

Indentifikasi Persamaan Sidik Jari Pada Mahaiswa Menggunakan Kombinasi Algoritma Sift Dan Algoritma Flann Pada Bahasa Pemograman Python

Identification of Fingerprint Similarities in Students Using a Combination of the Sift Algorithm and the Flann Algorithm On the Python Programming Language

Abdul Razaq^{a,1}, Mas Ayoe Elhias Nasution^{b,2}

^a Fakultas Teknik dan Ilmu Komputer, Universitas Potensi Utama

^b Fakultas Teknik dan Ilmu Komputer, Universitas Potensi Utama

razzaqkis79@gmail.com¹, masayoe02@gmail.com²

ABSTRAK

Seiring perkembangan teknologi informasi, kebutuhan akan sistem keamanan yang andal semakin meningkat. Salah satu metode identifikasi yang banyak digunakan yaitu pengenalan sidik jari karena sifatnya yang unik dan permanen. Penelitian ini mengembangkan sistem identifikasi sidik jari mahasiswa menggunakan kombinasi algoritma SIFT (Scale-Invariant Feature Transform) dan FLANN (Fast Library for Approximate Nearest Neighbors) yang diimplementasikan dalam bahasa pemrograman Python. Kombinasi kedua algoritma ini bertujuan untuk mengatasi permasalahan orientasi dan skala pada citra sidik jari serta meningkatkan efisiensi dan akurasi pencocokan. Sistem ini dirancang untuk berbagai keperluan akademik seperti absensi elektronik, akses ruang, sistem akademik, dan perpustakaan. SIFT memberikan ketahanan terhadap rotasi dan skala, sementara FLANN mempercepat proses pencarian fitur terdekat dalam dataset besar. Hasil integrasi keduanya diharapkan mampu menghasilkan sistem identifikasi sidik jari yang cepat, akurat, dan efisien untuk mendukung keamanan serta kenyamanan layanan digital di lingkungan kampus.

Kata Kunci : Sidik jari, Identifikasi, SIFT, FLANN, Pencocokan Citra, Python, Keamanan Sistem, Pengenalan Mahasiswa.

ABSTRACT

With the advancement of information technology, the need for reliable security systems is increasing. One widely used method of identification is fingerprint recognition due to its unique and permanent characteristics. This study develops a student fingerprint identification system using a combination of the Scale-Invariant Feature Transform (SIFT) dan Fast Library for Approximate Nearest Neighbors (FLANN) algorithms, implemented in the Python programming language. The integration of these algorithms aims to address issues of orientation and scale variations in fingerprint images while enhancing matching accuracy and processing efficiency. The system is designed for various academic applications such as electronic attendance, room access, academic systems, and library services. SIFT offers robustness against rotation and scale changes, while FLANN accelerates the nearest neighbor search in large datasets. The integration of both algorithms is expected to deliver a fast, accurate, and efficient fingerprint identification system to support secure and convenient digital services within the campus environment.

Keywords : Fingerprint, Identification, SIFT, FLANN, Image Matching, Python, System Security, Student Recognition

Disubmit: 10 Januari 2025

Info Artikel :
Direview: 12 Februari 2025

Diterima : 05 Maret 2025

Copyright © 2025 - Journal UPU. All rights reserved.

1. PENDAHULUAN

Sidik jari yaitu bagian tubuh manusia yang unik. Sidik jari yaitu pola ulur yang terdapat diujung jari manusia, pola ini terbentuk sejak manusia masih dalam kandungan dan tidak akna berubah sepanjang

hidupnya. Sidik jari pada setiap manusia berbeda-beda, bahkan pada kembar identic sekalipun. Oleh karena itu sidik jari sering digunakan sebagai identitas seseorang mengenai keaslian dirinya. Keunikan yang ada pada sidik jari biasanya tetap sama sepanjang hidup, pola sidik jari tidak akan berubah sejak manusia lahir hingga meninggal dunia, hal ini menjadikan sidik jari sebagai alat identifikasi yang sangat handal, dan sidik jari dapat digunakan untuk berbagai keperluan seperti identifikasi forensik, keamanan, otentikasi.

Pada permasalahan yang diangkat bagaimana melakukan persamaan mengenai mengidentifikasi sidik jari pada mahasiswa. Adapun tujuan pengenalan sidik jari mahasiswa bisa digunakan oleh kampus yang berguna untuk mengidentifikasi absensi elektronik, akses ruangan mahasiswa, akses sistem akademik dan sistem perpustakaan. Untuk memecahkan permasalahan ini digunakan sebuah sistem pengenalan sidik jari untuk mengidentifikasi identitas mahasiswa dengan menggunakan algoritma SIFT dan penggabungan dengan algoritma FLANN agar dapat mengidentifikasi sidik jari seorang mahasiswa. Perancangan aplikasi sistem sidik jari ini menggunakan software python untuk penggabungan sidik jari dengan penggabungan algoritma SIFT dan Algoritma FLANN.

