PENGEMBANGAN ALGORITMA *MATCHING* UNTUK AUTENTIKASI SIDIK JARI PADA SMARTCARD

Agus Bejo, Arfianto Dwi Cahyo, Risanuri Hidayat

Departemen Teknik Elektro dan Teknologi Informasi, Faklutas Teknik, Universitas Gadjah Mada

Abstract*— Fingerprint is a biometric authentication system which is commonly used because of the high accuracy and low-cost implementation. Fingerprint authentication consist of two parts: feature extraction and classification. The classification process is used for mathing two fingerprint data. Therefore, the classification is also called matching. There are many matching techniques. One of them is hough transform technique. The hough transfrom technique is capable of resulting good accuracy, however it requires heavy computational cost. This becomes a problem when it is implemented on embedded systems such as smart cards that have limited resources. This paper proposes a modification of hough transform technique by reducing the amount of fingerprint minutiae data in order to reduce the computation. In addition, the proposed technique also performs computation sharing in which some of the heavy computation part is done by the host computer and the rest is done on the smart card. Experimental results show that the proposed technique is capable of improving the matching computation by 156% without significantly impacting the accuracy of the algorithm.

***Intisari*—Biometrik sidik jari merupakan teknik autentikasi yang banyak digunakan karena memiliki kelebihan akurasi tinggi namun biaya relatif rendah. Proses autentikasi sidik jari terdiri dari dua bagian yaitu ekstraksi ciri dan klasifikasi. Proses klasifikasi berfungsi untuk mencocokan dua buah data sidik jari sehingga sering dikenal dengan *matching*. Ada banyak teknik *matching* salah satunya adalah *hough transform*. Teknik *hough transfrom* mampu menghasilkan akurasi yang baik namun memerlukan komputasi yang berat. Komputasi yang berat ini menjadi masalah ketika diimplementasikan pada *embedded system* seperti *smart card* yang memiliki sumber daya terbatas. Pada penelitian ini diajukan sebuah modifikasi teknik *hough transform* dengan cara mereduksi jumlah data *minutiae* sidik jari agar dapat menurunkan komputasi. Selain itu teknik yang diajukan juga melakukan *sharing* komputasi dimana sebagian komputasi yang berat dikerjakan oleh *host computer* dan sisanya dikerjakan di *smart card*. Hasil eksperimen menunjukkan bahwa teknik yang diusulkan mampu meningkatkan komputasi algoritma *matching* sebesar 156% tanpa berakibat pada perubahan akurasinya secara signifikan.**

Kata Kunci*— algoritma *matching*, sidik jari, *hough transform

I. PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat juga membuat keamanan informasi tidak dapat dipisahkan dari hal itu. Keamanan informasi mempunyai fungsi untuk melindungi informasi dari usaha pencurian, penggantian dan perusakan oleh pihak-pihak yang tidak mempunyai hak akses terhadap informasi tersebut. Oleh sebab itu diperlukan autentikasi pengguna oleh sistem keamanan untuk mencegah pengaksesan oleh pengguna yang tidak berhak [1]. Metode yang digunakan untuk melakukan autentikasi ada bermacam-macam, yaitu berdasarkan apa yang user miliki (*Ownership-based)*, berdasarkan apa yang *user* ketahui (*knowledge-based*) dan berdasarkan ciri atau karakteristik biologi yang yang melekat pada tubuh seseorang yang dikenal dengan biometrik (*inherence-based*) [2].

Tren autentikasi yang sedang berkembang dan populer belakangan ini adalah autentikasi menggunakan karakteristik biometrik. Karakteristik biometrik merupakan perpaduan antara karakter fisik dan perilaku khusus manusia. Sidik jari, iris mata, bentuk wajah, geometri tangan, tanda tangan, pembuluh darah tangan atau jari dan suara merupakan contoh dari karakteristik biometrik. Karakteristik biometrik bersifat unik, artinya setiap orang memiliki karakteristik biometrik yang berbeda. Sebagai contoh, sidik jari setiap manusia tidak ada yang sama, bahkan sidik jari milik manusia yang sama tetapi jari yang berbeda juga akan berbeda. Salah satu keunggulan dari biometrik dibanding metode lain adalalah *user* tidak perlu mengingat atau membawa apapun untuk melakukan autentikasi sehingga tidak akan ada lagi alasan lupa atau ketinggalan. Keunggulan lain adalah tingkat keamanan yang lebih baik dibanding dua metode lain karena informasi biometrik *user* tidak mudah diduplikat ataupun dicuri oleh orang lain.

