

PENGENALAN SIDIK JARI MANUSIA TERDISTORSI MENGGUNAKAN ALGORITMA SURF (*SPEEDED-UP ROBUST FEATURE EXTRACTION*)

RECOGNITION OF DISTORTED HUMAN FINGERPRINT USING SURF ALGORITHM (*SPEEDED-UP ROBUST FEATURE EXTRACTION*)

Enita Ambarwati¹, Anggunmeka Luhur Prasasti, S.T., M.T.² Ashri Dinimaharawati S.Pd., M.T.³

^{1,2,3}Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

¹enitaambarwati@student.telkomuniversity.ac.id, ²anggunmeka@telkomuniversity.ac.id, ³ashridini@telkomuniversity.ac.id

Abstrak

Salah satu sistem pengenalan identitas yang sudah berkembang adalah pengenalan identitas melalui sidik jari. Namun, dikarenakan oleh kondisi kulit yang berbeda-beda, terkadang sistem sidik jari tidak dapat melakukan identifikasi dengan baik. Beberapa permasalahan kulit yang umum terjadi yang mempengaruhi identifikasi melalui sidik jari antara lain kondisi kulit berminyak, kulit kering, dan kulit pecah-pecah.

Pada tugas akhir kali ini, telah dirancang suatu sistem identifikasi sidik jari untuk pola sidik jari yang terdistorsi berdasarkan pengolahan citra. Sistem sidik jari ini merupakan sistem pengenalan sidik jari dengan tingkat keakurasian dan kecepatan yang lebih tinggi. Sistem tersebut akan mengimplementasikan metode *Speeded-Up Robust Feature Extraction* dan *Brute Force Matching*. Hasilnya menunjukkan bahwa sistem sidik jari yang dikembangkan dalam penelitian ini berhasil mengenali sidik jari dengan tingkat akurasi 83.30% dan mampu melakukan pengenalan sidik jari dengan rata-rata total *respon time* sebesar 1.232 detik/kondisi.

Kata kunci: *Brute Force Matching*, Pengenalan Sidik Jari, *Speeded-Up Robust Feature Extraction*.

Abstract

One of the identity recognition systems that has developed is identity recognition through fingerprints. However, due to different skin conditions, sometimes the fingerprint system cannot properly identify it. Some of the common skin problems that affect fingerprint identification include oily skin, dry skin, and cracked skin.

In this final project, a fingerprint identification system has been designed for distorted fingerprint patterns based on image processing. This fingerprint system is a fingerprint recognition system with a higher level of accuracy and speed. The system will implement the Speeded-Up Robust Feature Extraction and Brute Force Matching methods. The results show that the fingerprint system developed in this study has successfully recognized fingerprints with an accuracy rate of 83.30% and is able to perform fingerprint recognition with an average total response time of 1.232 seconds/condition.

Keyword: *Brute Force Matching*, *Fingerprint Recognition*, *Speeded-Up Roust Feature Extraction*.

1. Pendahuluan

Di Indonesia, salah satu teknologi biometrik yang telah berkembang selama empat puluh tahun terakhir adalah sistem sidik jari[1]. Dalam penerapannya, sistem sidik jari telah banyak dijumpai pada berbagai aplikasi sistem absensi dan keamanan. Namun, sistem sidik jari masih mengalami masalah penelitian untuk mengenali sidik jari berkualitas rendah[1] seperti perbedaan karakteristik kulit terdistorsi antara lain kulit berminyak, kulit kering, dan kulit pecah-pecah. Hal ini sangat mungkin bahwa sistem sidik jari belum dirancang untuk menangani kulit yang terdistorsi. Oleh karena itu, menjadi permasalahan setiap individu yang memiliki kulit distorsi untuk mendeteksi sidik jarinya.

Metode yang digunakan dalam penelitian ini adalah SURF (*Speeded-Up Robust Feature*) dan *Brute Force Matching*. SURF digunakan untuk mengekstraksi fitur lokal kuat dimana dapat mengekstraksi dengan adanya rotasi, penskalaan, blurring, dan perubahan bentuk[2]. *Brute Force* merupakan pendekatan yang tidak beraturan untuk memecahkan suatu masalah (*straight forward*)[3]. Algoritma *brute force* memiliki kelebihan dibandingkan dengan

algoritma lain, yaitu dapat digunakan untuk memecahkan hampir sebagian besar masalah (*wide applicability*). Dengan penelitian ini, diharapkan dapat membuat sistem untuk mendeteksi sidik jari terdistorsi.

