

**VERIFIKASI CITRA SIDIK JARI POIN MINUTIAE  
DALAM VISUM ET REPERTUM (VER)  
MENGUNAKAN K-MEANS CLUSTERING**

**Andika Budi Pratama<sup>1</sup>**

Program Studi Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Brawijaya  
Email : andikabudi@gmail.com

**Abstrak**

Verifikasi sidik jari adalah proses pencocokan sidik jari. Peranan verifikasi ini salah satunya mendukung proses *visum et repertum* (VER) di bidang kedokteran forensik. Verifikasi yang sudah ada berdasarkan algoritma *minutiae*, tetapi ditemukan permasalahan terutama pada poin *minutiae* (*ridge*) yang belum bisa diproses secara lengkap. Kelemahan ini akan tampak pada proses pencocokan sidik jari yang mengandung perbedaan jumlah poin *minutiae* dua sidik jari yang berkorespondensi yang banyak ditemui pada sidik jari korban dalam proses *visum et repertum* (VER). Pada penelitian ini akan dihasilkan model perangkat lunak verifikasi citra sidik jari poin *minutiae* yang dapat mengatasi kelemahan tersebut.

Algoritma yang digunakan adalah *k-means clustering*. Uji coba diterapkan pada 5 orang. Masing-masing orang diambil 1 sampel sidik jari sebanyak 6 kali, pada sidik jari yang sama dengan posisi atau perlakuan yang berbeda. Pengujian hasil menggunakan metode statistika, yaitu analisis ragam dua arah (*Two Way ANOVA*) dan statistika deskriptif.

Hasil pengujian menunjukkan bahwa algoritma *k-means clustering* dapat diimplementasikan dalam model perangkat lunak untuk verifikasi citra sidik jari poin *minutiae*. Persentase keakuratan verifikasi citra sidik jari sangat dipengaruhi oleh pemilihan jenis *minutiae* dan tidak dipengaruhi oleh nilai *threshold*. Verifikasi yang menggunakan jenis *minutiae ridge all* dan *ridge bifurcation* akan menghasilkan persentase keakuratan terbaik, yaitu dengan mean (rata-rata) dan median (nilai tengah) sebesar 100%

**Kata kunci :** sidik jari, *visum et repertum*, *k-means clustering*

### **1.1 Latar Belakang**

Sidik jari merupakan salah satu teknologi yang dapat digunakan dalam mengidentifikasi seseorang. Bahkan sidik jari menjadi teknologi yang dirasa cukup handal, karena terbukti relatif akurat, aman, mudah, dan nyaman untuk dipakai sebagai identifikasi bila dibandingkan dengan sistem *biometric* yang lainnya seperti retina mata atau DNA.

Penerapan teknologi sidik jari ini tidak hanya pada sistem absensi pegawai perusahaan, tetapi juga berkembang di bidang kedokteran forensik, yaitu pada proses *visum et repertum* (VER). VER merupakan laporan tertulis dokter untuk memberikan keterangan untuk kepentingan peradilan. Salah satu tahap VER adalah verifikasi sidik jari. Verifikasi ini dilakukan untuk mengetahui identitas seseorang terhadap suatu masalah pidana atau perdata, contohnya kasus korban kecelakaan, korban tenggelam, dan lain-lain.

