

**EKSTRAKSI CIRI PADA TELAPAK TANGAN DENGAN METODE  
*LOCAL BINARY PATTERN (LBP)***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Dwi Retnoningrum

NIM: 155150200111167



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019

# PENGESAHAN

EKSTRAKSI CIRI PADA TELAPAK TANGAN DENGAN METODE LOCAL BINARY  
PATTERN (LBP)

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Dwi Retnoningrum  
NIM: 155150200111167

Skripsi ini telah diuji dan dinyatakan lulus pada  
2 Januari 2019

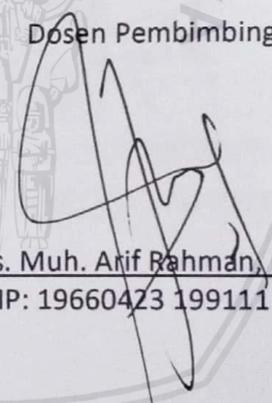
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Agus Wahyu Widodo, S.T, M.Cs  
NIP: 19740805 200112 1 001



Drs. Muh. Arif Rahman, M.Kom  
NIP: 19660423 199111 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Fitri H. Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 02 Januari 2019



Dwi Retnoningrum

NIM: 155150200111167



## PRAKATA

Dengan menyebut nama Allah SWT Yang Maha Pengasih Lagi Maha Penyayang. Syukur alhamdulillah, penulis berterima kasih atas segala limpahan rizki yang diberikan oleh Allah SWT, sehingga penelitian sebagai tugas akhir skripsi yang berjudul “Penerapan Metode LBP (*Local Binary Patter*) Untuk Ekstraksi Ciri Pada Telapak Tangan” dapat diselesaikan tepat pada waktunya. Tak lupa shawat serta salam penulis sampaikan kepada Nabi Muhammad SAW, beserta keluarga dan para sahabat, karena tanpa ajaran Islam, ilmu pengetahuan tidak akan berkembang.

Selama pelaksanaan penelitian pada skripsi ini, penulis banyak mendapatkan dukungan dari banyak pihak. Oleh sebab, itu, penulis menyampaikan rasa terima kasih kepada:

1. Bapak Agus Wahyu Widodo, S.T., M.Cs selaku dosen pembimbing I dan ketua Program Studi Teknik Informatika serta Bapak Drs. Muh. Arif Rahman, M.Kom selaku dosen pembimbing II yang telah menyediakan waktu untuk membimbing penulis dalam menyelesaikan laporan ini.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku ketua Jurusan Teknik Informatika.
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer.
4. Seluruh dosen dan karyawan Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya mengajarkan, membagikan ilmu yang bermanfaat, dan membantu penulis selama masa perkuliahan dan pengerjaan skripsi
5. Kedua orang tua saya dan kakak tercinta yang selalu hadir dalam suka maupun duka serta senantiasa mendoakan yang terbaik untuk saya.
6. Teman-teman kuliah (Andi Amaliyah M, Gilang Widiyanto, Farid Ulil , Jonathan Rovelano, M. Aji Taufan, M.Hanafi, Aryo Dimas, Bimo Rizki, Tri Kurniawan P., Kurnia A., I Gede Satya K. , Pengkuh A.) yang telah memberika dukungan motivasi dan dukungan moril selama pengerjaan penelitian skripsi.
7. Rekan seperjuangan (RR.Dea Annisayanti P., Nur Faiqoh, Angelika T., Grace T., Lita handayani) yang senantiasa berbagi ilmu.
8. Tim Futsal Putri Fakultas Ilmu Komputer khususnya Novy Annisa, Hanaria Rotua, Dinda Hudaifa, Vira Indriana, Azkia Samara Devi, Olivia Very, Winda Ika yang selalu memberikan doa dan dukungan.
9. Anggota Lab.Sistem Komputer dan Robotika yang memberikan pengetahuan *softskill* dan berorganisasi..
10. Pihak-pihak lain yang secara langsung maupun tidak langsung membantu dalam penyelesaian skripsi.

Penulis menyadari bahwa dalam masa penyusunan skripsi ini masih banyak kekurangan, dengan itu penulis sangat mengharapkan saran dan kritik yang membangun untuk skripsi ini. Semoga skripsi ini dapat membawa manfaat bagi semua pihak dan digunakan sebagaimana mestinya.

Malang, 02 Januari 2019

Penulis

[dwiretningrum3@gmail.com](mailto:dwiretningrum3@gmail.com)



## ABSTRAK

**Dwi Retnoningrum, Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP)**

**Pembimbing: Agus Wahyu Widodo, S.T., M.Cs. dan Dr. Drs. Muh. Arif Rahman, M.Kom.**

Ekstraksi ciri dapat dilakukan pada objek dalam bentuk citra dengan menggunakan beberapa fitur. Ekstraksi ciri dapat dipadukan dengan ilmu biometrik. Biometrik merupakan karakteristik fisik atau biologi dan tingkah laku yang unik serta terukur. Identifikasi biometrik dapat dimanfaatkan untuk meningkatkan keamanan maupun menghindari penggunaan identitas palsu. Dalam penelitian ini, penulis menggunakan telapak tangan sebagai objek penelitian dikarenakan telapak tangan memiliki fitur unik yang berbeda pada setiap individu. Selain fitur unik, luas permukaan telapak tangan menjadi salah satu pertimbangan penulis dalam menentukan objek penelitian. Luas permukaan telapak tangan lebih besar jika dibandingkan dengan luas permukaan salah satu jari. Salah satu metode yang dapat dimanfaatkan dalam proses identifikasi adalah metode ekstraksi ciri LBP (*Local Binary Pattern*) yang menerapkan jarak ketetanggaan dan jumlah tetangga yang dibandingkan. Dimulai dengan tahapan *Pre-processing* atau tahap persiapan citra berwarna yang akan dirubah menjadi citra keabuan kemudian dilanjutkan dengan proses *regioning* atau proses pembagian citra menjadi beberapa *sub-region*. Dilanjutkan dengan tahap ekstraksi ciri dengan metode LBP. Pada penelitian ini citra yang digunakan berjumlah 208 citra yang diperoleh dari *College of Engineering Pune* dengan 26 individu yang masing-masing memiliki 8 citra. Hasil akurasi tertinggi yang diperoleh dari penelitian ini sebesar 92,31% dengan jarak ketetanggaan 2, jumlah tetangga yang dibandingkan = 8, jumlah region = 16 dan jumlah pembagian *height* = 4 dan *width* = 4.

**Kata kunci:** ekstraksi ciri, biometrik, LBP, telapak tangan

## ABSTRACT

**Dwi Retnoningrum, Feature Extraction On Palm With *Local Binary Pattern* (LBP) Method**

**Supervisors: Agus Wahyu Widodo, S.T., M.Cs. dan Dr. Drs. Muh. Arif Rahman, M.Kom.**

*Feature extraction can be done on an objects in the form of images using several features. Feature extraction can be combined with the science of biometrics. Biometrics are unique and measurable physical or biological characteristics. Biometric identification can be used to improve security and avoid using fake identities. In this case, the author uses the palm of the object of research because palm has unique features that are different for each individual. In addition to unique features, the palm surface area is one of the authors' considerations in determining the object of research. The surface area of the palm is greater than the surface area of one finger. One method that can be utilized in the identification process is the LBP (Local Binary Pattern) feature extraction method that applies neighboring distances and the number of neighbors compared. It starts with the Pre-processing stage or the preparation stage of the color image which will be transformed into a gray image and then followed by a regioning process or image sharing process into several sub-regions. Followed by feature extraction stages with the LBP method. In this study the image used amounted to 208 images obtained from the Pune College of Engineering with 26 individuals each of which had 8 images. The highest accuracy results obtained from this study amounted to 92.31% with neighboring distance 2, number of neighbors compared = 8, number of regions = 16 and number of shares height = 4 and width = 4.*

