

PENCOCOKAN GAMBAR SIDIK JARI DENGAN KAMERA *HANDPHONE* MENGUNAKAN METODE RANSAC DAN TRANSFORMASI AFFINE BERBASIS ANDROID

Haruno Sajati¹, Dwi Nugraheny², Nova Adi Suwarso³

Program Studi Teknik Informatika

harunosajati@stta.ac.id¹, henynug@gmail.com², Novaadiso@gmail.com³

ABSTRACT

Fingerprints occur due to stroke differences. These stroke differences have occurred at a time when humans are still fetal form. A normal fingerprint pattern is formed of lines and spaces. These lines are called ridges whereas the spaces between these lines are called valleys. To make an introduction to the fingerprint image requires a variety of support tools. Starting from a fingerprint machine, a smartphone that has a fingerprint sensor and much more. In this research, the acquisition of image is done by grayscaling, histogram equalization, gabor filter, binary, thinning, 8 neighbors, matching. The result of making android application with the method that has been described to show unfavorable results seen from the calculation of the accuracy of 63%. Based on testing the specs android OS devices, this application can run on android with OS 4.4.2 specification kitkat.

Keywords : *OCR Fingerprint, Fingerprint recognition, Minutiae based matching, Fingerprint image processing.*

1. Pendahuluan

Perkembangan teknologi merupakan hal yang tidak dapat dihindari pada era ini. Teknologi berperan penting terhadap pergeseran pola kehidupan sehari-hari. Mulai dari pekerjaan yang dahulu hanya dapat dikerjakan oleh banyak orang hingga membutuhkan waktu sehari-hari dan bahkan lebih lama lagi saat ini dapat dikerjakan oleh seorang saja dengan waktu yang jauh lebih singkat.

Pengolahan citra digital telah memberikan manfaat yang banyak terhadap kehidupan manusia. Mulai dari pengenalan wajah hingga sistem presensi berbasis sidik jari. Sistem yang membutuhkan tingkat keamanan atau *privasi* yang tinggi dapat dibantu dengan teknologi ini.

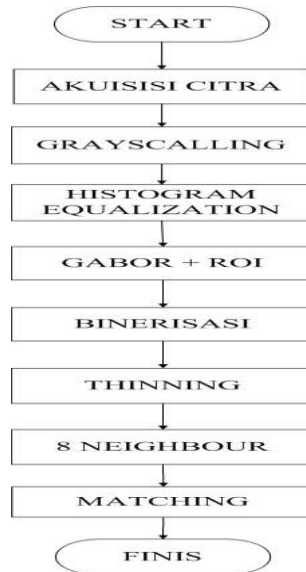
Dalam penggunaannya teknologi pengolahan citra sidik jari membutuhkan alat penunjang yang bervariasi. Mulai dari mesin *fingerprint, smartphone* yang memiliki sensor sidik jari dan masih banyak lagi. Hal ini terjadi karena kualitas citra yang dihasilkan dengan kamera biasa kurang baik. Sedangkan kebutuhan manusia saat ini mulai beranjak ke *mobile*. Hal ini akan mempersulit pengguna jika harus memiliki *devices* yang mendukung teknologi ini.

Pada penelitian ini penulis akan melakukan pengujian terhadap citra sidik jari yang dihasilkan melalui kamera *smartphone* android. Sehingga memungkinkan proses pembacaan sidik jari dapat dilakukan dari mana saja melalui *smartphone* dengan spesifikasi yang minimal.

2. Metodologi Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah eksperimen. Metode yang akan digunakan untuk pengujian objek citra sidik jari pada penelitian ini adalah akuisisi citra, *grayscale*, peningkatan kontras citra dengan *histogram equalization, gabor filter* dan seleksi *region of interest*, untuk memunculkan pola garis pada citra sidik jari. Setelah pola sidik jari didapatkan kemudian dilakukan binerisasi pada pola sidik jari. Citra biner kemudian dilakukan *thinning* untuk menipiskan garis-garis pola sidik jari. Untuk mengekstrak *minutiae* yaitu *ridge ending* dan *ridge bifurcation* dilakukan pendekatan 8 ketetanggaan / *neighbourhood*. Setelah

minutiae terdeteksi barulah dilakukan pencocokan. Untuk pencocokan sidik jari menggunakan metode *Minutiae Based Matching* menggunakan transformasi *affine* dan *Random Sample Consensus (RANSAC)*. Tahap-tahap penerapan metode-metode yang digunakan pada penelitian ini dapat dilihat Gambar 1.