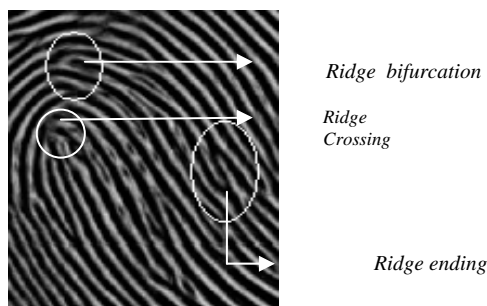
Verifikasi sidik jari ini menggunakan metode *minutiae based matching* dengan dua *template* sidik jari. Pertama *template* sidik jari di *data storage* (kepolisian) dan kedua, *template* sidik jari sampel di TKP (Tempat Kejadian Perkara). Dalam penggunaannya metode *minutiae based matching* ditemukan beberapa kelemahan. Salah satu kelemahan itu adalah kemungkinan tidak dapat diprosesnya citra sidik jari. Ada dua penyebab hal tersebut dapat terjadi; yaitu hilang atau bertambahnya poin-poin *minutiae* dalam pemrosesan citra digital pada salah satu citra atau dapat disebabkan perbedaan jumlah poin *minutiae* kedua *template* citra. Padahal pada proses *visum et repertum* itu sendiri terdapat kemungkinan perubahan fisik pada tubuh korban atau pengambilan sidik jari yang kurang sempurna di tempat kejadian perkara (TKP) oleh penyidik kepolisian.

Untuk mengatasi kelemahan pemrosesan sidik jari tersebut dilakukan percobaan dengan melakukan pengklasteran titik yang

berkorespondensi antara kedua *template*. Dilakukanlah perancangan model perangkat lunak untuk verifikasi citra sidik jari poin *minutiae* dalam visum et repertum ini. Tujuannya tidak lain tidak bukan mengetahui hasil uji coba terhadap metode yang nantinya akan dipakai dan sejauh mana metode yang digunakan dapat digunakan untuk mengatasi kelemahan algoritma *minutiae based matching*. Sehingga diharapkan pada akhirnya nanti dapat diciptakan sebuah model perangkat lunak yang mampu melakukan verifikasi citra sidik jari poin *minutiae* identitas seseorang, sehingga dapat mendukung proses *visum et repertum* (VER) bagi bidang kedokteran forensik.

## 2. Verifikasi Citra Sidik Jari Poin *Minutiae*

Poin *Minutiae* adalah sejenis titik yang terbentuk pada sidik jari. Ada beberapa jenis *minutiae* atau dapat juga disebut dengan *ridge*, antara lain *ridge ending* (akhir), *ridge crossing* (persilangan), dan Fitur kecil yang terbentuk dari percabangan *ridge* pada sidik jari disebut *ridge bifurcation* (Jain, dkk., 1999). Pada Gambar 2.1 ditunjukkan bentuk dari *minutiae* sidik jari.



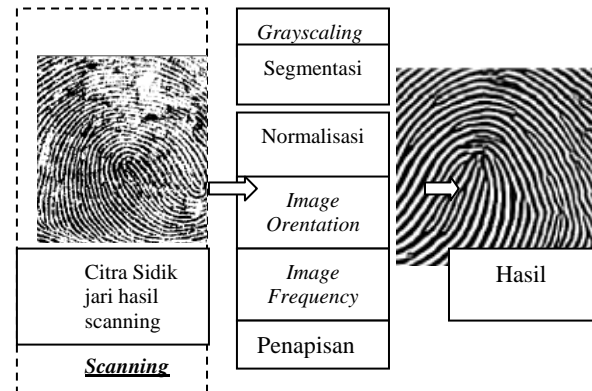
Gambar 1. Bentuk *Ridge* pada sidik jari

### 2.1 Proses Verifikasi

Verifikasi merupakan proses pencocokan sejenis dengan identifikasi hanya saja pada proses verifikasi, sidik jari dicocokkan satu-satu dimana setiap sidik jari masukan diperbandingkan dengan satu *template* sidik jari tertentu yang tersimpan sebelumnya. Keluaran dari program ini adalah keputusan apakah proses verifikasi berhasil (*valid*) atau gagal (*invalid*)

### 2.2 Perbaikan Citra (*Image Enhancement*)

Tahap pertama adalah pemrosesan citra sidik jari (*fingerprint enhancement*). Pada tahap ini citra sidik jari hasil *scanning* akan ditingkatkan kualitasnya melalui beberapa proses



