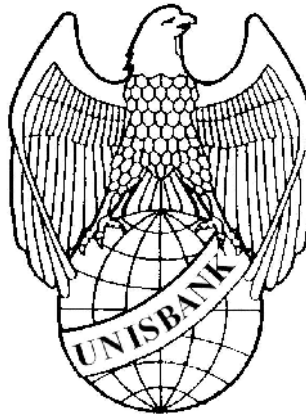


LAPORAN PENELITIAN



PENGOLAHAN CITRA DIGITAL IDENTIFIKASI SIDIKJARI BERBASIS MINUTIAE

Oleh :

Eka Ardhianto, S.Kom., M.Cs / YU.2.09.11.079 (Ketua)
Siti Munawaroh, S.Kom., M.Cs / YU.2.02.10.055 (Anggota)
Agung Prihandono, S.Kom / YU.2.09.11.0080 (Anggota)

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS STIKUBANK (UNISBANK)
SEMARANG
JUNI 2010**

Created with



nitro PDF[®]
created with

professional

download the free trial online at nitropdf.com/professional

download the free trial online at nitropdf.com/professional

HALAMAN PENGESAHAN LAPORAN PENELITIAN

1. a. Judul Penelitian	: PENGOLAH CITRA DIGITAL UNTUK IDENTIFIKASI CIRI SIDIKJARI BERBASIS MINUTIAE
b. Bidang Ilmu	: IPTEK
c. Kategori Penelitian	:
2. Ketua Peneliti	
a. Nama Lengkap dan Gelar	: Eka Ardhianto, S.Kom., M.Cs
b. Jenis Kelamin	: Laki-laki
c. Golongan Pangkat dan NIP	: III A / YU.2.09.11.079
d. Jabatan Fungsional:	: -- / --
e. Jabatan Struktural	: --
f. Fakultas / Program Studi	: Teknologi Informasi / Teknik Informatika
g. Pusat Penelitian	: Universitas Stikubank Semarang
3. Jumlah Anggota Penelitian	: 2 orang
a. Nama Anggota Peneliti I	: Siti Munawaroh, S. Kom., M.Cs
b. Nama Anggota Peneliti II	: Agung Prihandono, S.Kom
4. Lokasi Penelitian	: UNISBANK SEMARANG
5. Kerjasama dengan Institusi Lain	
a. Nama Institusi	: -
b. Alamat	: -
c. Telepon / Fax / e-mail	: -
6. Lama Penelitian	: 3 bulan
7. Biaya Yang Diperlukan	
a. Sumber dari Unisbank	: Rp. 1.500.000,00 (Satu juta lima ratus ribu rupiah)
b. Sumber Lain, sebutkan	: Rp. –

Mengetahui,
Dekan Fakultas Teknologi Informasi

Semarang, 30 Juni 2010
Ketua Peneliti,

(Dwi Agus Diartono, S.Kom, M.Kom)
NIY. Y. 2.92.05.074

(Eka Ardhianto, S.Kom., M.Cs)
NIY.YU.2.09.11.079

Menyetujui,
Ketua LPPM Unisbank

(DR. Dra.Lie Liana, MMSI)
NIY.Y.2.92.07.085

KATA PENGANTAR

Maha suci Allah yang telah memberikan ilmu, berkah kesehatan, inspirasi dan kekuatan malam yang benar keberadaanya. Dia hadir dengan kasih dan sayang yang menguatkan penulis untuk menyelesaikan penelitian ini.

Tiada sempurna suatu ciptaan kecuali dari Allah, sekedar langkah kecil semoga penulisan laporan ini dapat menjadikan sedikit sumbangan ilmu bagi semua.

Semarang, Juni 2010

Penulis

INTISARI

**PENGOLAH CITRA DIGITAL
UNTUK IDENTIFIKASI CIRI SIDIKJARI BERBASIS MINUTIAE**

Eka Ardianto, S.Kom., M.Cs
Siti Munawaroh, S.Kom., M.Cs
Agung Prihandono, S.Kom

Biometrik, termasuk di dalamnya *fingerprint recognition*, secara umum digunakan untuk identifikasi dan verifikasi. *Biometric recognition* merupakan sistem pengenalan atau identifikasi seseorang berdasarkan karakteristik biologis khusus yang dimiliki oleh orang tersebut. Setiap orang memiliki sidik jari yang unik terdiri dari pola garis-garis gelap dari kulit yang naik disebut bubungan (*ridges*). Masing-masing pola garis bubungan menghasilkan suatu bentuk pola area berbeda.

Pada penelitian ini dilakukan pengolahan gambar sidik jari sehingga didapatkan pola anatomi riges sidik jari yang nantinya dapat digunakan sebagai identifikator seseorang. Kegiatan yang dilakukan adalah dengan melakukan proses akuisisi citra, *grayscale*, *thresholding*, *thinning* (*skeletonizing*) dan *patern matching*.

Hasil yang didapatkan adalah bahwa gambar sidik jari dapat dikenali pola ridgesnya dan dari setiap gambar sidik jari menghasilkan pola yang berbeda sehingga dapat digunakan sebagai alat identifikasi.

Kata kunci : *Biometric, Fingerprint identification, image processing*

DAFTAR ISI

	Hal
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
KATA PENGANTAR	iii
INTISARI	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	viii
I. PENDAHULUAN	1
1. Latar belakang masalah	1
2. Perumusan masalah	2
3. Batasan masalah	2
4. Tujuan penelitian	2
5. Manfaat penelitian	2
6. Metode penelitian	3
II. LANDASAN TEORI	4
1. Biometrik sidik jari	4
2. Gambar digital	6
3. Pengolahan gambar digital	7
3.1. Grayscale	7
3.2. Pengembangan otsu	8
3.3. Skeletonizing	11
3.4. Template matching	12
III. RANCANGAN PENELITIAN	14
1. Bahan penelitian	14
2. Alur penelitian	14
2.1. Akuisisi gambar	15

2.2. Grayscale	16
2.3. Pengembangan otsu	16
2.4. Skeletonizing	16
3. Rancangan database	17
4. Rancangan antarmuka	17
IV. IMPLEMENTASI DAN PENGUJIAN	19
1. Implementasi	19
2. Pengujian	23
V. PENUTUP	26
1. Kesimpulan	26
2. Saran	26
DAFTAR PUSTAKA	27
LAMPIRAN	28

DAFTAR GAMBAR

	Hal.
Gambar 1. Grafik perbandingan sistem biometrik	4
Gambar 2. Specimen sidik jari.	5
Gambar 3. Fungsi <i>thresholding</i> nilai tunggal	9
Gambar 4. Notasi mask algoritma penipisan	12
Gambar 5. Corak ujung dan percabangan bubungan (<i>ridge</i>)	12
Gambar 6. Mask patern matching	13
Gambar 7. Alur Proses Identifikasi Sidik Jari	14
Gambar 8. Sample sidik jari	15
Gambar 9. Antarmuka Aplikasi Identifikasi Sidik jari	18
Gambar 10. Proses Grayscale	19
Gambar 11. Proses Otsu Treshoding	22
Gambar 12. Proses Transformasi Gambar ke Matrik	22
Gambar 13. Proses Transformasi Gambar ke Matrik	23
Gambar 14. Proses Identifikasi Sidik Jari	23

DAFTAR TABEL

	Hal.
Tabel 1. Macam anatomi bubungan sidikjari	6
Tabel 2. Rancangan Database	17
Tabel 3. Hasil Proses Identifikasi Sidik Jari	24
Table 4. Hasil Uji Coba Kecocokan Sidik Jari	25

BAB I

PENDAHULUAN

1. LATAR BELAKANG MASALAH

Biometrik, termasuk di dalamnya *fingerprint recognition*, secara umum digunakan untuk identifikasi dan verifikasi. Identifikasi ialah mengenali identitas seseorang, dilakukan perbandingan kecocokan antara data *biometric* seseorang dalam *database* berisi record karakter seseorang. *Biometric recognition* merupakan sistem pengenalan atau identifikasi seseorang berdasarkan karakteristik biologis khusus yang dimiliki oleh orang tersebut. Fungsinya antara lain adalah untuk sistem keamanan dengan mengenali identitas seseorang.

Keamanan adalah sesuatu yang penting untuk menjaga informasi. Teknologi yang dikembangkan tidak selalu memberikan perlindungan yang aman untuk menjaga keaslian data, sehingga dikembangkan teknologi yang menawarkan penggunaan biometrik. Teknologi biometrik menawarkan identifikasi secara biologis yang menggunakan sifat atau ciri yang dimiliki oleh individu sehingga memungkinkan sistem untuk dapat mengenali pengguna secara tepat, misalnya adalah iris mata (*iris*), sidik jari (*fingerprint*), pola tangan (*hand*), tanda tangan (*signature*), wajah (*face*) dan suara (*voice*).

Setiap orang memiliki sidik jari yang unik terdiri dari pola garis-garis gelap dari kulit yang naik disebut bubungan (*ridges*). Masing-masing pola garis bubungan menghasilkan suatu bentuk pola area berbeda. 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 identifikasi.

