

PERUMUSAN SIDIK JARI BERJENIS *LOOP* DENGAN MENGUNAKAN METODE *POINCARÉ INDEX*

LOOP TYPE FINGERPRINT FORMULATION BY USING POINCARÉ INDEX METHOD

Farhan Annur Mahmudi¹, Raditiana Patmasari. S.T., M.T.², Sofia Sa'idah. S.T., M.T.³

^{1,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹farhananr@student.telkomuniversity.ac.id, ²raditiana@telkomuniversity.co.id,

³sofiasaidahsfi@telkomuniversity.ac.id

Sidik jari adalah salah satu karakteristik fisiologis pada tubuh manusia yang dapat digunakan untuk identifikasi manusia. Sidik jari dapat digunakan sebagai pengenalan atau identitas karena sidik jari memiliki bentuk dan alur yang berbeda pada setiap manusia. Pihak kepolisian menggunakan sidik jari sebagai metode identifikasi menggunakan cara yang konvensional sampai saat ini. Oleh karena itu dibutuhkan sistem yang dapat digunakan untuk perumusan sidik jari berdasarkan citra digital.

Tugas akhir ini membuat simulasi sistem yang dapat melakukan perumusan sidik jari dengan metode *Poincaré Index*. *Input* sistem berupa citra digital *grayscale* sidik jari dari sumber data sekunder berupa *dataset*. Tahap pemrosesan diliputi oleh *pre-processing* yang mencakup filter *gaussian*, binerisasi, dan *spot removal*. Setelah tahap *pre-processing* dilakukan pendeteksian titik *core* dan *delta*, perhitungan jarak antara titik *core* dan *delta* dan diakhiri oleh tahap perumusan sidik jari. *Output* yang dikeluarkan oleh sistem adalah berupa rumus sidik jari yang mencakup informasi tipe sidik jari dan jarak antara titik *core* dan *delta*.

Sistem perumusan sidik jari yang dirancang menggunakan *input* citra digital sidik jari *grayscale* bertipe *loop* yang dapat menghasilkan rumus sidik jari menggunakan metode *poincaré index*. Hasil daripada penelitian ini adalah sebuah simulasi sistem yang dapat melakukan perumusan sidik jari dengan tingkat keberhasilan sistem sebesar 80.95%.

Kata kunci : sidik jari, biometrik, *Poincaré index*.

Fingerprints are one of the physiological characteristics of the human body that can be used for human identification. Fingerprints can be used as identifiers or identities because fingerprints have a different shape and path for each human being. The police use fingerprints as a method of identification using conventional methods until recently. Therefore we need a system that can be used for fingerprint formulation based on digital images.

This final project creates a simulation system that can perform fingerprint formulation using the Poincaré Index method. The system input is a grayscale digital image of fingerprints from a secondary data source in the form of a dataset. The processing stage includes pre-processing which includes gaussian filters, binaryzation, and spot removal. After the pre-processing stage, the detection of core and delta points is carried out, calculating the distance between core and delta points and ending with the fingerprint formulation stage. The output issued by the system is a fingerprint formula which includes information on the type of fingerprint and the distance between the core and delta points.

The fingerprint formulation system is designed to use a loop-type grayscale fingerprint digital image input that can generate fingerprint formulas using the poincaré index method. The result of this research is a system simulation that can perform fingerprint formulation with a system success rate of 80.95%.

Keywords: fingerprint, biometric, Poincaré index

1. Pendahuluan

1.1 Latar Belakang

Sidik jari adalah salah satu karakteristik fisiologis pada tubuh manusia yang dapat digunakan untuk identifikasi manusia yang dapat digunakan sebagai pengenalan atau identitas karena sidik jari memiliki bentuk dan alur yang berbeda pada setiap manusia [1]. Pihak kepolisian menggunakan sidik jari sebagai metode identifikasi menggunakan cara yang konvensional sampai saat ini.

Metode yang konvensional meliputi penggunaan tinta daktiloskopi, *roller*, dan kartu sidik jari memiliki kekurangan pada validitas hasil perumusan dikarenakan kurangnya tingkat ketelitian pada pembacaan sidik jari. Oleh karena itu dibutuhkan sistem yang dapat digunakan untuk perumusan sidik jari berdasarkan citra digital. Dengan adanya sistem perumusan sidik jari menggunakan citra digital, diharapkan dapat menggantikan metode konvensional yang digunakan oleh pihak kepolisian saat ini dan untuk waktu kedepannya.