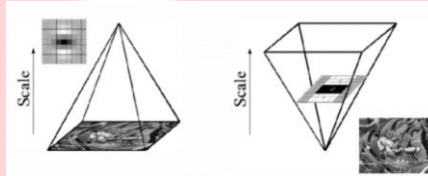
2. Metodologi Penelitian

2.1 Speeded-Up Robust Feature (SURF) Extraction

Tahap-tahap metode SURF adalah sebagai berikut:

1. Scale Space Representation

Dengan ukuran citra yang berbeda-beda, akan sangat sulit untuk membandingkan fitur yang terdapat pada citra tersebut, sehingga diperlukan proses untuk menangani perbedaan ukuran dengan cara menggunakan metode perbandingan skala[2].



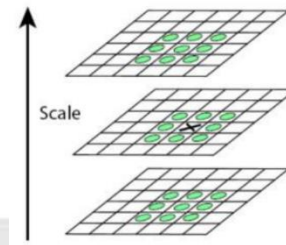
Gambar 1. Scale Space Representation[2]

Pada *scale space*, citra diimplementasikan dalam bentuk sebuah *image pyramid*[4]. Citra secara berulang akan diperhalus dengan fungsi *gaussian* dan dengan cara sub sampling untuk mencapai tingkat tertinggi *pyramid*. Dengan menggunakan *integral image*, perhitungan ini tidak perlu dilakukan secara iteratif dengan menggunakan filter yang sama, tetapi dapat juga menggunakan filter dengan ukuran sembarang ke dalam beberapa skala citra yang berbeda.

2. Lokalisasi Keypoint

Mencari lokasi *keypoint* yang invariant terhadap rotasi terdiri dari 3 proses, yaitu[2]:

- a. Menentukan nilai ambang atau batas penentuan lokasi. Ketika nilai ambang dinaikkan, jumlah *keypoint* yang terdeteksi lebih kecil dan sebaliknya. Oleh karena itu, nilai ambang bisa disesuaikan di setiap penelitian.
- b. Proses *non-maxima suppression* yang bertujuan untuk mencari sekumpulan *keypoint*.



Gambar 2. Non-Maxima Suppression[5]

Gambar 2 di atas, membandingkan setiap *pixel* citra pada *scale space 26-neighbours* yang terdiri 8 titik di *scale* asli dan 9 titik di *scale* atas dan bawah. Hasil dari *non-maxima suppression* adalah *keypoint* yang dihasilkan citra.

- c. Proses pencarian *keypoint* dengan menggunakan interpolasi data yang dekat dengan *keypoint*. Proses ini dilakukan dengan mencocokkan *quadratic* 3 dimensi[6]. Determinan *Hessian* yang dinyatakan dengan $H(x, y, \sigma)$ didefinisikan sebagai berikut:

$$H(x) = H \frac{\partial H^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 H}{\partial x^2} \quad (1)$$

Lokasi ekstrim yang diinterpolasi, didapatkan dengan mencari turunan dari fungsi $\hat{x} = (x, y, \sigma)$ dan diberi nilai nol, dijelaskan sebagai berikut:

$$\hat{x} = \frac{\partial^2 H^{-1}}{\partial x^2} \frac{\partial H}{\partial x} \quad (2)$$

Jika \hat{x} lebih besar dari 0.5 pada x,y atau arah σ , lokasi perlu di sesuaikan dengan interpolasi kembali. Prosedur ini diulang sampai nilai \hat{x} kurang dari 0.5. Bila interpolasi telah dilakukan beberapa kali namun hasilnya tidak memenuhi nilai di bawah 0.5, maka *keypoint* tersebut dihapus. *Keypoint* yang tersisa adalah yang stabil.

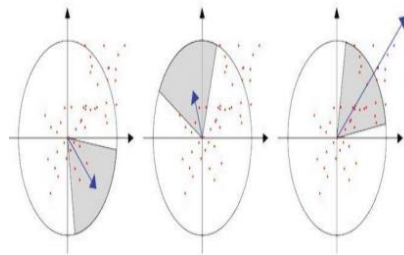
3. Deskripsi Fitur

Deskriptor ini menggambarkan distribusi intensitas *pixel neighbours* di sekitar *keypoint*. Pada SURF, dihitung distribusi *respon wavelet haar* order pertama dalam arah x dan y serta tidak menggunakan *gradient* seperti yang dilakukan SIFT. Selain itu, SURF menggunakan integral gambar untuk meningkatkan kecepatan proses. Hal tersebut bertujuan untuk mengurangi waktu komputasi pada proses ekstraksi dan *matching* fitur[7]. Berikut ini dua langkah perhitungan deskriptor *keypoint* yang akan dijelaskan lebih detail[2].