**Keywords:** *feature extraction, biometric, LBP, palm.*

## DAFTAR ISI

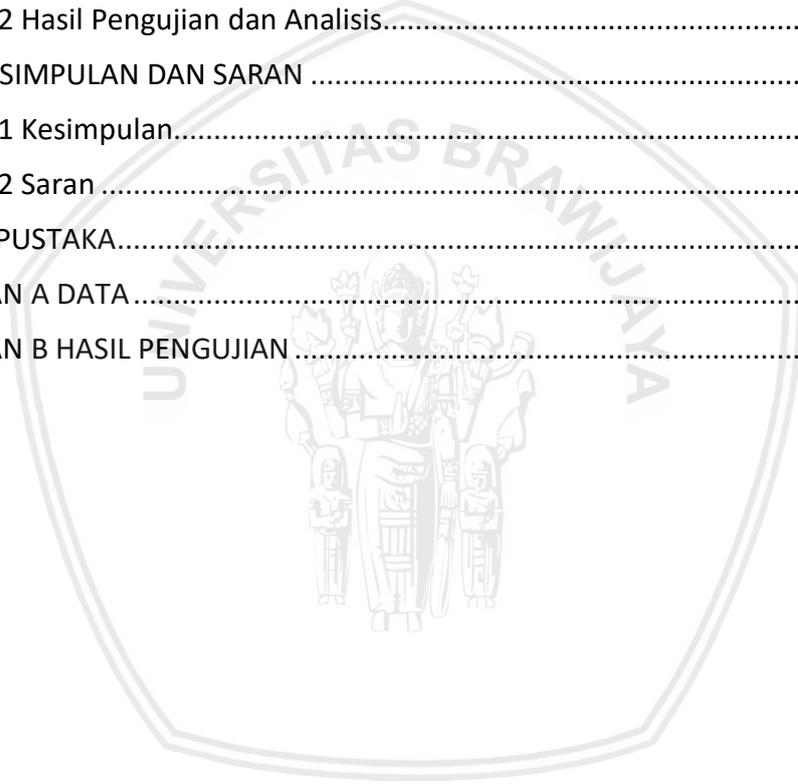
PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN .....	xiii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan masalah .....	3
1.6 Sistematika pembahasan.....	3
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>5</b>
2.1 Tinjauan Pustaka .....	5
2.2 Telapak Tangan (Palmprint).....	7
2.3 Biometrik.....	8
2.4 Ekstraksi Tekstur .....	9
2.5 Grayscale.....	9
2.6 Local Binary Pattern (LBP).....	9
2.7 Histogram.....	11
2.8 Normalisasi .....	11
2.9 K-NN (K- Nearest Neighbour).....	12
2.10 Pengukuran Jarak Euclidean .....	12
2.11 Perhitungan Akurasi.....	13
<b>BAB 3 METODOLOGI .....</b>	<b>14</b>
3.1 Tipe Penelitian .....	14



3.2 Strategi Penelitian.....	14
3.3 Peralatan Pendukung.....	15
3.3.1 Perangkat Keras Pendukung .....	15
3.3.2 Perangkat Lunak Pendukung.....	15
3.4 Pengumpulan Data .....	15
3.5 Lokasi Penelitian .....	16
3.6 Perancangan Algoritme .....	16
3.7 Pengujian dan Analisis .....	17
3.8 Penarikan Kesimpulan dan Saran .....	19
<b>BAB 4 PERANCANGAN.....</b>	<b>20</b>
4.1 Perancangan Algoritme .....	20
4.1.1 Pre-Processing.....	22
4.1.2 Regioning.....	22
4.1.3 Ekstraksi Ciri LBP .....	23
4.1.4 Hitung Frekuensi .....	25
4.1.5 Normalisasi.....	25
4.1.6 Konkatenasi.....	26
4.1.7 Klasifikasi K-NN.....	27
4.2 Perhitungan Manual .....	28
4.2.1 <i>Grayscale</i> .....	29
4.2.2 Regioning.....	30
4.2.3 Local Binary Pattern .....	31
4.2.4 Hitung Frekuensi .....	33
4.2.5 Normalisasi.....	33
4.2.6 Konkatenasi.....	34
4.2.7 K-NN .....	34
<b>BAB 5 IMPLEMENTASI .....</b>	<b>37</b>
5.1 Batasan Implementasi .....	37
5.2 Implementasi Algoritme .....	37
5.2.1 <i>Pre-Processing</i> .....	37
5.2.2 Algoritme Regioning.....	38
5.2.3 Algoritme Local Binary Pattern .....	38



5.2.4 Hitung Frekuensi .....	40
5.2.5 Normalisasi.....	40
5.2.6 Konkatenasi.....	41
5.2.7 Klasifikasi K-NN.....	41
<b>BAB 6 PENGUJIAN DAN ANALISIS.....</b>	<b>42</b>
6.1 Pengujian .....	42
6.1.1 Pengujian Jumlah <i>Region</i> dan Nilai <i>Height</i> dan <i>Width</i> .....	42
6.1.2 Pengujian Jarak Tetangga.....	42
6.2 Hasil Pengujian dan Analisis.....	43
<b>BAB 7 KESIMPULAN DAN SARAN .....</b>	<b>48</b>
7.1 Kesimpulan.....	48
7.2 Saran .....	48
DAFTAR PUSTAKA.....	49
LAMPIRAN A DATA .....	51
LAMPIRAN B HASIL PENGUJIAN .....	51



## DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka .....	5
Tabel 3.1 Jumlah <i>Region</i> dan Pembagian <i>Height</i> dan <i>Width</i> .....	17
Tabel 4.1 Nilai dari Layer <i>Red</i> .....	28
Tabel 4.2 Nilai dari Layer <i>Green</i> .....	28
Tabel 4.3 Nilai dari Layer <i>Blue</i> .....	29
Tabel 4.4 Nilai dari <i>Grayscale</i> .....	30
Tabel 4.5 <i>height = 2, width = 2</i> .....	30
Tabel 4.6 <i>height = 3, width = 2</i> .....	31
Tabel 4.7 Frekuensi Histogram.....	33
Tabel 4.8 Normalisasi .....	34
Tabel 4.9 Hasil Perhitungan Jarak .....	35
Tabel 4.10 Hasil Pengurutan Jarak .....	35
Tabel 4.11 Hasil Pengambilan 5 Teratas .....	36
Tabel 6.1 Hasil Pengujian $R = 1, P = 8$ .....	43
Tabel 6.2 Hasil Pengujian $R = 2, P = 8$ .....	44
Tabel 6.3 Hasil Pengujian $R = 3, P = 8$ .....	44
Tabel 6.4 Keterangan .....	46

## DAFTAR GAMBAR

Gambar 2.1 <i>Palmprint</i> .....	8
Gambar 2.2 <i>Cropping</i> Telapak Tangan .....	8
Gambar 2.3 Jarak dan jumlah tetangga yang dipilih.....	10
Gambar 2.4 <i>Thresholding</i> .....	11
Gambar 3.1 Alur Proses dalam Model .....	14
Gambar 3.2 Alur Algoritme .....	16
Gambar 4.1 Diagram Alir Algoritme .....	20
Gambar 4.2 Diagram Alir (a) <i>Training</i> (b) <i>Testing</i> .....	21
Gambar 4.3 <i>Pre-Processing</i> .....	22
Gambar 4.4 <i>Regioning</i> .....	23
Gambar 4.5 Ekstraksi Ciri LBP .....	24
Gambar 4.6 Hitung Histogram .....	25
Gambar 4.7 Normalisasi .....	26
Gambar 4.8 Konkatenasi .....	26
Gambar 4.9 Klasifikasi K-NN.....	27
Gambar 4.10 Sub Region.....	31
Gambar 4.11 Lokasi Tetangga .....	32
Gambar 4.12 <i>Thresholding</i> .....	32
Gambar 4.13 Penyusunan biner.....	32
Gambar 4.14 Hasil LBP .....	33
Gambar 4.15 Hasil Konkatenasi .....	34
Gambar 5.1 <i>Source Code Pre-Processing</i> .....	37
Gambar 5.2 <i>Source Code Regioning</i> .....	38
Gambar 5.3 <i>Source Code Local Binary Pattern</i> .....	39
Gambar 5.4 <i>Source Code</i> hitung frekuensi.....	40
Gambar 5.5 <i>Source Code</i> normalisasi.....	41
Gambar 5.6 <i>Source Code</i> Klasifikasi K-NN .....	41
Gambar 5.7 <i>Source Code</i> Klasifikasi K-NN .....	41

## DAFTAR LAMPIRAN

LAMPIRAN A DATA .....	51
LAMPIRAN B HASIL PENGUJIAN .....	51



## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

Sistem pengenalan individu merupakan sebuah sistem yang digunakan untuk mengenali atau mengetahui identitas individu (*user authentication*). *User authentication* sendiri merupakan sistem keamanan informasi yang melindungi data dan hak cipta atas informasi yang mereka miliki (Dharavath, Talukdar and Laskar, 2013). Pada awalnya sistem tersebut masih menggunakan kata sandi yang memiliki kekurangan yang mana pengguna sering lupa kata sandinya dan kata sandi tersebut mudah ditebak oleh orang lain sehingga timbul masalah baru. Contoh masalah tersebut antara lain adalah pencurian identitas, penggunaan identitas palsu, pelanggaran privasi, penipuan dan masalah-masalah lain yang merugikan orang lain (Bhattacharyya et al., 2009). Dengan demikian pengenalan individu berbasis biometrik diperkenalkan untuk mengatasi masalah tersebut.

Identifikasi biometrik muncul karena sebagai sarana yang kuat untuk mengenali identitas seseorang secara otomatis (Zhang et al., 2003). Biometrik adalah ilmu pengetahuan yang dapat mengenali identitas individu berdasarkan ciri kimia, fisik ataupun perilaku individu tersebut (Jain, Nandakumar and Nagar, 2008). Sistem biometrik memanfaatkan fitur seperti suara, iris, sidik jari, telapak tangan, wajah dan lain lain (Han, Guo and Zhang, 2008). Fitur tersebut bersifat unik dari lahir hingga meninggal dan tidak ada yang memiliki bentuk yang sama meskipun pada orang kembar (Ahmed, 2015). Salah satu contoh sistem yang menggunakan pengenalan biometrik adalah pada absensi menggunakan sidik jari (*fingerprint*).