Penelitian Siti Lailatul Afifah [2] sebelumnya telah dibahas mengenai perangkat lunak yang dapat melakukan perumusan sidik jari berjenis *loop* menggunakan metode *Walking Directional Fields* (WDFs) menggunakan skenario *noise* tambahan dengan nilai 0.10, 0.20, 0.30, 0.40 dengan menggunakan *dataset* primer dari sebuah *scanner* sidik jari. Hasil yang didapat dari penelitian sebelumnya adalah sebuah sistem perumusan sidik jari dengan tingkat keberhasilan penentuan titik *core* 49.33%, penentuan titik *delta* 55.33%, penentuan *ridge counting* 7.78%, dan tingkat keberhasilan sistem dalam melakukan perumusan sidik jari sebesar 9.11%. Pada penelitian oleh Kurnia Rizqiani [5] telah dibahas mengenai perancangan perangkat lunak yang dapat melakukan perhitungan rumus sidik jari tipe *loop* dengan sampel sebanyak 25 citra sidik jari, tetapi tanpa menggunakan metode khusus dengan akurasi perumusan sidik jari sebesar 70%. Pada penelitian oleh Fila Harmuningtyas [6] telah dibahas mengenai penggunaan metode *Poincaré Index* dalam pendeteksian letak titik *core* pada sidik jari dengan sampel sebanyak 160 citra sidik jari dengan akurasi pendeteksian titik *core* sebesar 65% pada skenario nilai *Poincaré Index* 0.47-0.51. Simulasi sistem yang dibuat pada penelitian ini melakukan perumusan sidik jari tipe *loop* dengan data input berupa citra digital sidik jari *grayscale*. Perumusan sidik jari terbentuk dari informasi jenis tipe *loop*, dan jarak antara titik *core* dan *delta* yang didapatkan dari tahap deteksi titik *core* dan *delta* menggunakan metode *Poincaré Index* dan perhitungan jarak titik *core* dan *delta* menggunakan metode *euclidean distance*.

Fokus pada tugas penelitian ini adalah perumusan sidik jari berjenis *loop*, kemudian dilakukan pengujian dengan skenario pada variabel filter *gaussian* di tahap *pre-processing* yaitu kombinasi variabel *block size* bernilai 4, 5 dan 6, dan variabel *sigma* (σ) bernilai 2, 3, 4 dan 5. Proses pada sistem meliputi tahap *pre-processing*, penentuan titik *core* dan *delta*, perhitungan jarak titik *core* dan *delta*, dan perumusan sidik jari. Proses *pre-processing* terdiri dari tahap filter *Gaussian*, binerisasi, dan *spot removal*. *Output* dari simulasi sistem ini adalah hasil rumus sidik jari jenis *loop* dengan menggunakan metode *Poincaré Index*. Hasil dari penelitian ini adalah simulasi sistem dengan tingkat kesuksesan perumusan sidik jari sebesar 80.95% dan disimpulkan bahwa metode *Poincaré Index* dapat diimplementasikan untuk melakukan proses deteksi titik *core* dan *delta* pada sistem perumusan sidik jari, tetapi variasi dari nilai variabel filter *gaussian* dapat mempengaruhi tingkat kesuksesan sistem.

1.2 Tujuan

Tujuan dari penelitian ini adalah:

1. Merancang simulasi sistem perumusan sidik jari dengan algoritma yang tepat untuk melakukan perumusan sidik jari jenis *loop*.
2. Melakukan analisis tingkat keberhasilan perumusan sidik jari pada sistem tipe *loop* dengan menggunakan metode *Poincaré index*.

2. Konsep Dasar

Ilmu yang mempelajari tentang sidik jari adalah Daktiloskopi yang berasal dari kata *dactylos* yang berarti jari dan kata *scopein* yang berarti meneliti atau mengamati [5]. Daktiloskopi mengamati sidik jari yang merupakan alur garis pada kulit ujung jari manusia, dimana sidik jari tersebut dapat digunakan untuk mengidentifikasi atau mengenali identitas seseorang [6].

2.1 Sidik Jari

Sidik jari adalah suatu jejak yang ditinggalkan oleh permukaan kulit jari manusia secara sengaja atau tidak sengaja pada permukaan tertentu yang disentuh oleh jari manusia [1]. Jejak tersebut disebabkan oleh tonjolan garis rapat yang dipisahkan oleh celah yang memiliki alur tertentu yang disebut dengan *ridge* yang menyebabkan kulit jari manusia memiliki tekstur dan pola tertentu [2].

2.2 Sidik Jari Tipe *Loop*

Tipe sidik jari *loop* dapat ditemukan pada 60 – 70% dari populasi manusia. *Ridge* pada tipe ini membentuk lengkungan berbalik arah tetapi tidak terpilin. Tipe *loop* memiliki sub-kategori sebagai berikut [7]:

2.2.1 Radial *Loop*

Radial *loop* dinamai dari nama tulang lengan manusia yang satu sisi dengan tulang ibu jari bernama “radius” dimana arah alur dari lengkungan *ridge* searah dengan dengan tulang radius yang diilustrasikan pada gambar 1. Tipe ini tidak mudah ditemui tetapi cenderung berada pada jari telunjuk [7].



Gambar 1. Radial loop [8]

2.2.2 Ulnar *Loop*

Ulnar *Loop* dinamai dari nama tulang lengan manusia yang satu sisi dengan tulang jari kelingking bernama “Ulna” dimana arah alur dari lengkungan *ridge* searah dengan tulang ulna sebagaimana diilustrasikan pada gambar 2 [7].



Gambar 2. Ulnar loop [8]

2.3 Pemrosesan Citra Digital

Pemrosesan citra digital atau Digital image processing adalah salah satu sub-kategori dari pemrosesan sinyal digital dengan metode teknik pemrosesan gambar dalam bentuk digital menggunakan algoritma yang dijalankan oleh komputer. Pemrosesan citra digital memungkinkan penerapan jangkauan algoritma yang lebih luas pada data *input* dan dapat mengatasi masalah seperti noise dan distorsi pada pemrosesan sinyal [9].