a. Pembuatan Orientasi

Langkah-langkah yang dilakukan sebagai berikut[2]:

1. Menghitung *Respon Haar Wavelet* dalam arah x dan y pada daerah *pixel* yang berbentuk lingkaran dengan jarak 6s, dengan s merupakan *scale* dari *keypoint* yang terdeteksi. Keluaran yang dihasilkan direpresentasikan sebagai titik dalam ruang dengan *respons horizontal* sepanjang sumbu absis dan *respons vertical* sebagai sumbu koordinat.
2. Membuat orientasi dominan dengan menghitung jumlah seluruh *respons* dalam pergeseran orientasi dengan ukuran window $\frac{\pi}{3}$,

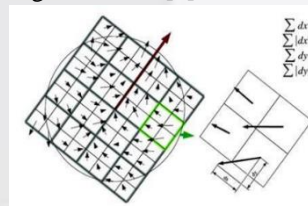


Gambar 3. Pergeseran Orientasi[2]

Gambar 3 di atas, menjelaskan penjumlahan *respons horizontal* dan *vertical*. Dari kedua *respons* yang dijumlahkan tersebut akan menghasilkan sebuah *vector* orientasi lokal. *Vector* terpanjang mendefinisikan orientasi *keypoint*.

b. Deskriptor Berdasarkan Jumlah *Respon Haar Wavelet*

Untuk proses ekstraksi deskriptor, langkah pertama yang dilakukan adalah membuat daerah kotak sekitar *keypoint*, dengan *keypoint* sebagai pusat dari daerah kotak tersebut, dan orientasinya di sekitar orientasi yang ditentukan[2].



Gambar 4. Perhitungan Deskriptor[2]

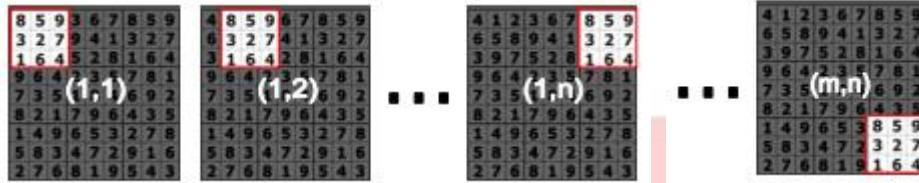
Gambar 4 di atas, *respon wavelet dx* dan *dy* dijumlahkan pada masing-masing *sub-region*. Selain memberikan informasi tentang polar dari perubahan intensitas, juga akan dihasilkan jumlah nilai absolut dari respon $|dx|$ dan $|dy|$. Masing-masing *sub-region* mempunyai empat dimensi deskriptor vektor v, yaitu dx, dy, $|dx|$ dan $|dy|$. Untuk 4x4 *sub-region*, maka panjang vektor deskriptornya adalah 64. Sementara untuk menghasilkan vektor v adalah sebagai berikut:

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \tag{3}$$

Respon *wavelet* juga invariant terhadap pencahayaan, sedangkan sifat invariant terhadap kontras dicapai melalui pembentukan deskriptor ke dalam satuan *vector*.

2.2 Brute Force Matching

Tahap penelusuran untuk mendapatkan tingkat kemiripan akan menggunakan algoritma *brute force*. Matriks akan ditelusuri secara dua dimensi[3].



Gambar 5. Matriks dari Koordinat (1,1) sampai (m,n)[3]

setiap penelusuran menghitung tingkat kemiripan matriks pada koordinat yang ditelusuri. Tingkat kemiripan dihitung dengan rumus:

$$kemiripan(x, y) = 1 - \left(\frac{\sum_{\substack{0 \leq i \leq x \\ 0 < j < y}} kesalahan(i, j)}{panjang \times lebar} \right) \tag{4}$$

Fungsi tersebut memiliki hasil dari 0 sampai dengan 1. Semakin tinggi nilai menunjukkan potongan matriks semakin mirip dengan daerah yang sedang ditelusurinya. Sedangkan fungsi kesalahan adalah fungsi yang menunjukkan sama atau tidaknya elemen potongan matriks.

$$kesalahan(x, y) = \begin{cases} 0, & A[x][y] \neq B[x][y] \\ 1, & A[x][y] = B[x][y] \end{cases} \tag{5}$$

Fungsi kesalahan bernilai 0 ketika elemen kedua matriks yang dibandingkan tidak sama, dan bernilai 1 ketika nilainya sama. Dengan penelusuran *brute force* dan kedua fungsi tersebut, kita dapat memperoleh tingkat kemiripan dari setiap koordinat dan menyimpan ke dalam baris. Tahap selanjutnya adalah mencari tingkat kemiripan tertinggi dari baris tersebut. Koordinat yang memiliki tingkat kemiripan tertinggi adalah koordinat letak potongan gambar tersebut pada gambar utama.[3].