Metode SIFT tahan terhadap perubahan pada benda atau objek, tetapi ada beberapa kelemahan yang akan terjadi pada sistem jika tidak menggunakan algoritma SIFT, salah satunya yaitu akurasi pencocokan gambar data menurun secara signifikan. Hal ini dapat berakibat pada kesalahan dalam pencocokan gambar. Kesulitan dalam menemukan gambar yang sama, dan penurunan kinerja dalam aplikasi yang menggunakan pencocokan gambar.

Algoritma FLANN (*Fast Library for Approximate Nearest Neighbors*) yaitu algoritma yang digunakan brute-force nearest neighbors search. FLANN banyak digunakan dalam berbagai aplikasi, seperti pengenalan wajah, klasifikasi gambar, rekomendasi sistem. Banyak hal yang didapat dengan memanfaatkan algoritma FLANN, tetapi ada beberapa kerugian yang akan terjadi jika kita menjalankan sistem tanpa algoritma ini, seperti proses pencarian tetangga terdekat dapat menjadi sangat lambat, terutama untuk dataset yang besar.

Sidik Jari yaitu metode verifikasi identitas manusia yang mengandalkan fitur fisik spesifik tubuh, seperti pola sidik jari tangan. Untuk penjelasan lebih lanjut tentang sidik jari, lihat Handbook of Biometric Anti-Spoofing. Pola sidik jari yaitu patokan biometrik yang populer dan akurat karena masing-masing individu memiliki karakteristik yang berbeda dan permanen[1]. Python yaitu bahasa pemrograman yang kuat dan serbaguna yang dapat digunakan untuk banyak aplikasi dan memiliki banyak komunitas yang aktif dan Python yaitu bahasa skrip tingkat tinggi yang mirip dengan bahasa Inggris yang dapat digunakan untuk berbagai tujuan, seperti otomatisasi tugas, manipulasi data, pemrosesan gambar, kecerdasan buatan, dan pembelajaran mesin[2]. Kombinasi yaitu proses yang penting untuk menggabungkan dua atau lebih elemen untuk membentuk sesuatu yang baru. Kombinasi yaitu bidang yang terus berkembang dengan munculnya metode dan algoritma baru.[3] FLANN sebagai alat yang serbaguna yang dapat digunakan untuk berbagai aplikasi, termasuk pengenalan gambar, pengenalan wajah, dan pencocokan string.[4] SFIT sebagai alat yang serbaguna yang dapat digunakan untuk berbagai aplikasi, termasuk ekstraksi informasi, pengenalan entitas, dan klasifikasi teks.[5]

2. METODE

Identifikasi persamaan sidik jari merupakan sebuah proses untuk menentukan apakah dua sidik jari berasal dari orang yang sama atau tidak. Proses ini memiliki beberapa tantangan seperti keunikan dalam sidik jari dimana setiap sidik jari manusia unik bentuknya, bahkan manusia kembar identic sekali pun memiliki sidik jari yang berbeda, keunikan ini terletak pada pola alur dan titik-titik pada sidik jari. Dan juga dalam beberapa aplikasi, seperti forensik dan kontrol akses, diperlukan identifikasi sidik jari

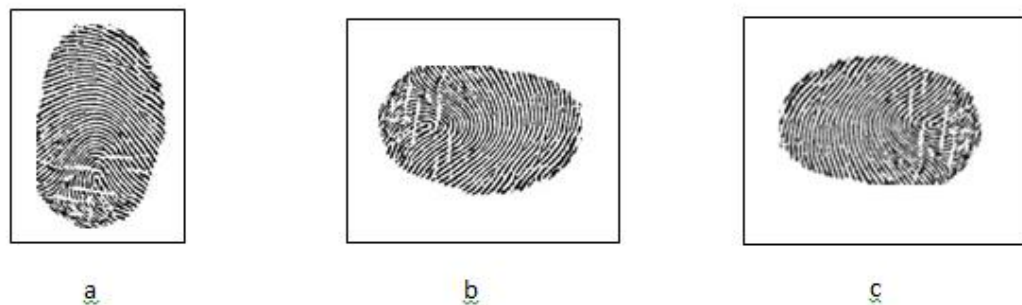
yang cepat dan akurat, hal ini menjadi tantangan karena proses identifikasi sidik jari dapat memakan waktu.

Untuk mendukung penelitian ini penulis menggunakan metode kombinasi SIFT dan FLANN untuk mendukung keakuratan nilai persamaan sidik jari, dimana penelitian ini menunjukkan bahwa kombinasi SIFT dan FLANN dapat digunakan untuk mengidentifikasi persamaan sidik jari pada mahasiswa atau pada manusia yang ingin kita lakukan uji coba persamaan sidik jari dengan tingkat akurasi yang tinggi. Masalah pada penelitian ini yaitu ingin dipecahkan oleh penelitian ini yaitu bagaimana mengidentifikasi persamaan sidik jari, kesulitan dalam identifikasi sidik jari dapat disebabkan oleh berbagai factor seperti kualitas sidik jari, variasi sidik jari antar individu. Ketidakmampuan untuk mengidentifikasi sidik jari dapat menyebabkan masalah keamanan, seperti penipuan identitas dan akses yang tidak sah.