2.4 MATLAB

MATLAB (Matrix laboratory) adalah suatu software pemrograman yang dikembangkan oleh Mathworks berbasis matriks yang digunakan para teknisi dan ilmuwan untuk keperluan komputasi matematika, analisa data, pengembangan algoritma, dan pembuatan aplikasi serta model saintifik dengan bahasa pemrograman C, C++, C#, Java, Fortran, dan Python [10]. Meskipun Matlab dapat digunakan untuk perhitungan numerik, Matlab juga dapat digunakan untuk perhitungan simbolik, serta memiliki fitur interface grafis yang dapat digunakan sebagai model dan representasi citra.

2.5 Perumusan Sidik Jari

Perumusan sidik jari adalah metode yang digunakan untuk mengidentifikasi karakteristik dari sidik jari seseorang. Titik fokus pada sidik jari adalah titik *core* dan titik *delta*. Titik *core* adalah titik tengah yang terdapat pada sidik jari yang terdalam dan terjauh dari titik *delta*, sedangkan titik *delta* adalah titik fokus yang terletak pada berpisahannya alir garis sidik jari di sisi kanan atau kiri titik *core* [3]. Tahap yang dilakukan untuk melakukan perumusan yaitu mencari titik *delta* dan titik *core*, dan menghitung jarak kedua titik tersebut. Rumus sidik jari meliputi deretan simbol yang melambangkan jenis sidik jari, hasil dari perhitungan jarak antara *core* dan *dela*. Pada sidik jari tipe *loop*, digunakan simbol *RL* untuk tipe *radial loop* dan simbol *UL* untuk tipe *ulnar loop*. Sebagai

contoh, jari telunjuk dengan sidik jari tipe *ulnar loop* dan jarak antara *core* dan *delta* bernilai 100 *pixel* memiliki rumus sidik jari sebagai berikut = UL_{100} .

2.6 Euclidean distance

Euclidean distance atau euclidean metric adalah jarak diantara dua titik pada euclidean space. Euclidean space pertama kali ditemukan oleh ilmuwan matematika Yunani bernama Elucid pada tahun 300 SM untuk mempelajari tentang hubungan antara sudut dan jarak. Sistem geometri ini masih digunakan hingga saat ini terutama pada bidang 2 dimensi dan ruang 3 dimensi, tetapi dapat digunakan juga pada dimensi yang lebih tinggi [11].

Perhitungan jarak euclidean pada dimensi satu dilakukan dengan operasi akar dari kuadrat pada pengurangan posisi dari kedua titik pada posisi yang berbeda. Operasi kuadrat yang diakarkan kembali dilakukan agar nilai absolut didapatkan, sehingga nilai jarak valid [12].

Perhitungan jarak euclidean pada dimensi dua dilakukan dengan cara yang sama tetapi dengan menggunakan set nilai yang terdiri dari dua nilai pada setiap sumbu yang menjelaskan tentang posisi atau koordinatnya sebelum dilakukan operasi kuadrat. Setelah itu dilakukan operasi penjumlahan antara kedua sumbu tersebut dan dilakukan operasi akar. Tahap yang dilakukan pada dimensi dua dapat digunakan pada dimensi yang lebih tinggi dari 2 dengan perbedaan bentuk nilai posisi dimana jumlah sumbu sama dengan tingkat dimensinya [11].

Pada koordinat kartesian, jika $\mathbf{p} = (p_1, p_2, \dots, p_n)$ dan $\mathbf{q} = (q_1, q_2, \dots, q_n)$ adalah dua titik pada bidang atau ruang euclidean dimensi- n , maka jarak (d) dari \mathbf{p} ke \mathbf{q} atau sebaliknya dijelaskan dengan rumus *pythagoras* berikut [11]:

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \quad (1)$$

$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$

2.7 Filter Gaussian

Filter *Gaussian* adalah teknik filter *blur* yang membuat transisi warna pada sebuah gambar menjadi gradien warna yang lebih lembut. *Gaussian* filter adalah sebuah filter yang menggunakan persamaan matematika untuk menciptakan efek kabur atau berkabut. Istilah *gaussian* diambil dari nama seorang matematikawan Jerman bernama *Karl Friedrich Gauss* [13]. Filter *gaussian* dapat dilakukan dengan persamaan berikut [14]:

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (2)$$

dimana x adalah jarak pada sumbu horizontal, y adalah jarak pada sumbu vertikal, dan σ adalah deviasi standar pada distribusi *gaussian*.

2.8 Local Orientation Estimation

Nilai orientasi gambar dihitung dengan tujuan untuk meningkatkan kualitas gambar dengan metode *least mean square estimation* yang berpusat pada *pixel*. Nilai *least square estimation* dihitung dengan menggunakan persamaan berikut [15]:

$$V_x(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u, v)\partial_y(u, v) \quad (3)$$

$$V_y(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \partial_x^2(u, v)\partial_y^2(u, v) \quad (4)$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{V_y(i, j)}{V_x(i, j)} \right), \quad (5)$$

dimana $\theta(i, j)$ merupakan variabel untuk nilai *least square estimation* dari orientasi blok yang berpusat pada *pixel* dengan koordinat (i, j) .