2.3 Gambaran Umum Sistem

Gambar 6 menunjukkan sistem pengenalan sidik jari yang dibuat dengan penjelasan sebagai berikut:

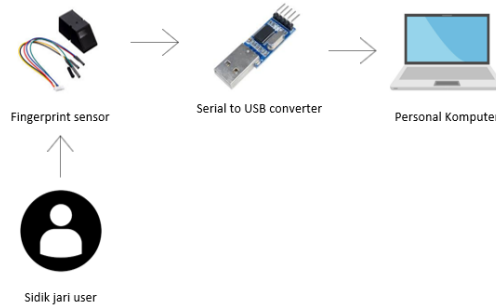


Gambar 6. Gambaran Umum Sistem

1. Pengguna melakukan pengambilan citra sidik jari menggunakan sensor sidik jari.
2. Citra tersebut diproses oleh oleh *personal computer* untuk dilakukan proses *pre-processing*
3. Pengguna mengambil sidik jari kembali pada sensor sidik jari untuk melakukan proses pencocokan dan melihat tingkat kesesuaian citra sidik jari.

2.4 Desain Kebutuhan Perangkat Keras

Sistem yang dibuat membutuhkan beberapa perangkat keras seperti yang ditunjukkan pada Gambar 7.

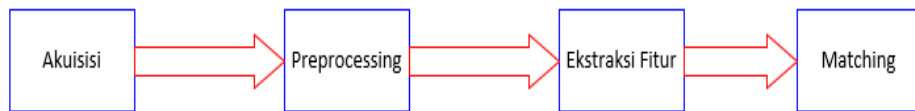


Gambar 7. Kebutuhan Perangkat Keras

Personal computer digunakan sebagai pusat kontrol sistem dalam mengatur jalur *input output* data. Pengguna memasukkan citra sidik jari melalui sensor sidik jari. Data citra yang telah di-*capture* oleh sensor sidik jari akan menjadi data masukan untuk *personal computer* yang kemudian akan diolah dan disimpan ke dalam *storage*. Pada saat program berjalan, pengguna diminta untuk memasukkan data citra uji. Citra uji tersebut diterima oleh *personal computer*, maka program akan membandingkan kecocokan antara citra uji dengan seluruh citra pada *storage* untuk mencari citra yang paling sesuai.

2.5 Desain Sistem

Dalam perancangan dan implementasinya, sistem yang dibuat terdiri dari 4 tahap, yaitu akuisisi gambar, atau pengambilan citra sidik jari, *preprocessing* untuk mendapatkan kualitas citra terbaik, ekstraksi fitur untuk mendapatkan ciri-ciri khusus pada tiap citra sidik jari, dan *matching* untuk mencocokkan citra sidik jari[8]. Keempat tahap tersebut ditunjukkan pada diagram blok seperti pada Gambar 8.



Gambar 8. Diagram Blok Sistem Pengenalan Sidik Jari[8]

Sistem dibuat menggunakan bahasa pemrograman python 3.6. Tahap awal adalah proses akuisisi citra sidik jari menggunakan sensor sidik jari. Setelah citra didapatkan, citra sidik jari tersebut mengalami proses *preprocessing* untuk menghilangkan *noise* yang tidak diperlukan. Lalu dilakukan proses ekstraksi fitur untuk mencari ciri khusus atau informasi penting dari citra sidik jari yang kemudian akan disimpan pada *storage*. Citra tersebut dijadikan sebagai citra *template* untuk dibandingkan dengan citra uji pada proses *matching*.

2.6 Akuisisi Sidik Jari

Tahap ini merupakan proses untuk pengambilan citra sidik jari menggunakan sensor sidik jari. Proses ini menghasilkan gambar sidik jari distorsi dengan format *joint photographic group* (.jpg) yang ditunjukkan pada Gambar 9.



