

**Penggabungan Fitur Local Binary Pattern (LBP) dan Ridge  
Density dalam Klasifikasi Gender Berdasarkan Sidik Jari  
Menggunakan Support Vector Machine**

Skripsi



oleh:

**Dedi Yanto**  
**71200593**

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA

2024

**Penggabungan Fitur Local Binary Pattern (LBP) dan Ridge  
Density dalam Klasifikasi Gender Berdasarkan Sidik Jari  
Menggunakan Support Vector Machine**

Skripsi



Diajukan kepada Program Studi Informatika Fakultas Teknologi Informasi  
Universitas Kristen Duta Wacana  
Sebagai Salah Satu Syarat dalam Memperoleh Gelar  
Sarjana Komputer

Disusun oleh

**Dedi Yanto**

**71200593**

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA

2024

## Daftar Isi

KATA PENGANTAR .....	<b>Error! Bookmark not defined.</b>
Daftar Isi.....	1
INTISARI.....	3
ABSTRACT.....	3
BAB I .....	5
PENDAHULUAN .....	5
1.1. Latar Belakang Masalah.....	5
1.2. Perumusan Masalah .....	6
1.3. Batasan Masalah.....	6
1.4. Tujuan Penelitian .....	6
1.5. Manfaat Penelitian .....	6
1.6. Metodologi Penelitian .....	7
1.7. Sistematika Penulisan .....	8
BAB II.....	9
TINJAUAN PUSTAKA DAN DASAR TEORI .....	9
2.1 Tinjauan Pustaka .....	9
2.2 Dasar Teori.....	9
BAB III .....	29
METODOLOGI PENELITIAN.....	29
3.1 Objek Penelitian .....	30
3.2 Analisis Kebutuhan Sistem .....	30
3.3 Perancangan Penelitian .....	31
BAB IV .....	41
IMPLEMENTASI DAN ANALISIS SISTEM.....	41
4.1 Implementasi sistem.....	41
4.2.1 Preprocessing Citra .....	41
4.2.2 Ekstraksi Fitur Citra .....	46
4.2.3 Penggabungan Fitur .....	54

4.2.4	Pembuatan Model SVM.....	56
4.2	Analisis Sistem.....	58
4.2.1	Pengujian terhadap 4730 sidik jari laki-laki dan 1230 sidik jari wanita	58
4.2.2	Pengujian terhadap 1230 sidik jari laki-laki dan 1230 sidik jari wanita	63
4.2.3	Pengujian terhadap 4770 sidik jari laki-laki dan 4770 sidik jari wanita	84
4.2.4	Kesimpulan menyeluruh dari ketiga percobaan.....	102
Bab V	.....	104
Kesimpulan	.....	104
5.1	Kesimpulan .....	104
5.2	Saran.....	105
DAFTAR PUSTAKA	.....	105
LAMPIRAN A	.....	109
KODE SUMBER PROGRAM	.....	109
LAMPIRAN C	.....	<b>Error! Bookmark not defined.</b>
KARTU KONSULTASI DOESEN 2	.....	<b>Error! Bookmark not defined.</b>

## **INTISARI**

### **Penggabungan Fitur Local Binary Pattern (LBP) dan Ridge Density dalam Klasifikasi Gender Berdasarkan Sidik Jari Menggunakan Support Vector Machine**

Oleh

Dedi Yanto

71200617

Penelitian ini bertujuan untuk mengklasifikasikan gender berdasarkan sidik jari dengan menggabungkan fitur Local Binary Pattern (LBP) dan Ridge Density menggunakan Support Vector Machine (SVM). Metode pre-processing citra yang digunakan meliputi pengurangan noise, binarisasi, penghalusan citra, dan skeletonisasi. Setelah pre-processing, fitur LBP dan Ridge Density diekstraksi dari citra sidik jari untuk digunakan dalam proses klasifikasi. Hasil penelitian menunjukkan bahwa model SVM dengan penggabungan fitur LBP dan Ridge Density berhasil mencapai akurasi tertinggi sebesar 87%. Kesimpulan dari penelitian ini adalah bahwa metode yang diusulkan efektif dalam mengklasifikasikan gender berdasarkan sidik jari, dan penggabungan kedua fitur tersebut memberikan hasil yang signifikan dalam meningkatkan akurasi klasifikasi.

**Kata Kunci:** Local Binary Pattern, Ridge Density, Support Vector Machine, Sidik Jari, Klasifikasi Gender

## **ABSTRACT**

### **Penggabungan Fitur Local Binary Pattern (LBP) dan Ridge Density dalam Klasifikasi Gender Berdasarkan Sidik Jari Menggunakan Support Vector Machine**

By

Dedi Yanto

71200593

This study aims to classify gender based on fingerprints by combining Local Binary Pattern (LBP) and Ridge Density features using Support Vector Machine (SVM). The image pre-processing methods used include noise reduction, binarization, image smoothing, and skeletonization. After pre-processing, LBP and Ridge Density features are extracted from the fingerprint images for the classification process. The results of the study show that the SVM model with the combination of LBP and Ridge Density features achieved the highest accuracy of 87%. The conclusion of this study is that the proposed method is effective in classifying gender based on fingerprints, and the combination of these two features provides significant results in improving classification accuracy.

**Keywords:** Local Binary Pattern, Ridge Density, Support Vector Machine, Fingerprints, Gender Classification

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Penelitian ini akan mengkaji bagaimana sidik jari manusia dapat digunakan untuk mengidentifikasi gender seseorang. Alur kulit sidik jari memiliki karakteristik khusus yang dapat membedakan gender seseorang. Sejauh pengetahuan penulis hubungan antara pola sidik jari dan gender seseorang masih dalam tahap penelitian dan banyak peneliti yang telah melakukan dengan berbagai metode. Penelitian ini merupakan lanjutan penelitian dari (Suwarno, 2022) yang meneliti bagaimana hubungan antara ketebalan ridge terhadap gender pada seorang individu. Proses pengidentifikasi gender pada penelitian ini akan dilakukan dengan membangun sebuah model *machine learning*. Model akan dilatih untuk dapat memprediksi gender berdasarkan dataset sidik jari sekunder.

Proses untuk melakukan pengklasifikasian gender berdasarkan sidik jari sangat bergantung pada ekstraksi fitur yang dilakukan pada gambar sidik jari. Proses ekstraksi fitur ini dilakukan pertama kali untuk mengenali pola yang ada didalam gambar sidik jari yang diproses, sehingga kualitas gambar sidik jari menjadi hal yang sangat krusial. Pada penelitian ini, peneliti akan menggabungkan ekstraksi fitur pada metode ekstraksi Local Binary Pattern (LBP) dan fitur ridge density untuk melakukan klasifikasi gender dengan mengukur performa klasifikasi menggunakan algoritma Support Vector Machine (SVM).

Sejauh yang penulis ketahui saat ini belum ada penelitian yang menggabungkan fitur yang didapatkan dari ekstraksi LBP dan ridge density dalam pengklasifikasian gender pada sidik jari. Metode ekstraksi ini dipilih karena memiliki karakteristik unik yang dapat membantu dalam klasifikasi gender. Ekstraksi LBP akan berfokus pada pola tekstur di dalam gambar sidik jari dan ekstraksi ridge density akan berfokus pada pola geometris di dalam gambar sidik jari. Dalam konteks pengklasifikasian gender, perbedaan dalam pola-pola ini dapat mengandung informasi penting yang membedakan antara sidik jari pria dan wanita.

## **1.2. Perumusan Masalah**

Berdasarkan yang sudah dijelaskan pada latarbelakang, rumusan masalah yang dapat diuraikan sebagai berikut:

1. Bagaimana hasil akurasi yang dihasilkan dengan menggabungkan fitur ekstraksi Local Binary Pattern (LBP) dan ridge density dalam klasifikasi gender berdasarkan sidik jari?

## **1.3. Batasan Masalah**

Batasan Masalah dari penelitian “Perbandingan Ekstraksi Fitur LBP dan Minutiae dalam Klasifikasi Gender Berdasarkan Sidik Jari Menggunakan SVM” dapat dijabarkan sebagai berikut:

1. Database gambar sidik jari yang dipakai hanya menggunakan dataset dari Sokoto Conventry Fingerprint Dataset (SOCOFing) yang tersedia pada platform kaggle
2. Penelitian ini akan menggunakan 4770 sidik jari laki-laki dan 4770 sidik jari perempuan dalam dataset Sokoto Conventry Fingerprint Dataset (SOCOFing) dari total 54000 sidik jari pada dataset tersebut.

## **1.4. Tujuan Penelitian**

Berikut ini adalah tujuan dari penelitian ini:

1. Menganalisis akurasi metode ekstraksi fitur LBP dan *ridge density* dalam klasifikasi gender berdasarkan sidik jari
2. Menghasilkan luaran sistem klasifikasi gender dengan memaksimalkan fitur LBP dan *ridge density* yang akurat

## **1.5. Manfaat Penelitian**

Adapun manfaat penelitian yang diperoleh dari dilakukannya penelitian ini, antara lain:



1. Penelitian ini akan memberikan kontribusi pada pengembangan teknologi klasifikasi gender berdasarkan sidik jari
2. Penelitian ini dapat membantu meningkatkan akurasi identifikasi gender berdasarkan sidik jari dengan menggabungkan ekstraksi fitur LBP dan *ridge density*
3. Penelitian ini dapat menjadi dasar untuk penelitian lanjutan dalam pengembangan metode ekstraksi fitur untuk pengenalan gender berdasarkan sidik jari
4. Hasil penelitian ini dapat berguna bagi tim forensik dalam identifikasi pelaku kejahatan yang tidak diketahui atau mengidentifikasi mayat yang tidak dikenal
5. Penelitian ini dapat memiliki dampak dalam penelitian sosial terkait perbedaan gender dan karakteristik fisik pada sidik jari

#### **1.6. Metodologi Penelitian**

Dalam penelitian ini akan terdiri dari beberapa tahapan yaitu sebagai berikut:

##### **1. Tahap Pengumpulan Data**

Tahap ini akan mengumpulkan citra sidik jari yang terdapat didalam dataset, citra sidik jari asli pria dan wanita akan disimpan pada satu folder untuk diproses nantinya

##### **2. Tahap Implementasi Model**

Tahap implementasi model akan terdiri dari 3 langkah diantaranya adalah *pre-processing* citra sidik jari, ekstraksi fitur citra sidik jari, dan pelatihan model klasifikasi

##### **3. Tahap Evaluasi Model**

Tahap evaluasi model akan dilakukan untuk menguji seberapa tinggi tingkat akurasi klasifikasi yang dihasilkan oleh model.

## **1.7. Sistematika Penulisan**

Laporan/Proposal skripsi ini disusun dengan sistematika bagian pertama, terdiri dari empat bab: Bab 1 yaitu Pendahuluan yang berisi tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan pernyataan keaslian disertasi. Bab 2 yaitu Tinjauan Pustaka dan Landasan Teori yang berisi tinjauan pustaka tentang penelitian-penelitian terkait, dan berbagai tinjauan pustaka spesifik, yaitu tentang *image pre-processing*, *feature extraction*, ekstraksi fitur LBP, ekstraksi fitur ridge density, algoritma support vector machine dan teknik evaluasi model. Bab 3 yaitu Metodologi Penelitian akan membahas tentang objek penelitian, analisis kebutuhan sistem, dan perancangan penelitian. BAB IV Implementasi dan Analisis Sistem pada bab ini akan berisi hasil dan evaluasi dari sistem yang telah dilakukan. BAB V Kesimpulan dan Saran pada bab ini akan berisi hasil penelitian yang akan menjadi pedoman dalam pembuatan kesimpulan dan saran untuk pengembangan selanjutnya.

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian terkait klasifikasi gender berdasarkan gender dengan menggunakan metode LBP pernah dilakukan oleh Gornale, dkk. (2017). Pada penelitian ini mereka melakukan ekstraksi fitur tekstur untuk mendeteksi dan mengklasifikasikan gender melalui sidik jari dengan menggunakan algoritma K-Nearest Neighbor (KNN). Sebanyak 400 sidik jari asli dikumpulkan dalam penelitian ini dari berbagai kelompok usia masyarakat perkotaan dan pedesaan sebagai bagian dari eksperimen. Model yang dibuat pada penelitian ini mengekstrak tiga fitur didalam sidik jari, yaitu kualitas gambar, tekstur gambar, dan karakteristik yang terkait dengan ekstraktabilitas fitur.

Pada penelitian ini memanfaatkan framework yang berbeda-beda untuk mendapatkan pola struktur lokal pada citra sidik jari, yaitu Completed LSP (CLSP), Robust LSP (RLSP), Completed Noise-invariant LSP (CNLP) dan Completed Noise-invariant Global-structure Pattern (CNGP). Penelitian yang dilakukan memanfaatkan intensitas lokal, intensitas global, dan informasi kontras pada gambar untuk mendapatkan pola dalam sidik jari yang akan dikodekan menggunakan metode LBP dengan nilai threshold yang sudah dihitung. Nilai LBP yang didapatkan kemudian akan digabungkan menggunakan framework CLSP untuk menggabungkan pola LBP yang sudah didapat kedalam satu sinyal histogram. Hasil pengamatan eksperimental menunjukkan tingkat keberhasilan klasifikasi sebesar 95,88% dengan menggunakan fitur Local Binary Pattern (LBP).

Penelitian lain pernah dilakukan oleh Kruthi, dkk. (2019) untuk mengidentifikasi gender berdasarkan sidik jari dengan menggabungkan metode ekstraksi citra LBP dan LPQ. Penelitian ini melakukan percobaan pada dua dataset yang berbeda, dataset pertama menggunakan dataset sidik jari yang disediakan oleh SDUMLA-HMT secara publik dan dataset kedua menggunakan dataset sidik jari

yang mereka kumpulkan secara mandiri. Dataset tersebut mencakup sidik jari 348 orang yang terdiri dari 183 relawan laki – laki dan 165 relawan perempuan.

Pada penelitian ini dilakukan percobaan pada algoritma klasifikasi yang berbeda-beda untuk melakukan klasifikasi gender, seperti SVM, LDA, QDA dan KNN. Penelitian ini mencoba untuk menggunakan nilai ekstraksi fitur pada metode LBP dan LPQ untuk melakukan klasifikasi, dengan mengambil nilai ekstraksi secara terpisah maupun bersamaan. Hasil dari penelitian yang didapatkan menunjukkan jika algoritma klasifikasi SVM, memperoleh nilai akurasi paling tinggi dengan memanfaatkan nilai ekstraksi fitur LBP dan LPQ secara bersamaan.

Pada penelitian yang telah disebutkan diatas, dapat diketahui jika ternyata tekstur yang ada pada sidik jari dan wajah dapat mendeskripsikan gender seseorang. Selain pola tekstur, ternyata pola geometris yang ada didalam sidik jari dapat membedakan tiap individu, salah satunya adalah gender. Seperti penelitian yang dilakukan oleh Yamini, dkk (2023), pada 223 volunteer berasal dari Thiruporur Taluk, ditemukan jika *ridge density* dapat menentukan gender pada seseorang. *Ridge density* adalah banyaknya pola garis-garis (*ridges*) yang terbentuk pada sidik jari manusia. Penelitian yang telah dilakukan oleh Yamini, dkk (2023) pada masyarakat Tamil, menemukan jika nilai *ridge density* pada wanita berkisar antara 11,2 hingga 12,2 / mm<sup>2</sup>, dengan rata-rata sebesar 11,95 ridges/ mm<sup>2</sup> dan nilai *ridge density* pada pria berkisar antara 12,1 -13,7 ridges/mm<sup>2</sup>, dengan rata-rata sebesar 12,9 ridges/mm<sup>2</sup>. Hasil penelitian ini menunjukkan bahwa rata-rata *ridge density* pada pria lebih tinggi daripada wanita.

Namun banyak penelitian yang dilakukan mendapatkan kesimpulan jika *ridge density* yang dimiliki oleh wanita lebih tinggi dari pada pria. Hal ini dapat dilihat pada penelitian yang dilakukan oleh Andres, dkk. (2018) yang melakukan percobaan evaluasi terhadap sidik jari 213 orang dewasa Spanyol. Partisipan pada penelitian ini dibagi menjadi dua kelompok berdasarkan usia, yaitu kelompok Junior (18 - 30 tahun) dan kelompok Senior (55-60 tahun). *Ridge density* direkam dari area radial, ulnar, dan area proksimal pada ujung jari-jari partisipan.

Pada kelompok Junior, ditemukan jika *ridge density* di area radial pada pria sebesar 16,19 *ridge*/25mm<sup>2</sup> dan wanita sebesar 18,24 *ridge*/25mm<sup>2</sup>. Demikian juga

di area ulnar, *ridge density* pada pria ditemukan sebesar 14,63 *ridge/25mm<sup>2</sup>* dan wanita sebesar 16,44 *ridge/25mm<sup>2</sup>*. Sementara itu, di are proksimal, ditemukan sebesar 12,25 *ridge/25mm<sup>2</sup>* pada pria dan 13,27 *ridge/25mm<sup>2</sup>* pada wanita. Dari Hasil tersebut dapat diketahui jika *ridge density* yang dimiliki oleh pria lebih rendah ketimbang wanita, pada kelompok Junior.

Hasil yang sama juga didapatkan pada kelompok Senior, yaitu pada area radial nilai *ridge density* pada pria sebesar 14,22 *ridge/25mm<sup>2</sup>* dan pada wanita sebesar 15,81 *ridge/25mm<sup>2</sup>*. Hasil yang didapatkan di area ulnar ditemukan sebesar 12,81 *ridge/25mm<sup>2</sup>* pada pria dan 14,25 *ridge/25mm<sup>2</sup>* pada wanita. Hasil yang didapatkan di area proksimal, ditemukan sebesar 12,07 *ridge/25mm<sup>2</sup>* pada pria dan 13,05 *ridge/25mm<sup>2</sup>* pada wanita.

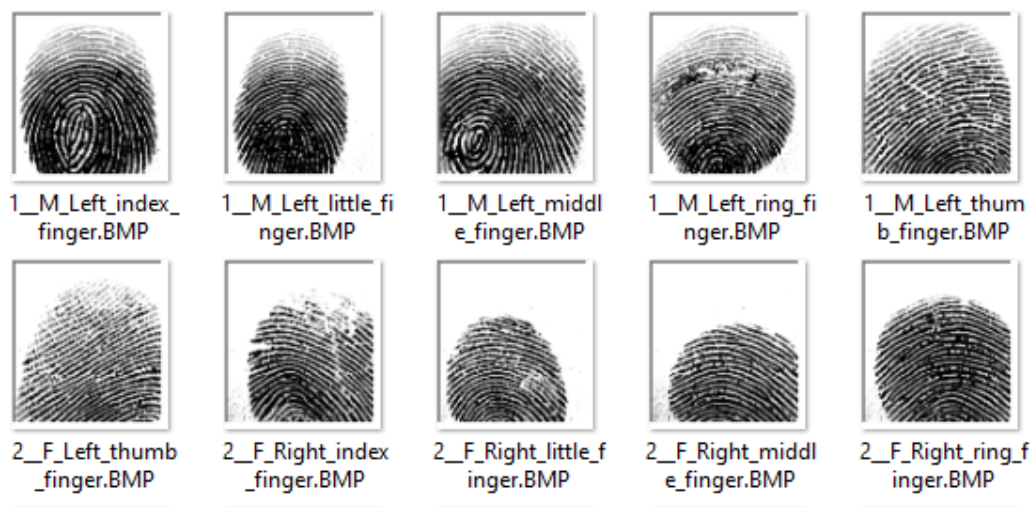
Kesimpulannya didapatkan jika terdapat hasil yang signifikan pada perbedaan *ridge density* yang dimiliki oleh pria dan wanita. Studi ini menunjukkan jika perbedaan *ridge density* tertinggi terlihat pada area radial. Dari kedua penelitian terkait *ridge density* didapatkan jika nilai pada fitur ini, dapat membantu membedakan gender pada sidik jari, karena antara pria dan wanita memiliki nilai *ridge density* yang signifikan.

Penelitian yang akan dilakukan kali ini bertujuan untuk menggabungkan kedua fitur yaitu fitur tekstur citra yang didapatkan pada hasil ekstraksi metode LBP dan *ridge density* pada citra sidik jari. Penelitian sebelumnya telah mengukur tingkat akurasi klasifikasi gender dengan menggunakan metode LBP pada sidik jari, dan penelitian lain menunjukkan jika *ridge density* dapat menjadi fitur dalam pembedaan gender pada sidik jari. Namun, penelitian sebelumnya hanya menggunakan salah satu fitur tersebut, sehingga pada penelitian ini akan menjalankan percobaan dengan menggabungkan kedua fitur ini untuk meningkatkan akurasi klasifikasi gender. Hasil penelitian ini tunjukkan untuk mengeksplorasi sejauh mana informasi tekstur dan karakteristik geometris pada sidik jari dapat berhubungan satu sama lain untuk meningkatkan akurasi klasifikasi gender.

## 2.2 Dasar Teori

### 2.2.1 Image Pre-Processing

Pemrosesan citra sangat penting untuk menghasilkan kualitas citra yang bagus, karena dapat mengurangi *noise* pada citra dan untuk meningkatkan kualitas dari gambar asli, agar dapat diproses dengan lebih optimal. Dalam citra sidik jari pemrosesan citra sangat bergantung pada metode yang digunakan, agar proses ekstraksi fitur dapat maksimal.



Gambar 2.1 Sampel dataset sidik jari yang akan digunakan

Gambar 2.1 menunjukkan sampel dari *dataset* gambar sidik jari yang akan dipakai pada penelitian ini. Dari sampel dataset tersebut, jika diperhatikan secara sekilas gambar sidik jari masih kurang baik dikarenakan ada citra sidik jari yang hanya terambil sebagian dan ada citra yang terlihat lebih gelap dari pada citra lainnya, sehingga detail sidik jari menjadi tidak konsisten. Oleh sebab itu citra sidik jari pada dataset yang digunakan, perlu dilakukan *image pre-processing*, hal ini bertujuan untuk meningkatkan kualitas citra agar fitur yang diekstraksi mendapatkan hasil yang maksimal. Tahapan *image pre-processing* bergantung pada kebutuhan dan citra yang digunakan. Berikut ini beberapa tahap yang akan dipakai untuk melakukan *pre-processing image* pada penelitian ini.

#### 1 Pengurangan Noise

Proses pengurangan noise merupakan proses untuk menghilangkan atau mengurangi noise yang diakibatkan oleh kerusakan di dalam citra. Noise terjadi

ketika piksel yang terdapat pada citra menunjukkan nilai intensitas yang berbeda daripada nilai piksel sebenarnya yang diperoleh dari gambar Hambal, dkk. (2015). Tujuan dari pengurangan noise ini adalah untuk meningkatkan kualitas gambar sehingga lebih mudah untuk dianalisa. Pengurangan noise diperlukan untuk menjaga konsistensi intensitas piksel yang ada pada citra sidik jari. Pengurangan noise ini akan bermanfaat agar proses identifikasi pola tekstur dalam citra sidik jari dapat lebih akurat, karena intensitas piksel yang ada didalam citra menjadi lebih konsisten dan dapat diandalkan. Proses ini akan di gunakan saat mengekstraksi fitur tekstur pada algoritma Local Binary Pattern (LBP) pada penelitian ini. Contoh hasil citra yang mengalami noise dapat dilihat pada Gambar 2.2 sebelum terjadi *noise* dan Gambar 2.3 setelah terjadi *noise*. Gambar 2.3 menunjukkan terjadinya *Gaussian noise*.



Gambar 2.2 Setelah noise gambar di kurangi.

Sumber (Hambal, Pei, & Ishabailu, 2015)



Gambar 2. 3 saat terjadi noise.

Sumber (Hambal, Pei, & Ishabailu, 2015)

Terdapat banyak sekali metode yang dapat digunakan untuk menghilangkan noise didalam citra, secara umum metode yang digunakan terbagi menjadi dua cabang yaitu metode *spatial filtering* dan *transform domain filtering*. *Mean filter* adalah salah satu metode yang dapat digunakan untuk menghilangkan noise didalam citra dan termasuk kedalam cabang metode *spatial filtering*. Rumus yang digunakan untuk menghitung nilai *mean filter* dapat dilihat pada persamaan (2.1) (Yuwono, 2010).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n X_i \quad (2.1)$$

Keterangan:

$\bar{x}$  = Nilai rata-rata (mean)

N = Jumlah data

$X_i$  = Nilai ke-i

i = Nilai awal

## 2 Peningkatan Kontras

Peningkatan kontras merupakan teknik yang digunakan untuk mengubah distribusi intensitas piksel dalam citra sidik jari, supaya memperjelas perbedaan antara area gelap dan terang. Tujuan dari tahap ini adalah agar pola tekstur dan gambar sidik



jari menjadi lebih jelas dan mudah untuk diidentifikasi, sehingga mempermudah proses ekstraksi fitur. Proses peningkatan kontras pada citra sangat berguna dalam membuat perbedaan warna dan kecerahan pada gambar menjadi lebih jelas, sehingga membantu untuk meningkatkan efek visual dan memunculkan detail-detail yang terdapat pada citra Zhang, dkk. (2023). Proses ini akan sangat berguna nantinya saat mengekstraksi fitur tekstur pada algoritma Local Binary Pattern (LBP) pada penelitian ini. Gambar 2.4 menunjukkan proses peningkatan kontras menggunakan teknik *image adjustment*, dimana gambar sebelah kiri menunjukkan gambar asli dan sebelah kanan menunjukkan gambar setelah peningkatan kontras.



Gambar 2.4 Peningkatan Kontras pada gambar sidik jari menggunakan teknik image adjustment.

Sumber (Asamoah, Oppong, Oppong, & Danso, 2018)

Salah satu metode yang dapat dipakai untuk meningkatkan kontras pada citra adalah *histogram equalization*. Rumus umum yang digunakan untuk menggunakan metode *histogram equalization* dapat dilihat pada persamaan (2.2).

$$h(v) = \text{round} \left( \frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right) \quad (2.2)$$

Keterangan:

$h(v)$  adalah nilai piksel yang telah diseimbangkan.

$cdf(v)$  adalah fungsi distribusi kumulatif dari piksel dengan intensitas  $v$ .

$cdf\_min$  adalah nilai minimum dari  $cdf(v)$ .

$M$  dan  $N$  adalah dimensi gambar (lebar dan tinggi).

$L$  adalah jumlah level intensitas dalam gambar.

$v$  adalah intensitas piksel dalam gambar

### 3 Binarization

Binarisasi merupakan proses untuk mengubah gambar grayscale menjadi gambar biner (hitam dan putih), sehingga piksel-piksel yang ada pada citra hanya memiliki dua nilai intensitas, yang biasanya ditandai dengan nilai 0 atau 1. Tujuannya adalah untuk mengidentifikasi objek dan latar belakang dalam citra dengan jelas. Dalam algoritma binarisasi pada citra sidik jari akan membuat citra dengan tipe 1-bit, dimana nilai 0 akan mewakili bagian *ridges* pada sidik jari dan nilai 1 akan mewakili *valleys* pada sidik jari Kumar dan Verma (2012). Proses pemorsesan citra ini sangat berguna untuk mendapatkan nilai fitur ketebalan *ridges* yang akan membantu membedakan objek sidik jari dari latar belakang, sehingga proses ekstraksi fitur ketebalan *ridges* dapat lebih cepat dan akurat. Gambar 2.5 menunjukkan contoh citra sidik jari sebelum binarisasi dan Gambar 2.6 sesudah binarisasi.



Gambar 2.5 Gambar sidik jari awal.

Sumber Kumar dan Verma (2012)



Gambar 2.6 Gambar sidik jari setelah binarisasi.

Sumber: Kumar dan Verma (2012)

Terdapat beberapa metode yang dapat dilakukan untuk mengubah citra abu-abu menjadi citra biner. Salah satu metode yang dapat digunakan adalah metode otsu, metode ini akan melakukan *thersholding* pada citra abu-abu dengan rumus yang dapat dilihat pada persamaan (2.3) Saddami, dkk (2019).

$$\sigma^2 B(T) = \omega_1(T)\omega_2(T)[\mu_1(T) - \mu_2(T)]^2 \quad (2.3)$$

Keterangan:

$\sigma^2 B(T)$  adalah varians antara dua kelas piksel.

$\omega_1(T)$  adalah bobot kelas piksel latar belakang (background).

$\omega_2(T)$  adalah bobot kelas piksel objek (foreground).

$\mu_1(T)$  adalah rata-rata intensitas piksel dalam kelas piksel latar belakang.

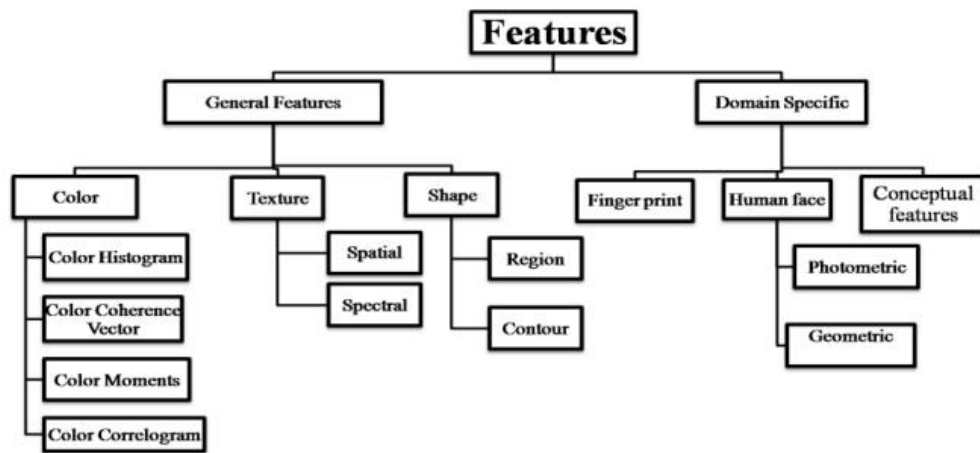
$\mu_2(T)$  adalah rata-rata intensitas piksel dalam kelas piksel objek.

### 2.2.2 Feature Extraction

*Feature extraction* dalam pengolahan citra merupakan proses untuk mengidentifikasi dan mengekstraksi informasi-informasi penting dari sebuah citra yang dilakukan dengan tujuan untuk melakukan proses analisis pada citra tersebut. Tujuan dari dilakukannya *feature extraction* adalah untuk mendapatkan fitur pada citra, yang dapat digunakan untuk melakukan pengenalan pada citra dan menghasilkan kumpulan fitur yang dapat diklasifikasi. Fitur yang diekstraksi pada sebuah citra dapat digunakan untuk memperoleh informasi pada citra, seperti informasi tepi, sudut, *ridge*, *blobs*, dan tekstur pada sebuah citra (Sharma & Abrol, 2020).