Dalam kasus pengenalan individu menggunakan sidik jari hanya menggunakan sidik jari sebagai parameter tunggal, namun ditemukan berbagai ketidak sempurnaan seperti jari kotor terkena noda, jari tergores, jari terdapat bekas luka bakar, kulit jari terkelupas (ganti kulit), jari dalam keadaan basah (Wibawa, Wirayuda and Hevanie, 2014). Untuk meminimalisir ketidak sempurnaan tersebut maka penulis menggunakan bagian telapak tangan untuk menjadi parameter pengenalan individu, karena telapak tangan memiliki luas permukaan yang lebih luas apabila dibandingkan dengan luas permukaan jari tangan. Telapak tangan digunakan karena memiliki permukaan yang lebih luas apabila dibandingkan dengan jari tangan. Selain luas permukaan, telapak tangan juga memiliki hasil biometrik yang lebih stabil (Basya et al., 2012). Salah satu proses dalam identifikasi biometrik adalah ekstraksi ciri yang dapat dilakukan pada objek dalam bentuk citra dengan menggunakan beberapa fitur. Fitur-fitur tersebut dapat menjadi parameter yang nantinya akan digunakan untuk membedakan individu satu dengan individu lainnya.

Dalam penelitian ini penulis menggunakan ekstraksi ciri berupa tekstur karena dianggap cocok dengan objek telapak tangan. Selain itu tekstur merupakan fitur dari citra yang sangat kuat untuk *computer vision* dan pengenalan pola (Abdesselam, 2013). Setiap objek di dunia nyata mempunyai tekstur yang berbeda

(Abdesselam, 2013). Sejumlah besar teknik yang dirancang yaitu untuk mendeskripsikan, mengklasifikasikan serta mengambil fitur. Ekstraksi fitur dapat dilakukan dengan menggunakan berbagai metode seperti LBP, *Haar Wavelet*, GLCM dan lain-lain.

Metode ekstraksi ciri yang digunakan adalah metode *Local Binary Pattern* (LBP) karena dianggap sebagai metode yang telah terbukti efektif untuk ekstraksi tekstur serta invarian terhadap pencahayaan yang berbeda (Ojala, Pietikäinen and Harwood, 1996). Pada sistem pengenalan wajah dengan menggunakan metode LBP berjalan dengan baik dan memiliki tingkat akurasi sebesar 100% (Mujib, Hidayatno and Prakoso, 2018). Sama halnya penelitian yang dilakukan untuk pengenalan *finger vein* menggunakan *Local Binary Pattern* memiliki tingkat akurasi sebesar 90% (Sari and Saputra, 2017). Untuk mengidentifikasi atau mencocokkan telapak tangan yang sama dengan menggunakan metode klasifikasi.

Klasifikasi adalah sebuah proses yang dimanfaatkan untuk menemukan fungsi atau model yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas suatu objek yang labelnya tidak diketahui (Bustami, 2014). Klasifikasi disebut metode *supervised learning* karena *pattern* yang memiliki kelas telah diketahui dan dijadikan data latih untuk mengidentifikasi objek berada di kelas mana. Ada beberapa metode klasifikasi yang dapat digunakan antara lain yaitu K-NN (*K-Nearest Neighbour*), SVM (*Super Vector Machine*), LDA (*Linear Discriminant Analysis*), FCM (*Fuzzy C Means*), Naïve Bayes, dan lain-lain. K-NN adalah metode klasifikasi yang terkenal dalam *machine learning* karena memiliki algoritma yang cukup sederhana. Metode ini menggunakan fungsi jarak untuk melakukan klasifikasi, salah satu contoh perhitungan jarak yaitu *Euclidean* yang digunakan untuk membandingkan sampel data lebih dekat dengan sampel data lain pada kelas tertentu (Amyntarto, Sari and Cahyawihandika, 2018).

Pada penelitian sebelumnya yang dilakukan oleh Hanif Ibadurrahman yang berjudul Biometrik Berbasis Telapak Tangan Menggunakan Metode *Local Binary Pattern* (LBP) dan Propagasi Balik Levenberg Marquardt menyarankan untuk menggunakan metode LBP dan membagi citra menjadi empat *region*. Setiap *region* akan dilakukan proses LBP, sehingga akan didapatkan empat histogram dan kemudian dilakukan konkatenasi histogram karena dengan cara ini ciri yang didapatkan akan menjadi semakin banyak.

Dengan mempertimbangkan masalah yang ada dan kelebihan dari metode LBP serta saran membagi citra menjadi beberapa *region* maka penulis akan melakukan penelitian mengenai penerapan Metode *Local Binary Pattern* (LBP) untuk ekstraksi ciri pada telapak tangan. Serta dalam penelitian ini penulis ingin menggunakan K-NN sebagai metode klasifikasinya.

## 1.2 Rumusan masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya maka dapat dirumuskan permasalahan sebagai berikut.

1. Bagaimana pengaruh pembagian citra menjadi beberapa *region* pada metode *Local Binary Pattern* (LBP) untuk ekstraksi ciri pada telapak tangan.
2. Bagaimana pengaruh parameter metode *Local Binary Pattern* (LBP) untuk ekstraksi ciri ada telapak tangan.

### 1.3 Tujuan

Tujuan penulis dari penelitian ini adalah.

1. Untuk mendapatkan pengaruh pembagian citra menjadi beberapa *region* pada penerapan metode *Local Binary Pattern* (LBP) untuk ekstraksi ciri pada telapak tangan.
2. Mendapat nilai parameter terbaik dari metode *Local Binary Pattern* (LBP).

### 1.4 Manfaat

Manfaat penelitian dapat diuraikan sebagai berikut.

1. Mempermudah analisis dalam ekstraksi ciri pada telapak tangan.
2. Sebagai salah satu sarana referensi metode untuk penelitian pada bidang ekstraksi ciri pada objek telapak tangan.

### 1.5 Batasan masalah

Pada penelitian ini masalah dan ruang lingkup dibatasi oleh:

1. Data yang digunakan pada penelitian ini yaitu berupa data sekunder yang diambil dari database College of Engineering, Pune.
2. Tidak menggunakan seluruh bagian tangan, melainkan hanya menggunakan bagian telapak tangan.
3. Citra yang digunakan merupakan rekaman dari telapak tangan bagian kanan.
4. Sudah dilakukan proses *cropping* manual untuk data yang akan digunakan.
5. Data berupa citra tegak atau vertikal.
6. Menggunakan citra berukuran 640 x 480 pixel.
7. *Input* berupa citra telapak tangan.
8. Hanya menguji salah satu parameter dari *Local Binary Pattern* yaitu jarak tetangga.

### 1.6 Sistematika pembahasan

Untuk memberikan gambaran tentang penelitian ini, maka penulis menuliskan gambaran besar pembahasan dari keseluruhan dari isi bab sebagai berikut.

#### BAB I : PENDAHULUAN

Pada bab ini berisikan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika penulisan.

**BAB II : DASAR TEORI**

Pada bab ini membahas penelitian-penelitian sebelumnya yang menguraikan teori-teori yang berkaitan dengan telapak tangan, biometrik, ekstraksi tekstur, *Grayscale*, *Local Binary Pattern*, Histogram, normalisasi, K-NN, pengukuran jarak, perhitungan akurasi.

**BAB III METODE PENELITIAN**

Pada bab ini berisi metode yang diharapkan dapat memberi solusi dari permasalahan yang telah dijelaskan pada latar belakang.

**BAB VI PERANCANGAN**

Pada bab ini membahas mengenai perancangan algoritme ekstraksi ciri pada telapak tangan dengan metode *Local Binary Pattern* (LBP)

**BAB V IMPLEMENTASI**

Pada bab ini membahas tentang implementasi yang telah dilakukan dan siap untuk bisa dilakukan proses pengujian dan analisis.

**BAB VI PENGUJIAN DAN ANALISIS**

Pada bab ini berisi pengujian data yang telah diperoleh, setelah itu akan dilakukan proses analisis pada telapak tangan dengan menggunakan metode *Local Binary Pattern* (LBP).

**BAB VII PENUTUP**

Pada bab ini berisi kesimpulan yang telah didapatkan dari penelitian yang telah dilakukan, serta saran yang diberikan untuk penelitian berikutnya.

## BAB 2 LANDASAN KEPUSTAKAAN

Bab ini terdiri dari kajian pustaka dan pembahasan dasar teori. Tinjauan pustaka membahas penelitian yang telah dilakukan sebelumnya dan memiliki topik atau obyek yang sama namun menggunakan metode yang berbeda dengan penelitian yang akan dilakukan. Pada dasar teori akan dijelaskan mengenai teori yang mendasari penelitian.

### 2.1 Tinjauan Pustaka

Tinjauan pustaka pada penelitian ini akan membahas penelitian sebelumnya yang dapat mendukung pengerjaan penelitian Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP). Pada penelitian yang pertama akan membahas biometrik telapak tangan dan memanfaatkan *Local Binary Pattern* sebagai metode untuk mengekstraksi fitur. Klasifikasi yang digunakan yaitu untuk mencocokkan fitur yaitu *propagasi balik levenberh marquardt*. Hasil yang tertinggi dari percobaan ini yaitu memiliki akurasi sebesar 86.67% dengan nilai jarak tetangga adalah 3 (Ibadurrahman, Wirayuda and Hevanie, 2014). Serta jumlah cek poin mempengaruhi hasil (Wibawa, Wirayuda and Hevanie, 2014).