Gambar 9. Hasil Akuisisi Citra Sidik Jari Berminyak

2.7 Preprocessing

Tahap ini merupakan proses yang dilakukan setelah mendapatkan citra sidik jari. Citra sidik jari yang sudah didapatkan mengalami proses *grayscale*. Proses ini bertujuan untuk mengubah agar warna citra sidik

jari hanya pada tingkat keabuan saja. Selain itu proses ini juga bertujuan untuk menyederhanakan model citra dan mempermudah proses perhitungan. Kemudian citra hasil *grayscale* dikonversi ke citra biner, pada proses ini ditentukan nilai ambang atau *threshold*. Nilai ini digunakan untuk mengubah citra *grayscale* ke dalam dua nilai yaitu hitam dan putih.

2.8 Ekstraksi Fitur

Tahap awal ekstraksi fitur adalah dengan mengkonversi citra RGB menjadi *grayscale* yang tujuannya untuk menghitung citra integral yang bertujuan mempercepat proses komputasi. Langkah selanjutnya adalah dengan mengimplementasikan algoritma SURF yang terdiri dari deteksi titik fitur, representasi skala dan deskripsi fitur yang akan dihasilkan *keypoint*. Hasil ekstraksi fitur *speeded-up robust feature* ditunjukkan pada Gambar 10. Citra sidik jari yang sudah mengalami proses ekstraksi dalam bentuk titik-titik *keypoint*. Citra ini yang akan disimpan pada *storage*.



Gambar 10. Citra Sidik Jari Hasil *Speeded-Up Robust Feature*

2.9 Matching

Tahap ini adalah tahap pencocokan antara citra *input* dengan citra *template* yang sudah mengalami tahapan *pre-processing* sebelumnya. Citra *input* akan dibandingkan dengan citra *template*. Nilai selisih antara kedua citra akan menunjukkan seberapa besar tingkat akurasi suatu metode dalam pengenalan sidik jari.

2.10 Tingkat Akurasi

Tingkat akurasi dapat dikalkulasikan dengan cara sebagai berikut :

$$\text{Tingkat Akurasi} = \frac{\sum \text{Data uji citra benar}}{\sum \text{Total data uji}} \times 100\% \quad (6)$$

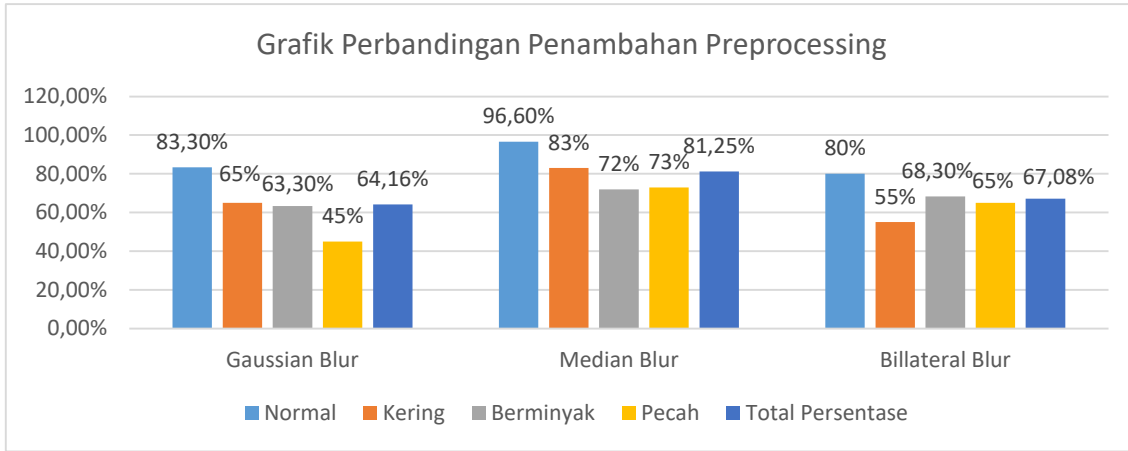
Citra benar apabila citra distorsi yang berhasil dikenali. Sistem dapat dikatakan baik jika memiliki tingkat akurasi yang tinggi.

3. Hasil dan Pembahasan

Data masukan yang digunakan dalam pengujian adalah citra sidik jari distorsi dengan format *joint photographic group* (.jpg) yang ditangkap menggunakan sensor sidik jari dan data yang disimpan adalah sidik jari normal. Target objek pada pengujian berupa sidik jari jempol dan telunjuk distorsi dari bagian kanan maupun kiri. Pengujian dilakukan sebanyak 5 kali percobaan dalam setiap kondisi normal dan distorsi pada masing-masing 3 subjek per kondisi.