Berdasarkan jurnal yang dibuat oleh Sharma (2020) ekstraksi fitur berguna untuk mengurangi dimensi data asli menjadi lebih kecil yang membantu dalam menganalisa, mengolah, dan mengambil keputusan menjadi lebih efisien. Proses ekstraksi fitur sangat penting dalam mengklasifikasikan objek berdasarkan citra. Proses ini dapat menghasilkan fitur yang tidak berlebihan dari data asli dengan mengambil fitur-fitur yang relevan sesuai keperluan, untuk mendapatkan pola-pola yang ada pada citra. Pada jurnal yang diterbitkan oleh Salau dan Jain (2019) fitur yang diekstraksi pada sebuah citra dapat dibagi menjadi dua klasifikasi yaitu

*general features (GF)* dan *domain-specific features (DSF)*. Klasifikasi GF mengekstraksi fitur seperti warna, bentuk dan tekstur pada citra, sedangkan DSF mengekstraksi fitur yang bergantung pada aplikasi seperti fitur pada wajah manusia, sidik jari, dan fitur konseptual. Kedua klasifikasi ini diklasifikasikan lagi menjadi beberapa kategori yang ditunjukkan pada Gambar 2.7.



Gambar 2.7 Klasifikasi teknik ekstraksi fitur.

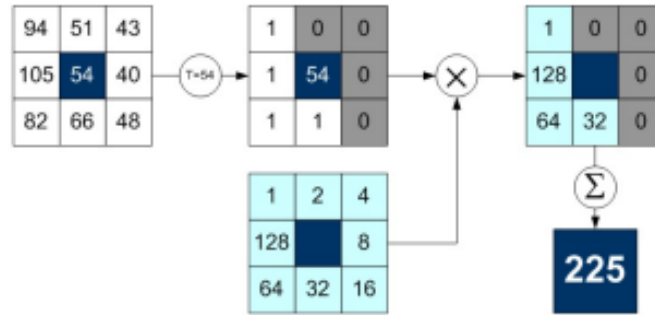
Sumber (Salau & Jain, 2019)

### 2.2.3 Local Binary Pattern (LBP)

LBP merupakan salah satu metode yang digunakan dalam penelitian ini sebagai metode untuk melakukan ekstraksi fitur pada gambar sidik jari. LBP digunakan untuk mengambil fitur tekstur yang ada didalam citra. Dalam penelitian yang dilakukan oleh Retnoningrum, dkk. (2019) dia melakukan ekstraksi ciri pada citra telapak tangan manusia untuk melakukan proses identifikasi seseorang. Penelitian tersebut memaparkan jika metode LBP sangat baik dalam melakukan ekstraksi fitur tekstur pada citra abu-abu (*grayscale*) dan dapat bekerja dengan baik dalam citra yang memiliki pencahayaan berbeda. LBP akan menghasilkan nilai sebaran histogram pada citra sidik jari yang dapat digunakan untuk menggambarkan pola tekstur lokal dan bentuk global dari citra, pada penelitian Retnoningrum, dkk. (2019) berarti citra telapak tangan.

Berbeda dengan yang dilakukan oleh Retnoningrum, dkk. (2019), penelitian kali ini akan memanfaatkan metode LBP untuk mendapatkan fitur tekstur yang

dimiliki oleh sidik jari sebagai bahan untuk melakukan klasifikasi gender. Metode LBP digunakan mendapatkan deskripsi tekstur pada sebuah citra, pada penelitian ini, LBP akan dibuat dengan cara membagi citra sidik jari menjadi beberapa wilayah-wilayah,  $W_0, W_1, W_2, \dots, W_m$ . Wilayah yang dibagi tersebut bertujuan untuk menyederhanakan sidik jari menjadi matrix 3x3 piksel yang digunakan untuk mencari nilai histogram pada gambar tersebut. Dari matrix 3x3 tersebut metode LBP akan memilih salah satu piksel menjadi nilai tengah yang nantinya akan digunakan untuk membandingkan nilai piksel tersebut dengan nilai setiap piksel tetangga yang kemudian akan menghasilkan nilai matrix baru menjadi nilai 0 atau 1 pada proses *thresholding*.



Gambar 2. 8 Ilustrasi Tahap Kalkulasi dengan Metode LBP.

Sumber (Prakasa, 2015)

Proses *thresholding* yang ditunjukkan pada Gambar 2.8, dilakukan dengan persyaratan apabila nilai piksel tetangga kurang dari nilai tengah yaitu nilai yang dipilih sebagai acuan piksel maka akan diberi nilai 0, sedangkan jika nilai piksel tetangga lebih dari nilai tengah maka akan diberi nilai 1. Proses *thresholding* dapat dilihat pada persamaan (2.4) dan (2.5). Persamaan (2.4) merupakan proses matematis untuk mengubah nilai biner yang didapatkan pada proses *thresholding* menjadi nilai desimal dan persamaan (2.5) merupakan proses *thresholding*.

$$LBP(x_c, y_c) = \sum_{p=0}^7 S(g_p - g_c) 2^p \quad (2.4)$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.5)$$

Keterangan

$LBP(x_c, y_c)$  : adalah nilai LBP pada piksel tengah suatu wilayah

$g_p$  : adalah nilai piksel tetangga

$g_c$  : adalah nilai piksel tengah.

nilai index p: merupakan urutan pada piksel tetangga.

#### 2.2.4 Ridge Denisity

Sidik jari (*fingerprint*) merupakan salah satu jenis data biometrik yang dimiliki oleh manusia. Sidik jari memiliki pola-pola dan karakteristik yang unik dan permanen pada setiap manusia, pola-pola yang terbentuk pada sidik jari memiliki visual garis-garis dan lekukan yang terdapat pada permukaan 10 ujung jari manusia. Dalam jurnal yang dibuat oleh (Sharma & Chawla, 2016) mengatakan jika sidik jari yang dimiliki oleh manusia bersifat unik dan akan tetap sama selama sisa hidupnya, jika sidik jari tersebut tidak mengalami luka, bahkan pada orang yang terlahir kembar identik, mereka tetap mempunyai sidik jari yang berbeda.

Sidik jari memiliki karakteristik yang unik pada tiap manusia, salah satunya adalah ridge density atau kepadatan lipatan. Ridge density merupakan ketebalan garis-garis (*ridge*) yang terdapat dalam area sidik jari, yang dapat digunakan untuk identifikasi individu dan penentuan jenis kelamin. Kepadatan ini dapat bervariasi dari satu individu ke individu lainnya, serta di berbagai bagian dari sidik jari yang sama (Chauhan & Shukla, 2017).

Berdasarkan dari penelitian yang telah dilakukan untuk mencari hubungan antara *ridge density* dengan gender, didapatkan jika wanita cenderung memiliki *ridge density* lebih banyak daripada pria (Chauhan & Shukla, 2017), dari hasil penelitian tersebut dapat disimpulkan jika ada perbedaan yang signifikan pada *ridge density* pada pria dan wanita. Hasil perbandingan pada *ridge density* yang dimiliki oleh pria dan wanita pada penelitian (Chauhan & Shukla, 2017), dapat dilihat pada Tabel 2.1. Hasil tersebut didapatkan dari percobaan yang dilakukan pada berbagai populasi di India. Hasil yang ditunjukkan pada Tabel 2.1 memberikan penjelasan mengenai signifikansi perbedaan *ridge density* pada sidik jari pria dan wanita.

Tabel 2.1 Studi Perbandingan Ridge Density pada berbagai populasi di India Berdasarkan Pada Pria dan Wanita.

Sumber (Chauhan & Shukla, 2017)

Wilayah sampel	Jumlah sampel yang diambil	Rentang Usia (Tahun)	Rata-rata jumlah <i>ridge</i> / 25mm <sup>2</sup>	Kesimpulan
India Selatan	100 – Pria 100 – Wanita	18-80	Pria – 14 Wanita – 16	Wanita mempunyai jumlah <i>ridge</i> lebih tinggi dari pada Pria
India Tengah	100 – Pria 100 – Wanita	18-30	Pria – 13 Wanita – 15	Wanita mempunyai jumlah <i>ridge</i> lebih tinggi dari pada Pria
India Utara	97 – Pria 97 – Wanita	18-25	Pria – 13 Wanita – 16	Wanita mempunyai jumlah <i>ridge</i> lebih tinggi dari pada Pria
Uttarakhand	100 – Pria 100 – Wanita	18-60	Pria – 14 Wanita – 16	Wanita mempunyai jumlah <i>ridge</i> lebih tinggi dari pada Pria
Punjabi	50 – Pria 50 – Wanita	18-50	Pria – 14 Wanita – 16	Wanita mempunyai jumlah <i>ridge</i> lebih tinggi dari pada Pria
Chandigarh	50 – Pria 50 – Wanita	18-40	Pria – 14 Wanita – 15	Wanita mempunyai jumlah <i>ridge</i> lebih tinggi dari pada Pria

Penelitian ini akan melakukan perhitungan fitur *ridge density* berdasarkan penelitian yang pernah dilakukan oleh Abdullah, dkk (2015) yang dapat dilihat pada persamaan (2.6).

$$Ridge\ density = \frac{Jumlah\ Ridge}{25mm^2} \quad (2.6)$$

Keterangan:

- Jumlah ridge adalah jumlah garis atau lipatan dalam area sidik jari
- $25mm^2$  merupakan area ukuran yang ditentukan

Setelah perhitungan *ridge density* telah dilakukan, selanjutnya adalah menghitung rata-rata *ridge density* yang dimiliki oleh individu tersebut untuk dijadikan sebagai fitur klasifikasi. Rumus untuk mendapatkan rata-rata *ridge density* dapat dilihat pada persamaan (2.7).

$$\mu = \frac{\sum_{n=1}^5 Xn}{N} \quad (2.7)$$

Keterangan:

$\mu$  = rata-rata *ridge density*

$\sum_{n=1}^5 Xn$  = jumlah semua *ridge density* pada kelima jari untuk tangan kanan maupun tangan kiri

$N$  = jumlah jari pada tangan, yaitu 5.

### 2.2.5 Support Vector Machine (SVM)

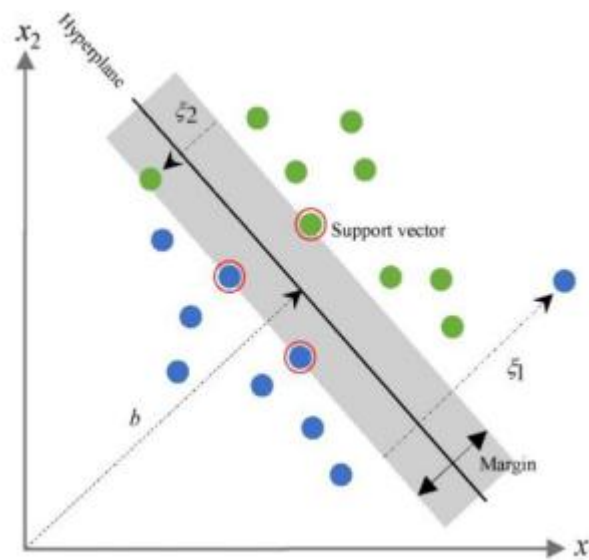
*Support Vector Machine* (SVM) merupakan algoritma klasifikasi yang dipakai pada penelitian ini. SVM merupakan salah satu algoritma dalam *supervised learning* yang digunakan untuk melakukan klasifikasi dan regresi. SVM dipilih karena kemampuannya yang sangat baik dalam mengklasifikasikan dua kelas yang berbeda (*binary classification*), karena desain SVM didesain untuk masalah klasifikasi biner Khouma, dkk. (2019). SVM digunakan untuk mencari *hyperlane* paling optimal dengan margin terjauh pada pemisahan dua buah kelas yang dibutuhkan dari *input space* Abdi, dkk. (2022).

*Hyperlane* adalah garis atau bidang yang membagi dua buah kelas pada kasus ini pria dan wanita menjadi dua kelas yang perlu dipisahkan pada *feature space*. *Feature space* adalah ruang vektor berdimensi tinggi yang berisi nilai vektor data *input*, yang merepresentasikan data yang didapatkan pada algoritma SVM. Untuk mengubah data *input* dari *input space* ke dalam *feature space* dibutuhkan



kernel yang digunakan untuk memisahkan kedua sampel klasifikasi secara *linear* melalui transformasi yang sesuai dalam ruang berdimensi tinggi, setelah memasukan data input ke dalam *input space*, SVM akan mencoba untuk menemukan *hyperlane* yang digunakan untuk memisahkan kedua kelas kedalam input space. Abdi, dkk. (2022).

Terdapat beberapa fungsi kernel yang umum digunakan dalam SVM, yaitu *linear kernel*, *polynomial kernel*, *radial basis function (RBF) kernel*, dan *sigmoid kernel* cevik, dkk (2015). Ilustrasi dari klasifikasi SVM dapat dilihat pada Gambar 2.9.



Gambar 2.9 Ilustrasi dari SVM.

Sumber (Khouma, Diop, Fall, Ndiaye, & Farssi, 2019)

Gambar 2.9 menunjukkan ilustrasi *hyperplane* dan margin pada SVM. Terlihat pada gambar tersebut menunjukkan data input yang terdiri dari dua fitur yang di tandai dengan titik berwarna biru dan hijau. Margin adalah jarak antara *hyperplane* dan data terdekat dari masing-masing kelas. Margin pada Gambar 2.9 ditunjukkan pada area berwarna abu-abu. Data-data yang terpisah oleh *hyperplane* menandakan data tersebut memiliki kelas yang berbeda. *Support vector* merupakan

data input yang berada paling dekat dengan dengan margin seperti yang terlihat pada Gambar 2.9.

Dalam buku *The Hundred-Page Machine Learning Book* (Burkov, 2019) dijelaskan jika setiap vektor fitur dalam penelitian ini berarti setiap fitur didalam citra sidik jari akan dijadikan sebagai sebuah titik dalam *high-dimensional space* (didalam penelitian ini *space*-nya adalah total fitur pada citra sidik jari). Algoritma SVM akan menempatkan semua vektor fitur pada *imaginary dimensional plot*, dengan jumlah dimensi sejumlah total fitur yang terdapat pada citra sidik jari. Algoritma SVM kemudian menggambar garis *imaginary dimensional (hyperplane)* yang memisahkan input data dengan label positif dan data dengan label negatif. Dalam machine learning, batas yang memisahkan data-data dari kelas yang berbeda biasa disebut sebagai *decision boundary*.

Persamaan dari *hyperplane* diberikan oleh dua parameter, yaitu vektor bernilai riil  $W$  dengan dimensi yang sama dengan vektor fitur input  $x$ , dan bilangan riil  $b$ , seperti yang terlihat pada persamaan (2.8).

$$wx - b = 0 \quad (2.8)$$

Dimana ekspresi  $wx$  berarti  $w(1) \times x(1) + w(2) \times x(2) + \dots + w(D) \times x(D)$ , dan  $D$  adalah jumlah dimensi dari vektor fitur  $x$ . Persamaan (2.8) digunakan untuk menentukan *hyperplane* yang memisahkan kelas-kelas secara linear dengan memaksimalkan margin antara *hyperplane* dan titik-titik terdekat dari setiap kelas. Untuk menentukan label data dari beberapa masukkan fitur vektor  $x$  pada saat pelatihan model, dapat dilihat pada persamaan (2.9).

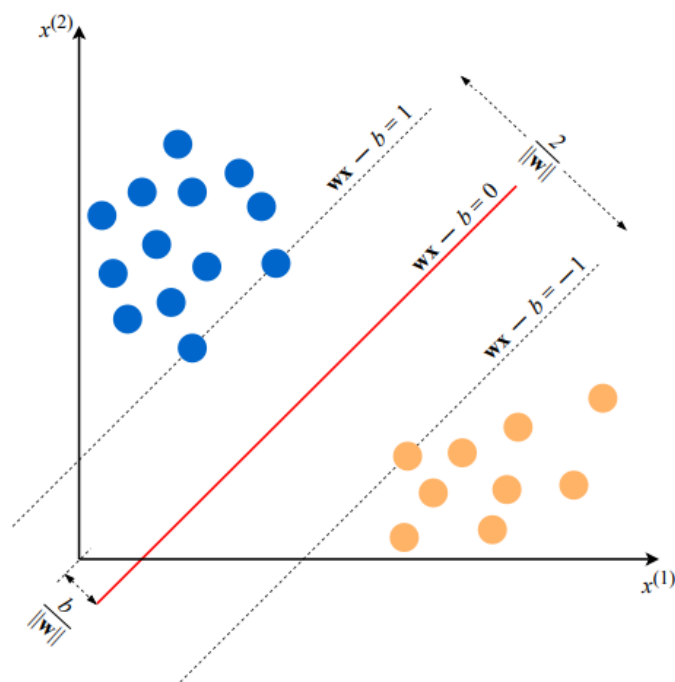
$$y = \text{sign}(wx - b) \quad (2.9)$$

Dimana *sign* merupakan operator matematika yang mengambil nilai apa pun sebagai masukan dan akan mengembalikan +1 jika masukan adalah bilangan positif atau -1 jika masukan adalah bilangan negatif. Persamaan (2.9) akan digunakan untuk menentukan nilai yang optimal untuk parameter  $w$  dan  $b$ , sehingga membantu dalam menghitung prediksi untuk data pelatihan dan melakukan penyesuaian parameter  $w$  dan  $b$  agar model dapat meminimalkan kesalahan

prediksi. Sehingga tujuan dari algoritma SVM adalah untuk memanfaatkan dataset yang diberikan kemudian menemukan nilai-nilai optimal  $w^*$  dan  $b^*$  untuk parameter  $w$  dan  $b$ . Ketika model machine learning yang dilatih menggunakan persamaan (2.9) mendapatkan nilai optimal, model  $f(x)$  akan memprediksi data baru dengan fungsi yang terlihat pada persamaan (2.10).

$$f(x) = \text{sign}(w * x - b) \quad (2.10)$$

Sehingga dalam penelitian ini untuk memprediksi sidik jari masuk kedalam kelas laki-laki atau perempuan menggunakan model SVM, maka perlu mendapatkan vektor fitur dari citra sidik jari, yang kemudian nilai tersebut akan dikalikan dengan nilai  $w^*$ , kemudian dikurangi dengan nilai  $b^*$  dan mengambil nilai sign dari hasil yang didapatkan. Nilai yang didapat dari persamaan (2.10) akan memberikan keluaran +1 yang artinya laki-laki dan -1 yang artinya perempuan.



**Gambar 2.10 Contoh model SVM dari vektor fitur dua dimensi**

Sumber (Burkov, 2019)

Elemen yang memisahkan antara contoh data positif dari yang negatif dalam hyperplane disebut margin. Margin adalah jarak antara data terdekat dari dua kelas, yang didefinisikan oleh sebuah *decision boundary*. Jarak margin yang besar

berkontribusi dalam generalisasi yang semakin baik, artinya model akan semakin baik dalam melakukan klasifikasi data baru yang dimasukkan. Untuk mengoptimalkan jarak margin, diperlukan untuk meminimalkan nilai Euclidean norm dari  $w$  yang dilambangkan dengan  $\|w\|$  dan dengan fungsi yang digunakan untuk mendapatkan margin terbesar seperti yang dapat dilihat pada persamaan (2.11).

$$\sqrt{\sum_{j=1}^D (w^{(j)})^2} \quad (2.11)$$

Sehingga masalah optimisasi yang harus diselesaikan yaitu meminimalkan  $\|w\|$  dengan batasan pada  $y_i(wx_i - b) \geq 1$  untuk  $i = 1, \dots, N$  (Burkov, 2019). Pada klasifikasi dua kelas, permasalahan dan solusinya dapat digambarkan secara visual, seperti yang terlihat pada Gambar 10. Dalam visualisasi ini, lingkaran biru dan oranye menggambarkan contoh positif dan negatif, sementara garis yang merepresentasikan  $wx - b = 0$  disebut sebagai *decision boundary*.

Untuk mengoptimisasi transformasi data input kedalam ruang dimensi vektor dalam model SVM dibutuhkan yang namanya fungsi kernel atau dapat disebut kernels. Kernels menyediakan cara efisien untuk melakukan perhitungan dalam ruang dimensi yang lebih tinggi tanpa harus secara eksplisit melakukan transformasi data ke ruang tersebut. memungkinkan SVM untuk bekerja dengan baik dalam menangani data yang tidak dapat dipisahkan secara linier di ruang dimensi rendah dengan memproyeksikan data ke ruang dimensi yang lebih tinggi, di mana pemisahan antara kelas-kelas data menjadi lebih mudah dilakukan. Jadi, secara efektif, kernels memungkinkan SVM untuk menemukan hiperplane pemisah yang optimal antara kelas-kelas data bahkan di ruang dimensi yang lebih tinggi, tanpa perlu transformasi vektor secara eksplisit. Banyak fungsi kernel yang tersedia, namun fungsi kernel yang paling banyak digunakan adalah RBF kernel (Burkov, 2019), dimana persamaan (2.12) menunjukkan fungsi dari *radial basis function* atau RBF kernel.

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \quad (2.12)$$

Dimana  $\|x - x'\|^2$  adalah nilai kuadra dari *Euclidean distance* antara dua vektor fitur. Dan untuk mendapatkan nilai dari *Euclidean distance* terlihat pada persamaan (2.13).

$$d(x_i, x_k) = \sqrt{\sum_{j=1}^D (x_i^{(j)} - x_k^{(j)})^2} \quad (2.13)$$

### 2.2.6 Evaluasi Model

Dalam mengevaluasi kinerja model klasifikasi yang dibangun pada penelitian ini, maka perlu dilakukan perhitungan untuk mengetahui tingkat keberhasilan model dalam melakukan klasifikasi gender. Beberapa indikator yang dipakai untuk menguji keberhasilan model yang dibangun pada penelitian ini adalah metrik akurasi, presisi, *recall*, dan *F1-score*.

#### 1 Akurasi

Akurasi adalah tingkat keberhasilan model dalam mengklasifikasikan kelas dengan benar dari keseluruhan data Lubanraja, dkk. (2021). Model yang baik dapat dinilai dari tingginya hasil akurasi yang dihasilkan Lubanraja, dkk. (2021). Pada penelitian ini berarti akurasi menjelaskan bagaimana model berhasil memprediksi sidik jari pria dan wanita dari keseluruhan data. Misalnya, jika model akan memprediksi 60 sidik jari pria dan 60 sidik jari wanita, dan ternyata model dapat memprediksi 55 sidik jari pria dan 35 sidik jari wanita dengan benar, maka akurasi model adalah 75%. Rumus yang dipakai untuk menghitung nilai akurasi terlihat pada persamaan (2.14).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.14)$$

#### 2 Presisi

Presisi adalah ukuran keberhasilan model dalam mengidentifikasi dengan benar kasus positif dari total kasus yang diklasifikasikan sebagai positif Lubanraja, dkk. (2021). Presisi dapat memberikan gambaran seberapa akurat model dalam

mengklasifikasikan label dari kelas tersebut. Pada penelitian ini berarti presisi menjelaskan bagaimana model berhasil memprediksi sidik jari pria atau wanita dengan benar dari keseluruhan data pada kelas pria atau wanita. Misalnya, jika model akan memprediksi 60 sidik jari pria dan ternyata model dapat memprediksi 55 sidik jari pria dengan benar, maka presisi model adalah 91.67%. Rumus yang dipakai untuk menghitung nilai presisi dapat dilihat pada persamaan (2.15).

$$Presisi = \frac{TP}{TP + FP} \quad (2.15)$$

### 3 Recall

Recall adalah ukuran keberhasilan model dalam mengidentifikasi semua kasus positif yang sebenarnya dari total kasus positif dalam data Lubanraja, dkk. (2021). Recall dapat memberikan gambaran seberapa sensitif model dalam menghindari kesalahan klasifikasi pada kelas tertentu. Pada penelitian ini berarti recal menjelaskan bagaimana model dapat memprediksi sidik jari pria atau wanita dengan benar dari keseluruhan data asli pada kelas pria atau wanita. Misalnya jika dari 100 data sidik jari, sebenarnya ada 70 sidik jari laki-laki dan model berhasil mengidentifikasi 55 sidik jari dengan benar, maka recall untuk klasifikasi laki-laki adalah 78,57%. Rumus yang dipakai untuk menghitung nilai recal dapat dilihat pada persamaan (2.16).

$$Recall = \frac{TP}{TP + FN} \quad (2.16)$$

### 4 F-1 Score

*F-1 Score* adalah nilai yang menunjukkan nilai keseimbangan antara nilai presisi dan nilai *recall*. Nilai yang dihasilkan oleh *F-1 score* akan menunjukkan nilai rata-rata harmonik dari presisi dan *recall*, yang akan memiliki nilai dalam rentang 0 (nilai terburuk) hingga 1 (nilai terbaik) Lubanraja, dkk. (2021). Rumus yang dipakai untuk menghitung nilai F-1 score dapat dilihat pada persamaan (2.17).

$$F1 = 2 \times \frac{Presisi \times Recall}{Presisi + R} \quad (2.18)$$

Keterangan:

- Nilai positif pada penelitian ini berarti adalah gender pria dan nilai negatif pada penelitian ini berarti adalah gender wanita
- TP (*true positive*): *True positive* adalah nilai yang diperoleh ketika model klasifikasi secara benar memprediksi bahwa suatu data termasuk dalam suatu kelas atau label dan prediksi tersebut sesuai dengan kelas atau label sebenarnya dari data tersebut. Misalnya dalam penelitian ini, model dengan benar memprediksi gambar sidik jari sebagai pria dan pada kenyataannya gambar tersebut adalah sidik jari pria.
- TN (*true negative*): *True negative* adalah nilai yang diperoleh ketika model klasifikasi secara benar memprediksi bahwa suatu data bukan termasuk dalam suatu kelas atau label dan prediksi tersebut. Misalnya dalam penelitian ini, model dengan benar memprediksi gambar sidik jari perempuan sebagai wanita dan pada kenyataannya gambar tersebut adalah sidik jari wanita.
- FP (*false positive*): *False positive* adalah nilai yang diperoleh ketika model klasifikasi salah memprediksi nilai positif. Misalnya dalam penelitian ini, model memprediksi suatu gambar sidik jari sebagai pria, yang pada kenyataannya gambar sidik jari tersebut merupakan wanita.
- FN (*false negative*): *False negative* adalah nilai yang diperoleh ketika model klasifikasi salah memprediksi nilai negatif. Misalnya dalam penelitian ini, model memprediksi suatu gambar sidik jari sebagai wanita, yang pada kenyataannya gambar sidik jari tersebut merupakan pria.
- 

### **BAB III**

### **METODOLOGI PENELITIAN**

### 3.1 Objek Penelitian

Penelitian ini akan bertujuan untuk membangun sebuah model machine learning yang dapat melakukan klasifikasi gender berdasarkan sidik jari. Data sidik jari yang digunakan berasal dari dataset yang disediakan oleh *Sokoto Coventry Fingerprint*. Dataset ini terdiri 6000 citra sidik jari yang berasal dari 600 orang Afrika. Terdapat 10 citra sidik jari yang diambil pada setiap subjek orang Afrika dan semua subjek berusia minimal 18 tahun. Penamaan file pada citra sidik jari yang diambil memiliki nama unik seperti label untuk gender, tangan, dan nama jari. Sidik jari diambil menggunakan Hamster plus (HSDU03PTM) dan pemindai sensor SecuGen SDU03PTM. Semua citra sidik jari memiliki resolusi 1 x 96 x 103 (abu-abu x lebar x tinggi).

### 3.2 Analisis Kebutuhan Sistem

#### 3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional dari sistem model klasifikasi sidik jari yang dibangun adalah sebagai berikut:

- Dapat menerima input berupa citra sidik jari
- Dapat melakukan ekstraksi fitur dari citra sidik jari
- Dapat melakukan klasifikasi gender berdasarkan fitur yang diekstrak

#### 3.2.2 Kebutuhan non-Fungsional

Kebutuhan non-fungsional dari sistem model klasifikasi sidik jari yang dibangun adalah sebagai berikut:

- Akurasi klasifikasi gender diatas 80%
- Model dapat mengklasifikasi citra sidik jari yang bagus secara baik

#### 3.2.3 Perangkat Keras

Untuk membangun model klasifikasi sidik jari, penulis akan menggunakan perangkat keras sebagai berikut:

- Laptop : Asus TUF A15



- Processor : Ryzen 7 4800H
- RAM : 16 GB
- GPU : Nvidia GeForce GTX 1650Ti

#### **3.2.4 Perangkat Lunak**

Untuk membangun model klasifikasi sidik jari, penulis akan menggunakan perangkat lunak sebagai berikut:

- Synder untuk membuat laporan yang terstruktur, menjalankan dan memantau eksperimen
- Python sebagai bahasa pemograman utama

#### **3.2.5 Library**

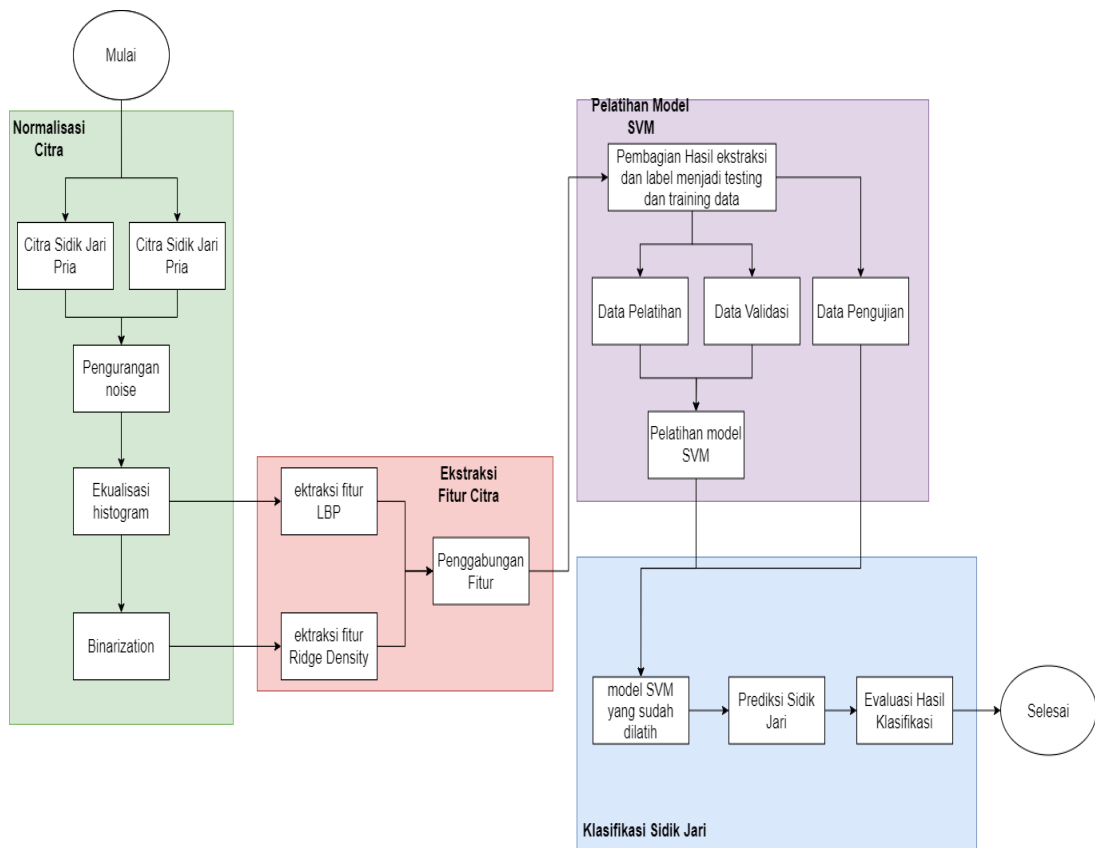
Untuk membangun model klasifikasi sidik jari, penulis juga akan menggunakan beberapa library pendukung untuk membangun model klasifikasi sidik jari sebagai berikut:

- Scikit-learn
- OpenCV
- Numpy
- Matplotlib
- Image-enhancer
- PIL
- Seabron

### **3.3 Perancangan Penelitian**

#### **3.3.1 Diagram sistem pembuatan model**

Diagram sistem pada penelitian ini akan terdiri dari 4 tahapan seperti yang dapat dilihat pada Gambar 3.1.