Penelitian mengenai telapak tangan juga bisa menggunakan metode lain, contohnya *haar wavelet* dan *Gray Level Co-occurent Matrix* (GLCM). Pemanfaatan metode *Haar Wavelet* sendiri bisa dilakukan dengan berbagai cara yang pertama menggabungkan *data-level* dengan *min-max strategy* untuk menggabungkan *image* (Han, Guo and Zhang, 2008), kedua memanfaatkan fitur CA, CH, CV, CD atau di *paper* lain disebut sebagai LL, LH, LV, CD (Ahmed, 2015), dan (Basya et al., 2012). Metode *Gray Level Co-occurent Matrix* (GLCM) mengubah citra *grayscale* ke bentuk matriks dan menentukan piksel tetangga berupa sudut  $0^{\circ}$ ,  $45^{\circ}$ ,  $90^{\circ}$ ,  $135^{\circ}$  dan jarak  $d = 1$ . Dan kemudian mengambil 4 ekstraksi fitur yaitu *contrast*, *correlation*, *energy* dan *homogeinity* serta mengklasifikasikan dengan K-NN dengan nilai  $k = 1, 3, 5, 7$ , dan 9. Penelitian terdahulu ditunjukkan oleh Tabel 2.1.

**Tabel 2.1 Tinjauan Pustaka**

No	Judul	Metode Ekstraksi Fitur			Klasifikasi	Hasil
		LBP	Wavelet	GLCM		
1	Biometrik Berbasis Telapak Tangan Menggunakan Metode Local Binary Pattern (LBP) dan Propagasi Balik Levenberg	V			Propagasi Balik Levenberg Marquardt	Jarak tetangga terbaik adalah 3, akurasi tertinggi yaitu sebesar 86,67%

Tabel 2.1 Tinjauan Pustaka(lanjutan)

No	Judul	Metode Ekstraksi Fitur			Klasifikasi	Hasil
		LBP	Wavelet	GLCM		
	Marquardt (Ibadurrahman, Wirayuda and Hevanie, 2014)					
2	<i>Multispectral Palmprint Recognition using Wavelet – based Image Fusion</i> (Han, Guo and Zhang, 2008)		V		Tidak dijelaskan	Hasil menunjukkan bahwa algoritma wavelet – based image fusion efektif pada <i>image blurr</i>
3	Pengenalan Ciri Garis Telapak Tangan Menggunakan Ekstraksi Fitur (GLCM) dan Metode K-NN(Purnamasari et al., 2017)			V	K-NN	Akurasi sebesar 92.3 %
4	Palmprint Recognition with Statistical, Wavelet and Local Feature Extraction Methods(Ahmed, 2015)		V		K-NN	Akurasi 76.59%
5	Sistem Biometrik Berbasis Pola Telapak Tangan Dengan Menggunakan Metode <i>Local Binary Pattern</i> Dan	V			<i>Probabilistic Neural Network</i>	Semakin banyak jumlah cek poin maka hasil semakin

**Tabel 2.1 Tinjauan Pustaka(lanjutan)**

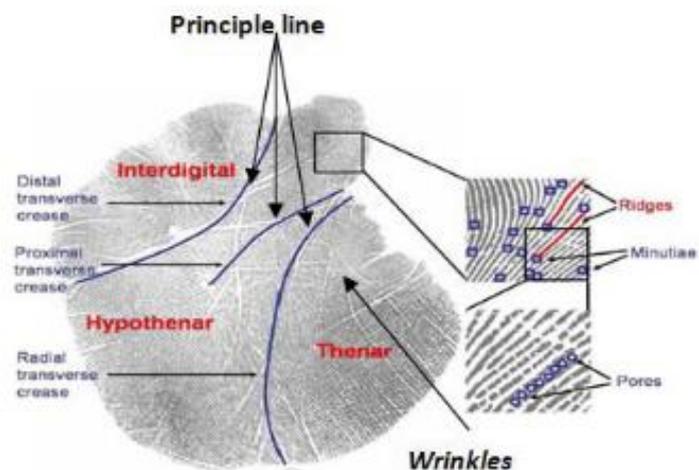
No	Judul	Metode Ekstraksi Fitur			Klasifikasi	Hasil
		LBP	Wave let	GLCM		
	<i>Probabilistic Neural Network</i> (Wibawa, Wirayuda and Hevanie, 2014)					bagus
6	Sistem Identifikasi Biometrik Telapak Tangan Menggunakan Metode <i>Haar Wavelet</i> dan <i>Support Vector Machine</i> (Basya et al., 2012)		V		<i>Support Vector Machine</i>	Akurasi yang di dapat adalah 97%
7	Diajukan	V ( <i>Regioning</i> + LBP)			K-NN	

## 2.2 Telapak Tangan (Palmprint)

Telapak tangan adalah permukaan bagian dalam tangan yang terletak di antara pergelangan tangan dan jari-jari. Telapak tangan adalah biometrik yang dapat diandalkan dan memiliki sifat unik dimana telapak tangan dapat ditangkap oleh perangkat beresolusi rendah serta mengandung fitur tambahan seperti garis utama (*principal lines*), kerutan (*wrinkle*), dan *ridge* (Manegopale, 2014). Oleh sebab itu telapak tangan sangat cocok untuk mengidentifikasi atau memverifikasi individu.

Telapak tangan memiliki luas permukaan yang lebih luas apabila dibandingkan dengan luas permukaan jari tangan. Selain luas permukaan yang lebih luas, telapak tangan juga memiliki hasil biometrik yang lebih stabil (Basya et al., 2012). Telapak tangan ditunjukkan pada Gambar 2.1

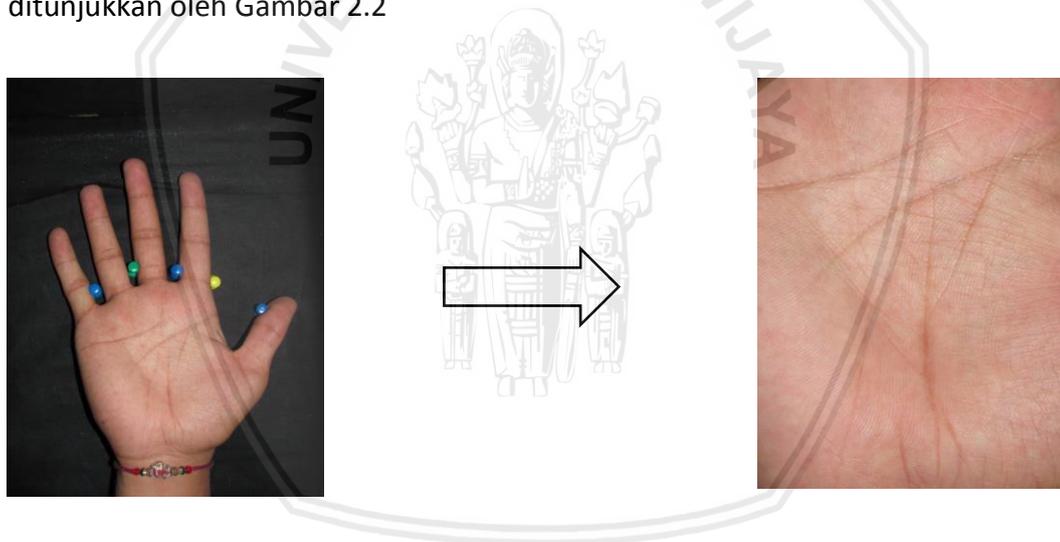




**Gambar 2.1 Palmprint**

Sumber : (Ali et al., 2016)

Telapak tangan yang nantinya akan digunakan sebagai data yaitu telapak tangan yang telah dilakukan proses *cropping* manual dengan ukuran 640x480 yang ditunjukkan oleh Gambar 2.2



**Gambar 2.2 Cropping Telapak Tangan**

Sumber : (Database College Engineering Of Pune)

### 2.3 Biometrik

Biometrik merupakan karakteristik fisik (biologi) serta tingkah laku yang unik dan terukur (Dharavath, Talukdar and Laskar, 2013). Dikatakan unik karena tidak terdapat pola fisik dari seseorang yang sama dari lahir hingga meninggal, meskipun pada orang yang kembar identik (Ahmed, 2015). *Biometric recognition* dianggap lebih aman dari pada metode autentifikasi tradisional karena tanda tangan seseorang dapat dengan mudah lupa dan ditiru (Ali et al., 2016).

Sistem biometrik dapat dimanfaatkan untuk mengatur akses ke ruang fisik, informasi dan lain lain. Manfaat biometrik yang telah terbukti efisien juga memiliki tujuan untuk mengurangi penipuan, penggunaan identitas palsu, pencurian informasi, serta meningkatkan kamanan publik dan kamanan nasional (N and Pato . Millett, 2010). *Biometric system* memanfaatkan fitur seperti suara, iris , sidik jari , telapak tangan, wajah, dan lain lain (Han, Guo and Zhang, 2008).

