

**KLASIFIKASI CITRA SIDIK JARI MENGGUNAKAN  
*CONVOLUTIONAL NEURAL NETWORK*  
BERDASARKAN TIPE PATTERN  
PADA SISTEM HENRY**

**SKRIPSI**

**TRIA RISKIANI**

**171402016**



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

**KLASIFIKASI CITRA SIDIK JARI MENGGUNAKAN *CONVOLUTIONAL  
NEURAL NETWORK* BERDASARKAN TIPE PATTERN  
PADA SISTEM HENRY**

**SKRIPSI**

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah  
Sarjana Teknologi Informasi

**TRIA RISKIANI**

171402016



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2024**

## PERSETUJUAN

Judul : KLASIFIKASI CITRA SIDIK JARI  
           MENGGUNAKAN *CONVOLUTIONAL NEURAL  
           NETWORK* BERDASARKAN TIPE *PATTERN* PADA  
           SISTEM HENRY

Kategori : SKRIPSI

Nama Mahasiswa : TRIA RISKIANI

Nomor Induk Mahasiswa : 171402016

Program Studi : S1 TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI  
           UNIVERSITAS SUMATERA UTARA

Medan, 21 Mei 2024

Komisi Pembimbing:

Pembimbing 2,

Ivan Jaya, S.Si., M.Kom.  
           NIP. 198407072015041001

Pembimbing 1,

Dr. Erna Budhiarti Nababan, M.IT  
           NIP. 196210262017042001

Diketahui/disetujui oleh  
           Program Studi S1 Teknologi Informasi

Ketua,

Dedy Anisan, S.T., M.Kom.  
           NIP. 197908312009121002



## **PERNYATAAN**

**KLASIFIKASI CITRA SIDIK JARI MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK* BERDASARKAN TIPEPATTERN PADA SISTEM HENRY**

## **SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, Mei 2024

Tria Riskiani

171402016

## **UCAPAN TERIMA KASIH**

Ucapan terima kasih dan rasa syukur disampaikan kepada Tuhan yang Maha Esa, karena dengan izin-Nya penulis dapat menyelesaikan penyusunan skripsi ini untuk melengkapi syarat pencapaian gelar Sarjana Komputer dari Program Studi S1 Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara.

Skripsi ini mampu diselesaikan karena doa yang tak pernah berhenti, rasa cinta, dukungan dan kasih sayang dari keluarga untuk penulis, Bapak Suwito, Mama Ani, serta saudara/i penulis Mas Ferry dan Mbak Enni. Serta penghargaan yang tulus juga disampaikan untuk semua pihak yang terlibat dan memberikan bantuan berharga sepanjang perjalanan perkuliahan hingga penyelesaian skripsi ini. Penulis dengan tulus mengungkapkan rasa terima kasih yang mendalam kepada:

1. Bapak Prof. Dr. Muryanto Amin, S.Sos, M.Si. selaku Rektor Universitas Sumatera Utara.
2. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi.
3. Bapak Dedy Arisandi, S.T., M.Kom. selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
4. Ibu Dr. Erna Budhiarti Nababan, M.IT sebagai Dosen Pembimbing 1 dan Bapak Ivan Jaya, S.Si., M.Kom. sebagai Dosen Pembimbing 2, yang sudah memberikan arahan, saran dan bimbingan kepada penulis.
5. Dosen Pembanding 1 Bapak Dedy Arisandi, S.T., M.Kom dan Dosen Pembanding 2 Bapak Niskarto Zendrato, S.Kom., M.Kom yang sudah memberikan saran dan bimbingan kepada penulis.
6. Semua dosen serta staf pegawai akademik program studi S1 Teknologi Informasi.
7. Sahabat seperjuangan masa perkuliahan, Nurul Andini, Nadia Siti Namira, Allia Rania dan Miranda yang membantu, mendukung, memberikan semangat serta motivasi kepada penulis.

8. Teman dimasa perkuliahan yang senantiasa membantu, memberikan semangat kepada penulis, Melati, Destri, Jessi, Fani, Eka.
9. Sahabat penulis, Zakiyah Nelma Rahmi, Ika Febri Khairawan, MiaTri Handayani.
10. Teman-teman perkuliahan Angkatan 2017 S1 Teknologi Informasi.
11. Semua pihak yang ikut berperan dalam mendukung dan membantu penulis menyelesaikan skripsi ini.

Semoga Tuhan yang Maha Esa, Allah *Subbahanna wa ta'ala* memberikan rahmat-Nya dan membalas kebaikan semua pihak yang terlibat dalam penyelesaian skripsi ini.

Medan, Mei 2024

Tria Riskiani  
Penulis

## ABSTRAK

Sidik jari adalah salah satu bentuk biometrik yang digunakan untuk mengidentifikasi seseorang. Sidik jari memiliki susunan garis (*ridge*) pada setiap jari manusia yang unik dan tidak berubah seiring bertambahnya usia. Susunan garis pada sidik jari membentuk suatu pola yang unik dan berbeda. Sistem Henry merupakan sebuah sistem identifikasi sidik jari yang memanfaatkan pola atau *pattern* pada permukaan jari untuk memverifikasi identitas seseorang. Sebagai metode klasifikasi, *Henry's classification system* adalah metode klasifikasi sidik jari yang paling banyak digunakan. Terdapat lima jenis pola sidik jari yakni *Arch*, *Left Loop*, *Right Loop*, *Tented Arch* dan *Whrol*. Klasifikasi sidik jari menjadi bagian penting dalam sistem pengidentifikasian individu. Namun pengenalan pola sidik jari secara manual sulit dilakukan karena pola yang rumit sehingga bergantung pada kemampuan individu dan kurang efektif dari segi waktu. Hal ini tidak efektif dari segi waktu dan bergantung pada kemampuan individu. Penelitian ini dilakukan dengan tujuan untuk mengklasifikasi pola sidik jari menggunakan *Convolutional Neural Network*. Metode *Convolutional Neural Network* digunakan dalam penelitian ini karena mempunyai hasil yang paling signifikan dalam pengenalam citra. Teknik pengolahan citra yang diterapkan di penelitian ini adalah *contrast enhancement* dan *thresholding*. Setelah dilakukan pengujian pada data, metode ini mampu mengklasifikasi citra sidik jari ke dalam lima pola sidik jari dengan akurasi sebesar 85% dari data uji sebanyak 75 citra sidik jari.

**Kata Kunci:** Sidik Jari, Biometrik, Sistem Henry, *Arch*, *Left Loop*, *Right Loop*, *Tented Arch*, *Whrol*, *Contrast Enhancement*, *Thresholding*, *Convolutional Neural Network*

***FINGERPRINT IMAGE CLASSIFICATION USING  
CONVOLUTIONAL NEURAL NETWORK BASED  
ON PATTERN TYPE IN HENRY SYSTEM***

***ABSTRACT***

Fingerprint is one of the biometric forms used for identifying a person. Fingerprint has unique ridge arrangement on each human finger and do not change with age. The ridges on fingerprint create distinct and different patterns. The Henry system is a fingerprint identification system that utilizes the patterns on the surface of a finger to verify a person's identity. As a classification method, Henry's classification system is the most widely used for fingerprint classification. There are five types of fingerprint patterns: Arch, Left Loop, Right Loop, Tented Arch, and Whrol. Fingerprint classification is an important part of individual identification system. However, identifying fingerprint manually is difficult to do because the pattern is complicated and this depends on individual capabilities and also inefficient in terms of time. This research aims to classify fingerprint patterns using a Convolutional Neural Network. The Convolutional Neural Network method is used in this research because it has shown the most significant results in image recognition. The image processing techniques applied in this research include contrast enhancement and thresholding. After testing the data, this method is capable to classify fingerprint images into five fingerprint patterns with 85% accuracy from test dataset containing 75 fingerprint images.

**Keyword:** Fingerprint, Biometric, Henry System, Arch, Left Loop, Right Loop, Tented Arch, Whrol, Contrast Enhancement, Thresholding, Convolutional Neural Network

## DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB 1	
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2. Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4. Batasan Masalah	4
1.5. Manfaat penelitian	5
1.6 Metodologi Penelitian	5
1.7. Sistematika Penulisan	6
BAB 2	
LANDASAN TEORI	8
2.1 Sidik Jari	8
2.2. Pola Sidik Jari	9
2.3. Sistem Henry	10
2.4. Citra ( <i>Image</i> )	13

2.5. Pengolahan Citra Digital ( <i>Digital Image Processing</i> )	14
2.6. Convolutional Neural Network	16
2.7. Penelitian Terdahulu	20
<b>BAB 3</b>	
ANALISIS DAN PERANCANGAN SISTEM	24
3.1. Data yang digunakan	24
3.2. Analisis Sistem	25
3.3. Perancangan Antarmuka Sistem	38
3.4. Metode Evaluasi	41
<b>BAB 4</b>	
IMPLEMENTASI DAN PENGUJIAN	43
4.1 Implementasi Sistem	43
4.1.1. Spesifikasi perangkat keras dan perangkat lunak	43
4.1.2. Implementasi Perancangan Antarmuka	44
4.1.2.3. Tampilan halaman testing	45
4.2. Prosedur Operasional	49
4.3. Implementasi Model	53
4.4. Pengujian Sistem	55
4.5. Evaluasi	61
<b>BAB 5</b>	
KESIMPULAN DAN SARAN	66
5.1. Kesimpulan	66
5.2. Saran	66
DAFTAR PUSTAKA	68

## **DAFTAR TABEL**

Tabel 2.1. Penelitian Terdahulu	21
Tabel 2.2. Penelitian Terdahulu (Lanjutan)	22
Tabel 2.3. Penelitian Terdahulu (Lanjutan)	23
Tabel 3.1. Pembagian Dataset	27
Tabel 4.1. Percobaan dengan beberapa <i>epoch</i> dan <i>batch size</i>	54
Tabel 4.2. Hasil <i>Preprocessing</i> Citra	56
Tabel 4.3. Hasil <i>Preprocessing</i> Citra (Lanjutan)	57
Tabel 4.4. Hasil <i>Preprocessing</i> Citra (Lanjutan)	58
Tabel 4.5. Hasil Pengujian Citra pada Sistem	59
Tabel 4.6. Hasil Pengujian Citra pada Sistem (Lanjutan)	60
Tabel 4.7. Hasil Pengujian Citra pada Sistem (Lanjutan)	61
Tabel 4.8. Nilai <i>True Positive</i> , <i>False Positive</i> dan <i>False Negative</i>	62

## DAFTAR GAMBAR

Gambar 2.1. <i>Pattern area</i> (A) dan <i>radiant</i> atau <i>type line</i> (B)	9
Gambar 2.2. Tiga tipe utama sidik jari (a) <i>arch</i> , (b) <i>loop</i> , (c) <i>whorl</i> .	10
Gambar 2.3. Pola sidik jari <i>arch</i>	11
Gambar 2.4. Pola sidik jari <i>tented arch</i>	11
Gambar 2.5. Pola sidik jari <i>right loop</i>	12
Gambar 2.6. Pola sidik jari <i>left loop</i>	12
Gambar 2.7. Pola sidik jari <i>whorl</i>	13
Gambar 2.8. Citra Keabuan	14
Gambar 2.9. Arsitektur <i>Convolutional Neural Network</i>	17
Gambar 2.10. Operasi Konvolusi	17
Gambar 2.11. Operasi <i>Max Pooling</i>	19
Gambar 2.12. Operasi <i>Fully Connected Layer</i>	19
Gambar 3.1. Contoh Data Sidik Jari	24
Gambar 3.2. Arsitektur Umum	26
Gambar 3.3. Nilai Piksel Citra	27
Gambar 3.4. Perbedaan Histogram citra sebelum dan setelah diterapkan CLAHE	28
Gambar 3.5. Citra pola sidik jari sebelum dan setelah diterapkan metode CLAHE	29
Gambar 3.6. Nilai <i>grayscale</i> pada setiap piksel	29
Gambar 3.7. Nilai citra dan citra setelah dilakukan <i>thresholding</i>	30
Gambar 3.8. Nilai piksel citra masukan	31
Gambar 3.9. Nilai piksel citra masukan dengan <i>padding</i> 0 dan <i>kernel</i> 3 x 3	32

Gambar 3.10. <i>Feature Detector</i> dengan kernel size 3x3	32
Gambar 3.11. Proses pergeseran dengan <i>stride</i> 1 dan <i>kernel</i> 3 x 3	33
Gambar 3.12. Proses konvolusi pada pergeseran pertama	33
Gambar 3.13. Proses konvolusi pada semua piksel	34
Gambar 3.14. Hasil konvolusi yang telah melalui proses aktivasi <i>ReLU</i>	35
Gambar 3.15. Hasil <i>pooled feature map</i>	35
Gambar 3.16. Hasil <i>flattening layer</i>	36
Gambar 3.17. <i>Fully connected layer</i>	36
Gambar 3.18. Grafik <i>Loss</i> tidak menggunakan <i>Early Stopping Function</i>	37
Gambar 3.19. Grafik <i>Loss</i> menggunakan <i>Early Stopping Function</i>	38
Gambar 3.20. Rancangan tampilan halaman beranda	38
Gambar 3.21. Rancangan tampilan halaman <i>training</i>	39
Gambar 3.22. Rancangan tampilan halaman <i>testing</i>	40
Gambar 4.1. Tampilan halaman Beranda	44
Gambar 4.2. Tampilan Halaman <i>Training</i>	45
Gambar 4.3. Tampilan Halaman <i>Testing</i>	45
Gambar 4.4. Data Citra <i>Training</i> Sidik Jari Pola <i>Arch</i>	46
Gambar 4.5. Data Citra <i>Training</i> Sidik Jari Pola <i>Left Loop</i>	46
Gambar 4.6. Data Citra <i>Training</i> Sidik Jari Pola <i>Right Loop</i>	47
Gambar 4.7. Data Citra <i>Training</i> Sidik Jari Pola <i>Tented Arch</i>	47
Gambar 4.8. Data Citra <i>Training</i> Sidik Jari Pola <i>Whrol</i>	48
Gambar 4.9. Data Citra <i>Testing</i>	48
Gambar 4.10. Tampilan Halaman Beranda	49
Gambar 4.11. <i>Subfolder</i> Folder <i>Train</i>	50
Gambar 4.12. <i>Folder Training</i> Citra Sidik Jari	50

Gambar 4.13. Tampilan Halaman <i>Testing</i> (Tahap <i>preprocessing</i> )	51
Gambar 4.14. Tampilan Halaman <i>Testing</i>	52
Gambar 4.15. Tampilan Halaman <i>input</i> data <i>Testing</i>	52
Gambar 4.16. Tampilan Halaman <i>Testing</i> (Hasil Klasifikasi)	53
Gambar 4.17. Grafik Loss pada Pelatihan Model	54
Gambar 4.18. Grafik Accuracy pada Pelatihan Model	55
Gambar 4.19. <i>Confusion Matrix</i>	62
Gambar 4.20. Contoh Citra Gagal karena <i>ridge</i> sidik jari tidak terlalu jelas	64
Gambar 4.21. Contoh Citra yang Gagal diidentifikasi oleh <i>CNN</i>	65

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Sidik jari secara luas saat ini digunakan dalam ranah teknologi , terutama dalam bidang digital. Pengamanan berbasis sidik jari dalam kehidupan sehari-hari juga sudah banyak digunakan. Salah penyebabnya karena sidik jari terbukti unik, akurat, aman, mudah dan nyaman untuk digunakan jika dibandingkan dengan sistem biometrik yang lainnya (Sediyono *et al*, 2011).

Sidik jari menjadi salah satu identitas diri seseorang yang bersifat alamiah. Sidik jari memiliki sifat *immutability* yaitu sidik jari akan tetap sama seumur hidup kecuali mendapatkan kecelakaan serius. Berbeda dengan bagian tubuh lainnya yang dapat mengalami perubahan seiring berjalannya waktu. Sidik jari memiliki guratan-guratan atau garis-garis pada epidermis yang membentuk suatu pola yang unik dan berbeda untuk setiap orang, bahkan untuk kembar identik sekalipun.

Pola pada sidik jari merujuk pada tata letak garis-garis (*ridges*) dan karakteristik khusus yang terdapat pada permukaan kulit pada ujung jari manusia. Setiap jari pada tangan seseorang juga mempunyai pola sidik jari yang tak sama. Terdapat pola tertentu yang khusus yang diperlihatkan oleh cap jari. Tipe pada pola sidik jari ini digunakan untuk pemeriksaan sidik jari secara kualitatif. Berdasarkan karakteristik-karakteristik ini, sidik jari digunakan dalam konteks identifikasi forensik untuk menentukan identitas seseorang dan sering digunakan dalam sistem keamanan (Sudiro & Lukman, 2019).

Pemeriksaan sidik jari secara kualitatif merupakan analisis dan pengamatan sidik jari tanpa melibatkan perhitungan atau pengukuran kuantitatif. Ini adalah metode pemeriksaan yang lebih umum digunakan dalam forensik dan keamanan untuk mengidentifikasi pola, karakteristik unik dan fitur-fitur lainnya dalam sidik jari.

Dalam pemeriksaan sidik jari secara kualitatif, seorang ahli forensik atau seorang analisis sidik jari akan melihat dan membandingkan karakteristik visual dalam sidik jari, seperti jenis pola, jumlah garis atau *ridge*, lokasi dan bentuk delta (titik pertemuan garis-garis), fitur khusus seperti *minutiae points* (titik minutiae) dan karakteristik lainnya yang dapat membantu mengidentifikasi individu atau mengklasifikasi sidik jari.

Metode kualitatif membantu mengkonfirmasi atau mengevaluasi persamaan atau perbedaan antara sidik jari yang ditemukan di tempat kejadian perkara, dalam database forensik atau di dalam keamanan. Pemeriksaan sidik jari secara kualitatif biasanya merupakan langkah awal dalam analisis sidik jari sebelum mengaplikasikan metode kuantitatif yang lebih canggih dan teknologi pengenalan sidik jari seperti sistem otomatis.

Sidik jari yang bersifat unik memiliki kelebihan sebagai alat identifikasi individu, namun sebaliknya hal ini juga menyulitkan proses pencocokannya. Ini menjadi masalah tersendiri bagi efisiensi dalam proses identifikasi. Oleh karena itu, pendekatan dan metode klasifikasi beragam dilakukan berdasarkan karakteristik umum yang terlihat pada sidik jari. Dengan adanya klasifikasi diharapkan dapat memudahkan dalam pencarian sidik jari berdasarkan ciri umum yang dijadikan sebagai acuan. Ciri umum yang dijadikan acuan pada penelitian ini ialah pola sidik jari.

Francis Galton melalui penelitiannya pada tahun 1892 membagi pola sidik jari kedalam tiga kelompok besar yaitu, *arch*, *loop*, dan *whorl*. Kemudian, tahun 1901 Edwar Henry memperbaharui klasifikasi Galton yang dikenal sebagai sistem klasifikasi Henry atau “*Henry classification system*” dan membagi pola sidik jari kedalam lima kelas yaitu *arch* (A), *tented arch* (T), *left loop* (L), *right loop* (R), dan

*whorl* (W). Pola sidik jari yang ditentukan oleh Edwar Henry adalah yang paling banyak digunakan dalam proses identifikasi seseorang (Kusmiati, 2020).