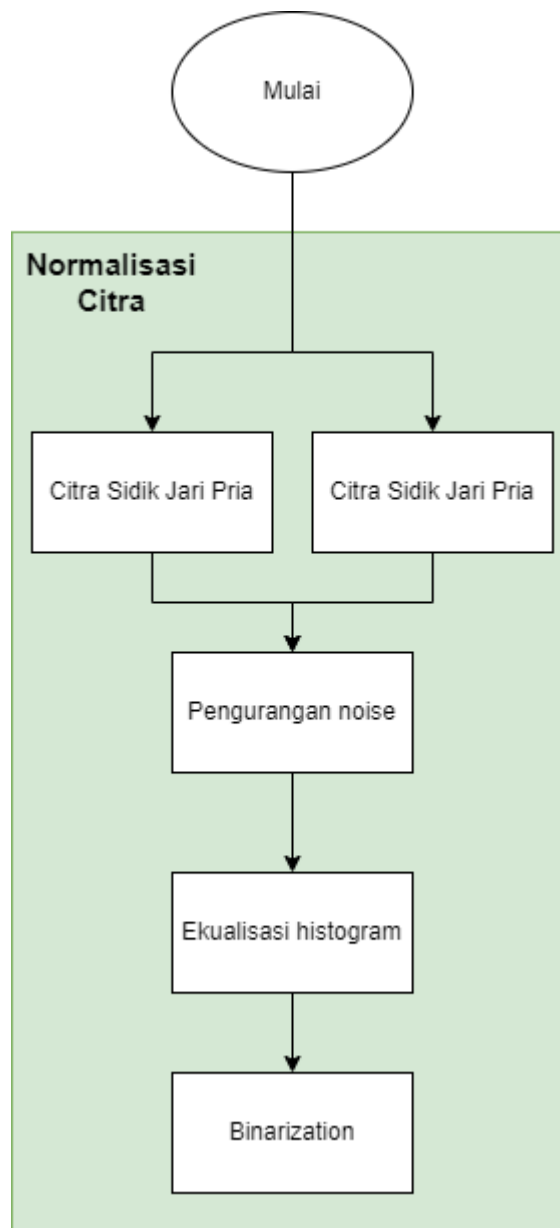
Gambar 3. 1 Blok diagram sistem

Blok diagram yang ditunjukkan pada Gambar 3.1, terbagi menjadi 4 tahapan yang dapat dijelaskan sebagai berikut:

#### 1. Normalisasi Citra

Tahapan normalisasi citra merupakan tahapan yang digunakan untuk meningkatkan kualitas citra sidik jari, agar nantinya fitur yang diekstraksi memiliki hasil yang baik. Untuk menghasilkan ekstraksi fitur LBP akan melalui dua tahapan normalisasi yaitu pengurangan noise dan ekualisasi histogram yang akan digunakan untuk meningkatkan kontras pada citra sidik jari. Kedua tahapan ini akan digunakan untuk mendapatkan fitur LBP yang berfokus pada tekstur dalam citra. Proses ekstraksi fitur LBP tidak melalui tahapan binarisasi, thinning, dan skeletisasi, dikarenakan ketiga tahapan ini digunakan untuk mendapatkan morfologi citra yang berfokus pada struktur geometris objek dalam citra, yang nantinya akan digunakan untuk

mendapatkan fitur ridge density. Proses ekstraksi fitur ridge density membutuhkan citra sidik jari yang telah dibinasisasi, dithining, dan diskeletonisasi. Proses-proses ini bertujuan untuk memperkecil ukuran citra sidik jari dan menghilangkan informasi yang tidak relevan. Informasi yang tidak relevan dapat mengganggu proses ekstraksi fitur ridge density. Tahapan normalisasi citra secara berturut dapat dilihat pada Gambar 3.2. Tahapan akan dimulai dengan menerapkan pengurangan noise pada citra sidik jari pria dan wanita, kemudian setelah itu maka akan ekualisasi histogram pada citra yang sudah diproses sebelumnya, data citra yang sudah diproses ini kemudian akan di di proses terlebih dahulu untuk mengekstraksi fitur LBP, ketika fitur LBP sudah didapatkan, maka citra kemudian akan dilanjutkan untuk diproses menjadi citra biner yang akan digunakan untuk proses ekstraksi fitur *ridge density*.

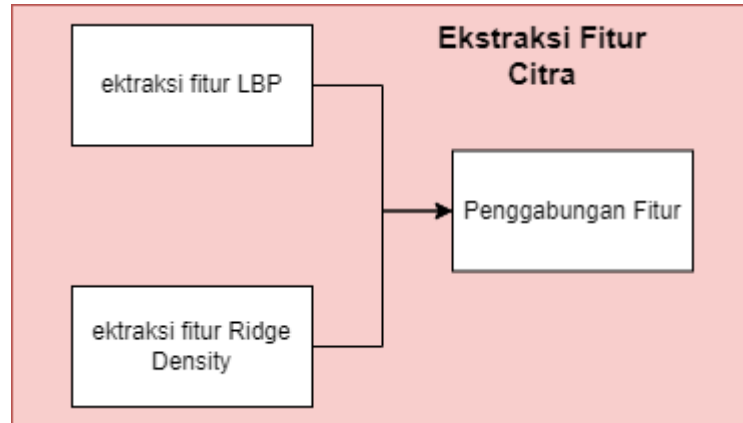


Gambar 3.2 Diagram alir proses normalisasi citra

## 2. Ekstraksi Fitur Citra

Setelah proses normalisasi citra, setiap citra sidik jari akan diekstraksi fiturnya dengan metode LBP untuk mendapatkan fitur tekstur dan metode ridge density untuk mendapatkan informasi kepadatan ridge. Kedua fitur tersebut kemudian digabungkan menjadi satu untuk merepresentasikan satu fitur yang terdapat pada citra sidik jari. Tahapan ekstraksi fitur citra secara berturut dapat dilihat pada

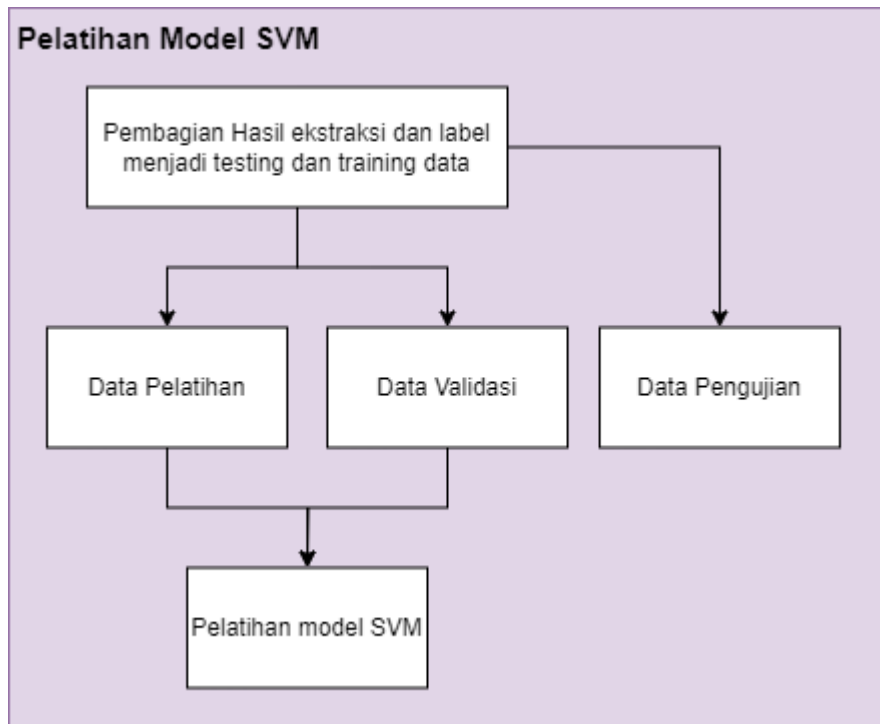
Gambar 3.3. Proses ini akan menghasilkan fitur vektor yang terdiri dari penggabungan hasil ekstraksi fitur LBP dan *ridge density*.



Gambar 3.3 Diagram alir proses ekstraksi fitur citra

### 3. Pelatihan Model SVM

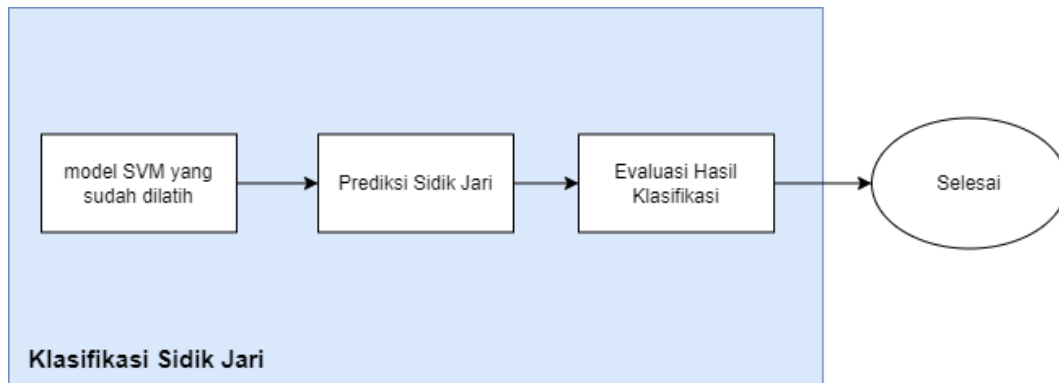
Pelatihan model akan dimulai dengan melakukan pembagian data citra sidik jari menjadi data training yang akan digunakan untuk melakukan pelatihan model klasifikasi dan data testing yang akan digunakan untuk mengevaluasi kinerja model klasifikasi. Data training akan dibagi menjadi data pelatihan dan data validasi. Data pelatihan ini akan digunakan untuk melatih model dalam mengenali pola-pola yang terdapat pada citra sidik jari untuk dapat melakukan klasifikasi pada data baru dan data validasi akan digunakan untuk menguji kinerja model selama proses pelatihan. Sedangkan untuk data testing akan digunakan sebagai proses untuk mengevaluasi kinerja model yang sudah dilatih, dengan melakukan percobaan menggunakan data pengujian. Tahapan pelatihan model SVM secara berturut dapat dilihat pada Gambar 3.4. Proses pelatihan model SVM akan dimulai dengan membagikan dataset yang sudah memiliki fitur vektor pada tiap data beserta dengan label yang sudah diberikan, pembagian ini akan menghasilkan data pelatihan, data validasi, dan data pengujian. Data pelatihan dan data validasi nantinya akan diterapkan teknik *k-fold validation* untuk melatih model SVM, dan data pengujian akan digunakan untuk menguji model yang sudah dilatih.



Gambar 3.4 Diagram alir proses pelatihan model machine learning

#### 4. Klasifikasi Sidik Jari

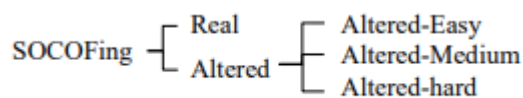
Pada tahap klasifikasi sidik jari, model SVM yang sudah dilatih akan digunakan untuk memprediksi gender pada citra sidik jari. Prediksi dilakukan dengan dua kategori kelas, yaitu pria dan wanita. Hasil prediksi kemudian akan dievaluasi berdasarkan label sebenarnya pada citra sidik jari yang terdapat pada data pengujian. Tahapan pelatihan model SVM secara berturut dapat dilihat pada Gambar 3.5. Pada proses ini model yang sudah dilatih akan diujikan dengan data pengujian yang belum memiliki label namun sudah memiliki fitur vektor, model kemudian akan memprediksi kelas dari tiap data tersebut, yang akan menghasilkan keluaran kelas pria atau wanita. Setelah proses klasifikasi pada model yang sudah dilatih selesai, maka selanjutnya akan dilakukan evaluasi kinerja model untuk menilai seberapa optimal model yang dibuat dengan membandingkan label sebenarnya dari tiap data pengujian dengan label yang diprediksi oleh model.



Gambar 3. 5 Diagram alir proses klasifikasi sidik jari

### 3.3.2 Tahap Pengumpulan Data

Data citra sidik jari yang digunakan dalam penelitian ini adalah data yang telah diperoleh dari database sidik jari Sokoto Conventry Fingerprint Dataset (SOCOFing) yang tersedia pada platform kaggle. SOCOFing sendiri merupakan database yang khusus disediakan untuk penelitian sidik jari. SOCOFing berisikan 6000 sidik jari dari 600 warga Afrika dan sudah diberi label gender, tangan kiri atau tangan kanan, dan nama jari. Pada dataset ini terdapat 10 sidik jari pada setiap subjek dan diambil dari warga yang berusia 18 tahun ke atas. Dataset ini terbagi menjadi dua folder yaitu folder real dan altered. Pada folder real terbagi menjadi 1 subfolder yaitu folder yang menyimpan gambar sidik jari asli (*original images*) yang berisikan sidik jari dari warga Afrika yang tidak mengalami manipulasi gambar. Pada folder altered terbagi menjadi 3 subfolder, dimana subfolder tersebut adalah altered-easy, altered-medium, dan altered-hard, yang menandakan tingkat kesusahan yang berbeda dalam manipulasi gambar. Hierarki folder pada dataset ini dapat dilihat pada Gambar 3.6.



Gambar 3. 6 Struktur Folder SOCOFing.

Sumber Shehu, dkk (2022)

Pada penelitian ini akan menggunakan citra sidik jari asli pada tangan kanan dan tangan kiri di setiap subjek untuk melakukan klasifikasi gender. Penelitian ini akan menggunakan 70% dari dataset untuk digunakan sebagai data pelatihan untuk pelatihan model, yang nantinya akan dibagi lagi menggunakan k-fold validation yang akan membuat setiap iterasi pada fold memiliki jumlah data pelatihan 89-90% dan data validasi 9-10% dan 30% untuk data pengujian untuk pengujian performa model.

Data yang digunakan untuk membangun modal akan menerapkan 3 kemungkinan:

Percobaan Ke-	Jumlah citra laki-laki	Jumlah citra wanita	Folder	Data pelatihan	Data validasi	Data pengujian
1	4770	1230	Semua di ambil pada folder <i>real</i>	4200	420	1800
2	1230	1230	Semua di ambil pada folder <i>real</i>	1549 - 1550	172 - 173	738
3	4770	4770	Citra laki-laki di ambil pada folder <i>real</i> , 1230 citra wanita di ambil dari folder <i>real</i> , dan 3540 citra wanita di ambil dari folder <i>altered easy</i>	6010 - 6011	667 - 668	2862

**Tabel 3. 1 Tabel percobaan sebaran nilai yang akan digunakan pada penelitian ini**





**Gambar 3.7** Contoh citra sidik jari pada folder *Real*



**Gambar 3.8** Contoh citra sidik jari pada folder *Altered Easy*

### 3.3.3 Tahap Implementasi Model

Tahap-tahap implementasi model secara berurut dapat dijelaskan sebagai berikut:

1. Data citra sidik jari yang digunakan akan dilakukan *pre-processing* sebelum dilakukan ekstraksi fitur, proses ini akan dilakukan menggunakan bahasa pemrograman Python dengan memanfaatkan *library* OpenCV dan scikit-image. Proses *pre-processing* akan dimulai dengan dilakukan *scaling image* pada tiap citra menggunakan *library* cv2, kemudian menghapus noise pada citra menggunakan *mean filter*, kemudian meningkatkan kontras pada citra sidik jari menggunakan histogram equalization, dan terakhir citra sidik jari abu-abu akan diproses kedalam citra biner dengan metode otsu.
2. Setelah dilakukan *pre-processing* data, maka akan dilakukan ekstraksi fitur untuk mendapatkan nilai LBP dan *Ridge Density*. *Library* yang akan digunakan untuk melakukan ekstraksi fitur citra dengan metode LBP dan *ridge density* adalah scikit-image. Scikit-image telah menyediakan fungsi yang dapat digunakan langsung untuk menghitung nilai LBP dan *ridge density* dari gambar. Hasil ekstraksi fitur yang dihasilkan akan disimpan kedalam dimensi *array* menggunakan *library* Numpy. Array tersebut nantinya akan berisikan hasil sebaran frekuensi histogram yang didapatkan menggunakan metode LBP pada setiap sidik jari tiap individu dan nilai dari *ridge density* pada tiap individu. Setelah fitur ekstraksi telah diekstrak dan disimpan, data hasil ekstraksi LBP

dan *ridge density* akan digabungkan menjadi vektor fitur. Sehingga setiap baris pada data array akan berisikan 20 kolom yang terbagi menjadi 10 jari kanan dan kiri yang berisikan nilai histogram LBP dan 10 jari kanan dan kiri yang berisikan nilai *ridge density*.

3. Fitur vektor ini kemudian akan digunakan untuk melatih model klasifikasi. Library *scikit-learn* akan digunakan untuk menerapkan *Support Vector Machine* (SVM) sebagai classifier pada penelitian ini. Setelah model SVM dilatih dengan vektor fitur, model tersebut dapat digunakan untuk memprediksi gender berdasarkan data pengujian.

#### **3.3.4 Tahap Evaluasi Model**

Model klasifikasi yang telah dibuat akan dilakukan evaluasi, untuk mengukur tingkat keberhasilan klasifikasi pada model SVM. Evaluasi model dilakukan dengan melihat nilai yang didapatkan pada perhitungan akurasi, presisi, *recall*, dan *F1-Score*.

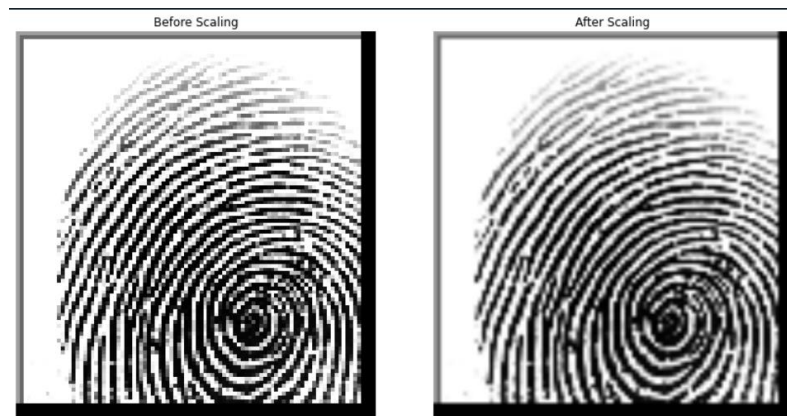
## BAB IV

### IMPLEMENTASI DAN ANALISIS SISTEM

#### 4.1 Implementasi sistem

##### 4.2.1 Preprocessing Citra

Preprocessing citra adalah tahap awal dalam analisis sidik jari yang bertujuan untuk mempersiapkan citra untuk proses analisis selanjutnya. Pada tahap ini akan dimulai dengan memastikan jika semua citra yang akan diproses dalam bentuk grayscale, selanjutnya akan dilakukan *resized image* dengan melakukan *scaling* sebesar 5 kali pada tiap citra sidik jari yang akan diproses. Ukuran awal citra adalah 96 x 103 pixels sehingga setelah dilakukan *resized image* akan menjadi 480 x 515 pixels, ukuran tersebut dapat di lihat pada Gambar 4. 2. Hasil citra yang sudah di *resized* dapat di lihat pada Gambar 4. 1, gambar sebelah kiri menunjukkan citra sidik jari sebelum di lakukan perubahan ukuran pixel dan gambar sebelah kanan sudah melalui perubahan ukuran pixel.



**Gambar 4. 1** Perbandingan citra sebelum dan sesudah *scaling*

<code>image_after_scaling</code>	Array of uint8	(515, 480, 4)	<code>[[[160 160 160 255]</code> <code>[160 160 160 255]</code>
<code>image_before_scaling</code>	Array of uint8	(103, 96, 4)	<code>[[[160 160 160 255]</code> <code>[160 160 160 255]</code>

**Gambar 4. 2** Nilai pixel citra sebelum dan sesudah *scaling*

Implementasi preprocessing selanjutnya akan terbagi menjadi dua tahap yang berbeda

#### 1. Preprocessing untuk ekstraksi *Ridge Density*

Setelah melakukan resized image citra tahap selanjutnya adalah mengubah citra sidik jari hitam putih menjadi citra sidik jari biner untuk melakukan ekstraksi fitur ridge density, hasil citra yang telah melalui tahap binarisasi dapat dilihat pada Gambar 4.3. Setelah citra sudah melewati tahap binerisasi, maka citra akan dihaluskan menggunakan library *image\_enhancer* yang menerapkan algoritma *gabor filter*, hasil citra sidik jari yang telah melalui tahap ini dapat dilihat pada Gambar 4.4. Setelah citra dihaluskan, maka tahap terakhir adalah melakukan skeletonisasi pada citra untuk menghasilkan citra dengan garis ridge yang sangat tipis, hasil yang telah melalui tahap ini dapat dilihat pada Gambar 4.5. Codingan untuk melakukan pre-processing pada tahap ini dapat dilihat pada Code 4.1

```
# Apply local thresholding
local_binary = filters.threshold_local(resized_img,
block_size=55, method='gaussian')
# Binarize the image using local threshold
binarized_img = resized_img > local_binary

# Enhance image
out = fingerprint_enhancer.enhance_Fingerprint(binarized_img)
out[out == 255] = 1
skeleton = skeletonize(out)
```

**Code 4. 1** codingan untuk pre-procesing ekstraksi *ridge density*

Image After Binarization



**Gambar 4. 3** Citra sidik jari sesudah binarisasi

Image After Thining



**Gambar 4. 4** Citra sidik jari sesudah di haluskan

Image After Skeletonize



**Gambar 4. 5** Citra sidik jari setelah skeletonisasi

2. ***Pre-processing untuk ekstraksi Local Binary Pattern(LBP)***

Setelah melakukan resized image citra yang terlihat pada Gambar 4.6 tahapan pada *pre-processing* ini hanya akan melakukan dua tahapan *pre-processing* yaitu dengan melalui pengurangan noise menggunakan *Gaussian filter*, hasil dari citra yang telah melalui tahapan pengurangan *noise* dapat di lihat pada Gambar 4.7 dan melakukan penyesuaian kontras menggunakan *histogram equalization* hasil citra yang telah melalui tahap ini dapat di lihat pada Gambar 4.8. *Codingan* yang di gunakan untuk melakukan *pre-processing* pengurangan noise dan penyesuaian kontras dapat di lihat pada Code 4.2

```
# Apply noise reduction using Gaussian filter
blurred_img = filters.gaussian(resized_img, sigma=1)

# Apply histogram equalization for contrast enhancement
equalized_img = exposure.equalize_hist(blurred_img)
skeleton = skeletonize(out)
```

**Code 4. 2** Code *Pre-processing* untuk ekstraksi *Local Binary Pattern*



**Gambar 4. 6** Citra sidik jari sebelum di kurangi noise



**Gambar 4. 7** Citra sidik jari setelah di kurangi noise



**Gambar 4. 8** Citra sidik jari setelah penerapa histogram equalization

#### **4.2.2 Ekstraksi Fitur Citra**

Ekstraksi fitur merupakan tahap krusial dalam analisis citra sidik jari di mana informasi paling relevan diekstraksi dari citra tersebut. Dua metode utama yang digunakan dalam ekstraksi fitur adalah Local Binary Pattern (LBP) dan Ridge Density. Kedua metode ini memiliki kelebihan dan karakteristik khusus dalam menangkap pola sidik jari. Setelah fitur-fitur ini diekstraksi, mereka digabungkan untuk meningkatkan representasi sidik jari secara keseluruhan, sehingga memungkinkan analisis yang lebih akurat dan efisien.

##### **4.1.2.1 Ekstraksi Fitur LBP**

Pada tahap ini, fitur-fitur penting dari citra sidik jari diekstraksi menggunakan metode Local Binary Pattern (LBP). Nilai citra sebelum melalui tahapan LBP akan dapat di lihat pada persebaran histogram yang di tunjukkan pada Gambar 4.9. Implementasi LBP pada penelitian ini akan menggunakan radius 1 dan



nilai tetangga 8. Implementasi untuk melakukan ekstraksi fitur LBP menggunakan salah satu fungsi pada library *scikit-image* yaitu *LocalBinaryPatterns*, implementasi tersebut dapat di lihat pada Code 4.3 di mana hasil perubahan gambar yang telah melalui tahap ini dapat di lihat pada Gambar 4.10, hasil perbedaan histogram sebelum melalui perhitungan nilai LBP dapat di lihat pada Gambar 4.11 dan histogram setelah melalui perhitungan nilai LBP dapat di lihat pada Gambar 4.12. *Output* yang dihasilkan pada fungsi *LocalBinaryPatterns* akan menghasilkan *numpy array* yang berisikan keseluruhan piksel yang telah melalui proses perhitungan fungsi ini, sehingga dimensi fitur yang ada dapat sangat tinggi yang dapat meningkatkan waktu komputasi model, sehingga perlu adanya normalisasi nilai pada keseluruhan pixel. Hasil persebaran histogram yang sudah melalui proses normalisasi dapat dilihat dari Gambar 4.13. Hasil dari normalisasi tersebut akan menghasilkan *numpy array* berdimensi 1x1, yang dapat di lihat pada Gambar 4.14. Untuk mengecilkan waktu komputasi akan dilakukan normalisasi piksel yang bertujuan agar pengenalan pola lainnya dapat menjadi lebih stabil dan lebih mudah dikonvergensi.

```
#Function to store LBP feature
def store_LBP_features(arr_file):
    array_lbp = []

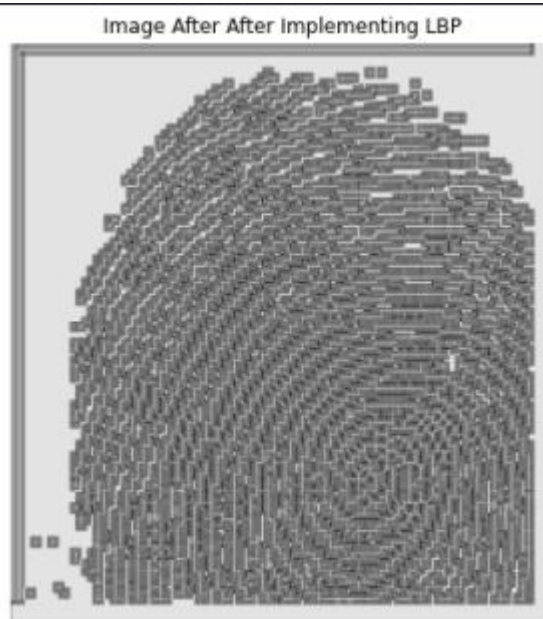
    for i in range(0,len(arr_file)):
        desc = LocalBinaryPatterns(8,1)
        histogram = desc.describe(arr_file[i])
        array_lbp.append(histogram)

    return array_lbp
```

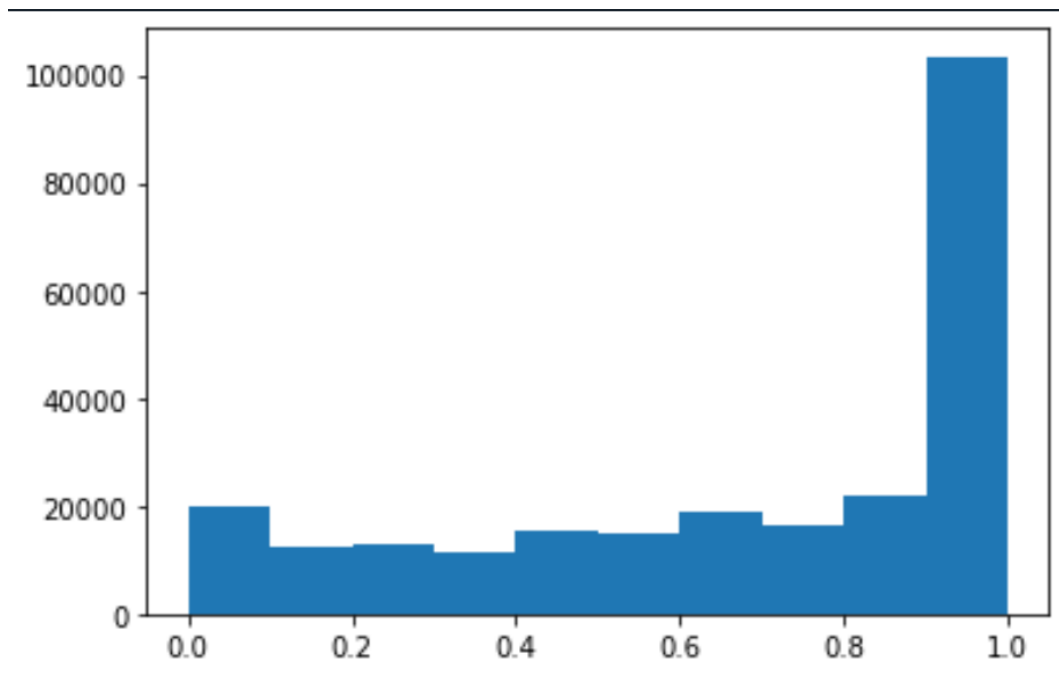
**Code 4. 3 Implementasi ekstraksi LBP**



