ARTIKEL

IDENTIFIKASI TELAPAK TANGAN MENGGUNAKAN IMAGE PROCESSING

Oleh:

Fitri Permata Sari

13.1.03.02.0002

Dibimbing oleh :

1. **Fatkur Rhohman, M.Pd.**
2. **Resty Wulanningrum, M.Kom.**

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS NUSANTARA PGRI KEDIRI

2017

SURAT PERNYATAAN
ARTIKEL SKRIPSI TAHUN 2017

Yang bertanda tangan di bawah ini:

Nama Lengkap	: Fitri Permata Sari
NPM	: 13.1.03.02.0002
Telepun/HP	: 085646682601
Alamat Surel (Email)	: fitripermata38@gmail.com
Judul Artikel	: Identifikasi Telapak Tangan Menggunakan Image Processing
Fakultas – Program Studi	: Fakultas Teknik - Teknik Informatika
Nama Perguruan Tinggi	: Universitas Nusantara PGRI Kediri
Alamat Perguruan Tinggi	: JL. K.H. Achmad Dahlan No. 76, Kediri

Dengan ini menyatakan bahwa :

a. artikel yang saya tulis merupakan karya saya pribadi (bersama tim penulis) dan bebas plagiarisme;
b. artikel telah diteliti dan disetujui untuk diterbitkan oleh Dosen Pembimbing I dan II.

Demikian surat pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari ditemukan ketidaksesuaian data dengan pernyataan ini dan atau ada tuntutan dari pihak lain, saya bersedia bertanggungjawab dan diproses sesuai dengan ketentuan yang berlaku.

Mengetahui		Kediri, 10 Agustus 2017
Pembimbing I	Pembimbing II	Penulis,
Fatkur Rhohman, M.Pd. NIDN. 0728088503	Resty Wulanningrum, M.Kom. NIDN. 0704108701	Fitri Permata Sari 13.1.03.02.0002

IDENTIFIKASI TELAPAK TANGAN MENGGUNAKAN IMAGE PROCESSING

Fitri Permata Sari
13.1.03.02.0002
Fakultas Teknik – Teknik Informatika
Email : fitripermata38@gmail.com
Fatkur Rohman, M.Pd. dan Resty Wulanningrum, M.Kom.
UNIVERSITAS NUSANTARA PGRI KEDIRI

ABSTRAK

Fitri Permata Sari: Identifikasi Telapak Tangan Menggunakan *Image Processing*, Skripsi, Teknik Informatika, FT UN PGRI Kediri, 2017.
Kata Kunci : *K-Nearest Neighbor,Image Processing*.

Penelitian ini dilatar belakangi dari pengamatan peneliti dan pengalaman peneliti, bahwa pengenalan (sistem biometrika) juga dapat diterapkan menggunakan telapak tangan . Pada saat ini sudah banyak pengguna sistem biometrik yang menggunakan sidik jari. Berbeda dengan pengenalan menggunakan telapak tangan, sistem penggunaan sidik jari ini sangat rentan terjadinya ketidak akuratan atau juga bisa dikatakan kelemahannya. Dikarenakan jari tangan lebih rentan terluka dibandingkan dengan telapak tangan yang kemungkinannya sangat kecil untuk terluka atau terkena goresan benda maupun lainya.

Permasalahan peneliti ini adalah (1) Bagaimana cara menerapkan metode *K-Nearest Neighbors (K-NN)* pada pengenalan ciri garis telapak tangan seseorang. (2) Bagaimana mengimplementasikan pengenalan telapak tangan seseorang menggunakan teknik pengolahan citra.

Penelitian ini menggunakan metode *K-Nearest Neighbors (K-NN).* Sebelum dilakukan pengenalan, *file* citra sebelumnya dilakukan proses *preprocessing* terlebih dahulu. Proses tersebut meliputi *greyscale* serta pendeteksian tepi *laplacian.* Dari hasil pengujian pada skenario pertama diperoleh akurasi sebesar 40% pada *dataset* sebanyak 60 dan data *test* 40. Pada skenario kedua, diperoleh akurasi sebesar 60% untuk *dataset* 40 dan data *test* sebanyak 60. Pada pengujian ketiga diperoleh akurasi sebesar 45% untuk *dataset* 60 dan data *test* sebanyak 20. Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa metode *K-Nearest Neighbors (K-NN)* tingkat akurasi yang didapatkan masih kurang untuk mengidentifikasi telapak tangan.