Banyak upaya penelitian yang telah dilakukan mengenai klasifikasi sidik jari, seperti Sulistiyani & Winarko (2014) yang melakukan penelitian tentang klasifikasi sidik jari manusia kedalam tiga kategori, yakni *whorl*, *arch* dan *loop*. Klasifikasi dilaksanakan melalui penerapan algoritma jaringan syaraf tiruan *Backpropagation* dan didapatkan akurasi mencapai 87,5%. Kemudian Pradana (2015) mengklasifikasikan citra sidik jari dengan berdasarkan pada enam tipe *pattern* dengan menerapkan metode *Euclidean Distance* dan *Template Matching*. Pada penelitian ini, akurasi yang diperoleh mencapai 76,6% namun data yang dipakai hanya berjumlah 30 citra.

Penelitian serupa dilakukan oleh Maulana & Rochmawati (2019) menggunakan metode CNN untuk klasifikasi citra buah dan dari proses pengujian didapatkan akurasi sebesar 97,97% menggunakan 345 citra uji. Kemudian Rahmawati *et al* (2021) melakukan penelitian menggunakan CNN untuk pengenalan aksara sunda dengan akurasi tertinggi yang diperoleh sebesar 98.03%.

Penggunaan *Convolutional Neural Network* dipilih dalam penelitian ini karena mempunyai hasil yang paling signifikan dalam pengenalan citra digital. Karena CNN diimplementasikan berdasarkan sistem pengenalan citra pada *visual cortex* manusia (Harjoseputro, 2018). Jaringan CNN menerima citra atau gambar sebagai masukan, selanjutnya akan melalui lapisan konvolusi dimana data diolah dengan filter yang sudah ditentukan. Masing-masing lapisan menghasilkan pola dari bagian citra yang memudahkan proses klasifikasi (Sudiro & Lukman, 2019).

Berdasarkan latar belakang dan penelitian terdahulu yang telah dijelaskan, dilakukan penelitian dengan judul **KLASIFIKASI CITRA SIDIK JARI MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK BERDASARKAN TIPE PATTERN PADA SISTEM HENRY**.

## 1.2. Rumusan Masalah

Pemeriksaan sidik jari secara kualitatif dilakukan oleh seorang ahli forensik yang kemudian akan melihat dan membandingkan jenis pola sidik jari. Sidik jari yang bersifat unik menguntungkan sebagai alat identifikasi seseorang, namun disisi lain mempersulit proses pencocokannya karena pola sidik diamati secara manual dan membutuhkan waktu serta ketelitian. Hal ini tidak efisien dari segi waktu dan bergantung pada kemampuan individu. Maka diperlukan sebuah sistem yang dapat mengklasifikasikan pola-pola sidik jari untuk mempercepat dan menyederhanakan proses pencocokannya.

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengklasifikasikan citra pola sidik jari menggunakan metode *Convolutional Neural Network*.

## 1.4. Batasan Masalah

Terdapat beberapa batasan masalah yang telah diberlakukan supaya pembahasan yang terkait dengan penelitian lebih terfokus. Beberapa dari batasan masalah tersebut meliputi:

1. Data latih yang digunakan ialah data sidik jari NIST (*National Institute of standards and Technology Fingerprint database*).
2. Data uji yang digunakan ialah data sidik jari *FVC* (*Fingerprint Verification Competition*) 2000 dan 2004.
3. Penelitian ini menggunakan data citra sidik jari dengan format *.png*
4. Pada penelitian ini digunakan data yang bersifat statis yang berarti data memerlukan *input* secara manual ke dalam sistem.
5. *Output* yang dihasilkan merupakan sistem berbasis *desktop* yang dapat mengklasifikasikan pola sidik jari *arch* (A), *tented arch* (T), *left loop* (L), *right loop* (R), dan *whorl* (W).

## 1.5. Manfaat penelitian

1. Dapat membantu mempercepat dan menyederhanakan proses pencocokkan sidik jari pada sistem identifikasi sidik jari. Hal ini membantu pihak yang berkepentingan dalam mengelompokkan sidik jari, karena setelah klasifikasi dilakukan, proses pencarian akan lebih efisien karena pencarinya difokuskan pada kelas pola sidik jari yang sesuai.
2. Menjadi referensi mengenai penerapan metode *Convolutional Neural Network* pada pengolahan citra sidik jari.

## 1.6 Metodologi Penelitian

Pada penelitian ini terdapat sejumlah tahapan yang diterapkan diantaranya:

### 1. Studi Literatur

Studi literatur adalah langkah pengumpulan beragam referensi dari berbagai sumber seperti skripsi, buku, artikel jurnal serta sumber informasi relevan dan terkait dengan penelitian ini.

### 2. Analisis Permasalahan

Setelah menyelesaikan studi literatur, langkah berikutnya yakni melakukan analisis masalah. Pada tahap ini, informasi dari berbagai sumber terkait penelitian akan dianalisis untuk menemukan metode yang sesuai guna mengatasi masalah yang ada dalam penelitian.

### 3. Perancangan Sistem

Setelah menyelesaikan tahap analisis permasalahan, langkah berikutnya adalah tahap perancangan sistem yang mencakup proses perancangan mengenai arsitektur umum, pengumpulan data serta desain antarmuka sistem.

### 4. Implementasi

Setelah menyelesaikan tahap perancangan sistem, langkah berikutnya yaitu tahap implementasi. *Convolutional Neural Network* ialah metode yang

diimplementasikan dalam penelitian ini dan akan diterapkan ke sistem menggunakan bahasa pemrograman Python.

### 5. Pengujian

Setelah menyelesaikan tahap implementasi, langkah berikutnya adalah tahap pengujian. Dalam tahap ini, sistem yang telah dirancang akan diuji untuk mengevaluasi metode *Convolutional Neural Network* dalam menyelesaikan masalah.

### 6. Penyusunan Laporan

Setelah menyelesaikan tahap pengujian, langkah berikutnya adalah tahap penyusunan laporan. Dalam langkah ini, penulis akan menyusun sebuah laporan yang mencakup seluruh proses penelitian dan menguraikan hasil dari penelitian yang sudah dilakukan.

## **1.7. Sistematika Penulisan**

Penulisan dalam skripsi ini mengikuti sistematika penulisan yang terdiri atas lima bagian, yakni:

### **Bab 1: Pendahuluan**

Bagian ini mencakup beberapa bagian, diantaranya latar belakang penelitian, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

### **Bab 2: Landasan Teori**

Beberapa teori yang diterapkan pada penelitian akan dijelaskan guna menyelesaikan serta memahami masalah yang dibahas dalam penelitian serta mencapai tujuan yang diinginkan pada penelitian. Bagian ini akan menjelaskan teori-teori terkait sidik jari, citra, pengolahan citra digital, metode yang digunakan yaitu *Convolutional Neural Network*. Penelitian-penelitian yang telah dilakukan sebelumnya juga dijelaskan pada bagian ini dengan tujuan sebagai acuan dalam pelaksanaan penelitian ini.

### **Bab 3: Analisis dan Perancangan Sistem**

Dalam bab ini, akan diuraikan analisis terkait permasalahan penelitian dan perancangan metode *Convolutional Neural Network* dalam mengklasifikasikan sidik jari berdasarkan tipe *pattern* pada Sistem Henry. Setiap tahap yang dilakukan dalam tahap *preprocessing*, segmentasi, *training*, *testing* serta perancangan sistem yang diimplementasikan di penelitian juga akan diuraikan dalam bagian ini.

### **Bab 4: Implementasi dan Pengujian Sistem**

Pembahasan terkait perancangan dan implementasi yang telah diuraikan dalam Bab 3 akan dilakukan di bagian ini. Disamping itu, bab ini juga akan memaparkan hasil-hasil yang diperoleh dari tahap pengujian terhadap implementasi yang sudah dilaksanakan.

### **Bab 5: Kesimpulan dan Saran**

Hasil akhir dari semua rangkaian penelitian yang sudah dilaksanakan akan diuraikan pada bab ini sebagai simpulan. Dalam upaya untuk menilai pencapaian tujuan penelitian, bagian ini juga akan mengulas apakah tujuan dari penelitian dapat tercapai. Selain itu, bagian ini juga akan mengemukakan saran untuk pengembangan dan penelitian mendatang.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Sidik Jari**

Sidik jari telah digunakan sejak tiga ribu tahun lalu untuk menandatangani dokumen hukum di Cina (Win *et al*, 2019). Pada tahun 1897, sidik jari juga telah digunakan untuk identifikasi sidik jari tanpa berbasis komputer atau *dactyloscopy*. Daktiloskopi ialah ilmu yang mempelajari dan menganalisis pola sidik jari untuk keperluan mengenali atau mengidentifikasi kembali identitas seseorang dengan mengamati pola garis unik yang ada di permukaan sidik jari tangan atau telapak kaki.

Awal mula terbentuknya sidik jari yakni sejak bulan ketiga hingga keempat saat manusia di dalam kandungan. Garis-garis yang menonjol pada sidik jari manusia mulai berkembang secara unik dan acak. Kondisi janin di dalam rahim yang dinilai mempengaruhi pola sidik jari manusia sehingga menghasilkan pola sidik jari yang berbeda pada setiap manusia. Francis Galton (1892) mengatakan bahwa tidak ada sidik jari yang serupa pada manusia bahkan manusia kembar sekalipun, karena sidik jari manusia memiliki banyak karakteristik unik yang dapat membedakan setiap manusia. Sidik jari memiliki beberapa sifat umum, antara lain:

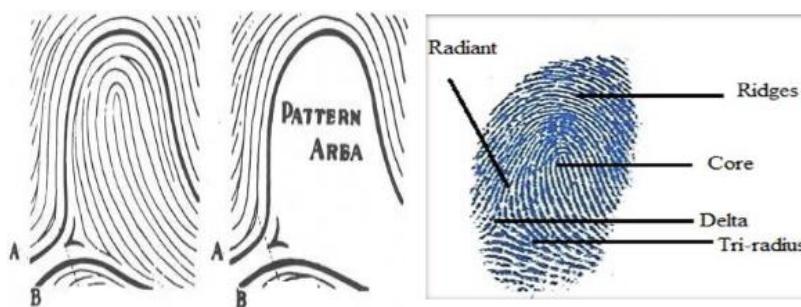
- a) Sifat *Perennial nature*, yakni guratan-guratan pada sidik jari tetap melekat pada kulit manusia sepanjang hidupnya.
- b) Sifat *Immutability*, yang berarti bahwa sidik jari tidak akan mengalami perubahan kecuali mengalami kecelakan serius.
- c) Sifat *Individuality*, yang menunjukkan bahwa setiap individu memiliki pola sidik jari unik dan berbeda satu sama lain.

## 2.2. Pola Sidik Jari

Sidik jari memiliki *pattern area* (area pola) yang merupakan suatu area dari sidik jari yang akan diartikan dan diklasifikasikan menjadi beberapa tipe pola.

Pola pada sidik jari terdiri dari beberapa komponen dan fitur penting, diantaranya:

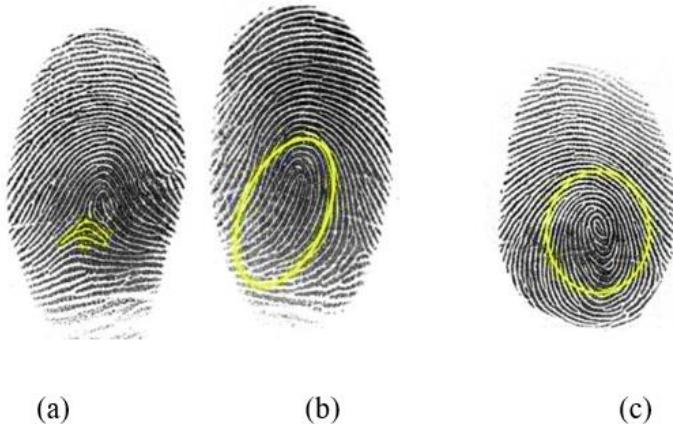
- *Pore* (pori), merupakan pembukaan kecil di kulit yang terdapat di sepanjang garis-garis sidik jari.
- *Ridge* (garis-garis), yaitu garis tipis yang membentuk pola sidik jari. *Ridge* ialah fitur utama pada sidik jari dan membentuk pola unik.
- *Re-entry Point* (titik kembali), yakni titik tempat garis lengkung dari pola *loop* berputar kembali ke jalurnya. Titik kembali ini hanya terdapat pada pola *loop*.
- *Core* (inti), yakni pusat dari suatu pola dan mempunyai variasi bentuk yang berbeda-beda tergantung pada tipe atau jenis polanya.
- *Delta*, yakni tempat di mana dua garis berjalan berdampingan lalu membentuk segitiga.
- *Triradius*, yakni titik pertemuan tiga sistem guratan yang saling berlawanan arah.
- *Radiant* atau *type line*, yakni garis yang berasal dari *triradius*.



**Gambar 2.1.** *Pattern area* (A) dan *radian* atau *type line* (B) (I Permatasari, 2019)

Menurut klasifikasi Galton (1892), terdapat tiga kategori utama dalam pola sidik jari, yakni *arch* (garis melengkung), *loop* (garis melingkar) dan *whorl* (pusaran).

Gambar 2.2. memperlihatkan tiga tipe utama pola sidik jari menurut Galton.



**Gambar 2.2.** Tiga tipe utama sidik jari (a) *arch*, (b) *loop*, (c) *whorl*. (Win et al, 2019)

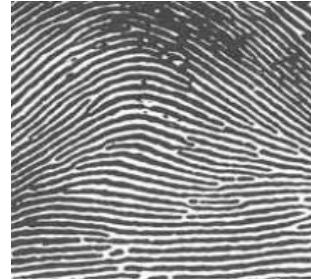
### 2.3. Sistem Henry

Edwar Henry pada tahun 1901 memperbaharui klasifikasi Galton yang dikenal sebagai “*Henry classification system*”. Sistem Henry merupakan sebuah sistem identifikasi sidik jari yang memanfaatkan pola atau *pattern* pada permukaan jari untuk memverifikasi identitas seseorang. Sistem ini sangat berguna dalam sistem keamanan dan identifikasi. Pola sidik jari dibagi menjadi lima kelas, yakni *arch* (A), *tented arch* (T), *left loop* (L), *right loop* (R), dan *whorl* (W). Klasifikasi sidik jari telah dikembangkan dari awal *computer vision* ada. Sebagai metode klasifikasi, *Henry's classification system* adalah metode klasifikasi sidik jari yang paling banyak digunakan (Win et al, 2019).

#### 2.3.1. Arch

*Arch* ialah jenis pola sidik jari dimana semua guratannya datang atau bermula dari satu sisi pola, kemudian cenderung mengalir ke sisi lainnya dan membentuk gelombang naik di bagian tengah. Pola sidik jari *arch* tidak mempunyai *triradius*. Pola ini seringkali terlihat sebagai lengkungan yang mulus tanpa pola ulang yang jelas

dalam sidik jari. Guratan-guratannya berjalan sejajar dari satu sisi ke sisi lain tanpa membentuk lengkungan yang curam.



**Gambar 2.3.** Pola sidik jari *arch*

### 2.3.2. *Tented Arch*

*Tented arch* ialah pola ini juga membentuk lengkungan, tetapi memiliki puncak kecil atau *tent* ditengah garis lengkungan pola *arch*. Garis sidik jari dari *triradius* cenderung mengarah lurus ke puncak *distal phalanx* (tulang yang paling ujung pada jari), sementara guratan lainnya membentuk lengkungan yang meninggi, menciptakan pola yang menyerupai bentuk tenda (*tent*).



**Gambar 2.4.** Pola sidik jari *tented arch*

### 2.3.3. *Right Loop*

Daerah pola pada jenis *loop* dikelilingi *type line* atau *radiant*. Pola ini terdiri atas serangkaian guratan yang berjalan sejajar, memasuki *pattern area*, kemudian berbelok 180° dan keluar dari *pattern area* pada sisi yang sama saat guratan tersebut masuk. Pola sidik jari *Right Loop* merujuk pada pola *loop* yang mempunyai garis lengkung

yang mengarah ke arah sisi kanan dari jari yang bersangkutan. Ini berarti *loop* tersebut akan memiliki titik kembali (*re-entry point*) di sisi kanan dari jalur lengkungannya.



**Gambar 2.5.** Pola sidik jari *right loop*

#### 2.3.4. *Left Loop*

Pola sidik jari *Left Loop* merujuk pada pola *loop* yang memiliki garis lengkung yang mengarah ke arah sisi kiri dari jari yang bersangkutan. Ini berarti *loop* tersebut akan memiliki titik kembali (*re-entry point*) di sisi kiri dari jalur lengkungannya.



**Gambar 2.6.** Pola sidik jari *left loop*

#### 2.3.5. *Whrol*

*Whrol* memiliki *pattern area* yang mirip dengan *pattern area* pada *loop*, yang dikelilingi oleh *radiant* atau *type line*. Pola sidik jari yang dibentuk oleh guratan yang membentuk pusaran. Pola terbentuk dari serangkaian guratan yang mengelilingi inti.

Inti (*core*) pada pola *whorl* berbentuk guratan pendek atau titik, serta mempunyai dua atau lebih *triradius*. Whrol memiliki tampilan seperti kumpulan lingkaran atau spiral.



**Gambar 2.7.** Pola sidik jari *whrol*

## 2.4. Citra (*Image*)

Pengertian citra secara matematis yaitu fungsi kontinyu dengan intensitas cahaya pada bidang dua dimensi. Sebuah citra perlu dipresentasikan dalam bentuk numerik dengan nilai-nilai distrik untuk dapat diproses oleh komputer. Representasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut dengan digitalisasi citra (Kusumanto & Tompunu, 2011).

Citra digital ialah istilah lain dari gambar, sebuah komponen dalam multimedia yang memiliki peran sangat penting sebagai bentuk informasi visual (Leksono, 2014). Citra digital dibagi kedalam 3 kategori, yakni citra berwarna (*color image*), citra biner (*binary image*) dan citrate keabuan (*grayscale image*).

### 2.4.1. Citra Keabuan (*Grayscale Image*)

Pada citra keabuan, warna yang terkandung didalamnya ialah warna keabuan, hitam dan putih. Tingkatan keabuan mengacu pada warna abu-abu mulai dari hitam sampai mendekati putih. *Grayscale image* mempunyai penyimpanan 8 bit dengan 256 variasi warna keabuan (Saputra, 2016). Citra *grayscale* memiliki kapasitas untuk memuat 256 tingkatan skala abu-abu, dengan setiap piksel mempunyai rentang nilai antara 0 hingga 255. Nilai 0 mewakili warna hitam, sedangkan 255 mewakili warna putih (Kumar & Verma, 2010) seperti yang ditunjukkan dalam Gambar 2.8.



**Gambar 2.8.** Citra Keabuan (*Grayscale Image*) (Kumar & Verma, 2010)

## 2.5. Pengolahan Citra Digital (*Digital Image Processing*)

*Digital Image Processing* adalah suatu teknik yang diterapkan pada suatu citra untuk meningkatkan kualitas atau mutu citra sehingga informasi yang terkandung dalam citra tersebut menjadi lebih terlihat dan jelas.

Pengolahan citra digital diartikan sebagai bidang ilmu yang mempelajari mengenai cara mengolah citra. Citra diolah dengan beberapa teknik yang dilakukan secara digital dengan teknologi komputer. Pada pengolahan citra digital, citra yang dapat diproses adalah gambar diam (foto) dan citra bergerak (*webcam*) (Kusumanto & Tompunu, 2011).