2. PERUMUSAN MASALAH

Dari uraian diatas, rumusan masalah yang akan dibahas dalam penelitian ini meliputi :

1. Bagaimana pengolahan citra yang dilakukan untuk melakukan identifikasi.
2. Bagaimana proses identifikasi dilakukan untuk mendapatkan ciri sidikjari berdasar *minutiae*.

3. BATASAN MASALAH

Untuk memperjelas permasalahan yang akan dibahas, sekaligus membatasi permasalahan yang akan diteliti, maka batasan – batasan masalah ditentukan sebagai berikut :

1. Sidikjari yang diteliti adalah sidikjari manusia normal.
2. Pola minutiae yang digunakan adalah *ridges end point, island* dan *bifurcation*.

4. TUJUAN PENELITIAN

Tujuan dari penelitian ini adalah: mendapatkan pola ciri sidikjari untuk kebutuhan identifikasi yang disesuaikan dengan data pada database.

5. MANFAAT PENELITIAN

Manfaat yang dapat diperoleh dari adanya penelitian ini adalah: Algoritma yang dihasilkan dari penelitian ini dapat digunakan untuk melakukan identifikasi sidikjari dengan berdasarkan *minutiae*.

6. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah dengan melakukan perekayasa perangkat lunak dengan menggunakan model prototipe. Model prototipe difokuskan pada perancangan prototipe secara cepat pada pendekatan model input dan out-putnya sehingga memungkinkan program dapat dimunculkan secara cepat.

Pengembangan perangkat lunak dengan model prototipe memiliki tahapan sebagai berikut:

1. Pengumpulan Kebutuhan

Pengumpulan kebutuhan adalah proses pendefinisian secara objektif mengenai perangkat lunak yang akan dibangun, mengidentifikasi kebutuhan input dan output. Langkah yang dilakukan dalam mengidentifikasi input adalah dengan melakukan pengambilan gambar sidik jari dan studi kepustakaan dengan mempelajari macam dan jenis riges sidik jari serta model riges sidik jari.

2. Perancangan Kilat

Perancangan kilat yang dilakukan adalah melakukan desain perangkat lunak dan bentuk pengolahan citra yang digunakan untuk mendeteksi model riges sidik jari.

3. Membangun Prototipe

Proses membangun prototipe adalah bentuk implementasi kedalam bahasa pemrograman dari hasil perancangan kilat.

4. Evaluasi Prototipe

Tahap evaluasi prototipe adalah proses dimana hasil pembuatan prototipe disesuaikan dengan kebutuhan, pada tahap ini dimungkinkan untuk melakukan prancangan ulang sehingga dihasilkan bentuk prototipe yang menghasilkan output sesuai dengan kebutuhan.

5. Engineer Product

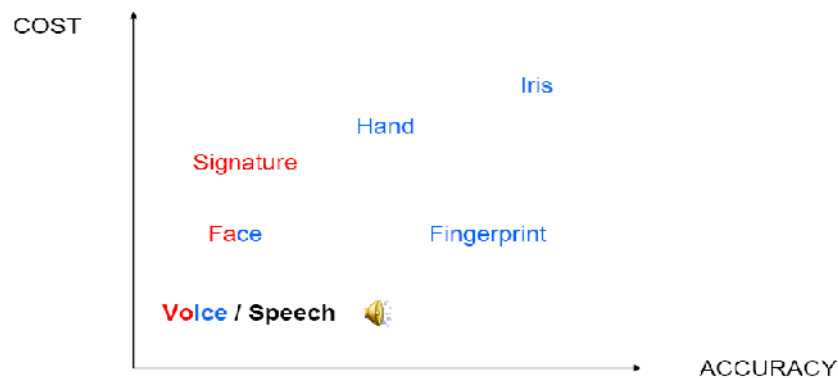
Tahap engineer product adalah bentuk akhir dari prototipe yang dibangun yang telah sesuai dengan kebutuhan.

BAB II

LANDASAN TEORI

1. BIOMETRIK SIDIK JARI

Teknologi biometrik menawarkan otentikasi secara biologis yang menggunakan sifat atau ciri yang dimiliki oleh individu sehingga memungkinkan sistem untuk dapat mengenali pengguna secara tepat, misalnya adalah iris mata (*iris*), sidik jari (*fingerprint*), pola tangan (*hand*), tanda tangan (*signature*), wajah (*face*) dan suara (*voice*). Penggunaan sidik jari merupakan salah satu pola biometrik yang baik untuk melakukan pengujian autentikasi data. Gambar 1 menerangkan bahwa penggunaan sidik jari (*fingerprint*) akan memberikan hasil yang lebih akurat dari mayoritas sistem biometrik yang lain, juga tidak banyak memerlukan biaya (*cost*) dalam pembuatannya.

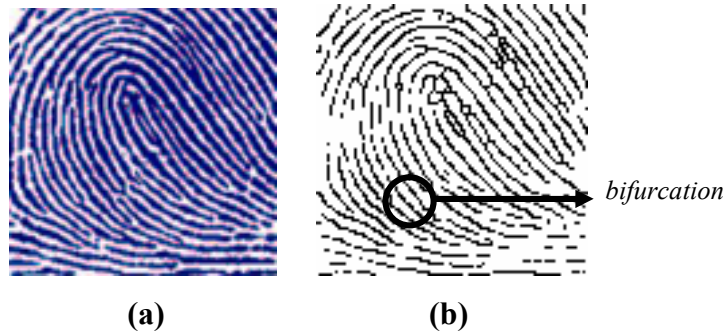


Gambar 1. Grafik perbandingan sistem biometrik

(sumber: <http://scgwww.efpl.ch.courses>)

Setiap orang memiliki sidik jari yang unik terdiri dari pola garis-garis gelap dari kulit yang naik disebut bubungan (*ridges*) yang diperlihatkan sebagai warna putih dan garis-garis terang dari kulit yang turun disebut kerutan (*furrows*) yang diperlihatkan sebagai warna gelap pada gambar 2(a). Area garis bubungan kadang-kadang dikenal sebagai area pola. Masing-masing pola garis bubungan menghasilkan suatu bentuk pola area berbeda. Pusat gambar jari mencerminkan pola area, dikenal sebagai inti dasar (*core*

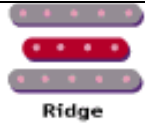








point). Titik awal pencabangan ganda atau corak anatomi pada penyimpangan dua bentuk garis disebut *delta*. Bubungan pada sidik jari yang terputus disebut ujung bubungan. Gambar 2(b) memperlihatkan bentuk *delta* sebagai bentuk garis percabangan (*bifurcation*).



Gambar 2. Specimen sidik jari.
(a) hasil scanning (b) hasil penipisan

Dalam hidup, pola bubungan hanya diubah secara kebetulan akibat luka-luka, terbakar, penyakit atau penyebab lain yang *tidak* wajar. Identifikasi dari sidik jari memerlukan pembedaan tentang bentuk keliling garis bubungan tak terputuskan yang diikuti oleh pemetaan tentang gangguan atau tanda anatomi bubungan yang sama. Berdasarkan bentuk bubungan, menurut penelitian yang dilakukan oleh Syamsa (2004) macam ukuran karakteristik anatomi bubungan adalah seperti terlihat pada tabel 1.

Tabel 1. Macam anatomi bubungan sidikjari

 Ridge	Ridge	Mempunyai ketegasan jarak ganda dari permulaan ke-akhir, sebagai lebar ridges satu dengan lainnya
 Evading Ends	Evading Ends	dua ridge dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3mm.
 Bifurcation	Bifurcation/ percabangan	dua ridge dengan arah berbeda berjalan sejajar satu sama lain kurang dari 3mm.
 Hook	Hook	ridges merobek; satu ridges tidaklah lebih panjang dibanding 3mm
 Fork	Fork	Dua ridges dihubungkan oleh sepertiga ridges tiddak lebih panjang dibanding 3mm
 Dot	Dot	Bagian ridges adalah tidak lagi dibanding ridges yang berdekatan
 Eye/Island	Eye	ridges merobek dan menggabungkan lagi di dalam 3mm
 Enclosed Ridge	Enclosed Ridge	Ridges tidak lebih panjang dibanding 6mm antara dua ridges
 Enclosed Loop	Enclosed Loop	yang tidak mempola menentukan pengulangan antar dua atau lebih ridges paralel

2. GAMBAR DIGITAL

Suatu gambar sering disebut dengan citra atau *image*. Sebuah gambar didefinisikan sebagai gambar yang terdapat pada bidang dwi matra atau dua dimensi. Menurut Munir (2004), secara matematis gambar merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi. Sebuah objek dua dimensi yang diterangi oleh sumber cahaya akan memantulkan kembali sebagian dari berkas cahaya yang diterima. Alat optik yang menerima pantulan cahaya, misalnya kamera ataupun alat pemindai akan merekam

bayangan objek yang hasilnya direpresentasikan dalam bentuk nilai kontinyu yang dapat diolah disebut dengan gambar digital. Suatu gambar digital terbentuk dari kumpulan titik pembentuk gambar yang disebut dengan *pixel* (piksel). Ahmad (2005) menyatakan bahwa piksel merupakan sampel dari pemandangan yang mengandung intensitas cahaya yang dinyatakan dalam bilangan bulat. Sehingga gambar digital adalah kumpulan piksel yang dinotasikan dalam bentuk bilangan pada sejumlah baris dan kolom. Gambar yang dimaksud didalam penulisan ini adalah "gambar diam" yaitu gambar tunggal yang tidak bergerak. Untuk selanjutnya gambar diam akan disebut dengan gambar saja.