3.1 Perbandingan Penambahan *Pre-processing*

Pada awalnya, sistem yang dibuat hanya melakukan proses *grayscale* pada tahap *preprocessing*. Pengujian ini dilakukan melakukan proses penambahan *Gaussian*, *median* dan *bilateral blur*. Hasil perbandingan ditunjukkan pada Gambar 11.

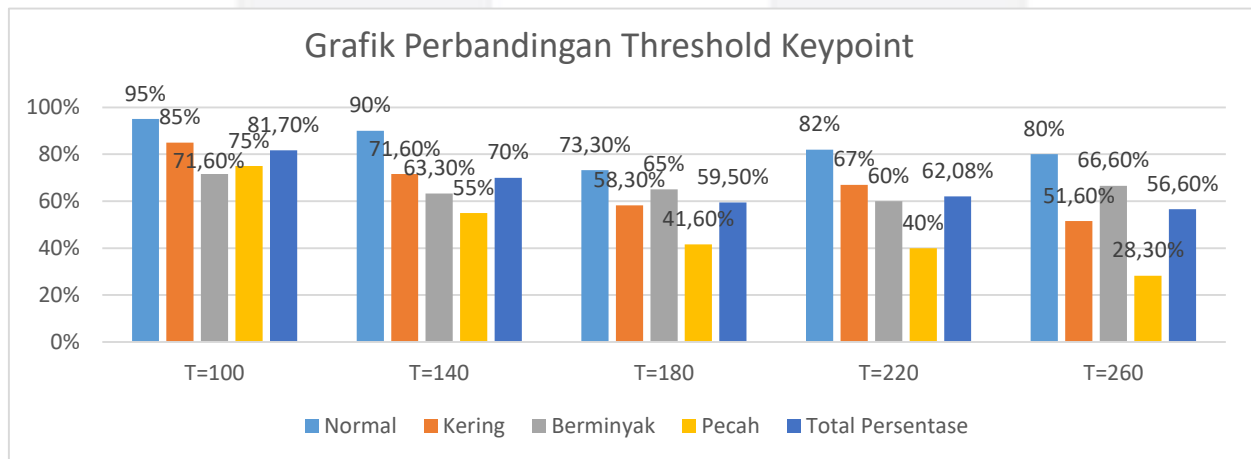


Gambar 11. Hasil Perbandingan Penambahan Pre-processing

Gambar 11 dapat dilihat bahwa sistem pengenalan sidik jari terdistorsi memiliki total persentase yang paling tinggi dengan penambahan parameter *median blur*, yaitu **81.25%**. Berdasarkan hasil pengujian penambahan parameter *preprocessing*, parameter *gaussian blur* memiliki tingkat akurasi rendah dari *median* dan *billateral blur*, dikarenakan *preprocessing* pada *gaussian blur* tidak menjaga tepi ketika menghilangkan *noise* dan tidak menggantikan intensitas setiap piksel dengan rata-rata nilai intensitas dari piksel terdekat. Sehingga garis tepi yang tajam akan dibuang. oleh sebab itu, sulit untuk dideteksi. Sidik jari distorsi yang mudah dikenali, yaitu kering dikarenakan perubahan bentuk pola sidik jari kering tidak jauh berbeda dengan citra *template* yang ada di *storage*, oleh karena itu titik-titik *keypoint* yang terdapat dalam sidik jari kering banyak dan mengalami kesamaan, sehingga mudah untuk dikenali.

3.2 Perbandingan Perubahan Nilai Threshold Keypoint

Pada pengujian ini, dilakukan untuk mengetahui kestabilan jumlah *keypoint* dan kondisi ideal pada sistem dengan cara mengubah nilai *threshold*. Hasil perbandingan ditunjukkan pada Gambar 12.