Adapun tujuan dari penerapan teknik pengolahan citra digital yakni untuk membuat peningkatan kualitas pada citra. Sehingga informasi yang terdapat pada citra akan terlihat dengan jelas dan dapat mengekstraksi informasi ciri pada citra. Dalam pengolahan citra digital terdapat sejumlah metode yang digunakan, yakni:

### 2.5.1. Contrast Limited Adaptive Histogram Equalization (CLAHE)

Metode peningkatan kontras atau *contrast enhancement* memiliki dua tujuan utama dalam meningkatkan kontras suatu citra, yaitu memperbaiki tampilan citra untuk interpretasi visual dan memfasilitasi atau meningkatkan kinerja untuk proses atau tahap selanjutnya (seperti analisis gambar, deteksi objek maupun segmentasi citra). Sebagian besar metode *contrast enhancement* mengandalkan modifikasi histogram, yang dapat diterapkan secara global atau secara lokal. Metode CLAHE mengatasi batasan pendekatan global dengan meningkatkan kontras lokal.

CLAHE yang merupakan singkatan dari *Contrast Limited Adaptive Histogram Equalization*, yakni metode ekualisasi citra. CLAHE ialah variasi dari *Adaptive Histogram Equalization* (AHE) yang mencegah penguatan kontras berlebihan.

Metode ini digunakan dengan tujuan mengatasi permasalahan kontras yang rendah pada citra digital. Pada dasarnya CLAHE bekerja membatasi peningkatan kontras yang biasanya dilakukan oleh *Histogram Equalization* biasa yang justru juga dapat menghasilkan peningkatan *noise* pada citra digital. Maka dengan membatasi peningkatan kontras pada *Histogram Equalization*, hasil yang diinginkan dapat dicapai dalam kasus dimana *noise* tidak terlalu terlihat dengan adanya peningkatan kontras.

CLAHE didefinisikan oleh dua parameter, yaitu *Block Size* (BS) dan *Clip Limit* (CL). Kedua parameter ini secara umum mengatur peningkatan kualitas citra. Ketika *clip limit* ditingkatkan, citra menjadi lebih terang karena citra masukan memiliki intensitas yang sangat rendah dan *clip limit* yang lebih besar membuat histogram citra menjadi lebih rata.

- *clipLimit* – parameter ini mengatur ambang batas peningkatan kontras. Pengaturan *default* adalah 40.
- *tileGrdeSize* – menentukan jumlah blok atau *tiles* dalam setiap baris dan kolom. Secara *default* nilainya 8x8.

### 2.5.2. Pengambangan (*Thresholding*)

*Thresholding* merupakan operasi non-liner yang dimanfaatkan untuk segmentasi citra, bertujuan untuk mengubah citra *grayscale* menjadi citra biner yang dapat diolah sebaik mungkin sehingga objek dan latar belakang pada citra masukan menjadi tampak dengan jelas. Dalam tahap ini, citra masukan yang digunakan ialah citra yang sudah melewati tahap sebelumnya. *Thresholding* memanfaatkan nilai ambang pada

citra untuk mengubah nilai piksel, yang melebihi nilai ambang diubah menjadi 1 (putih), sementara nilai yang kurang diubah menjadi 0 (hitam). Proses ini terlihat dalam Persamaan 2.1.

$$g(x,y) = \begin{cases} 1, & \text{if } f(x,y) \geq T \\ 0, & \text{if } f(x,y) < T \end{cases} \quad (2.1)$$

Dengan:

$g(x,y)$  : piksel citra yang telah diterapkan binerisasi

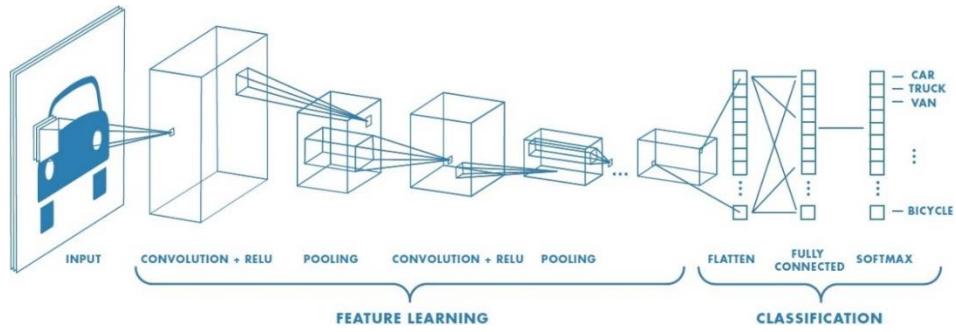
$f(x,y)$  : piksel citra *input-an*

$T$  : nilai ambang *thresholding*

## 2.6. Convolutional Neural Network

*Deep Learning* ialah bagian dari *machine learning* yang terfokus pada pengembangan dan penggunaan algoritma yang disebut jaringan saraf tiruan (*neural network*) yang di dalamnya terdapat sejumlah lapisan (*layer*) yang terhubung satu sama lain. Setiap *layer* ini merupakan algoritma atau metode yang berperan untuk melakukan pemrosesan dari *input* hingga menghasilkan *output* yang diinginkan. *Deep learning* memanfaatkan arsitektur jaringan saraf tiruan yang mendalam dan kompleks, salah satunya seperti *Convolutional Neural Network* yang digunakan untuk pengolahan gambar. Model yang ada pada *deep learning* mampu mengekstraksi fitur-fitur yang semakin abstrak dan kompleks dari data yang diberikan, sehingga sesuai untuk klasifikasi, deteksi objek, segmentasi, pengenalan suara dan banyak lagi.

CNN merupakan hasil pengembangan dari model jaringan saraf tiruan yang terinspirasi dari struktur jaringan saraf manusia. Umumnya CNN diterapkan dalam data gambar untuk mendeteksi serta pengenalan objek sebuah citra. Hampir sama seperti *neural network* pada umumnya, CNN mempunyai neuron yang terdiri atas bias dan bobot. Gambar 2.9 menunjukkan arsitektur CNN.



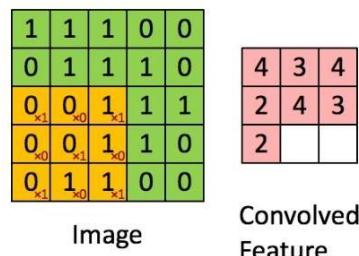
**Gambar 2.9.** Arsitektur *Convolutional Neural Network* (Putra, 2020)

CNN terdiri atas beberapa lapisan (*layer*) yang meliputi *convolutional layer*, *pooling layer* dan *fully connected layer*.

#### 2.6.1. *Convolutional Layer*

Lapisan konvolusi (*convolutional layer*) ialah tahap utama yang menjadi dasar dari sebuah CNN. Pada lapisan ini terdapat kumpulan *filter* atau kernel yang digunakan untuk mempelajari citra masukan. Fungsi utama dari lapisan ini adalah untuk melakukan ekstraksi pada fitur-fitur dari citra yang nantinya akan diolah lebih lanjut pada lapisan berikutnya untuk mendapatkan fitur yang lebih kompleks. Lapisan konvolusi melakukan tahap konvolusi pada *input* maupun *output* dari *layer* atau lapisan sebelumnya. Secara matematis, konvolusi yaitu mengaplikasikan suatu fungsi pada *output* fungsi lain secara berulang.

Dalam pengolahan citra, istilah konvolusi mengacu pada mengaplikaskan sebuah *kernel* pada citra diseluruh posisi atau *offset* yang memungkinkan, terlihat dalam Gambar 2.10. Seluruh area berwarna hijau ialah citra yang nantinya dilakukan konvolusi. *Kernel* bergerak dari sudut kiri atas ke kanan bawah. Gambar 2.10 menunjukkan hasil dari operasi konvolusi yang digambarkan pada area berwarna merah muda.



**Gambar 2.10.** Operasi Konvolusi (Illahiyah & Nilogiri, 2018)

Hasil konvolusi menjadi transformasi linear dari data *input* sesuai dengan informasi spasial yang terdapat dalam data tersebut. Bobot yang terdapat dalam lapisan ini menentukan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat melalui penelitian dengan memanfaatkan data *input* pada CNN (Suartika, 2016).

Selain parameter *kernel*, terdapat juga parameter lainnya pada *layer* ini yaitu *stride* dan *padding*. *Stride* berfungsi untuk mengatur pergerakan *kernel* untuk pengekstraksian. Informasi pada citra akan semakin *detail* didapatkan jika semakin sedikit jumlah *stride* yang digunakan. *Padding* berfungsi untuk menambahkan beberapa piksel jumlah tertentu pada citra sesuai yang dibutuhkan. Persamaan 2.2 merupakan cara untuk mendapatkan hasil akhir dari proses konvolusi.

$$\text{Output} = \frac{W - F + 2P}{S} + 1 \quad (2.2)$$

Dengan:

*W* = Ukuran citra

*F* = Ukuran *filter* atau *kernel*

*P* = Padding

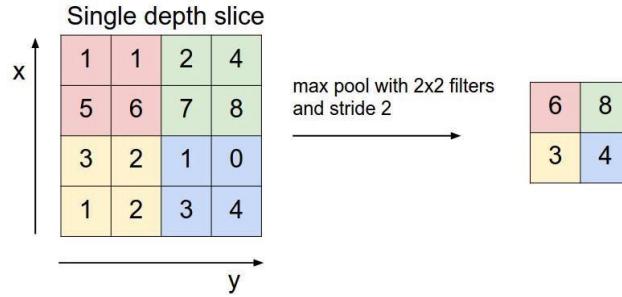
*S* = *Stride*

### 2.6.2. Pooling Layer

*Pooling layer* berperan dalam melakukan *down sampling* atau pengurangan dimensi dari *feature map* yang nantinya dapat membuat proses komputasi lebih cepat, sebab jumlah parameter yang harus diperbarui menjadi lebih sedikit. Selain itu, lapisan ini juga membantu mengatasi adanya *overfitting*.

Umumnya *pooling layer* terdiri atas *filter* dengan ukuran dan *stride* tertentu yang bergeser melintasi seluruh bagian *feature map*. *Max pooling* dan *average pooling* ialah *pooling* yang sering diterapkan. Dalam *max pooling*, nilai terbesar atau maksimum diambil dari area *filter* yang dipilih selama pergeseran. Pada *Max pooling*, hasil *output* dari *convolutional layer* dibagi kedalam sejumlah *grid* kecil dan setiap

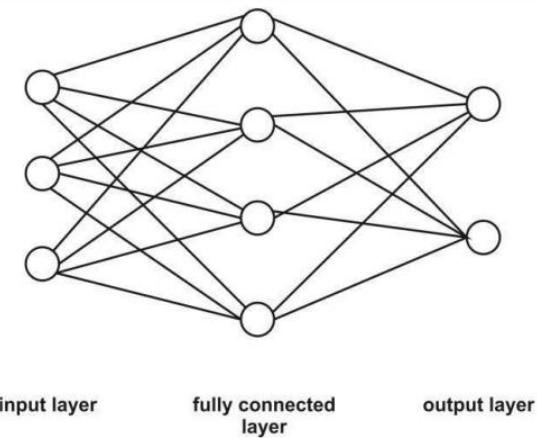
pergeseran *filter* nilai maksimum akan diambil dari masing-masing *grid* untuk membentuk matriks citra yang sudah direduksi, sebagaimana diperlihatkan dalam Gambar 2.11. Sementara dalam *average pooling*, nilai rata-ratanya diambil dari *filter* yang dipilih.



**Gambar 2.11.** Operasi *Max Pooling* (Suartika *et al*, 2016)

### 2.6.3. Fully Connected Layer

Lapisan ini digunakan untuk mengambil seluruh *neuron* pada lapisan sebelumnya dan menghubungkannya dengan setiap *single neuron* yang ada hal ini berguna untuk mengolah data sehingga dapat diklasifikasikan. Gambar 2.12 menunjukkan tahap *fully connected layer*.



**Gambar 2.12.** Operasi *Fully Connected Layer*

## 2.7. Penelitian Terdahulu

Penelitian terkait klasifikasi pola sidik jari dengan menerapkan metode yang beragam telah banyak dilakukan. Sulistiyan & Winarko (2014), melakukan penelitian tentang klasifikasi sidik jari manusia kedalam tiga kategori, diantaranya *whorl*, *arch* dan *loop*. Klasifikasi tersebut diimplementasikan dengan jaringan syaraf tiruan *backpropagation* dan didapatkan akurasi 87,5% pada hasil penelitian. Penelitian ini menggunakan 110 citra dengan data latih sebanyak 70 citra yang terdiri dari kelas *whorl*, *arch* dan *loop* serta data uji 40 citra yang terdiri atas kelas *whorl*, *arch* dan *loop*.

Kemudian Pradana (2015) melakukan pengklasifikasian citra sidik jari berdasarkan enam tipe pola atau *pattern* (*plain arch*, *plain whorl*, *tented arch*, *radial loop*, *ulnar loop* dan *double loop*) dengan metode *Euclidean Distance* dan *Template Matching*. *Euclidean Distance* yakni metode klasifikasi tetangga terdekat dengan melakukan perhitungan pada jarak antara dua objek. Dalam algoritma ini akan dihitung jarak antara matriks yang diambil dari citra masukan dengan citra sampel (referensi). Pada penelitian ini, akurasi yang diperoleh mencapai 76,6% namun data yang digunakan hanya berjumlah 30 citra.

Wang *et al.* (2014), mereka melakukan klasifikasi sidik jari berdasarkan empat tipe *pattern* (*arch*, *right loop*, *left loop* dan *whorl*) dengan metode *Fuzzy*. Berdasarkan *Depth Neural Network*, peneliti mengusulkan algoritma sidik jari. Mereka menggunakan *softmax regression* untuk klasifikasi *fuzzy* untuk peningkatan akurasi klasifikasi. Dengan hanya mengatur ambang batas probabilitas (*probability threshold*) diperoleh akurasi klasifikasi sebesar 96,1% (*threshold* = 0,85), akurasi sebesar 97,2% (*threshold* = 0,90) dan akurasi sebesar 98,0% (*threshold* = 0,95).

Penelitian serupa dilakukan oleh Maulana & Rochmawati (2019) menggunakan metode CNN untuk klasifikasi citra buah. Dalam penelitian ini, digunakan gabungan 3 *convolutional layer* dan 2 *fully connected layer* agar dapat mengklasifikasikan citra buah-buahan dengan baik. Didapatkan akurasi 97,97% pada tahap *testing* dengan menggunakan 345 citra uji. Dataset yang digunakan adalah dataset *Fruit-360* menggunakan 15 kelas dari 111 kelas data. Dalam proses pembelajaran, model CNN memperoleh akurasi 100% serta *loss* mencapai 0,012.

CNN mendapatkan akurasi sebesar 91,42% saat proses pengujian dengan 45 sampel citra buah.

Selanjutnya Rahmawati *et al.* (2021) mengimplementasikan *deep learning* pada pengenalan aksara sunda menggunakan metode *convolutional neural network*. Digunakan data latih sebanyak 756 dan data *testing* 188. Optimasi ADAM digunakan dengan percobaan *epoch* 20, 50, 100 dan 500. Penggunaan *epoch* 500 dan *learning rate* 0.1 didapatkan akurasi tertinggi sebesar 98.03%. Sementara hasil data *training* dengan nilai *epoch* 100 dan *learning rate* 0.001 didapatkan akurasi sebesar 96.71% untuk data latih dan 92.02% untuk data uji.

Kemudian Fadlia & Kosasih (2020) melakukan penelitian menggunakan CNN untuk melakukan klasifikasi terhadap jenis kendaraan. Penelitian ini menggunakan 120 citra. Jenis kendaraan yang diklasifikasikan ialah sepeda, motor dan mobil. Evaluasi dan pengujian model pada ketiga tipe kendaraan menggunakan *package Keras* menghasilkan akurasi 94,4% saat proses pelatihan dan akurasi 73,3% pada proses pengujian

**Tabel 2.1.** Penelitian Terdahulu

No.	Peneliti	Metode	Keterangan
1.	Sulistiyani & Winarko (2014)	<i>Backpropagation</i>	Mengklasifikasi tiga kategori sidik jari <i>arch</i> , <i>whorl</i> dan <i>loop</i> . Hasil penelitian dengan akurasi 87,5%. Penelitian ini menggunakan dataset 110 citra, dimana terdapat 70 citra data latih, 40 citra data uji.
2.	Pradana (2015)	<i>Euclidean Distance</i> dan <i>Template Matching</i>	Mengklasifikasikan enam pola sidik jari plain <i>arch</i> , plain <i>whorl</i> , tented <i>arch</i> , <i>ulnar loop</i> dan <i>double loop</i> . Didapatkan akurasi mencapai 76,6% namun data yang dipakai hanya berjumlah 30 citra.

**Tabel 2.2.** Penelitian Terdahulu (Lanjutan)

No.	Peneliti	Metode	Keterangan
3.	Wang <i>et al.</i> (2014)	<i>Fuzzy dan Back Propagation Neural Network</i>	Mengklasifikasi empat tipe sidik jari <i>arch</i> , <i>right loop</i> , <i>left loop</i> dan <i>whorl</i> . Dengan hanya mengatur ambang batas probabilitas ( <i>probability threshold</i> ) diperoleh akurasi klasifikasi sebesar 96,1% ( <i>threshold</i> = 0,85), akurasi sebesar 97,2% ( <i>threshold</i> = 0,90) dan akurasi sebesar 98,0% ( <i>threshold</i> = 0,95).
4.	Maulana & Rochmawati (2019)	<i>Convolutional Neural Network</i>	Mengklasifikasi citra buah dengan menggabungkan 3 <i>convolutional layer</i> dan 2 <i>fully connected layer</i> . Didapatkan akurasi 97,97% pada tahap <i>testing</i> menggunakan 345 citra uji. Proses pembelajaran atau <i>learning</i> model CNN memperoleh <i>loss</i> sebesar 0,012 dan akurasi 100%. Sementara proses <i>testing</i> model CNN, tercapai tingkat akurasi 91,42% dengan 45 sampel citra buah.
5.	Rahmawati <i>et al.</i> (2021)	<i>Convolutional Neural Network</i>	Implementasi <i>deep learning</i> pada pengenalan aksara sunda dengan data latih sebanyak 756 dan data uji 188. Dengan menggunakan optimasi ADAM , <i>epoch</i> 500 dan <i>learning rate</i> 0.1 didapatkan akurasi paling tinggi 98.03%. Sementara hasil data <i>training</i> dengan nilai <i>epoch</i> 100 dan <i>learning rate</i> 0.001 didapatkan akurasi 96.71% untuk data latih dan 92.02% untuk data uji.

**Tabel 2.3.** Penelitian Terdahulu (Lanjutan)

No.	Peneliti	Metode	Keterangan
6.	Fadlia & Kosasih (2020)	<i>Convolutional Neural Network</i>	Mengklasifikasikan terhadap jenis kendaraan. Jumlah data yang diterapkan pada penelitian mencakup 120 citra, diantaranya citra kendaraan seperti sepeda, mobil dan motor. Evaluasi dan uji coba menggunakan <i>package Keras</i> dilakukan terhadap tiga jenis kendaraan, dengan akurasi 94,4% saat proses pelatihan serta 73,3% saat proses pengujian.