Beberapa tindakan harus diambil untuk memecahkan masalah tersebut, seperti Normalisasi ukuran: Mengubah ukuran sidik jari menjadi lebih elastis dan mengatasi perbedaan luasan kulit dengan normalisasi Pengolahan gambar: Memperbaiki kualitas gambar sidik jari dengan menggunakan filter seperti Gabor untuk menghasilkan gambar yang lebih baik Pengenalan fitur minutiae: Menggali fitur minutiae yang ada pada gambar sidik jari, seperti ujung, inti, dan bifurcation, yang dapat digunakan untuk pencocokan metode identifikasi.

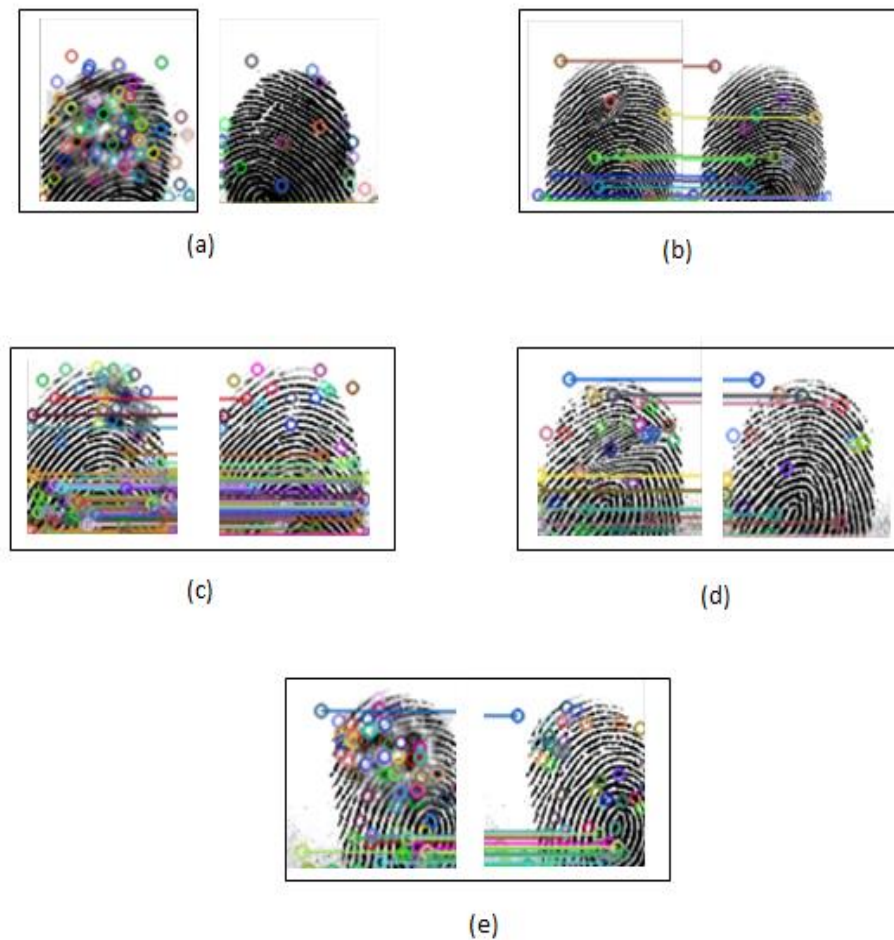
Penelitian ini memanfaatkan data hasil ujian berbasis kertas yang kemudian dibandingkan dengan referensi dari buku dan jurnal, serta dilengkapi dengan data sidik jari. Data utama yang digunakan meliputi sidik jari mahasiswa yang dianalisis untuk keperluan deteksi. Alat yang digunakan dalam penelitian mencakup perangkat keras dan perangkat lunak. Untuk perangkat keras, digunakan sebuah laptop dengan spesifikasi prosesor Intel Core i3 berkecepatan 1,8 GHz, RAM sebesar 4 GB, dan penyimpanan sebesar 500 GB. Adapun perangkat lunak yang digunakan meliputi sistem operasi Microsoft Windows 10 serta bahasa pemrograman Python yang dilengkapi dengan script khusus.

Kumpulan data yaitu sekumpulan data yang terstruktur. Data dalam kumpulan data dapat saling terkait dengan cara tertentu. Objek yang digunakan untuk melakukan pengujian seluruhnya berjumlah 5 buah sidik jari dimana masing-masing terdiri dari dua muka. Masing-masing objek tersebut akan dikenai berbagai pengujian rotasi, yaitu tanpa rotasi atau 0o , rotasi 90o , rotasi 180o.. Contoh citra pecahan nominal uang kertas rupiah dalam beberapa variasi rotasi seperti gambar dibawah ini.