Biometrik sidik jari merupakan autentikasi biometrik yang berkembang pesat dan sudah banyak digunakan secara komersial. Penggunaan sidik jari dalam proses autentikasi menjadi populer karena pengaplikasiannya tidak sulit dan dari segi harga juga terjangkau [3]. Teknologi yang ada sekarang juga membuat proses pengambilan informasi sidik jari menjadi mudah. Dari segi akurasi, sidik jari bukanlah yang terbaik, namun akurasinya sudah cukup untuk beberapa aplikasi yang memerlukan tingkat keamanan tidak terlalu tinggi seperti sistem absensi kehadiran dan akses ruangan. Jika dibandingkan dengan DNA atau iris mata, sidik jari memiliki tingkat akurasi yang lebih rendah namun dari sisi biaya infrastruktur sidik jari lebih murah dan praktis dibandingkan DNA atau iris mata. Secara keseluruhan, autentikasi menggunakan biometrik sidik jari merupakan metode yang tingkat keamanannya baik, performa akurasi, waktu eksekusi, penggunaan memori, ketahanan juga baik dan biaya infrastruktur murah [4].

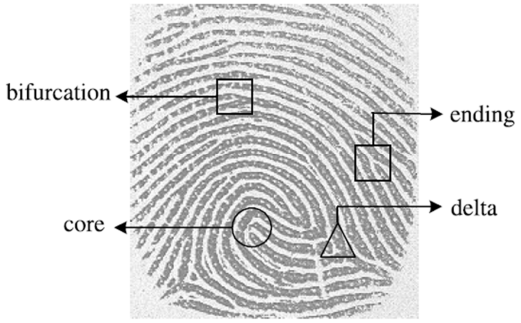
Aplikasi autentikasi sidik jari umumnya menggunakan komputer yang memiliki sumber daya besar baik kemampuan komputasi maupun memori. Tantangan implementasi autentikasi sidik jari adalah ketika diimplementasikan pada *embedded system* seperti menanamkan proses *matching* pada pada *smart card* atau sering dikenal dengan istilah *match-on-card*. Sistem ini akan lebih efektif dan aman karena data sidik jari terregistrasi akan disimpan dalam kartu tersebut dan

ketika autentikasi akan diproses langsung oleh di dalam kartu tersebut. Penelitian yang telah dilakukan selama ini kebanyakan hanya fokus pada peningkatan sisi akurasi tanpa mempertimbangkan komputasi dan kebutuhan memori yang digunakan. Padahal aplikasi *embedded system* seperti *smart card* memiliki memori yang terbatas dan komputasi yang rendah. Dengan demikian menjalankan algoritma *matching* di dalam *smart card* memerlukan waktu yang lama dan tidak dapat ditolerir oleh pengguna. Untuk mengatasi hal tersebut pada penelitian ini diajukan sebuah konsep modifikasi teknik *hough transform* yang dilakukan dengan cara reduksi data *minutiae* sidik jari dan *sharing* komputasi antara *host* komputer dan *smart card.* Tujuannya adalah agar dapat mengurangi beban komputasi algoritma *matching* pada *smart card* sehingga dapat diterima oleh pengguna.