Perbedaan penelitian ini dengan penelitian terdahulu yaitu Sulistiyani & Winarko (2014) melakukan klasifikasi sidik jari kedalam tiga kategori pola, diantaranya *whorl*, *arch* dan *loop*. Klasifikasi tersebut diimplementasikan dengan jaringan syaraf tiruan *backpropagation*. Kemudian Wang *et al.* (2014), mengklasifikasikan sidik jari berdasarkan empat tipe *pattern* (*arch*, *right loop*, *left loop* dan *whorl*) dengan metode *Fuzzy* dan *Backpropagation neural network*. Mereka menggunakan *softmax regression* untuk klasifikasi *fuzzy* untuk peningkatan akurasi klasifikasi. Pradana (2015) mengklasifikasikan citra sidik jari berdasarkan enam tipe pola atau *pattern* (*plain arch*, *plain whorl*, *tented arch*, *radial loop*, *ulnar loop* dan *double loop*) dengan metode *Euclidean Distance* dan *Template Matching* namun data yang digunakan hanya berjumlah 30 citra.

Sedangkan pada penelitian ini mengklasifikasikan lima pola sidik jari berdasarkan pola sidik jari yang terdapat pada Sistem Henry diantaranya yaitu sidik jari *arch* (A), *tented arch* (T), *left loop* (L), *right loop* (R), dan *whorl* (W). Metode *Convolutional Neural Network* digunakan karena metode ini dapat melakukan ekstraksi fitur secara baik yang nantinya ekstraksi fitur akan sangat diperlukan untuk pengenalan ciri objek.

## **BAB 3**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1. Data yang digunakan**

Penelitian ini menggunakan data *input* berupa data sidik jari yang didapat melalui NIST (*National Institute of standards and Technology*) *Fingerprint database*. Seluruh data berekstensi PNG dengan ukuran 512 x 512 piksel dengan jumlah data sebesar 1745 citra sidik jari. Terdapat 5 jenis data citra sidik jari yang akan di uji dalam penelitian ini, yakni sidik jari dengan pola *Arch*, *Left loop*, *Right loop*, *Tented arch* dan *Whorl*. Gambar 3.1 memperlihatkan contoh data citra sidik jari yang digunakan pada penelitian ini.



**Gambar 3.1.** Contoh Data Sidik Jari

Kemudian dataset NIST yang berjumlah 1745 tersebut dibagi menjadi data *training* dan data validasi. Pembagian data dilakukan dengan data latih 80% dan data validasi sebesar 20%. Pengujian model dilakukan dengan menggunakan dataset sidik jari dari *FVC* (*Fingerprint Verification Competition*) dengan jumlah data 75 data citra sidik jari yang terlihat dalam Tabel 3.1.

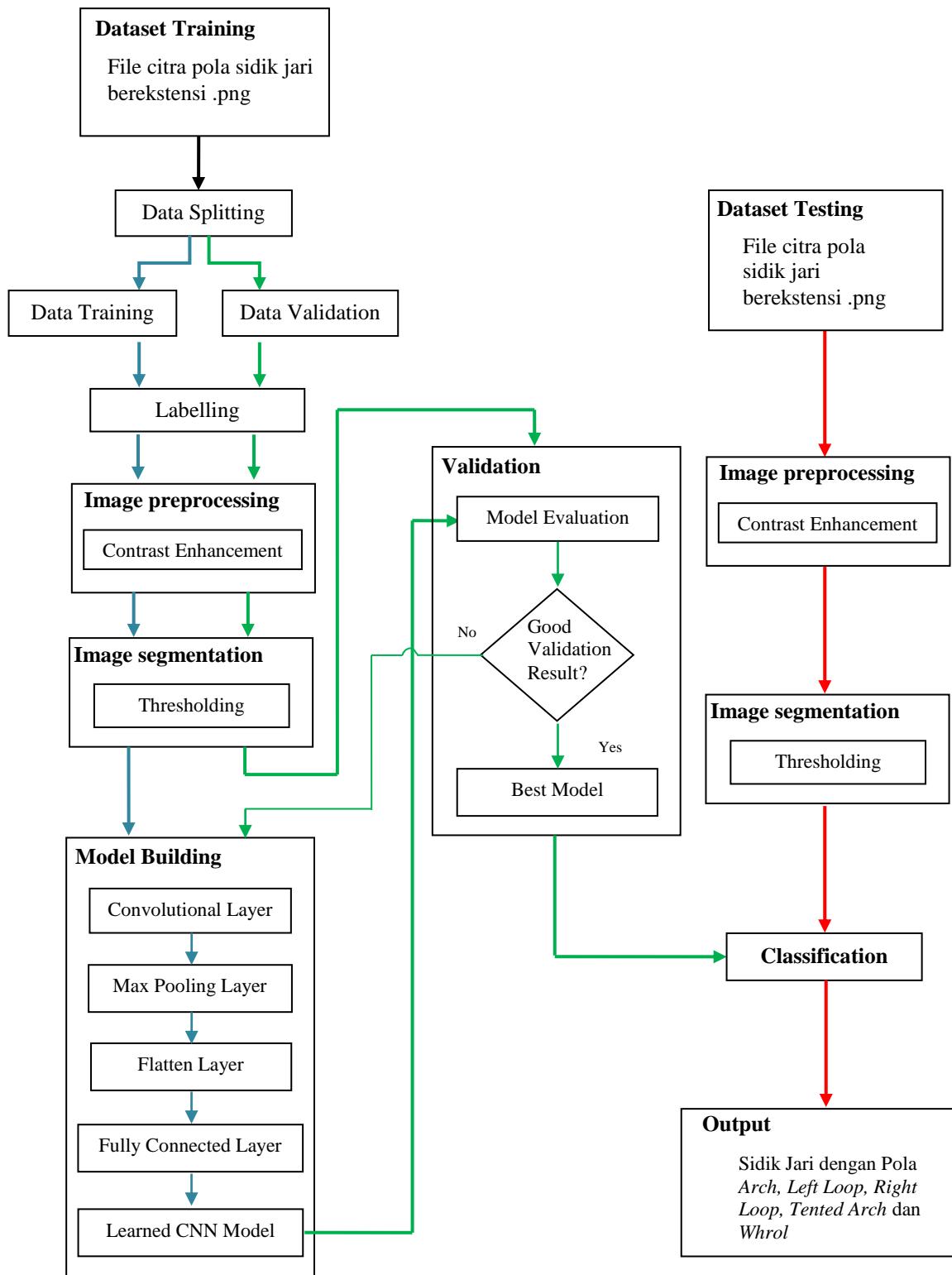
**Tabel 3.1.** Pembagian Dataset

Dataset Training		Data Testing
Training	Validation	75
1396	349	

### 3.2. Analisis Sistem

Beberapa tahapan akan dilakukan dalam penelitian ini, termasuk diantaranya: pengumpulan data yang diperoleh dari kumpulan citra sidik jari yakni dari NIST *fingerprint database* berupa file berekstensi *png*. Kemudian, dataset *training* citra sidik jari akan dibagi menjadi data *training* dan validasi. Data latih digunakan dalam proses pelatihan, data validasi untuk proses validasi sedangkan data uji yang diperoleh dari FVC *database* 2022 dan 2004 digunakan untuk proses pengujian hasil pelatihan. Tahap pertama, dataset *training* akan melalui tahap *data labeling* dengan tujuan untuk mengkategorikan citra ke dalam kelas tertentu. Sehingga memungkinkan pembuatan model klasifikasi dapat mengklasifikasikan objek dalam citra.

Selanjutnya dataset memasuki tahap *preprocessing*. Dalam tahap ini citra diperbaiki supaya memperoleh hasil yang lebih baik sebelum memasuki tahap berikutnya. Tahap ini mencakup proses *contrast enhacement* dengan CLAHE yang bertujuan untuk memperbaiki nilai kontras pada citra sidik jari yang akan diproses sehingga guratan – guratan sidik jari yang ada pada citra dapat terlihat lebih jelas. Tahap selanjunya yakni *image segmentation* menggunakan metode *thresholding*, dimana tujuannya untuk membedakan antara objek (*foreground*) dan latar belakang (*background*). Selanjutnya data yang sudah melalui tahap *preprocessing* akan menjadi masukan untuk model *Convolutional Neural Network* untuk melalui proses pelatihan (*training*) sehingga didapatkan model CNN yang telah belajar untuk mengklasifikasikan pola sidik jari. Kemudian model yang telah belajar akan diuji menggunakan *testing dataset* untuk mengetahui kemampuannya dalam mengklasifikasikan citra sidik jari. Gambar 3.2 menunjukkan gambaran tentang arsitektur umum yang memaparkan metodologi yang digunakan pada penelitian ini.



Keterangan :

- Proses Training
- Proses Validasi
- Proses Testing

**Gambar 3.2.** Arsitektur Umum

### 3.2.1. Labelling

*Labelling* dibutuhkan dalam proses pelatihan karena CNN ialah salah satu jenis algoritma *supervised learning*, maka dibutuhkan *labeling* data dengan tujuan memberikan nilai target untuk masing-masing citra. Dengan target dari citra *arch* adalah kategori 0, target dari citra *left loop* adalah 1, target dari citra *right loop* adalah 2, target dari citra *tented arch* adalah 3 dan target dari citra *whorl* adalah 4.

### 3.2.2. Image Preprocessing

Tahap ini meliputi *contrast enhancement*, yakni langkah dimana citra diproses dengan tujuan menghasilkan citra yang lebih optimal untuk memasuki proses selanjutnya. Penelitian ini menggunakan citra *grayscale* pada data *training*. Pada citra keabuan, intensitas setiap piksel dinyatakan dalam nilai tunggal, dengan intensitas piksel untuk setiap komponen merah, hijau dan memiliki nilai yang seragam atau sama. Gambar 3.3 menampilkan nilai piksel salah satu citra sidik jari.

```

[[184 184 184]
 [181 181 181]
 [183 183 183]
 [184 184 184]
 [184 184 184]
 [181 181 181]
 [171 171 171]
 [163 163 163]
 [154 154 154]
 [137 137 137]
 [116 116 116]
 [ 94  94  94]
 [ 86  86  86]
 [ 83  83  83]
 [ 85  85  85]
 [ 87  87  87]
 [ 94  94  94]
 [115 115 115]
 [142 142 142]
 [160 160 160]

...
[183 183 183]
[186 186 186]
[190 190 190]
]
(512, 512, 3)

```

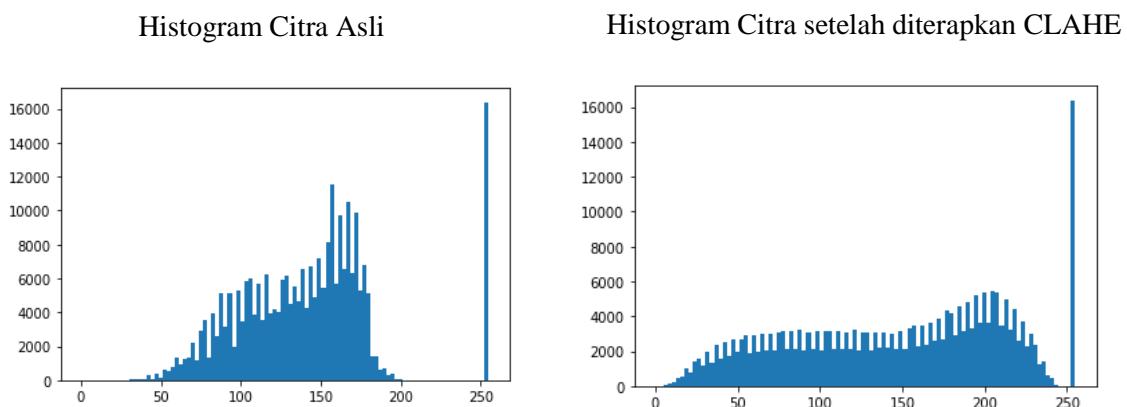
**Gambar 3.3.** Nilai piksel citra

### 3.2.2.1. Contrast Enhancement

Langkah selanjutnya yaitu menerapkan *contrast enhancement* pada citra. Penggunaan metode CLAHE dimaksudkan untuk memperbaiki nilai kontras pada citra sehingga guratan-guratan pola sidik jari pada citra menjadi terlihat dengan jelas.

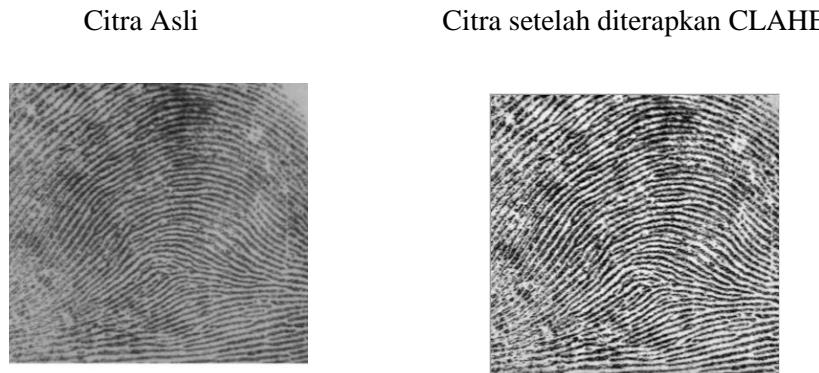
Terlihat pada Gambar 3.5 histogram dari intensitas piksel citra asli, sebagian besar informasi pada citra terdapat pada wilayah atau bagian kecil saja, sehingga memiliki kontras yang rendah (*low contrast*). Dengan menerapkan CLAHE, persebaran informasi pada citra dapat diubah sehingga nilai piksel pada histogram akan merata.

Pada CLAHE, *Adaptive histogram* berarti bahwa dalam prosesnya citra dibagi menjadi blok kecil, penelitian ini menggunakan nilai  $10 \times 10$  *tiles*. Kemudian, dalam setiap blok atau *tiles* dilakukan *histogram equalization* dan untuk meminimalkan peningkatan kontras diterapkanlah *contrast limiting*, pada penelitian ini digunakan *clipLimit* = 2.0. Terlihat pada gambar 3.4 histogram citra yang telah melalui proses CLAHE terlihat lebih baik, informasi pada citra lebih merata.



**Gambar 3.4.** Perbedaan Histogram citra sebelum dan setelah diterapkan CLAHE

Gambar 3.5 menampilkan contoh citra yang telah diterapkan metode CLAHE



**Gambar 3.5.** Citra pola sidik jari sebelum dan setelah diterapkan metode CLAHE

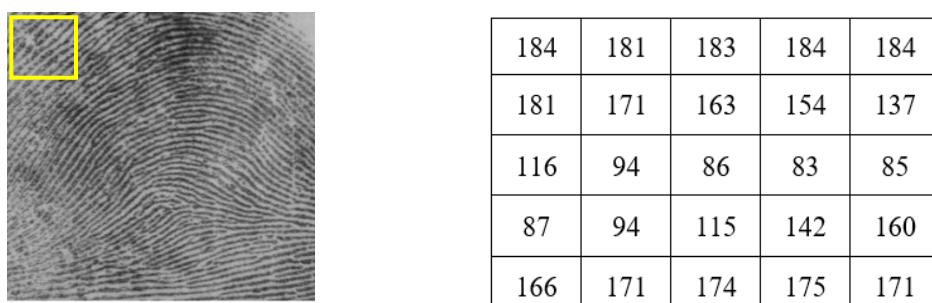
### 3.2.3. Image Segmentation

Tujuan dari segmentasi citra yakni untuk memisahkan antara objek (*foreground*) dengan *background*. Metode yang diterapkan dalam penelitian ini untuk yakni *thresholding*.

#### 3.2.3.1. Thresholding

*Thresholding* pada citra dilakukan setelah melalui tahap *contrast enhancement*. Tujuan dilakukan *thresholding* yakni untuk memisahkan guratan-guratan pada citra sidik jari dengan latar yang terdapat pada citra. *Thresholding* mengubah citra menjadi 2 warna yakni putih dan hitam dimana putih merepresentasikan guratan-guratan pada pola sidik jari dan hitam merepresentasikan latar.

Citra masukan yang digunakan berukuran 512 x 512 piksel, dalam contoh perhitungan yang akan dibahas pada bab ini hanya menggunakan sebagian kecil saja dari citra. Setelah mendapatkan nilai *grayscale* dari tiap piksel, maka nilai piksel pada citra akan direpresentasikan sebagaimana tampak dalam Gambar 3.6.



**Gambar 3.6.** Nilai *grayscale* pada setiap piksel

Perhitungan *thresholding* dapat dilakukan perhitungannya berdasarkan persamaan 2.2. Pada persamaan tersebut, ketika nilai piksel pada citra input lebih rendah daripada nilai *threshold* ( $T$ ), maka nilai piksel akan diubah menjadi 0 atau berwarna hitam. Tetapi, jika nilai piksel pada citra masukan melebihi atau setara dengan *threshold* ( $T$ ), maka piksel akan dinayatakan bernilai 1 atau berwarna putih. Berdasarkan nilai piksel yang telah diperoleh dari proses sebelumnya, diketahui:

Nilai maksimum : 184

Nilai minimum : 83

Sehingga,

$$T = \frac{184 + 83}{2} = 133.5$$

Dengan demikian, maka:

$$184 > 133.5 = 1,$$

$$181 > 133.5 = 1,$$

$$183 > 133.5 = 1,$$

$$184 > 133.5 = 1,$$

$$184 > 133.5 = 1,$$

.

$$166 \geq 133.5 = 1,$$

$$171 \geq 133.5 = 1,$$

$$174 \geq 133.5 = 1,$$

$$175 \geq 133.5 = 1,$$

$$171 \geq 133.5 = 1$$

Gambar 3.7 menampilkan nilai *thresholding* pada citra dengan ukuran matriks  $5 \times 5$  dan contoh citra yang telah dilakukan *thresholding*.

1	1	1	1	1
1	1	1	1	1
0	0	0	0	0
0	0	0	1	1
1	1	1	1	1



**Gambar 3.7.** Nilai *thresholding* dan citra setelah dilakukan *thresholding*

### 3.2.4. Model Building

Pada proses ini terjadi pembentukan model (*model building*) dengan menggunakan metode CNN. Model terdiri atas sejumlah lapisan yang bertujuan untuk menerima citra inputan. Tahapan - tahapan pada CNN yakni *convolutional layer*, *max-pooling*, *flattening* dan *fully-connected layer*. Model yang sudah terbentuk nantinya akan disimpan dan digunakan untuk tahap pengujian (*testing*), yaitu mengklasifikasikan pola sidik jari.

#### 3.2.4.1. Convolutional Layer

Citra masukan melewati tahap konvolusi untuk mengekstraksi ciri dari citra tersebut. Jumlah *layer* konvolusi yang digunakan dalam penelitian ini ialah sebanyak 3 *layer*. Gambar 3.8 adalah representasi dari salah satu lapisan konvolusi. Representasi citra yang diambil dari proses pengolahan citra yang telah dilakukan sebelumnya dengan dimensi 5 x 5.