Gambar 1. Tahap- Tahap yang Digunakan Pada Peneletian

A. Citra Digital

Citra adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi terus menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek ditangkap oleh alat-alat optic misalnya mata pada manusia, kamera, pemindai (*scanner*) dan sebagainya sehingga bayangan objek yang disebut citra tersebut terekam. Citra digital merupakan suatu larik dua dimensi atau suatu matriks yang elemen elemennya menyatakan tingkat keabuan dari elemen gambar.

B. Grayscale

Grayscale adalah proses penyederhanaan gambar dari format gambar berwarna RGB menjadi gambar berwarna abu-abu (*gray*). 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*. Untuk mengubah gambar berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi gambar *grayscale* dengan nilai *k*, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B sehingga secara mudah dapat dituliskan seperti persamaan (2). (Ardhianto, 2010)

$$k = (R + G + B) / 3 \quad (2)$$

C. Region of Interest

Region of Interest (ROI) adalah daerah yang dipilih sebagai daerah yang paling signifikan di dalam sebuah data yang akan diidentifikasi untuk tujuan tertentu. Tujuan dari penggunaan seleksi ROI adalah memperkecil jumlah fitur citra yang akan diproses, dengan demikian akan meningkatkan kecepatan proses.

D. Histogram Equalization

Histogram Equalization (HE) merupakan teknik yang digunakan untuk mengatur intensitas suatu citra dengan meningkatkan nilai kontras (Gonzalez dan Woods, 2008). Cara kerja HE adalah

dengan meratakan derajat histogram dari setiap sumbu warna. Prinsip kerjanya sangat sederhana, yaitu dengan membuat frekuensi warna lebih merata dari batas bawah ke batas atas sehingga tidak ada bagian histogram yang terlalu dominan. Di sisi lain, histogram akan membuat persebaran kumulatif *gray values* (dari paling gelap ke paling terang) menjadi merata dan membentuk persamaan $X = Y$

Untuk dapat melakukan HE ini diperlukan suatu fungsi distribusi kumulatif yang merupakan perhitungan kumulatif dari histogram. Fungsi distribusi kumulatif/*cumulative distribution function* (CDF).

E. Gabor filter

Filter Gabor merupakan fungsi Gaussian yang dikalikan dengan fungsi harmonik. Hal ini secara optimal terbatas sesuai prinsip ketidak pastian baik dalam frekuensi dan domain khusus yaitu $\Delta x \cdot \Delta \omega$ yang dekat dengan h , metrik ketidak pastian. Ini berarti bahwa Filter Gabor sangat selektif dalam kedua frekuensi dan posisi, sehingga mengakibatkan tajam tekstur deteksi batas. Paradigma segmentasi terkait dengan Filter Gabor didasarkan pada model filter bank di mana beberapa Filter diterapkan serentak ke gambar *input*. Filter fokus pada berbagai tertentu frekuensi. Jika gambar masukan berisi dua wilayah tekstur yang berbeda, perbedaan frekuensi lokal antara daerah akan mendeteksi tekstur dalam satu atau lebih filter *output* sub-gambar. Fungsi Gabor dasar dapat melakukan dekomposisi ruang sendi. Setiap Filter Gabor ditentukan oleh fungsi *Gabor* dasar. Karena spasial dan spasial frekuensi lokalisasi Filter Gabor secara luas digunakan untuk segmentasi tekstur, dan dituliskan

$$g(x, y) = \exp\left(-\left(\left(\frac{a}{2\sigma}\right)^2 + y^2\left(\frac{b}{2\sigma}\right)^2\right)\right) * x \cos\left(\frac{2\pi}{\lambda}(x \cos\theta + y \sin\theta) + \phi\right) \quad (1)$$