3. PENGOLAHAN GAMBAR DIGITAL

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. Sebagai contoh adalah gambar 2 (a) merupakan gambar sidik jari yang diambil secara langsung, namun untuk mengetahui ciri sidik jari maka gambar awal ditipiskan seperti terlihat pada gambar 2 (b) sebelum dilakukan proses lebih lanjut.

Image processing pada dasarnya adalah memodifikasi setiap piksel dalam gambar sesuai dengan kebutuhan. Operasi gambar yang digunakan pada penulisan ini adalah *grayscale*, pengambangan (*thresholding*), *thinning* (penipisan) dan *template matching*.

3.1. Grayscale

Grayscale adalah proses penyederhanaan gambar dari format gambar berwarna RGB menjadi gambar berwarna abu-abu (*grey*). Suatu gambar berwarna RGB memiliki tiga lapisan matrik yaitu *R-layer*, *G-layer* dan *B-layer*. Bila setiap proses perhitungan dilakukan pada setiap lapisan, maka satu piksel akan dikenakan tiga kali operasi, sehingga konsep tiga layer RGB

disederhanakan menjadi sebuah lapisan yaitu lapisan *greyscale*. Dalam gambar ini tidak ada lagi warna, yang ada adalah derajat keabuan. Untuk mengubah gambar berwarna yang mempunyai nilai matrik masing-masing R , G dan B menjadi gambar *gray scale* dengan nilai k , maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R , G dan B (Ahmad, 2005) sehingga secara mudah dapat dituliskan seperti persamaan (2.1).

$$k = (R + G + B) / 3 \quad (2.1)$$

Ahmad (2005) menuliskan juga bahwa karena ketiga warna R , G dan B dianggap tidak seragam dalam hal kemampuan kontribusi terhadap kecerahan, ada yang berpendapat bahwa cara konversi lebih tepat menggunakan persamaan (2.2).

$$k = (0,299 R + 0,587 G + 0,114 B) \quad (2.2)$$

3.2. Pengambangan Otsu (*Otsu Thresholding*)

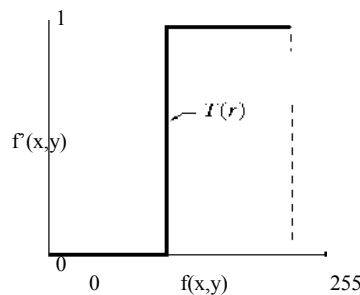
Pengambangan atau *thresholding* merupakan salah satu cara melakukan segmentasi gambar atau pemisahan gambar menjadi kelompok-kelompok yang mewakili suatu wilayah. Proses segmentasi gambar dengan menggunakan *thresholding* pada dasarnya adalah memisahkan nilai piksel gambar berdasar sebuah atau beberapa nilai batas (*threshold*). Dalam penulisan ini gambar sidik jari disegmentasikan menjadi dua wilayah, yaitu wilayah *ridge* dan wilayah *furrows*, sehingga gambar hasil hanya akan memiliki satu nilai *threshold* (T) dengan dua macam warna saja yaitu putih dan hitam atau identik dengan 0 dan 1.

$$f^o(x,y) = \begin{cases} 0, & f(x,y) < T \\ 1, & f(x,y) > T \end{cases} \quad (2.3)$$

atau

$$f^*(x,y) = \begin{cases} 0, & f(x,y) > T \\ 1, & f(x,y) < T \end{cases} \quad (2.4)$$

Fungsi (2.3) dan (2.4) (Munir, 2004) dapat dijelaskan bahwa nilai piksel (x,y) dari gambar hasil $f^*(x,y)$ akan bernilai 0 jika nilai piksel (x,y) gambar awal $f(x,y)$ kurang dari nilai *threshold* (T) dan Nilai piksel (x,y) dari gambar hasil $f^*(x,y)$ akan bernilai 1 jika nilai piksel (x,y) gambar awal $f(x,y)$ menyamai atau melebihi dari nilai *threshold* (T), hal ini juga berlaku untuk sebaliknya. Lebih jelas bentuk visualisasi fungsi pada gambar



Gambar 3. Fungsi *thresholding* nilai tunggal

Penentuan nilai *threshold* (T) dapat dianalisa dengan menggunakan histogram gambar, dimana nilai T diambil pada nilai terendah pada wilayah lembah histogram. Metode penentuan nilai T pada penulisan in dipilih metode Otsu.

Metode Otsu (Otsu, 1979) menghitung nilai ambang (T) secara otomatis berdasarkan gambar masukan. Pendekatan yang digunakan oleh metode Otsu adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis Diskriminan akan memaksimumkan variabel tersebut agar dapat memisahkan objek dengan latar belakang. Misalkan nilai ambang yang akan dicari dinyatakan dengan k . Nilai k berkisar antara 1 sampai dengan L , dengan $L = 255$. Probabilitas untuk piksel i dinyatakan dengan :

$$P_i = \frac{n_i}{N} \quad (2.5)$$

dengan n_i menyatakan jumlah piksel dengan tingkat keabuan i , dan N menyatakan banyaknya piksel pada gambar. Nilai momen kumulatif ke-nol, momen kumulatif ke-satu, dan nilai rata-rata berturut-turut dapat dinyatakan sebagai berikut :

$$\omega(k) = \sum_{i=1}^k P_i \quad (2.6)$$

$$\mu(k) = \sum_{i=1}^k i.P_i \quad (2.7)$$

$$\mu_T = \sum_{i=1}^L i.P_i \quad (2.8)$$

nilai ambang k dapat ditentukan dengan memaksimumkan persamaan:

$$\sigma_B^2(k^*) = \max_{1 \leq k \leq L} \sigma_B^2(k)$$

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad (2.9)$$

untuk menghitung nilai ambang (T), pertama adalah dengan membuat histogram dari gambar yang dimaksud yaitu dengan menambah nilai k jika terdapat piksel yang memiliki nilai sama dengan level (L), dalam hal ini level (L) adalah level untuk gambar dengan format *greyscale* mulai dari 0 (nol) sampai dengan 255. Selanjutnya adalah menghitung nilai probabilitas untuk setiap level (L) dengan membandingkan antara jumlah piksel pada level dengan total piksel pada gambar atau dapat dituliskan seperti pada persamaan (2.5). Kemudian untuk setiap level dihitung jumlah nilai momen kumulatif ke-0 dengan persamaan (2.6), momen kumulatif ke-1 dengan persamaan (2.7) dan

nilai rata-rata dengan persamaan (2.8) yang selanjutnya dihitung nilai varian dengan persamaan (2.9). Jika seluruh level telah dihitung, selanjutnya dicari nilai varian yang paling tinggi untuk digunakan sebagai nilai ambang (T).

3.3. Skeletonizing (*Thinning*)

Suatu pendekatan representasi bentuk struktur dari sebuah objek adalah dengan mereduksi menjadi suatu *graph* (Gonzalez dan Woods, 1993). *Graph* yang dimaksud adalah bentuk objek yang lebih tipis atau sebuah garis kerangka (*skeleton*) yang dihasilkan dari algoritma *thinning* (penipisan) atau sering disebut dengan *skeletonizing*. Algoritma penipisan (*thinning algorithm*) (Davies dan Plummer, 1980) adalah proses berulang (*recuring*) yang dimaksudkan untuk membuang atau mengurangi bagian dari objek yang tidak perlu sehingga hanya dihasilkan informasi yang mendasar untuk memfasilitasi proses selanjutnya. Dalam Munir (2004), Pitas dan Ioannis menyatakan bahwa algoritma penipisan objek harus memenuhi persyaratan sebagai berikut :

1. Mempertahankan keterhubungan piksel piksel objek. Dengan kata lain, tidak menyebabkan bentuk objek menjadi terputus.
2. Tidak memperpendek ujung lengan dari bentuk yang ditipiskan.