Gambar 1 Citra sidik jari: a) rotasi 0 o , b) rotasi 90o , c) rotasi 180o

Pada proses ekstraksi ciri citra dengan metode SIFT, diperoleh keypoint yang merupakan vektor ciri yang mendeskripsikan citra itu sendiri. Keypoint dari hasil ekstraksi ciri citra sidik jari.



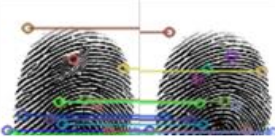
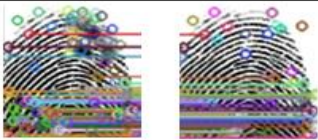
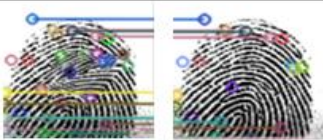
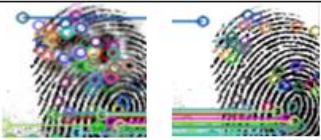
Gambar 2 Keypoint hasil ekstraksi ciri citra sidik jari dengan metode SIFT

Pengujian dilakukan pada ketujuh pecahan nominal uang kertas pada setiap muka pada variasi rotasi dengan cara mengubah orientasi default dari akuisisi citra sidik jari sebagai berikut ini.

Tabel 1. Hasil pengujian variasi rotasi dari 5 sidik jari

Gambar	Pengujian Rotasi	Citra Uji	Berhasil dikenali	Dikenali Objek	Akurasi
a	0°	4	4	0	100%
b	90°	4	4	0	100%
c	180°	4	4	0	100%
d	270°	4	4	0	100%
e	360°	4	4	0	100%

Untuk membuat dataset menjadi data training dengan metode SIFT (Scale-Invariant Feature Transform) dan FLANN (Fast Library for Approximate Nearest Neighbors) dalam Python. Berikut yaitu langkah-langkah umum yang bisa Anda ikuti:

Dataset	Keypoint Sift dan Flann	Akurasi%
		100%
		100%
		100%
		100%
		100%

Pengujian dilakukan pada ketujuh pecahan nominal uang kertas pada setiap muka pada variasi rotasi dengan cara mengubah orientasi default dari akuisisi citra sidik jari sebagai berikut ini.

Tabel 2. Hasil pengujian variasi rotasi dari 5 sidik jari

Gambar	Pengujian Rotasi	Citra Uji	Berhasil dikenali	Dikenali Objek	Akurasi
a	0°	4	4	0	100%
b	90°	4	4	0	100%
c	180°	4	4	0	100%
d	270°	4	4	0	100%
e	360°	4	4	0	100%

Dataset ini mencakup beberapa kolom yang menunjukkan informasi dasar tentang citra sidik jari dan hasil ekstraksi fitur serta pencocokan:

Tabel 3 Dataset Sidik Jari

Id Citra	Nama Citra	Jumlah Fitur Sift	Waktu Sift	Jumlah Fitur Flann	Waktu (ms)
1	Fingerprint1	145	14	97	14
2	Fingerprint2	160	18	96	18
3	Fingerprint3	110	12	92	12
4	Fingerprint4	135	13	95	13
5	Fingerprint5	175	20	98	20
6	Fingerprint6	150	15	96	15
7	Fingerprint7	120	11	93	11
8	Fingerprint8	140	14	94	14
9	Fingerprint9	170	19	97	19
10	Fingerprint10	115	12	91	12

- 1) ID_Citra: ID unik untuk setiap citra sidik jari.
- 2) Nama_Citra: Nama atau identifikasi citra sidik jari.
- 3) Jumlah_Fitur_SIFT: Jumlah fitur yang diekstraksi menggunakan metode SIFT dari citra sidik jari.
- 4) Waktu_Ekstraksi_SIFT (ms): Waktu yang dibutuhkan untuk mengekstrak fitur SIFT dalam milidetik.
- 5) Jumlah_Pasangan_FLANN: Jumlah pasangan fitur yang berhasil dicocokkan menggunakan metode FLANN.
- 6) Waktu_Pencocokan_FLANN (ms): Waktu yang dibutuhkan untuk mencocokkan fitur menggunakan metode FLANN dalam milidetik.

Dengan berdasarkan rumus dari metode SIFT untuk mencari persamaan pola pada sidik jari, disini penulis lebih mendekati penggunaan metode SIFT untuk Pencocokan Fitur (Feature Matching) dalam mencari kesamaan pada dua gambar atau dikenal juga sebagai Euclidean Distance: Euclidean distance digunakan untuk mengukur kesamaan antara dua SIFT descriptor, yaitu :