## 2.4 Ekstraksi Tekstur

Ekstraksi ciri merupakan tahapan untuk mengesktrak ciri atau informasi dari objek di dalam sebuah citra yang hendak dikenali atau diidentifikasi yang membedakan objek satu dengan lainnya. Fitur yang dapat digunakan untuk ekstraksi ciri yaitu berdasarkan bentuk, ukuran, geometri, tekstur, dan warna.

Ekstraksi ciri tekstur dapat menggunakan ciri statistik orde pertama atau orde dua. Orde pertama didasarkan pada karakteristik histogram citra dan pada umumnya digunakan untuk membedakan tekstur makrostruktur (perulangan periodik pada pola lokal). Orde dua digunakan untuk membedakan tekstur mikrostruktur (pola lokal dan perulangan yang bersifat random dan tidak jelas). Ciri yang telah didapat kemudian digunakan sebagai parameter atau nilai masukan untuk membedakan antara objek satu dengan yang lainnya menggunakan metode klasifikasi.

## 2.5 Grayscale

*Grayscale* atau citra keabuan hanya memiliki satu nilai yang merepresentasikan nilai *red, green, blue*. Derajat citra keabuan memiliki 8 bit nilai biner atau skala 0-255, dimana nilai 0 berwarna hitam dan 255 berwarna putih. Konversi citra dari warna RGB (*Red, Green, Blue*) menjadi citra keabuan ditunjukkan pada Persamaan 2.1.

$$gray = (0.299 \times R) + (0.587 \times G) + (0.114 \times B) \quad (2.1)$$

Keterangan:

- R* = nilai citra pada *layer red*
- G* = nilai citra pada *layer green*
- B* = nilai citra pada *layer blue*

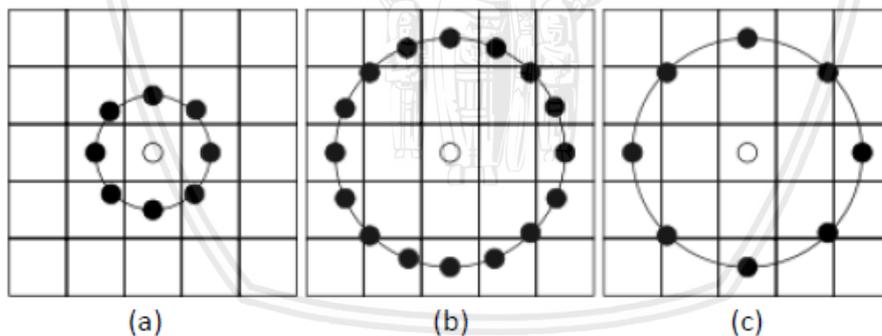
## 2.6 Local Binary Pattern (LBP)

Metode *Local Binary Pattern* pertama kali diperkenalkan oleh Ojala et al. Menurut Ojala et al. *Local Binary Pattern* merupakan metode yang digunakan sebagai ukuran tekstur *grayscale* yang terbukti efektif dan *invariant* terhadap pencahayaan yang berbeda. Metode ini teruji ampuh untuk mendeskripsikan tekstur, karena memiliki daya pembeda yang akurat, serta mempunyai toleransi terhadap perubahan *grayscale* yang *monotatic* (Turiyanto, Purwanto and

Dikairono, 2014). LBP dimanfaatkan untuk deskripsi tekstur dan didukung oleh komposisi pola mikro yang dapat dijelaskan oleh sebuah operator (Achsani, Atmaja and Purnamasari, 2015) .

Operator tersebut bekerja dengan cara mengubah gambar menjadi gambar berlabel bilangan bulat yang menggambarkan tampilan *small-scale*. Representasi dari label untuk menggambarkan statistik dari LBP yang sering digunakan yaitu histogram (Amynarto, Sari and Cahyawihandika, 2018). Label diberikan pada setiap piksel gambar dengan proses *thresholding window 3x3 neighbourhood* (ketetanggaan) sebagai nilai tengah dan mengubah hasilnya menjadi nilai 0 atau 1 (biner). Oleh karena itu Ojala, et al menyebut bahwa *fundamental pattern* ini sebagai *uniform pattern* karena mengandung dua *bitwise transition* dari 0 ke 1 dan sebaliknya. Apabila piksel bernilai kurang dari nilai tengah (piksel yang diolah) maka akan diberi nilai 0, sedangkan piksel yang memiliki nilai lebih dari nilai tengah maka akan diberi nilai 1.

Window *3x3 neighbourhood* (ketetanggaan) yang merupakan operator dasar LBP tidak dapat menangkap ciri domain yang memiliki struktur dengan skala besar. Oleh sebab itu operator dikembangkan untuk menggunakan ukuran tingkat ketetanggaan yang berbeda (Ojala, Pietikäinen and Maenpaa, 2002). Untuk menentukan ketetanggaan yang hendak digunakan jumlah tetangga atau jumlah cek poin disimbolkan dengan P dan jarak disimbolkan dengan R. P dan R ditunjukkan oleh Gambar 2.3



**Gambar 2.3 Jarak dan jumlah tetangga yang dipilih**

(a) R=1, P=8 (b) R=2, P=16 (c) R=2, P=8

Sumber : Pietikäinen, et al. (2011)

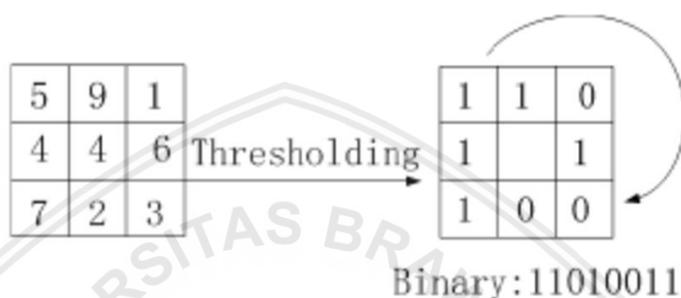
Secara matematis *thresholding* ditunjukkan oleh Persamaan 2.2, dan Persamaan 2.3 untuk menghitung nilai biner dari hasil *thresholding* menjadi angka desimal, sedangkan proses *thresholding* ditunjukkan pada Gambar 2.4.

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.2)$$

$$LBP_{p,r}(x_c, y) = \sum_{p=0}^{p-1} S(g_p - g_c)2^p \tag{2.3}$$

Keterangan :

- $P$  : banyaknya piksel tetangga
- $R$  : nilai jarak/radius
- $g_c$  : nilai dari piksel x dan y
- $g_p$  : nilai piksel tetangga
- $x_c, y_c$  : koordinat pusat



**Gambar 2.4 Thresholding**

Sumber : (Adi, Saputra, Tjokorda, 2015)

Untuk mengikut sertakan *micro-pattern* dari telapak tangan, ekstraksi LBP dilakukan dengan cara membagi citra sama rata menjadi *region-region*  $R_0, R_1, R_2, \dots, R_m$ . Pada setiap *region* dilakukan proses LBP yang kemudian menghasilkan histogram. Hasil histogram dari tiap *region* kemudian di-concat menjadi histogram ciri.

Histogram ciri yang terbentuk menggambarkan tekstur lokal dan bentuk global dari telapak tangan. Beberapa parameter yang dioptimalkan untuk menghasilkan ekstraksi ciri yang lebih baik. Parameter tersebut adalah operator dari LBP dan jumlah pembagian citra (*region*).

## 2.7 Histogram

Histogram merupakan diagram yang menggambarkan frekuensi setiap nilai yang muncul di seluruh piksel citra. Histogram sendiri berguna untuk mengamati penyebaran warna serta dapat dipakai sebagai pengambilan keputusan. Akan tetapi histogram tidak mencerminkan urutan posisi warna piksel dalam citra (Kadir, 2014).

## 2.8 Normalisasi

Normalisasi dilakukan dengan maksud agar tidak terjadi perbedaan nilai yang cukup jauh antara satu fitur dengan fitur histogram yang lain. Secara matematis normalisasi ditunjukkan oleh Persamaan 2.4.

$$X_{baru} = \left( \frac{x_{lama}}{\text{sum } x} \right) \tag{2.4}$$



Keterangan:

$X_{baru}$  : nilai normalisasi

$X_{lama}$  : nilai x lama yang belum dilakukan proses normalisasi

$Sum\ x$  : merupakan jumlah dari nilai x

## 2.9 K-NN (K- Nearest Neighbour)

K-Nearest Neighbor (KNN) merupakan algoritme klasifikasi yang cukup sederhana dan terkenal yang diperkenalkan oleh T.M Cover dan P.E Hart San. K-NN tergolong dalam *supervised learning*. *Supervised learning* adalah metode klasifikasi yang terlebih dahulu menyediakan label atau kategori pada data latih.