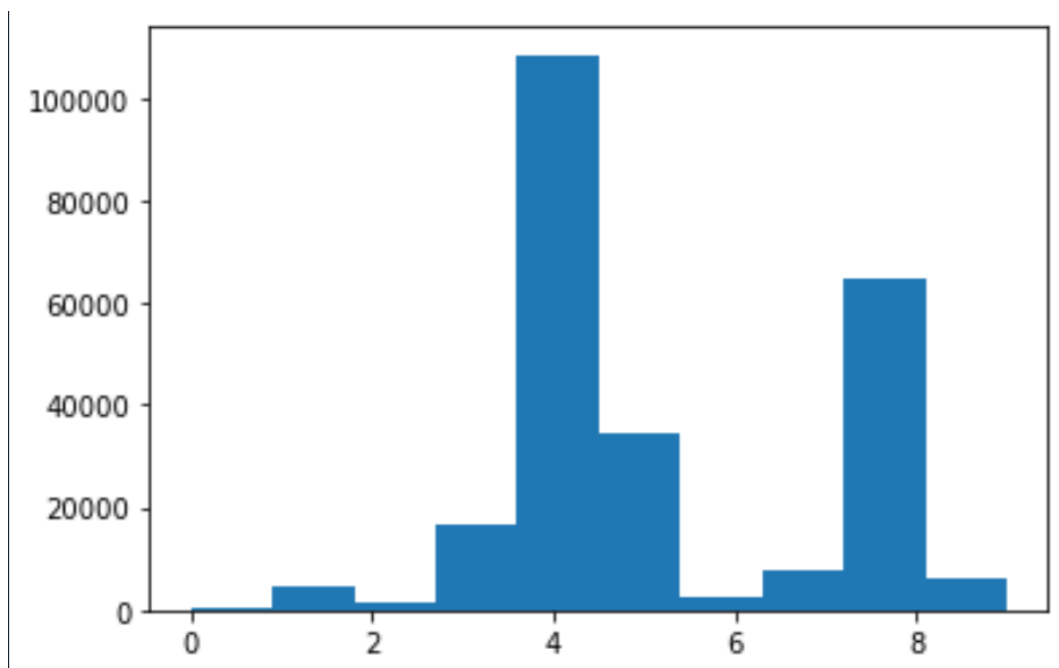
**Gambar 4. 9** Citra sidik jari sebelum implementasi LBP



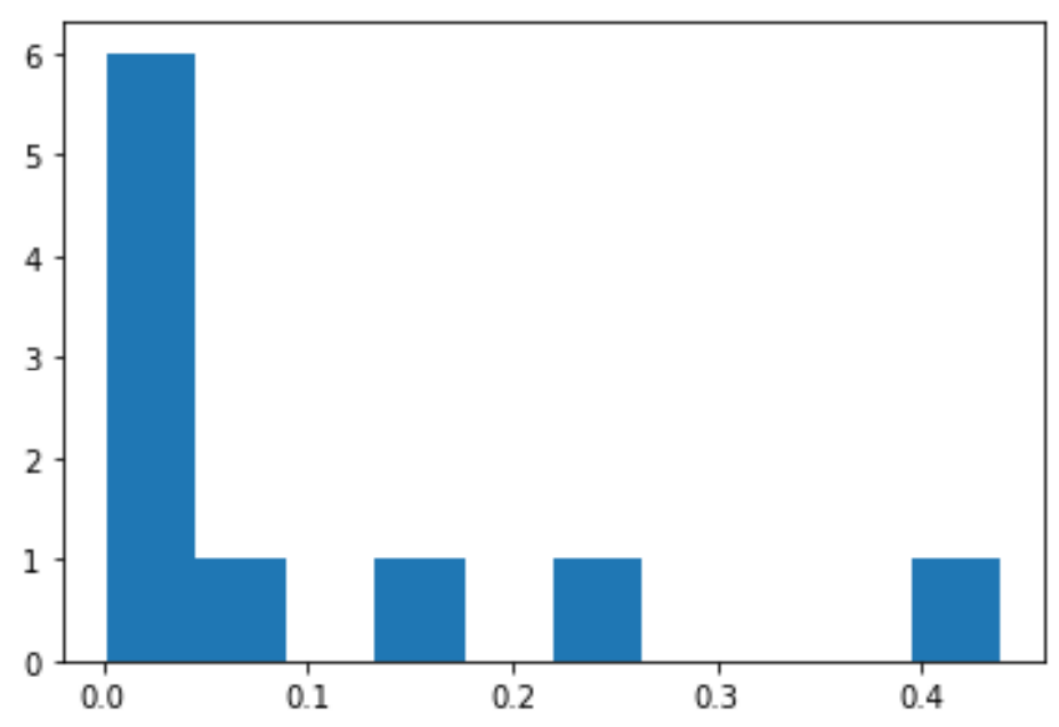
**Gambar 4. 10** Citra sidik jari setelah implementasi LBP



**Gambar 4. 11** Histogram citra original sebelum implementasi LBP



**Gambar 4. 12** Histogram citra original sesudah implementasi LBP



**Gambar 4.13** Histogram citra original sesudah implementasi LBP dan di normalisasi

	0
0	0.00150081
1	0.0192314
2	0.00688511
3	0.0676861
4	0.438358
5	0.139057
6	0.00967637
7	0.0312904
8	0.261655
9	0.0246602

**Gambar 4.14** Fitur citra LBP yang sudah di normalisasi

#### 4.1.2.1 Ekstraksi Fitur Ridge Density

Selain ekstraksi fitur menggunakan LBP, ridge density juga diekstraksi sebagai fitur tambahan pada penelitian ini. Ridge density menggambarkan kepadatan ridge yang terdapat dalam citra sidik jari, yang merupakan karakteristik penting dalam mengidentifikasi dan membedakan sidik jari antar individu. Berikut adalah kode ekstraksi fitur ridge density. Penghitungan *Ridge Density* pada penelitian ini diterapkan dengan menghitung pixel hitam pada keseluruhan citra sidik jari yang kemudian akan di bagi dengan luas piksel citra. Hal ini dilakukan dengan memisahkan foreground dan background pada citra sidik jari, yang mana foreground akan mendeskripsikan ridge atau garis pada sidik jari dan background akan mendeskripsikan area di sekitarnya. Dengan cara ini, penghitungan ridge density (kerapatan ridge) dapat memberikan informasi tentang seberapa padat atau jarang garis-garis (ridge) pada sidik jari tersebut dibandingkan dengan luas total area citra. Semakin tinggi nilai ridge density, semakin padat dan kompleks pola sidik jari tersebut.

```
#function to store ridge density features
def store_ridge_density_features(arr_file):
    array_ridge_density = []
    for i in range(0, len(arr_file)):
        convert_to_int = (1 - arr_file[i]).astype(np.uint8) * 255
        area = np.prod(arr_file[i].shape)
        ridge_count = sum(255 for row in convert_to_int for pixel in row if
pixel == 255)
        ridge_density = RidgeDensityCalculator(ridge_count, area)
        count_ridge_density = ridge_density.calculate_ridge_density()
        array_ridge_density.append(count_ridge_density)
    return array_ridge_density
```

Kode tersebut mengimplementasikan perhitungan kepadatan ridge dari citra sidik jari. Pertama, kelas `RidgeDensityCalculator` didefinisikan untuk menghitung kepadatan ridge berdasarkan jumlah ridge dan luas citra. Kemudian, fungsi `store_ridge_density_features` digunakan untuk menghitung kepadatan ridge dari setiap citra dalam `arr_file`. Setiap citra diubah menjadi citra biner, dan jumlah piksel putih (yang mewakili ridge) dihitung untuk mendapatkan jumlah ridge. Luas citra dihitung sebagai produk dari panjang dan lebar citra. Selanjutnya, kepadatan ridge dihitung menggunakan objek `RidgeDensityCalculator` dan disimpan dalam

`array\_ridge\_density`. Hasil sidik jari yang telah selesai pada tahap *pre-processing* ini, dapat di lihat pada Gambar 4.15



**Gambar 4. 15** Contoh gambar citra untuk ekstraksi fitur *ridge density*

#### **4.1.2.3 Ekstraksi fitur pada file sidik jari**

Langkah ini mencakup pengumpulan informasi penting yang terkandung dalam penamaan gambar citra sidik jari. Fitur ini merupakan fitur tambahan yang digunakan dalam proses pembuatan model. Informasi yang dihasilkan seperti lokasi tangan sidik jari yaitu sidik jari kanan atau sidik jari kiri dan jenis sidik jari yang mencakup jari jempol, jari telunjuk, jari tengah, jari manis, jari kelingking.

```

def store_hand(arr_file):
    temp_array = []
    for i in arr_file:
        path = os.path.basename(i)
        file = path.split("_")
        array = []
        location = file[4]

        #array.append(file[0])

        if location == "thumb":
            array.append(0)
            if file[3] == "Left":
                array.append(0)
            elif file[3] == "Right":
                array.append(1)
        elif location == "middle":
            array.append(1)
            if file[3] == "Left":
                array.append(0)
            elif file[3] == "Right":
                array.append(1)
        elif location == "ring":
            array.append(2)
            if file[3] == "Left":
                array.append(0)
            elif file[3] == "Right":
                array.append(1)
        elif location == "index":
            array.append(3)
            if file[3] == "Left":
                array.append(0)
            elif file[3] == "Right":
                array.append(1)
        else:
            array.append(4)
            if file[3] == "Left":
                array.append(0)
            elif file[3] == "Right":
                array.append(1)
        temp_array.append(array)
    return temp_array
arr_hand_location = store_hand(image_files)

```

Fungsi `store_hand` menerima sebuah array `arr_file` yang berisi path file dari citra sidik jari. Selanjutnya, setiap path file tersebut diproses untuk mengambil informasi tentang lokasi dan sisi tangan dari citra tersebut. Informasi ini kemudian dimasukkan ke dalam sebuah array sementara yang disebut `temp_array`. Setiap elemen dalam `temp_array` terdiri dari nomor identifikasi citra, lokasi tangan (dalam bentuk angka), dan sisi tangan (0 untuk tangan kiri, 1 untuk tangan kanan), yang merepresentasikan informasi penting yang diperoleh dari nama file citra.

Hasil akhir dari pemrosesan ini adalah `arr_hand_location`, sebuah array yang berisi array-array dengan informasi lokasi dan sisi tangan dari setiap citra sidik jari yang terdapat dalam `image_files`.

### 4.2.3 Penggabungan Fitur

Fitur - fitur *local binary pattern (LBP)* dan *ridge density* yang sudah diekstraksi dari citra sidik jari akan dilakukan *feature fusion*. Penggabungan ini bertujuan untuk memperkaya representasi fitur dari citra sidik jari secara keseluruhan, dengan memanfaatkan kelebihan dari masing-masing metode ekstraksi fitur untuk meningkatkan kemampuan sistem dalam mengenali pola-pola sidik jari dengan lebih akurat. Penelitian ini melakukan 4 percobaan penggabungan fitur untuk dapat meninjau sejauh mana setiap fitur mempengaruhi performa model. Percobaan pertama akan menggabungkan fitur *LBP* dan *ridge density* menjadi satu, percobaan kedua akan menggabungkan fitur *LBP*, *ridge density* dan fitur pada file citra sidik jari yang akan menghasilkan lokasi jari, percobaan ketiga akan menggabungkan fitur *LBP* dan fitur pada file citra sidik jari, dan percobaan keempat akan menggabungkan fitur *ridge density* dan fitur pada file citra sidik jari.

Berikut adalah kode penggabungan fitur yang terlihat pada Code 4.4.

```
#training
# Convert lists to numpy arrays
X_lbp = np.array(arr_lbp_features)
X_ridge_density = np.array(arr_ridge_density_features)
X_array_hand_location = np.array(arr_hand_location)
y = np.array(labels)

vector_features = []

for i in range(0, len(X_lbp)):
    temp_array = np.concatenate((X_lbp[i], [X_ridge_density[i]]))
    #temp_array = np.concatenate((X_lbp[i],
    [X_ridge_density[i]], X_array_hand_location[i]))
    #temp_array = np.concatenate((X_lbp[i], X_array_hand_location[i]))
    #temp_array =
    np.concatenate((X_ridge_density[i], X_array_hand_location[i]))

    vector_features.append(temp_array)
```

**Code 4. 4 kode penggabungan fitur**

Dalam kode di atas, fitur-fitur *LBP*, *ridge density*, dan file citra sidik jari digabungkan untuk setiap citra sidik jari secara iteratif. Prosesnya dimulai dengan



mengubah list fitur LBP, *ridge density*, dan lokasi tangan menjadi *array numpy*. Selanjutnya, dilakukan iterasi untuk setiap citra sidik jari. Pada setiap iterasi, fitur-fitur LBP, *ridge density*, dan lokasi tangan untuk citra sidik jari tersebut digabungkan menjadi satu array menggunakan metode konkatenasi. Hasil penggabungan ini kemudian dimasukkan ke dalam list *vector\_features*. Dengan demikian, setiap citra sidik jari akan direpresentasikan oleh satu vektor fitur yang menggabungkan informasi dari ketiga metode ekstraksi fitur tersebut. Hasil dari penggabungan tersebut dapat di lihat dari persebaran array yang di tunjukkan oleh Gambar 4.16

0.00427647	0.0350235	0.0140266	0.105268	0.562704	0.12313	0.0121852	0.025116	0.0998909	0.0183792	21.942	3	0
0.00360752	0.0271563	0.00987662	0.106434	0.511939	0.116363	0.0102115	0.0216921	0.172926	0.0197944	19.1849	4	0
0.004357	0.0361265	0.0144136	0.106687	0.566497	0.119707	0.0140175	0.0316512	0.0862037	0.0203395	18.7026	2	0
0.0035311	0.0314498	0.0152201	0.0948947	0.540799	0.149689	0.0140478	0.0278421	0.0952105	0.0273158	22.3851	0	0
0.00481404	0.0363228	0.0146246	0.100779	0.52847	0.11786	0.0136351	0.0290316	0.138182	0.0162807	22.9643	4	1
0.00263636	0.0312778	0.0138333	0.0804848	0.52599	0.154167	0.0145657	0.026404	0.122121	0.0285202	19.9273	0	1
0.00433553	0.0291579	0.0120592	0.110724	0.499388	0.107533	0.00818421	0.0158197	0.203316	0.0102829	20.863	3	0
0.00412233	0.0367024	0.0139216	0.0937305	0.546977	0.111169	0.0123479	0.0302262	0.134992	0.0157311	22.5511	4	0
0.00447561	0.0367439	0.0111402	0.0993415	0.539476	0.10211	0.0112378	0.0312195	0.148823	0.0154329	21.902	2	0
0.0042464	0.0332708	0.0122389	0.0951032	0.501426	0.0998249	0.00915572	0.0244041	0.208662	0.0115885	19.8371	4	1
0.0044715	0.0345747	0.0103839	0.110205	0.52391	0.0975613	0.00966449	0.0220937	0.175579	0.0112566	21.1124	2	1
0.00325086	0.0326697	0.0122267	0.094683	0.567975	0.112135	0.00831991	0.0173303	0.137002	0.0144879	23.3902	0	1
0.00348101	0.0369102	0.0145972	0.0863119	0.537083	0.126738	0.0149402	0.0270196	0.131392	0.021519	20.8049	4	0
0.00394329	0.033954	0.0116158	0.101113	0.525559	0.102028	0.00890851	0.0186035	0.182841	0.0114339	21.939	2	0
0.00433266	0.0333132	0.0133333	0.0986787	0.513387	0.11338	0.0114085	0.0258618	0.172207	0.0140979	20.1537	4	1
0.00428177	0.023634	0.00979521	0.115805	0.463163	0.109961	0.0068032	0.0131474	0.242022	0.0113871	19.68	1	1
0.00419485	0.027518	0.00873239	0.116407	0.47841	0.0987981	0.00805008	0.012651	0.236413	0.00882629	20.8246	2	1
0.00422456	0.027186	0.0118877	0.113885	0.530344	0.13745	0.00879532	0.0213474	0.125062	0.0198175	22.1024	0	1
0.00375169	0.0370271	0.0134359	0.0996383	0.566197	0.113943	0.0138839	0.0286802	0.106332	0.0162105	20.6629	3	0
0.0035697	0.0330364	0.0175939	0.0770545	0.491358	0.149818	0.0165273	0.0291455	0.154855	0.0270424	16.3507	4	0
0.0050641	0.0394872	0.0152278	0.111111	0.570935	0.111229	0.0175801	0.0349306	0.0796154	0.0147703	20.1848	1	0
0.00340686	0.0393627	0.0146471	0.08475	0.557368	0.123598	0.0179412	0.0357157	0.102475	0.0207353	20.9187	2	0
0.00345552	0.0301205	0.0124994	0.0980159	0.511125	0.112284	0.0108946	0.0254294	0.176606	0.0195693	18.8809	0	0
0.00420491	0.0383607	0.0118007	0.109036	0.565884	0.102894	0.0111888	0.0218891	0.123234	0.0115088	23.1186	3	1
0.00151874	0.016309	0.0177092	0.0789965	0.734691	0.10138	0.0203374	0.0157189	0.00657561	0.00676357	14.8257	4	1
0.00375217	0.0403358	0.0136069	0.0909047	0.508039	0.116418	0.0137935	0.033567	0.0893804	0.0174017	22.4427	1	1
0.00482942	0.0362257	0.0154775	0.107328	0.535635	0.112994	0.0166848	0.0318326	0.124317	0.014676	18.7675	2	1
0.00400926	0.0312269	0.00791204	0.114394	0.511671	0.0914398	0.00756944	0.014625	0.207458	0.00969444	21.9867	0	1
0.00322581	0.0272718	0.0112578	0.10183	0.524086	0.125946	0.011868	0.0248262	0.146767	0.0222006	21.3399	3	0
0.00414424	0.0275621	0.0137497	0.108769	0.513394	0.119368	0.0205696	0.0373034	0.135161	0.0199791	20.5253	4	0
0.0035085	0.031203	0.0136471	0.104559	0.542348	0.113584	0.0221176	0.0309739	0.10869	0.020251	20.5467	2	0

**Gambar 4. 16** Contoh penggabungan fitur LBP, ridge density, dan lokasi jari

Pelatihan model melibatkan penggunaan data training, testing, dan validasi untuk melatih model SVM (Support Vector Machine). Berdasarkan sebaran data yang ditunjukkan pada gambar 4. Menunjukkan contoh fitur LBP, ridge density dan lokasi jari yang sudah diekstraksi kemudian di gabungkan. Dari hasil penggabungan fitur ini, terdapat 13 atribut yang dihasilkan untuk melatih model machine learning nantinya. Data kolom 1-10 menunjukkan hasil ekstraksi fitur

LBP, kolom 11 menunjukkan hasil ekstraksi ridge density, dan kolom 12-13 menunjukkan hasil ekstraksi lokasi jari

#### 4.2.4 Pembuatan Model SVM

Pelatihan model melibatkan penggunaan data training, testing, dan validasi untuk melatih model SVM (Support Vector Machine). Data training digunakan untuk mengajarkan model SVM mengenali pola-pola pada sidik jari, sedangkan data testing digunakan untuk menguji performa model yang telah dilatih. Data validasi digunakan untuk menyesuaikan parameter model agar sesuai dengan data yang digunakan. Langkah terakhir dalam proses pelatihan model adalah melatih model SVM menggunakan algoritma yang sesuai. Proses pelatihan model menggunakan *K-fold validation*, yang mana data training dibagi menjadi k subset yang sama besar, nilai k yang dipakai pada penelitian ini adalah k=10. Setiap subset kemudian di pilih secara bergantian sebagai data validasi, sementara subset lainnya digunakan sebagai data training. Proses ini diulang 10 kali sehingga setiap subset menjadi data validasi sekali. Dengan demikian, model dilatih dan dievaluasi menggunakan semua data training dan testing dengan kombinasi yang berbeda, memungkinkan evaluasi yang lebih reliabel terhadap performa model.

Berikut ini merupakan implementasi pembangunan model SVM yang dapat di lihat pada Code 4.5:

```
# Split dataset into training and test set
X_train, X_test, y_train, y_test = train_test_split(vector_features, labels,
test_size=0.3, random_state=100)

# Initialize the SVM classifier
svm_linear = SVC(kernel='linear', C=100, gamma='auto')

# Define k for k-fold cross-validation
k = 10
kf = KFold(n_splits=k, shuffle=True, random_state=100)

# Store metrics for each fold
fold accuracies = []
precision_scores = []
recall_scores = []
f1_scores = []

test accuracies = []

train_vold_size = dict()
val_vold_size = dict()

# Iterate through each fold and train the model
for fold, (train_index, val_index) in enumerate(kf.split(X_train), 1):
    X_train_fold, X_val_fold = X_train[train_index], X_train[val_index]
```

```

y_train_fold, y_val_fold = y_train[train_index], y_train[val_index]

# Track the size of training and validation sets for each fold
train_vold_size[fold] = len(train_index)
val_vold_size[fold] = len(val_index)
svm_linear.fit(X_train_fold, y_train_fold)

# Calculate accuracy for the fold
val_accuracy = svm_linear.score(X_val_fold, y_val_fold) * 100
fold accuracies.append(val_accuracy)
print(f"Accuracy report - Fold {fold}: {val_accuracy}")

#test check
test_accuracy = svm_linear.score(X_test,y_test) * 100
test_accurrencies.append(test_accuracy)
y_pred_test = svm_linear.predict(X_test)
precision = precision_score(y_test,y_pred_test)

# Predict labels for the validation set
y_pred_val = svm_linear.predict(X_val_fold)

# Generate confusion matrix for the validation set
cm = confusion_matrix(y_val_fold, y_pred_val)

# Calculate additional metrics
precision = precision_score(y_val_fold, y_pred_val, average='weighted')
recall = recall_score(y_val_fold, y_pred_val, average='weighted')
f1 = f1_score(y_val_fold, y_pred_val, average='weighted')
precision_scores.append(precision)
recall_scores.append(recall)
f1_scores.append(f1)
print(f"Precision: {precision:.2f}, Recall: {recall:.2f}, F1-Score: {f1:.2f}")

# Plot confusion matrix
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title(f"Confusion Matrix - Fold {fold}")
plt.xticks([0.5, 1.5], ["female", "male"])
plt.yticks([0.5, 1.5], ["female", "male"])
plt.show()
plt.close()
# Generate and print classification report
report = classification_report(y_val_fold, y_pred_val)
print(f"Classification Report - Fold {fold}:\n{report}")

```

**Code 4. 5** Kode untuk melakukan proses pembangunan model

Dalam implementasi k-fold validation ini, data latih dibagi menjadi 10 subkelompok yang sebanding. Pada setiap iterasi, model SVM dengan kernel linear (svm\_linear) dilatih menggunakan data latih dari subkelompok yang dipilih. Selanjutnya, akurasi model diukur dengan menggunakan data validasi dari subkelompok yang sama. Akurasi validasi untuk setiap lipatan (*fold*) dicetak dan ditambahkan ke dalam daftar fold\_accurrencies.

Selain itu, dilakukan prediksi label untuk data validasi menggunakan model yang telah dilatih. Selanjutnya, dilakukan perhitungan *matric confusion* untuk

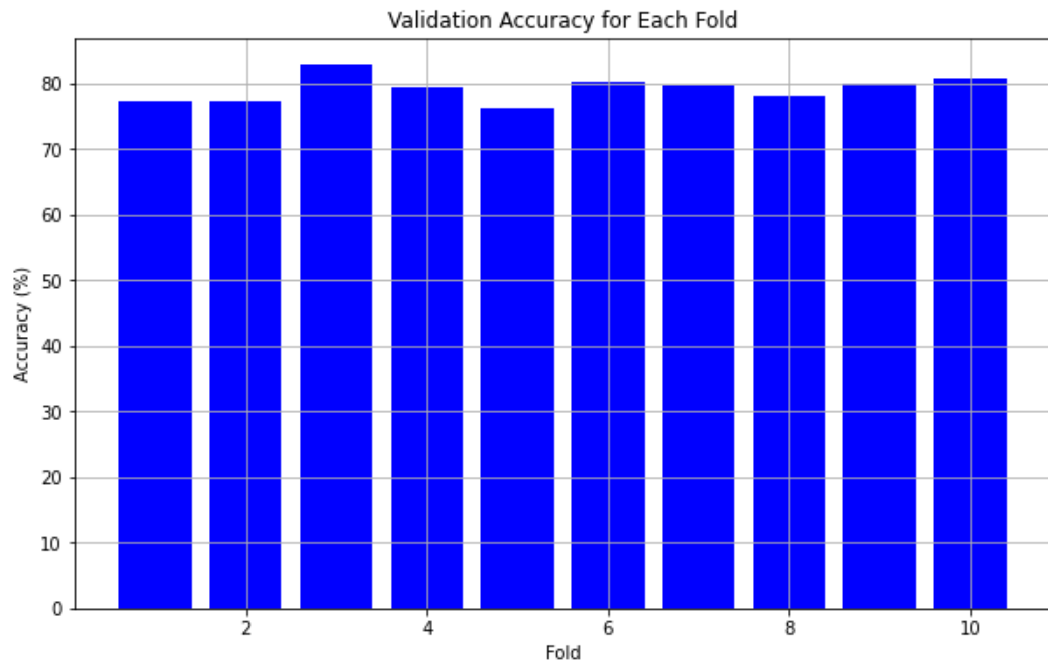
mengevaluasi performa model lebih lanjut. *Matric confusion* untuk setiap lipatan akan ditampilkan menggunakan menggunakan library matplotlib. Setelah seluruh iterasi selesai, sebuah diagram batang yang menunjukkan akurasi validasi untuk setiap lipatan akan dihasilkan. Ini memungkinkan visualisasi performa model pada setiap lipatan k-fold. Seluruh proses ini memungkinkan evaluasi yang lebih dapat diandalkan terhadap performa model SVM menggunakan k-fold validation dengan nilai k sebesar 10.

## **4.2 Analisis Sistem**

Percobaan pengujian pada penelitian ini dilakukan dengan 3 tahap pengujian yang di dapati hasil sebagai berikut:

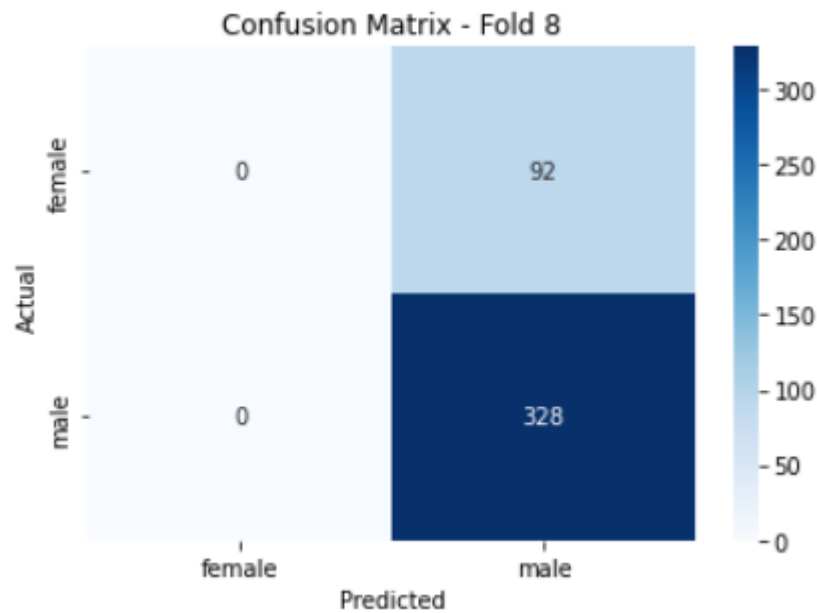
### **4.2.1 Pengujian terhadap 4730 sidik jari laki-laki dan 1230 sidik jari wanita**

Percobaan awal dalam penelitian ini adalah dengan menguji semua sidik jari yang terdapat pada folder Real. Citra yang terdapat pada folder ini menunjukkan ketimpangan antar kelas, di mana kelas pria memiliki citra lebih banyak dari pada kelas wanita yaitu sebanyak 3520 data. Hal ini menunjukkan adanya ketimbangan sebesar 58.33%. Pada Gambar 4.17 menunjukkan bagaimana akurasi yang dihasilkan oleh model ketika melakukan klasifikasi. Terlihat hasil akurasi yang cukup tinggi hingga mencapai 83%. Dengan rata-rata akurasi mencapai 79.21% dan dengan standar deviasi sebesar 1.8%, hal ini menunjukkan jika akurasi yang di hasilkan oleh percobaan ini cukup konsisten.



**Gambar 4. 17** Sebaran akurasi pada model pada penggabungan fitur LBP, *ridge density*, dan lokasi jari

Namun ketika di teliti lebih lanjut pada nilai *classification report* pada tiap iterasi k-fold di dapati jika model mengalami overtunning, dimana hasil prediksi kelas wanita memperoleh nilai akurasi sebesar 0%, yang mengartikan jika tidak ada sidik jari wanita yang berhasil diprediksi benar oleh model. Hal ini dapat di sebabkan oleh sebaran citra sidik jari yang tidak seimbang pada kelas wanita. Hasil *confusion metric* pada salah satu iterasi k-fold dapat dilihat pada Gambar 4.18.



**Gambar 4. 18** Hasil *confusion matrix* pada k-fold ke-8 pada hasil ekstraksi fitur LBP, *ridge density*, dan lokasi jari

Classification Report - Fold 8:				
	precision	recall	f1-score	support
0	0.00	0.00	0.00	92
1	0.78	1.00	0.88	328
accuracy			0.78	420
macro avg	0.39	0.50	0.44	420
weighted avg	0.61	0.78	0.68	420

**Gambar 4. 19** Hasil *classification report* pada k-fold ke-8 pada hasil ekstraksi fitur LBP, *ridge density*, dan lokasi jari

Dari data yang di tunjukkan pada Gambar 4.19 menunjukkan jika terjadi ketimbangan yang sangat tinggi pada sampel kelas wanita yang hanya sekitar 21.9% dari data validasi. Hal ini menyebabkan model cenderung mempelajari pola dari kelas pria dengan lebih baik daripada kelas wanita. Sebagai akibatnya, ketika dihadapkan pada data uji, model tidak dapat mengenali atau memprediksi kelas wanita dengan baik. Hal ini menghasilkan tidak ada kelas wanita yang berhasil di prediksi secara benar pada iterasi tersebut. Perhatikan hasil sebaran *confusion matrix* yang di dihasilkan dari setiap percobaan penggabungan fitur pada percobaan ini yang di tunjukkan pada Tabel 4.1.