Algoritma penipisan (Gonzalez dan Woods, 1993) menggunakan *mask* (matrik) dengan ordo 3 x 3 yang pada penulisan ini akan disebut sebagai *mask*, dan mask tersebut mengupas piksel bagian luar dari objek, bentuk notasi matrik 3 x 3 diperlihatkan pada gambar 3.5. Mask penipisan dilakukan secara berulang mulai dari posisi awal gambar hingga akhir gambar dengan langkah sebagai berikut :

1. Tandai titik P_8 sebagai titik yang akan dihapus, bila P_8 memenuhi persyaratan berikut :

a. $(2 \leq N(P_t) \leq 6)$ and $(S(P_t) = 1)$

dengan asumsi nilai $P_8 = 1$, $N(P_t)$ adalah jumlah tetangga dari P_8 yang nilainya sama dengan P_8 dirumuskan seperti persamaan

(2.10), dan $S(P_t)$ adalah jumlah transisi dari 0 ke 1 dalam barisan $P_0 + P_1 + \dots + P_6 + P_7$.

$$N(P_t) = P_0 + P_1 + \dots + P_6 + P_7 \quad (2.10)$$

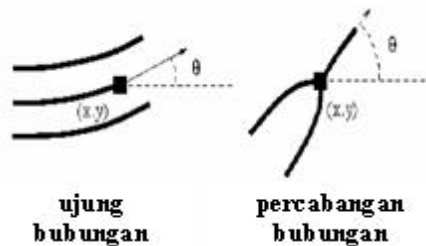
2. Hapus semua titik yang ditandai pada langkah 1.
3. Ulangi langkah 1 dan 2 sampai tidak ada lagi yang dihapus.
4. Simpan hasil akhir sebagai gambar baru.

P_0	P_1	P_2
P_7	P_8	P_3
P_6	P_5	P_4

Gambar 4. Notasi mask algoritma penipisan

3.4. Template Matching

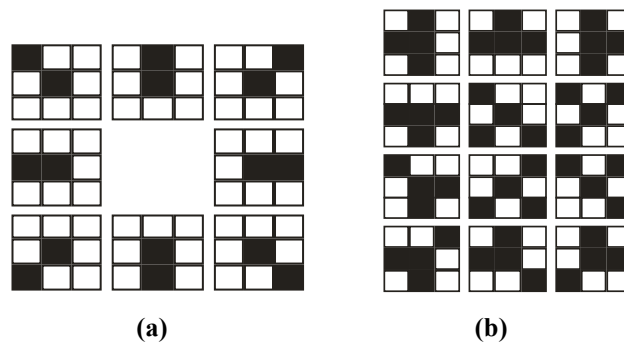
Template matching merupakan langkah akhir dalam mencari ciri *ridge* dasar sidik jari. Teknik *template matching* berdasarkan corak merupakan teknik paling terkenal dan paling banyak digunakan (Budiman, 2006). Dalam pendekatan ini, sidik jari digambarkan sebagai vektor garis dengan lebar satu piksel dari bentuk *ridge* percabangan (*bifurcation*) dan ujung bubungan (*end point*) seperti diperlihatkan pada gambar 5 (Budiman, 2006).



Gambar 5. Corak ujung dan percabangan bubungan (*ridge*)

Corak sidik jari dinotasikan dengan matrik (*mask*) dengan ordo 3 x 3. Untuk tipe *ridge end point*, terdapat 8 pola mask seperti diperlihatkan pada

gambar (6.a), sedangkan untuk pola *ridge bifurcation* terdapat 12 pola *mask* yang ditunjukkan seperti pada gambar (6.b). Selanjutnya operasi dilakukan untuk mendapatkan jumlah *end point* dan bifurcation dengan cara mencocokkan setiap *mask* dengan posisi piksel yang bersesuaian.



Gambar 6. Mask patern matching

(a) *end point ridge*, (b) *bifurcation ridge*

untuk menambah keyakinan bahwa sidikjari setiap manusia adalah berbeda, maka model dot ditambahkan dalam bentuk satu piksel.

BAB III

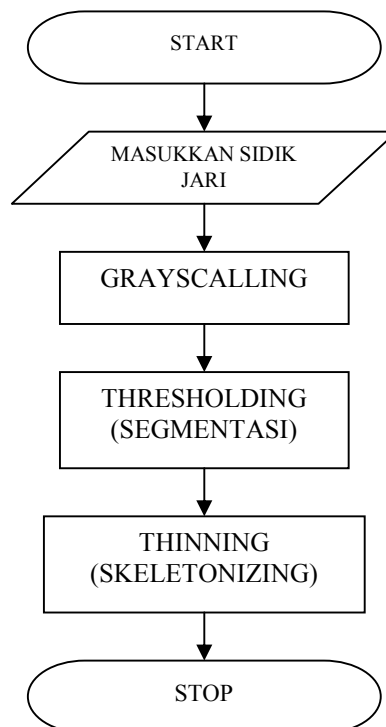
RANCANGAN PENELITIAN

1. BAHAN PENELITIAN

Penelitian ini membutuhkan bahan berupa dua macam gambar, yaitu gambar sidik jari dan gambar sidik jari untuk melakukan pengujian. Gambar yang digunakan adalah gambar dalam bentuk file gambar dengan format .bmp yang dimaksudkan untuk mempermudah pengolahan yang menggunakan delphi sebagai alat bantu penelitian.

2. ALUR PENELITIAN

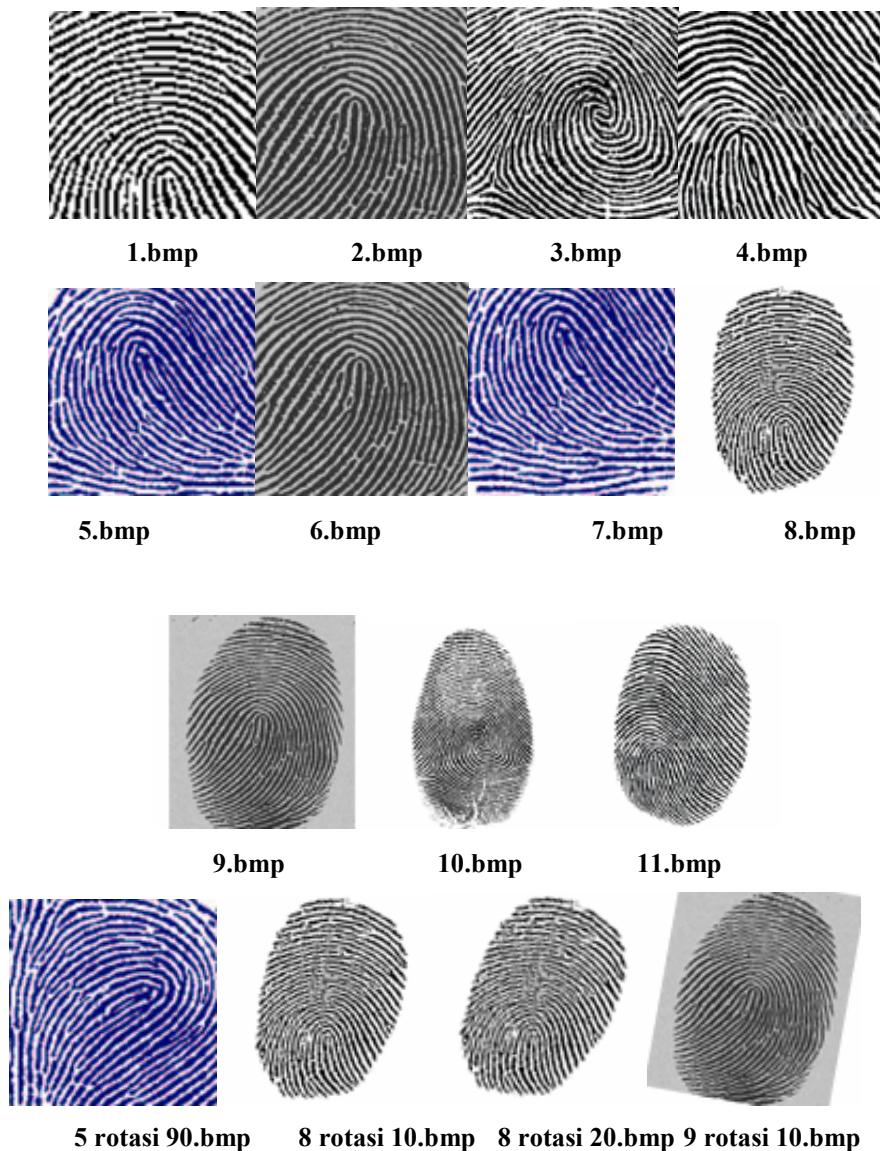
Perancangan alur proses ini bertujuan untuk memudahkan pemahaman terhadap jalannya penelitian. Proses Identifikasi Sidikjari akan melalui beberapa tahapan yaitu akuisisi sidik jari, *grayscale*, *thresholding*, *thinning* seperti ditunjukkan pada gambar



Gambar 7. Alur Proses Identifikasi Sidik Jari

2.1. Akuisisi Gambar

Akuisisi gambar merupakan langkah awal dari proses Pengolahan Sidikjari. Akuisisi gambar dilakukan dengan cara memindai kertas yang sudah terdapat cap jari melalui alat *scanner*. Selanjutnya sidik jari diolah dengan alat bantu *photoshop* untuk mendapatkan ukuran 100 x 100 dengan resolusi standar 72 dpi (*dot per inch*). Sidik jari diambil pada tiga orang yang berbeda yang kemudian diberi nama yang berbeda seperti terlihat pada gambar 8.