II. ALGORITMA MATCHING HOUGH TRANSFORM

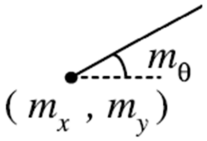
A. Data *Minutiae* Sidik Jari

Algoritma *matching* diklasifikasikan menjadi tiga yaitu: berbasis korelasi, berbasis *minutiae* dan berbasis *non-minutiae* [7]. Pada teknik berbasis korelasi, dua buah citra sidik jari saling ditumpuk dan dihitung korelasi antar piksel untuk mengukur tingkat kemiripannya. Teknik berbasis *minutiae* menghitung kemiripan dua citra sidik jari berdasarkan data *minutiae*. Teknik *non-minutiae* hampir sama dengan teknik *minutiae* hanya bedanya yang digunakan untuk menghitung kemiripan bukan data *minutiae* melainkan pola *ridge/valley* seperti orientasi, frekuensi, bentuk kontur dan tekstur [5].



Gambar 1. *Data sidik jari berbasis minutiae.*

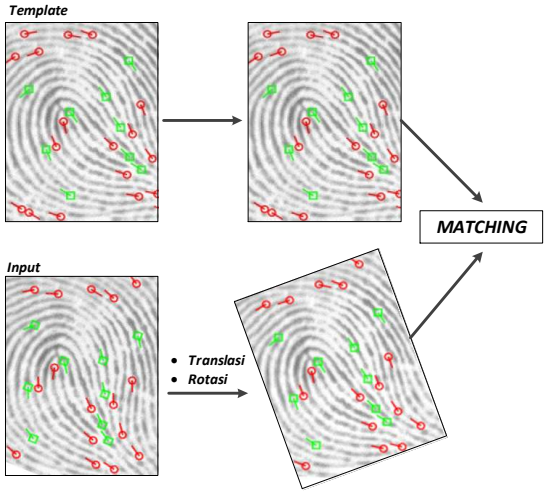
Data sidik jari yang umumnya digunakan pada algoritma berbasis *minutiae* adalah *core*, *delta*, *endpoint* dan *bifurcation* seperti ditunjukkan pada gambar 1. Data tersebut disebut dengan istilah *minutiae*. Menurut standard ISO 19794 setiap *minutiae* akan memiliki informasi berupa tipe *minutiae* (m_{type}), posisi x (m_x), posisi y (m_y) dan sudut orientasi (m_θ) seperti terlihat pada gambar 2.



Gambar 2. *Standard data minutiae menurut ISO 19794.*

B. Teknik *Matching Hough Transform*

Konsep *hough transform* adalah membandingkan data *minutiae* antara sidik jari *input* dan *template* dengan melakukan berbagai kombinasi transformasi geometri rotasi dan translasi untuk mengukur kemiripanya. Nilai kemiripan yang dihasilkan nanti akan dibandingkan dengan sebuah *treshold* tertentu. Jika nilai kemiripan melebihi *treshold* maka kedua sidik jari dinyatakan sama (*match*). Sebaliknya jika nilai kemiripan lebih rendah dari *threshold* maka kedua sidik jari dinyatakan berbeda (*unmatch*). Gambar 3 memperlihatkan proses *matching* menggunakan teknik *hough transform* berbasis data *minutiae* [6]. *T* adalah data *mintiae* sidik jari template sedangkan *I* adalah data *minutiae* sidik jari input. Data *minutiae* input *I* akan dirotasi dan translasi dengan berbagai kombinasi θ, Δx dan Δy kemudian dibandingkan dengan data *minutiae* template *T*. Setiap kombinasi θ, Δx dan Δy akan dihitung tingkat kemiripannya dengan menghasilkan nilai kemiripan atau *matching score*. Nilai kemiripan tertinggi inilah yang nantinya akan dibandingkan dengan *treshold* untuk memutuskan apakah sidik jari input sama dengan sidik jari template.