**Tabel 4. 1 Sebaran nilai *confusion matrix* pada setiap percobaan penggabungan fitur**

Fold ke-	Fitur	Kelas	TP	TN	FP	FN
1	LBP, ridge density, dan lokasi jari	Pria	325	0	0	95
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	325	0	0	95
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	325	0	0	95
		Wanita	0	0	0	0
	Ridge density dan lokasi jari	Pria	325	0	0	95
		Wanita	0	0	0	0
2	LBP, ridge density, dan lokasi jari	Pria	325	0	0	95
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	325	0	0	95
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	325	0	0	95
		Wanita	0	0	0	0
	Ridge density dan lokasi jari	Pria	325	0	0	95
		Wanita	0	0	0	0
3	LBP, ridge density, dan lokasi jari	Pria	348	0	0	72
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	348	0	0	72
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	348	0	0	72
		Wanita	0	0	0	0
	Ridge density dan lokasi jari	Pria	348	0	0	72
		Wanita	0	0	0	0
4	LBP, ridge density, dan lokasi jari	Pria	334	0	0	86
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	334	0	0	86
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	334	0	0	86
		Wanita	0	0	0	0
	Ridge density dan lokasi jari	Pria	334	0	0	86
		Wanita	0	0	0	0
5	LBP, ridge density, dan lokasi jari	Pria	320	0	0	100
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	320	0	0	100
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	320	0	0	100
		Wanita	0	0	0	0

	Ridge density dan lokasi jari	Pria	320	0	0	100
		Wanita	0	0	0	0
6	LBP, ridge density, dan lokasi jari	Pria	337	0	0	83
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	337	0	0	83
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	337	0	0	83
		Wanita	0	0	0	0
	Ridge density dan lokasi jari	Pria	337	0	0	83
		Wanita	0	0	0	0
7	LBP, ridge density, dan lokasi jari	Pria	335	0	0	85
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	335	0	0	85
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	337	0	0	83
		Wanita	0	0	0	0
	Ridge density dan lokasi jari	Pria	337	0	0	83
		Wanita	0	0	0	0
8	LBP, ridge density, dan lokasi jari	Pria	328	0	0	92
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	328	0	0	92
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	328	0	0	92
		Wanita	0	0	0	0
	Ridge density dan lokasi jari	Pria	328	0	0	92
		Wanita	0	0	0	0
9	LBP, ridge density, dan lokasi jari	Pria	336	0	0	84
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	336	0	0	84
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	336	0	0	84
		Wanita	0	0	0	0
	Ridge density dan lokasi jari	Pria	336	0	0	84
		Wanita	0	0	0	0
10	LBP, ridge density, dan lokasi jari	Pria	339	0	0	81
		Wanita	0	0	0	0
	LBP dan ridge density	Pria	339	0	0	81
		Wanita	0	0	0	0
	LBP dan lokasi jari	Pria	339	0	0	81
		Wanita	0	0	0	0
	Ridge density dan lokasi jari	Pria	339	0	0	81
		Wanita	0	0	0	0



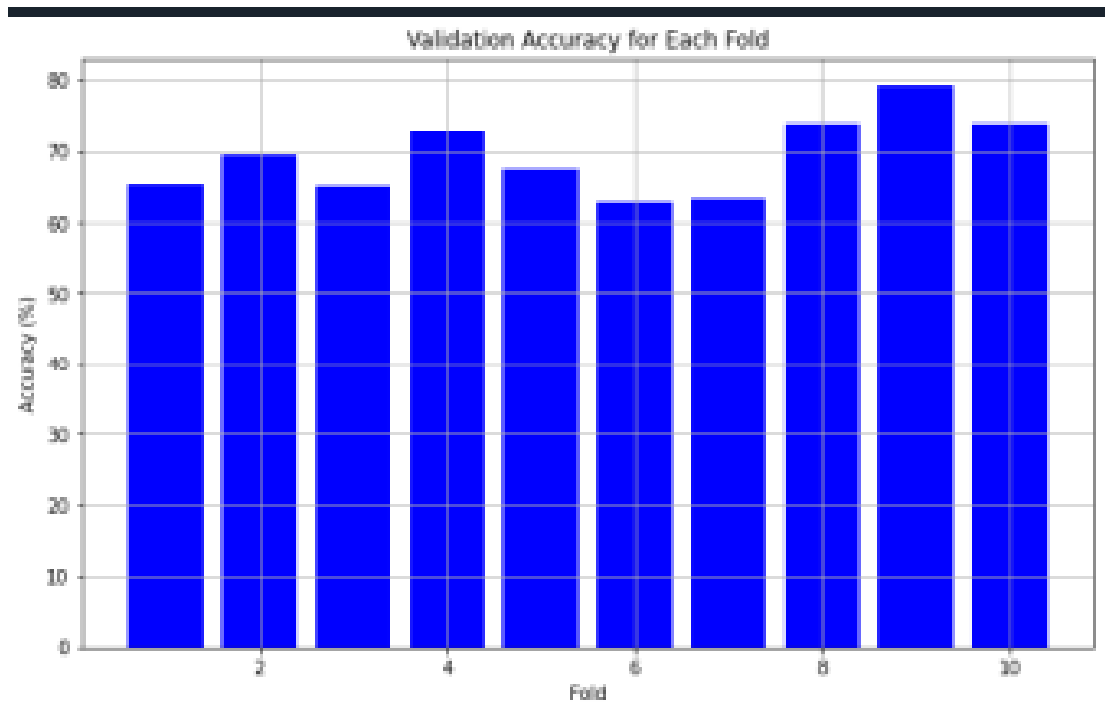
		Wanita	0	0	0	0
--	--	--------	---	---	---	---

### **Kesimpulan Pengujian terhadap 4730 sidik jari laki-laki dan 1230 sidik jari wanita**

Berdasarkan pada informasi yang ditampilkan pada Tabel 4.1 dapat diketahui jika nilai *True Negative*(TN) pada semua kelas termasuk kelas pria bernilai 0, hal ini menandakan jika model tidak dapat mengidentifikasi atau membedakan sidik jari wanita dan hanya sidik jari pria saja yang dapat diidentifikasi oleh model. Untuk semua percobaan penggabungan fitur nilai *confusion matrix* tidak mengalami perubahan sama sekali untuk kelas pria maupun wanita, yang artinya bahwa fitur yang ditambahkan tidak memberikan informasi tambahan yang signifikan bagi model untuk membedakan antara kedua kelas tersebut. Dari hasil tersebut dapat disimpulkan jika perbedaan jumlah data sidik jari pada tiap kelas sangat mempengaruhi pada kinerja model dalam melakukan prediksi dengan benar.

#### **4.2.2 Pengujian terhadap 1230 sidik jari laki-laki dan 1230 sidik jari wanita**

Pada percobaan ini, model SVM yang dilatih menggunakan kernel Linear dengan penggabungan tiga fitur utama, yaitu Local Binary Patterns (LBP), ridge density, dan informasi file citra sidik jari, telah menunjukkan variasi performa yang signifikan dalam 10-fold cross-validation. Hasil akurasi tertinggi dicapai pada fold ke-9 dengan nilai 79%, sedangkan akurasi terendah terjadi pada fold ke-3 dengan nilai 65%. Secara keseluruhan, rata-rata akurasi dari seluruh fold adalah 69.27%, dengan standar deviasi sebesar 5.15%. Hasil setiap fold dapat dilihat pada Gambar 4.20.



**Gambar 4. 20** Sebaran akurasi pada penggabungan fitur LBP, *ridge density*, dan lokasi jari

Fold dengan akurasi tertinggi (79% pada fold ke-9) menunjukkan potensi model yang baik ketika dilatih dan divalidasi dengan data yang memiliki distribusi yang mungkin lebih ideal. Namun, adanya fold dengan akurasi rendah (65% pada fold ke-3) menandakan bahwa model mungkin mengalami kesulitan dengan data yang berbeda secara signifikan dari data pelatihan lainnya, yang mengarah pada kemungkinan adanya overfitting atau underfitting pada beberapa subset data. Rata-rata akurasi sebesar 69.27% mengindikasikan bahwa model SVM dengan kernel Linear mampu mengklasifikasikan data dengan cukup baik. Namun, variasi akurasi yang cukup besar, yang ditunjukkan oleh standar deviasi 5.15%, mengungkapkan adanya inkonsistensi performa model di berbagai fold. Variasi ini mungkin disebabkan oleh perbedaan dalam distribusi data pada setiap subset pelatihan dan validasi, yang menunjukkan bahwa model ini sensitif terhadap perbedaan-perbedaan tersebut.

Dalam penelitian ini, dilakukan empat percobaan pengujian terhadap variasi fitur untuk memahami bagaimana setiap fitur mempengaruhi performa model.

Tabel 4.2 menunjukkan hasil dari empat kombinasi fitur yang digunakan untuk melatih model.

**Tabel 4. 2** hasil metrik evaluasi dan sebaran data dari empat kombinasi fitur

Fold-ke	Fitur	Akurasi	Recal	F-1 score	Presisi	Train Data	Test Data	Val data
1	LBP, ridge density, dan lokasi jari	0.65	0.65	0.65	0.66	1549	738	173
	LBP dan ridge density	65.8	0.66	0.66	0.67	1549	738	173
	LBP dan lokasi jari	0.65	0.65	0.65	0.66	1549	738	173
	Ridge density dan lokasi jari	0.58	0.58	0.59	0.59	1549	738	173
2	LBP, ridge density, dan lokasi jari	0.69	0.69	0.69	0.69	1549	738	173
	LBP dan ridge density	67.0	0.67	0.67	0.67	1549	738	173
	LBP dan lokasi jari	0.66	0.66	0.66	0.66	1549	738	173
	Ridge density dan lokasi jari	0.62	0.62	0.62	0.62	1549	738	173
3	LBP, ridge density, dan lokasi jari	0.65	0.65	0.65	0.65	1550	738	172
	LBP dan ridge density	60.4	0.60	0.60	0.61	1550	738	172
	LBP dan lokasi jari	0.65	0.65	0.64	0.65	1550	738	172
	Ridge density dan lokasi jari	0.59	0.59	0.59	0.59	1550	738	172
4	LBP, ridge density, dan lokasi jari	0.73	0.73	0.73	0.73	1550	738	172
	LBP dan ridge density	70.3	0.70	0.70	0.70	1550	738	172
	LBP dan lokasi jari	0.74	0.74	0.74	0.75	1550	738	172
	Ridge density dan lokasi jari	0.65	0.65	0.65	0.65	1550	738	172
5	LBP, ridge density, dan lokasi jari	0.67	0.67	0.68	0.68	1550	738	172
	LBP dan ridge density	66.2	0.66	0.66	0.67	1550	738	172
	LBP dan lokasi jari	0.67	0.67	0.67	0.67	1550	738	172
	Ridge density dan lokasi jari	0.65	0.65	0.64	0.64	1550	738	172

6	LBP, ridge density, dan lokasi jari	0.63	0.63	0.63	0.63	1550	738	172
	LBP dan ridge density	62.7	0.63	0.63	0.63	1550	738	172
	LBP dan lokasi jari	0.65	0.65	0.65	0.65	1550	738	172
	Ridge density dan lokasi jari	0.57	0.57	0.57	0.57	1550	738	172
7	LBP, ridge density, dan lokasi jari	0.63	0.63	0.63	0.64	1550	738	172
	LBP dan ridge density	59.8	0.60	0.60	0.60	1550	738	172
	LBP dan lokasi jari	0.62	0.62	0.61	0.62	1550	738	172
	Ridge density dan lokasi jari	0.65	0.65	0.64	0.65	1550	738	172
8	LBP, ridge density, dan lokasi jari	0.74	0.74	0.74	0.74	1550	738	172
	LBP dan ridge density	67.4	0.67	0.67	0.68	1550	738	172
	LBP dan lokasi jari	0.73	0.73	0.73	0.73	1550	738	172
	Ridge density dan lokasi jari	0.65	0.65	0.65	0.66	1550	738	172
9	LBP, ridge density, dan lokasi jari	0.79	0.79	0.79	0.79	1550	738	172
	LBP dan ridge density	77.9	0.78	0.78	0.78	1550	738	172
	LBP dan lokasi jari	0.76	0.76	0.76	0.76	1550	738	172
	Ridge density dan lokasi jari	0.71	0.71	0.71	0.71	1550	738	172
10	LBP, ridge density, dan lokasi jari	0.74	0.74	0.74	0.74	1550	738	172
	LBP dan ridge density	77.3	0.77	0.77	0.78	1550	738	172
	LBP dan lokasi jari	0.73	0.73	0.73	0.74	1550	738	172
	Ridge density dan lokasi jari	0.73	0.73	0.73	0.74	1550	738	172

## 2.a Percobaan penggabungan fitur LBP, ridge density, dan lokasi jari

**Tabel 4. 3** Metrik evaluasi dari penggabungan fitur LBP, ridge density, dan lokasi jari

No	Metrik evaluasi	Rata-rata	Standar deviasi
----	-----------------	-----------	-----------------

1	Akurasi	69.28	5.122
2	Presisi	0.70	0.051
3	Recall	0.69	0.051
4	F1-Score	0.69	0.051

Pada percobaan ini memperoleh hasil rata – rata akurasi pada 10 percobaan k-fold sebesar 0.69 atau 69% dan Standar deviasi yang diperoleh dari tiap perubahan akurasi di setiap percobaan pelatihan sebesar 0.051 atau 5.1%. Meskipun akurasi rata-rata sebesar 69% menunjukkan performa yang layak, variasi yang signifikan dalam performa model antara percobaan k-fold menunjukkan adanya perbedaan dalam distribusi data atau masalah dalam generalisasi model. Hasil presisi rata rata yang di peroleh pada percobaan ini sebesar 0.7 atau 70%. Dengan nilai rata-rata 70%, model memiliki kemampuan untuk memprediksi hasil positif secara akurat. Ini berarti sekitar 70% dari hasil positif yang diprediksi oleh model adalah benar. Hasil recall rata rata yang di peroleh pada percobaan ini sebesar 0.7 atau 70%. Dengan nilai rata-rata 70%, model mampu mengidentifikasi sekitar 70% dari seluruh sampel positif yang ada. Hasil F-1 score rata rata yang di peroleh pada percobaan ini sebesar 0.7 atau 70%. Dengan nilai rata-rata 70%, model menunjukkan keseimbangan yang baik antara presisi dan recall. F-1 score adalah *harmonic mean* dari kedua metrik tersebut, sehingga nilai yang tinggi menunjukkan keseimbangan yang lebih baik antara kemampuan model dalam mengklasifikasikan hasil positif dan negatif.

Dalam Gambar 4.21, disajikan hasil akurasi yang diperoleh dari pengujian pada data testing menggunakan model yang telah dilatih pada setiap percobaan k-fold. Total 738 citra sidik jari digunakan dalam pengujian performa k-fold. Hasil menunjukkan bahwa akurasi tertinggi dicapai pada percobaan fold ke-10, mencapai 68.69%, sementara akurasi terendah terjadi pada percobaan fold ke-3, dengan tingkat akurasi sebesar 67.07%.

test accuracies - List (10 elements)

Ind ▲	Type	Size	
0	float64	1	68.42818428184282
1	float64	1	67.88617886178862
2	float64	1	68.15718157181571
3	float64	1	67.07317073170732
4	float64	1	68.02168021680217
5	float64	1	67.75067750677508
6	float64	1	67.75067750677508
7	float64	1	68.29268292682927
8	float64	1	68.15718157181571
9	float64	1	68.69918699186992

**Gambar 4. 21** Sebaran hasil akurasi pada pengujian data *testing*

Rata-rata akurasi dari keseluruhan percobaan k-fold pada data *testing* adalah sebesar 68.19%, dengan standar deviasi sebesar 0.57%. Rata-rata akurasi mencerminkan kemampuan model dalam mengklasifikasikan data secara umum, sementara standar deviasi mengindikasikan variasi atau fluktuasi dalam performa model antar percobaan k-fold. Berdasarkan nilai rata – rata akurasi yang di dapat menunjukkan jika sekitar kurang lebih 503 citra sidik jari telah diklasifikasikan secara akurat dan 235 sidik jari di klasifikasikan secara tidak akurat. Nilai standar deviasi yang relatif rendah menunjukkan bahwa variasi performa model antar fold adalah relatif kecil, menandakan konsistensi yang baik dalam hasil akurasi. Meskipun akurasi 68.19% menunjukkan kualitas yang cukup baik, masih ada ruang untuk meningkatkan performa model. Pengembangan lebih lanjut dalam hal pengolahan data, pemilihan fitur, atau penyesuaian parameter model dapat membantu meningkatkan akurasi dan kinerja keseluruhan model.

Untuk menilai bagaimana performa model dalam memprediksi tiap kelas dapat dilihat dari hasil *confusion matrix* pada tiap iterasi fold. Hasil total *confusion matrix* pada setiap iterasi fold dapat dilihat pada tabel 4.4

No	Kelas	TP	TN	FP	FN
1	Pria	637	540	278	267
2	Wanita	540	637	267	278

**Tabel 4. 4 Total *confusion matrix* pada setiap iterasi fold**

Berdasarkan dari nilai *confusion matrix* pada Tabel 4.4 di ketahui jika nilai metrik evaluasi pada kelas pria adalah sebagai berikut:

- Akurasi: 68.36%
- Presisi: 69.62%
- Recall: 70.42%
- F1-Score: 70.02%

Hal ini dapat di ketahui jika kelas Pria memiliki tingkat akurasi yang baik dalam memprediksi kelas Pria adalah Pria secara tepat dengan nilai persentase sebesar 70.42%. Ini ditunjukkan oleh nilai recall, yang mengukur kemampuan model untuk mendeteksi semua instance dari kelas pria secara tepat. Recall menunjukkan seberapa baik model dapat mengidentifikasi semua kasus positif sebenarnya (pria) di antara semua kasus yang sebenarnya positif (pria). Akurasi menunjukkan bahwa model mampu mengklasifikasikan dengan benar sekitar 68.36% dari seluruh prediksi untuk kelas pria. Ini mencerminkan kemampuan model untuk melakukan prediksi yang benar secara keseluruhan, meskipun nilai ini menunjukkan bahwa masih ada ruang untuk perbaikan dalam mengklasifikasikan pria dan wanita. Presisi untuk kelas pria adalah 69.62%. Ini berarti dari semua prediksi yang diklasifikasikan sebagai pria, 69.62% di antaranya benar-benar pria. Tingginya nilai presisi menunjukkan bahwa model cukup efektif dalam menghindari false positive (FP), yaitu kasus di mana model salah mengklasifikasikan wanita sebagai pria. F1-Score untuk kelas pria adalah 70.02%. F1-Score adalah gabungan dari presisi dan recall, memberikan gambaran seimbang tentang performa model. Nilai 70.02% menunjukkan bahwa model memiliki keseimbangan yang baik antara kemampuan untuk mendeteksi pria (recall) dan menghindari kesalahan prediksi pria (presisi).

Untuk kelas wanita nilai metrik evaluasi adalah sebagai berikut:

- Akurasi: 68.36%
- Presisi: 66.94%
- Recall: 66.01%

- F1-Score: 66.47%

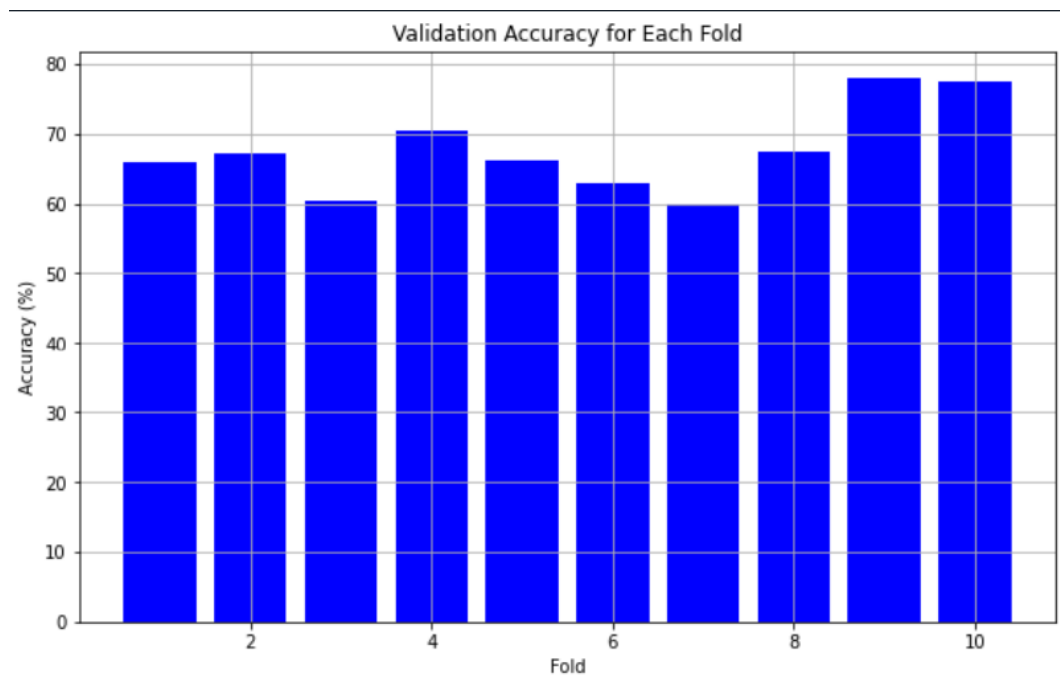
Hal ini dapat diketahui jika performa presisi pada model ketika di uji untuk kelas wanita lebih tinggi dari pada metrik evaluasi lainnya. Model dapat mengenali perbedaan antara kelas pria dan wanita lebih baik, semua prediksi yang diklasifikasikan sebagai wanita, sekitar 66.94% di antaranya benar-benar wanita. Nilai presisi ini menunjukkan bahwa model cukup efektif dalam menghindari false positive (FP), yaitu kasus di mana model salah mengklasifikasikan pria sebagai wanita. Recall untuk kelas wanita adalah 66.01%. Ini menunjukkan bahwa model mampu mendeteksi 66.01% dari semua wanita yang benar-benar wanita (true positive rate). Meskipun lebih rendah dari recall untuk kelas pria, ini masih menunjukkan bahwa model cukup baik dalam menemukan instance positif (wanita) di antara data yang sebenarnya positif. F1-Score untuk kelas wanita adalah 66.47%. F1-Score adalah gabungan dari presisi dan recall, memberikan gambaran seimbang tentang performa model. Nilai ini menunjukkan bahwa model memiliki keseimbangan yang cukup baik antara kemampuan untuk mendeteksi wanita (recall) dan menghindari kesalahan prediksi wanita (presisi).

Akurasi untuk kedua kelas adalah sama, yaitu 68.36%, yang menunjukkan bahwa model memiliki performa yang konsisten secara keseluruhan dalam memprediksi kedua kelas. Namun, presisi, recall, dan F1-Score menunjukkan perbedaan yang signifikan antara kedua kelas. Presisi untuk kelas pria adalah 69.62%, sedangkan untuk kelas wanita adalah 66.94%. Hal ini berarti bahwa ketika model memprediksi "pria," kemungkinan prediksi tersebut benar lebih tinggi dibandingkan ketika model memprediksi "wanita." Ini menunjukkan bahwa model lebih jarang membuat kesalahan dengan memprediksi pria secara salah (false positive) dibandingkan wanita. Recall untuk kelas pria adalah 70.42%, lebih tinggi dibandingkan recall untuk kelas wanita yang sebesar 66.01%. Ini berarti model lebih baik dalam mengidentifikasi semua instance yang benar-benar pria dibandingkan wanita. Dengan recall yang lebih tinggi, model cenderung lebih sedikit membuat kesalahan dengan tidak mengenali pria yang sebenarnya pria (false negative) dibandingkan wanita. F1-Score, yang menggabungkan presisi dan recall,



adalah 70.02% untuk pria dan 66.47% untuk wanita. F1-Score yang lebih tinggi untuk pria mengindikasikan bahwa keseimbangan antara presisi dan recall untuk pria lebih baik dibandingkan wanita. Secara keseluruhan, model lebih andal dalam mengklasifikasikan pria dibandingkan wanita.

## 2.b Penggabungan fitur LBP dan *ridge density*



**Gambar 4. 22** Sebaran hasil akurasi pada percobaan penggabungan fitur LBP dan *ridge density*

**Tabel 4. 5** nilai metrik evaluasi pada penggabungan fitur LBP dan *ridge density*

No	Metrik evaluasi	Rata-rata	Standar deviasi
1	Akurasi	67.54	5.890
2	Presisi	0.68	0.059
3	Recall	0.68	0.059
4	F1-Score	0.68	0.059

Akurasi tertinggi yang didapatkan pada hasil percobaan ini di dapati pada fold ke-9 dengan nilai akurasi sebesar 78% dan hasil akurasi terendah pada fold ke-6 sebesar 63%. Meskipun pada proses pelatihan model berhasil mencapai tingkat akurasi sebesar 78%, namun hasil rata – rata akurasi pada 10 percobaan k-fold sebesar 0.67 atau 67% yang menunjukkan adanya fluktuasi dalam performa model di berbagai fold dan Standar deviasi yang diperoleh dari tiap perubahan akurasi di

setiap percobaan pelatihan sebesar 0.059 atau 5.9%. Meskipun akurasi rata-rata sebesar 67% menunjukkan performa yang layak, namun akurasi yang diperoleh lebih rendah dari pada akurasi saat menambahkan fitur lokasi jari pada *vector features*. Standar deviasi yang di hasilkan pada percobaan ini juga lebih tinggi 0.008 dari pada percobaan pertama yang mengindikasikan jika terdapat nilai variasi yang relatif lebih tinggi pada distribusi data pada percobaan ini. Selain itu, model prediksi ini juga menghasilkan nilai presisi, recall, dan F1-score yang masing-masing sebesar 0,68. Presisi dan recall yang sama-sama sebesar 0,68 mengindikasikan keseimbangan yang baik antara tingkat prediksi benar positif dan cakupan pengenalan semua kasus positif. Nilai F1-score yang juga 0,68 mengonfirmasi bahwa keseimbangan antara presisi dan recall terjaga dengan baik.

Dalam Gambar 4.33, disajikan hasil akurasi yang diperoleh dari pengujian pada data *testing* menggunakan model yang telah dilatih pada setiap percobaan k-fold. Total 738 citra sidik jari digunakan dalam pengujian performa k-fold. Hasil menunjukkan bahwa akurasi tertinggi dicapai pada percobaan fold ke-10, mencapai 68.02%, sementara akurasi terendah terjadi pada percobaan fold ke-3, dengan tingkat akurasi sebesar 67.20%.

test accuracies - List (10 elements)

Ind ▲	Type	Size	
0	float64	1	67.88617886178862
1	float64	1	67.61517615176152
2	float64	1	67.20867208672087
3	float64	1	67.61517615176152
4	float64	1	67.75067750677508
5	float64	1	66.93766937669376
6	float64	1	67.47967479674797
7	float64	1	67.88617886178862
8	float64	1	67.47967479674797
9	float64	1	68.02168021680217

**Gambar 4. 23** Hasil uji akurasi pada data *tesing* pada penggabungan fitur LBP dan *ridge density*

Rata-rata akurasi dari keseluruhan percobaan k-fold pada data *testing* adalah sebesar 67.78%, dengan standar deviasi sebesar 0.32%. Dengan nilai standar deviasi yang kecil menunjukkan bahwa variasi performa model antar percobaan k-fold pada percobaan data *testing* relatif kecil yang menandakan konsistensi yang baik dalam hasil akurasi. Berdasarkan nilai rata-rata akurasi yang diperoleh, sekitar kurang lebih 500 citra sidik jari telah diklasifikasikan secara akurat dan 238 citra sidik jari diklasifikasikan secara tidak akurat. Nilai rentang akurasi yang relatif kecil sebesar 0.82% menunjukkan bahwa variasi performa model antar percobaan k-fold adalah relatif kecil, menandakan konsistensi yang baik dalam hasil akurasi. Meskipun akurasi 67.78% menunjukkan kualitas yang cukup baik, masih ada ruang untuk meningkatkan performa model.

**Tabel 4. 6** Hasil total *confusion matrix* penggabungan fitur LBP dan *ridge density*

No	Kelas	TP	TN	FP	FN
1	Pria	612	551	255	304

2	Wanita	551	612	304	255
---	--------	-----	-----	-----	-----

Berdasarkan dari nilai *confusion matrix* pada Tabel 4.6 di ketahui jika nilai metrik evaluasi pada kelas pria adalah sebagai berikut:

- Akurasi: 67.54%
- Presisi: 70.59%
- Recall: 68.81%
- F1-Score: 68.65%

Hal ini dapat di ketahui jika kelas Pria memiliki tingkat akurasi yang baik dalam memprediksi kelas Pria adalah Pria secara tepat dengan nilai persentase sebesar 70.59%. Ini ditunjukkan oleh nilai presisi, yang mengukur kemampuan model untuk mendeteksi mengklasifikasikan citra sebagai pria adalah benar-benar pria. Tingginya nilai presisi menunjukkan bahwa model cukup efektif dalam menghindari false positive (FP), yaitu kasus di mana model salah mengklasifikasikan wanita sebagai pria. Recall menunjukkan seberapa baik model dapat mengidentifikasi semua kasus positif sebenarnya (pria) di antara semua kasus yang sebenarnya positif (pria). Akurasi menunjukkan bahwa model mampu mengklasifikasikan dengan benar sekitar 67.54% dari seluruh prediksi untuk kelas pria. F1-Score untuk kelas pria adalah 68.65%. F1-Score adalah gabungan dari presisi dan recall, memberikan gambaran seimbang tentang performa model. Nilai 68.65% menunjukkan bahwa model memiliki keseimbangan yang baik antara kemampuan untuk mendeteksi pria (recall) dan menghindari kesalahan prediksi pria (presisi).

Untuk kelas wanita nilai metrik evaluasi adalah sebagai berikut:

- Akurasi: 68.36%
- Presisi: 66.94%
- Recall: 66.01%
- F1-Score: 66.47%

Hal ini dapat di ketahui jika peforma recal pada model ketika di uji untuk kelas wanita lebih tinggi dari pada metrik evaluasi lainnya. Model dapat mengenali

mengidentifikasi semua contoh wanita yang sebenarnya sebagai wanita (nilai TP yang tinggi) sekitar 66.94% di antaranya benar-benar wanita. Nilai presisi ini menunjukkan bahwa model cukup efektif dalam menghindari false positive (FP), yaitu kasus di mana model salah mengklasifikasikan pria sebagai wanita. Recall untuk kelas wanita adalah 66.01%. Ini menunjukkan bahwa model mampu mendeteksi 66.01% dari semua wanita yang benar-benar wanita (true positive rate). Meskipun lebih rendah dari recall untuk kelas pria, ini masih menunjukkan bahwa model cukup baik dalam menemukan instance positif (wanita) di antara data yang sebenarnya positif. F1-Score untuk kelas wanita adalah 66.47%. F1-Score adalah gabungan dari presisi dan recall, memberikan gambaran seimbang tentang performa model. Nilai ini menunjukkan bahwa model memiliki keseimbangan yang cukup baik antara kemampuan untuk mendeteksi wanita (recall) dan menghindari kesalahan prediksi wanita (presisi).