Dimana:

$$a = (x \cos \theta + y \sin \theta)^2$$

$$b = (-x \sin \theta + y \cos \theta)^2$$

$\sigma = \textit{bandwidth}$ menyatakan nilai efektif dari width suatu citra

$\lambda = \textit{lambda}$ menyatakan panjang gelombang suatu citra

$\theta = \textit{theta}$ menyatakan sudut suatu citra

$\gamma = \textit{gamma}$ menyatakan tingkat kecerahan (*brightness*) suatu citra

$\phi = \textit{phase}$ menyatakan bentuk suatu citra

F. Binerisasi

Proses binerisasi adalah proses menentukan nilai ambang (*Threshold*). nilai ini digunakan untuk membagi citra *grayscale* ke dalam dua nilai yaitu hitam dan putih. *Thresholding* adalah metode sederhana untuk membuat gambar biner dari gambar *grayscale*. Dalam proses *thresholding*, setiap piksel dari citra digital ditandai sebagai "objek" piksel, jika nilai piksel lebih besar dibandingkan nilai ambang (*Threshold*) biasanya, obyek piksel diberi nilai "1" sedangkan piksel latar belakang diberi nilai "0". (Ahmad,2015)

G. Thinning

Proses thinning merupakan salah satu pemrosesan citra (*image processing*). Proses *thinning* digunakan dengan tujuan :

- Mengurangi suatu daerah (*region*) menjadi suatu grafik / kurva dengan memperoleh kerangka (*skeleton*) dari daerah tersebut. Dengan demikian, *image* tersebut ditransformasikan menjadi bentuk struktural.
- Mengurangi suatu daerah yang tebal atau bergumpal menjadi unit-unit dengan *pixel-pixel* tunggal. Dengan demikian, *image* tersebut ditransformasikan menjadi garis-garis *pixel*. Proses *thinning* merupakan salah satu pemrosesan citra (*image processing*).

Proses *thinning* digunakan dengan tujuan :

- a. Mengurangi suatu daerah (*region*) menjadi suatu grafik/ kurva dengan memperoleh kerangka (*skeleton*) dari daerah tersebut. Dengan demikian, *image* tersebut ditransformasikan menjadi bentuk struktural.
- b. Mengurangi suatu daerah yang tebal atau bergumpal menjadi unit-unit dengan *pixel-pixel tunggal*. Dengan demikian, *image* tersebut ditransformasikan menjadi garis-garis *pixel*. (Suroto, 2009)

H. Minutiae Extraction

Pendeteksian minutia dilakukan dengan konvolusi 8 tetangga, yang dicari adalah akhir dari sebuah *ridge* (*ridge ending*) dan percabangan sebuah *ridge* (*bifurcation*). Sebuah *ridge* didefinisikan sebagai *ridge ending* apabila sebuah *pixel ridge* $r(i,j)$ hanya memiliki 1 tetangga, sedangkan sebuah *ridge* didefinisikan sebagai percabangan atau *bifurcation* apabila *pixel ridge* $r(i,j)$ memiliki lebih dari 2 tetangga *pixel*.

I. Matching

Pada tahapan pencocokan (*matching*) digunakan penggabungan dua metode, yaitu algoritma *Random Sample Consensus* (*RANSAC*) dan metode transformasi 2 dimensi Transformasi Affine. *Random Sample Consensus* (*RANSAC*) adalah metode iterasi untuk memperkirakan parameter dari model matematika dari sekumpulan data yang terdiri atas *inliers* dan *outliers*. Langkah pertama adalah menemukan *pixel* tepian, kemudian menghubungkan tepian tersebut menjadi segmen-segmen kecil menggunakan metode *RANSAC*. Setelah itu, menggabungkan segmen menjadi garis dan menemukan semua titik potong dari semua garis yang saling tegak lurus. Langkah terakhir, menemukan dua pasang titik potong dengan jarak terjauh, maka akan didapatkan empat garis terluar marker yang merupakan langkah penting dalam mendeteksi marker. (Anisa, dkk, 2013) Pada dasarnya transformasi afin terdiri dari unsur-unsur transformasi *scaling*, *skewing*, *rotating*, dan *translating*, masing-masing unsur transformasi tersebut dapat diilustrasikan dengan skema perubahan dimensi, bentuk, dan posisi.