Algoritme K-NN dilakukan dengan cara mencari jarak yang paling dekat (mirip) pada data baru (data uji) dengan kelompok k objek dalam data latih (Leidiyana, 2013). Pada umumnya ada beberapa cara yang digunakan untuk mengukur objek data uji dengan objek data latih dan mengukur jarak yaitu *Euclidean*, *Manhattan*, *Minowski*, *Chebyshev*, dll. Tahapan berikutnya adalah dengan melakukan pengurutan nilai kemiripan (similaritas) berdasarkan nilai terbesar dari pemilihan k tetangga. Berikut adalah tahapan dari algoritme K-Nearest Neighbour (Priambodo, Dewi, & Triwiratno, 2015):

1. Menentukan berapa banyak k atau tetangga terdekat dari data latih terhadap data yang diuji.
2. Mencari jarak berdasarkan data latih dan data uji.
3. Mengurutkan data dari yang paling kecil hingga paling besar. Data yang dipakai yaitu telah dilakukan proses perhitungan jarak antara data latih dan data uji.
4. Mengelompokkan data sejumlah k data teratas.
5. Memilih label atau kategori yang paling banyak muncul.

## 2.10 Pengukuran Jarak Euclidean

*Euclidean* merupakan metode pengukuran jarak yang digunakan untuk mencari kedekatan antar dua vektor atau lebih. Kedekatan atau kemiripan dapat dicari dengan menggunakan nilai fitur yang dimiliki untuk mengisi vektor. Hasil yang paling kecil merepresentasikan jarak terdekat atau memiliki tingkat kemiripan yang tinggi. Pengukuran jarak *Euclidean* dilakukan dengan menerapkan Persamaan 2.5.

$$d(x, y) = \sqrt{\sum_{r=1}^n (x_i - y_i)^2} \quad (2.5)$$

Keterangan:

$(x, y)$  : jarak *Euclidean* antara  $x_i$  dengan  $y_i$ .

$x_i$  : data pada x ke- $i$  yang akan dilakukan proses perhitungan.

$y_i$  : data pada y ke- $l$  yang akan dilakukan proses perhitungan.

### 2.11 Perhitungan Akurasi

Perhitungan akurasi digunakan untuk mengetahui tingkat keberhasilan dari pengujian. Tingkat keberhasilan dapat dicari dengan membagi total data yang benar dengan jumlah seluruh data yang diuji dikalikan 100. Secara matematis perhitungan akurasi ditunjukkan pada Persamaan 2.6.

$$Akurasi = \frac{\sum Benar}{\sum Data Uji} \times 100\% \quad (2.6)$$



## BAB 3 METODOLOGI

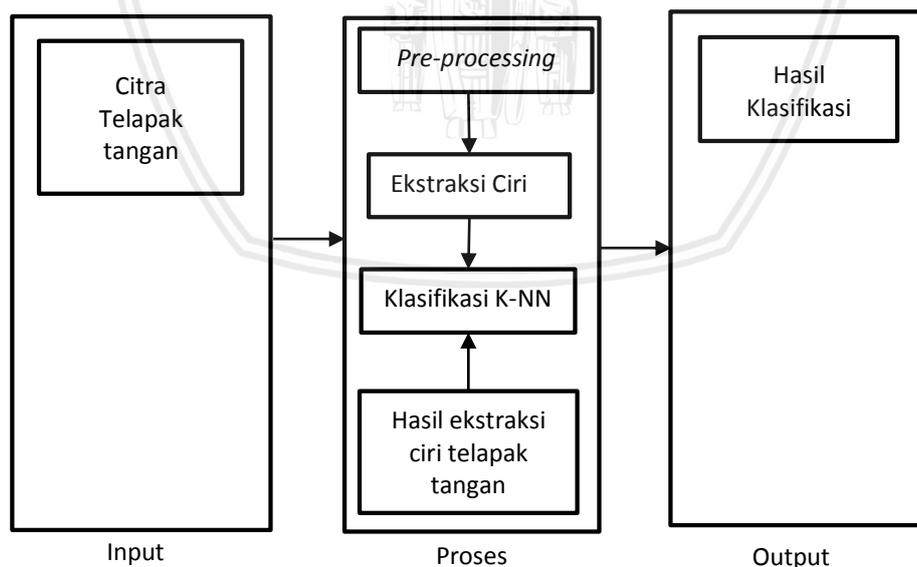
Pada bab ini akan membahas mengenai metodologi yang digunakan dalam penelitian ekstraksi ciri pada telapak tangan dengan metode *Local Binary Pattern* (LBP). Adapun subbab yang akan dibahas antara lain adalah tipe penelitian, strategi penelitian, analisis kebutuhan, teknik pengumpulan data, lokasi penelitian, perancangan pengujian dan analisis, penarikan kesimpulan dan saran.

### 3.1 Tipe Penelitian

Tipe penelitian yang digunakan dalam penelitian “Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP)” adalah bertipe non-implimentatif analitik. Tipe ini dipilih disebabkan oleh adanya hypotesis yang realistis, dapat diuji, spesifik dan terdiri dari penentuan variabel yang dipakai berlandaskan studi yang telah dilakukan pada literature sebelumnya. Penelitian ini merupakan penelitian analitik karena mengangkat analisis sebagai hasil utama dari sistem ini, yaitu analisis apakah pemakaian algoritme *Analytical Hierrarchy Process* dan algoritme *Technique for Order of Preferency by Similiarity to Ideal Solution* sudah cukup efektif dalam memberikan rekomendasi.

### 3.2 Strategi Penelitian

Pada subbab ini strategi penelitian yang dilakukan yaitu dengan cara mengumpulkan citra data telapak tangan dan melakukan proses dari algoritme yang digunakan yang ditunjukkan pada Gambar 3.1.



**Gambar 3.1 Alur Proses dalam Model**

Pada Gambar 3.1 hal yang pertama dilakukan adalah memasukkan citra telapak tangan yang telah dilakukan proses *cropping* manual, kemudian melakukan proses *pre-processing* dengan cara mengubah citra berwarna menjadi

citra keabuan, selanjutnya melakukan proses ekstraksi ciri menggunakan metode *Local Binary Pattern*, setelah didapatkan ciri dari telapak tangan maka dilakukan proses klasifikasi yang memanfaatkan metode *K-Nearest Neighbour* (K-NN), dan hasil *output* berupa identifikasi telapak tangan.

### 3.3 Peralatan Pendukung

Peralatan pendukung yang digunakan pada penelitian ini antara lain perangkat keras pendukung, perangkat lunak pendukung.

#### 3.3.1 Perangkat Keras Pendukung

Perangkat keras yang dimanfaatkan dalam implementasi sistem adalah sebagai berikut:

1. Processor : Intel® Core™ i7-5550U CPU @ 2.4GHz (4CPUs), ~2.4GHz
2. Memory : 4096MB
3. VGA : NVIDIA GeForce 940MX
4. ROM : 1000 GB HDD
5. Monitor
6. Keyboard
7. Mouse

#### 3.3.2 Perangkat Lunak Pendukung

Perangkat lunak yang digunakan dalam implementasi algoritme penelitian adalah sebagai berikut:

1. *Operating system* Microsoft Windows 8.1 Pro.
2. Microsoft Office Word 2013 sebagai media pembuatan laporan.
3. Microsoft Office PowerPoint 2013 sebagai media pembuatan presentasi hasil.
4. Microsoft Office Excel 2013 sebagai media pembuatan data pengujian dan manualisasi.
5. Bahasa pemrograman yang digunakan yaitu *python*.
6. JetBrains PyCharm Community Edition 2107.3 sebagai media pembuatan kode program.
7. Adobe Photoshop CC 2017 sebagai media *cropping* manual citra telapak tangan.

### 3.4 Pengumpulan Data

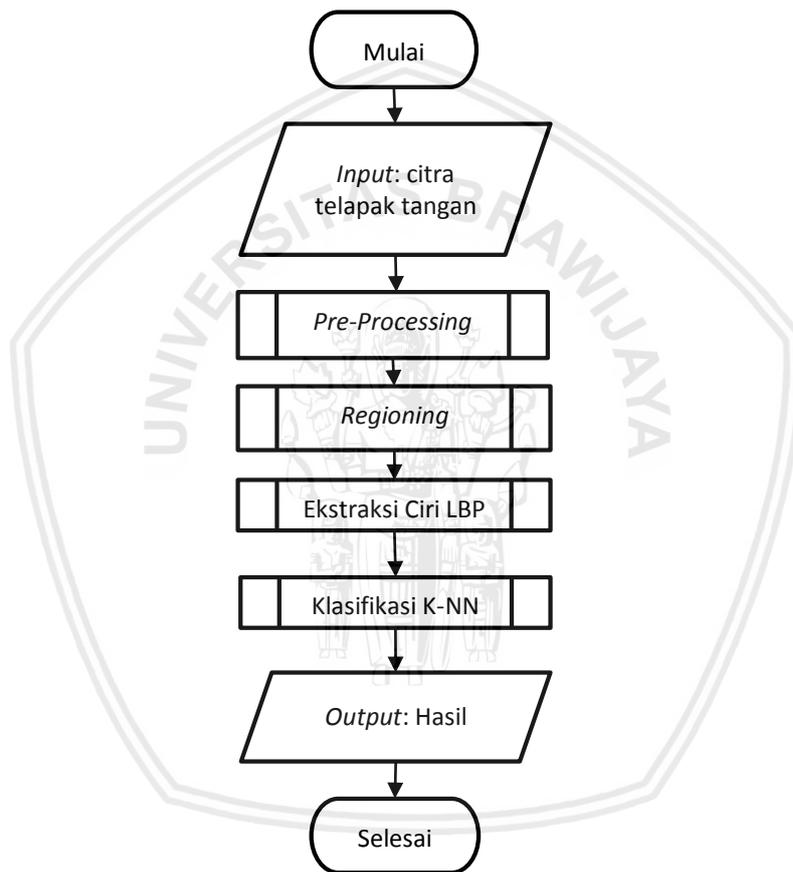
Data yang digunakan dalam penelitian ini merupakan data sekunder. Data diambil dari database College of Engineering, Pune (<http://www.coep.org.in/resources/coeppalmprintdatabase>).