Gambar 3. *Algoritma matching hough transform.*

Secara matematis, perhitungan nilai kemiripan dijelaskan sebagai berikut. Didefinisikan *m* adalah vektor yang berisi data *minutiae* sebuah sidik jari.

$$m = (m_x, m_y, m_\theta) \qquad (1)$$

M didefinisikan sebagai kumpulan dari semua data *minutia* dari sebuah sidik jari. *Template* dilambangkan dengan *T* dan *input* dilambangkan dengan *I*. *T* dan *I* didefinisikan sebagai :

$$\boldsymbol{T} = (\boldsymbol{t_1}, \boldsymbol{t_2}, \dots, \boldsymbol{t_m} | \boldsymbol{t_i} \in M, i = 1..m) \qquad (2)$$
$$\mathbf{I} = (\boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_n} | \boldsymbol{s_j} \in M, j = 1..n) \qquad (3)$$

Didefinisikan $F_{\Delta x \Delta y \Delta \theta}$ sebagai fungsi yang memetakan vektor *minutiae* $\boldsymbol{s_j}$ menjadi $\boldsymbol{s'_j}$ menggunakan transformasi geometri. Transformasi geometri yang digunakan adalah pergeseran$(\Delta x, \Delta y)$ dan rotasi berlawanan arah jarum jam dari titik origin $(\Delta\theta)$. Selanjutnya $\boldsymbol{s'_j}$ dapat di rumuskan menjadi :

$$\begin{bmatrix} s'^{j}_{x} \\ s'^{j}_{y} \\ s'^{j}_{\theta} \end{bmatrix} = \begin{bmatrix} cos\Delta\theta & -sin\Delta\theta & 0 \\ sin\Delta\theta & cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s^{j}_{x} \\ s^{j}_{y} \\ s^{j}_{\theta} \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} \quad (4)$$

Sepasang vektor *minutia* ($\mathbf{t_j}, \mathbf{s'_j}$) dapat dikatakan cocok jika jarak spasial (*sD*) antara *T* dan *I* lebih kecil dari toleransi r_0 dan perbedaan arah orientasi (*dD*) antara *T* dan *I* lebih kecil dari toleransi θ_0. Kedua kriteria tersebut disebut dengan kriteria kecocokan, yang dirumuskan sebagai berikut.

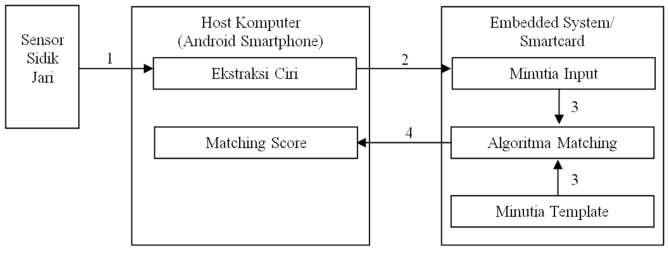
$$sD(t_i, s'_j) = \sqrt{\left(t^i_x - s'^{j}_{x}\right)^2 + \left(t^i_y - s'^{j}_{y}\right)^2} \quad (5)$$

$$dD(t_i, s'_j) = min\left(\left|t^i_\theta - s'^{j}_{\theta}\right|, 360^o - \left|t^i_\theta - s'^{j}_{\theta}\right|\right) \quad (6)$$

Nilai kemiripan yang memenuhi kriteria persamaan 5 dan 6 dari setiap kombinasi transformasi $(\Delta x, \Delta y, \Delta\theta)$ ditampung pada sebuah matrik akumulator $A[\Delta x, \Delta y, \Delta\theta]$. Untuk setiap pasangan$(t_i, s_j)$ akan dihitung semua kombinasi transformasi yang memetakan s_j menjadi s'_j. Nilai kemiripan akan di tambahkan dengan 1 jika kriteria $sD(t_i, s'_j) \le r_0$ dan $dD(t_i, s'_j) \le \theta_0$ terpenuhi. Setelah semua kombinasi data *minutiae I* dan *T* dihitung maka nilai matrik akumulator $A[\Delta x, \Delta y, \Delta\theta]$ tertinggi merepresentasikan nilai kemiripan antara sidik jari input *I* dengan sidik jari template *T*.