J. OpenCV

OpenCV adalah sebuah *library* yang berisi fungsi-fungsi pemrograman untuk teknologi *computer vision* secara *real time*. OpenCV sudah menggunakan antarmuka bahasa C++ dan seluruh pengembangannya terdapat dalam format bahasa C++. Contoh aplikasi dari OpenCV yaitu interaksi manusia dan komputer; identifikasi, segmentasi dan pengenalan objek, pengenalan wajah, pengenalan gerakan dan penelusuran gerakan. Pada penelitian ini akan digunakan versi *opencv4Android*. (Fajar, 2014)

K. Peralatan dan Bahan

Dalam penelitian ini, alat dan bahan yang digunakan sebagai berikut:

1. *Hardware: Smartphone* Android versi 4.4.2 Kitkat
2. *Software: Android development tools Eclipse, Opencv library, Android ndk*

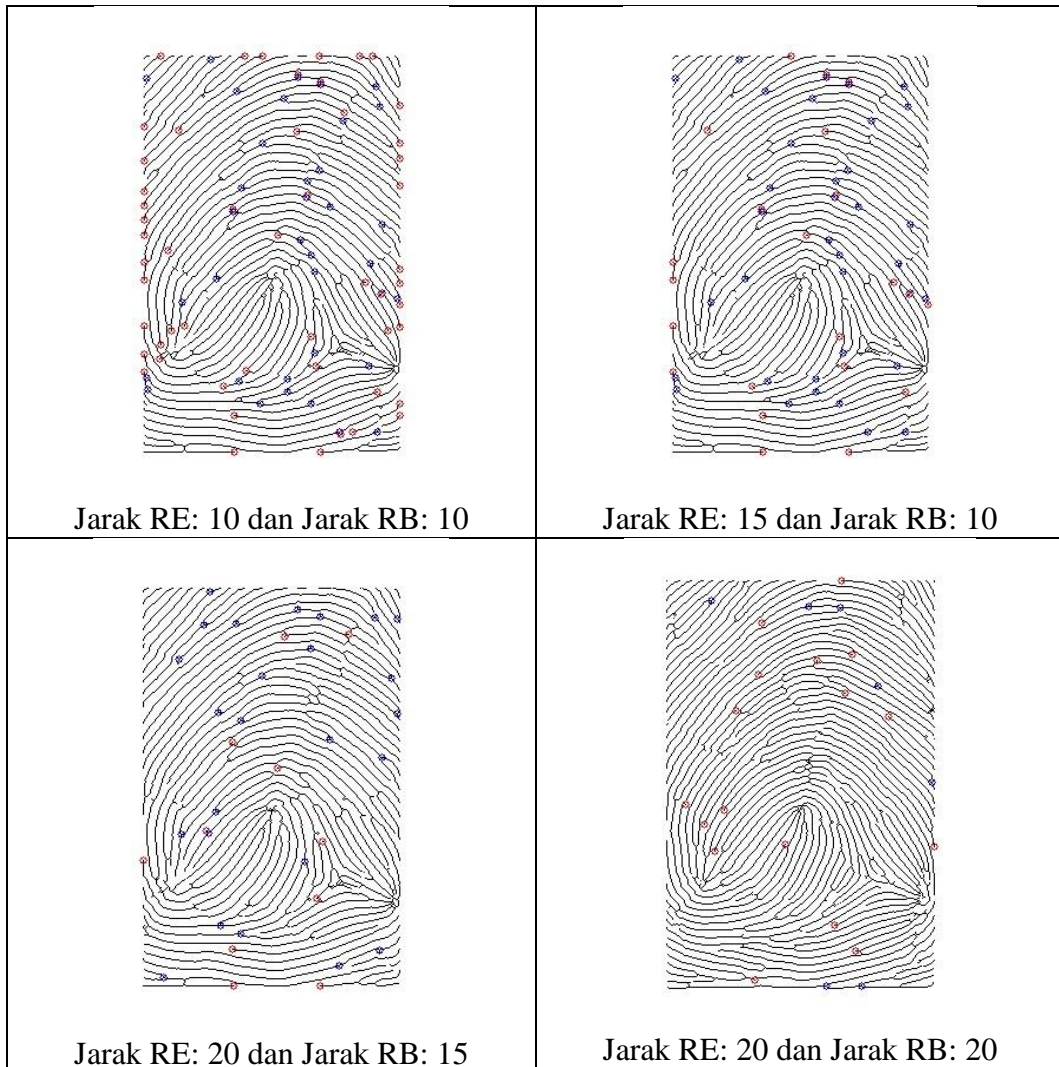
Objek: Citra sidik jari bersumber dari kamera *smartphone* dengan format *jpg* dengan ukuran 300x400.