Gambar 2. Proses Perbaikan citra

Proses *Grayscale* dilakukan dengan mengkonversi citra warna (*true color*) menjadi citra hitam putih dengan merata-rata nilai ketiga elemen warna (RGB) setiap piksel. *Segmentasi* merupakan proses untuk memisahkan obyek pada suatu citra dari daerah latar belakangnya. Setelah itu citra yang disegmentasi dinormalisasi dengan menstandarisasi nilai intensitas suatu citra dengan menyesuaikan cakupan derajat keabuan sehingga berada pada cakupan nilai yang diharapkan. Proses *Image Orentation* dan *Image Frequency* digunakan untuk proses penapisan citra sidik jari. Penapisan yang digunakan adalah *Penapisan Gabor*.

$$G(x, y; \theta, f) = \exp \left\{ -\frac{1}{2} \left[ \frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2} \right] \right\} \cos(2\pi f x_{\theta}) \quad (2.1)$$

dimana:

$$x_{\theta} = x \cos \theta + y \sin \theta, \quad (2.2)$$

$$y_{\theta} = -x \sin \theta + y \cos \theta, \quad (2.3)$$

$$E(i, j) = \sum_{u=-\frac{w_x}{2}}^{\frac{w_x}{2}} \sum_{v=-\frac{w_y}{2}}^{\frac{w_y}{2}} G(u, v; \theta(i, j), F(i, j)) N(i-u, j-v) \quad (2.4)$$

### 2.3 Ekstraksi *Minutiae* (*Image Extraction*)

Ada tiga Tahap dalam *Image Extraction* ini, diantaranya: *Binerisasi*, *Penipisan pola* (*skeletonization*), dan *deteksi minutiae*. Konversi citra pada proses binerisasi dilakukan dengan operasi pengambungan (*thresholding*) sehingga didapatkan keberadaan obyek berupa alur guratan sidik jari. *Penipisan pola* bertujuan mengurangi bagian yang tidak perlu (*redundant*). Kemudian citra hasil penipisan di deteksi *minutiae* menggunakan metode *crossing number* (CN). Poin *minutiae* dideteksi dengan memindai tetangga lokal pada masing-masing piksel *ridge* pada citra menggunakan ukuran *window* 3 x 3.

Kemudian nilai CN dihitung, yang didefinisikan sebagai separuh penjumlahan dari perbedaan antara pasangan-pasangan piksel yang bersebelahan pada *eight-neighbourhood*.

Tabel 1. Properti *Crossing Number*

CN	Property
0	Isolated Point
1	Ridge ending point
2	Continuing ridge point
3	Bifurcation point
4	Crossing point

Nilai CN pada *ridge* piksel  $P$  didapat dari Persamaan 2.16.

$$CN = 0.5 \sum_{t=1}^8 |P_t - P_{t+1}| \quad (2.5)$$

$$P_9 = P_1 \quad (2.6)$$

dimana  $P_i$  merupakan nilai tetangga dari  $P$ . Untuk suatu piksel  $P$ , kedelapan piksel tetangganya diperiksa dengan arah berlawanan jarum jam seperti pada Gambar 2.10.

$P_4$	$P_3$	$P_2$
$P_5$	$P$	$P_1$
$P_6$	$P_7$	$P_8$

Gambar 3. Pemberian nomor pada 8-tetangga piksel  $P$  untuk proses *minutiae detection*

#### 2.4 Visum Et Repertum (VER)

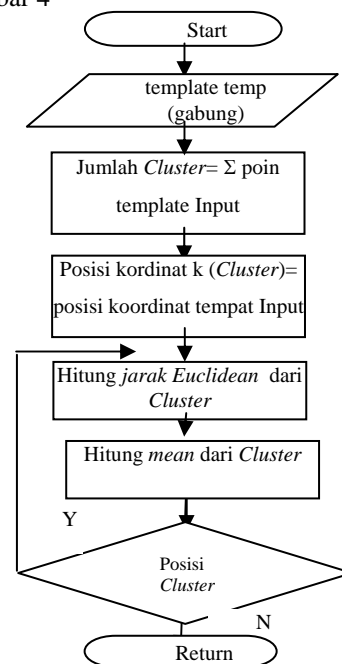
Istilah *Visum et repertum* ini dapat ditemukan dalam lembaran Negara tahun 1937 Nomor : 350 Pasal I yang terjemahannya adalah *visum et repertum* pada Dokter yang dibuat baik atas sumpah Dokter yang diucapkan pada waktu menyelesaikan pelajarannya di Negeri Belanda atau Indonesia, maupun atas sumpah khusus seperti tercantum dalam pasal 2, mempunyai daya bukti yang syah dalam perkara pidana selama *visa et Reperta* tersebut berisi keterangan mengenai hal hal yang diamati oleh Dokter itu pada benda-benda yang diperiksa.