1	1	1	1	1
1	1	1	1	1
0	0	0	0	0
0	0	0	1	1
1	1	1	1	1

**Gambar 3.8.** Nilai piksel citra masukan

Penelitian ini menggunakan *padding=’same’* atau *zero padding*, yaitu parameter yang menentukan jumlah pixel berisi nilai 0 yang akan ditambahkan di setiap sisi dari input. *Padding* digunakan untuk meningkatkan performa dari model karena *convolutional filter* akan fokus pada informasi yang sebenarnya yaitu yang berada diantara zero padding tersebut. Dengan menggunakan *padding*, dimensi *output* dapat diatur agar

tetap sama dengan dimensi *input* atau setidaknya tidak terlalu banyak berkurang. Representasi citra yang ditambahkan *zero padding* terlihat dalam Gambar 3.9.

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0
0	0	0	0	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

**Gambar 3.9.** Nilai piksel citra masukan dengan *padding* 0

Pada proses konvolusi, akan dilakukan *define kernel size* yang akan digunakan. Kernel mengekstrak *feature* tertentu dari citra input. Ukuran *kernel* atau dikenal juga sebagai *filter* yang digunakan sebagai pengkalian dari nilai piksel pada citra inputan adalah kernel ukuran 3x3.

1	1	1
0	0	0
-1	-1	-1

**Gambar 3.10.** *Feature Detector* dengan kernel size 3x3

Persamaan 2.3 dapat digunakan untuk mengetahui besar *output* atau *feature map* yang akan diperoleh. Berdasarkan Gambar 3.10 dilakukan perhitungan:

$$\text{Output} = \frac{W - F + 2P}{S} + 1$$

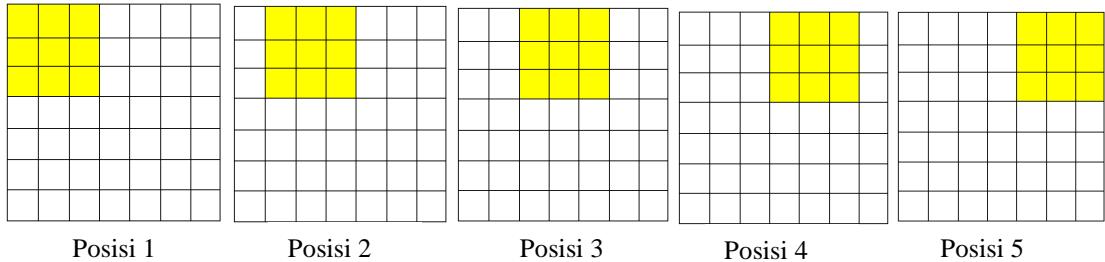
Dengan:

W = 5, F = 3, P = 1, S = 1

Maka,

$$\text{Output} = \frac{5 - 3 + 2(1)}{1} + 1 = 5$$

Dari perhitungan yang dilakukan, maka diperoleh *feature map* dengan ukuran  $5 \times 5$ .



**Gambar 3.11.** Proses pergeseran dengan *stride* 1 dan *kernel*  $3 \times 3$

Hasil operasi konvolusi ini dihasilkan dari perkalian dua fungsi yang akan menghasilkan matrik baru.

$  \begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}  \quad  \begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}  \quad  \begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}  \quad  \begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}  \quad  \begin{array}{ c c c } \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}  $	<b>X</b>	$  \begin{array}{ c c c } \hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}  =   \begin{array}{ c c c } \hline -2 & & \\ \hline \end{array}  $
<i>Feature Detector</i> $3 \times 3$		<i>Feature Map</i> $5 \times 5$

**Gambar 3.12.** Proses konvolusi pada pergeseran pertama

Setiap posisi dilakukan perhitungan dengan jumlah elemen perkalian antara *kernel* dengan matriks pada citra masukan. Hasil dari operasi ini akan membentuk *feature map*. Berikut merupakan perhitungan dari pergeseran matriks pada proses konvolusi.

$$C_{1,1} = (0*1) + (0*1) + (0*1) + (0*0) + (1*0) + (1*0) + (0*(-1)) + (1*(-1)) + (1*(-1)) = -2$$

$$C_{1,2} = (0*1) + (0*1) + (0*1) + (1*0) + (1*0) + (1*0) + (1*(-1)) + (1*(-1)) + (1*(-1)) = -3$$

$$C_{1,3} = (0*1) + (0*1) + (0*1) + (1*0) + (1*0) + (1*0) + (1*(-1)) + (1*(-1)) + (1*(-1)) = -3$$

$$C_{1,4} = (0*1) + (0*1) + (0*1) + (1*0) + (1*0) + (1*0) + (1*(-1)) + (1*(-1)) + (1*(-1)) = -3$$

$$C_{1,5} = (0*1) + (0*1) + (0*1) + (0*0) + (1*0) + (1*0) + (0*(-1)) + (1*(-1)) + (1*(-1)) = -2$$

$$C_{2,1} = (0*1) + (1*1) + (1*1) + (0*0) + (1*0) + (1*0) + (0*(-1)) + (0*(-1)) + (0*(-1)) = 2$$

$$C_{2,2} = (1*1) + (1*1) + (1*1) + (1*0) + (1*0) + (1*0) + (0*(-1)) + (0*(-1)) + (0*(-1)) = 3$$

.

.

$$C_{5,5} = (1*1) + (1*1) + (0*1) + (1*0) + (1*0) + (0*0) + (0*(-1)) + (0*(-1)) + (0*(-1)) = 2$$

Proses perhitungan terus dilakukan sampai ke  $C_{5,5}$  sehingga didapatkan hasil konvolusi layer.

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0
0	0	0	0	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

**X** =

-2	-3	-3	-3	-2
2	3	3	3	2
2	3	2	1	0
-2	-3	-3	-3	-2
0	0	1	2	2

*Feature Detector 3 x 3*                           *Feature Map 5 x 5*

**Gambar 3.13.** Proses konvolusi pada semua piksel

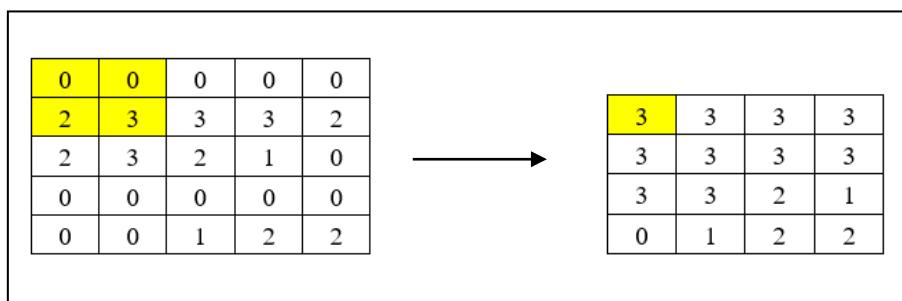
Karena pada penelitian ini menggunakan aktivasi *ReLU* (*Retrified Linear Unit*), maka semua nilai yang bernilai negatif akan diubah menjadi 0, maka didapatkanlah hasil konvolusi dan terlihat di Gambar 3.14.

0	0	0	0	0
2	3	3	3	2
2	3	2	1	0
0	0	0	0	0
0	0	1	2	2

**Gambar 3.14.** Hasil konvolusi yang telah melalui proses aktivasi *ReLU*

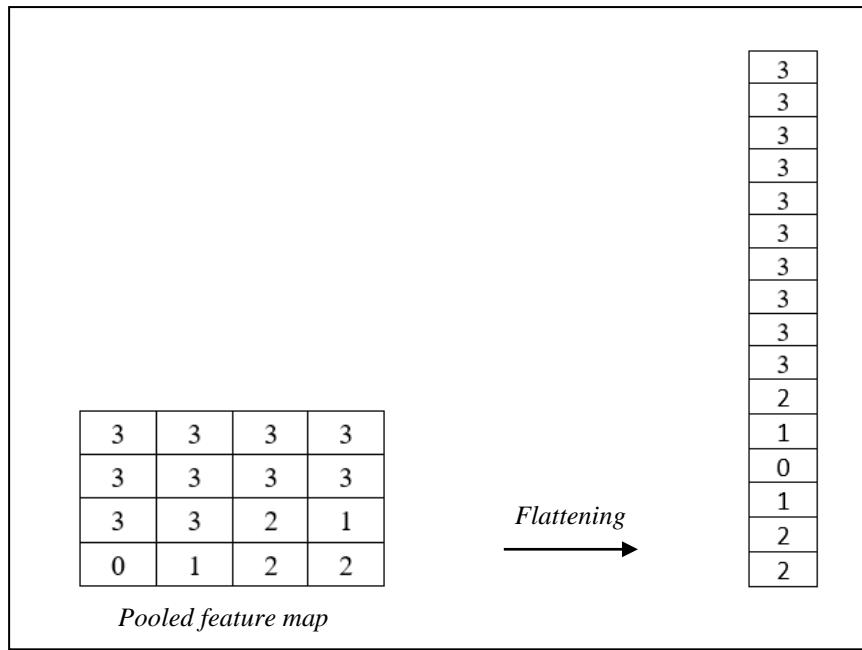
### 3.2.4.2. Pooling Layer

Dalam CNN, *output* dari suatu *layer* itu akan menjadi inputan dalam *layer* selanjutnya. Langkah selanjutnya ialah *max pooling*, dimana lapisan konvolusi dipecah menjadi *grid* yang lebih kecil, dan nilai yang paling besar diambil dari matriks tersebut. *Pool size* yang digunakan yakni ukuran  $2 \times 2$  dengan *stride* 1. Gambar 3.15 menampilkan hasil *pooled feature map*.



**Gambar 3.15.** Hasil *pooled feature map*

Setelah mendapatkan nilai dari proses konvolusi, *flatten layer* akan mengubah nilai yang awalnya bentuk matriks menjadi bentuk vektor. Kemudian nilai tersebut akan digunakan dalam langkah selanjutnya. Proses ini dikenal dengan proses *flattening*. Gambar 3.16 menampilkan hasil *flattening*.

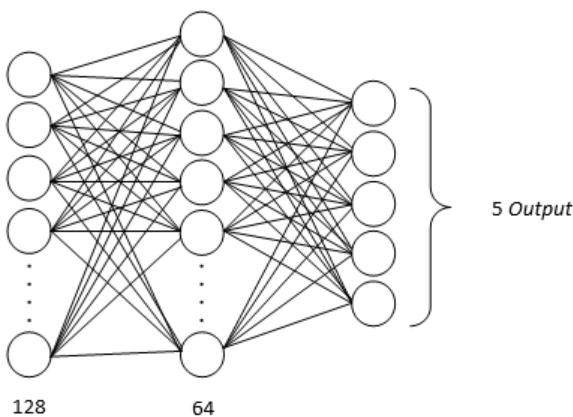


**Gambar 3.16.** Hasil *flattening layer*

Kemudian hasil *flattening layer* akan melalui proses *dropout*, yakni proses memilih beberapa neuron yang tidak digunakan selama proses pelatihan. Neuron tersebut nantinya akan dibuang secara acak supaya tidak mengakibatkan *overfitting* pada sebuah proses training. Neuron yang akan dilakukan *dropout* yakni 20% dari unit neuron.

#### 3.2.4.3. Fully-connected Layer

Output dari *flatten layer* nantinya digunakan sebagai inputan pada *fully connected layer*. Dimana *fully connected layer* artinya seluruh *neural network* yang terdapat pada layer terhubung satu sama lain. Penelitian ini menggunakan 1 *hidden layer* dan jumlah *node* atau *neuron* yang digunakan adalah sebanyak 64 unit neuron. Gambar 3.17 menampilkan *fully connected layer*.

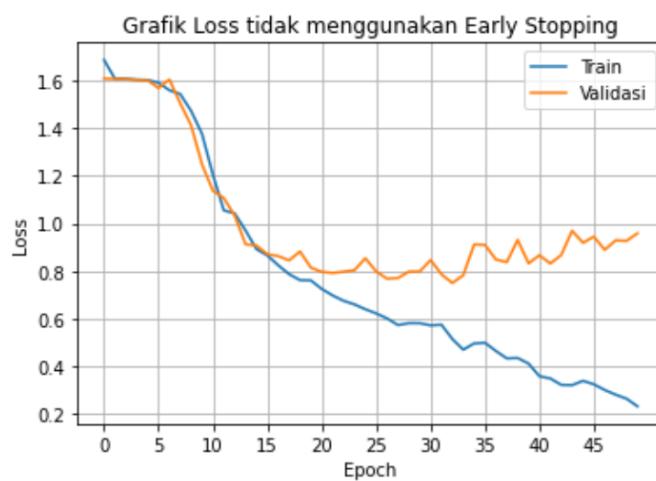


**Gambar 3.17.** *Fully connected layer*

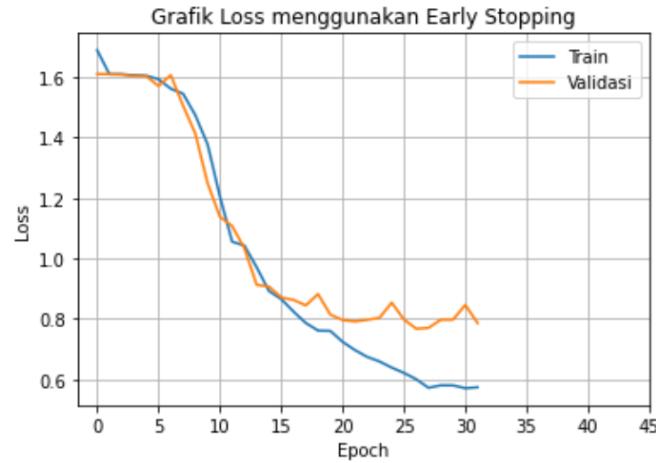
Pada lapisan *fully connected* terdapat *dense layer*. Fungsi aktivasi *softmax* digunakan untuk *output* layer karena penelitian ini termasuk metode klasifikasi multikelas.

Pemodelan menggunakan fungsi optimasi Adam (*Adaptive Moment Estimation*) untuk menentukan parameter yang tepat dengan meminimalkan nilai *loss*. Pada pemodelan *deep learning* fungsi optimasi Adam sangat umum digunakan. Untuk menentukan cepat atau lambatnya sistem dalam pembelajaran digunakan sutau nilai *learning rate*. *Learing rate* ialah sebuah parameter yang mempengaruhi kinerja model CNN.

Pada penelitian ini, tahap *training* dilakukan dengan menggunakan 30 epoch dan *batch size* sebesar 128, yang menghasilkan 11 iterasi dalam 1 epoch. Untuk menghindari model yang *overfitting*, dalam proses iterasi digunakan salah satu teknik regulasi yakni *Early Stopping Function*, yang bertujuan untuk memberhentikan proses pelatihan (*training*) jika nilai *loss* pada data validasi telah mencapai titik minimun dan tidak menunjukkan peningkatan yang signifikan. Gambar 3.18 dan Gambar 3.19 memperlihatkan perbedaan proses *training* yang tidak menggunakan dan yang menggunakan *early stopping function*.



**Gambar 3.18.** Grafik *Loss* Tidak menggunakan *Early Stopping Function*



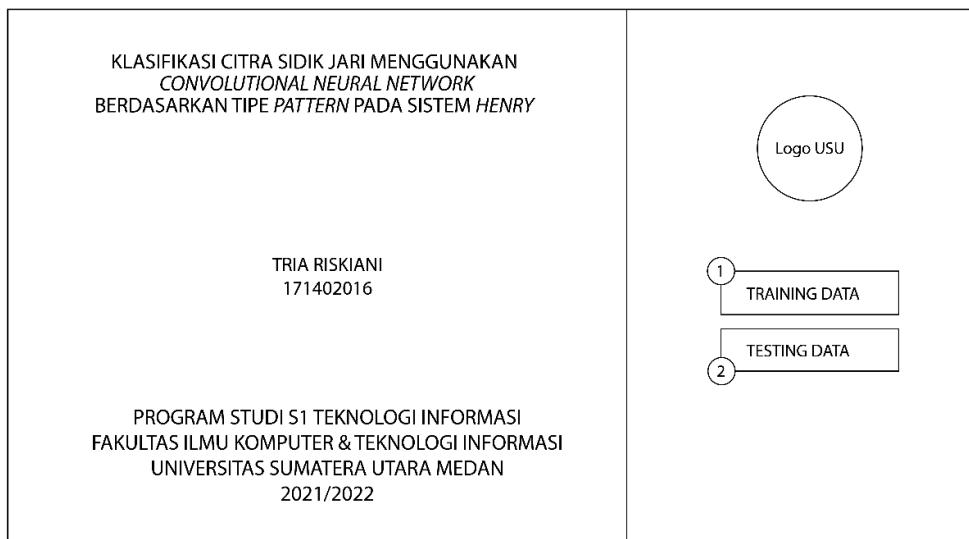
**Gambar 3.19.** Grafik Loss menggunakan *Early Stopping Function*

### 3.3. Perancangan Antarmuka Sistem

Perancangan tampilan antarmuka sistem bertujuan mendeskripsikan dan menggambarkan tentang sistem klasifikasi citra sidik jari berdasarkan pola pada sidik jari yang akan dibuat. Dalam tahap perancangan antarmuka sistem berisikan 3 menu, yakni beranda, *training* dan *testing*.

#### 3.3.1. Rancangan Tampilan Beranda

Rancangan ini sebagai tampilan awal ketika sistem pertama kali diakses. Di halaman ini, terdapat informasi judul penelitian, nama, NIM, institusi penulis, serta tombol-tombol seperti tombol “*training*” dan “*testing*”. Gambar 3.20 menampilkan contoh rancangan tampilan beranda.



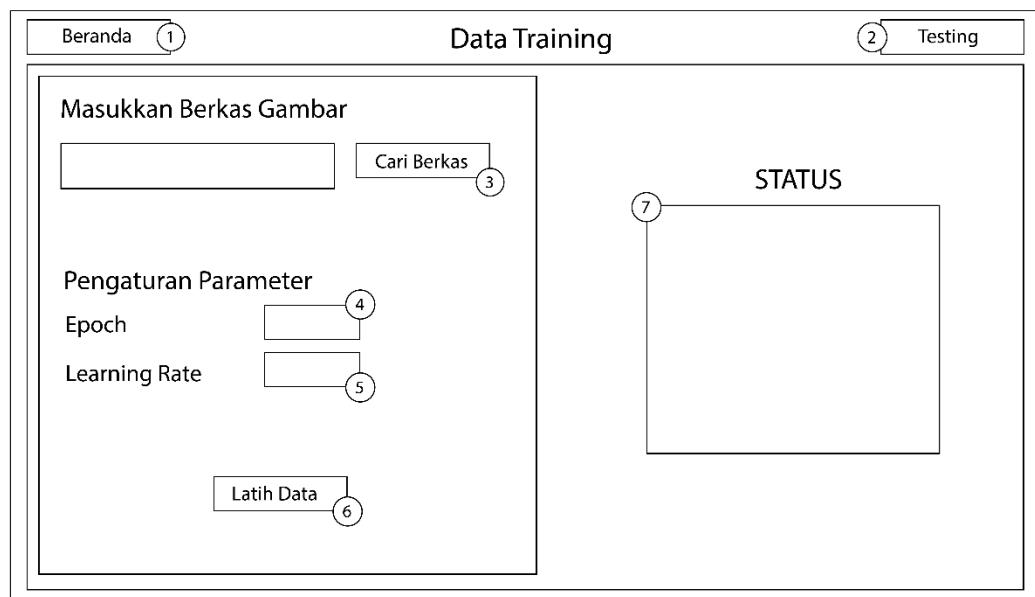
**Gambar 3.20.** Rancangan tampilan halaman beranda

Detail tampilan halaman beranda diantaranya:

1. Tombol *training data*, mengalihkan sistem ke halaman *training*.
2. Tombol *testing data*, mengalihkan sistem ke halaman *testing*.