Gambar 8. Sample sidik jari

2.2. Grayscale

Grayscale merupakan suatu proses yang digunakan untuk merubah gambar dari format RGB (*full color*) menjadi tipe warna skala abu-abu. Pada dasarnya proses perubahan gambar dengan tipe warna *fullcolor* menjadi abu-abu dilakukan dengan menggunakan persamaan (2.1).

2.3. Otsu Thresholding

Selanjutnya adalah proses pengubahan gambar dari format *grayscale* menjadi gambar dengan format warna biner dengan menggunakan nilai pengambang (*threshold*). Proses *thresholding* ini intinya adalah membandingkan nilai level abu-abu gambar dengan sebuah nilai ambang (T) untuk setiap piksel pada sidik jari. Pada penelitian ini akan digunakan metode Otsu untuk menentukan nilai T.

Metode Otsu (Otsu, 1979) menghitung *nilai ambang* (T) secara otomatis berdasarkan gambar masukan seperti yang telah dijelaskan pada bab. Pendekatan yang digunakan oleh metode Otsu adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis diskriminan akan memaksimumkan variabel tersebut agar dapat memisahkan objek dengan latar belakang.

2.4. Thinning

Algoritma penipisan (*thinning algorithm*) (Davies dan Plummer, 1980) adalah proses berulang (*recuring*) yang dimaksudkan untuk membuang atau mengurangi bagian dari objek yang tidak perlu sehingga hanya dihasilkan informasi yang mendasar untuk memfasilitasi proses selanjutnya yaitu pencocokan pola *ridge*.

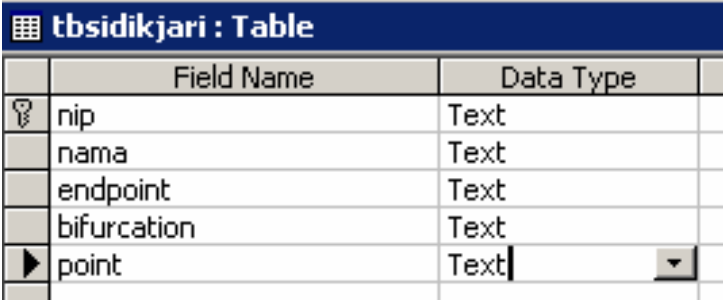
Algoritma penipisan (Gonzalez dan Woods, 1993) menggunakan *mask* (matrik) dengan ordo 3 x 3 sebagai *mask*, dan mask tersebut mengupas piksel bagian luar dari objek, bentuk notasi matrik 3 x 3 diperlihatkan pada gambar 6. Mask penipisan dilakukan secara berulang mulai dari posisi awal gambar

hingga akhir gambar sehingga dihasilkan gambar yang memiliki objek dengan tebal hanya satu piksel saja. Dalam penelitian ini yang digunakan adalah algoritma thinning seperti yang dituliskan oleh Gonzallez dan Woods (1993).

3. RANCANGAN DATABASE

Dalam perancangan ini, digunakan sebuah tabel yang dimaksudkan untuk menyimpan data sidik jari pengirim gambar yang diberinama *tbsidikjari*. Tbfinger digunakan pada proses identifikasi gambar untuk mengenali sidik jari pengirim gambar. Untuk membuat tabel *tbsidikjari* digunakan perangkat database dari perangkat Microsoft Access. Bentuk rancangan tabel *tbsidikjari* diperlihatkan pada tabel 2.

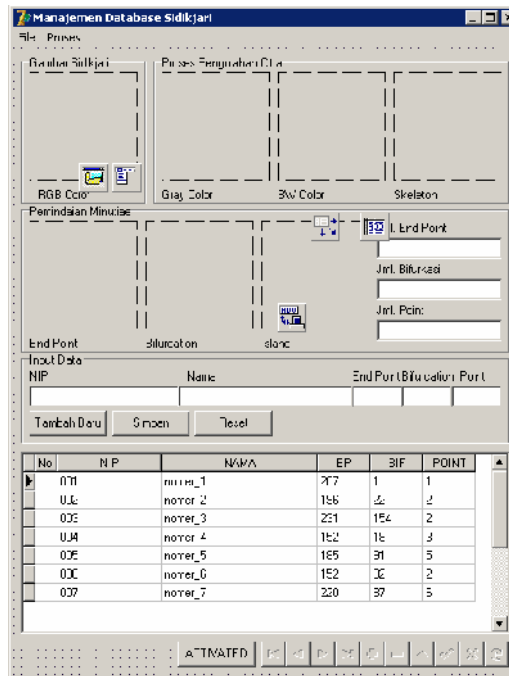
Tabel 2. Rancangan Database



	Field Name	Data Type
🔑	nip	Text
	nama	Text
	endpoint	Text
	bifurcation	Text
▶	point	Text

4. RANCANGAN ANTARMUKA

Antarmuka merupakan bentuk visual aplikasi yang dimaksudkan sebagai perantara antara pengguna dengan program aplikasi. Bentuk antarmuka yang digunakan terdapat beberapa bagian yaitu: Bagian Gambar sidik jari yang memuat gambar sidik jari asli, Bagian Proses Pengolahan Gambar yaitu menunjukkan langkah-langkah hasil pengolahan gambar, Bagian Pemindaian Minutiae yaitu proses Identifikasi anatomi riges sidik jari dan Bagian Input data yang digunakan untuk melakukan penyimpanan hasil identifikasi. Model antarmuka diperlihatkan pada gambar 9.



Gambar 9. Antarmuka Aplikasi Identifikasi Sidik jari

BAB IV

IMPLEMENTASI DAN PENGUJIAN

1. IMPLEMENTASI

Implementasi yang dibuat adalah berupa penterjemahan rancangan kedalam bahasa pemrograman. Hasil dari implementasi adalah Aplikasi Identifikasi Gambar Sidik Jari. Bagian per bagian akan dijelaskan pada sub bab bagian implementasi. Implementasi dan pengujian dilakukan dengan menggunakan alat bantu pemrograman Delphi.

Proses Identifikasi memerlukan gambar berupa gambar sidik jari yang telah dilakukan proses akuisisi gambar yang kemudian dilakukan proses *grayscale*, *thresholding* dan *thinning*. Proses *grayscale* yang difungsikan untuk merubah gambar RGB kebentuk abu-abu (*gray*). Dalam implementasi, gambar sidikjari dimasukkan kedalam komponen penampung gambar yaitu *TImage*. Tipe gambar yang dikenal delphi secara dasar adalah file berformat bitmap dengan tiga bentuk warna, sehingga bentuk gambar gambar yang disimpan secara *grayscale*, dikenal delphi sebagai *grayscale* dengan tiga warna yaitu RGB. Gambar 10 menunjukkan penggalan bentuk implementasi *grayscale*.

```
For i:=0 to Image1.Picture.Height-1 do
begin
  PC:=Image1.Picture.Bitmap.ScanLine[i];
  PH:=Image2.Picture.Bitmap.ScanLine[i];
  for j:=0 to Image2.Picture.Width-1 do
  begin
    PH[j]:=round((PC[3*j]+PC[3*j+1]+PC[3*j+2])/3);
  end;
end;
```

Gambar 10. Proses Grayscale

Pada dasarnya proses yang terjadi adalah mengubah nilai salah satu dari tiga matrik RGB dengan persamaan (2.1). Supaya bentuk visual gambar tetap terlihat abu-abu maka matrik yang lain juga diubah nilainya sesuai dengan persamaan yang digunakan. Untuk selanjutnya proses tiga matrik dianggap sebagai satu matrik saja, meskipun proses implementasi dikenakan pada ketiga matrik RGB.

Proses dilanjutkan dengan pemisahan area latar belakang dengan objek yaitu dengan menggunakan proses *thresholding*. Sesuai dengan rancangan, Metode Otsu digunakan sebagai proses *thresholding* untuk merubah gambar menjadi dua bagian yaitu bagian latar yang berwarna putih dan bagian objek yang berwarna hitam dengan nilai batas ambang (*threshold*) adalah k . Bentuk penggalan *sourcecode* metode Otsu diperlihatkan pada gambar 11. Metode Otsu menghitung nilai ambang (T) secara otomatis berdasarkan citra masukan. Pendekatan yang digunakan oleh metode Otsu adalah dengan melakukan analisis diskriminan yaitu menentukan suatu variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis diskriminan akan memaksimalkan variabel tersebut agar dapat memisahkan objek dengan latar belakang. Untuk mendapatkan nilai ambang k pertama kali dibentuk histogram gambar selanjutnya dilakukan penghitungan nilai probabilitas untuk setiap level. Proses selanjutnya adalah menghitung *TotalMean*, *zerothCumumoment*, *firstCumumoment* dan mencari nilai maksimum *variance* untuk setiap perhitungan *TotalMean*, *zerothCumumoment*, *firstCumumoment*. Jika diketemukan nilai *variance* yang paling tinggi, posisi level tersebut yang digunakan untuk nilai ambang (*threshold*).