2.9 Poincaré index

Dalam ilmu matematika *Poincaré index* (juga dikenal sebagai *Poincaré–Hopf theorem*) adalah teorema yang digunakan dalam topologi diferensial yang dinamai dari *Henri Poincaré* dan *Heinz Hopf*. Teorema ini dapat diilustrasikan dengan sebuah kasus dimana suatu bola memiliki vektor area yang tidak memiliki sumber maupun ujung [16]. *Poincaré index* menunjukkan perubahan dari rotasi orientasi area sekitar sebuah titik pusat dengan arah berlawanan arah jarum jam [4]. *Poincaré index* didapat dari hasil perhitungan orientation field estimation. Jika nilai *poincaré index* menghasilkan (-0.5) maka terdeteksi titik core sedangkan, jika menghasilkan (0.5) terdeteksi titik delta [4]. Persamaan umum dari metode *Poincaré index* adalah sebagai berikut [18]:

$$PC(i, j) = \frac{1}{2\pi} \sum_{k=0}^{Np} \Delta(k). \quad (6)$$

$$\Delta(k)(i, j) = \begin{cases} \delta(k) & \text{if } |\delta(k)| < \pi/2 \\ \pi + \delta(k) & \text{if } |\delta(k)| \leq -\pi/2 \\ \pi - \delta(k) & \text{if } |\delta(k)| \geq \pi/2 \end{cases} \quad (7)$$

$$\delta(k) = \theta' \left(x_{(k+1) \bmod Np}, y_{(k+1) \bmod Np} \right) - \theta' (x_k, y_k), \quad (8)$$

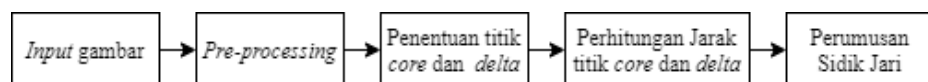
dimana $PC(i, j)$ adalah *poincaré index* (x, y) , $\delta(k)$ adalah standar deviasi, θ' adalah sudut arah hasil perhitungan dari orientation field estimation, dan Np adalah nilai partikular.

3. Model Sistem dan Perancangan

Model sistem yang dirancang pada penelitian ini adalah sistem perumusan sidik jari berjenis *loop* dengan metode *poincaré index*. *Input* dari sistem ini berupa citra digital dan *output* sistem berupa rumus sidik jari jenis *loop*.

3.1 Diagram Blok Prototype Sistem

Terdapat beberapa langkah yang dilakukan sebelum melakukan perumusan sidik jari, seperti *input* gambar, *pre-processing*, penentuan titik *core* dan *delta*, dan perhitungan jarak titik *core* dan *delta*. Berikut diagram blok dari prototype sistem:



Gambar 3. Diagram blok *prototype* sistem

Pada diagram blok diatas, tahap pertama yang dilakukan adalah *input* gambar. Gambar *input* diambil dari *database* biometrik *NIST Special Database* dengan format *grayscale* [19]. Jari yang digunakan sebagai *sample* adalah jari tangan kanan dan jari tangan kiri.

Tahap ini terdiri dari beberapa proses, yaitu filter *Gaussian* untuk memberikan efek *blur* yang memperhalus garis-garis pada sidik jari, binerisasi untuk mengubah format gambar sidik jari menjadi biner, *spot removal* untuk menghilangkan titik-titik pada permukaan garis-garis sidik jari, dan *local orientation estimation* untuk mengetahui nilai orientasi pada gambar sidik jari

Tahap ketiga adalah penentuan titik *core* dan *delta*. Pada tahap ini digunakan metode *poincaré index* untuk menentukan posisi titik *core* dan *delta* berdasarkan perubahan arah orientasi dikarenakan lengkungan *ridge* yang ada pada daerah *core* dan *delta*.

Tahap keempat adalah perhitungan jarak antara titik *core* dan *delta*. Tahap ini dilakukan setelah posisi titik *core* dan *delta* didapatkan. Jarak dihitung berdasarkan koordinat titik *core* dan *delta* menggunakan metode *euclidean distance*.

Tahap kelima adalah perumusan sidik jari. Tahap ini dilakukan setelah posisi *delta* diketahui dan jumlah *ridge* antara titik *core* dan *delta* diketahui. Rumus sidik jari dibentuk dari informasi tipe sub-kategori sidik jari pada jenis *loop* dan jarak antara titik *core* dan *delta* dalam *pixel*.

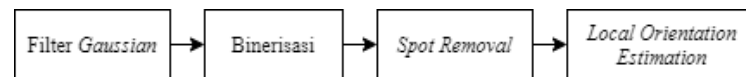
3.1.1 Input Gambar

Tahap yang harus dilakukan untuk melakukan perumusan sidik jari dimulai dengan pengambilan citra digital dari sidik jari yang akan diproses. Gambar yang digunakan adalah

gambar sidik jari yang tidak memiliki cacat atau kerusakan pada permukaannya. Gambar sidik jari diambil dari sumber data sekunder yang berupa dataset sidik jari dari *database* biometrik *NIST Special Database* [19]. Gambar yang ada pada *database* tersebut terdiri dari gambar sidik jari dari beragam tipe, maka dari itu peneliti hanya mengambil gambar sidik jari dengan tipe *loop* sesuai dengan judul penelitian.