Akurasi untuk kedua kelas adalah sama, yaitu 68.36%, yang menunjukkan bahwa model memiliki performa yang konsisten secara keseluruhan dalam memprediksi kedua kelas. Namun, presisi, recall, dan F1-Score menunjukkan perbedaan yang signifikan antara kedua kelas. Presisi untuk kelas pria adalah 69.62%, sedangkan untuk kelas wanita adalah 66.94%. Hal ini berarti bahwa ketika model memprediksi "pria," kemungkinan prediksi tersebut benar lebih tinggi dibandingkan ketika model memprediksi "wanita." Ini menunjukkan bahwa model lebih jarang membuat kesalahan dengan memprediksi pria secara salah (false positive) dibandingkan wanita. Recall untuk kelas pria adalah 70.42%, lebih tinggi dibandingkan recall untuk kelas wanita yang sebesar 66.01%. Ini berarti model lebih baik dalam mengidentifikasi semua instance yang benar-benar pria dibandingkan wanita. Dengan recall yang lebih tinggi, model cenderung lebih sedikit membuat kesalahan dengan tidak mengenali pria yang sebenarnya pria (false negative) dibandingkan wanita. F1-Score, yang menggabungkan presisi dan recall, adalah 70.02% untuk pria dan 66.47% untuk wanita. F1-Score yang lebih tinggi untuk pria mengindikasikan bahwa keseimbangan antara presisi dan recall untuk pria lebih baik dibandingkan wanita. Secara keseluruhan, model lebih andal dalam mengklasifikasikan pria dibandingkan wanita.

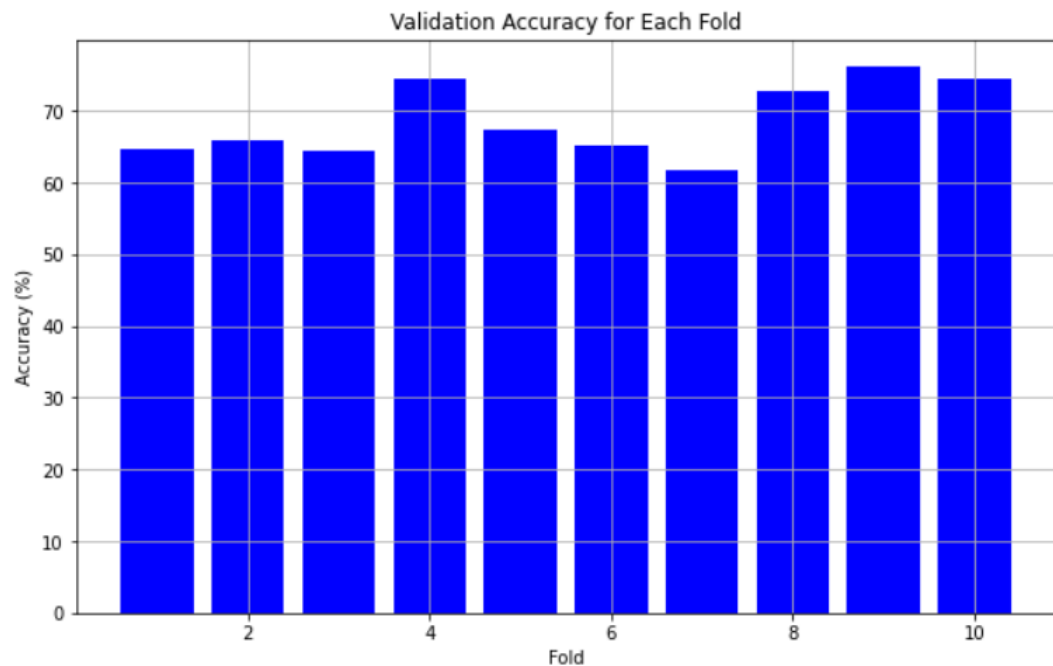
## 2.c Penggabungan fitur LBP dan lokasi jari

**Tabel 4. 7** Hasil Metrik evaluasi pada penggabungan fitur LBP dan lokasi jari

No	Metrik evaluasi	Rata-rata	Standar deviasi
1	Akurasi	68.70	0.049
2	Presisi	0.69	0.049
3	Recall	0.69	0.049
4	F1-Score	0.69	0.049

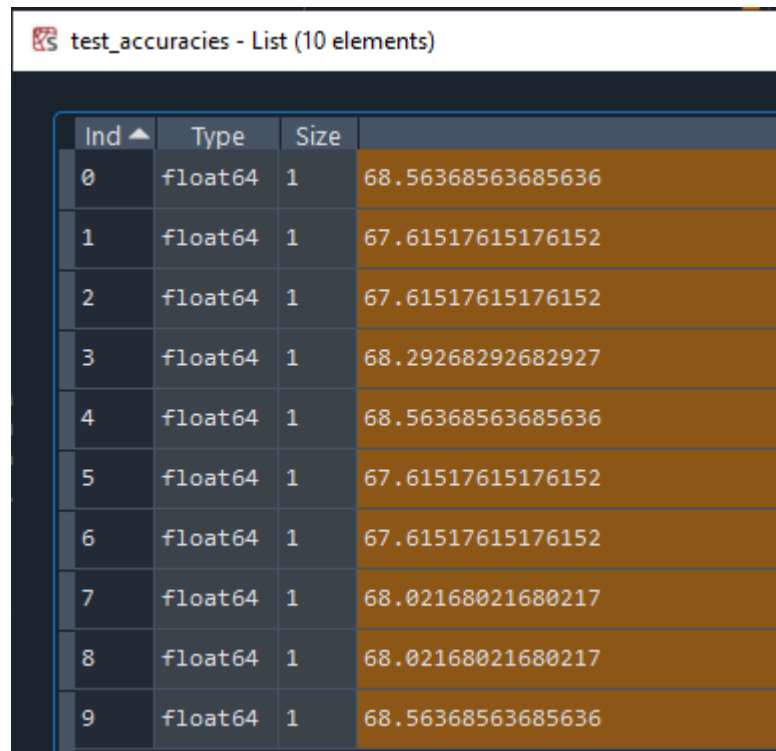
Akurasi tertinggi yang didapatkan pada hasil percobaan ini di dapati pada fold ke-9 dengan nilai akurasi sebesar 76% dan hasil akurasi terendah pada fold ke-7 sebesar 62%. Meskipun pada proses pelatihan model berhasil mencapai tingkat akurasi sebesar 76%, namun hasil rata – rata akurasi pada 10 percobaan k-fold sebesar 0.687 atau 68,7% yang menunjukkan adanya fluktuasi dalam performa model di berbagai fold, namun tidak begitu besar pada dua percobaan sebelumnya dan Standar deviasi yang diperoleh dari tiap perubahan akurasi di setiap percobaan pelatihan sebesar 0.049 atau 4.9%. Meskipun akurasi rata-rata sebesar 68,7% menunjukkan performa yang layak, namun akurasi yang diperoleh lebih rendah dari pada akurasi saat menambahkan fitur lokasi jari pada *vector features*. Akan tetapi Standar deviasi yang di hasilkan pada percobaan ini lebih rendah 0.003 dari pada percobaan pertama yang mengindikasikan jika nilai variasi yang di hasilkan lebih rendah pada distribusi data dalam percobaan ini. Selain itu, model prediksi ini juga menghasilkan nilai presisi, recall, dan F1-score yang masing-masing sebesar 0,69. Presisi dan recall yang sama-sama sebesar 0,69 mengindikasikan keseimbangan yang baik antara tingkat prediksi benar positif dan cakupan pengenalan semua kasus positif. Nilai F1-score yang juga 0,69 mengonfirmasi bahwa keseimbangan antara

presisi dan recall terjaga dengan baik.



**Gambar 4. 24** Hasil sebaran akurasi data validasi pada Penggabungan fitur LBP dan lokasi jari

Dalam Gambar 4.45, disajikan hasil akurasi yang diperoleh dari pengujian pada data *testing* menggunakan model yang telah dilatih pada setiap percobaan k-fold. Total 738 citra sidik jari digunakan dalam pengujian performa k-fold. Hasil menunjukkan bahwa akurasi tertinggi dicapai pada percobaan fold ke-1, ke-3 dan ke-10, mencapai 68.56%, sementara akurasi terendah terjadi pada percobaan fold ke-2, ke-3, ke-6, dan ke-7, dengan tingkat akurasi sebesar 67.61%.



Ind	Type	Size	
0	float64	1	68.56368563685636
1	float64	1	67.61517615176152
2	float64	1	67.61517615176152
3	float64	1	68.29268292682927
4	float64	1	68.56368563685636
5	float64	1	67.61517615176152
6	float64	1	67.61517615176152
7	float64	1	68.02168021680217
8	float64	1	68.02168021680217
9	float64	1	68.56368563685636

**Gambar 4. 25** Hasil akurasi pada data *testing* pada penggabungan fitur LBP dan lokasi jari

Rata-rata akurasi dari keseluruhan percobaan k-fold pada data testing adalah sebesar 68.045%, dengan standar deviasi sebesar 0.38%. Dengan nilai standar deviasi yang kecil ini, variasi performa model antar percobaan k-fold pada data testing relatif kecil, yang menandakan konsistensi yang baik dalam hasil akurasi. Berdasarkan nilai rata-rata akurasi yang diperoleh, sekitar kurang lebih 502 citra sidik jari telah diklasifikasikan secara akurat dan 236 citra sidik jari diklasifikasikan secara tidak akurat. Nilai rentang akurasi yang relatif kecil sebesar 0.95% menunjukkan bahwa variasi performa model antar percobaan k-fold adalah relatif kecil, menandakan konsistensi yang baik dalam hasil akurasi. Meskipun akurasi 68.045% menunjukkan kualitas yang cukup baik, masih ada ruang untuk meningkatkan performa model.

**Tabel 4. 8** Hasil *confusion matrix* pada penggabungan fitur LBP dan lokasi jari

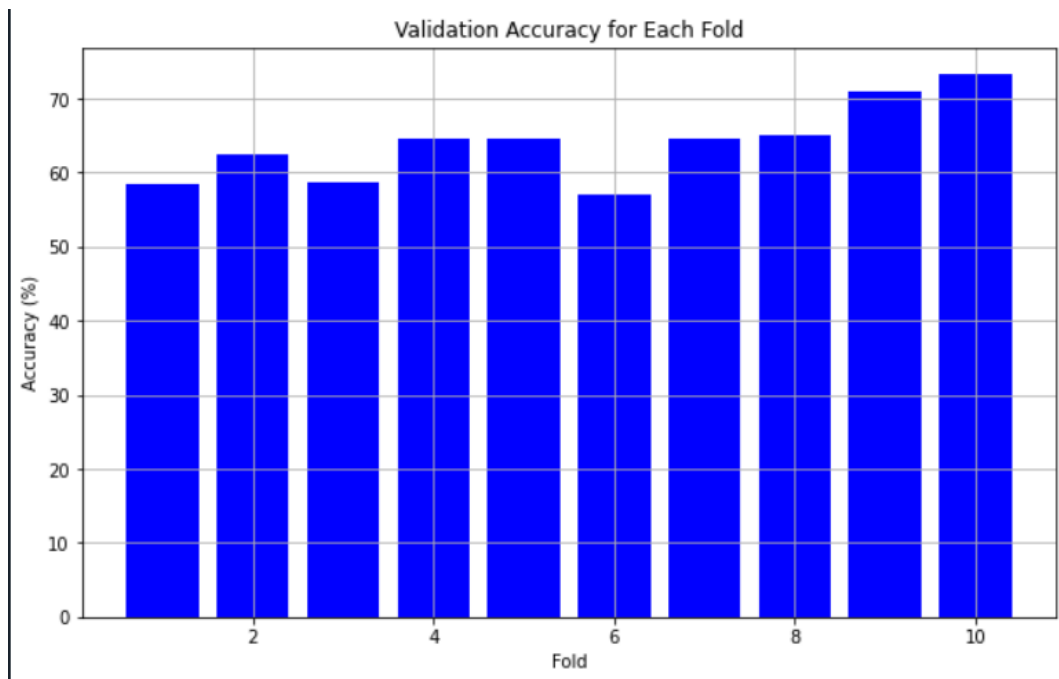
		TP	TN	FP	FN
Fold-1	Pria	50	62	24	37



	Wanita	62	50	37	24
Fold-2	Pria	62	52	25	34
	Wanita	52	62	34	25
Fold-3	Pria	59	52	23	38
	Wanita	52	59	38	23
Fold-4	Pria	73	55	17	27
	Wanita	55	73	27	17
Fold-5	Pria	70	46	27	29
	Wanita	46	70	29	27
Fold-6	Pria	63	49	27	33
	Wanita	49	63	33	27
Fold-7	Pria	59	47	27	39
	Wanita	47	59	39	27
Fold-8	Pria	68	57	17	30
	Wanita	57	68	30	17
Fold-9	Pria	70	61	17	24
	Wanita	61	70	24	17
Fold-10	Pria	75	53	14	30
	Wanita	53	75	30	14

Analisis hasil dari data confusion matrix menunjukkan adanya variasi performa yang menarik perhatian dalam model klasifikasi. Dalam kesepuluh fold evaluasi, terlihat fluktuasi yang signifikan antara jumlah True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN), menunjukkan bahwa kinerja model tidak stabil di setiap fold. Meskipun demikian, observasi yang menarik adalah tidak adanya ketidakseimbangan kelas yang signifikan, yang dilihat dari distribusi seimbang jumlah sampel yang diprediksi sebagai pria dan wanita. Meskipun model menunjukkan kinerja yang relatif baik dalam membedakan kedua kategori, terdapat ruang untuk analisis lebih lanjut guna memahami faktor-faktor yang berkontribusi terhadap fluktuasi tersebut. Evaluasi lebih lanjut diperlukan untuk mendapatkan pemahaman yang lebih komprehensif tentang kekuatan dan kelemahan model ini, serta untuk mengevaluasi efektivitas penggabungan fitur LBP dan lokasi jari dalam tugas klasifikasi ini secara keseluruhan.

## 2.d Penggabungan fitur *ridge density* dan lokasi jari



**Gambar 4. 26** Hasil sebaran akurasi pada pengujian data validasi pada penggabungan fitur *ridge density* dan lokasi jari

**Tabel 4. 9** Hasil metrik evaluasi pada penggabungan fitur *ridge density* dan lokasi jari

No	Metrik evaluasi	Rata-rata	Standar deviasi
1	Akurasi	63.94	4.970
2	Presisi	0.64	0.0497
3	Recall	0.64	0.0497
4	F1-Score	0.64	0.0497

Akurasi tertinggi yang didapatkan pada hasil percobaan ini di dapati pada fold ke-10 dengan nilai akurasi sebesar 73% dan hasil akurasi terendah pada fold ke-6 sebesar 57%. Model pada percobaan ini mengalami penurunan akurasi jika di bandingkan dengan akurasi tertinggi yang dihasilkan pada percobaan penggabungan fitur LBP dan lokasi jari, sehingga hal ini menunjukkan jika fitur LBP lebih mempengaruhi peforma model, jika di bandingkan dengan fitur *ridge density* secara keseluruhan. Meskipun pada proses pelatihan model berhasil mencapai tingkat akurasi sebesar 73%, namun hasil rata – rata akurasi pada 10 percobaan k-fold sebesar 0.639 atau 63.9% yang menunjukkan adanya fluktuasi dalam performa model di berbagai fold, nilai rata – rata akurasi yang di hasilkan

pada percobaan ini merupakan hasil terendah dari 3 percobaan penggabungan fitur sebelumnya. Standar deviasi yang di peroleh sebesar 0.0497 atau 5. Standar deviasi yang di hasilkan pada percobaan ini merupakan yang terendah dari semua percobaan penggabungan fitur, hal ini mengindikasikan jika nilai variasi yang di hasilkan pada percobaan ini relatif lebih rendah pada distribusi data dari pada percobaan lainnya.

**Tabel 4. 10** Hasil *confusion matrix* pada penggabungan fitur *ridge density* dan lokasi jari

		TP	TN	FP	FN
Fold-1	Pria	41	60	33	39
	Wanita	60	41	39	33
Fold-2	Pria	55	53	32	33
	Wanita	53	55	33	32
Fold-3	Pria	55	46	27	44
	Wanita	46	55	44	27
Fold-4	Pria	59	52	31	30
	Wanita	52	59	30	31
Fold-5	Pria	69	42	28	33
	Wanita	42	69	33	28
Fold-6	Pria	55	42	35	39
	Wanita	42	55	39	35
Fold-7	Pria	61	50	25	36
	Wanita	50	61	36	25
Fold-8	Pria	66	46	19	41
	Wanita	46	66	41	10
Fold-9	Pria	65	57	22	28
	Wanita	57	65	28	22
Fold-10	Pria	74	52	15	31
	Wanita	52	74	31	15

Dari hasil *confusion matrix* untuk penggabungan fitur *ridge density* dan lokasi jari, terlihat variasi dalam kinerja model klasifikasi antara fold-fold yang berbeda. Perhatikan bahwa untuk setiap fold, terjadi fluktuasi dalam jumlah True

Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Ini menunjukkan bahwa kinerja model tidak stabil di setiap iterasi fold.

Pada umumnya, terdapat kecenderungan bahwa jumlah True Positive (TP) lebih rendah daripada jumlah True Negative (TN) pada banyak fold, terutama pada kategori pria. Namun, pada beberapa fold seperti fold ke-8, terdapat peningkatan yang signifikan dalam jumlah False Negative (FN) pada kategori wanita, yang mungkin perlu ditelusuri lebih lanjut.

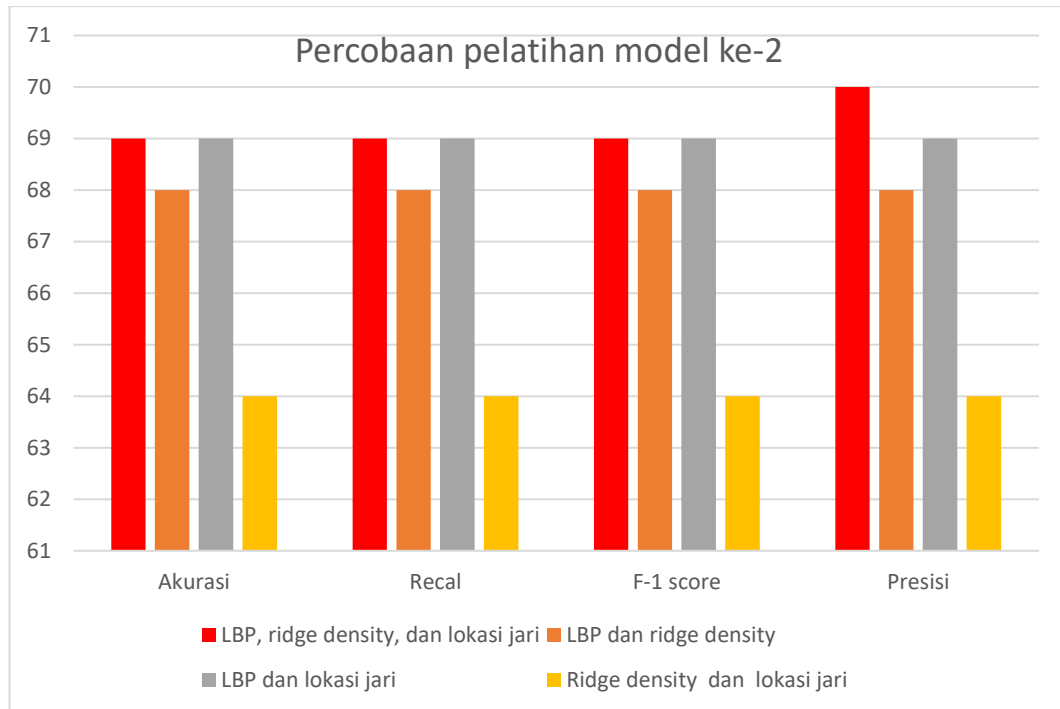
Selain itu, terlihat bahwa terdapat perbedaan yang cukup besar antara jumlah FP dan FN di beberapa fold, menandakan adanya potensi untuk meningkatkan kinerja model dalam mengurangi kesalahan prediksi salah (False Negatives dan False Positives).

Kesimpulannya, meskipun terdapat variasi dalam kinerja model antara fold-fold, penggabungan fitur *ridge density* dan lokasi jari menunjukkan potensi sebagai fitur yang berguna dalam membedakan antara pria dan wanita. Namun, evaluasi lebih lanjut diperlukan untuk memahami faktor-faktor yang menyebabkan fluktuasi dalam kinerja model dan untuk meningkatkan akurasi serta konsistensi prediksi.

### **Kesimpulan Pengujian terhadap 1230 sidik jari laki-laki dan 1230 sidik jari wanita**

Berdasarkan hasil perhitungan 10-fold pada matriks *confusion* menggunakan kombinasi fitur *ridge density* dan lokasi jari untuk klasifikasi jenis kelamin, ditemukan bahwa model menunjukkan kinerja yang stabil dan cukup baik secara keseluruhan. Meskipun terdapat variasi dalam performa antara kelas wanita dan pria di beberapa fold, secara umum, model cenderung lebih baik dalam mengenali pria daripada wanita, terutama terlihat dari nilai recall yang lebih tinggi untuk kelas pria. Meskipun demikian, performa model dalam mengenali wanita juga cukup memuaskan, dengan nilai presisi dan F1-score yang tinggi.

Untuk melihat kinerja metrik evaluasi secara menyeluruh dari keempat percobaan penggabungan fitur pada percobaan pembangunan model kedua pada penelitian ini, dapat dilihat melalui Grafik 1.



**Grafik 4. 1 Grafik nilai rata – rata metrik evaluasi pada uji coba data validasi**

Secara menyeluruh hasil analisis pada ke-4 percobaan penggabungan fitur dengan metode k-fold validation dalam melatih model untuk melakukan klasifikasi sidik jari dapat dirangkum jika Kombinasi LBP, *ridge density*, dan lokasi jari menghasilkan performa terbaik dengan akurasi, recall, dan F1 score tertinggi (0.79) pada percobaan ke-9. Fitur lokasi jari membantu dalam meningkatkan rata – rata akurasi model sebanyak 1.74%, fitur *ridge density* membantu dalam meningkatkan rata – rata akurasi model sebanyak 0.58%, dan fitur LBP membantu dalam meningkatkan rata – rata akurasi model sebanyak 5.34%. Dari hasil tersebut menunjukkan jika fitur yang paling krusial dalam meningkatkan akurasi model dalam setiap percobaan adalah LBP, hal ini dikarenakan atribut fitur yang dihasilkan lebih banyak ketimbang dengan fitur *ridge density* dan lokasi jari.

Metode LBP, *ridge density*, dan lokasi jari merupakan metode identifikasi sidik jari yang paling akurat dengan akurasi 69%, recall 69%, F-1 score 69%, dan presisi

70%. Nilai presisi 70% menandakan bahwa dari 172 data yang diidentifikasi oleh metode tersebut pada data validasi, 70 data benar-benar merupakan sidik jari yang sesuai. Dengan kata lain, metode tersebut memiliki tingkat kepastian yang tinggi dalam mengidentifikasi sidik jari. Nilai presisi ini menunjukkan bahwa meskipun metode tersebut tidak dapat mengidentifikasi semua sidik jari dengan benar (recall 69%), namun sidik jari yang diidentifikasinya sebagian besar benar-benar sidik jari yang sesuai. Nilai akurasi 69%, recall 69%, presisi 70%, dan F-1 score 69% menunjukkan bahwa metode identifikasi sidik jari tersebut memiliki kinerja yang baik, namun masih ada ruang untuk perbaikan.

Metode LBP dan lokasi jari memiliki performa yang setara dengan metode 1, namun dengan kompleksitas perhitungan yang lebih rendah karena tidak melibatkan ridge density. Metode ridge density dan lokasi jari memiliki performa yang paling rendah di antara 4 metode yang diuji. Penggunaan fitur secara terpisah, baik LBP, *ridge density*, atau lokasi jari, cenderung memberikan performa yang lebih rendah dibandingkan dengan kombinasi fitur. Misalnya, penggunaan hanya ridge density dan lokasi jari menghasilkan akurasi rata-rata sekitar 0.62 dan F1-score rata-rata sekitar 0.62. Terdapat variasi performa antara fold-folds, namun secara umum, model memiliki kinerja yang stabil dan cukup baik dalam mengenali kedua jenis kelamin.

#### 4.2.3 Pengujian terhadap 4770 sidik jari laki-laki dan 4770 sidik jari wanita

**Tabel 4. 11** hasil metrik evaluasi dan sebaran data dari empat kombinasi fitur pada pengujian ke-3

Fold-ke	Fitur	Akurasi	Recal	F-1 score	Presisi	Train Data	Test Data	Val data
1	LBP, ridge density, dan lokasi jari	0.87	0.88	0.87	0.90	6010	2862	668
	LBP dan ridge density	0.87	0.87	0.87	0.90	6010	2862	668
	LBP dan lokasi jari	0.87	0.87	0.87	0.88	6010	2862	668
	Ridge density dan lokasi jari	0.62	0.62	0.62	0.62	6010	2862	668

2	LBP, ridge density, dan lokasi jari	0.86	0.87	0.86	0.88	6010	2862	668
	LBP dan ridge density	0.86	0.86	0.86	0.88	6010	2862	668
	LBP dan lokasi jari	0.87	0.87	0.87	0.88	6010	2862	668
	Ridge density dan lokasi jari	0.62	0.62	0.62	0.62	6010	2862	668
3	LBP, ridge density, dan lokasi jari	0.85	0.86	0.86	0.87	6010	2862	668
	LBP dan ridge density	0.85	0.85	0.85	0.87	6010	2862	668
	LBP dan lokasi jari	0.84	0.84	0.84	0.86	6010	2862	668
	Ridge density dan lokasi jari	0.59	0.59	0.59	0.59	6010	2862	668
4	LBP, ridge density, dan lokasi jari	0.87	0.87	0.87	0.89	6010	2862	668
	LBP dan ridge density	0.87	0.87	0.87	0.89	6010	2862	668
	LBP dan lokasi jari	0.87	0.87	0.87	0.88	6010	2862	668
	Ridge density dan lokasi jari	0.62	0.61	0.61	0.61	6010	2862	668
5	LBP, ridge density, dan lokasi jari	0.87	0.88	0.88	0.90	6010	2862	668
	LBP dan ridge density	0.88	0.88	0.88	0.90	6010	2862	668
	LBP dan lokasi jari	0.87	0.87	0.87	0.88	6010	2862	668
	Ridge density dan lokasi jari	0.61	0.61	0.61	0.61	6010	2862	668
6	LBP, ridge density, dan lokasi jari	0.86	0.87	0.87	0.89	6010	2862	668
	LBP dan ridge density	0.87	0.87	0.87	0.89	6010	2862	668
	LBP dan lokasi jari	0.86	0.86	0.86	0.88	6010	2862	668
	Ridge density dan lokasi jari	0.62	0.62	0.62	0.63	6010	2862	668
7	LBP, ridge density, dan lokasi jari	0.85	0.85	0.85	0.88	6010	2862	668
	LBP dan ridge density	0.85	0.85	0.85	0.88	6010	2862	668
	LBP dan lokasi jari	0.85	0.85	0.85	0.87	6010	2862	668
	Ridge density dan lokasi jari	0.59	0.59	0.59	0.59	6010	2862	668
8	LBP, ridge density, dan lokasi jari	0.86	0.87	0.86	0.88	6010	2862	668
	LBP dan ridge density	0.87	0.87	0.87	0.89	6010	2862	668
	LBP dan lokasi jari	0.86	0.86	0.86	0.87	6010	2862	668
	Ridge density dan lokasi jari	0.61	0.61	0.61	0.61	6010	2862	668

9	LBP, ridge density, dan lokasi jari	0.86	0.86	0.86	0.88	6011	2862	667
	LBP dan ridge density	0.87	0.87	0.86	0.89	6011	2862	667
	LBP dan lokasi jari	0.85	0.85	0.85	0.86	6011	2862	667
	Ridge density dan lokasi jari	0.61	0.61	0.61	0.61	6011	2862	667
10	LBP, ridge density, dan lokasi jari	0.86	0.86	0.86	0.88	6011	2862	667
	LBP dan ridge density	0.86	0.86	0.85	0.87	6011	2862	667
	LBP dan lokasi jari	0.85	0.85	0.85	0.86	6011	2862	667
	Ridge density dan lokasi jari	0.63	0.63	0.63	0.63	6011	2862	667

Dalam percobaan terakhir menggunakan total 9520 citra sidik jari, dengan jumlah kelas pria dan wanita yang seimbang, diperoleh hasil yang signifikan. Berdasarkan data performa model yang ditunjukkan oleh Tabel 4, diperoleh hasil akurasi tertinggi yang dapat dicapai oleh model sebesar 0,88 atau 88%. Dari uji performa model pada percobaan ini, dapat disimpulkan bahwa jumlah data set sangat mempengaruhi hasil akurasi. Hasil uji performa model untuk setiap percobaan penggabungan fitur adalah sebagai berikut:

### 3.a Penggabungan fitur LBP, *ridge density*, dan lokasi jari

Hasil analisis dari percobaan penggabungan fitur LBP, ridge density, dan lokasi jari pada model yang di bangun pada percobaan ini menunjukkan kinerja yang sangat sangat baik. Tabel 4.12 menunjukkan nilai rata-rata metrik evaluasi yang di peroleh oleh pada percobaan ini.

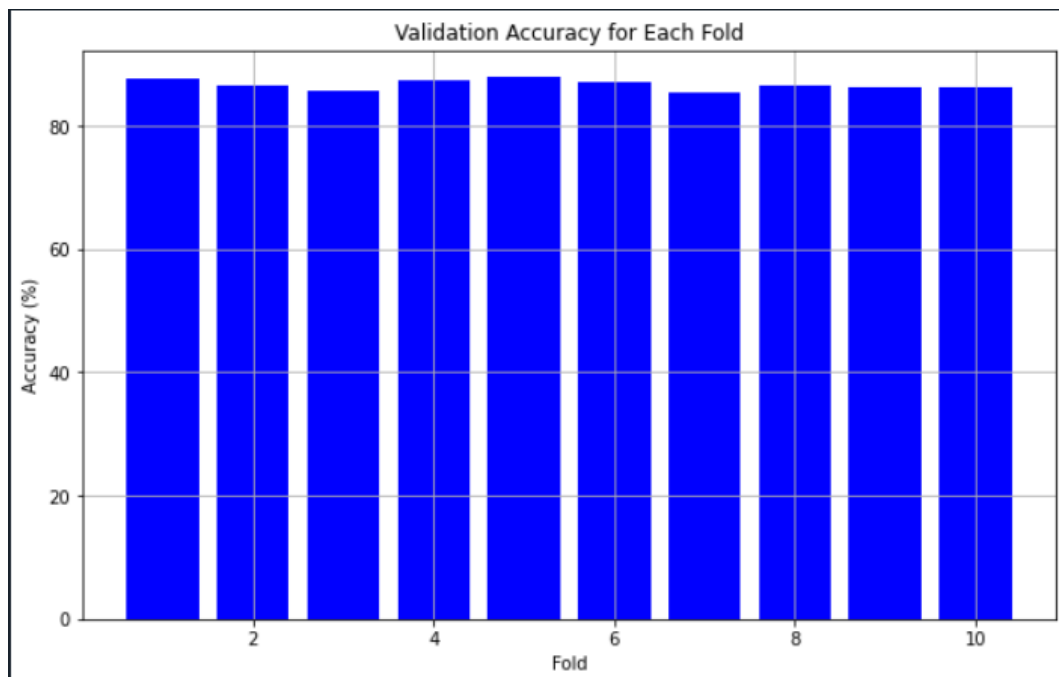
**Tabel 4. 12** hasil metrik evaluasi pada penggabungan fitur LBP, ridge density, dan lokasi jari

No	Metrik evaluasi	Rata-rata	Standar deviasi
1	Akurasi	86.64	0.008
2	Presisi	0.88	0.008
3	Recall	0.87	0.008



4	F1-Score	0.86	0.008
---	----------	------	-------

Dari metrik evaluasi yang diperoleh pada tabel 1.1, rata – rata akurasi sebesar 86.64% mengindikasikan bahwa model berhasil mengklasifikasikan sebagian besar dari total 667 sidik jari dengan benar. Jadi, sekitar 579 sidik jari dengan benar oleh model klasifikasi. Presisi sebesar 0.88 menunjukkan bahwa sekitar 88% dari semua sidik jari yang diprediksi sebagai milik satu jenis kelamin tertentu adalah benar-benar milik jenis kelamin tersebut. Recall, yang mencapai 0.87, menunjukkan bahwa model mampu mengidentifikasi sekitar 87% dari semua sidik jari yang sebenarnya milik jenis kelamin tertentu. F-1 Score, dengan nilai 0.86, menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall. Standar deviasi yang rendah (0.008) untuk setiap metrik evaluasi menunjukkan bahwa model memiliki konsistensi yang baik dalam performanya saat diuji pada data validasi. Ini menunjukkan bahwa model cenderung memberikan hasil yang stabil dan dapat diandalkan dalam mengklasifikasikan jenis kelamin berdasarkan citra sidik jari.



