan citra latihan , aplikasi mencocokkan citra dengan persentase keberhasilan 95% pada metode linear discriminant analysis dengan probabilitas metode naïve bayesian.
2. Metode linear discriminant analysis dengan probabilitas naïve bayesian dapat mengenali citra yang diberi noise salt and pepper 0,05 dan blur Gaussian filter 2.5. Hal ini dibuktikan dengan tingkat akurasi masing-masing hasil pencocokan yaitu 93,34 % dan 93,34%.
3. Nilai akurasi dari pengujian sampel yang diambil dari CASIA Palmprint Image database adalah 100%, lebih baik dari citra yang diambil secara acak dari lingkungan sekitar yaitu 92,5%, namun nilai rata-rata akurasi sampel yang diambil dari CASIA Palmprint Image database adalah 0.0000008019, lebih rendah dari citra yang diambil secara acak dari lingkungan sekitar yaitu 0.0000008988.

### 2. Saran

Berdasarkan hasil dan pembahasan yang telah dijabarkan sebelumnya, maka saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

Untuk Penelitian lebih lanjut aplikasi mampu melakukan pencocokan langsung melalui kamera *webcam*.

## REFERENSI

- [1] S. Liu and M. Silverman, *A Practical Guide to Biometric Security Technology*, p. 1, 2001.
- [2] A. Muntasa, *Pengenalan Pola*, Yogyakarta: Graha Ilmu, 2015.
- [3] M. C. Wijaya and A. Prijono, *Pengolahan Citra Digital Menggunakan MatLAB Image Processing Toolbox.*, Bandung: Informatika, 2007.
- [4] A. Chandra and T. G. Calderon, *Toward a Biometric Security Layer*, *Journal of Information Systems*, pp. 51-70, 2003.
- [5] A. Linda, *Penerapan Region of Interest (ROI) pada Metode Kompresi*, Institut Teknologi Bandung, Bandung, 2014.
- [6] U. Ahmad, *Pengolahan Citra Digital*, Graha Ilmu, Bogor, 2005.
- [7] D. Sundar, *Software Engineering*, New Delhi: University Science Press, 2010.
- [8] H. Ramza and Y. Dewanto, *Teknik Pemrograman Menggunakan Matlab*, Jakarta: Grasindo, 2007
- [9] D. Putra, "*Sistem Biometrika Konsep Dasar*," Yogyakarta, ANDI, 2009.
- [10] A. Dennis, B. H. Wixom and D. Tegarden, *Sistem Analysis and Design with UML Version 2.0*, United States of America: John Willey & Sons, Inc., 2005.