3. Hasil Pengamatan dan Pembahasan

3.1. Ketepatan Identifikasi *Minutiae*

Pengujian ini bertujuan untuk melihat seberapa akuratkah metode yang digunakan dalam mengekstraksi *minutiae* dari citra sidik jari. Pengujian ini dapat dilakukan dengan identifikasi secara langsung terhadap citra hasil ekstraksi *minutiae*.

Hal yang sangat berpengaruh terhadap hasil pendeteksian *minutiae* adalah proses *filtering minutiae*. Semakin pendek jarak yang diberikan untuk mengidentifikasi suatu *minutiae* maka semakin banyak jumlah *minutiae* yang teridentifikasi. Sedangkan semakin banyak jumlah *minutiae* akan semakin besar kemungkinan terjadi kesalahan pencocokan karena semakin banyak pula kemungkinan pola *minutiae* yang sama antara citra kunci dan citra uji. Lihat pada Gambar 2.











Gambar 2. Pengaruh *Filtering Minutiae* Terhadap Jumlah *Minutiae* yang Terdeteksi

3.2. Kemampuan Metode Dalam Mengidentifikasi Kemiripan Citra Sidik Jari

Pada pengujian ini dilakukan pencocokan citra sidik jari dimana sebuah citra sidik jari akan dijadikan sebagai kunci atau sebagai citra database sedangkan citra lainnya digunakan sebagai citra uji. Citra yang diuji merupakan hasil dari stempel secara manual sidik jari menggunakan tinta berwarna hitam yang ditempelkan pada selembar kertas. Kemudian sidik jari yang terlihat pada kertas akan difoto satu persatu sesuai kelompok dan pemilik citra sidik jari, Hasil merupakan kemiripan dari citra kunci dan citra uji yang diproses dengan menggunakan metode RANSAC dan transformasi AFFINE.

Hasil dari pengujian ini dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian

NO	Citra Kunci	Citra Uji	Hasil
1			67%
2			100%
3			100%
4			69%

Tabel 2. Tabel perhitungan akurasi (Zhu, 2010)

Aktual	Pendeteksian Aplikasi	
	Positif	Negatif
Positif	TP (True Positif)	FP (False Positif)
Negatif	FN (False Negatif)	TN (True Negatif)

$$\text{Akurasi} = \frac{TN+TP}{TN+TP+FN+FP}$$

Aktual adalah kepemilikan sidik jari, aktual positif adalah citra sidik jari yang dimiliki oleh satu orang yang sama sedangkan aktual negatif adalah citra sidik jari yang dimiliki oleh orang yang berbeda. Pendeteksian positif adalah kesuksesan identifikasi sidik jari sedangkan negatif adalah kegagalan identifikasi sidik jari. Kesuksesan aplikasi dilihat dari nilai kecocokan pada citra sidik jari dengan pemilik yang sama lebih besar jika dibandingkan dengan citra sidik jari dengan pemilik yang berbeda.