**Gambar 4. 27** Hasil sebaran akurasi pada penggabungan fitur LBP, ridge density, dan lokasi jari

Gambar 4.68 menunjukkan jika persebaran akurasi yang di hasilkan oleh percobaan ini. Nilai akurasi tertinggi yang dicapai adalah sebesar 87.87%, terjadi pada fold kelima dari proses validasi silang k=10, sementara nilai akurasi terendah sebesar 85.33% diamati pada fold ketujuh. Perbedaan yang ditemukan antara nilai akurasi tertinggi dan terendah dari proses validasi adalah sekitar 2.54%, yang secara relatif kecil dalam evaluasi kinerja model. Hal ini menunjukkan bahwa model memiliki konsistensi yang cukup baik dalam memprediksi jenis kelamin berdasarkan sidik jari di seluruh fold validas.

test accuracies - List (10 elements)

Ind ▲	Type	Size	
0	float64	1	86.23340321453529
1	float64	1	86.16352201257862
2	float64	1	86.26834381551363
3	float64	1	86.19846261355696
4	float64	1	86.19846261355696
5	float64	1	86.16352201257862
6	float64	1	86.19846261355696
7	float64	1	86.23340321453529
8	float64	1	86.12858141160028
9	float64	1	86.19846261355696

**Gambar 4. 28** Hasil sebaran akurasi pada pengujian data *testing* pada penggabungan fitur *ridge density* dan lokasi jari

Hasil pengujian terhadap data *testing* yang terdiri dari 2862 data sidik jari menunjukkan bahwa model klasifikasi sidik jari memiliki nilai rata-rata akurasi sebesar 86%. Ini adalah hasil yang signifikan dan menunjukkan bahwa model dapat memprediksi dengan benar dalam sebagian besar kasus. Selain itu, nilai akurasi ini tidak jauh berbeda dengan hasil yang diperoleh saat menguji data validasi, yang mengindikasikan bahwa performa model konsisten dan andal ketika dihadapkan pada data uji baru. Konsistensi ini memperkuat kepercayaan bahwa model tidak

hanya terlatih dengan baik pada data yang ada tetapi juga memiliki generalisasi yang baik untuk data yang belum pernah dilihat sebelumnya.

**Tabel 4. 13** Hasil *confusion matrix* pada penggabungan fitur LBP, ridge density, dan lokasi jari

		TP	TN	FP	FN
Fold-1	Pria	337	248	3	80
	Wanita	248	337	80	3
Fold-2	Pria	334	244	8	82
	Wanita	244	334	82	8
Fold-3	Pria	315	258	13	82
	Wanita	258	315	82	13
Fold-4	Pria	306	278	6	78
	Wanita	278	306	77	6
Fold-5	Pria	309	278	6	75
	Wanita	278	309	75	6
Fold-6	Pria	338	243	5	82
	Wanita	243	338	82	5
Fold-7	Pria	313	257	9	89
	Wanita	257	315	89	9
Fold-8	Pria	323	255	13	77
	Wanita	255	323	77	13
Fold-9	Pria	325	250	12	80
	Wanita	250	325	80	12
Fold-10	Pria	329	250	8	80
	Wanita	250	329	80	8

Dari hasil confusion matrix untuk penggabungan fitur LBP, ridge density, dan lokasi jari, terlihat bahwa model klasifikasi menunjukkan kinerja yang konsisten dan cukup baik dalam membedakan antara pria dan wanita. Dalam kesepuluh fold evaluasi, jumlah True Positive (TP) dan True Negative (TN) cenderung tinggi, sementara jumlah False Positive (FP) dan False Negative (FN) relatif rendah.

Hal ini menunjukkan bahwa model mampu dengan baik dalam mengidentifikasi kedua kategori dengan akurasi yang tinggi. Perlu dicatat bahwa fluktuasi yang terlihat dalam jumlah FP dan FN antara fold-fold mungkin disebabkan oleh variasi dalam data pelatihan yang digunakan dalam setiap iterasi fold.

Secara umum, konsistensi kinerja model dalam mengklasifikasikan pria dan wanita dengan akurasi tinggi menunjukkan bahwa penggabungan fitur LBP, ridge density, dan lokasi jari merupakan pendekatan yang efektif dalam tugas klasifikasi ini. Meskipun demikian, evaluasi lebih lanjut mungkin diperlukan untuk memvalidasi kinerja model pada dataset yang lebih luas dan untuk memastikan generalisasi yang baik dari model ini dalam kasus penggunaan yang berbeda.

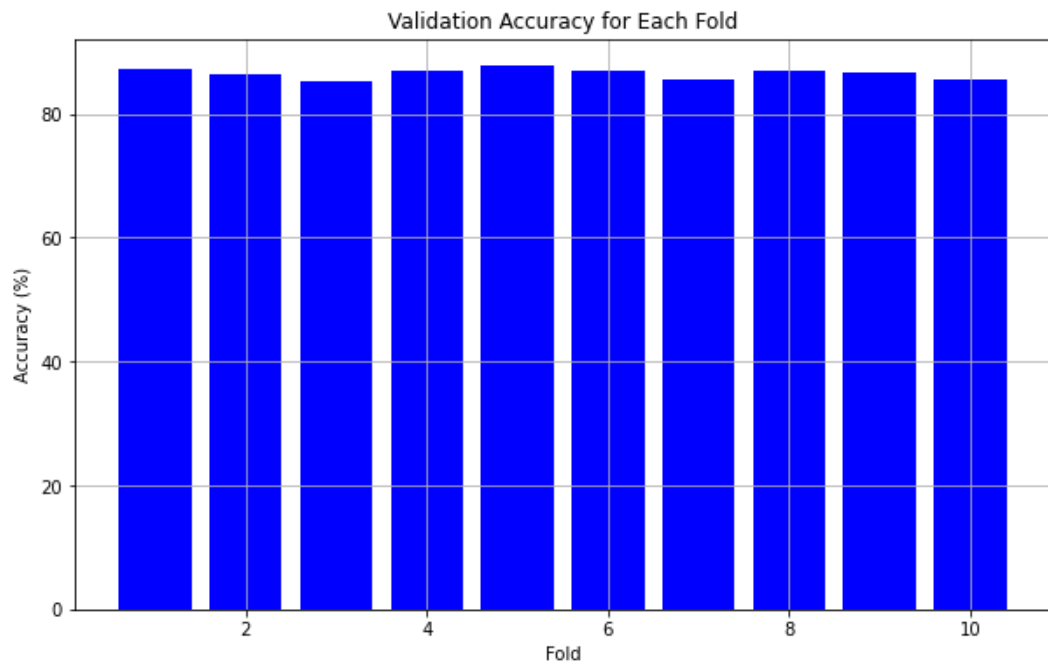
### 3.b Penggabungan fitur LBP dan *ridge density*

Hasil analisis dari percobaan penggabungan fitur LBP dan ridge density pada model yang di bangun pada percobaan ini menunjukkan kinerja yang sangat sangat baik seperti pada percobaan pertama. Tabel 4.14 menunjukkan nilai rata-rata metrik evaluasi yang di peroleh oleh pada percobaan ini.

**Tabel 4. 14** hasil metrik evaluasi pada penggabungan fitur LBP dan *ridge density*

No	Metrik evaluasi	Rata-rata	Standar deviasi
1	Akurasi	86.49	0.008
2	Presisi	0.89	0.008
3	Recall	0.86	0.008
4	F1-Score	0.86	0.008

Dari metrik evaluasi yang diperoleh pada tabel 1.1, rata – rata akurasi sebesar 86.49% mengindikasikan bahwa model berhasil mengklasifikasikan sebagian besar dari total 667 sidik jari dengan benar. Jadi, sekitar 577 sidik jari dengan benar oleh model klasifikasi. Presisi sebesar 0.89 menunjukkan bahwa sekitar 89% dari semua sidik jari yang diprediksi sebagai milik satu jenis kelamin tertentu adalah benar-benar milik jenis kelamin tersebut. Recall, yang mencapai 0.86, menunjukkan bahwa model mampu mengidentifikasi sekitar 86% dari semua sidik jari yang sebenarnya milik jenis kelamin tertentu. F-1 Score, dengan nilai 0.86, menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall. Standar deviasi yang rendah (0.008) untuk setiap metrik evaluasi menunjukkan bahwa model memiliki konsistensi yang baik dalam performanya saat diuji pada data validasi. Ini menunjukkan bahwa model cenderung memberikan hasil yang stabil dan dapat diandalkan dalam mengklasifikasikan jenis kelamin berdasarkan citra sidik jari.



**Gambar 4. 29** Hasil sebaran akurasi pada pengujian data validasi pada penggabungan fitur LBP dan *ridge density*

Gambar 4.78 menunjukkan persebaran akurasi yang dihasilkan oleh percobaan ini. Nilai akurasi tertinggi yang dicapai adalah sebesar 87.72%, terjadi pada fold kelima dari proses validasi silang k=10, sementara nilai akurasi terendah sebesar 85.33% diamati pada fold ketiga. Perbedaan yang ditemukan antara nilai akurasi tertinggi dan terendah dari proses validasi adalah sekitar 2.40%, yang secara relatif kecil dalam evaluasi kinerja model. Hal ini menunjukkan bahwa model memiliki konsistensi yang cukup baik dalam memprediksi jenis kelamin berdasarkan sidik jari di seluruh fold validasi.

test accuracies - List (10 elements)

Ind ▲	Type	Size	
0	float64	1	86.16352201257862
1	float64	1	86.16352201257862
2	float64	1	86.19846261355696
3	float64	1	86.23340321453529
4	float64	1	86.12858141160028
5	float64	1	86.12858141160028
6	float64	1	86.16352201257862
7	float64	1	86.12858141160028
8	float64	1	86.12858141160028
9	float64	1	86.19846261355696

**Gambar 4. 30** Hasil sebaran akurasi pada pengujian data *testing* pada penggabungan fitur *ridge density* dan lokasi jari

Hasil pengujian terhadap data *testing* yang terdiri dari 2862 data sidik jari menunjukkan bahwa model klasifikasi sidik jari memiliki nilai rata-rata akurasi sebesar 86%. Ini adalah hasil yang signifikan dan menunjukkan bahwa model dapat memprediksi dengan benar dalam sebagian besar kasus. Selain itu, nilai akurasi ini tidak jauh berbeda dengan hasil yang diperoleh saat menguji data validasi, yang mengindikasikan bahwa performa model konsisten dan andal ketika dihadapkan pada data uji baru. Konsistensi ini memperkuat kepercayaan bahwa model tidak hanya terlatih dengan baik pada data yang ada tetapi juga memiliki generalisasi yang baik untuk data yang belum pernah dilihat sebelumnya.

**Tabel 4. 15** Hasil *confusion matrix* pada penggabungan fitur LBP dan *ridge density*

		TP	TN	FP	FN
Fold-1	Pria	338	245	2	83
	Wanita	245	338	83	2

Fold-2	Pria	334	242	8	84
	Wanita	242	334	84	8
Fold-3	Pria	316	254	12	86
	Wanita	254	316	86	12
Fold-4	Pria	304	277	8	79
	Wanita	277	304	79	8
Fold-5	Pria	313	273	2	80
	Wanita	273	313	80	2
Fold-6	Pria	340	240	3	85
	Wanita	240	340	85	3
Fold-7	Pria	317	254	5	92
	Wanita	254	317	92	5
Fold-8	Pria	328	253	8	79
	Wanita	253	328	79	8
Fold-9	Pria	332	245	5	85
	Wanita	245	332	85	5
Fold-10	Pria	326	245	11	85
	Wanita	245	326	85	11

Dari hasil confusion matrix untuk penggabungan fitur LBP dan ridge density, terlihat bahwa model klasifikasi menunjukkan kinerja yang cukup baik dalam membedakan antara pria dan wanita. Dalam kesepuluh fold evaluasi, jumlah True Positive (TP) dan True Negative (TN) relatif tinggi, sedangkan jumlah False Positive (FP) dan False Negative (FN) cenderung rendah.

Hal ini menunjukkan bahwa model dapat mengidentifikasi kedua kategori dengan akurasi yang tinggi dan memiliki kemampuan dalam mengurangi kesalahan prediksi. Meskipun terdapat fluktuasi dalam jumlah FP dan FN antara fold-fold, kinerja model secara keseluruhan masih cukup konsisten.

Penggabungan fitur LBP dan ridge density nampaknya merupakan pendekatan yang efektif dalam tugas klasifikasi ini, dengan model mampu memanfaatkan informasi dari kedua jenis fitur untuk melakukan prediksi dengan baik. Namun, evaluasi lebih lanjut mungkin diperlukan untuk memvalidasi kinerja model pada dataset yang lebih besar dan untuk mengevaluasi generalisasi model ini dalam skenario penggunaan yang berbeda.

### 3.c Penggabungan fitur LBP dan lokasi jari

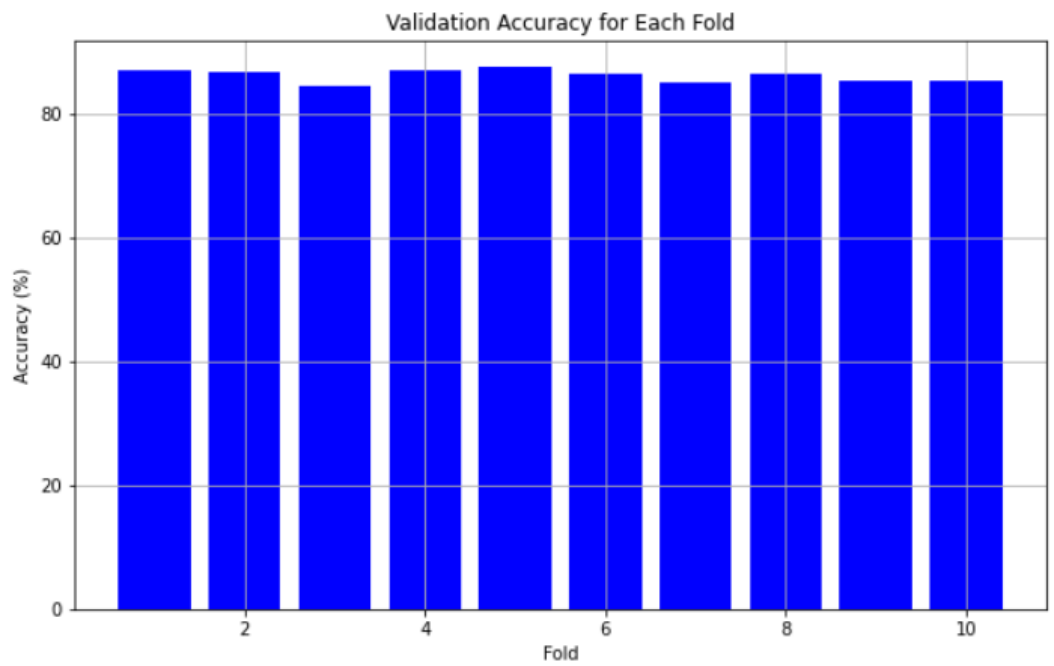
Hasil analisis dari percobaan penggabungan fitur LBP dan lokasi jari pada model yang di bangun pada percobaan ini menunjukkan kinerja yang sangat sangat baik. Tabel 1.1 menunjukkan nilai rata-rata metrik evaluasi yang di peroleh oleh pada percobaan ini.

**Tabel 4. 16** Hasil metrik evaluasi pada penggabungan fitur LBP dan lokasi jari

No	Metrik evaluasi	Rata-rata	Standar deviasi
1	Akurasi	86.06	0.010
2	Presisi	0.87	0.011
3	Recall	0.86	0.010
4	F1-Score	0.86	0.010

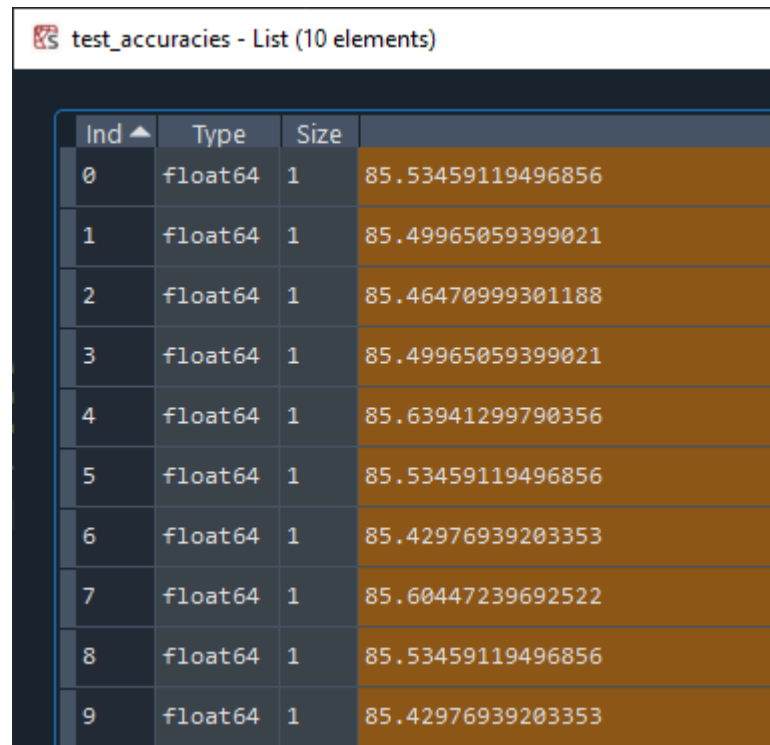
Dari metrik evaluasi yang diperoleh pada tabel 1.1, rata – rata akurasi sebesar 86.06% mengindikasikan bahwa model berhasil mengklasifikasikan sebagian besar dari total 667 sidik jari dengan benar. Jadi, sekitar 577 sidik jari dengan benar oleh model klasifikasi. Presisi sebesar 0.87 menunjukkan bahwa sekitar 87% dari semua sidik jari yang diprediksi sebagai milik satu jenis kelamin tertentu adalah benar-benar milik jenis kelamin tersebut. Recall, yang mencapai 0.86, menunjukkan bahwa model mampu mengidentifikasi sekitar 86% dari semua sidik jari yang sebenarnya milik jenis kelamin tertentu. F-1 Score, dengan nilai 0.86, menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall. Standar deviasi yang rendah (0.010 – 0.011) untuk setiap metrik evaluasi menunjukkan bahwa model memiliki konsistensi yang baik dalam performanya saat diuji pada data validasi. Ini menunjukkan bahwa model cenderung memberikan hasil yang stabil dan dapat diandalkan dalam mengklasifikasikan jenis kelamin berdasarkan citra sidik jari.





**Gambar 4. 31** Hasil sebaran akurasi pada pengujian data validasi pada penggabungan fitur LBP dan lokasi jari

Gambar 4.90 menunjukkan persebaran akurasi yang dihasilkan oleh percobaan ini. Nilai akurasi tertinggi yang dicapai adalah sebesar 87.43%, terjadi pada fold kelima dari proses validasi silang  $k=10$ , sementara nilai akurasi terendah sebesar 84.43% diamati pada fold ketiga. Perbedaan yang ditemukan antara nilai akurasi tertinggi dan terendah dari proses validasi adalah sekitar 3.00%, yang secara relatif kecil dalam evaluasi kinerja model. Hal ini menunjukkan bahwa model memiliki konsistensi yang cukup baik dalam memprediksi jenis kelamin berdasarkan sidik jari di seluruh fold validasi.



Ind	Type	Size	
0	float64	1	85.53459119496856
1	float64	1	85.49965059399021
2	float64	1	85.46470999301188
3	float64	1	85.49965059399021
4	float64	1	85.63941299790356
5	float64	1	85.53459119496856
6	float64	1	85.42976939203353
7	float64	1	85.60447239692522
8	float64	1	85.53459119496856
9	float64	1	85.42976939203353

**Gambar 4. 32** Hasil sebaran akurasi pada pengujian data testing pada penggabungan fitur LBP dan lokasi jari

Hasil pengujian terhadap data testing yang terdiri dari 2862 data sidik jari menunjukkan bahwa model klasifikasi sidik jari memiliki nilai rata-rata akurasi sebesar 85%. Ini adalah hasil yang signifikan dan menunjukkan bahwa model dapat memprediksi dengan benar dalam sebagian besar kasus. Selain itu, nilai akurasi ini tidak jauh berbeda dengan hasil yang diperoleh saat menguji data validasi, yang mengindikasikan bahwa performa model konsisten dan andal ketika dihadapkan pada data uji baru. Konsistensi ini memperkuat kepercayaan bahwa model tidak hanya terlatih dengan baik pada data yang ada tetapi juga memiliki generalisasi yang baik untuk data yang belum pernah dilihat sebelumnya.

**Tabel 4. 17** Hasil *confusion matrix* pada penggabungan fitur LBP dan lokasi jari

		TP	TN	FP	FN
Fold-1	Pria	330	251	10	77
	Wanita	251	330	77	10
Fold-2	Pria	327	252	15	74
	Wanita	252	327	74	15

Fold-3	Pria	306	258	22	82
	Wanita	258	306	82	22
Fold-4	Pria	299	282	13	74
	Wanita	282	299	74	13
Fold-5	Pria	304	280	11	73
	Wanita	280	304	73	11
Fold-6	Pria	332	245	11	80
	Wanita	245	332	80	11
Fold-7	Pria	306	262	16	84
	Wanita	262	306	84	16
Fold-8	Pria	314	262	22	70
	Wanita	262	314	70	22
Fold-9	Pria	317	252	20	78
	Wanita	252	317	78	20
Fold-10	Pria	316	252	21	78
	Wanita	252	316	78	21

Dari hasil confusion matrix untuk penggabungan fitur LBP dan lokasi jari, terlihat bahwa model klasifikasi menunjukkan kinerja yang cukup baik dalam membedakan antara pria dan wanita. Dalam kesepuluh fold evaluasi, jumlah True Positive (TP) dan True Negative (TN) relatif tinggi, sementara jumlah False Positive (FP) dan False Negative (FN) cenderung rendah.

Hal ini menunjukkan bahwa model dapat mengidentifikasi kedua kategori dengan akurasi yang baik dan memiliki kemampuan dalam mengurangi kesalahan prediksi. Meskipun terdapat fluktuasi dalam jumlah FP dan FN antara fold-fold, kinerja model secara keseluruhan masih cukup konsisten.

Penggabungan fitur LBP dan lokasi jari sepertinya merupakan pendekatan yang efektif dalam tugas klasifikasi ini, dengan model mampu memanfaatkan informasi dari kedua jenis fitur untuk melakukan prediksi dengan baik. Namun, evaluasi lebih lanjut mungkin diperlukan untuk memvalidasi kinerja model pada dataset yang lebih besar dan untuk mengevaluasi generalisasi model ini dalam skenario penggunaan yang berbeda.

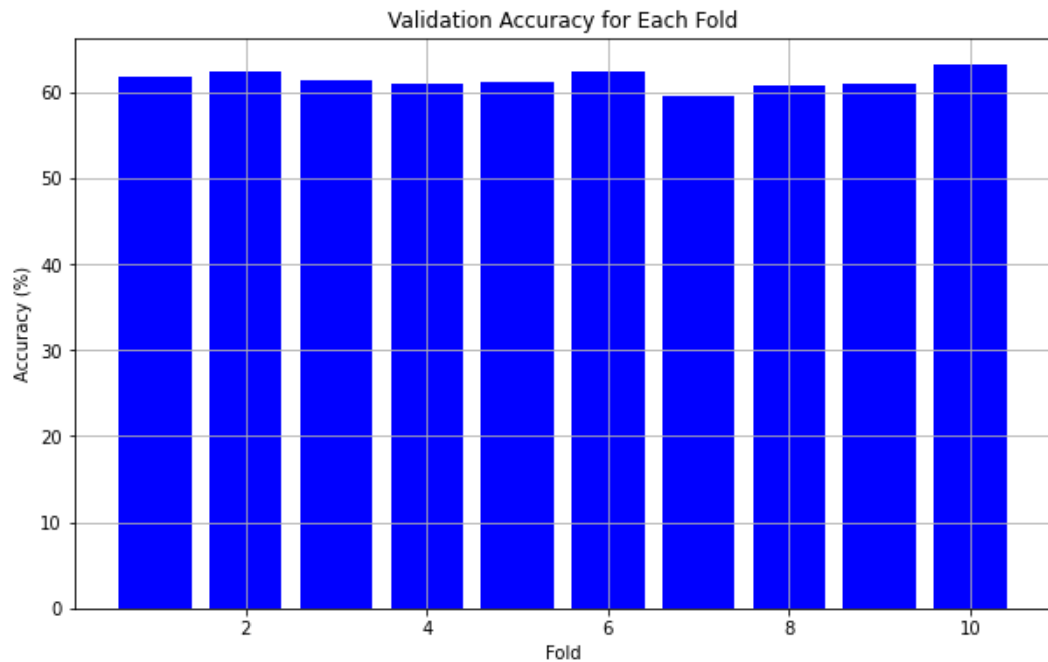
### 3.d Percobana penggabungan fitur *ridge density* dan lokasi jari

Hasil analisis dari percobaan penggabungan ridge density dan lokasi jari pada model yang di bangun pada percobaan ini menunjukkan kinerja yang sangat kurang baik di bandingkan percobaan sebelumnya. Tabel 4.18 menunjukkan nilai rata-rata metrik evaluasi yang di peroleh oleh pada percobaan ini.

**Tabel 4. 18** hasil metrik evaluasi pada penggabungan fitur ridge density dan lokasi jari

No	Metrik evaluasi	Rata-rata	Standar deviasi
1	Akurasi	61.41	0.010
2	Presisi	0.61	0.010
3	Recall	0.61	0.010
4	F1-Score	0.61	0.010

Dari tabel 4.18, akurasi rata-rata sebesar 61.41% menunjukkan bahwa model hanya mampu memprediksi setengah sidik jari dengan benar, menandakan hasil akurasi yang rendah. Sehingga, percobaan ini merupakan yang memiliki akurasi terendah. Jadi, sekitar 409 sidik jari dengan benar oleh model klasifikasi. Presisi sebesar 0.61 menunjukkan bahwa sekitar 61% dari semua sidik jari yang diprediksi sebagai milik satu jenis kelamin tertentu adalah benar-benar milik jenis kelamin tersebut. Recall, yang mencapai 0.61, menunjukkan bahwa model mampu mengidentifikasi sekitar 61% dari semua sidik jari yang sebenarnya milik jenis kelamin tertentu. F-1 Score, dengan nilai 0.61, menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi dan recall. Standar deviasi yang rendah (0.010) untuk setiap metrik evaluasi menunjukkan bahwa model memiliki konsistensi yang baik dalam performanya saat diuji pada data validasi. Ini menunjukkan bahwa model cenderung memberikan hasil yang stabil dalam mengklasifikasikan jenis kelamin berdasarkan citra sidik jari.



**Gambar 4. 33** Hasil sebaran akurasi pada pengujian data validasi pada penggabungan fitur ridge density dan lokasi jari

Gambar 4.102 menunjukkan persebaran akurasi yang dihasilkan oleh percobaan ini. Nilai akurasi tertinggi yang dicapai adalah sebesar 63.12%, terjadi pada fold kesepuluh dari proses validasi silang  $k=10$ , sementara nilai akurasi terendah sebesar 59.43% diamati pada fold ketujuh. Perbedaan yang ditemukan antara nilai akurasi tertinggi dan terendah dari proses validasi adalah sekitar 3.69%, yang relatif besar dalam evaluasi kinerja model. Hal ini menunjukkan bahwa terdapat variasi yang signifikan dalam performa model pada masing-masing fold validasi.

**Tabel 4. 19** Hasil *confusion matrix* pada penggabungan fitur ridge density dan lokasi jari

		TP	TN	FP	FN
Fold-1	Pria	207	206	133	122
	Wanita	206	207	122	133
Fold-2	Pria	203	213	139	113
	Wanita	213	203	113	139
Fold-3	Pria	196	213	132	127
	Wanita	213	196	127	132
Fold-4	Pria	185	222	127	134
	Wanita	222	185	134	127
Fold-5	Pria	185	223	130	130
	Wanita	223	185	130	130

Fold-6	Pria	208	209	135	116
	Wanita	209	208	116	135
Fold-7	Pria	188	209	134	137
	Wanita	209	188	137	134
Fold-8	Pria	196	210	140	122
	Wanita	210	196	122	140
Fold-9	Pria	205	202	132	128
	Wanita	202	205	128	132
Fold-10	Pria	215	206	122	124
	Wanita	206	215	124	122

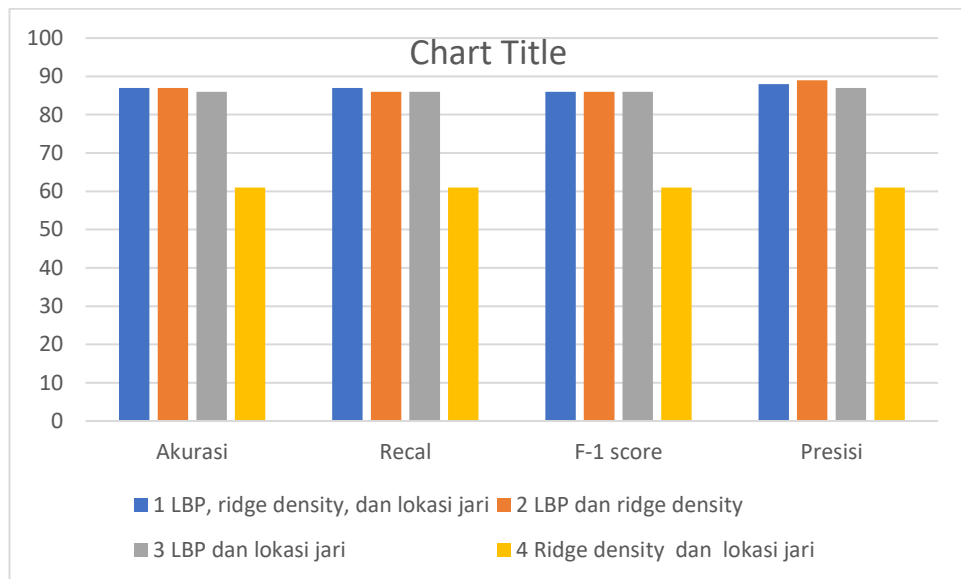
Dari hasil confusion matrix untuk penggabungan fitur ridge density dan lokasi jari, terlihat bahwa model klasifikasi menunjukkan kinerja yang bervariasi dalam membedakan antara pria dan wanita. Terdapat fluktuasi dalam jumlah True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN) antara fold-fold evaluasi.