Penegak hukum mengartikan *visum et repertum* sebagai laporan tertulis yang dibuat dokter berdasarkan sumpah atas permintaan yang berwajib untuk kepentingan peradilan tentang segala hal yang dilihat dan ditemukan menurut pengetahuan yang sebaik-baiknya.

### 3. K-MEANS CLUSTERING

Algoritma *K-Means clustering* merupakan algoritma yang digunakan untuk melakukan pencocokan dua *template* sidik jari yang telah digabung. Adanya pengklasteran titik titik dalam proses *k-means clustering* ini diharapkan dapat mengatasi kelemahan algoritma *minutiae based* yakni, tidak dapat diprosesnya pencocokan karena perbedaan jumlah *minutiae* dalam *visum et repertum*.

Proses pengklasteran poin *minutiae* tampak pada Gambar 4



Gambar 4. Diagram Alir Proses *K-means Clustering*

Berikut merupakan langkah-langkah dari *K-means clustering*

1. Menentukan  $K$  titik yang merepresentasikan obyek (*centroid*).
2. *repeat*
3. Menetapkan setiap obyek pada kelompok dengan posisi *centroid* terdekat berdasarkan nilai *mean* dari *object Cluster*, menggunakan Rumus *Euclidean*
4. Update *mean cluster*, Hitung Nilai *mean* masing-masing *cluster*
5. *until* posisi *cluster* tidak berubah (Konvergen)

### 3.1 Nilai Kecocokan (*matching score*)

Adapun langkah-langkah untuk menentukan nilai kecocokan (*matching score*) berdasarkan percobaan adalah:

1. Menghitung *Alone cluster* (AC) dengan memeriksa apakah jumlah poin *minutiae* Gabung (TotMin) lebih besar dari pada dua kali jumlah *minutiae* k (*cluster*).

If  $2k \geq \text{TotMin}$  then

$$AC \leftarrow (\text{TotMin} - 2k) * \text{Threshold}$$

Else

$$AC \leftarrow 0;$$

2. Menentukan nilai *fitness* setiap klasternya

$$\text{fitness} = \left( 1 - \frac{1}{\text{Threshold}} \times \left( \sum_{k=1}^m (E(T_{1,2})) + \frac{AC}{n} \right) \right) \cdot \left( \frac{\text{bataslooping} - \text{loopingstate}}{\text{bataslooping} - 2} \right) \quad (3.1)$$

Pada Persamaan 3.1 ditunjukkan bahwa E adalah jarak *euclidean* dua poin *minutiae* terdekat,  $k =$  Poin *minutiae* dalam satu *cluster*,  $T =$  template gabungan *minutiae*, *threshold* adalah ambang jarak tertentu antara pusat *cluster* dengan *cluster* yang bersesuaian dalam satu *cluster*,  $m$  adalah jumlah titik *cluster* gabungan, dan hasil *looping* merupakan titik henti (konvergen *state*) ketika *cluster* mencapai konvergen.