Pada Tabel 2 dapat dilihat hasil dari pengujian pencocokan terhadap citra sidik jari. Pada kolom Positif adalah hasil pencocokan citra sidik jari terhadap sidik jari yang sama atau orang yang sama. Sedangkan pada kolom Negatif 1, Negatif 2 dan Negatif 3 merupakan hasil pencocokan citra sidik jari terhadap sidik jari orang yang berbeda atau sidik jari yang berbeda. Kemudian hasil tersebut dibandingkan antara pencocokan terhadap sidik jari orang yang sama dan orang yang berbeda. Jika hasil dari pencocokan terhadap sidik jari yang sama lebih besar dibandingkan sidik jari yang berbeda maka hasilnya adalah *true* jika hasil pengujian terhadap sidik jari yang berbeda lebih besar jika dibandingkan dengan pengujian terhadap sidik jari yang sama maka hasilnya adalah *false*. Jumlah *true* dan *false* akan dihitung untuk mencari akurasi dari aplikasi.

Tabel 3. Hasil perhitungan akurasi

NO	Positif (%)	Negatif 1 (%)	Negatif 2 (%)	Negatif 3 (%)	HASIL
1	67	67	67	89	False
2	100	27	47	60	True
3	100	27	20	33	True
4	69	20	33	27	True
5	75	31	23	38	True
6	100	23	62	31	True
7	54	40	30	80	False
8	80	23	100	57	False
9	50	27	20	33	True
10	47	26	21	16	True
11	100	16	74	16	True
12	80	21	37	26	True
13	67	38	23	23	True
14	30	17	28	17	True
15	22	21	21	19	True
16	60	18	14	14	True
17	27	17	67	17	False
18	33	20	15	45	False
19	27	10	17	10	True
20	67	50	20	15	True
21	71	15	20	20	True
22	67	17	22	28	True
23	33	28	22	28	True
24	48	15	12	50	False
25	25	83	25	25	False
26	67	27	91	55	False
27	33	45	27	64	False
28	16	7	11	7	True
29	11	44	26	9	False
30	12	12	21	9	False

Tabel 4. Hasil Pendeteksian Aplikasi

Aktual	Pendeteksian Aplikasi	
	Positif	Negatif
Positif	19	11
Negatif	11	19

$$\text{Akurasi} = \frac{19+19}{19+19+11+11} = \frac{38}{60} = 0,63$$

$$\text{Akurasi} = 63\%$$

3.3. Pengujian Spesifikasi Perangkat Android

Berdasarkan percobaan pengujian aplikasi pada *smartphone* android, untuk menggunakan aplikasi ini agar berjalan dengan stabil pada *smartphone* android maka spesifikasi minimum yang dapat digunakan adalah *smartphone* android dengan sistem operasi Android Kitkat dan Ram 429 MB. Dapat dilihat pada hasil ujicoba Tabel 5.

Tabel 5. Hasil Uji Coba Perangkat OS

NO	Versi OS	Ya	Tidak
1	Ice Cream Sandwich (versi 4.0)		✓
2	Jelly Bean (versi 4.1)		✓
3	Kitkat (versi 4.4)	✓	
4	Lolipop (versi 5.0)	✓	
5	Marshmallow (versi 6.0)	✓	

4. Penutup

4.1. Kesimpulan

Berdasarkan kegiatan yang telah dilaksanakan dalam penelitian ini, maka dapat diambil beberapa kesimpulan, diantaranya:

1. Metode yang digunakan pada penelitian ini dapat mendeteksi *minutiae*. Hal ini ditunjukkan pada proses identifikasi *minutiae*, *minutiae* dapat teridentifikasi, terutama dalam hal identifikasi *ridge ending (RE)* maupun *ridge bifurcation (RB.)*
2. Hasil dari pengenalan sidik jari menunjukkan hasil yang kurang baik. Hal ini dapat dilihat dari hasil pengujian pencocokan sidik jari. Nilai rata-rata akurasi pendeteksian sidik jari sebesar 63%.
3. Berdasarkan pengujian OS, Aplikasi dapat berjalan pada *smartphone* android dengan OS minimal versi 4.4 kitkat