Threshold (k) yang dihasilkan adalah nilai *threshold* yang digunakan untuk gambar dengan format *grayscale* yang memiliki warna antara 0 sampai dengan 255, sedangkan yang dihadapi adalah gambar dengan format RGB. Nilai *threshold (k)* diimplementasikan kedalam gambar dengan cara menghitung ulang nilai *threshold* yang dikalikan dengan perbandingan nilai

warna putih dengan nilai maksimum skala abu-abu sehingga nilai *threshold* yang diimplementasikan terhadap gambar adalah sebanding dengan nilai *threshold* yang dihitung untuk model gambar *grayscale*, dalam implementasi ini nilai *threshold* yang baru disimpan dalam variable *kt*. Nilai *kt* selanjutnya dibandingkan dengan nilai warna gambar untuk membuat gambar hanya memiliki dua warna yaitu warna putih dan warna hitam, bentuk implementasi perhitungan *kt* dan pemisahan wilayah.

```

Threshold:=0;
  totalMean := 0;
  maxVariance := 0;
  firstCumumoment := 0;
  zerothCumumoment := 0;
  area := Image.Height * Image.Width;
  for k:= 0 to level do
    TotalMean := TotalMean + (k * histogram[k] / area);
    for k:= 0 to level do
      begin
        zerothCumumoment := zerothCumumoment + histogram[k] / area;
        firstCumumoment := firstCumumoment + (k * histogram[k] /
area);
        variance := totalMean * zerothCumumoment - firstCumumoment;
        variance := variance * variance;
        if ((zerothCumumoment <> 0) and (zerothCumumoment <> 1))
then
          begin
            variance := variance / (zerothCumumoment * (1 -
zerothCumumoment));
            if (maxVariance < variance) then
              begin
                maxVariance := variance;
                threshold := k;
              end;
            end;
          end;
        for i:=0 to Image.Height-1 do begin
          for j:=0 to Image.Width-1 do begin
            kt:= round(threshold*(clWhite div 255));

```

```

if Image.Canvas.Pixels[j,i] < kt then
  px[j,i]:= clBlack
else
  px[j,i]:= clWhite;
end;
end;
for i:=0 to Image.Height-1 do begin
  for j:=0 to Image.Width-1 do begin
    Image3.Picture.Bitmap.Canvas.Pixels[j,i]:=px[j,i];
  end;
end;
end;

```

Gambar 11. Proses Otsu Treshoding

Proses selanjutnya adalah proses penipisan objek hingga terbentuk kerangka (*skeleton*) dari objek. Dalam implementasi, untuk mempermudah proses, gambar yang dihasilkan dari proses *thresholding* ditransformasikan kedalam matrik yang berukuran sama dengan gambar dan diberi nilai 0 untuk piksel hitam dan nilai 1 untuk piksel warna putih. Bentuk *sourcecodenya* dapat dilihat pada gambar 12.

```

for k:=0 to m2 do begin
  for l:=0 to m1 do begin
    if image.Canvas.Pixels[l,k]=clBlack then begin
      img1[l,k]:=0;    img2[l,k]:=0;
    end else begin
      img1[l,k]:=1;    img2[l,k]:=1;
    end;
  end;
end;
end;

```

Gambar 12. Proses Transformasi Gambar ke Matrik

penggunaan dua matrik yang diimplementasikan dimaksudkan untuk melakukan perbandingan antara matrik awal dan matrik yang diproses. Selanjutnya matrik yang dihasilkan diolah sesuai dengan dasar teori penipisan yang diterangkan pada bab 3.3.3 dan bab 3.3.4.

Hasil pengolahan matrik 0 dan 1, selanjutnya diimplementasikan kedalam bentuk gambar. Untuk setiap posisi nilai yang bersesuaian dilakukan transformasi angka kedalam bentuk warna. Jika dalam matrik bernilai 0 maka ditransformasikan dalam warna hitam, dan sebaliknya jika dalam matrik bernilai 1 maka ditransformasikan dalam warna putih. Bentuk implementasi ditunjukkan pada penggalan *sourcecode* pada gambar 13.

```

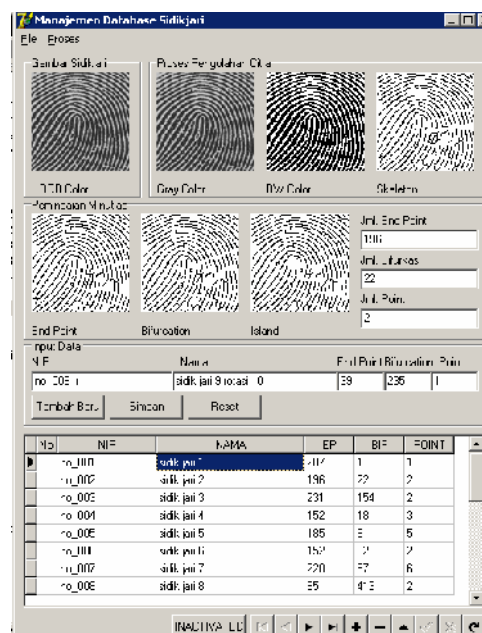
if img1[l,k]=1 then
    Image1.Picture.Bitmap.Canvas.Pixels[l,k]:= clWhite
else
    Image1.Picture.Bitmap.Canvas.Pixels[l,k]:= clBlack;
End;

```

Gambar 13. Proses Transformasi Gambar ke Matrik

2. PENGUJIAN

Sub bab ini akan membahas mengenai pengujian hasil implementasi untuk melihat seberapa jauh hasil dari konsep yang dirancang. Pengujian dilakukan sebanyak dua macam: pencarian pola sidik jari, uji coba kecocokan sidik jari.



Gambar 14. Proses Identifikasi Sidik Jari

Pada proses Identifikasi Sidikjari, digunakan 11 macam gambar sidikjari yang berbeda seperti diperlihatkan pada gambar 8. Identifikasi dilakukan pada pencarian anatomi riges end point, point dan bifurcation seperti terlihat pada tabel 3.

Tabel 3. Hasil Proses Identifikasi Sidik Jari

tbsidikjari				
nip	nama	endpoint	bifurcation	point
no_001	sidik jari 1	207	1	1
no_002	sidik jari 2	196	22	2
no_003	sidik jari 3	231	154	2
no_004	sidik jari 4	152	18	3
no_005	sidik jari 5	185	91	5
no_006	sidik jari 6	152	32	2
no_007	sidik jari 7	220	87	6
no_008	sidik jari 8	85	413	2
no_009	sidik jari 9	46	496	4
no_010	sidik jari 10	17	174	2
no_011	sidik jari 11	27	971	0

Dari tabel 3, dapat dikatakan bahwa setiap sidikjari yang dijadikan sample memiliki anatomi riges yang berbeda.

Selanjutnya adalah melakukan ujicoba dengan menggunakan gambar sisik jari yang sama, namun dilakukan perubahan pada gambar yaitu dengan memutar gambar, sehingga hasilnya adalah seperti terlihat pada tabel 4.

Table 4. Hasil Uji Coba Kecocokan Sidik Jari

tbsidikjari				
nip	nama	endpoint	bifurcation	point
no_005	sidik jari 5	185	91	5
no_005_r	sidik jari 5 rotasi 90	183	91	6
no_008	sidik jari 8	85	413	2
no_008_r1	sidik jari 8 rotasi 10	64	354	3
no_008_r2	sidik jari 8 rotasi 20	49	286	3
no_009	sidik jari 9	46	496	4
no_009_r	sidik jari 9 rotasi 10	39	235	1

Dari tabel 4, dapat dilihat bahwa gambar sidik jari yang dilakukan perubahan akan menghasilkan nilai anatomi yang berbeda. Terlihat sebagai contoh adalah no_005 yang memiliki kombinasi anatomi 185, 91, 5 dan no_005_r adalah gambar sidik jari yang dirotasi sebesar 10 derajat yang menghasilkan kombinasi anatomi 183, 91, 6.

BAB V

KESIMPULAN

1. KESIMPULAN

Dari penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan yaitu :

1. Proses pengolahan citra yang dilakukan dapat mengenali bentuk anatomi ridges sidik jari.
2. Setiap sidikjari memiliki pola kombinasi anatomi yang berbeda sehingga dapat digunakan sebagai salah satu cara pengidentifikasian seseorang.

2. SARAN

Untuk perbaikan dan proses penelitian selanjutnya sehingga diusulkan saran sebagai berikut :

1. *Sidik jari* pada setiap orang sebaiknya diambil lebih dari satu kali, karena pergeseran sidik jari pada saat akuisisi gambar mungkin dapat mengakibatkan perubahan ciri data *ridge* yang diolah.
2. Pada proses pengambilan sample sidikjari diharapkan dapat menggunakan perangkat scanner khusus sesuai dengan bentuk anatomi jari manusia.