3. Menentukan nilai *matching score*. Setiap *cluster* yang terbentuk mempunyai nilai *fitness*, maka dilakukan rata-rata (*mean*). Jumlah jarak terdekat tiap *cluster* dibagi dengan jumlah *cluster* ( $n$ ), tampak pada Persamaan 3.2

Matching Score =  $\bar{f}$

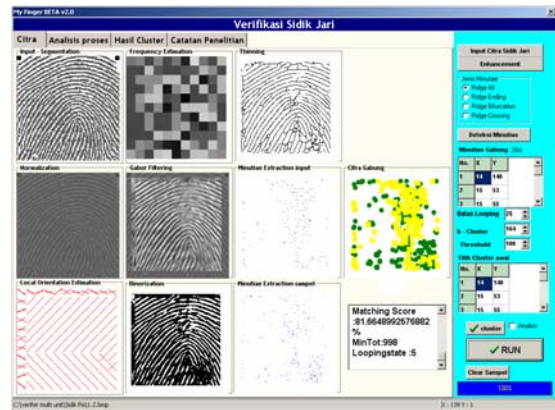
$$\begin{aligned} &= \frac{f_{\text{cluster}(1)} + f_{\text{cluster}(2)} + f_{\text{cluster}(3)} + \dots + f_{\text{cluster}(n)}}{n} \\ &= \left( \frac{\left( 1 - \frac{(E_1 + E_2)}{\text{Threshold}} + \dots + \left( 1 - \frac{(E_{m-1} + E_m)}{\text{Threshold}} \right) \right)}{n} \right) \\ &\quad \left( \frac{\text{bataslooping} - \text{loopingstate}}{\text{bataslooping} - 2} \right) \\ &= \left( \frac{\sum_{k=1}^m (E(T_{1,2}) + AC)}{n} \right) \cdot \left( \frac{\text{bataslooping} - \text{loopingstate}}{\text{bataslooping} - 2} \right) \quad (3.2) \end{aligned}$$

### 3.2 Persentase Keakuratan Verifikasi (*acc*)

Persentase keakuratan verifikasi didapatkan dari jumlah verifikasi yang berhasil dalam suatu percobaan dengan perlakuan sampel tertentu. Verifikasi dikatakan berhasil (*valid*), apabila nilai kecocokan (*matching score*) antara identitas obyek sampel dengan identitas seseorang di data storage menunjukkan nilai rata-rata terbesar diantara sampel yang lain

## 4. Hasil dan Pembahasan

Perancangan *interface* model perangkat lunak pada ditunjukkan pada Gambar 5



Gambar 5. Tampilan hasil deteksi *minutiae* clustering

### 4.1 Analisa Hasil

Tabel 2. Persentase Keakuratan Verifikasi

		Threshold					Mean	Median
		50	100	120	200	375		
Jenis Minutiae	All	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	Ridge Ending	80.00	80.00	80.00	100.00	100.00	88.00	80.00
	Bifurcation	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	Crossing	40.00	40.00	40.00	40.00	40.00	40.00	40.00

Pada Tabel 2 ditunjukkan persentase keakuratan verifikasi berdasarkan jenis *minutiae* dan *threshold*. Penggunaan nilai *threshold* yang semakin besar dalam verifikasi, akan memperbesar persentase keakuratan verifikasi, seperti yang terlihat pada baris tabel 4.1 jenis *minutiae* *ridge ending*. Percobaan dengan *threshold* 50, 100, dan 120 keakuratannya sebesar 80%, tetapi dengan nilai *threshold* 200 dan 375 persentase keakuratannya meningkat sebesar 20% menjadi 100%. Verifikasi dengan jenis *minutiae* *crossing* menghasilkan persentase keakuratan yang paling kecil diantara jenis *minutiae* yang lain, sebesar 40%. Hal ini berarti setiap dilakukan 10 verifikasi, akan terdapat sedikitnya empat verifikasi yang gagal (*invalid*). Hasil persentase keakuratan terbaik terjadi pada *ridge all* dan *ridge ending*, menghasilkan *mean* (rata-rata) dan median (nilai tengah) dengan keakuratan verifikasi sebesar 100%.