Gambar 12. Hasil Perbandingan Nilai Threshold Keypoint

Gambar 12 dapat dilihat bahwa kondisi ideal sistem adalah pada saat nilai T = 100 dengan total persentase terbesar, yaitu **81.70%**. Berdasarkan hasil pengujian perubahan nilai *threshold keypoint*, kondisi terdistorsi yang sulit dikenali adalah berminyak dan pecah-pecah dikarenakan ada faktor yang mempengaruhi diantaranya:

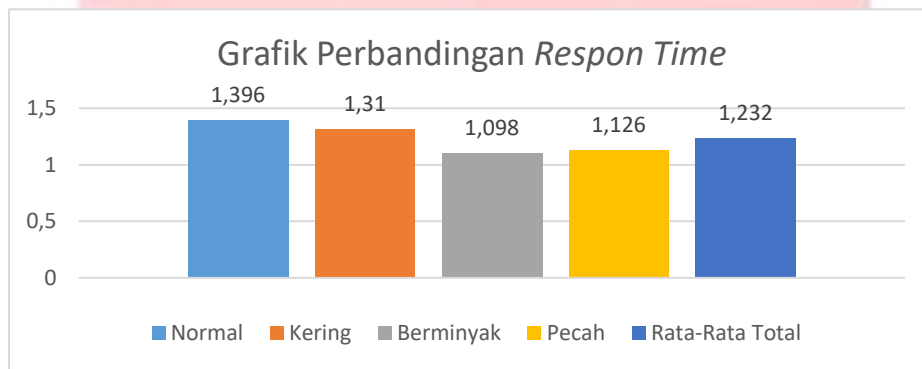
1. Pengambilan gambar yang tidak sempurna sehingga kualitas gambarnya tidak baik.
2. Kadar distorsi minyak dan pecah-pecah terlalu pekat, sehingga *ridge* dan *valley* yang terdapat pada citra sidik jari tidak terlihat dan teratur pembentukannya. oleh karena itu, titik-titik *keypoint* terdapat dalam sidik jari sedikit dan tidak dapat terdeteksi.

3. Perubahan bentuk bisa terjadi karena bentuk citra data awal yang tidak utuh atau tidak sempurna gambar yang ada di dalam citra sampel tersebut. Citra uji yang tidak utuh mungkin karena ada objek lain yang menutupi, sehingga tidak cocok pada citra template yang ada di *storage*.

Dalam pengujian *threshold keypoint* juga memiliki total persentase rendah, yaitu ketika $T=260$ dengan total persentase **56.60%**. Ketika nilai *threshold* dinaikkan, umumnya jumlah *keypoint* yang terdeteksi lebih kecil. Oleh karena itu, menyebabkan penurunan distorsi, sehingga sulit untuk identifikasi sidik jari. Dalam pengujian metode SURF yang dilakukan, jumlah *keypoint* yang dihasilkan lebih besar dari nilai *threshold* dikarenakan metode ini tidak memiliki kestabilan antara jumlah *keypoint* dan nilai *threshold*. Faktor ketelitian sistem dalam mendeteksi dan mencocokkan *keypoint* juga berpengaruh, sedikit perubahan *keypoint* bisa dianggap tidak cocok oleh sistem.

3.3 Perbandingan Respon Time

Pengujian ini dilakukan untuk mengukur seberapa cepat respon *time* pengenalan sidik jari seseorang berdasarkan kondisi kulit terdistorsi. *Respon time* dilakukan pada proses deteksi. Hasil perbandingan ditunjukkan pada Gambar 13.

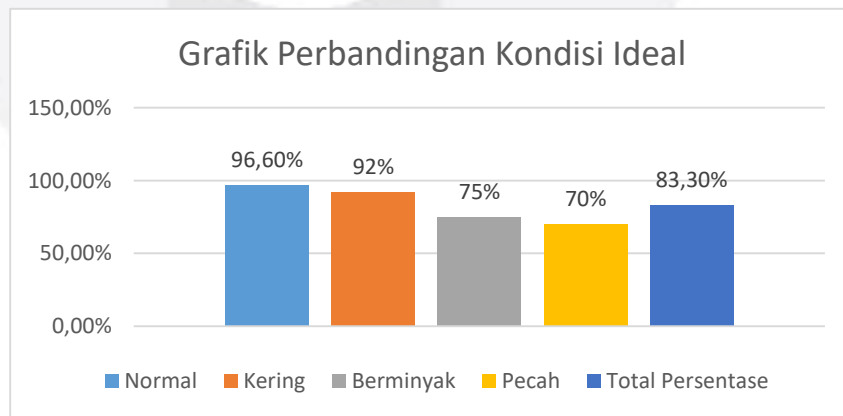


Gambar 13. Hasil Perbandingan Respon Time

Gambar 13 menunjukkan bahwa waktu komputasi pada pengenalan sidik jari memiliki *respon time* yang relatif cepat sebesar **1,232** detik/kondisi.

3.4 Perbandingan Kondisi Ideal Sistem

Pada pengujian ini, kondisi sistem ideal yang akan diuji adalah pada saat sistem memiliki total persentase tertinggi, yaitu nilai $T=100$ dengan menambahkan parameter *median blur*. Hasil perbandingan ditunjukkan pada Gambar 14.