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}$$

di mana:

x dan y yaitu dua SIFT descriptor

n yaitu dimensi dari SIFT descriptor

kemudian berdasarkan rumus tersebut penulis mengimplementasikan algoritma metode SIFT pada python seperti berikut :

```
keypoints_1, descriptors_1 = sift.detectAndCompute(sample, None)
keypoints_2, descriptors_2 = sift.detectAndCompute(fingerprint_image, None)
matches = cv2.FlannBasedMatcher({'algorithm':1, 'trees':10},
                                {}).knnMatch(descriptors_1, descriptors_2, k=2)
match_points = []
for p, q in matches:
    if p.distance < 0.1 * q.distance:
        match_points.append(p)
```

```

keypoints = 0
if len(keypoints_1) < len(keypoints_2):
    keypoints = len(keypoints_1)
else:
    keypoints = len(keypoints_2)
if len(match_points) / keypoints*100 > best_score:
    best_score = len(match_points)/keypoints*100
    filename = file
    image = fingerprint_image
    kp1, kp2, mp = keypoints_1, keypoints_2, match_points

```

Metode SIFT (Scale Invariant Feature Transform) diterapkan pada citra digital guna mendeteksi titik-titik kunci yang akan dimanfaatkan dalam proses identifikasi pola pada gambar. Algoritma SIFT terdiri dari beberapa tahapan utama yang saling terintegrasi.

- a. Pembentukan *Gaussian Scale Space* dan *Difference of Gaussian Scale Space*. Mencari nilai *gaussian* menggunakan rumus:

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

Setelah diperoleh nilai dari Gaussian Scale Space, tahap berikutnya adalah membentuk Difference of Gaussian (DoG) Scale. Proses pembentukan DoG dilakukan dengan menggunakan rumus sebagai berikut:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

Setelah titik-titik ekstremum berhasil diidentifikasi, langkah selanjutnya adalah meningkatkan ketepatan lokasinya hingga mencapai tingkat sub-piksel. Hal ini dilakukan melalui pendekatan ekspansi Taylor orde dua terhadap fungsi ruang skala, sehingga posisi ekstremum yang lebih akurat dapat diperoleh menggunakan rumus sebagai berikut:

$$Z = -\left(\frac{\partial^2 D}{\partial x^2}\right)^{-1} \frac{\partial D}{\partial x}$$

Proses penentuan *keypoint* dilakukan dengan mencari nilai *maxima* / *minima* DoG yang dideteksi dengan membandingkan sebuah piksel dengan tetangganya.

$$D(Z) = D + \frac{1}{2} \frac{\partial^2 D}{\partial x^2} Z^2$$

Jika nilai (z) tidak melebihi nilai *threshold* (0,03) maka *keypoint* akan dihilangkan dan tidak terpakai lagi. Penghapusan *keypoint* yang tidak memenuhi syarat *threshold* dilakukan juga penghapusan *keypoint* tak stabil yang berada di daerah *edge*.

Penentuan ini dilakukan dengan menggunakan citra Gaussian smooth L yang memiliki skala paling mendekati skala *keypoint* yang teridentifikasi. Untuk setiap sampel citra (x,) magnitu de(x,y) dan orientasi $\theta(x,y)$ dihitung dengan rumus berikut:

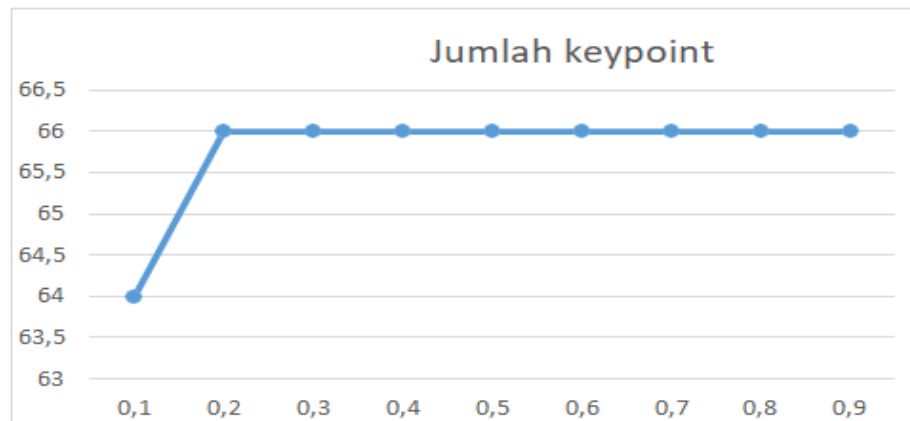
$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (3.5)$$

Setelah keseluruhan langkah tersebut akan memperoleh hasil dari perhitungan berupa titik-titik yang nantinya akan menjadi *keypoint* atau titik deteksi yang akan dikenali sistem.