Selanjutnya digunakan pendekatan statistika menggunakan analisis ragam dua-arah (Two-Way ANOVA). Metode ANOVA ini dapat digunakan untuk mengetahui pengaruh variabel-variabel yang ada di dalam hasil uji coba, antara lain variabel; jenis *minutiae*, *threshold* ,dan persentase keakuratan verifikasi. Pengujian ini menggunakan *software* MINITAB 13, yang hasilnya tampak pada Gambar 6

Factorial Design							
General Factorial Design							
Factors:	2	Factor Levels:	4, 5				
Runs:	20	Replicates:	1				
Two-way ANOVA: acc(%) versus Jenis Minutiae, Threshold							
Analysis of Variance for acc(%)							
Factor	Type	Levels Values					
Jenis Mi	fixed	4	1	2	3	4	
Threshold	fixed	5	50	100	120	200	375
Analysis of Variance for acc(%), using Adjusted SS for Tests							
Source	DF	Seq SS	Adj SS	Adj MS	F	P	
Jenis Mi	3	12240.0	12240.0	4080.0	136.00	0.000	
Threshold	4	120.0	120.0	30.0	1.00	0.445	
Error	12	360.0	360.0	30.0			
Total	19	12720.0					

Gambar 6. Pengujian analisis ragam 2-arah

Uji Hipotesis yang pertama adalah memeriksa pengaruh *Threshold*. Hipotesis:

$$H_0 = T_1 = T_2 = T_3 = T_4 = T_5 = 0$$

$$H_1 = \text{paling sedikit satu } T_i \neq 0 \text{ } i=1,2,3,4,5$$

(Perbedaan nilai *threshold* berpengaruh terhadap persentase keakuratan sidik jari)

Selama p-value (nilai *significantly*) kurang dari  $\alpha$ , maka Tolak  $H_0$ . Pada Gambar 6, statistik *threshold* adalah 0,445, dengan taraf nyata ( $\alpha$ ) 0,01 atau 0,05 hipotesis  $H_0$  diterima. Artinya tidak perbedaan yang cukup signifikan dari perbedaan nilai *threshold* terhadap hasil persentase keakuratan verifikasi. Selanjutnya yang kedua, diperiksa pengaruh jenis *minutiae* dalam proses *clustering* terhadap persentase keakuratan. Hipotesis yang kedua yang diuji:

$$H_0 = M_1 = M_2 = M_3 = M_4 = M_5 = 0$$

$$H_1 = \text{paling sedikit satu } M_i \neq 0 \text{ } i=1,2,3,4,5$$

(Perbedaan Jenis *minutiae* berpengaruh terhadap persentase keakuratan sidik jari)

Selama p-value (nilai *significantly*) kurang dari  $\alpha$ , maka Tolak  $H_0$ . Pada Gambar 6., statistik jenis *minutiae* adalah 0,00, dengan taraf nyata ( $\alpha$ ) 0,01 atau 0,05 hipotesis  $H_0$  ditolak. Artinya ada perbedaan yang cukup signifikan dari penggunaan jenis *minutiae* pada proses *clustering* dalam menentukan persentase keakuratan kedua sidik jari.

#### 4.1 Kesimpulan

Dari hasil pembahasan dan analisis yang telah dilakukan pada bab sebelumnya, dapat disimpulkan bahwa:

1. Telah berhasil dibuat model verifikasi citra sidik jari poin *minutiae* yang tersimulasikan dalam perangkat lunak yang telah dibangun. Perangkat lunak tersebut dirancang dalam *visum et repertum* (VER) menggunakan *k-means clustering*. Secara keseluruhan perangkat lunak dibangun menggunakan bahasa pemrograman *delphi*.
2. *K-means clustering* dapat diimplementasikan dalam model perangkat lunak untuk verifikasi citra sidik jari poin *minutiae*. Karena mampu melakukan pengklasteran poin *minutiae* antara *template* masukan dengan *template* sampel, sehingga kelemahan algoritma *minutiae* dalam *visum et repertum* berupa perbedaan jumlah poin *minutiae* dua sidik jari dapat diatasi
3. Persentase keakuratan verifikasi citra sidik jari poin *minutiae* dalam *visum et repertum* tidak dipengaruhi perbedaan nilai *threshold*, tetapi dipengaruhi pemilihan jenis *minutiae*. Penggunaan jenis *minutiae ridge crossing* menghasilkan keakuratan verifikasi hanya sebesar 40%. Hal ini disebabkan tidak terbentuknya *ridge* pada sidik jari, sehingga *k-means clustering* tidak dapat dilakukan. Sedangkan verifikasi yang menggunakan jenis *minutiae ridge all* dan *ridge bifurcation* akan menghasilkan persentase keakuratan verifikasi terbaik, dengan median (nilai tengah) sebesar 100%.