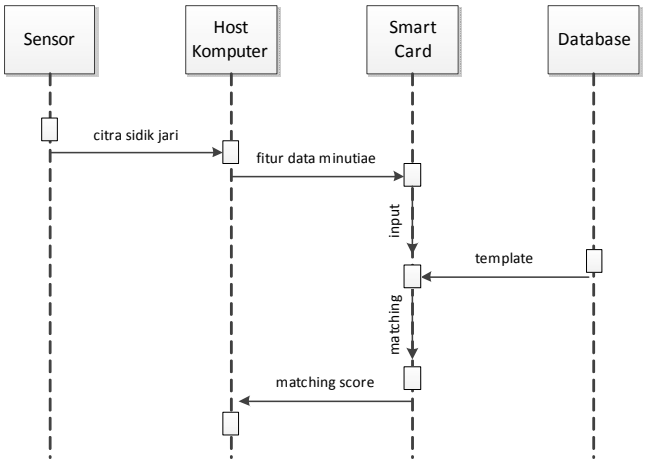
III. MODIFIKASI HOUGH TRANSFORM

Dalam algoritma *mathing* berbasis *hough transform*, bagian yang memberikan kontribusi beban komputasi paling berat terdapat pada iterasi penghitungan rotasi dan translasi untuk menemukan θ, Δx dan Δy paling optimal. Jika nilai θ, Δx dan Δy ini bisa diperoleh sebelumnya, tentu beban komputasi pada algoritma *matching* akan dapat berkurang dengan drastis. Oleh karena itulah dalam penelitian ini diajukan sebuah ide untuk memindahkan proses penghitungan nilai θ, Δx dan Δy di luar *smart card*. Teknisnya adalah dengan memberikan informasi sampel beberapa data minutia template sebagai referensi yang dapat digunakan oleh perangkat eksternal untuk mengestimasi nilai θ, Δx dan Δy. Setelah nilai θ, Δx dan Δy diketahui maka nilai ini bisa diberikan kepada algoritma *matching* sehingga dalam algoritma *matching* tinggal menghitung nilai kemiripan atau *matching score* saja. Detil mekanisme *sharing* komputasi ditunjukkan pada gambar 4.

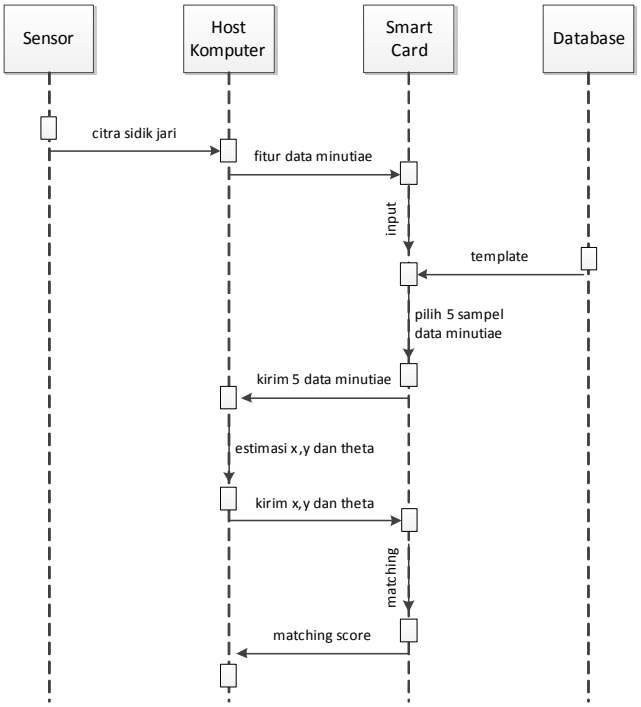


Gambar 4. *Konsep sharing komputasi hough transform.*

(1) Gambar input sidik jari akan dibaca oleh sensor kemudian dikirimkan ke *host* komputer.
(2) Proses ekstraksi ciri untuk mendapatkan data ciri (*features*) yang dalam hal ini berupa data *minutiae* dilakukan di *host* komputer. Selain itu *host* komputer juga melakukan kalkulasi estimasi nilai nilai θ, Δx dan Δy yang optimal. Data minutiae dan estimasi nilai θ, Δx dan Δy akan dikirimkan ke *smart card*.
(3) Selanjutnya data *minutiae* input dan template akan diproses oleh algoritma *matching* di dalam *smart card*. Data *minutiae* template akan dipilih sampel sebanyak 5-10 data saja dan pencarian nilai θ, Δx dan Δy yang optimal dilakukan disekitar data estimasi yang sudah diberikan oleh *host* komputer sehingga komputasi algoritma *matching* menjadi lebih ringan dan cepat.
(4) Hasil *matching* dikirimkan ke *host* komputer kembali.