### 3.5 Lokasi Penelitian

Penelitian “Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP)” dilakukan di Laboratorium Riset Komputasi Cerdas dan Visualisasi Fakultas Ilmu Komputer Universitas Brawijaya, Kota Malang, Provinsi Jawa Timur.

### 3.6 Perancangan Algoritme

Pada subbab perancangan algoritme akan dijelaskan mengenai tahap perancangan yang akan dijadikan dasar dari implementasi algoritme. Alur algoritme penelitian ditunjukkan pada Gambar 3.2



**Gambar 3.2 Alur Algoritme**

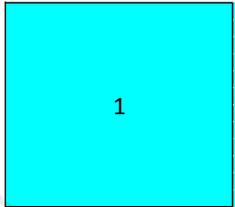
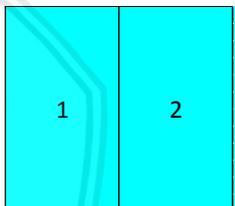
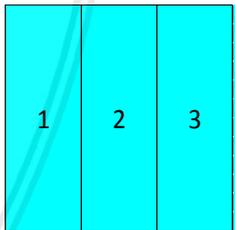
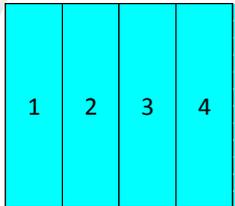
Pada Gambar 3.2 dimulai dari memasukkan input berupa citra telapak tangan yang telah dilakukan proses *cropping* manual, kemudian dilakukan proses *pre-processing* yaitu dengan cara mengubah citra berwarna menjadi citra keabuan, setelah mendapat citra keabuan kemudian dilakukan proses *regioning* yaitu dengan membagi citra menjadi beberapa *region*, setiap *region* diterapkan metode *Local Binary Pattern* sebagai metode ekstraksi ciri, ketika proses lbp selesai maka akan didapat ciri yang nantinya akan dilakukan proses klasifikasi menggunakan K-NN (*K-Nearest Neighbour*).

### 3.7 Pengujian dan Analisis

Pengujian dilakukan bertujuan untuk mengetahui apakah algoritme dari Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP) yaitu:

1. Menguji jumlah *region* dan nilai pembagian dari *height* dan *width*, dimana *region* yang diuji merupakan hasil perkalian dari *height* dan *width*. Jumlah *region* dan pembagian *height* dan *width* citra yang ditunjukkan oleh Tabel 3.1

**Tabel 3.1 Jumlah Region dan Pembagian Height dan Width**

No	Jumlah Region	Height	Width	Representasi
1	1	1	1	
2	2	1	2	
3	3	1	3	
4	4	1	4	

**Tabel 3.1 Jumlah Region dan Pembagian Height dan Width(lanjutan)**

No	Jumlah Region	Height	Width	Representasi
----	---------------	--------	-------	--------------

5	2	2	1	<table border="1"> <tr><td>1</td></tr> <tr><td>2</td></tr> </table>	1	2							
1													
2													
6	4	2	2	<table border="1"> <tr><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td></tr> </table>	1	2	3	4					
1	2												
3	4												
7	6	2	3	<table border="1"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> </table>	1	2	3	4	5	6			
1	2	3											
4	5	6											
8	8	2	4	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table>	1	2	3	4	5	6	7	8	
1	2	3	4										
5	6	7	8										
9	3	3	1	<table border="1"> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> </table>	1	2	3						
1													
2													
3													
10	6	3	2	<table border="1"> <tr><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td></tr> </table>	1	2	3	4	5	6			
1	2												
3	4												
5	6												
11	9	3	3	<table border="1"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> </table>	1	2	3	4	5	6	7	8	9
1	2	3											
4	5	6											
7	8	9											

**Tabel 3.1 Jumlah Region dan Pembagian Height dan Width(lanjutan)**

No	Jumlah Region	Height	Width	Representasi
----	---------------	--------	-------	--------------

12	12	3	4	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>9</td><td>10</td><td>11</td><td>12</td></tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12				
1	2	3	4																	
5	6	7	8																	
9	10	11	12																	
13	4	4	1	<table border="1"> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> </table>	1	2	3	4												
1																				
2																				
3																				
4																				
14	8	4	2	<table border="1"> <tr><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td></tr> </table>	1	2	3	4	5	6	7	8								
1	2																			
3	4																			
5	6																			
7	8																			
15	12	4	3	<table border="1"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> <tr><td>10</td><td>11</td><td>12</td></tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12				
1	2	3																		
4	5	6																		
7	8	9																		
10	11	12																		
16	16	4	4	<table border="1"> <tr><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td>9</td><td>10</td><td>11</td><td>12</td></tr> <tr><td>13</td><td>14</td><td>15</td><td>16</td></tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	3	4																	
5	6	7	8																	
9	10	11	12																	
13	14	15	16																	

2. Menguji parameter *Local Binary Pattern* yaitu nilai jarak tetangga 1,2 dan 3.

Untuk mendapatkan hasil dari skenario pengujian tersebut maka digunakan perhitungan akurasi yang sesuai dengan Persamaan 2.6.

### 3.8 Penarikan Kesimpulan dan Saran

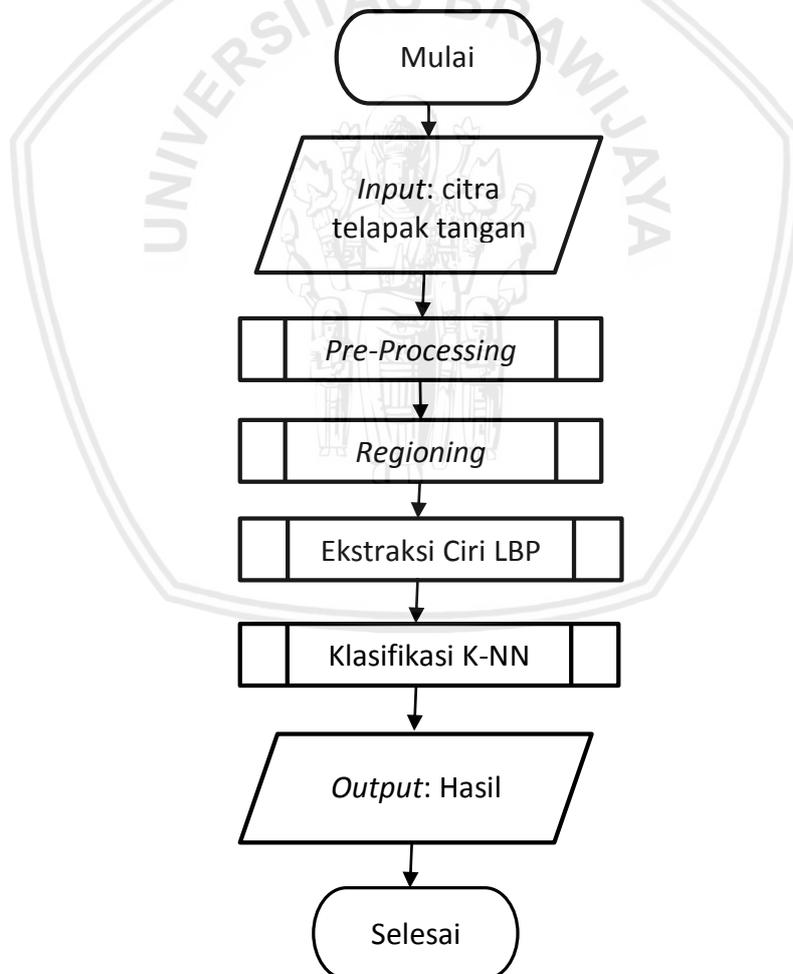
Penarikan kesimpulan dan saran dilakukan setelah melakukan pengujian dan analisis terhadap data. Ini bertujuan untuk mendapat hasil yang didapat setelah melakukan proses penelitian serta memberikan saran untuk penelitian yang mendatang agar lebih baik lagi hasil yang didapatkan pada Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP).

## BAB 4 PERANCANGAN

Pada bab ini akan menjelaskan mengenai perancangan algoritme, perancangan dari skenario pengujian dan perhitungan manual yang digunakan dalam Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP).