### 3.3.2. Rancangan Tampilan Training

Bagian ini bertujuan untuk menjalankan proses latih data. Dalam halaman ini terdapat *form* yang berperan untuk menjalankan proses pelatihan atau *training*. Setelah selesai dilakukannya proses *training*, maka akan muncul informasi yang menyatakan bahwa tahapan-tahapan pada proses *training* telah selesai. Gambar 3.21 menampilkan contoh rancangan tampilan *training*.



**Gambar 3.21.** Rancangan tampilan halaman *training*

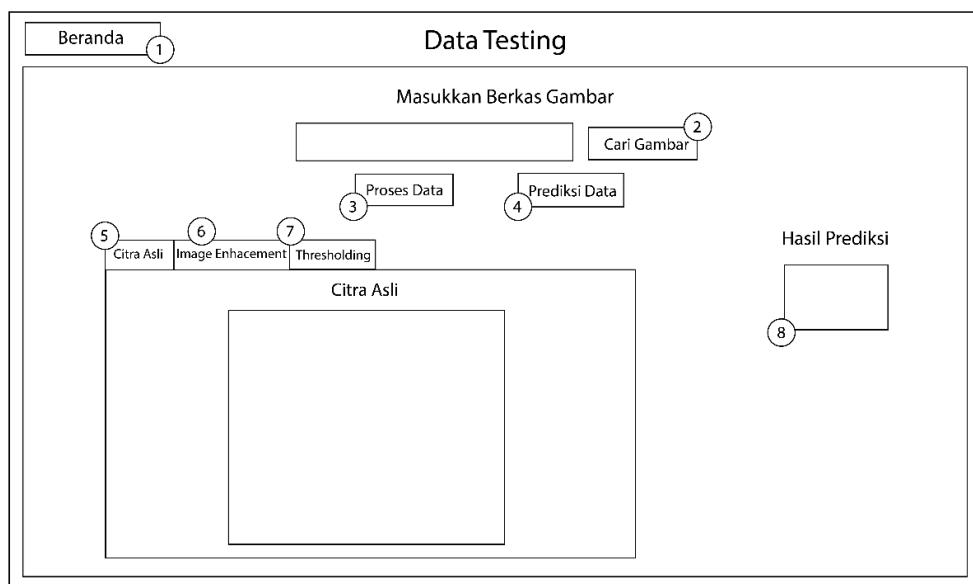
Detail dari rancangan tampilan halaman *training*, yakni:

1. Tombol Beranda, menampilkan tampilan awal pada sistem.
2. Tombol *Testing*, mengarahkan sistem ke halaman *testing*.
3. Tombol Cari Berkas, untuk mengunggah data citra yang akan dilatih.
4. *Epoch*, bagian ini memiliki fungsi untuk mengatur *epoch* yang akan digunakan.
5. *Learning Rate*, bagian ini memiliki fungsi untuk mengatur *learning rate* yang akan digunakan.

6. Tombol Latih data, untuk memulai proses pelatihan pada data yang telah dipilih atau di-input.
7. Status, bagian ini akan menampilkan status aktivitas *upload* data dan proses *preprocessing*.

### 3.3.3. Rancangan Tampilan Testing

Tahap pengujian atau *testing* dilakukan pada halaman ini. Data yang dapat di-input di halaman ini ialah data tunggal (*single data*). Pada tampilan antarmuka ini ada tombol “Pilih File” untuk meng-input data yang akan diklasifikasikan dengan format file “.png”. Proses klasifikasi dimulai dengan mengklik tombol “Proses Data”. Setelah tahap *testing* atau pengujian selesai, hasil dari proses pengujian akan ditampilkan di halaman ini, terlihat pada Gambar 3.22.



**Gambar 3.22.** Rancangan tampilan halaman *testing*

Detail rancangan antarmuka halaman *testing*, diantaranya:

1. Tombol Beranda, untuk menampilkan tampilan awal pada sistem.
2. Tombol Cari Gambar, untuk mengunggah data citra yang akan diuji.
3. Tombol Proses Data, memproses data citra uji untuk melewati tahap *preprocessing* dan segmentasi.
4. Tombol Prediksi Data, memproses data citra uji yang sudah melewati tahap *preprocessing* dan segmentasi untuk diklasifikasi.
5. Tab Citra Asli, menampilkan citra asli yang diunggah

6. *Image enhancement*, menampilkan hasil citra setelah melalui tahap *contrast enhancement*.
7. *Thresholding*, memperlihatkan hasil citra setelah melalui tahap *thresholding*.
8. Hasil Prediksi, menampilkan hasil dari klasifikasi citra.

### 3.4. Metode Evaluasi

Evaluasi diperlukan untuk mendapatkan pemahaman tentang keberhasilan dan keakuratan sistem dalam mengklasifikasikan citra pola sidik jari *arch*, *tented arch*, *right loop*, *left loop* dan *whorl*. Dalam penelitian ini metode evaluasi yang diaplikasikan adalah *confusion matrix*, yaitu salah satu metode yang dimanfaatkan untuk melakukan evaluasi terhadap metode-metode klasifikasi. Proses evaluasi menggunakan *confusion matrix* mengandung beberapa *performance metrics*, diantaranya *precision*, *recall*, *f1-score* dan *accuracy*.

*Accuracy* menggambarkan seberapa akurat model dapat melakukan klasifikasi dengan benar. Jadi akurasi adalah rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Akurasi ialah tingkat kedekatan nilai prediksi dengan nilai sebenarnya. Nilai akurasi dapat dihitung dengan Persamaan 3.1.

$$\text{Accuracy} = \frac{\sum TP_i}{\text{Jumlah Citra testing}} \quad (3.1)$$

*Precision* menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model. Jadi *precision* ialah perbandingan antara rasio prediksi benar positif dengan keseluruhan hasil yang diprediksi positif. Dari semua kelas positif yang telah diprediksi dengan benar, precision menunjukkan berapa banyak data yang benar-benar positif. *Precision* menjawab pertanyaan “Berapa persen sidik jari yang benar berpola *arch* dari keseluruhan sidik jari yang diprediksi berpola *arch*?”. Nilai *precision* dapat dihitung menggunakan persamaan 3.2.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3.2)$$

*Recall* menggambarkan kemampuan model dalam menemukan kembali informasi yang benar. Jadi *recall* adalah rasio antara prediksi benar positif dibandingkan dengan keseluruhan data yang benar-benar positif. *Recall* menjawab pertanyaan “Berapa persen sidik jari yang diprediksi berpola *arch* dibandingkan keseluruhan sidik jari yang sebenarnya berpola *arch*?”. Perhitungan nilai *recall* dapat dilakukan menggunakan Persamaan 3.3.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3.3)$$

*f1-score* merupakan rata-rata dari *precision* dan *recall*.

$$f1 - score = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall}+\text{Precision}} \quad (3.4)$$

*True Positive* (TP) mengacu pada data yang termasuk dalam kelas positif dan diklasifikasikan dengan benar. *True Negative* (TN) adalah data yang termasuk dalam kelas negatif dan diklasifikasikan dengan benar. *False Positive* (FP) mengarah pada data yang sebenarnya termasuk dalam kelas negatif tetapi salah diklasifikasikan sebagai kelas positif. *False Negative* (FN) ialah data yang diklasifikasi dengan kurang tepat, data yang sebenarnya termasuk dalam kelas positif salah diklasifikasikan sebagai kelas negatif.

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Implementasi Sistem

Rancangan sistem yang telah disebutkan dalam Bab 3 akan diimplementasikan melalui penggunaan berbagai perangkat keras dan perangkat lunak sebagai alat bantu dan sarana pendukung, termasuk diantaranya:

##### 4.1.1. Spesifikasi perangkat keras dan perangkat lunak

Spesifikasi perangkat keras yang digunakan dalam implementasi sistem dalam penelitian ini, meliputi:

1. Processor Intel CORE i5 8th Gen
2. Kapasitas Memory (RAM): 8 GB DDR 4

Spesifikasi perangkat lunak yang digunakan dalam implementasi sistem dalam penelitian ini, meliputi:

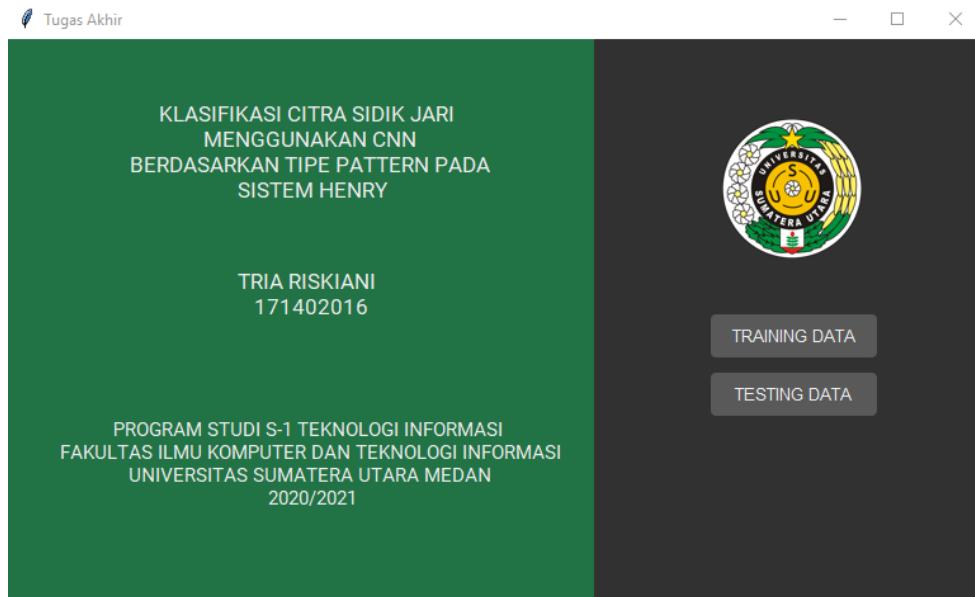
1. Sistem operasi yang digunakan *Windows 10 Home Single Languange 64-bit.*
2. *Python Version 3.6.8*
3. Beberapa contoh *Library* bahasa pemrograman *python* yang digunakan meliputi:
  - a. TensorFlow dan Keras
  - b. Pandas
  - c. Numpy
  - d. Matplotlib
4. Microsoft Visual Studio Code *Version 1.75.0*

#### *4.1.2. Implementasi Perancangan Antarmuka*

Implementasi antarmuka didasarkan pada perancangan sistem yang sudah diuraikan di Bab 3 dinyatakan seperti berikut:

##### *4.1.2.1. Tampilan halaman beranda*

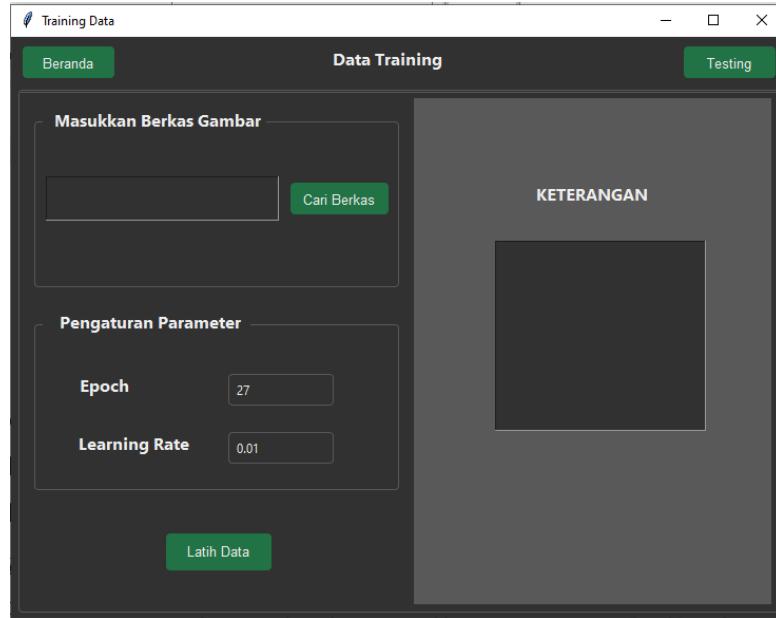
Laman beranda menjadi tampilan pertama ketika sistem dijalankan. Gambar 4.1 memperlihatkan tampilan halaman beranda.



**Gambar 4.1.** Tampilan Halaman Beranda

##### *4.1.2.2. Tampilan halaman training*

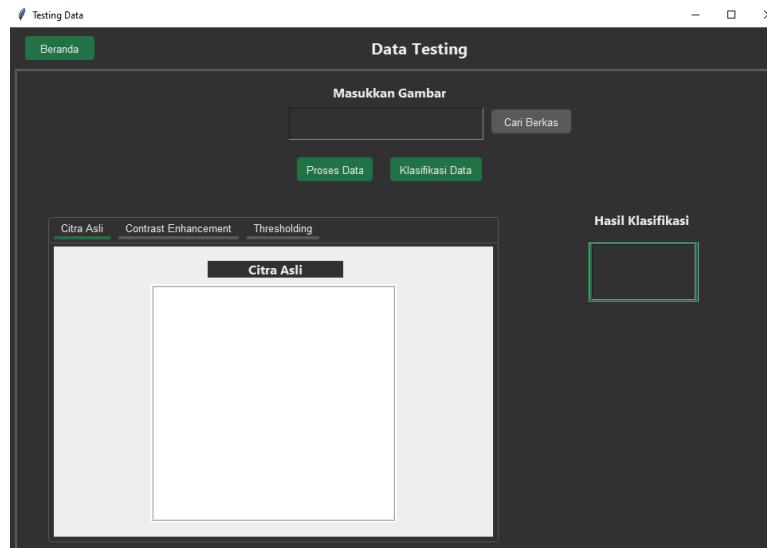
Halaman *training* yaitu halaman tempat mengunggah data *training* dalam format .png. Pada halaman ini tersedia dua pilihan *button* yakni “Cari Berkas” dan “Latih Data” serta terdapat *form* untuk pengaturan parameter seperti *learning rate* dan *epoch* yang digunakan dalam proses *training*. Selain itu ditampilkan juga rangkuman dari citra yang telah melalui tahap *pre-processing*. Ditunjukkan dalam Gambar 4.2 mengenai gambaran tampilan halaman *training*.



**Gambar 4.2.** Tampilan Halaman *Training*

#### 4.1.2.3. Tampilan halaman *testing*

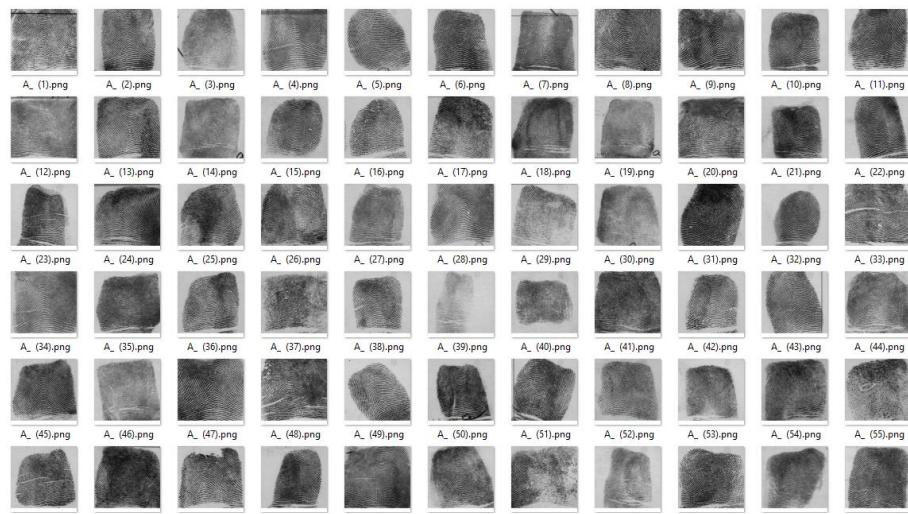
Pada menu *testing*, pengguna hanya dapat melakukan *input* satu citra sidik jari dengan ekstensi .png untuk kemudian diklasifikasi. Setelah data *testing* dimasukkan, hasil dari citra asli yang telah melalui tahap *contrast enhancement* dan *thresholding* serta hasil prediksi dari data uji yang diklasifikasi ditunjukkan di halaman ini. Gambar 4.3 menunjukkan halaman *testing* sebelum data di-*input*.



**Gambar 4.3.** Tampilan Halaman *Testing*

#### 4.1.3 Implementasi Data

Penelitian ini menggunakan data dari NIST *Fingerprint Database*. Citra yang digunakan adalah citra sidik jari dengan 5 kategori pola yakni *Arch*, *Left Loop*, *Right Loop*, *Tented Arch* dan *Whrol* yang berekstensi .png. Jumlah data latih yang digunakan sebanyak 1396 citra. Contoh citra latih sidik jari dapat tampak dalam Gambar 4.4, 4.5, 4.6, 4.7 dan 4.8.