Gambar 14. Hasil Perbandingan Kondisi Ideal

Gambar 14 menunjukkan grafik perbandingan kondisi ideal sistem untuk mengenali citra sidik jari seseorang dengan memiliki total persentase sebesar **83.30%**. Sementara pada pengujian ini, gambar hasil akuisisi sidik jari sudah relatif jelas dan baik sehingga proses *scoring* menyebabkan peningkatan pada proses *bridging* sidik jari yang terputus. Akurasi juga dipengaruhi proses *enrollment*, yaitu pendaftaran jari saat melakukan proses pemindai. Semakin ditekan kuat jarinya, semakin banyak pola sidik jari yang terakuisisi namun semakin banyak pula kesalahannya karena garis-garis sejajar yang seharusnya terpisah pada pola sidik jari menjadi tersambung.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Berdasarkan hasil pengujian, *algoritma speeded-up robust feature extraction* mampu mengimplementasi dan menganalisis sidik jari terdistorsi dengan total persentase sebesar 83.30% dengan menambahkan parameter *threshold* 100 dan parameter *median blur*.
2. Keandalan sistem yang dibuat, cepat mendeteksi sidik jari terdistorsi dengan rata-rata total waktu komputasi sebesar 1.232 detik/kondisi. Hal ini dikarenakan dalam penelitian sebelumnya metode SURF menggunakan citra integral untuk mempercepat waktu komputasi.

4.2 Saran

Adapun saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Kualitas citra sidik jari dapat ditingkatkan melalui 2 cara, yaitu dengan pemilihan sensor sidik jari dengan kualitas baik atau memanfaatkan *image enhancement* pada citra sidik jari.
2. Alat *scan* atau sensor yang digunakan diharapkan membuat penghilangan *noise* citra sidik jari yang dihasilkan.

Daftar Pustaka

- [1] X. Si, S. Member, and J. Feng, "Detection and Rectification of Distorted Fingerprints," vol. 37, no. 3, pp. 555–568, 2015.
- [2] H. Kusumo, "Aplikasi Pengenalan Pola Citra Tanda Tangan dengan Metode SURF pada Single Board Computer Beaglebone Black," 2015.
- [3] A. P. Wicaksana, "Algoritma Brute Force dalam Pattern Matching pada Aplikasi Pendeteksian Potongan Citra," 2013.
- [4] D. G. Low, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, pp. 91–110, 2004, [Online]. Available: <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>.
- [5] R. Lionnie, T. M. Kadarina, and M. Alaydrus, "Analisis Metode SIFT dan SURF untuk Sistem Pendeteksi Gambar Termanipulasi Penyerangan Copy-Move Forgery," *J. Telekomun. dan Komput.*, vol. 8, no. 3, p. 183, 2018, doi: 10.22441/incomtech.v8i3.3074.
- [6] M. Brown and D. Lowe, "Invariant Features from Interest Point Groups," pp. 23.1-23.10, 2013, doi: 10.5244/c.16.23.
- [7] W. E. Holzinger, H. Löcker, and B. Löcker, "Fulgoromorpha of Seychelles: A preliminary checklist," *Bull. Insectology*, vol. 61, no. 1, pp. 121–122, 2008.
- [8] A. L. PRASASTI, B. IRAWAN, S. E. FAJRI, A. RENDIKA, and S. HADIYOSO, "Perbandingan Ekstraksi Fitur dan Proses Matching pada Autentikasi Sidik Jari Manusia," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 8, no. 1, p. 95, 2020, doi: 10.26760/elkomika.v8i1.95.
- [9] N. A. Nugraha, B. Irawan, and A. L. Prasasti, "Singapore Dollar Recognition Using ORB Feature Based on Android," *Proc. - 2018 Int. Conf. Control. Electron. Renew. Energy Commun. ICCEREC 2018*, no. August 2019, pp. 142–148, 2018, doi: 10.1109/ICCEREC.2018.8711993.
- [10] M. Rizky Adhiguna, B. Irawan, and A. Luhur Prasasti, "Design of Foreign Currency Recognition Application using Scale Invariant Feature Transform (SIFT) Method based on Android (Case Study: Singapore Dollar)," *J. Eng. Appl. Sci.*, vol. 14, no. 19, pp. 6991–6997, 2019, doi: 10.36478/jeasci.2019.6991.6997.