#### REFERENSI

- [1] Achmad,B dan Firdausy,K. 2005. *Teknik Pengolahan Citra Digital Menggunakan Delphi*. Ardhi Publishing. Yogyakarta
- [2] Agusta, Y.2007. *K-Means-Penerapan, Permasalahan dan Metode Terkait*. Jurnal Sistem dan Informatika. 3:47-60
- [3] Ananta, A; Haruyani, H; Wulansari, I; Azhari,R; Kemala,R. 2003. *Penggunaan Metode Thinning Pada Pemrosesan Gambar*. Fak.Ilm Komputer,UI. Jakarta
- [4] Atmodiriono,H dan Hamdani,N. 1980. *Visum et Repertum dan Pelaksanaannya*. Edisi I., Airlangga University Press. Surabaya

- [5] Chamma,N. *Fingerprint Image Enhancement and Minutiae Extraction*. Dept of Electrical and Computer Engineering Clemson University
- [6] Han,J and Khamber,M. 2006. *Data Mining: Concept and Techniques*. Second Ed.,Morgan Kaufmann Publishers. USA
- [7] Handhani,O. 2007. *Pendeteksian Poin Poin Minutiae Pada Citra Sidik Jari Dengan Berbasis Pada Pemrosesan Citra Digital*. Fakultas MIPA,Ilmu Komputer UB. Malang
- [8][http://home.dei.polimi.it/matteucc/Clustering/tutorial\\_html/kmeans.html](http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html) , tanggal akses 1 Maret 2008
- [9] <http://fkui.org> , tanggal akses 1 Maret 2008
- [10] Iqbal,A.M. *Implementasi Dan Analisis Performansi Authentikasi Sistem Biometrik Sidik Jari*. Departemen Teknik Elektro,ITB. Bandung
- [11] Jain,L.C; Halici,U; Hayashi,I; Lee,S.B and Tsutsui,S. 1999. *Intelligent Biometric Techniques in Fingerprint and Face Recognition*. CRC Press. United States of America
- [12] Jea, T. *Minutiae Based partial Fingerprint Recognition*. Departement Of Computer Science Univeristy at Buffalo, the state University of New York
- [13] Kwon, Dongjin; Yun, Il Dong; Kim, Duck Hoon; Lee, Sang Uk. *Fingerprint Matching Method Using Minutiae Clustering and Warping*. Hankuk Univeristy of Foreign Study,Yongin. Korea
- [14] Meidiyanto,T. 2002. *Aplikasi FPGA Sebagai Pendeteksi Sidik Jari dengan Menggunakan Algoritma Gabor Filter*. Jurusan Teknik Elektro,ITS. Surabaya
- [15] Munir,Rinaldi. 2004. *Algoritma dan Pemrograman dalam bahasa Pascal dan C*. Informatika:Bandung
- [16] Neel, Julian. 2005. *Cluster Analysis Methods for Speech Recognition*. Royal Institute of Technology Stockholm
- [17] Simpson,K. 1979. *Forensic Medicine*. Eight Ed.,Edward Arnlod Publishers Ltd. London
- [18] Thai,R. 2003. *Fingerprint Image Enhancement and Minutiae Extraction*. The University Western Australia. Australia
- [19] Zhao, De Qun. 2006. *Fingerprint Registration Using Minutia Clusters and Centroid Structure I*. Beijing University of Posts and Telecommunications. China