**Gambar 4.4.** Data Citra *Training* Sidik Jari Pola *Arch*



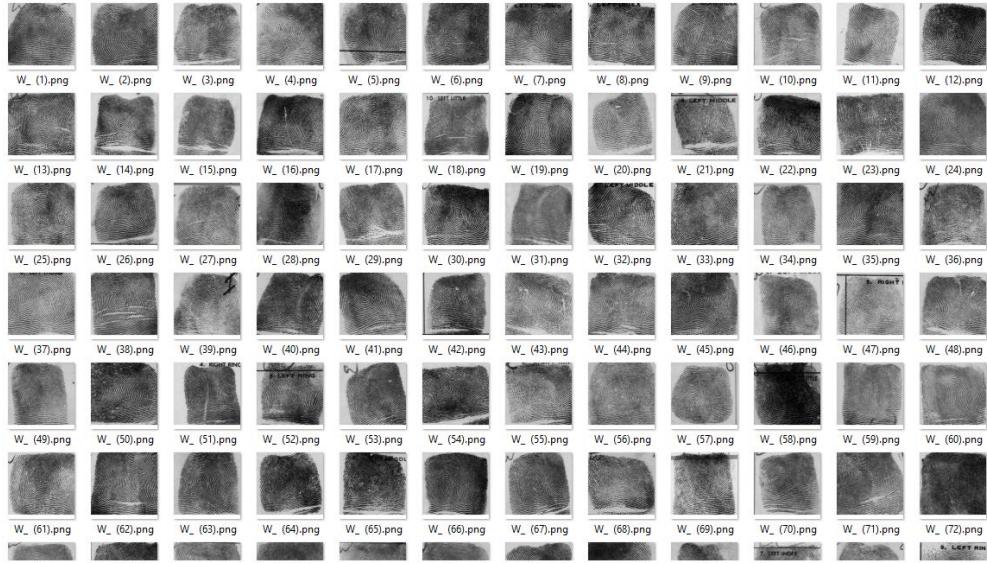
**Gambar 4.5.** Data Citra *Training* Sidik Jari Pola *Left Loop*



**Gambar 4.6.** Data Citra Training Sidik Jari Pola Right Loop

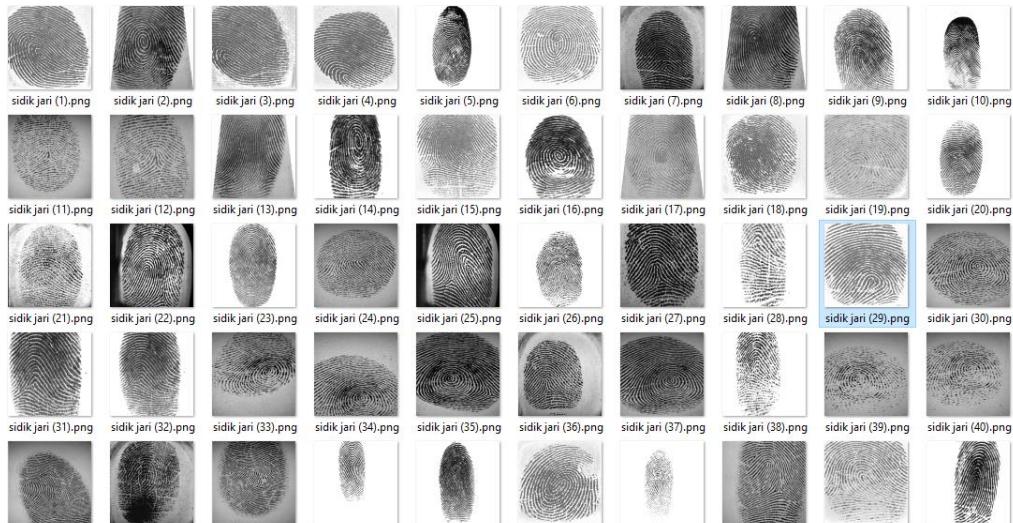


**Gambar 4.7.** Data Citra Training Sidik Jari Pola Tented Arch



**Gambar 4.8.** Data Citra *Training* Sidik Jari Pola Whrol

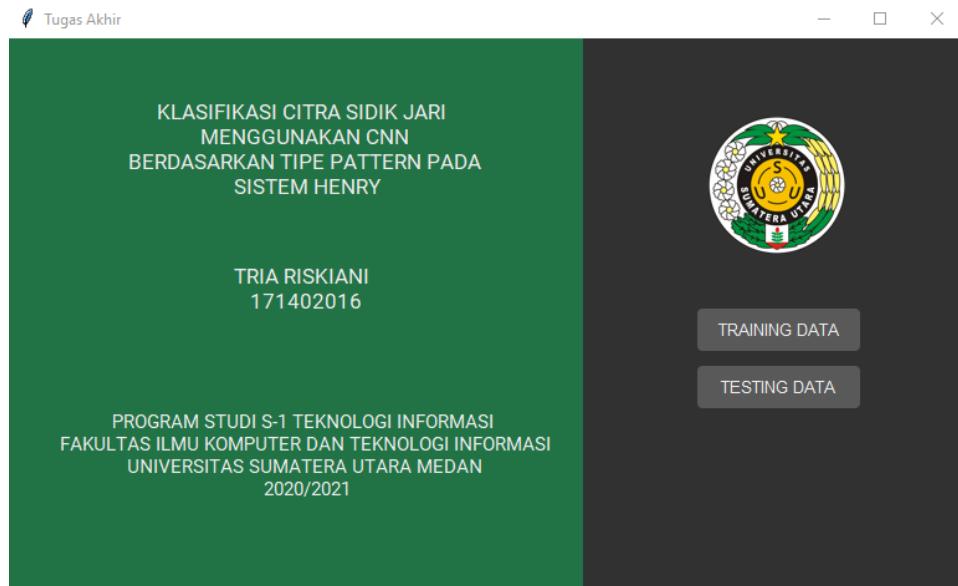
Jumlah citra yang digunakan dalam proses *testing* yaitu sebanyak 75 citra. Contoh citra uji sidik jari dapat tampak dalam Gambar 4.9.



**Gambar 4.9.** Data Citra *Testing*

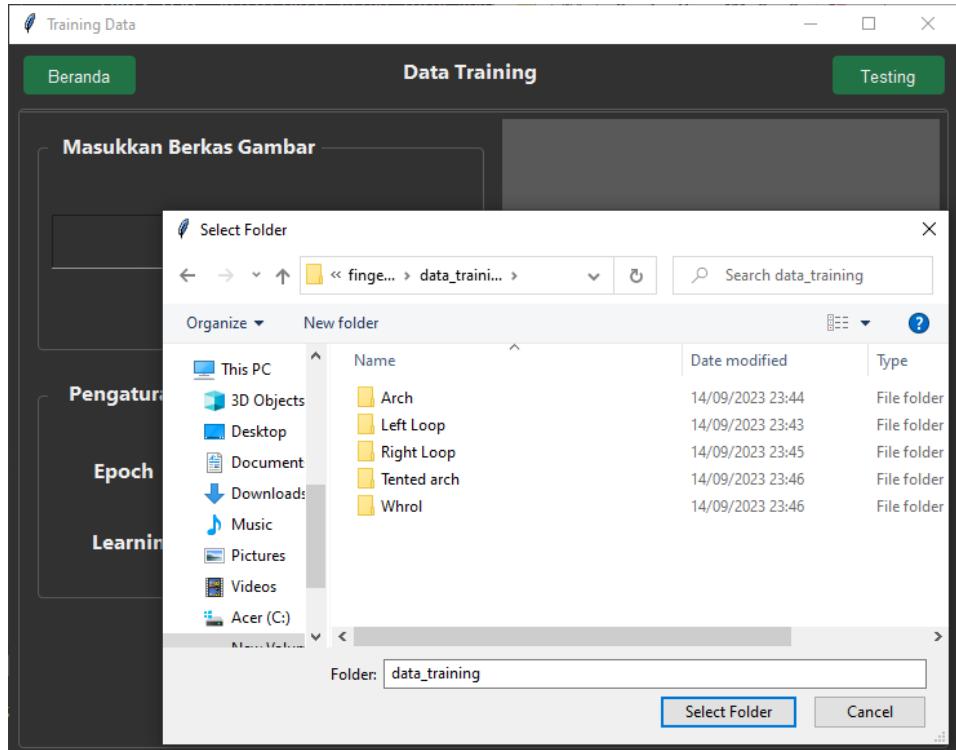
## 4.2. Prosedur Operasional

Sistem ini meliputi 3 tampilan antarmuka, yakni halaman beranda, *training* dan *testing*. Halaman beranda menjadi tampilan utama yang pertama kali ditampilkan ketika sistem dijalankan. Tampilan halaman beranda menampilkan sejumlah opsi yang berfungsi sebagai tautan menuju halaman lain dan mencakup rincian mengenai penulis serta judul dari penelitian yang sedang dilaksanakan. Terdapat pilihan untuk mengarahkan pengguna menuju halaman lain diantaranya tombol *training data* dan *testing data*. Gambar 4.10 ialah gambaran tampilan halaman beranda.

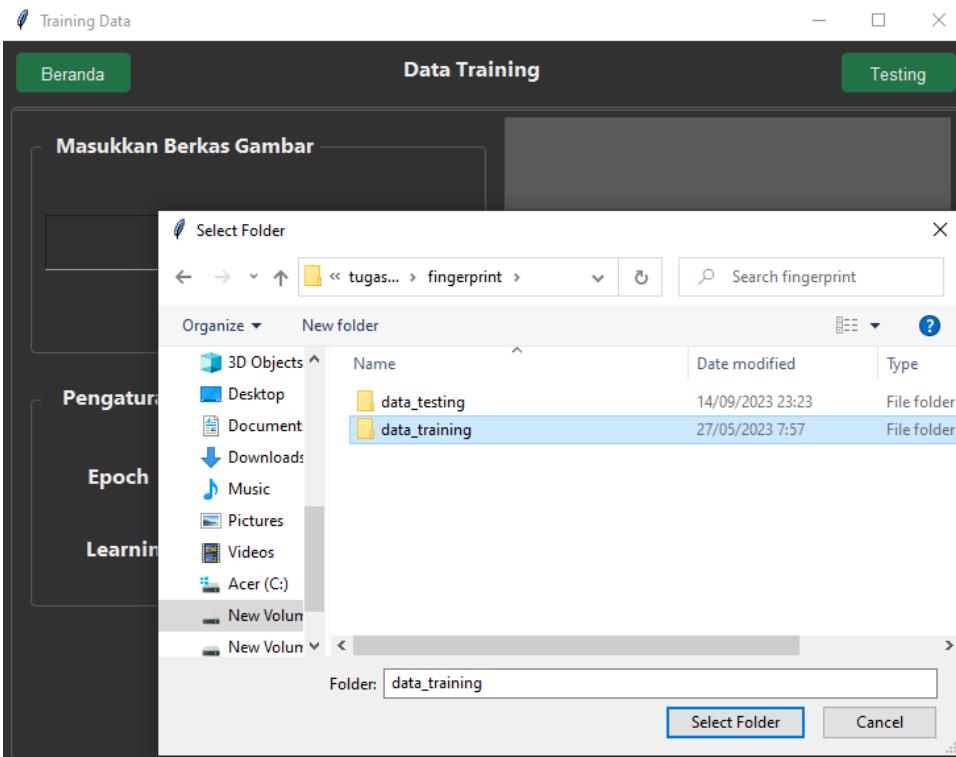


**Gambar 4.10.** Tampilan Halaman Beranda

Halaman *training* terdapat tombol Cari Folder untuk memasukkan data dalam bentuk folder yang akan dilakukan proses *training*. *Folder* citra yang akan digunakan untuk proses *training* berisikan 5 *subfolder* dari pola sidik jari yang berbeda-beda. Contoh *subfolder* tersebut terlihat pada Gambar 4.11.

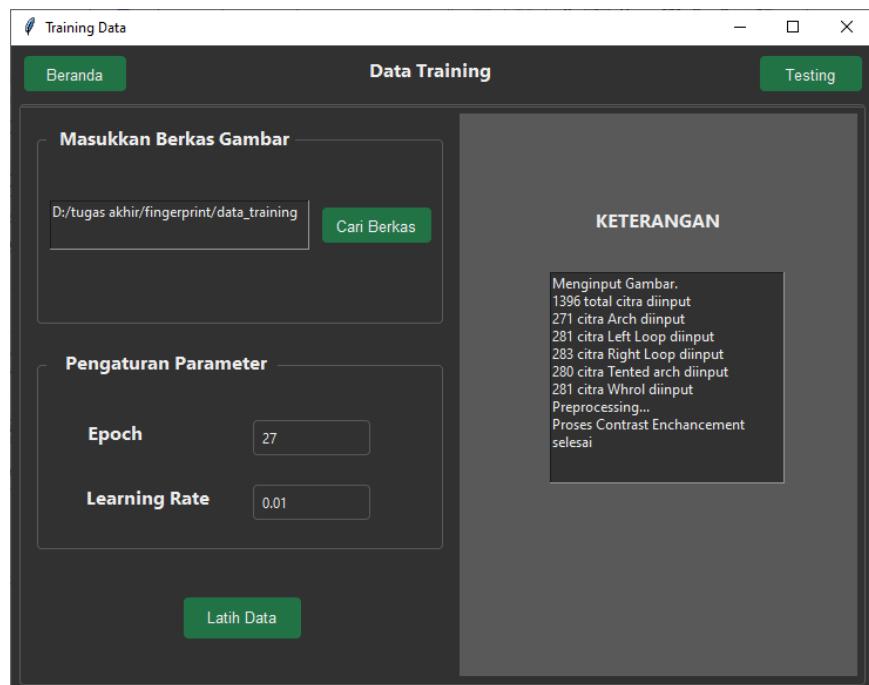


Gambar 4.11. Subfolder Folder Train



Gambar 4.12. Folder Training Citra Sidik Jari

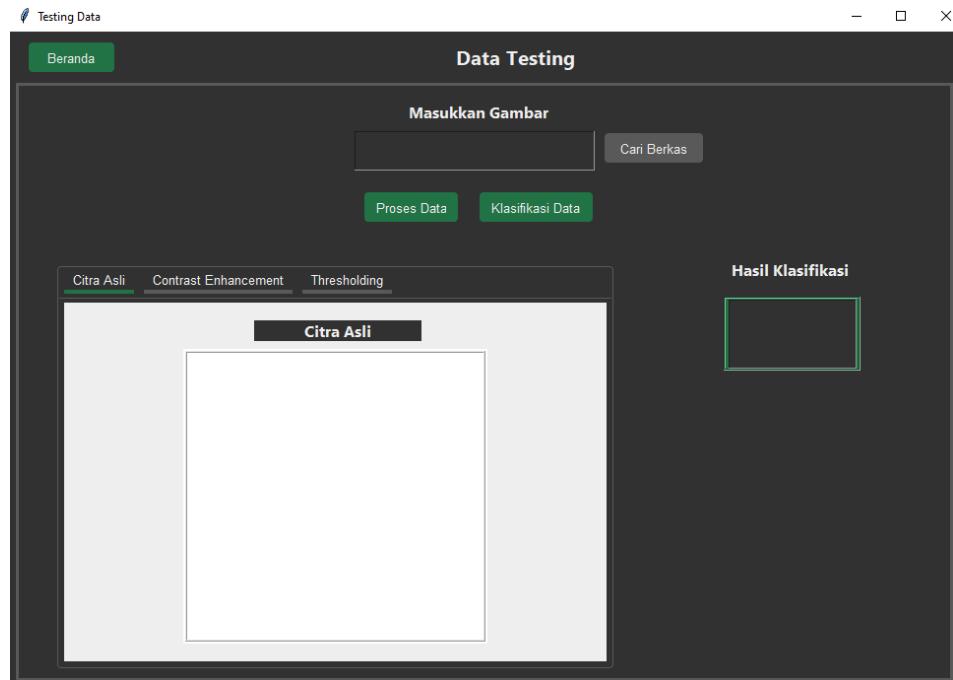
Setelah memilih data yang sesuai seperti pada Gambar 4.12, maka data akan melalui tahap *training* dengan memilih tombol Latih Data. Setelah data melewati tahap *training*, selanjutnya akan ditampilkan ringkasan dari citra yang telah melalui tahap *pre-processing*. Rangkuman tersebut berisi mengenai total semua citra yang di-*input*, jumlah dari masing-masing kategori pola sidik jari yang di-*input*, serta informasi bahwa proses *pre-processing* telah selesai yang diperlihatkan dalam Gambar 4.13.



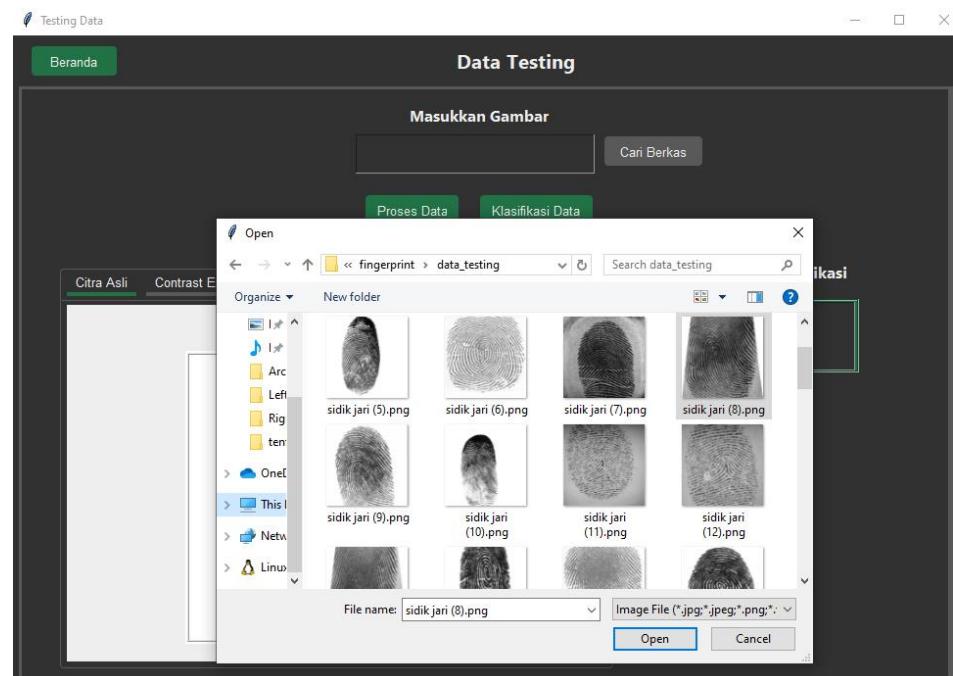
**Gambar 4.13.** Tampilan Halaman *Training* (Tahap *Pre-processing*)

Setelah menyelesaikan tahap *training* dengan metode CNN, berarti dihasilkanlah model yang dapat melakukan proses *testing*.

Gambar 4.14 menampilkan halaman *testing*, dimana akan dilakukan pengujian citra sidik jari untuk dilakukan klasifikasi. Halaman ini terdapat tombol Cari Berkas untuk memilih citra yang akan dilakukan klasifikasi oleh pengguna. Pada Gambar 4.15 terdapat tampilan *input* dari halaman *testing*.

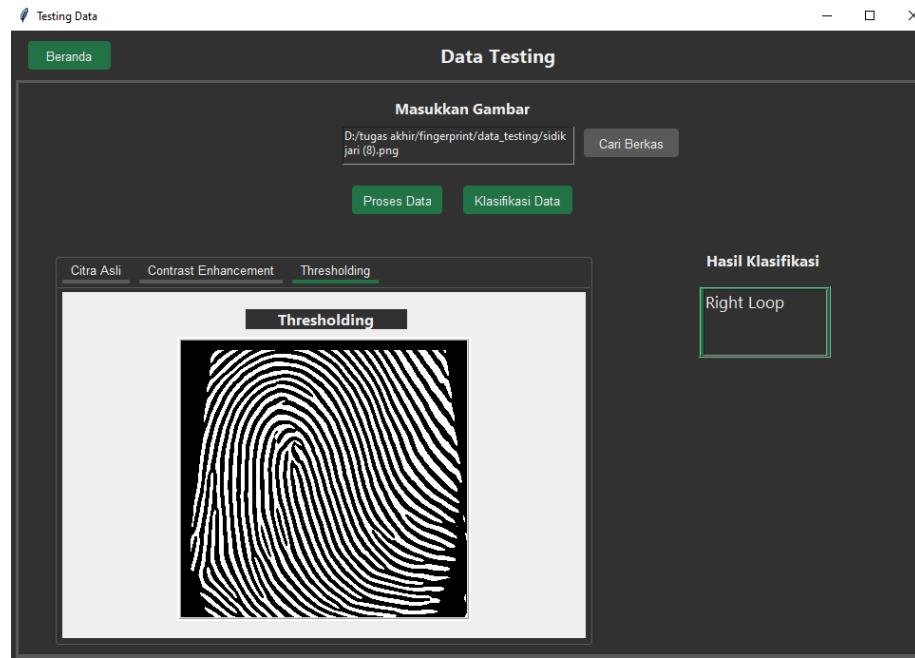


**Gambar 4.14.** Tampilan Halaman *Testing*



**Gambar 4.15.** Tampilan Halaman *Input Data Testing*

Bagian ini juga menunjukkan hasil klasifikasi dari citra yang sudah diolah. Ditampilkan hasil yakni citra asli yang di-*input* oleh pengguna, hasil citra yang sudah melewati tahap *contrast enhancement*, hasil citra yang sudah melewati tahap *thresholding* dan hasil prediksi dari pola sidik jari yang sudah diklasifikasi. Hasil klasifikasi pada pola sidik jari terdapat dalam Gambar 4.16.