Tabel 4 Keypoint

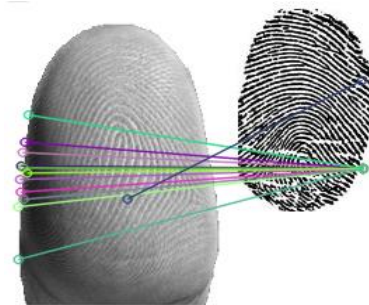
No	Nilai Threshold	Jumlah keypoint
1	0,1	64
2	0,2	66
3	0,3	66

4	0,4	66
5	0,5	66
6	0,6	66
7	0,7	66
8	0,8	66
9	0,9	66



Gambar 3 Jumlah Keypoint

Langkah pertama dilakukan dengan mengekstraksi fitur dan deskriptor menggunakan metode ORB, kemudian mencocokkan fitur tersebut dengan pendekatan Brute Force. Selanjutnya, fitur dan deskriptor diekstraksi menggunakan metode SIFT, yang kemudian diuji melalui proses pencocokan menggunakan metode Brute-Force dan FLANN. Adapun hasil dari proses deteksi fitur dengan menggunakan SIFT menunjukkan bahwa:



Gambar 4 Membandingkan Keypoint Dari Kedua Citra

Perhitungan matematika dalam metode FLANN (Fast Library for Approximate Nearest Neighbors) melibatkan beberapa konsep utama dalam pencarian tetangga terdekat pada ruang dimensi tinggi.

Perhitungan Matematika dalam Implementasi

KD-Tree Construction:

- 1) Memilih Dimensi untuk Partisi:
Dimensi dipilih bergantian($x_1, x_2, \dots, x_k$)

2) Median Partisi:

$$\text{Median} = \frac{\text{nilai ke}-(n/2)}{\text{dalam dimensi yang dipilih}}$$

Nearest Neighbor Search:

3) Distance Calculation:

$$\text{Jarak Euclidean} = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

4) Tree Traversal:

Ikuti cabang pohon yang lebih dekat ke titik query.

5) Approximate Checks:

Maksimal checks yang dilakukan=Nilai tertentu, misalnya 50

Ratio Test:

6) Ratio Comparison:

$$\frac{d_1}{d_2} < 0.7$$

3. HASIL DAN PEMBAHASAN

Bab ini membahas penyajian data hasil penelitian tentang analisis sidik jari mahasiswa menggunakan metode SIFT (Scale-Invariant Feature Transform) dan FLANN (Fast Library for Approximate Nearest Neighbors). Penelitian bertujuan untuk mengidentifikasi pola sidik jari dan mengevaluasi efektivitas kedua metode dalam proses ekstraksi dan pencocokan ciri. Data yang diperoleh disajikan dalam bentuk tabel, grafik, dan pembahasan analitis.

Dataset terdiri atas 10 sidik jari dari mahasiswa dengan berbagai pola (Loop, Whorl, Arch). Data diolah menggunakan algoritma SIFT untuk mengekstraksi fitur utama dan FLANN untuk pencocokan fitur.

Tabel 5 Dataset Yang Digunakan

No	Pola Sidik Jari	Jumlah SIFT	Waktu (ms)	FLANN (%)
1	Loop	145	14	97
2	Whorl	160	18	96
3	Arch	110	12	92
4	Loop	135	13	95

Pada tahap ini, algoritma SIFT digunakan untuk mendeteksi keypoints (titik kunci) dan mengekstraksi deskriptor unik dari masing-masing sidik jari. Berikut hasil ekstraksi fitur:

Tabel 6 Dataset Pola

No	Pola Sidik Jari	Jumlah Keypoint	Waktu Ekstraksi (ms)
1	Loop	145	14
2	Whorl	160	18

3	Arch	110	12
4	Loop	135	13
5	Whorl	175	20
6	Loop	150	15
7	Arch	120	11
8	Loop	140	14
9	Whorl	170	19
10	Arch	115	12

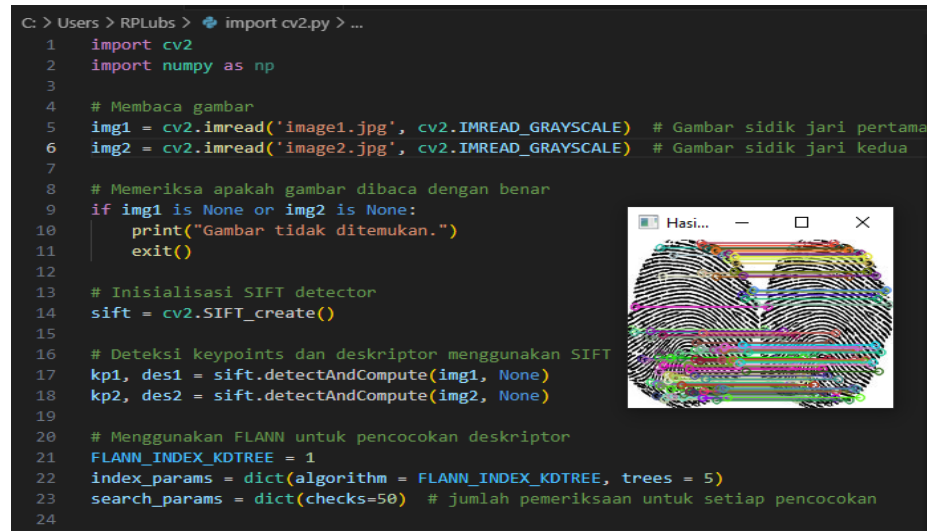
Analisis lanjutan terhadap hasil deteksi sidik jari menggunakan algoritma SIFT (Scale-Invariant Feature Transform) dan FLANN (Fast Library for Approximate Nearest Neighbors). Analisa mencakup evaluasi kinerja, pola pencocokan fitur, dan efektivitas metode berdasarkan 10 gambar sidik jari yang mewakili berbagai pola, yaitu Loop, Whorl, dan Arch.