I. LATAR BELAKANG

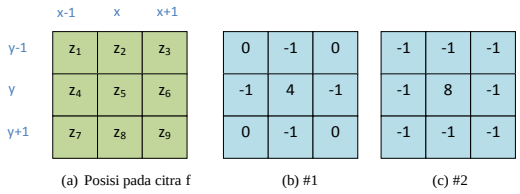
Pada era informasi ini, sistem pengenalan seseorang (*personal recognition*) secara otomatis menjadi sesuatu yang sangat penting. Apalagi dengan perkembangan teknologi yang begitu pesat seperti sekarang. Dengan meningkatnya sistem pengenalan individu yang handal dan dapat dipercaya semakin meningkat untuk keamanan sistem. Sistem pengenalan diri secara otomatis sangat diperlukan pada era informasi seperti sekarang ini. Teknologi pengenalan diri dengan menggunakan bagian tubuh atau perilaku manusia dikenal dengan istilah biometrika. Menurut Maltoni (2009), menyatakan bahwa pengenalan biometrik atau disebut juga identifikasi biometrik merupakan pengenalan seseorang secara otomatis berdasarkan karakteristik unik dari fisiologis (bagian –bagian tubuh tertentu seperti sidik jari, wajah, reina) maupun perilakunya. Pada saat ini sudah banyak pengguna sistem biometrik yang menggunakan sidik jari. Berbeda dengan pengenalan menggunakan telapak tangan, sistem penggunaan sidik jari ini sangat rentan terjadinya ketidak akuratan atau juga bisa dikatakan kelemahannya. Dikarenakan jari tangan lebih rentan terluka dibandingkan dengan telapak tangan yang kemungkinannya sangat kecil untuk terluka atau terkena goresan benda maupun lainya.

II. METODE

a. Operator Laplace

Operator Laplacian merupakan contoh operator yang berdasarkan pada turunan kedua. Operator ini bersifat *omnidirectional*, yakni menebalkan bagian tepi ke segala arah. Namun, operator *Laplacian* memiliki kelemahan, yakni peka terhadap derau, memberikan ketebalan ganda dan tidak mampu mendeteksi arah tepi. (Kadir, 2013).

Gambar 1. Matrik Laplacian



b. *K-Nearest Neighbor*

KNN adalah sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. KNN termasuk adalam golongan *supervised learning*, dimana hasil *query instance* yang baru diklasifikasikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam KNN. Nantinya kelas yang baru dari suatu data akan dipilih berdasarkan grup kelas yang paling

dekat jarak vektornya (Wardani, 2013).

Algoritma metode *K-Nearest Neighbor* (KNN) sangatlah sederhana, bekerja berdasarkan jarak terpendek dari *query instance* ke *training sample* untuk menentukan KNN-nya. *Training sample* diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi memepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasrakan klasifikasi *traning sample*. Sebuah titik pada ruang ini ditandai kelas c jika kelas merupakan yang paling banyak ditemui pada k buah tetangga terdekat dari titik tersebut. Dekat jauhnya tetengga biasanya dihitung berdasarkan *Euclidean Distance* (Wardani, 2013).

Jarak *Euclidean* paling sering digunakan menghitung jarak. Jarak *Euclidean* berfungsi menguji ukuran yang digunakan sebagai interpretasi kedekatan jarak antara dua obyek. yang dipresentasikan sebagai berikut :

$$D(a,b) \quad = \sqrt{\sum_{k-1}^{d}(a_K - bk)^2}$$

Dimana D (a,b) adalah jarak skalar dari dua buah vektor a dan b dari matrik berukuran dua dimensi. Pada fase training algoritma ini hanya melakukan penyimpanan vektor-vektor firut dan klasifikasi data *training sample*. Pada klasifikasi fitur-fitur yang sama dihitung untuk testing data (yang klsifikasinya tidak diketahui). Jarak dari vektor yang baru ini terhadap seluruh vektor training dihitung, dan sejumlah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan masuk pada klasifikasi terbanyak dari titik-titik tersebut.

Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasab antara setiap klasifikasi menjadi semakin kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-volidation*. Kasus khusus dimana kalsifikasi diprediksikan berdasarkan training data yang paling dekat (dengan kata lain, k=1) disebut algoritma *nearest neighbor* (Wardani, 2013). Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur itu tidak setara dengan relevansinya terhadap klasifikasi.

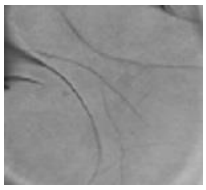
c. Analisa Data Logika Metode

1. Simulasi Algoritma

a. Data testing

Pada piksel ini diambil dari citra telapak tangan yang sudah melalui proses grayscale dengan simulasi sebagai berikut :

Tabel 1. Matrik Citra Grayscale



142	255	89
101	55	120
100	40	102

Untuk memperoleh nilai dari piksel deteksi tepi, dengan mengkalikan piksel citra grayscale dengan matrik *laplacian* dengan mempartisi 3x3 , dan dihitung setiap partisinya.

Tabel 2. Matrik Laplacian

-1	-1	-1
-1	8	-1
-1	-1	-1

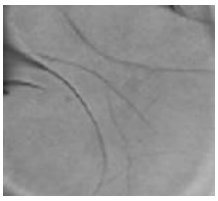
Tabel 3. Hasil Laplacian

0	255	0
0	219	0
0	118	0

Setelah melalui proses deteksi tepi *laplacian* kemudian hasil nilai citra telapak tangan dibandingkan dengan nilai citra telapak tangan yang sudah disimpan dalam database.

1) Data Training

a. Data training 1 citra telapak tangan pada piksel ini diambil dari citra telapak tangan yang sudah melalui proses grayscale.



123	45	255
231	251	265
237	112	121

Untuk memperoleh nilai dari piksel deteksi tepi, dengan mengkalikan piksel citra *grayscale* dengan matrik *laplacian* dengan mempartisi 3x3, dan dan dihitung setiap partisinya.

Tabel 4. Matrik Laplacian

-1	-1	-1
-1	8	-1
-1	-1	-1

Tabel 5. Matrik Laplacian

0	0	0
0	255	0
0	255	0

Dari matrik diatas diketahui nilai matrik *training Laplaciant* citra telapak tangan adalah 0,0,0,0,255,0,0,255,0 dengan mengkalikan piksel citra *grayscale* dengan matrik *laplacian*.

b. Data training 2 pada piksel ini diambil dari citra telapak tangan yang sudah melalui proses grayscale.

Tabel 6. Matrik Citra *Grayscale*

254	50	127
235	70	50
264	23	234

Untuk memperoleh nilai dari piksel deteksi tepi, dengan mengkalikan piksel

Gambar 2. Citra Grayscale

citra grayscale dengan matrik laplacian dengan mempartisi 3x3, dan dan dihitung setiap partisinya.

Tabel 7. Matrik *Laplacian*

-1	-1	-1
-1	8	-1
-1	-1	-1

Tabel 8. Hasil Matrik *Laplacian*

0	19	0
0	255	0
0	0	0

Setelah melalui proses deteksi tepi laplacian diperoleh hasil 0,19,0,0,255,0,0,255,0 matrik tersebut yang akan dibandingkan dengan fitur yang ada didalam database yang sebelumnya sudah di olah dalam proses *training*.

d. Identifikasi KNN

Identifikasi telapak tangan dengan membandingkan antara data training dan data testing dengan menggunakan algoritma KNN.