**Gambar 4.16.** Tampilan Halaman *Testing* (Hasil Klasifikasi)

### 4.3. Implementasi Model

#### 4.3.1 Pelatihan Model

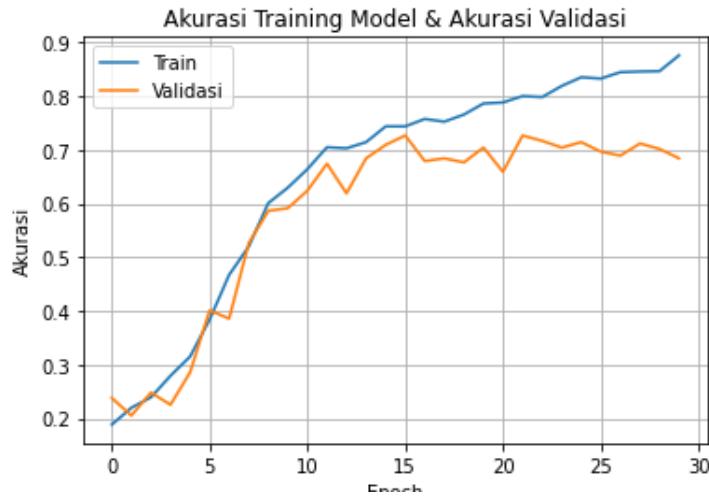
Dalam penelitian ini, dilakukan beberapa kali percobaan pada tahap pelatihan dengan mengubah jumlah *epoch* dan *batch size*, tujuannya adalah mencapai performa terbaik pada model. Sebuah model dikatakan memiliki performa terbaik apabila model telah mencapai akurasi tertinggi tetapi tidak *overfit*. Upaya untuk menghindari terjadinya *overfit*, pada Tabel 4.1. terlihat hasil dari dilakukanya beberapa percobaan menggunakan *epoch* dan *batch size* yang berbeda.

**Tabel 4.1.** Percobaan dengan beberapa epoch dan batch size

<i>Epoch</i>	<i>Batch Size</i>	<i>Accuracy</i>	<i>Loss</i>
20	32	0.7100	0.8025
	64	0.7300	0.6545
	128	0.7300	0.6251
	256	0.7150	0.6593
25	32	0.7250	0.6886
	64	0.8300	0.5139
	128	0.7850	0.5692
	256	0.7700	0.6541
30	32	0.9333	0.5228
	64	0.8933	0.4449
	128	0.8533	0.4384
	256	0.7867	0.6951

Dalam penelitian ini, digunakan 30 *epoch* dan ukuran *batch size* sebesar 128, menghasilkan 11 iterasi dalam 1 epoch. Pemilihan *epoch* dan *batch size* disesuaikan dengan data serta kapabilitas perangkat yang digunakan. *Learning rate* yang digunakan sebesar 0,01. Proses pelatihan pada model ditampilkan dalam Gambar 4.17 (Grafik Loss) serta 4.18 (Grafik Akurasi).

**Gambar 4.17.** Grafik Loss pada Pelatihan Model



**Gambar 4.18.** Grafik Accuracy pada Pelatihan Model

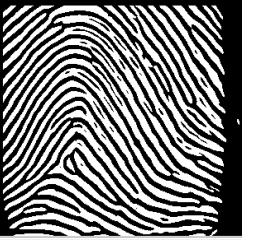
Melalui visualisasi hasil per *epoch*-nya, dapat dilihat apakah data mengalami *overfit* atau tidak. Idealnya, *training* dan validasi tidak boleh terlalu jauh selisihnya. Semakin kecil *loss* maka berarti semakin baik dan apabila akurasi semakin besar maka berarti semakin baik.

#### 4.4. Pengujian Sistem

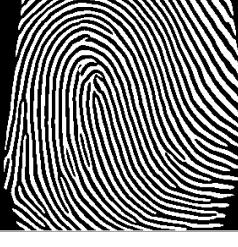
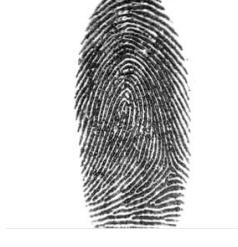
Pengujian dilakukan pada 75 citra sidik jari yang diambil dari FVC *Database*. Jumlah 75 citra untuk lima kategori pola sidik jari *Arch*, *Left Loop*, *right Loop*, *Tented Arch* dan *Whrol*.

Data uji akan melalui tahap *preprocessing* sesuai dengan yang telah diuraikan dalam Bab 3. Citra asli data uji akan ditingkatkan nilai kontrasnya dan akan dilakukan *thresholding* untuk memperjelas guratan-guratan sidik jari sehingga terlihat lebih jelas. Hasil *preprocessing* data uji citra sidik jari terlihat di Tabel 4.2, Tabel 4.3 serta Tabel 4.4.

**Tabel 4.2.** Hasil *Preprocessing* Citra

No	Pola Sidik Jari	Contrast Enhancement	Thresholding
1			
2			
3			
4			

**Tabel 4.3.** Hasil *Preprocessing* Citra (Lanjutan)

No	Pola Sidik Jari	Contrast Enhancement	Thresholding
5			
6			
7			
8			
9			

**Tabel 4.4.** Hasil *Preprocessing* Citra (Lanjutan)

No	Pola Sidik Jari	Contrast Enhancement	Thresholding
10			
12			

Data uji yang sudah melalui tahap *preprocessing* selanjutnya akan diklasifikasi oleh model yang sudah dilatih sebelumnya. Hasil dari pengujian sistem klasifikasi citra sidik jari menggunakan metode *Convolutional Neural Network* berdasarkan tipe *pattern* pada Sistem Henry dapat diamati di Tabel 4.5 hingga Tabel 4.6.

**Tabel 4.5.** Hasil Pengujian Citra pada Sistem

No	Pola Sidik Jari	Nama Citra	Hasil Prediksi
1		<i>Arch</i>	<i>Arch</i>
2		<i>Arch</i>	<i>Arch</i>
3		<i>Left Loop</i>	<i>Left Loop</i>
4		<i>Right Loop</i>	<i>Right Loop</i>
5		<i>whrol</i>	<i>whrol</i>

**Tabel 4.6.** Hasil Pengujian Citra pada Sistem (Lanjutan)

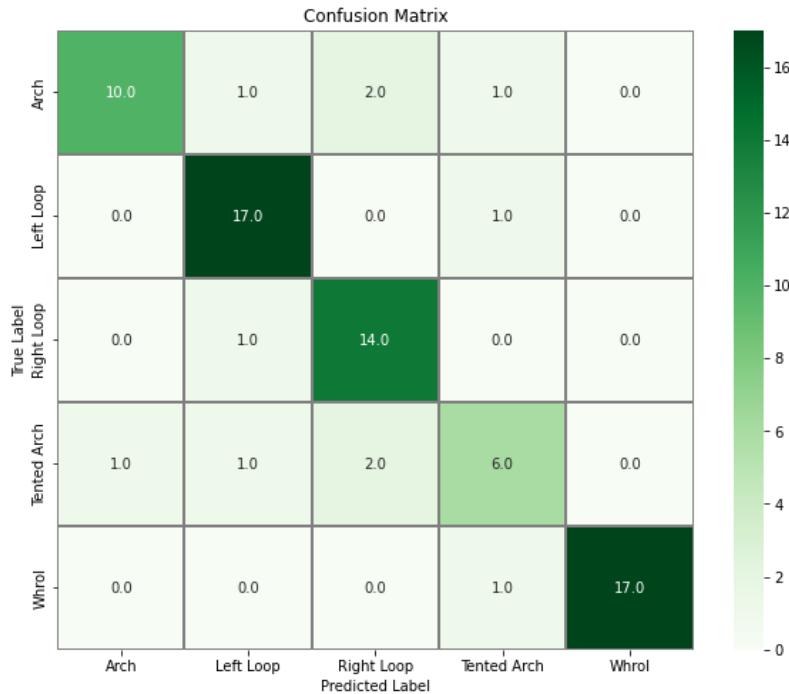
No	Pola Sidik Jari	Nsma Citra	Hasil Prediksi
6		<i>Tented Arch</i>	<i>Tented Arch</i>
7		<i>Tented Arch</i>	<i>Tented Arch</i>
8		<i>Right Loop</i>	<i>Right Loop</i>
9		<i>Left Loop</i>	<i>Left Loop</i>
10		<i>Whrol</i>	<i>Whrol</i>

**Tabel 4.7.** Hasil Pengujian Citra pada Sistem (Lanjutan)

No	Pola Sidik Jari	Nama Citra	Hasil Prediksi
11		<i>Right Loop</i>	<i>Right Loop</i>
...	.....	.....	.....
74		<i>Whrol</i>	<i>Whrol</i>
75		<i>Whrol</i>	<i>Whrol</i>

#### 4.5. Evaluasi

Evaluasi performa model dilakukan melalui perhitungan akurasi dengan menggunakan *confussion matrix*. Dibutuhkan jumlah data yang telah diprediksi dengan benar untuk menghitung nilai akurasi, sebagaimana yang terlihat dalam Gambar 4.17 (*Confussion Matrix*). Gambar 4.19 menunjukkan pengukuran pada performa model.



**Gambar 4.19.** Confusion Matrix

Dari Gambar 4.19 diperoleh nilai *True Positive* jumlah data yang diprediksi benar, *False Positive* yakni jumlah data positif yang diprediksi salah dan *False Negative* untuk jumlah data negatif yang diprediksi salah.

**Tabel 4.8.** Nilai *True Positive*, *False Positive* dan *False Negative*

	<b>TP</b>	<b>FP</b>	<b>FN</b>
Arch	10	1	4
Left Loop	17	3	1
Right Loop	14	4	1
Tented Arch	6	3	4
Whrol	17	0	1

Perhitungan *precision*, *recall* dan *f1-score*

- a. Pola sidik jari arch

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\% = \frac{10}{10+1} \times 100\% = 90\%$$

$$Recall = \frac{TP}{TP+FN} \times 100\% = \frac{10}{10+4} \times 100\% = 71\%$$

$$f1-score = \frac{2 \times precision \times recall}{precision + recall} \times 100\% = \frac{2 \times 0.90 \times 0.71}{0.90 + 0.71} \times 100\% = 79\%$$

b. Pola sidik jari *Left Loop*

$$Precision = \frac{TP}{TP+FP} \times 100\% = \frac{17}{17+3} \times 100\% = 85\%$$

$$Recall = \frac{TP}{TP+FN} \times 100\% = \frac{17}{17+1} \times 100\% = 94\%$$

$$f1-score = \frac{2 \times precision \times recall}{precision + recall} \times 100\% = \frac{2 \times 0.85 \times 0.94}{0.85 + 0.94} \times 100\% = 89\%$$

c. Pola sidik jari *Right Loop*

$$Precision = \frac{TP}{TP+FP} \times 100\% = \frac{14}{14+4} \times 100\% = 77\%$$

$$Recall = \frac{TP}{TP+FN} \times 100\% = \frac{14}{14+1} \times 100\% = 93\%$$

$$f1-score = \frac{2 \times precision \times recall}{precision + recall} \times 100\% = \frac{2 \times 0.77 \times 0.93}{0.77 + 0.93} \times 100\% = 84\%$$

d. Pola sidik jari *Tented Arch*

$$Precision = \frac{TP}{TP+FP} \times 100\% = \frac{6}{6+3} \times 100\% = 67\%$$

$$Recall = \frac{TP}{TP+FN} \times 100\% = \frac{6}{6+4} \times 100\% = 60\%$$

$$f1-score = \frac{2 \times precision \times recall}{precision + recall} \times 100\% = \frac{2 \times 0.67 \times 0.60}{0.67 + 0.60} \times 100\% = 63\%$$

e. Pola sidik jari *Whrol*

$$Precision = \frac{TP}{TP+FP} \times 100\% = \frac{17}{17+0} \times 100\% = 100\%$$

$$Recall = \frac{TP}{TP+FN} \times 100\% = \frac{17}{17+1} \times 100\% = 94\%$$

$$f1-score = \frac{2 \times precision \times recall}{precision + recall} \times 100\% = \frac{2 \times 1 \times 0.94}{1 + 0.94} \times 100\% = 97\%$$

Kemudian untuk mendapatkan nilai akurasi dari hasil pengujian, dilakukan perhitungan metrik *accuracy* berdasarkan persamaan 3.1 untuk mendapatkan tingkat akurasi model dalam melakukan klasifikasi citra sidik jari berdasarkan tipe *pattern* pada Sistem Henry.

$$Accuracy = \frac{TP Arch + TP Left Loop + TP Right Loop + TP Tented Arch + TP Whrol}{Jumlah Citra testing}$$

$$Accuracy = \frac{10+17+36+6+17}{75}$$

$$Accuracy = 0,85$$

$$Percentase Accuracy = 0,85 \times 100\% = 85\%$$

Dari perhitungan yang telah dilakukan terlihat bahwa akurasi yang diperoleh dari penelitian ini adalah 85%. Berdasarkan nilai akurasi tersebut, dapat ditarik kesimpulan bahwa model *Convolutional Neural Network* mampu melakukan pengklasifikasian citra sidik jari berdasarkan tipe *pattern* pada sistem henry dengan cukup baik.

Pada penelitian ini terdapat beberapa citra yang diklasifikasikan kurang tepat pada sistem ini. Ada faktor yang menyebabkan hal tersebut, yaitu guratan (*ridge*) pada citra sidik jari tidak terlalu jelas yang ditampilkan dalam Gambar 4.20.



**Gambar 4.20.** *ridge* pada sidik jari tidak terlalu jelas

Kemudian pola pada citra sidik jari terlihat mirip sehingga model *convolutional neural network* mendeteksi pola tersebut lebih menyerupai pola sidik jari lain yang kemudian menyebabkan adanya salah dikenali oleh sistem seperti sidik jari pada Gambar 4.21. Sistem mengklasifikasikan sidik jari ke dalam pola *whorl* yang seharusnya masuk ke kelas *left loop*.



**Gambar 4.21.** Contoh Citra yang gagal diidentifikasi oleh *CNN* karena memiliki pola yang hampir sama

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan penelitian yang telah dilakukan, pengujian klasifikasi citra sidik jari menggunakan *Convolutional Neural Network* berdasarkan tipe *pattern* pada sistem Henry dapat diambil kesimpulan bahwa:

1. Metode *Convolutional Neural Network* yang diterapkan dalam mengklasifikasi pola sidik jari *arch*, *left loop*, *right loop*, *tented arch* dan *whorl* dengan jumlah data latih 1396 citra dan data uji 75 citra berhasil mencapai tingkat akurasi sebesar 85%.
2. Pemilihan nilai *epoch* dan *learning rate* sangat berpengaruh pada tingkat keakuratan sistem. Penelitian ini menggunakan *epoch* sebesar 30, *batch size* 128 dan *learning rate* 0,01.
3. Berdasarkan hasil evaluasi Bab 4, maka model *Convolutional Neural Network* mempunyai hasil yang cukup baik dalam mengklasifikasi citra sidik jari berdasarkan tipe *pattern* pada Sistem Henry.

#### **5.2. Saran**

Dalam proses pengeraannya pada penelitian ini masih ditemukan hal-hal yang memerlukan perbaikan. Oleh karena itu, beberapa saran untuk penelitian di masa depan yakni:

1. Dataset yang digunakan bersifat dinamis sehingga model dapat mengklasifikasikan pola sidik jari secara langsung.

2. Sistem dapat mengklasifikasikan citra pola sidik jari dalam jumlah tertentu dalam proses pengujian data, tidak hanya dapat mengklasifikasikan satu citra saja.
3. Menggunakan teknik pengolahan citra yang lainnya agar citra menjadi lebih bagus kualitasnya dan dapat menampilkan guratan (*ridge*) sidik jari lebih jelas untuk perbandingan klasifikasi yang lebih baik dari hasil yang telah didapatkan pada penelitian ini.
4. Menggunakan metode *neural network* lainnya agar dapat menjadi perbandingan antara hasil klasifikasi citra sidik jari yang dihasilkan dari *Convolutional Neural Network* dengan metode lainnya.

## DAFTAR PUSTAKA

- Fadlia, N., & Kosasih, R. 2020. Klasifikasi Jenis Kendaraan Menggunakan Metode Convolutional Neural Network (Cnn). *Jurnal Ilmiah Teknologi dan Rekayasa*, 24(3): 207-215.
- Harjoseputro, Y. 2018. Classifying Javanese Letters with Convolutional Neural Network (CNN) Method. In *The First Internasional Conference and Exibition on Sciences and Technology (ICEST)2018* pp. 52.
- Kumar, T., & Verma, K. 2010. A Theory Based on Conversion of RGB image to Gray image. *International Journal of Computer Applications* 7(2): 7-10.
- Kusmiati, D. 2010. Klasifikasi Citra Sidik Jari Dengan Metode Templete Matching. (Doctoral dissertation, Mathematics and Natural science).
- Kusumanto, R. D., & Tompunu, A. N. 2011. pengolahan citra digital untuk mendeteksi obyek menggunakan pengolahan warna model normalisasi RGB. *Semantik* I(1).
- Maulana, F.F., & Rochmawati, N. 2019. Klasifikasi Citra Buah Menggunakan Convolutional Neural Network. *Journal of Informatics and Computer Science* 1(2): 104-108
- Pradana, I. H. 2015. Klasifikasi Citra Sidik Jari Berdasarkan Enam Tipe Pattern Menggunakan Metode Euclidean Distance. *Inform. Jur. Tek. Komputer, Fak. Ilmu Nuswantoro, Univ. Dian* 1-5.
- Rahmawati, S. N., Hidayat, E. W., & Mubarok, H. 2021. Implementasi Deep Learning Pada Pengenalan Aksara Sunda Menggunakan Metode Convolutional Neural Network. *INSERT: Information System and Emerging Technology Journal* 2(1): 46-58.
- Sharma, A., & Phonsa, G. 2021. Image Classification Using CNN. In *Proceedings of the International Conference on Innovative Computing & Communication (ICICC)*.
- Sharma, N., Jain, V., & Mishra, A. 2018. An analysis of convolutional neural networks for image classification. *Procedia computer science* 132: 377-384.

- Sediyono, E., Nataliani, Y., & Rorimpandey, C. M. 2011. Klasifikasi Sidik Jari dengan Menggunakan Metode Wavelet Symlet. *Jurnal Informatika* 5(2).
- Suartika, I. W. E. P., Wijaya, A. R & Soelaiman, R. 2016. Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS* Vol. 5 No. 1.
- Sudiro, S. A., & Lukman, S. 2019. Keakuratan Deteksi Keaslian Sidik Jari Menggunakan Metode Cnn. *Prosiding SeNTIK* 3(1).
- Sulistiyasni, S., & Winarko, E. 2014. Klasifikasi Pola Sidik Jari Menggunakan Jaringan Syaraf Tiruan Backpropagation. *BIMIPA* 24(3): 215-224.
- Wang, R., Han, C., Wu, Y., & Guo, T. 2014. Fingerprint classification based on depth neural network. *arXiv preprint arXiv* 1409.5188.
- Win, K. N., Li, K., Chen, J., Viger, P. F., & Li, K. 2020. Fingerprint classification and identification algorithms for criminal investigation: A survey. *Future Generation Computer Systems* 110: 758-771.