DAFTAR PUSTAKA

- Ahmad, U., 2005, *Pengolahan Citra Digital Dan Teknik Pemrogramannya*, Graha Ilmu, Yogyakarta.
- Ardisasmita, M. S., 2004, Pengembangan Model Matematika untuk Analisis Sistem Identifikasi Sidik Jari Otomatis, *Komputasi dalam sains dan teknologi nuklir XII*, Pusat Pengembangan Teknologi Informatika dan Komputasi, BATAN.
- Blyvas, I., Bruckstein, A., Kimmel, R., 2005, Efficient Computation of Adaptive Threshold Surfaces for Image Binarization, *Pattern Recognition*, No. 39, 89-101, www.elsevier.com/locate/patcog
- Gonzales, R. C., Woods, R. E., 1993, *Digital Image Processing*, Wesley Publishing Company, USA.
- Nobuyuki, O., 1979, A Threshold Selection Method From Grey-Level Histograms, *IEEE Trans. Syst. Man Cybern*, hal. 62-66.
- Putra, D., 2004, Binerisasi Citra Tangan dengan Metode Otsu, *Jurnal Teknologi Elektro Fak. Teknik Universitas Udayana*, No. 2, Vol. 3, Hal. 11-13.
- Rinaldi Munir, "Pengolahan Citra Digital dengan Pendekatan Algoritmik", Informatika Bandung, 2004
- Suyanto, A.H., 2005, *Review Metodologi Pengembangan Perangkat Lunak*, www.asep_hs.web.ugm.ac.id

Created with



PERSONALIA PENELITIAN

1. Ketua Penelitian :
 - a. Nama Lengkap : Eka Ardhianto, S.Kom., M.Cs.
 - b. Jenis Kelamin : Laki – Laki
 - c. NIP : YU.2.09.11.0079
 - d. Disiplin Ilmu : Teknik Informatika
 - e. Pangkat / Golongan : III A
 - f. Jabatan Fungsional / Struktural : -
 - g. Fakultas / Jurusan : Teknologi Informasi / Teknik Informatika
 - h. Waktu Penelitian : 3 bulan
 - i.
2. Anggota Penelitian :
 - Anggota I :
 - a. Nama Lengkap : Siti Munawaroh, S.Kom., M.Cs
 - b. Jenis Kelamin : Perempuan
 - c. NIP : YU.2.02.10.055
 - d. Disiplin Ilmu : Teknik Informatika
 - e. Pangkat / Golongan : III B
 - f. Jabatan Fungsional / Struktural : Lektor - Penata Muda Tk. I.
 - g. Fakultas / Jurusan : Teknologi Informasi / Teknik Informatika
 - Anggota II :
 - a. Nama Lengkap : Agung Prihandono, S.Kom
 - b. Jenis Kelamin : Laki - Laki
 - c. NIP : YU.2.09.11.0080
 - d. Disiplin Ilmu : Teknik Informatika
 - e. Pangkat / Golongan : III A
 - f. Jabatan Fungsional / Struktural : -
 - g. Fakultas / Jurusan : Teknologi Informasi / Teknik Informatika
3. Tenaga Laboran / Teknisi : -
4. Pekerja Lapangan : -
5. Tenaga Administrasi : -

LISTING PROGRAM

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Menus, ExtDlgs, ComCtrls, DB, DBTables,
  ADODB, Mask, DBCtrls, Grids, DBGrids;

type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    Image1: TImage;
    GroupBox2: TGroupBox;
    Image2: TImage;
    Image3: TImage;
    Image4: TImage;
    MainMenu: TMainMenu;
    File1: TMenuItem;
    AmbilGambarSidikjaril: TMenuItem;
    Proses1: TMenuItem;
    AmbilCiriSidikjaril: TMenuItem;
    Exit1: TMenuItem;
    SimpanInformasiKeDatabasel: TMenuItem;
    OpenPictureDialog1: TOpenPictureDialog;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    GroupBox3: TGroupBox;
    Image5: TImage;
    Label4: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Label5: TLabel;
    Label6: TLabel;
    Image6: TImage;
    Label7: TLabel;
    Label8: TLabel;
    Image7: TImage;
    Label9: TLabel;
    Label10: TLabel;
    Edit3: TEdit;
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    DBNavigator1: TDBNavigator;
    GroupBox4: TGroupBox;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Edit8: TEdit;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    DBGrid1: TDBGrid;
    Button4: TButton;
    ADOConnection1: TADOConnection;
    Edit4: TEdit;
    procedure analisa;
    procedure Thin(image:TBitmap;m2,m1:Integer);
    procedure Bwotsu;
    procedure Greyfinger;
  end;
end;
```

```

    procedure AmbilGambarSidikjarilClick(Sender: TObject);
    procedure AmbilCiriSidikjarilClick(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure SimpanInformasiKeDatabase1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;
    img1:array[0..99,0..99] of Integer;
    img2:array[0..99,0..99] of Integer;

implementation

{$R *.dfm}
//proses analisa
procedure TForm1.analisa;
var i,j,a,h,epc,bc,isl,b,trans,x1,y1,x2,y2,ecount:integer;
    ep:array [1..9] of Integer;
    ec:array [1..9] of Integer;
    ei:array [1..9] of Integer;
    img:TBitmap;
begin
    Image5.Picture.Bitmap:=Image4.Picture.Bitmap;
    //ekstraksi fingre base
    img:=Image5.Picture.Bitmap;

    //endpoint finger minutae
    epc:=0;
    for i:=1 to img.Height do
    begin
        for j:=1 to img.Width do
        begin
            ep[1]:=img.Canvas.Pixels[j+1,i+1];
            ecount:=0;
            if ep[1]=clBlack then
            begin
                ep[2]:=img.Canvas.Pixels[j+1,i];
                ep[3]:=img.Canvas.Pixels[j+2,i];
                ep[4]:=img.Canvas.Pixels[j+2,i+1];
                ep[5]:=img.Canvas.Pixels[j+2,i+2];
                ep[6]:=img.Canvas.Pixels[j+1,i+2];
                ep[7]:=img.Canvas.Pixels[j,i+2];
                ep[8]:=img.Canvas.Pixels[j,i+1];
                ep[9]:=img.Canvas.Pixels[j,i];
            end;

            h:=0;
            for a:=1 to 9 do
            begin
                if ep[a]=clBlack then inc(h);
            end;

            if (h=2) and (ep[1]=clBlack) then
            begin
                inc(epc);
            end;
        end;
    end;
    Edit1.Text:=IntToStr(epc);
    Image5.Refresh;
//=====

```

```

Image6.Picture.Bitmap:=Image4.Picture.Bitmap;
//ekstraksi fingre base
img:=Image6.Picture.Bitmap;

//bifurcation finger minutae
bc:=0; i:=0; j:=0;

for i:=1 to img.Height do
begin
for j:=1 to img.Width do
begin

ec[1]:=img.Canvas.Pixels[j+1,i+1];
if ColorToString(ec[1])<>'$00FEFEFE' then
begin
ec[2]:=img.Canvas.Pixels[j+1,i];
ec[3]:=img.Canvas.Pixels[j+2,i];
ec[4]:=img.Canvas.Pixels[j+2,i+1];
ec[5]:=img.Canvas.Pixels[j+2,i+2];
ec[6]:=img.Canvas.Pixels[j+1,i+2];
ec[7]:=img.Canvas.Pixels[j,i+2];
ec[8]:=img.Canvas.Pixels[j,i+1];
ec[9]:=img.Canvas.Pixels[j,i];
end;
h:=0;
for a:=1 to 9 do
begin
if ColorToString(ec[a])<>'$00FEFEFE' then inc(h);
end;
trans:=0;
for a:=2 to 8 do
begin
if (ColorToString(ec[a])='$00FEFEFE') and
(ColorToString(ec[a+1])<>'$00FEFEFE') then
trans:=trans+1;
end;
if (ColorToString(ec[9])='$00FEFEFE') and
(ColorToString(ec[2])<>'$00FEFEFE') then
trans:=trans+1;
if (trans=3) and (h>4) and (ColorToString(ec[1])<>'$00FEFEFE') then
begin
inc(bc);
// img.Canvas.Pixels[j,i]:=clRed;
end;

end;
end;
Edit2.Text:=IntToStr(bc);
Image6.Refresh;
//=====
Image7.Picture.Bitmap:=Image4.Picture.Bitmap;
//ekstraksi fingre base
img:=Image7.Picture.Bitmap;

//endpoint finger minutae
isl:=0;
for i:=1 to img.Height do
begin
for j:=1 to img.Width do
begin
ei[1]:=img.Canvas.Pixels[j+1,i+1];
ecount:=0;
if ei[1]=clBlack then
begin
ei[2]:=img.Canvas.Pixels[j+1,i];
ei[3]:=img.Canvas.Pixels[j+2,i];
ei[4]:=img.Canvas.Pixels[j+2,i+1];
ei[5]:=img.Canvas.Pixels[j+2,i+2];
ei[6]:=img.Canvas.Pixels[j+1,i+2];
ei[7]:=img.Canvas.Pixels[j,i+2];

```