$$d = \sqrt{\sum_{i=1}^{p}(x_{2i} - x_{1i})^2}$$

1) Membandingkan dari data *testing* dengan data *training* 1 Diperoleh nilai matrik data testing : 0,255,0,0,129,0,0,118,0. Sementara nilai matrik data *training* 1 adalah: 0,0,0,0,255,0,0,255,0

$$d = \sqrt{(0-0)^2 + (255-0)^2 + (0-0)^2 + (0-0)^2 + (219-255)^2 + (0-0)^2 + +(0-0)^2 + (118-255)^2 + (0-0)^2}$$

$$= \sqrt{255^2} + (-36)^2 + (-137)^2$$

$$= \sqrt{65025 + 1296 + 18769}$$

$$= \sqrt{85090} = 291{,}7019$$

2) Membandingkan dari data *testing* dengan data *training* 2 Diperoleh nilai matrik data testing : 0,255,0,0,129,0,0,118,0. Sementara nilai matrik data *training* 2 adalah: 0,19,0,0,255,0,0,0,0

$$d = \sqrt{(0-0)^2 + (255-19)^2 + (0-0)^2 + (0-0)^2 + (219-255)^2 + (0-0)^2 + +(0-0)^2 + (118-0)^2 + (0-0)^2}$$

$$= \sqrt{236^2} + (-36)^2 + 118^2$$

$$= \sqrt{55696 + 1296 + 13924}$$

$$= \sqrt{70916} = 266{,}3006$$

Kesimpulan : diperoleh dari hasil perhitungan dengan mengunakan algoritma KNN dengan proses perhitungan antara data *testing* dan data *training* 1 dengan jarak =291,7019 dan proses perhitungan antara perhitungan data *testing* dan data *training* 2 dengan jarak=266,3006 . Proses pencocokan pada data *testing* jenis data *training* 2 Hasil diperoleh jarak terdekat adalah data training 2 dengan jarak = 266,3006.

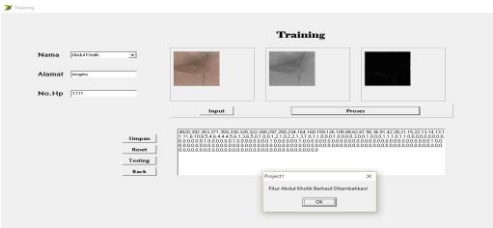
III. Tampilan Progam

Dalam pembuatan aplikasi identifikasi telapak tangan ini terdapat beberapa tampilan program. Berikut ini akan digambarkan dan dijelaskan mengenai tampilan progran tersebut.

Tampilan program ini merupakan *form-form* yang terdapat pada aplikasi identifikasi telapak tangan. *User* dapat memilih beberapa menu yang tersedia pada menu utama. Pada *form* ini pengguna akan ditampilkan menu utama yang mudah digunakan pada aplikasi identifikasi telapak tangan. Terdapat 3 pilihan yaitu tombol image processsing, testing, dan keluar.

Gambar 2 Tampilan Utama Program

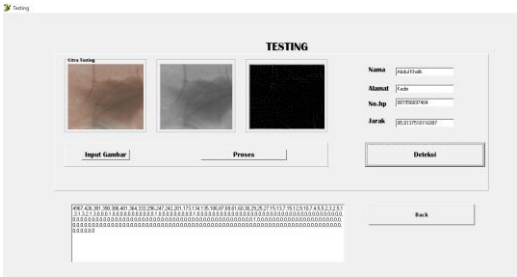
Ketika penguna menekan tombol *Image Processing* pengguna akan disuguhkan tampilan form proses klasifikasi training yang dapat dilihat pada gambar 3.



Gambar 3 Tampilan pada tombol *proses* pada *form training*

Ketika aplikasi sudah menampilkan proses pengguna diwajibkan menekan tombol simpan untuk menyimpan data fitur citra ke database sebagai acuan citra testing. Ketika pengguna telah menekan tombol simpan, aplikasi tersebut akan menampilkan pesan bahwa data fitur telah tersimpan. Hal ini dapat dilihat pada gambar 5.13.