3.1.2 Pre-processing

Pre-Processing adalah tahap pertama yang dilakukan setelah *input* gambar. Tahap ini dilakukan untuk mempersiapkan gambar untuk tahap perumusan sidik jari. Tahap ini terdiri dari beberapa langkah terurut yang diilustrasikan pada gambar diagram blok 4.



Gambar 4. Diagram blok *pre-processing*

Berikut penjelasan dari setiap blok pada diagram blok *pre-processing*

3.1.2.1 Filter Gaussian

Tahap ini dilakukan untuk memberikan efek kabur kepada gambar *input*. Ini dilakukan untuk menghaluskan tepian garis-garis sidik jari. Tahap ini adalah proses yang penting untuk tahap selanjutnya. Proses ini dilakukan agar *pixel* yang membentuk tepi *ridge* menjadi lebih halus. Pada tahap ini terdapat dua variabel yang akan digunakan sebagai skenario pengujian dalam perumusan sidik jari, yaitu *block size* dan *sigma* (σ).

3.1.2.2 Binerisasi

Tahap ini dilakukan untuk mengubah format gambar *grayscale* menjadi gambar dengan format *biner* dimana bagian *ridge* menjadi hitam dan *valley* menjadi putih. Proses ini dapat menghilangkan elemen gradasi abu-abu pada tepian garis-garis sidik jari sehingga alur garis menjadi lebih halus. Proses ini dilakukan untuk mengkilangkan efek kabur pada filter *Gaussian* tetapi tetap menjaga struktur tepi *ridge* yang sudah dihaluskan oleh filter *Gaussian*.

3.1.2.3 Spot Removal

Tahap ini adalah proses eliminasi titik-titik pada *ridge* dan *valley* di permukaan gambar sidik jari. Ini dilakukan dengan tujuan untuk membersihkan gambar dari elemen-elemen yang tidak diperlukan. Proses ini dilakukan untuk menghilangkan *pixel-pixel* kecil yang membentuk bintik-bintik putih serta lubang-lubang kecil pada *ridge*.

3.1.2.4 Local Orientation Estimation

Tahap ini melakukan orientasi daerah *ridge* pada gambar. Tujuan dari langkah ini adalah untuk mengetahui nilai sudut arah dari *orientation fields*.

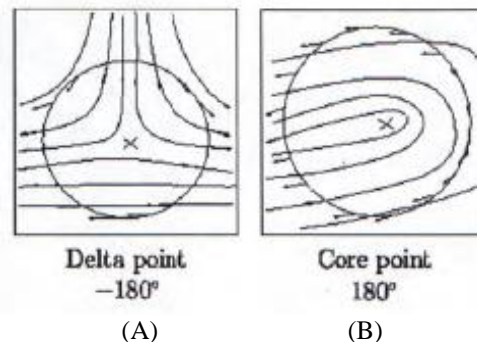
Tahap ini memiliki beberapa tahapan proses. Pertama gambar dibagi menjadi beberapa blok dengan ukuran tertentu. Setelah itu nilai gradien dihitung pada setiap *pixel* di dalam blok tersebut dalam bentuk matriks. Terdapat beberapa langkah pada proses *local orientation estimation*. Langkah pertama yaitu membagi gambar *input* menjadi blok-blok berukuran 3x3. Kemudian nilai gradien dihitung untuk setiap *pixel* yang ada di dalam blok tersebut. Selanjutnya mengestimasi orientasi lokal masing-masing blok dengan persamaan *local orientation estimation*. Selanjutnya hitung orientasi lokal *ridge* pada *pixel* koordinat (i,j) dengan persamaan *local orientation estimation* [15].

3.1.3 Penentuan titik *core* dan *delta*

Titik *core* dan *delta* dideteksi dengan tujuan pengukuran jarak antara titik *core* dan *delta*. Tahap ini dilakukan setelah mengetahui arah dari orientasi karena metode yang digunakan bergantung kepada proses tersebut. Apabila sistem *error* pada tahap ini maka sistem tidak akan memberikan *output* rumus sidik jari.

Metode yang digunakan adalah *Poincare Index*, dimana metode ini bertujuan untuk mendeteksi *point* seperti *core* dan *delta*. Titik *core* dan *delta* disebut *point* karena arah dari orientasi pada area tertentu menjadi terputus-putus yang diakibatkan karena lengkungan ekstrim *ridge* yang membentuk pola *core* dan *delta* [16]. Pendekatan yang digunakan dalam mencapai tujuan metode tersebut adalah *Poincaré Index*. Ketika nilai dari perubahan arah orientasi sudah

diketahui, maka posisi titik *core* dan *delta* dapat diketahui. Pada sidik jari tipe *loop*, sebuah *core Poincaré index* 180 derajat, sedangkan *delta* memiliki *Poincaré index* -180 derajat [16] yang diilustrasikan pada gambar 5.