Ekstraksi fitur yaitu langkah awal dalam analisis citra sidik jari. Algoritma SIFT digunakan untuk mendeteksi keypoints yang relevan dan menghasilkan deskriptor unik untuk setiap gambar. Rata-rata jumlah keypoints per pola:

Tabel 7 Dataset Pola Keypoint

No	Pola Sidik Jari	Rata-rata Keypoint
1	Loop	141
2	Whorl	161
3	Arch	101

Bagian hasil pengujian dalam penelitian ini memuat penjelasan terperinci mengenai temuan yang diperoleh selama proses penelitian. Penyampaian hasil perlu dilakukan secara jelas, baik dalam bentuk penemuan teoritis maupun aplikatif, yang dapat didukung dengan gambar, diagram, atau bentuk visual lainnya. Pada tahap pengujian program, terdapat beberapa metode yang digunakan, di antaranya adalah pengujian terhadap kesalahan sintaksis dan kesalahan logika.



Gambar 5 Pengujian Hasil Sidik Jari

- 1) IFT Detector: Detektor SIFT digunakan untuk mendeteksi fitur-fitur penting (keypoints) pada gambar sidik jari.
- 2) FLANN Matcher: Pustaka FLANN digunakan untuk mencocokkan fitur yang terdeteksi dari dua gambar.
- 3) Lowe's Ratio Test: Ini digunakan untuk menghilangkan pencocokan yang buruk dengan membandingkan jarak dua tetangga terdekat (KNN).