Ketika aplikasi sudah menampilkan semua proses, tahapan akhir adalah *user* menekan tombol deteksi untuk mengetahui identitas dan jarak pada perhitungan *euclidean.* tampilan akhir tersebut dapat dilihat pada gambar 5.18



Gambar 4. Tampilan tombol Deteksi pada *form testing*.

IV. Skenario Uji Coba 1

Skenario ini menggunakan data *training* sebanyak 60 citra telapak tangan. Dengan rincian mengambil 3 citra telapak tangan dari setiap responden. Daftar citra *training* dapat dilihat tabel 5.6.

Tabel.5.6 Skenario Uji Coba 1

No	Citra Training	Citra Testing
1.	Abdul_001, Abdul_002, Abdul_003,	Abdul_004, Abdul_005
2.	Agus_001,Agus_002,Agus_003	Agus_004,Agus_005
3.	Andik_001, Andik_002, Andik_003	Andik_004,Andik_005
4.	Bagus_001, Bagus_002, Bagus_003	Bagus_004,Bagus_005
5.	Pras_001, Pras_002, Pras_003	Pras_004,Pras_005
6.	Pras_001, Pras_002, Pras_003	Bam_004,Bam_005
7.	Bayu A_001, Bayu A_002, Bayu A_003	Bayu A_004,Bayu A_005
8.	Bayu K_001, Bayu K_002, Bayu K_003	Bayu K_004,Bayu K_005
9.	Dony_001, Dony_002, Dony_003	Dony_004, Dony_005
10.	Faisal_001, Faisal_002, Faisal_003	Faisal_004,Faisal_005
11.	Fitri_001, Fitri_002, Fitri_003	Fitri_004,Fitri_005
12.	Lia_001, Lia_002, Lia_003	Lia_004,Lia_005
13.	Niko_001, Niko_002, Niko_003	Niko_004,Niko_005
14.	Puji_001, Puji_002, Puji_003	Puji_004, Puji_005

2.Skenario Uji Coba 2

Skenario ini menggunakan data *training* sebanyak 20 citra telapak tangan. Dengan rincian mengambil 2 citra telapak tangan dari setiap responden. Daftar citra *training* dapat dilihat tabel 5.7.

Tabel.5.7 Skenario Uji coba 2

No	Citra Training	Citra Testing
1.	Abdul_001, Abdul_002	Abdul_003, Abdul_004, Abdul_005
2.	Agus_001,Agus_002	Agus_003,Agus_004,Agus_005
3.	Andik_001, Andik_002	Andik_003,004,Andik_005
4.	Bagus_001, Bagus_002	Bagus_003,Bagus_004,Bagus_005
5.	Pras_001, Pras_002	Pras_003,Pras_004,Pras_005
6.	Pras_001, Pras_002	Bam_003,Bam_004,Bam_005
7.	Bayu A_001, Bayu A_002	Bayu A_003, Bayu A_004,Bayu A_005
8.	Bayu K_001, Bayu K_002	Bayu K_003,Bayu K_004,Bayu K_005
9.	Dony_001, Dony_002	Dony_003,Dony_004, Dony_005
10.	Faisal_001, Faisal_002	Faisal_003,Faisal_004,Faisal_005
11.	Fitri_001, Fitri_002	Fitri_003,Fitri_004,Fitri_005
12.	Lia_001, Lia_002	Lia_003,Lia_004,Lia_005

IV. Skenario Uji Coba 3

Skenario ini menggunakan data *training* sebanyak 60 citra telapak tangan. Dengan rincian mengambil 3 citra telapak tangan dari setiap responden. Daftar citra *training* dapat dilihat tabel 5.