Gambar 5. (A) Perubahan arah orientasi pada area *delta* dan (B) perubahan arah orientasi pada area *core*

3.1.4 Perhitungan Jarak

Tahap perhitungan jarak dilakukan setelah posisi *delta* dan *core* diketahui. Tahap ini dilakukan untuk mengetahui nilai jarak diantara titik *core* dan *delta*. Nilai jarak ini diperlukan sebagai salah satu bagian pembentuk rumus sidik jari. Berikut ilustrasi perhitungan jarak antara titik *core* dan *delta*. Jarak antara *core* dan *delta* adalah jumlah dari banyaknya *pixel* yang ada diantara *core* dan *delta*. Jarak *core* dan *delta* dihitung menggunakan rumus *Pythagoras*. Dengan menggunakan informasi nilai jarak sumbu x *delta* dan nilai jarak sumbu y *core*, maka jarak diagonal antara *core* dan *delta* dapat diketahui. Apabila sistem *error* pada tahap ini maka sistem tidak akan memberikan *output* rumus sidik jari.

3.1.5 Perumusan Sidik Jari

Tahap ini dilakukan setelah jarak *delta* dan *core* didapatkan serta setelah jarak antara titik *core* dan *delta* didapatkan. Perumusan sidik jari dilakukan dengan menggabungkan informasi posisi *delta* dan jarak antara *core* dan *delta*.

Rumus sidik jari terdiri dari informasi tentang jenis sidik jari dan jarak antara *core* dan *delta*. Jarak antara *core* dan *delta* dirumuskan setelah simbol tipe sidik jari. Format perumusan adalah “tipe_jarak”, dengan elemen “tipe” adalah tipe dari sidik jari pada jenis *loop* yaitu *RL* sebagai *radial loop* dan *UL* sebagai *ulnar loop*, dan elemen “jarak” adalah jarak antara titik *core* dan *delta* dalam *pixel*. Rumus yang akan dibuat berbentuk “UL_jarak” untuk tipe *ulnar loop* dan “RL_jarak” untuk tipe *radial loop*.

Tipe *radial* dan *ulnar* dapat dibedakan dengan cara membandingkan posisi *delta* terhadap posisi *core* secara horizontal. Elemen jarak pada rumus menggunakan data yang didapatkan dari proses perhitungan jarak antara *core* dan *delta*. Apabila nilai sumbu x koordinat *delta* lebih kecil daripada nilai sumbu x koordinat *core*, maka sidik jari dinyatakan sebagai *RL* atau *radial loop*, sebaliknya jika dinyatakan sebagai *UL* atau *ulnar loop*. Apabila nilai sumbu x koordinat *delta* sama dengan nilai sumbu x *core*, maka dinyatakan tidak valid karena posisi kedua titik tersebut tidak boleh sejajar secara vertikal. Sebagai contoh, apabila sebuah citra sidik jari terdeteksi sebagai tipe *ulnar* dengan jarak antara titik *core* dan *delta* sebanyak 100 *pixel*, maka sistem akan memberikan *output* rumus *UL_100*. Sistem tidak akan memberikan *output* rumus sidik jari apabila sistem *error* pada salah satu proses yang ada.

3.3 Pengukuran Performansi Kinerja Sistem

Pada penelitian ini performansi kinerja sistem diukur dari tingkat kesuksesan sistem dalam melakukan perumusan sidik jari pada setiap citra sidik jari. Perumusan sidik jari adalah tahap terakhir yang ada pada sistem. Perumusan sidik jari dinyatakan sukses ketika sistem tidak mengalami *error* pada semua proses yang ada pada sistem dan sistem dapat memberikan *output* berupa rumus sidik jari. Tingkat kesuksesan sistem dihitung dengan persamaan sebagai berikut:

$$\text{Tingkat Kesuksesan} = \frac{n}{\text{Jumlah Data Keseluruhan}} \times 100\%, \quad (9)$$

dengan n sebagai jumlah citra sidik jari yang dinyatakan sukses dalam perumusan sidik jari.

4. Hasil dan Analisis

4.1 Spesifikasi Sistem

Perancangan simulasi sistem membutuhkan perangkat keras dan lunak dengan spesifikasi tertentu agar simulasi sistem dapat dijalankan dengan baik. Adapun uraian spesifikasi *hardware* dan *software* yang digunakan dalam perancangan sistem.

4.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Laptop ASUS
 - a. *Processor* : AMD RYZEN 5
 - b. *RAM* : 8 GB
 - c. *Harddisk* : 1 TB
 - d. *GPU* : AMD Radeon Vega 8

4.1.2 Perangkat Lunak

Terdapat beberapa perangkat lunak yang digunakan dalam perancangan sistem pada penelitian ini, yaitu:

1. MATLAB R2019a
2. OS Windows 10 64-bit
3. Microsoft Word 2010
4. Microsoft Excel 2010

4.2 Pengujian Sistem

Pengujian sistem dilakukan menggunakan 100% dari *dataset*. Pengujian berfokus kepada perumusan sidik jari dengan tambahan skenario yang terletak pada variasi nilai variabel filter *gaussian* pada tahap *pre-processing*.