4.2. Saran

Dalam penelitian mengenai pengenalan sidik jari ini tidak terlepas dari beberapa kekurangan. Oleh karena itu, penulis menyarankan beberapa hal, antara lain:

1. Pengembangan terhadap metode *filtering minutiae* agar pengidentifikasian *minutiae* menjadi lebih baik.
2. Akuisisi citra sebaiknya tidak dilakukan dengan media kamera untuk memberikan citra yang lebih baik.
3. Identifikasi terhadap komponen-komponen sidik jari dengan lebih kompleks sehingga akurasi pengenalan menjadi lebih baik.
4. Lakukan pengujian dengan metode lain dalam menampilkan pola dari guratan citra sidik jari.

Daftar Pustaka

- [1] Adi, Kusworo., 2003. *Perancangan dan Realisasi Sistem Ekstraksi Ciri Sidik Jari Berbasis Algoritma Filterbank Gabor*. Semarang: Jurusan Fisika, Universitas Diponegoro.

- [2] Arifin, AnisaAini., dkk. 2013. *Optimasi Deteksi Marker Pada Nyartoolkit Menggunakan MetodeRansac*. Malang: Universitas Brawijaya.
- [3] Cahyana, Fajar MIT., 2014. *Perancangan Program Penghitung Jumlah Kendaraan Di Lintasan Jalan Raya Satu Arah Menggunakan Bahasa Pemrograman C++ DenganPustakaOpencv*. Malang: Universitas Brawijaya.
- [4] Elia, Tiara., 2015. *Aplikasi Peningkatan Kualitas Citra Menggunakan Metode Histogram Equalization*. Medan: Jurusan Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer.
- [5] Juheri, Ahmad., 2015. *Identifikasi Pola Sidik Jari Berbasis Transaformasi Wavelet dan Jaringan Syaraf Tiruan Propagasi Balik*. Semarang: Jurusan Fisika, Universitas Negeri Semarang.
- [6] Elvayandri. 2002. *Sistem Keamanan Akses Menggunakan Pola Sidik Jari Berbasis Jaringan Saraf Tiruan*. Proyek Akhir Keamanan Sistem Informasi. Bandung: Institut Teknologi Bandung.
- [7] Legawa, Tri., dkk. 2011. *Pengenalan Sidik Jari Menggunakan Algoritma Pencocokan Adaptif Berdasarkan Penjajaran Minutiae*. Semarang: Jurusan Teknik Elektro Universitas Diponegoro.
- [8] Nampira, YustiFitriyani., 2012. *Aplikasi Deteksi Mikrokalsifikasi dan Klasifikasi Citra Mammogram Berbasis Tekstur Sebagai Pendukung Diagnosis Kanker Payudara*. Depok: Jurusan Teknik Informatika, Universitas Gunadarma.
- [9] Nasir, Muhammad., dkk., 2012. *Pengujian Kualitas Citra Sidik Jari Kotor Menggunakan Learning Vector Quantization*. Aceh: Jurusan Teknik Elektro, Politeknik Negeri Lhokseumawe.
- [10] Pangestu, Peter., 2015. *Penerapan Histogram Equalization pada Optical Character Recognition Preprocessing*. Tangerang: Jurusan Teknik Informatika, Universitas Multimedia Nusantara, Tangerang, Indonesia.
- [11] Suroto., 2009. *Studi Penyempurnaan Identifikasi Sidik Jari Pada Algoritma Minutia*. Depok Jawa Barat: Jurusan Teknik Elektro, Universitas Indonesia.
- [12] Tanzil, RobbinKristanto., 2015. *Pengenalan Sidik Jari Menggunakan Jaringan Saraf*. Surabaya: Sekolah Tinggi Teknik Surabaya.
- [13] Ratnadewi., dkk. 2004. *Identifikasi Sidik jari menggunakan Metoda Modified Gabor Filter (MGF)*. Bandung: FakultasTeknik, Universitas Kristen Maranatha.
- [14] Zhu, Wen., dkk. 2010. *Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS® Implementations*. Washington: Octagon Research Solution, Wayne, Pa.