Tabel 5. Skenario Uji coba 3

No	Citra Training	Citra Testing
1.	Abdul_001, Abdul_002, Abdul_003	Abdul_005
2.	Agus_001,Agus_002, Agus_003	Agus_005
3.	Andik_001, Andik_002, Andik_003	Andik_005
4.	Bagus_001, Bagus_002, Bagus_003	Bagus_005
5.	Pras_001, Pras_002, Pras_003	Pras_005
6.	Bam_001, Bam_002, Bam_003	Bam_005
7.	Bayu A_001, Bayu A_002, Bayu A_003	Bayu A_005
8.	Bayu K_001, Bayu K_002, Bayu K_003	Bayu K_005
9.	Dony_001, Dony_002, Dony_003	Dony_005
10.	Faisal_001, Faisal_002, Faisal_003	Faisal_005
11.	Fitri_001, Fitri_002, Fitri_003	Fitri_005
12.	Lia_001, Lia_002, Lia_003	Lia_005

1. Hasil Uji Coba

Dari ketiga skenario yang telah diuji coba, didapatkan hasil sebagai berikut **:**

Skenario 1	Training	Testing	Benar	Salah	Accuracy
	60	40	16	24	40%
Skenario 2	Training	Testing	Benar	Salah	Accuracy
	40	60	16	34	60%
Skenario 3	Training	Testing	Benar	Salah	Accuracy
	60	20	9	11	45%

2. Kesimpulan Uji Coba

Dari uji coba telah dilakukan pada sistem, dapat ditarik kesimpulan sebagai berikut :

a. Semakin banyak jumlah data training akan menghasilkan nilai prosentase akurasi yang lebih sedikit. Hal ini dibuktikan dengan skenario 1 dan 3 dimana pads skenario tersebut menggunkan data yang sama, yakni menggunakan *scan* dengan 100*100 *pixel.* Terdapat kesaamaan pada skenario 1 menggunakan data training sebanyak 60 buah menghasilkan akurasi sebesar 40%, sedangkan pada skenario 2 menggunakan data training sebanyak 60% yang mengasilkan 45%.

b. Proses *scan* yang berbeda pada data training dan testing dapat mempengaruhi proses pengenalan telapak tangan. Hal ini dibuktikan pada skenario 2, dimana menggunakan data traning 40 dan data testing 60 yang

menghasilkan prosentase akurasi sebesar 60%.

V. Kesimpulan

A. Simpulan

Berdasarkan analisis, hasil dan pembahasan mengenai aplikasi identifikasi telapak tangan menggunakan *image processing* yang telah dibuat, dapat disimpulkan beberapa hal sebagai berikut:

1. Pada penelitian ini peneliti menggunakan inputan telapak tangan sebagai bahan penelitian sistem ini menggunakan metode K-NN dan menggunakan beberapa *preprocessing* yakni dengan menggunakan proses *greyscale* dan *lapalacian* proses ini dilakukan sebelum melakukan proses pada metode K-NN.
2. Pada penelitian ini pengambilan telapak tangan menggunakan *scan* dan dilakukan *crop* pada setiap telapak tangan.

B. Saran

Pembuatan aplikasi identifikasi telapak tangan menggunakan *image processing* ini tidak terlepas dari permasalahan dan kekurangan, baik dalam proses penelitian maupun kekurangan dari pihak peneliti sistem. Untuk itu pengembangan sistem selanjutnya diharapkan :

1. Aplikasi dapat dikembangkan agar mampu mengenali citra telapak tangan bukan hanya bagian kiri namun juga bagian kanan manusia.
2. Sistem dapat dikembangkan dengan menggunakan algoritma yang lain untuk proses pencocokan citra telapak tangan dengan ekstraksi fitur warna, bentuk, dan tekstur agar hasil pencocokan lebih optimal.

VI. Daftar Pustaka

Putra, Toni W A. 2013. *Pengenalan Wajah Dengan Matriks Kookurensi Aras Keabuan Dan Jaringan Syaraf Tiruan Probabilistik*. Tesis. Semarang: Universitas Diponegoro.

Wardani, E. W. 2013. *Pengenalan Motif Batik Mengunakan Tarnsformasi Paket Wavelet.* Tuga Akhir. Bandung: Universitas Widyatama.

Kadir. 2013. *Dasar Pengolahan Citra dengan Delphi*. Yogyakarta: ANDI.