Gambar 5. *Proses verifikasi dengan metode konvensional.*



Gambar 6. *Proses verifikasi dengan metode yang diajukan.*

Secara keseluruhan perbandingan proses verifikasi sidik jari antara metode konvensional dan metode yang diajukan ditunjukkan pada gambar 5 dan gambar 6. Seperti terlihat pada alur proses verifikasi di atas, metode yang diajukan membagi komputasi algoritma *matching* dan mereduksi jumlah data *minutiae* dengan tujuan meringankan beban komputasi pada sisi *smart card*. *Smart card* mengirimkan sekitar 5 sampel data *minutiae* ke *host* komputer agar *host* komputer dapat melakukan estimasi nilai θ, Δx dan Δy. Hasil esimasi nilai θ, Δx dan Δy selanjutnya dikembalikan ke *smart card* untuk membantu mempercepat komputasi penghitungan nilai kemiripan atau *matching score*.

IV. Hasil Eksperimen dan Pembahasan

Untuk mengevaluasi kinerja algoritma *matching* yang diajukan digunakan dataset sidik jari sejumlah 100 gambar yang terdiri dari 10 jari masing-masing 10 gambar. Setiap gambar sidik jari akan dilakukan proses ekstraksi ciri untuk mendapatkan fitur yang berupa data *minuteae* seperti ditunjukkan pada gambar 7.

00000000	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	f0	ff	38	00	00	00	00	00	fd	ff	00	00	ff	ff	00	01
00000010	d8	ff	92	00	00	00	07	00	20	00	20	00	ff	ff	00	01
00000020	ea	ff	dc	ff	00	00	09	00	ff	ff	ff	ff	00	00	00	01
00000030	e0	ff	b9	ff	01	00	09	00	ff	ff	01	00	fe	ff	00	01
00000040	1f	00	d5	ff	00	00	0b	00	ff	ff	00	00	20	00	00	01
00000050	14	00	05	00	00	00	0c	00	20	00	fe	ff	ff	ff	00	01
00000060	24	00	ad	00	01	00	10	00	ff	ff	20	00	01	00	00	01
00000070	e7	ff	5b	00	00	00	11	00	03	00	20	00	20	00	00	01
00000080	e4	ff	7b	ff	01	00	12	00	20	00	00	00	20	00	00	01
00000090	02	00	17	00	01	00	14	00	03	00	02	00	00	00	00	01
000000a0	d7	ff	5e	00	01	00	14	00	20	00	ff	ff	20	00	00	01
000000b0	d8	ff	7e	ff	00	00	14	00	20	00	fe	ff	20	00	00	01
000000c0	30	00	d4	ff	00	00	1b	00	20	00	00	00	00	00	00	01
000000d0	04	00	9f	ff	00	00	1c	00	02	00	ff	ff	20	00	db	00
000000e0	02	00	2b	00	00	00	1d	00	00	00	ff	ff	ff	ff	00	01
000000f0	e9	ff	a2	ff	00	00	1d	00	01	00	fe	ff	02	00	00	01
00000100	f6	ff	8b	ff	01	00	1d	00	20	00	20	00	00	00	00	01
00000110	fe	ff	4c	00	00	00	1e	00	ff	ff	ff	ff	00	00	95	00
00000120	10	00	83	ff	01	00	1e	00	00	00	20	00	fd	ff	00	01
00000130	01	00	6e	ff	01	00	1f	00	ff	ff	01	00	20	00	00	01

Gambar 7. *Data minutiae pada fitur sidik jari*

Fitur sidik jari terdiri dari beberapa *minutia* dan setiap *minutiae* berisi informasi x, y, θ, type. Setiap data minutiae disimpan dalam bentuk data berukuran 16 byte. 2 byte pertama menunjukkan nilai x, 2 byte selanjutnya nilai y, 2 byte selanjutnya nilai θ, 2 byte berikutnya tipe dan sisanya tidak digunakan. Contoh untuk baris pertama pada gambar 7 diatas nilai x = 0xF0FF, y = 0x3800, θ = 0x0000, type = 0x0000.