4.2.1 Skenario Pengujian

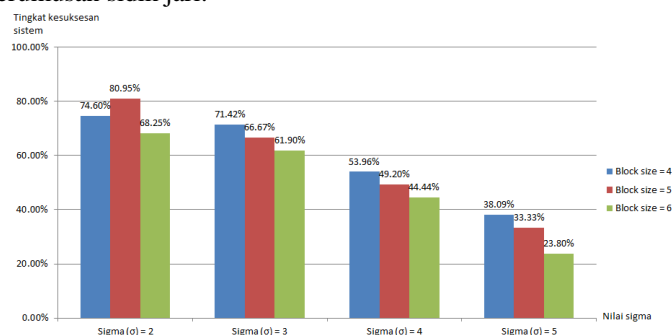
Skenario pada pengujian sistem menggunakan gabungan dari 2 variabel pada filter *gaussian*. Perumusan sidik jari akan mewakili tingkat keberhasilan seluruh sistem. Nilai dari variabel yang akan digunakan sebagai pengujian perumusan sidik jari adalah *block size* bernilai 2, 3 dan 4, dan *sigma* (σ) bernilai 4, 5 dan 6. Tingkat keberhasilan dari seluruh sistem sendiri adalah tingkat keberhasilan perumusan sidik jari dengan nilai variabel skenario yang paling optimal.

4.2.2 Hasil Pengujian Perumusan Sidik Jari

Pada tahap pengujian sistem digunakan sebanyak 63 gambar sidik jari dengan tipe *loop*. Sistem dapat dikatakan berhasil apabila sistem dapat menentukan titik *core* dan *delta*, mengukur jarak antara *core* dan *delta*, dan melakukan perumusan sidik jari.

Pengujian sistem hanya dilakukan pada tahap perumusan karena *output* dari sistem hanya bisa didapatkan ketika tahap perumusan berhasil dilakukan, sementara itu tingkat keberhasilan tahap perumusan bergantung sepenuhnya kepada tahap sebelumnya yaitu penentuan titik *core* dan *delta* dan tahap perhitungan jarak antara titik *core* dan *delta*.

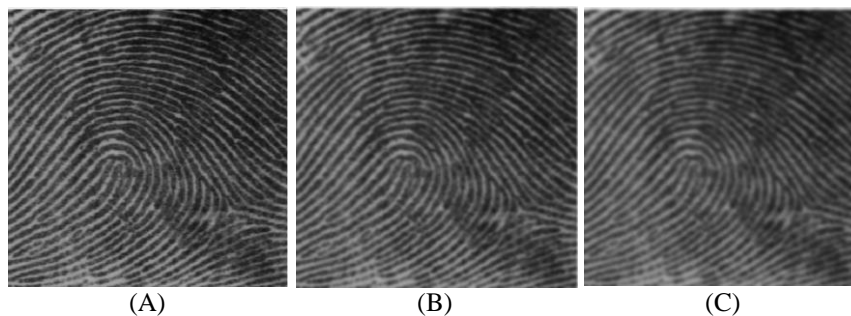
Skenario pengujian yang dilakukan adalah gabungan dari 2 variabel filter *gaussian* pada tahap *pre-processing* yaitu nilai *block size* dan nilai *sigma* (σ). Nilai *block size* yang digunakan adalah 2, 3 dan 4, sedangkan nilai *sigma* (σ) yang digunakan adalah 4, 5, dan 6. Berikut grafik hasil pengujian perumusan sidik jari:



Gambar 6. Grafik Hasil Pengujian Perumusan Sidik Jari

Pada grafik diatas, sumbu y menyatakan tingkat kesuksesan sistem dalam perumusan sidik jari oleh sistem, sedangkan sumbu x menyatakan nilai variabel *Sigma* (σ) dari skenario filter *gaussian*. Setiap blok yang ada pada grafik diatas menggambarkan variasi nilai dari *blok size*. Berdasarkan grafik diatas, didapatkan analisis sebagai berikut:

1. Nilai variabel yang paling optimal pada filter *gaussian* untuk mencapai persentase kesuksesan tertinggi pada perumusan sidik jari adalah *blok size* dengan nilai 5 dan *Sigma* (σ) dengan nilai 2 dengan jumlah gambar yang sukses dalam perumusan sidik jari yaitu 51 dari 63 gambar dengan persentase sebesar 80,95%. Nilai variabel tersebut dipilih karena memiliki tingkat kesuksesan pada tahap perumusan sidik jari yang paling tinggi dibandingkan dengan variasi nilai variabel yang lainnya.
2. Perbedaan nilai *blok size* dan *Sigma* (σ) pada filter *gaussian* mempengaruhi citra sidik jari dengan memberikan efek kabur dengan intensitas yang berbeda pada citra di tahap *pre-processing*. Nilai *blok size* dan *Sigma* (σ) yang terlalu kecil menyebabkan citra sidik jari menjadi terlalu kasar, sedangkan nilai *blok size* dan *Sigma* (σ) yang terlalu besar menyebabkan citra sidik jari menjadi terlalu kabur. Ini mengakibatkan sistem kesulitan dalam tahap penentuan titik *core* dan *delta* sehingga terdapat lebih banyak *error* pada sistem dan proses perumusan sidik jari tidak dapat dilakukan. Berikut contoh gambar hasil *pre-processing* dengan nilai variabel filter *Gaussian* yang terlalu rendah, optimal dan terlalu tinggi:



Gambar 7. (A) *blok size* = 4 dengan *Sigma* (σ) = 2, (B) *blok size* = 5 dengan *Sigma* (σ) = 2 dan (C) *blok size* = 6 dengan *Sigma* = 5 (σ).