```

    ei[8]:=img.Canvas.Pixels[j,i+1];
    ei[9]:=img.Canvas.Pixels[j,i];
end;

h:=0;
for a:=2 to 9 do
begin
    if ei[a]=clBlack then inc(h);
end;

if (h=0) and (ei[1]=clBlack) then
begin
    inc(isl);
end;
end;
Edit3.Text:=IntToStr(isl);
Image7.Refresh;

end;

//proses thinning
procedure TForm1.Thin(image:TBitmap;m2,m1:Integer);
var k,l,i,j,count,trans,m,ok:Integer;
    cek2:Boolean;
    y:array[0..7]of Integer;
begin
    //read picture
    for k:=0 to m2 do
    begin
        for l:=0 to m1 do
        begin
            if image.Canvas.Pixels[l,k]=clBlack then
            begin
                img1[l,k]:=0;
                img2[l,k]:=0;
            end
            else
            begin
                img1[l,k]:=1;
                img2[l,k]:=1;
            end;
        end;
    end;
    //proses thinning
    ok:=0;
    while ok=0 do
    begin
        ok:=1;
        //iterasi 1
        for k:=0 to m2 do
        begin
            for l:=0 to m1 do
            begin
                if img1[l,k]=0 then
                begin
                    count:=0;
                    for j:=-1 to 1 do
                    begin
                        for i:=-1 to 1 do
                        begin
                            if img1[l+i,k+j]=0 then count:=count+1;
                        end;
                    end;
                    if (count>2) and (count<7) then
                    begin
                        y[0]:=img1[l-1,k-1];
                        y[1]:=img1[l,k-1];
                        y[2]:=img1[l+1,k-1];
                        y[3]:=img1[l+1,k];
                    end;
                end;
            end;
        end;
    end;
end;

```

```

y[4]:=img1[l+1,k+1];
y[5]:=img1[l,k+1];
y[6]:=img1[l-1,k+1];
y[7]:=img1[l-1,k];
trans:=0;
for m:=0 to 6 do
begin
if (y[m]=1) and (y[m+1]=0)then trans:=trans+1;
end;
if (y[7]=1) and (y[0]=0) then trans:=trans+1;
if (trans=1) and ((count>2) and (count<7)) then
begin
img2[l,k]:=1; ok:=0;
end;
end;
end;
end;
end;
//copy img2 to img1
for k:=0 to m2 do
begin
for l:=0 to m1 do
begin
img1[l,k]:=img2[l,k];
end;
end;
end;
//end while
//gambar ke image1
for k:=0 to m2 do
begin
for l:=0 to m1 do
begin
if img1[l,k]=1 then
Image4.Picture.Bitmap.Canvas.Pixels[l,k]:= clWhite
else
Image4.Picture.Bitmap.Canvas.Pixels[l,k]:= clBlack;
end;
end;
end;
end;

//proses otsu
procedure TForm1.Bwotsu;
const level=255;
var
px:array[0..99,0..99] of TColor;
histogram: array[0..255] of integer;
PH: PByteArray;
area: Word;
TotalMean, Variance, maxVariance, zerothCumumoment, firstCumumoment : real;
threshold:Byte;
i,j,kt,k: integer;
Image:TBitmap;
begin
Image3.Picture:=Image2.Picture;
Image:=Image3.Picture.Bitmap;
if Image.PixelFormat=pf8bit then
begin
for i:=0 to level do
begin
histogram[i]:=0;
end;
for i:=0 to Image.Height-1 do
begin
PH:=Image.ScanLine[i];
for j:= 0 to Image.Width-1 do
begin
inc(histogram[PH[j]]);
end;
end;
end;
end;
end;

```

```

//compute otsu method
threshold:=0;
totalMean := 0;
maxVariance := 0;
firstCumumoment := 0;
zerothCumumoment := 0;
area := Image.Height * Image.Width;
for k:= 0 to level do
  TotalMean := TotalMean + (k * histogram[k] / area);
  for k:= 0 to level do
    begin
      zerothCumumoment := zerothCumumoment + histogram[k] / area;
      firstCumumoment := firstCumumoment + (k * histogram[k] / area);
      variance := totalMean * zerothCumumoment - firstCumumoment;
      variance := variance * variance;
      if ((zerothCumumoment <> 0) and (zerothCumumoment <> 1)) then
        begin
          variance := variance / (zerothCumumoment * (1 - zerothCumumoment));
          if (maxVariance < variance) then
            begin
              maxVariance := variance;
              threshold := k;
            end;
          end;
        end;
      end;
    for i:=0 to Image.Height-1 do begin
      for j:=0 to Image.Width-1 do begin
        kt:= round(threshold*(clWhite div 255));
        if Image.Canvas.Pixels[j,i] < kt then
          px[j,i]:= clBlack
        else
          px[j,i]:= clWhite;
        end;
      end;
    for i:=0 to Image.Height-1 do begin
      for j:=0 to Image.Width-1 do begin
        Image3.Picture.Bitmap.Canvas.Pixels[j,i]:=px[j,i];
      end;
    end;
    Image3.Refresh;
  end
  else
    ShowMessage('format salah');
end;

//== proses graysalling
procedure TForm1.Greyfingger;
type LogPal=record
  lpal:TLogPalette;
  entry:array[0..255] of TPaletteEntry;
end;

var
  paletKeabuan:LogPal;
  i,j : integer;
  PC,PH : PByteArray;
begin
  //atur palette
  paletKeabuan.lpal.palVersion:=$300;
  paletKeabuan.lpal.palNumEntries:=256;
  for i:=0 to 255 do
    begin
      paletKeabuan.entry[i].peRed:=i;
      paletKeabuan.entry[i].peGreen:=i;
      paletKeabuan.entry[i].peBlue:=i;
    end;
  //ubah ke grey
  Image2.Picture.Bitmap.PixelFormat:=pf8bit;
  Image2.Picture.Bitmap.Palette:=CreatePalette(paletKeabuan.lpal);
  for i:=0 to Image1.Picture.Height-1 do
    begin

```

```

PC:=Image1.Picture.Bitmap.ScanLine[i];
PH:=Image2.Picture.Bitmap.ScanLine[i];
for j:=0 to Image2.Picture.Width-1 do
begin
  PH[j]:=round((PC[3*j]+PC[3*j+1]+PC[3*j+2])/3);
end;
end;
Image2.Repaint;
Image2.Picture.Bitmap.PixelFormat:=pf8bit;
end;

//== buka file gambar sidikjari
procedure TForm1.AmbilGambarSidikjarilClick(Sender: TObject);
var namafile:string;
begin
  namafile:=OpenPictureDialog1.FileName;
  if OpenPictureDialog1.Execute then
  begin
    Image1.Picture:=TPicture.Create;
    Image1.Picture.LoadFromFile(OpenPictureDialog1.FileName );
  end;
end;

procedure TForm1.AmbilCiriSidikjarilClick(Sender: TObject);
begin
  //ubah rgb ke grayscale
  Image2.Picture:=Image1.Picture;
  Greyfinger;
  //ubah gray ke WB dengan otsu
  Bwotsu;
  //ubah gambar menjadi skeleton
  Image4.Picture:=Image3.Picture;
  Thin(Image4.Picture.Bitmap,100,100);
  analisa;

end;

procedure TForm1.Button4Click(Sender: TObject);
begin
  if Button4.Caption='ACTIVATED' then
  begin
    DBNavigator1.Enabled:=True;
    Button4.Caption:='INACTIVATED';
  end
  else
  begin
    DBNavigator1.Enabled:=False;
    Button4.Caption:='ACTIVATED';
  end;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  Edit4.ReadOnly:=False;
  Edit5.ReadOnly:=False;
  Edit4.SetFocus;
  Edit6.Text:=Edit1.Text;
  Edit7.Text:=Edit2.Text;
  Edit8.Text:=Edit3.Text;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  Edit4.Text:='';
  Edit5.Text:='';
  Edit6.Text:='';
  Edit7.Text:='';
  Edit8.Text:='';
  Edit4.ReadOnly:=True;

```

```
Edit5.ReadOnly:=True;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
ADOTable1.Open;
ADOTable1.Append;
ADOTable1.FieldByName('nip').AsString:=Edit4.Text;
ADOTable1.FieldByName('nama').AsString:=Edit5.Text;
ADOTable1.FieldByName('endpoint').AsString:=Edit6.Text;
ADOTable1.FieldByName('bifurcation').AsString:=Edit7.Text;
ADOTable1.FieldByName('point').AsString:=Edit8.Text;
ADOTable1.Post;

end;

procedure TForm1.SimpanInformasiKeDatabase1Click(Sender: TObject);
begin
GroupBox4.Enabled:=True;
end;

end.
```

Created with



nitro PDF[®]
created with

professional

download the free trial online at nitropdf.com/professional

download the free trial online at nitropdf.com/professional