Algoritma *matching* untuk autentikasi sidik jari diinstall dan dijalankan sebagai applet pada *smart card* tipe Java Card Kona 251. Beberapa dataset yang terdiri dari 100 gambar, 80 gambar dan 60 gambar digunakan untuk membandingkan waktu yang diperlukan untuk melakukan komputasi algoritma matching antara metode konvensional hough transform dan metode yang diajukan. Tabel 1 menunjukkan hasil perbandingan baik dari sisi komputasi maupun dari sisi nilai error.

Tabel 1. *Perbandingan komputasi dan error.*

Dataset	Perbandingan	Konvensional	Metode yang diajukan
100	Komputasi	7.3 ms	4.8 ms
	Error	1.67%	1.40%
80	Komputasi	7.6 ms	4.8 ms
	Error	0.4%	0.76%
60	Komputasi	7.8 ms	4.8 ms
	Error	0.67%	0.50%
Rata-rata	Komputasi	7.5 ms	4.8 ms
	Error	0.91%	0,88%

Hasil pengujian di atas menunjukkan bahwa metode yang diajukan dapat mempercepat waktu komputasi *matching* secara rata-rata dari 7.5 ms menjadi 4.8 ms atau setara dengan 156% lebih cepat dibandingkan dengan metode konvensional. Sementara itu dari sisi tingkat error atau akurasi tidak mengalami perubahan secara signifikan. Nilai error algoritma *matching* yang diajukan berubah dari 0.91% menjadi 0.88% atau setara dengan perubahan sebesar 0.03%.

V. Kesimpulan

Modifikasi algoritma *matching hough transform* telah dilakukan dengan target implementasi pada *smart card*. Modifikasi dilakukan dengan cara mereduksi jumlah sampel data *minutiae* dan melakukan *sharing* komputasi antara *smart card* dan *host* komputer. Hasilnya adalah komputasi dapat dipercepat sampai dengan 156% lebih cepat tanpa mempengaruhi tingkat error atau akurasi secara signifikan.

Referensi

[1] D. Bhattacharyya, R. Ranjan, P. Das, T.-h. Kim, & S.K. Bandyopadhyay, "*Biometric Authentication Techniques and its Future Possibilities*", IEEE Second International Conference on Computer and Electrical Engineering, hal. 652-655, 2009.

[2] C. Barral, *"Biometrics & Security : Combining Fingerprints , Smart Cards and Cryptography"*, PhD Thesis, École Polytechnique Fédérale De Lausanne, 2010.

[3] A. M. Bazen and S. H. Gerez, *"Fingerprint matching by thin-plate spline modelling of elastic deformations"*, Journal of Pattern Recognition., Vol. 36, No. 8, pp. 1859–1867, 2003.

[4] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, "*Handbook of Fingerprint Recognition"*, 2nd ed. Springer, 2003.

[5] A. Surya Rikin, D. Li, T. Isshiki, and H. Kunieda, *"A fingerprint matching using minutia ridge shape for low cost Match-on-Card systems"*, IEICE Transaction on. Fundamental Electronics, Communication, Computer Science, Vol. E88-A, No. 5, pp. 1305–1312, 2005.

[6] A. Mohammad Abdel-Mawgoud Saleh, *"Enhanced Secure Algorithm for Fingerprint Recognition"*, PhD Thesis, Ain Shams University, 2011.

[7] L. Wieclaw, *"A minutiae-based matching algorithms in fingerprint recognition"*, Journal of Medical Informatics & Technologies, Vol. 13, 2009.