4. Kesimpulan dan Saran

5.1 Kesimpulan

Dari hasil pengujian yang diperoleh serta analisis pada Perumusan Sidik Jari Berjenis *Loop* Menggunakan Metode *Poincaré Index* maka dapat diambil kesimpulan sebagai berikut:

1. Tingkat keberhasilan keseluruhan sistem adalah sebesar 80.952% dimana tingkat keberhasilan ini lebih besar daripada tingkat keberhasilan pada penelitian yang sebelumnya yaitu 9.11%, maka dari itu dinyatakan tingkat keberhasilan sistem perumusan sidik jari dapat ditingkatkan menggunakan metode *Poincaré index*.
2. Nilai dari variabel filter *Gaussian* yang paling optimal adalah *blok size* dengan nilai 5 dan *Sigma* (σ) dengan nilai 2. Kombinasi dari kedua nilai ini pada filter *Gaussian* menghasilkan tingkat keberhasilan perumusan yang paling baik.

5.2 Saran

Pada penelitian ini terdapat sejumlah kekurangan yang dapat ditingkatkan kembali untuk pengembangan berikutnya. Berikut saran yang diberikan oleh peneliti:

1. Penelitian dapat dikembangkan dengan meneliti semua jenis sidik jari sehingga dapat menghasilkan rumus sidik jari dengan standar kepolisian.
2. Penambahan tahap *image enhancing* sebelum tahap *pre-processing* agar didapatkan kualitas citra sidik jari yang lebih baik.
3. Diharapkan untuk selanjutnya dapat mengembangkan penelitian ini lebih jauh dan dapat dikembangkan menjadi alat yang mampu mempermudah proses perumusan sidik jari.

Daftar Pustaka

- [1] J. Hemanth and V. E. Balas, "*Biologically rationalized computing techniques for image processing applications*". Springer International Publishing, 2017.
- [2] S. L. Afifah¹, M. T., Mohammad Ramdhani, and M. T., Rita Purnamasari, "Perumusan Sidik Ibu Jari Berjenis *Loop* dengan Menggunakan Metode WDFs" e-Proceeding of Engineering, vol. 4, no. 2, 2017.
- [3] K. Rizqiani, M. Ramdhani, A. Rizal. "Perancangan Perangkat Lunak Penghitungan Rumus Sidik Jari Tipe Loop". Prosiding SENTIA, 2009.
- [4] F. Harmuningtyas, I. Agustien and F. Damayanti, "Penggunaan Metode *Poincaré index* Dalam Pendeteksian Letak *Corepoint* Pada Sidik Jari," KNTIA, 2011.
- [5] Misbach, Ifa H. "Dahsyatnya Sidik Jari". VisiMedia, 2010.
- [6] P. Yudha. "Peran Daktiloskopi dalam Mengungkapkan Kasus Tindakan Pidana Pencurian (Studi Kasus di Wilayah Hukum Polres Sragen)". UNS Digital Library, 2011.
- [7] A. Davis, "8 Most Common Fingerprint Patterns", Feb 13, 2017. Accessed on: Oct 23, 2019. [Online]. Available: <https://www.touchngoid.com/8-common-fingerprint-patterns/>
- [8] Federal Bureau of Investigation, "*The Science of Fingerprints Classification and Uses*". Empire Books, 1993.
- [9] P. Chakravorty, "What Is a Signal? [Lecture Notes]," IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 175-177, Sept, 2018.
- [10] The MathWorks, "*Company Overview*", Apr 2018.
- [11] R. Allan. "*How to Calculate Euclidean Distance*" sciencing.com. Accessed On: 26 Nov, 2019. [Online]. Available: <https://sciencing.com/how-to-calculate-euclidean-distance-12751761.html>.
- [12] A. Howard, "*Elementary Linear Algebra*" (7th ed.), John Wiley & Sons, pp. 170–171, 1994.
- [13] W. Andre, L. S. Herlina, S. H. Yanolanda. "*Comparative Analysis Of Gaussian, Mean and Median of Noise Reduction*". Jurnal Media Infotama Vol. 12 No. 1, Feb 2016.
- [14] L. G. Shapiro, G. C. Stockman "*Computer Vision*", page 137, 150. Prentice Hall, 2001
- [15] M. H. N. H. Khan, M. A. N. Momin, and M. S. Shaikh. "*Fingerprint Enhancement Using Different Techniques*". AIKTC, 2016.
- [16] Hazewinkel. Michiel, ed. (2001), "*Poincaré–Hopf theorem*". Springer Science+Business Media, 1994.
- [17] K. C. Chan, Y.S. Moon, P. S. Cheng, "*Fast Fingerprint Verification using Sub-regions of Fingerprint Images*". IEEE, 30 Jan, 2004.
- [18] J. Bo, T. H Ping, X. M. Lan "Fingerprint Singular Point Detection Algorithm by *Poincaré Index*". WSEAS TRANSACTIONS on SYSTEMS, Issue 12. Vol. 7: 1453-1462, 2008.
- [19] Standard Reference Data Program, "*NIST 8-Bit Gray Scale Images of Fingerprint Image Groups (FIGS)*", National Institute of Standards and Technology, Jul, 2018. Accessed on: Apr 21, 2020. [Online]. Available: <https://www.nist.gov/srd/nist-special-database-4/>