Dalam beberapa fold, terlihat bahwa model cenderung memiliki kesulitan dalam mengklasifikasikan kedua kategori dengan akurat, dengan jumlah FP dan FN yang cukup tinggi. Hal ini menunjukkan adanya potensi untuk meningkatkan kinerja model dalam mengurangi kesalahan prediksi.

Meskipun demikian, terlihat bahwa terdapat fold-fold tertentu di mana model menunjukkan kinerja yang lebih baik, dengan jumlah TP dan TN yang lebih tinggi dan jumlah FP dan FN yang lebih rendah. Hal ini menunjukkan bahwa penggabungan fitur ridge density dan lokasi jari memiliki potensi sebagai pendekatan yang efektif dalam membedakan antara pria dan wanita, namun perlu dilakukan evaluasi lebih lanjut untuk memahami faktor-faktor yang mempengaruhi fluktuasi dalam kinerja model.

### **Kesimpulan Pengujian terhadap 4770 sidik jari laki-laki dan 4770 sidik jari wanita**

Untuk melihat kinerja metrik evaluasi secara menyeluruh dari keembatan percobaan penggabungan fitur pada percobaan pembangunan model ketiga pada penelitian ini, dapat dilihat melalui Grafik 4.2.



**Grafik 4. 2 Nilai rata – rata metrik evaluasi pada uji data validasi**

Secara menyeluruh hasil analisis pada ke-4 percobaan penggabungan fitur dengan metode k-fold validation dalam melatih model untuk melakukan klasifikasi sidik jari dapat di rangkum yaitu jika kombinasi LBP, *ridge density*, dan lokasi jari menghasilkan performa yang tidak jauh dengan kombinasi LBP dan *ridge density*. Sehingga dapat di simpulkan jika fitur lokasi jari tidak berpengaruh secara signifikan pada percobaan ini. Fitur lokasi jari membantu dalam meningkatkan rata – rata akurasi model sebanyak 0.15%, fitur *ridge density* membantu dalam meningkatkan rata – rata akurasi model sebanyak 0.58%, dan fitur LBP membantu dalam meningkatkan rata – rata akurasi model secara signifikan yaitu sebanyak 25.23%. Dari hasil tersebut menunjukkan jika fitur yang paling krusial dalam meningkatkan akurasi model dalam setiap percobaan adalah LBP, hal ini di karenakan atribut fitur yang di hasilkan lebih banyak ketimbang dengan fitur *ridge density* dan lokasi jari.

Ekstraksi fitur LBP, *ridge density*, dan lokasi jari mempunyai tingkat akurasi yang sama dengan ekstraksi fitur LBP dan *ridge density*, merupakan metode identifikasi sidik jari yang paling akurat dengan akurasi 87%, Nilai recall Ekstraksi fitur LBP, *ridge density*, dan lokasi jari merupakan tertinggi dari hasil

penggabungan fitur lainnya dengan nilai 87% yang mengartikan jika model yang di bangun dengan ekstraksi ini mampu mengidentifikasi 87% sidik jari dengan benar, nilai F-1 score terendah di peroleh pada ekstraksi fitur *ridge density* dan lokasi jari yaitu sebesar 69%, dan nilai presisi tertinggi di peroleh pada metode ekstraksi LBP dan *ridge density* yaitu sebesar 89%. Nilai presisi 89% menandakan bahwa dari 668 data yang diidentifikasi positif oleh metode tersebut pada data validasi, Dari 668, sebanyak 597 data benar-benar merupakan sidik jari yang sesuai. Dengan kata lain, metode tersebut memiliki tingkat kepastian yang tinggi dalam mengidentifikasi sidik jari secara positif atau nilai TP yang dihasilkan tiap kelas besar. Penggabungan fitur *ridge density* dan lokasi jari memperoleh performa yang paling rendah di dibandingkan yang lain dengan perolehan akurasi, presisi, recall, dan F-1 score sebesar 61% hal ini menunjukkan sebanyak 39% data salah di prediksi oleh model, hal ini terjadi karena metode ini menghasilkan nilai vector features yang paling sedikit. Sedangkan ketiga metode lainnya memberikan hasil peforma di atas 80% untuk semua metrik evaluasi dengan perbedaan satu sama lain yang tidak terlalu signifikan.

#### 4.2.4 Kesimpulan menyeluruh dari ketiga percobaan

**Tabel 4. 20 Jumlah vector features pada tiap percobaan**

No	Fitur	Jumlah vector features
1	LBP, <i>ridge density</i> , dan lokasi jari	13
2	LBP dan <i>ridge density</i>	11
3	LBP dan lokasi jari	12
4	<i>Ridge density</i> dan lokasi jari	3

Berdasarkan ketiga percobaan yang telah di lakukan, di dapati jika semakin banyak perbedaan pada jumlah sample data tiap kelas dapat mempengaruhi bias pada hasil uji performa model. Hal ini terlihat pada percobaan pertama ketika terjadi perbedaan jumlah sample kelas wanita dengan kelas pria sebanyak 3520 citra sidik jari kelas pria lebih banyak, menghasilkan model yang overfitting, dimana model tidak dapat mengenali pola citra sidik jari kelas wanita, sehingga model hanya



mengetahui pola kelas pria. Berdasarkan hasil metrik evaluasi yang di hasilkan pada percobaan ini, di dapati jika model selalu memprediksi data validasi sebagai pria, yang menyebabkan nilai FP dan TN pada kelas pria 0 dan nilai TP,TN,FP,FN pada kelas wanita 0.

Untuk mengatasi masalah ini, percobaan kedua di lakukan untuk menyeimbangkan sample data pada tiap kelas dengan mengambil sample kelas wanita dan kelas pria secara seimbang. Hasil yang di peroleh pada percobaan ini naik secara signifikan dalam reabilitas datanya, dimana model sudah dapat mengenali sample kelas wanita. Namun pada percobaan ini di temui penemuan dimana jumlah vector features yang di gunakan dalam membangun model sangat mempengaruhi hasil kinerja model, dimana vector features paling sedikit yaitu pada percobaan penggabungan fitur *ridge density* dan lokasi jari dan model dengan metrik evalusi paling optimal terjadi pada percobaan penggabungan ketiga fitur di mana hasil akurasi rata – rata yang dihasilkan sebesar 69%, naik +4% dari nilai akurasi penggabungan fitur *ridge density* dan lokasi jari yang hanya memperoleh nilai rata – rata akurasi sebesar 64%.

Percobaan 3 di lakukan untuk mencari tahu bagaimana pengaruh jumlah sample data pada tiap kelas mempengaruhi tiap peforma, sehingga percobaan ke-3 menambahkan sebanyak 3520 sample data pada tiap kelas. Hasil metrik evaluasi yang di hasilkan meningkat secara signifikan di mana terjadi kenaikan akurasi sebanyak 18%, dengan perolehan akurasi tertinggi sebesar 87%. Percobaan ini menunjukkan jika semakin banyak data sample pada tiap kelas, model semakin dapat mengenali pola – pola yang ada pada tiap kelas. Secara keseluruhan model yang di latih cenderung melakukan prediksi secara benar pada kelas laki – laki yang dapat di lihat pada tiap nilai confusion metric dari tiap percobaan.

## Bab V

### Kesimpulan

#### 5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan terdapat beberapa kesimpulan yang dapat diambil yaitu sebagai berikut:

1. Banyaknya sample data pada tiap kelas sangat memengaruhi hasil akurasi model secara signifikan dan memberi hasil yang signifikan dalam memperkaya pembelajaran pada model, terlihat pada hasil percobaan 1 sampai percobaan tiga yang terus mengalami kenaikan nilai metrik evaluasi selama proses penambahan sample data pada tiap kelas.
2. Menyeimbangkan jumlah sampel antara kelas wanita dan pria sangat penting untuk menghindari bias dan meningkatkan kemampuan model dalam mengenali pola dari kedua kelas. Hal ini terbukti ketika model pada percobaan 1 tidak mampu mengenali sample kelas wanita di karena terjadinya perbedaan jumlah sample sebanyak 20.5%. Namun ketika di seimbangkan pada percobaan 2 model sudah dapat mengenali sample kelas wanita.
3. Menggunakan dataset yang lebih besar atau melakukan augmentasi data dapat membantu model untuk belajar lebih baik dan meningkatkan kemampuannya dalam menggeneralisasi data baru. Hal ini terbukti ketika model pada percobaan 2 hanya dapat menghasilkan rata – rata akurasi tertinggi sebesar 69%, namun ketika percobaan 3 di lakukan dengan menambahkan data augmentasi dari folder *altered easy*, akurasi yang di hasilkan naik sebesar 18%.
4. Menggunakan kombinasi fitur yang lebih lengkap (LBP, ridge density, dan lokasi jari) memberikan performa terbaik. Jumlah dan jenis fitur vektor yang digunakan sangat mempengaruhi hasil kinerja model. Hal ini di buktikan pada percobaan ke 2 dan ke 3, di mana penggabungan fitur yang menghasilkan *vector features* terendah yaitu penggabungan

fitur *ridge density* dan lokasi jari menghasilkan nilai metrik evaluasi terendah pada setiap percobaan.

## 5.2 Saran

Berdasarkan hasil yang telah di peroleh oleh peneliti selama melakukan penelitian ini, terdapat beberapa saran yang dapat di ikuti untuk meningkatkan hasil peforma dari model, antara lain adalah sebagai berikut:

1. Dapat menambahkan sample data tiap kelas dari dataset lainnya untuk memastikan varasi pola dapat di kenali dengan baik oleh model saat dilatih dengan jumlah data yang besar dan seimbang
2. Memastikan jika jumlah sample data antara kelas pria dan wanita seimbang.
3. Dapat melakukan augmentasi data untuk meningkatkan jumlah dan variasi data yang tersedia
4. Disarankan untuk menggunakan dataset yang seimbang antara kelas pria dan kelas wanita minimal sebanyak 4770 data untuk tiap kelas
5. Menambahkan fitur baru untuk menghasilkan pola yang lebih unik untuk setiap sidik jari

## DAFTAR PUSTAKA

- Abdi, M. F., Kusrini, & Kurniawan, M. P. (2022). LOCAL BINARY PATTERN UNTUK EKTRAKSI TEKSTUR GAMBAR WAJAH MENGGUNAKAN MASKER DAN TANPA MASKER. *Technologia*, 1. Retrieved from <https://ojs.uniska-bjm.ac.id/index.php/JIT/article/view/6275>
- Abdullah, S. F., Rahman, A. F., & Abas, Z. A. (2015). CLASSIFICATION OF GENDER BY USING FINGERPRINT RIDGE DENSITY IN NORTHERN PART OF MALAYSIA. *ARPJ Journal of Engineering and*

- Applied Sciences*, 3. Retrieved from  
[https://www.researchgate.net/publication/290495401\\_Classification\\_of\\_gender\\_by\\_using\\_fingerprint\\_ridge\\_density\\_in\\_northern\\_part\\_of\\_Malaysia](https://www.researchgate.net/publication/290495401_Classification_of_gender_by_using_fingerprint_ridge_density_in_northern_part_of_Malaysia)
- Andrés, A. S., Barea, J. A., Rivaldería, N., Rodríguez, C. A., & Redomero, E. G. (2018). Impact of aging on fingerprint ridge density: Anthropometry and forensic implications in sex inference. *Science & Justice*, 1. Retrieved from  
[https://www.researchgate.net/publication/324934309\\_Impact\\_of\\_aging\\_on\\_fingerprint\\_ridge\\_density\\_Anthropometry\\_and\\_forensic\\_implications\\_in\\_sex\\_inference](https://www.researchgate.net/publication/324934309_Impact_of_aging_on_fingerprint_ridge_density_Anthropometry_and_forensic_implications_in_sex_inference)
- Asamoah, D., Oppong, E. O., Oppong, S. O., & Danso, J. (2018). Measuring the Performance of Image Contrast Enhancement Technique. *International Journal of Computer Applications*, 5.
- Burkov, A. (2019). *The Hundred-Page Machine Learning Book*. Andriy Burkov.
- ÇEVİK, A., KURTOĞLU, A. E., & BİLGEHAN, M. (2015). SUPPORT VECTOR MACHINES IN STRUCTURAL ENGINEERING: A REVIEW. *JOURNAL OF CIVIL ENGINEERING AND MANAGEMENT*. Retrieved from  
[https://www.researchgate.net/publication/274265161\\_Support\\_vector\\_machines\\_in\\_structural\\_engineering\\_A\\_review](https://www.researchgate.net/publication/274265161_Support_vector_machines_in_structural_engineering_A_review)
- Chauhan, A., & Shukla, S. (2017). Feasibility of Ridge Density: A Comparative Study of Fingerprint Ridge Densities among Different Indian Population. *Journal of Forensic Sciences & Criminal Investigation*. Retrieved from  
<https://juniperpublishers.com/jfsci/pdf/JFSCI.MS.ID.555654.pdf>
- Gornale, S. S., M, B., & R, K. (2017). fingerprint based gender classification using local binary patterns. *International Journal of Computational Intelligence Research*. Retrieved from  
[https://www.ripublication.com/ijcir17/ijcirv13n2\\_09.pdf](https://www.ripublication.com/ijcir17/ijcirv13n2_09.pdf)
- Hambal, A. M., Pei, D. Z., & Ishabailu, F. L. (2015). Image Noise Reduction and Filtering Techniques . *International Journal of Science and Research (IJSR)* , 1.

- Khouma, O., Diop, I., Fall, P. A., Ndiaye, M. L., & Farssi, S. M. (2019). Novel Classification Method of Spikes Morphology in EEG Signal Using Machine Learning. *Second International Conference on Intelligent Computing in Data Sciences (ICDS 2018)*, 3-5. Retrieved from [https://www.researchgate.net/publication/331308937\\_Novel\\_Classification\\_Method\\_of\\_Spikes\\_Morphology\\_in\\_EEG\\_Signal\\_Using\\_Machine\\_Learning](https://www.researchgate.net/publication/331308937_Novel_Classification_Method_of_Spikes_Morphology_in_EEG_Signal_Using_Machine_Learning)
- Kruthi, Patil, A., & Gornale, S. (2019). Fusion of Local Binary Pattern and Local Phase Quantization features set for Gender Classification using Fingerprints. *International Journal of Computer Sciences and Engineering*. Retrieved from [https://www.researchgate.net/publication/331703416\\_Fusion\\_of\\_Local\\_Binary\\_Pattern\\_and\\_Local\\_Phase\\_Quantization\\_features\\_set\\_for\\_Gender\\_Classification\\_using\\_Fingerprints](https://www.researchgate.net/publication/331703416_Fusion_of_Local_Binary_Pattern_and_Local_Phase_Quantization_features_set_for_Gender_Classification_using_Fingerprints)
- Kumar, N., & Verma, P. (2012). FINGERPRINT IMAGE ENHANCEMENT AND MINUTIA. *International Journal of Engineering Sciences & Emerging Technologies*.
- Lubanjaja, F. R., Fitri, E., Ardiansyah, Junaidi, A., & Prabowo, R. (2021). Abstract Classification Using Support Vector Machine Algorithm (Case Study: Abstract in a Computer Science Journal). *Journal of Physics: Conference Series*, 9. Retrieved from <https://iopscience.iop.org/article/10.1088/1742-6596/1751/1/012042/pdf>
- Prakasa, E. (2015). Ekstraksi Ciri Tekstur dengan Menggunakan Local Binary Pattern. *Pusat Penelitian Informatika, Lembaga Ilmu Pengetahuan Indonesia (INKOM)*. Retrieved from [https://www.researchgate.net/publication/305152373\\_Texture\\_Feature\\_Extraction\\_by\\_Using\\_Local\\_Binary\\_Pattern](https://www.researchgate.net/publication/305152373_Texture_Feature_Extraction_by_Using_Local_Binary_Pattern)
- Retnoningrum, D., Widodo, A. W., & Rahman, A. M. (2019). Ekstraksi Ciri Pada Telapak Tangan Dengan Metode Local Binary Pattern (LBP). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*. Retrieved from <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4772/2207>

- Saddami, K., Munadi, K., Yuwaldi, A., & Arnia, F. (2019). Improvement of binarization performance using local otsu thresholding. *International Journal of Electrical and Computer Engineering (IJECE)*, 3. Retrieved from [https://www.researchgate.net/profile/Khairun-Saddami/publication/328715979\\_Improvement\\_of\\_Binarization\\_Performance\\_using\\_Local\\_Otsu\\_Thresholding/links/5e7c6a4b92851caef49da0c7/Improvement-of-Binarization-Performance-using-Local-Otsu-Thresholding.pdf](https://www.researchgate.net/profile/Khairun-Saddami/publication/328715979_Improvement_of_Binarization_Performance_using_Local_Otsu_Thresholding/links/5e7c6a4b92851caef49da0c7/Improvement-of-Binarization-Performance-using-Local-Otsu-Thresholding.pdf)
- Salau, A. O., & Jain, S. (2019). Feature Extraction: A Survey of the Types, Techniques, Applications. *IEEE*. Retrieved from [https://www.researchgate.net/publication/338366329\\_Feature\\_Extraction\\_A\\_Survey\\_of\\_the\\_Types\\_Techniques\\_Applications](https://www.researchgate.net/publication/338366329_Feature_Extraction_A_Survey_of_the_Types_Techniques_Applications)
- Salsabila, H., Rachmawati, E., & Sthevanie, F. (2021). Klasifikasi Gender Berdasarkan Citra Wajah Menggunakan Metode Local Binary Pattern dan K-Nearest Neighbor. *Jurnal Tugas Akhir Fakultas Informatika*, 1. Retrieved from <https://repository.telkomuniversity.ac.id/pustaka/167771/klasifikasi-gender-berdasarkan-citra-wajah-menggunakan-metode-local-binary-pattern-dan-k-nearest-neighbor.html>
- Sharma, A., & Chawla, M. (2016). Fingerprint matching Using Minutiae Extraction Techniques. *Journal of Advances in Electrical Devices*, 2-4. Retrieved from <https://core.ac.uk/download/pdf/230495427.pdf>
- Sharma, R., & Abrol, P. (2020). Image Feature Extraction Techniques. *International Journal of Scientific and Technical Advancements (IJSTA)*, 1. Retrieved from [https://www.researchgate.net/publication/350975404\\_Image\\_feature\\_extraction\\_techniques](https://www.researchgate.net/publication/350975404_Image_feature_extraction_techniques)
- Shehu, Y. I., Garcia, A. R., Palade, V., & James, A. (2022). Sokoto Coventry Fingerprint Dataset. *arxiv*.
- Yuwono, B. (2010). IMAGE SMOOTHING MENGGUNAKAN MEAN FILTERING, MEDIAN FILTERING, MODUS FILTERING DAN

GAUSSIAN FILTERING. *TELEMATIKA*, 2. Retrieved from

<http://jurnal.upnyk.ac.id/index.php/telematika/article/view/416/377>

Zhang, X., Ren, Y., Zhen, G., Shan, Y., & Chu, C. (2023). A color image contrast enhancement method. *PLOS ONE*, 1.

## LAMPIRAN A

### KODE SUMBER PROGRAM

```
import matplotlib.pyplot as plt
import fingerprint_enhancer
import numpy as np
import os
import cv2
from localbinarypatterns import LocalBinaryPatterns
from ridgedensity import RidgeDensityCalculator
from sklearn.svm import LinearSVC
from skimage import io, color, filters
from skimage.morphology import skeletonize
from skimage.color import rgba2rgb
from skimage import exposure
from sklearn.model_selection import train_test_split, KFold
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, classification_report,
f1_score, precision_score, recall_score
import seaborn as sns

def convert_to_grayscale_and_enchance_image(image_path):
    img = io.imread(image_path)

    if img.ndim == 3 and img.shape[2] == 3: # RGB image
        gray_img = color.rgb2gray(img)
        print("123412421")
    elif img.ndim == 3 and img.shape[2] == 4: # RGBA image
        gray_img = color.rgb2gray(color.rgb2rgb(img))
        print("3333")
    elif img.ndim == 2: # Grayscale image
        gray_img = img
        print("12")
    else:
        raise ValueError("Unsupported image format")
```

```

    ## Resize the image
    scaling_factor = 5
    resized_img = cv2.resize(gray_img, None, fx=scaling_factor,
fy=scaling_factor)
    print("keren")
    print(image_path)

    # Apply local thresholding
    local_binary = filters.threshold_local(resized_img,
block_size=55, method='gaussian')

    # Binarize the image using local threshold
    binarized_img = resized_img > local_binary

    # Enhance image
    out = fingerprint_enhancer.enhance_Fingerprint(binarized_img)
    out[out == 255] = 1 # Replace 255 with 1
    skeleton = skeletonize(out)
    inverted_skeleton = np.invert(skeleton)

    return inverted_skeleton

def image_preprocessing_for_LBP(image_path):
    img = io.imread(image_path)
    print(image_path)

    if img.ndim == 3 and img.shape[2] == 3: # RGB image
        gray_img = color.rgb2gray(img)
    elif img.ndim == 3 and img.shape[2] == 4: # RGBA image
        gray_img = color.rgb2gray(color rgba2rgb(img))
    elif img.ndim == 2: # Grayscale image
        gray_img = img
    else:
        raise ValueError("Unsupported image format")

    ## Resize the image
    scaling_factor = 5
    resized_img = cv2.resize(gray_img, None, fx=scaling_factor,
fy=scaling_factor)

    # Apply noise reduction using Gaussian filter
    blurred_img = filters.gaussian(resized_img, sigma=1)

```



```

    # Apply histogram equalization for contrast enhancement
    equalized_img = exposure.equalize_hist(blurred_img)

    return equalized_img

def get_label_from_filename(file_path):
    # Split the file path based on the directory separator
    parts = file_path.split('\\')

    # Get the last part containing the filename
    filename = parts[-1]

    # Split the filename based on double underscore
    label = filename.split('__')[1][0]
    return 0 if label == "F" else 1

def store_ridge_density_features(arr_file):
    array_ridge_density = []

    for i in range(0, len(arr_file)):
        convert_to_int = (1 - arr_file[i]).astype(np.uint8) * 255
        area = np.prod(arr_file[i].shape)
        ridge_count = sum(255 for row in convert_to_int for pixel in
row if pixel == 255)
        ridge_density = RidgeDensityCalculator(ridge_count, area)
        count_ridge_density =
ridge_density.calculate_ridge_density()
        array_ridge_density.append(count_ridge_density)
    return array_ridge_density

def store_LBP_features(arr_file):
    array_lbp = []

    for i in range(0, len(arr_file)):
        desc = LocalBinaryPatterns(8, 1)
        histogram = desc.describe(arr_file[i])
        array_lbp.append(histogram)

    return array_lbp

def store_hand(arr_file):
    temp_array = []

```

```

for i in arr_file:
    path = os.path.basename(i)
    file = path.split("_")
    array = []
    location = file[4]

    #array.append(file[0])

    if location == "thumb":
        array.append(0)
        if file[3] == "Left":
            array.append(0)
        elif file[3] == "Right":
            array.append(1)
    elif location == "middle":
        array.append(1)
        if file[3] == "Left":
            array.append(0)
        elif file[3] == "Right":
            array.append(1)
    elif location == "ring":
        array.append(2)
        if file[3] == "Left":
            array.append(0)
        elif file[3] == "Right":
            array.append(1)
    elif location == "index":
        array.append(3)
        if file[3] == "Left":
            array.append(0)
        elif file[3] == "Right":
            array.append(1)
    else:
        array.append(4)
        if file[3] == "Left":
            array.append(0)
        elif file[3] == "Right":
            array.append(1)
    temp_array.append(array)
return temp_array

if __name__ == '__main__':

```

```

    #Directory containing 6000 fingerprint image - Percobaan 1
    images_directory = r'D:\SEMESTER 7\RTI\Sidik
Jari\SOCOFing\SOCOFing\Real'

    #Directory containing equal 1230 fingerprint image - Percobaan 2
    #images_directory = r'D:\SEMESTER 8\ngoding\equal_image'

    #Directory containing equal 4770 fingerprint image - Percobaan 3
    #images_directory = r'D:\SEMESTER 7\RTI\Sidik
Jari\SOCOFing\SOCOFing\TRAIN'

    #Image pre-processing

    image_files = sorted([os.path.join(images_directory, f) for f in
os.listdir(images_directory) if
os.path.isfile(os.path.join(images_directory, f))])

    images_test_directory = r'D:\SEMESTER 8\ngoding\test'

    image_labels = {filename: get_label_from_filename(filename) for
filename in image_files}
    labels = [label for label in image_labels.values()]

    print(labels.count(0))
    print(labels.count(1))

    images_for_LBP = []
    images_for_ridge_density = []

    # Process each image file for LBP
    for image_file in image_files:
        processed_image = image_preprocessing_for_LBP(image_file)
        images_for_LBP.append(processed_image)

    print("Pre-processing LBP Image done")

    # Process each image file for ridge density
    for image_file in image_files:

```

```

        processed_image =
convert_to_grayscale_and_enhance_image(image_file)
        images_for_ridge_density.append(processed_image)

# Features extraction
arr_lbp_features = store_LBP_features(images_for_LBP)
arr_ridge_density_features =
store_ridge_density_features(images_for_ridge_density)
arr_hand_location = store_hand(image_files)

X_lbp = np.array(arr_lbp_features)
X_ridge_density = np.array(arr_ridge_density_features)
X_array_hand_location = np.array(arr_hand_location)
y = np.array(labels)

#Feature fussion
vector_features = []

for i in range(0,len(X_lbp)):
    #temp_array = np.concatenate((X_lbp[i],
[X_ridge_density[i]]))
    #temp_array = np.concatenate((X_lbp[i],
[X_ridge_density[i]],X_array_hand_location[i]))
    temp_array =
np.concatenate((X_lbp[i],X_array_hand_location[i]))
    #temp_array =
np.concatenate((X_ridge_density[i]],X_array_hand_location[i]))

    vector_features.append(temp_array)

vector_features = np.array(vector_features)
labels = np.array(labels)

#Build the model
# Split dataset into training and test set
X_train, X_test, y_train, y_test =
train_test_split(vector_features, labels, test_size=0.3,
random_state=100)

```

```

    # Initialize the SVM classifier
svm_linear = SVC(kernel='linear', C=100, gamma='auto')

    # Define k for k-fold cross-validation
k = 10
kf = KFold(n_splits=k, shuffle=True, random_state=100)

    # Store metrics for each fold
fold accuracies = []
precision_scores = []
recall_scores = []
f1_scores = []

test accuracies = []

train_vold_size = dict()
val_vold_size = dict()

    # Iterate through each fold and train the model
for fold, (train_index, val_index) in
enumerate(kf.split(X_train), 1):
    X_train_fold, X_val_fold = X_train[train_index],
X_train[val_index]
    y_train_fold, y_val_fold = y_train[train_index],
y_train[val_index]

    # Track the size of training and validation sets for each
fold
train_vold_size[fold] = len(train_index)
val_vold_size[fold] = len(val_index)
svm_linear.fit(X_train_fold, y_train_fold)

    # Calculate accuracy for the fold
val_accuracy = svm_linear.score(X_val_fold, y_val_fold) * 100
fold accuracies.append(val_accuracy)
print(f"Accuracy report - Fold {fold}: {val_accuracy}")

#test check
test_accuracy = svm_linear.score(X_test,y_test) * 100
test accuracies.append(test_accuracy)
y_pred_test = svm_linear.predict(X_test)

```

```

        precision =
precision_score(y_test,y_pred_test,zero_division=1)

        # Predict labels for the validation set
y_pred_val = svm_linear.predict(X_val_fold)

        # Generate confusion matrix for the validation set
cm = confusion_matrix(y_val_fold, y_pred_val)

        # Calculate additional metrics
precision = precision_score(y_val_fold, y_pred_val,
average='weighted',zero_division=1)
recall = recall_score(y_val_fold, y_pred_val,
average='weighted',zero_division=1)
f1 = f1_score(y_val_fold, y_pred_val,
average='weighted',zero_division=1)
precision_scores.append(precision)
recall_scores.append(recall)
f1_scores.append(f1)
print(f"Precision: {precision:.2f}, Recall: {recall:.2f}, F1-
Score: {f1:.2f}")

        # Plot confusion matrix
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title(f"Confusion Matrix - Fold {fold}")
plt.xticks([0.5, 1.5], ["female", "male"])
plt.yticks([0.5, 1.5], ["female", "male"])
plt.show()
plt.close()

        # Generate and print classification report
report = classification_report(y_val_fold, y_pred_val)
print(f"Classification Report - Fold {fold}:\n{report}")

        # Plot accuracy for each fold
plt.figure(figsize=(10, 6))
plt.bar(range(1, k + 1), fold accuracies, color='b')
plt.title('Validation Accuracy for Each Fold')
plt.xlabel('Fold')
plt.ylabel('Accuracy (%)')

```

```

plt.grid(True)
plt.show()
plt.close()

# Print overall metrics
print(f"Overall Accuracy: Rata-rata:
{np.mean(fold accuracies):.2f} ± Standar Deviasi:
{np.std(fold accuracies):.3f}")
print(f"Overall Precision: Rata-rata:
{np.mean(precision_scores):.2f} ± Standar Deviasi:
{np.std(precision_scores):.3f}")
print(f"Overall Recall: Rata-rata: {np.mean(recall_scores):.2f} ±
Standar Deviasi: {np.std(recall_scores):.3f}")
print(f"Overall F1-Score: Rata-rata: {np.mean(f1_scores):.2f} ±
Standar Deviasi: {np.std(f1_scores):.3f}")

```