Listing program digunakan untuk menerjemahkan bahasa manusia sehingga dapat dimengerti oleh komputer, sehingga sistem yang akan dibangun sesuai dengan keinginan pembuat sistem. Berikut ini merupakan listing program dari masing-masing menu halaman yang dibuat.

```
import cv2
import numpy as np

# Membaca gambar
img1 = cv2.imread('image1.jpg', cv2.IMREAD_GRAYSCALE) # Gambar sidik jari pertama
img2 = cv2.imread('image2.jpg', cv2.IMREAD_GRAYSCALE) # Gambar sidik jari kedua

# Memeriksa apakah gambar dibaca dengan benar
if img1 is None or img2 is None:
    print("Gambar tidak ditemukan.")
    exit()

# Inisialisasi SIFT detector
sift = cv2.SIFT_create()

# Deteksi keypoints dan deskriptor menggunakan SIFT
kp1, des1 = sift.detectAndCompute(img1, None)
kp2, des2 = sift.detectAndCompute(img2, None)

# Menggunakan FLANN untuk pencocokan deskriptor
FLANN_INDEX_KDTREE = 1
index_params = dict(algorithm = FLANN_INDEX_KDTREE, trees = 5)
search_params = dict(checks=50) # jumlah pemeriksaan untuk setiap pencocokan
```

```

search_params = dict(checks=50) # jumlah pemeriksaan untuk setiap pencocokan

flann = cv2.FlannBasedMatcher(index_params, search_params)

# Melakukan pencocokan menggunakan K-Nearest Neighbors (k=2)
matches = flann.knnMatch(des1, des2, k=2)

# Menyimpan pencocokan yang baik menggunakan metode Lowe's ratio test
good_matches = []
for m, n in matches:
    if m.distance < 0.7 * n.distance:
        good_matches.append(m)

# Menggambar hasil pencocokan
result_img = cv2.drawMatches(img1, kp1, img2, kp2, good_matches, None,
flags=cv2.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)

# Menampilkan hasil pencocokan
cv2.imshow('Hasil Pencocokan Sidik Jari', result_img)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

SIFT Detector: Detektor SIFT digunakan untuk mendeteksi fitur-fitur penting (keypoints) pada gambar sidik jari.

FLANN Matcher: Pustaka FLANN digunakan untuk mencocokkan fitur yang terdeteksi dari dua gambar.

Lowe's Ratio Test: Ini digunakan untuk menghilangkan pencocokan yang buruk dengan membandingkan jarak dua tetangga terdekat (KNN).

4. KESIMPULAN

Berikut ini kesimpulan dari penelitian ini yaitu sebagai berikut:

- 1) Kombinasi antara algoritma Scale-Invariant Feature Transform (SIFT) untuk ekstraksi ciri (feature extraction) dan algoritma Fast Library for Approximate Nearest Neighbors (FLANN) untuk pencocokan ciri (feature matching) terbukti memberikan hasil identifikasi sidik jari yang cepat dan akurat. Algoritma SIFT unggul dalam mendeteksi dan mengekstraksi fitur sidik jari, sementara FLANN mempercepat proses pencocokan.
- 2) mplementasi kedua algoritma ini menghasilkan tingkat akurasi yang tinggi dalam mengidentifikasi persamaan sidik jari antar mahasiswa. Tingkat akurasi ini didukung oleh kemampuan SIFT dalam mendeteksi ciri unik pada pola sidik jari, serta efisiensi FLANN dalam membandingkan banyak data secara simultan.
- 3) Kinerja sistem sangat bergantung pada parameter yang digunakan dalam algoritma SIFT dan FLANN, seperti jumlah keypoints yang diekstraksi dan ambang batas pencocokan (matching threshold). Penyesuaian parameter yang optimal mampu meningkatkan kecepatan dan akurasi sistem.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

REFERENSI

- [1] Anisah R, Jimmy. 2017. "Implementasi Customer Relationship Management (CRM) Pada Sistem Reservasi Hotel Berbasis Website Dan Desktop." *Bandung, Universitas Kristen Maranatha* 6(2):113–26.
- [2] Aji, S., & Prاتمanto, D. (2021). Sistem Informasi Inventory Barang Menggunakan Metode Waterfall. *Indonesian Journal on Software Engineering (IJSE)*, 7(1), 93–99. <https://doi.org/10.31294/ijse.v7i1.10601>
- [3] Arista, A., & Firmansyah, B. (2022). Perancangan Sistem Informasi Pemesanan Menu Makanan dan Minuman Berbasis WEB. *Junif: Jurnal Nasional Informatika*, 3(1), 36–41.
- [4] Fanggidae, A., Sihotang, D. M., & Rihi Pati, A. P. (2020). Pengenalan Pola Sidik Jari Dengan Metode Local Binary Pattern Dan Learning Vector Quantization. *Jurnal Komputer Dan Informatika*, 7(2), 148–156. <https://doi.org/10.35508/jicon.v7i2.1635>
- [5] Badrul, M. (2021). Penerapan Metode waterfall untuk Perancangan Sistem Informasi Inventory Pada Toko Keramik Bintang Terang. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 8(2), 57–52. <https://doi.org/10.30656/prosisko.v8i2.3852>.
- [6] Casuarina, I. P., Hayati, M. N., & Prangga, D. S. (2022). Klasifikasi Status Pembayaran Kredit Barang Elektronik dan Furniture Menggunakan Support Vector Machine Classification of Credit Payment Status for Electronic and Furniture Using Support Vector Machine. *Jurnal Eksponensial*, 13(1), 71–78.
- [7] Maudi, M., A. Nugraha, and B. Sasmito. 2015. "Desain Aplikasi Sistem Informasi Pelanggan Pdam Berbasis Webgis (Studi Kasus : Kota Demak)." *Jurnal Geodesi Undip* 3(3):98–110.
- [8] Rahman, A., Putra, Y., & Wibowo, T. 2022. "Penerapan IoT Untuk Prediksi Kebutuhan Spare Part Pada Mesin Produksi." *Jurnal Teknologi Dan Sistem Informasi* (10(2)):112–20.
- [9] Muhammad Romzi, & Kurniawan, B. (2020). Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma. *JTIM: Jurnal Teknik Informatika Mahakarya*, 03(2), 37–44.
- [10] Noviana, Rina. 2022. "Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan Php Dan Mysql." *JTS* 1(2):112–24.
- [11] Nugraha, A. R., & Hasan, A. (2019). Kendali Perangkat Elektronik Menggunakan Aplikasi Berbasis Web Menggunakan Arduino. *Jumantaka*, 03(1), 1. <http://jurnal.stmik-dci.ac.id/index.php/jumantaka/article/view/364>
- [12] Salamun, S., & Sukri, S. (2021). Analisa Pemanfaatan dan Peran Software Open-source bagi Mahasiswa Universitas Abdurrab. *Jurnal Buana Informatika*, 12(1), 49–57. <https://doi.org/10.24002/jbi.v12i1.4145>
- [13] Romadhoni, A. (2023). Pengaruh Fotografi Jurnalistik Pada Media Online. *IMAJI: Film, Fotografi, Televisi, & Media Baru*, 14(2), 98–107. <https://doi.org/10.52290/i.v14i2.115>
- [14] Rajagukguk, A. L. (2022). Kombinasi Metode Unsharp Masking dan Filter High Boost Dalam Meningkatkan Kualitas Video Call. *Journal of Informatics Management and ...*, 2(4), 151–158. <http://www.hostjournals.com/jimat/article/view/180>
- [15] Warsito, A. B., Ananda, A., & Triyanjaya, D. (2019). Penerapan Data JSON Untuk Mendukung Pengembangan Aplikasi Pada Perguruan Tinggi Dengan Teknik Restfull Dan Web Service. *Technomedia Journal*, 2(1), 26–36. <https://doi.org/10.33050/tmj.v2i1.313>