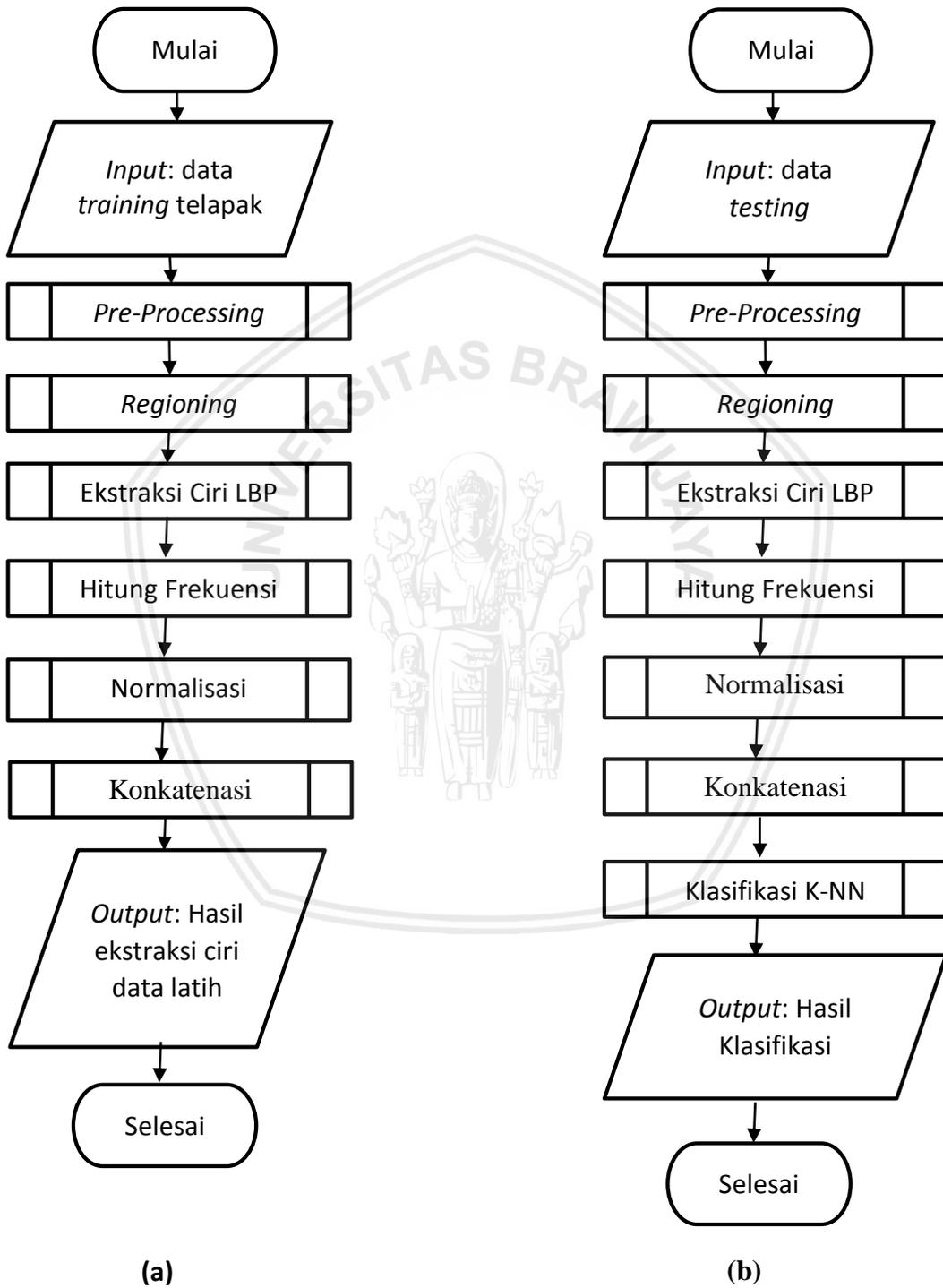
### 4.1 Perancangan Algoritme

Tahap perancangan algoritme bertujuan untuk merancang langkah-langkah algoritme yang akan diimplementasikan. Perancangan algoritme ini dibangun berdasarkan data set dari obyek permasalahan yang telah dilakukan pada tahap sebelumnya. Masukkan (*input*) diolah sedemikian rupa sehingga menghasilkan keluaran (*output*) yang sesuai dengan tujuan awal. Tahap ini juga bertujuan untuk mempermudah implementasi, pengujian dan analisis. Perancangan algoritme secara general ditunjukkan pada Gambar 4.1



Gambar 4.1 Diagram Alir Algoritme

Sedangkan gambaran algoritme “Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP)” secara detail yaitu terdiri dari algoritme *training* dan *testing*. Diagram alir *training* dan diagram alir *testing* ditunjukkan oleh Gambar 4.2

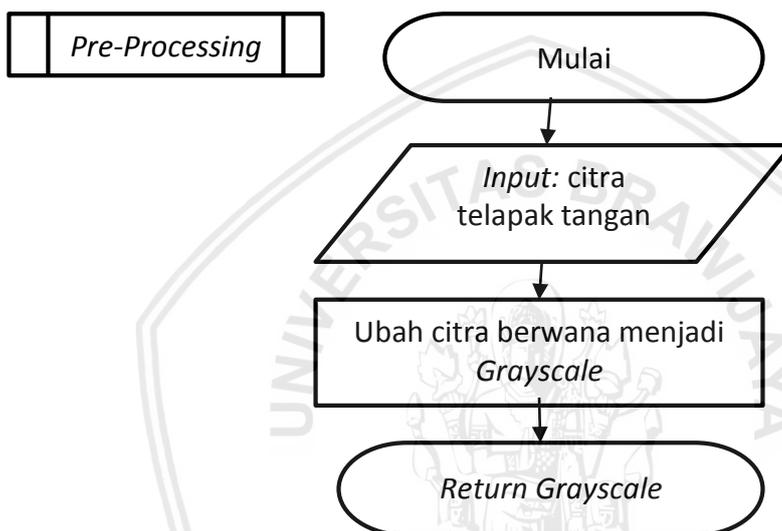


Gambar 4.2 Diagram Alir (a) *Training* (b) *Testing*

Langkah yang terdapat pada diagram alir *training* hampir mirip dengan *testing* yakni terdiri dari lima proses utama yaitu *Pre-processing*, *Regioning*, ekstraksi ciri LBP, hitung *Histogram* dan normalisasi *Histogram* akan tetapi terdapat sedikit perbedaan dari keduanya yakni terdapat satu proses yang hanya diterapkan pada algoritme *testing* proses tersebut adalah algoritme klasifikasi *K-Nearest Neighbour*.

#### 4.1.1 Pre-Processing

*Pre-Processing* berguna untuk menghilangkan *noise* atau membuat citra masukkan memiliki kualitas yang lebih berkualitas dari citra sebelumnya. Proses tersebut ditunjukkan oleh Gambar 4.3.

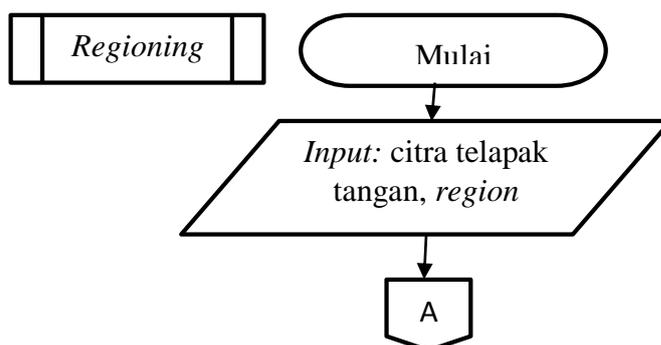


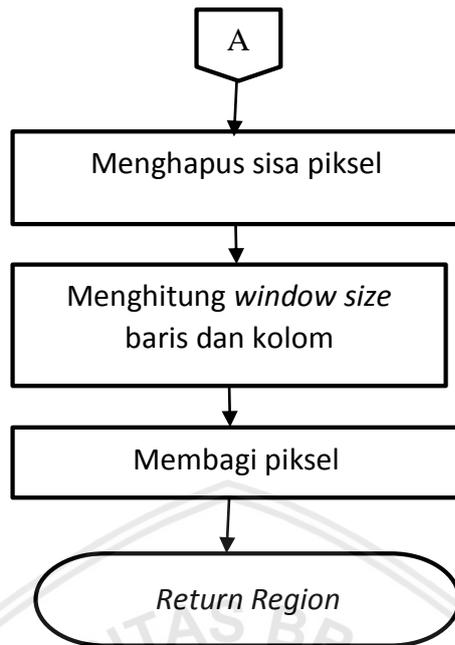
Gambar 4.3 Pre-Processing

Pertama citra masukkan dilakukan proses mengubah citra berwarna mejadi keabuabng (*grayscale*) kemudian nilai dari *grayscale* dari citra tersebut dikembalikan agar dapat diproses berikutnya yang direpresentasikan oleh Gambar 4.4.

#### 4.1.2 Regioning

*Regioning* dilakukan untuk memperoleh sub *region*. Tiap sub *region* akan dilakukan proses ekstraksi ciri. Langkah untuk mendapatkan sub *region* ditunjukkkkan oleh Gambar 4.4.



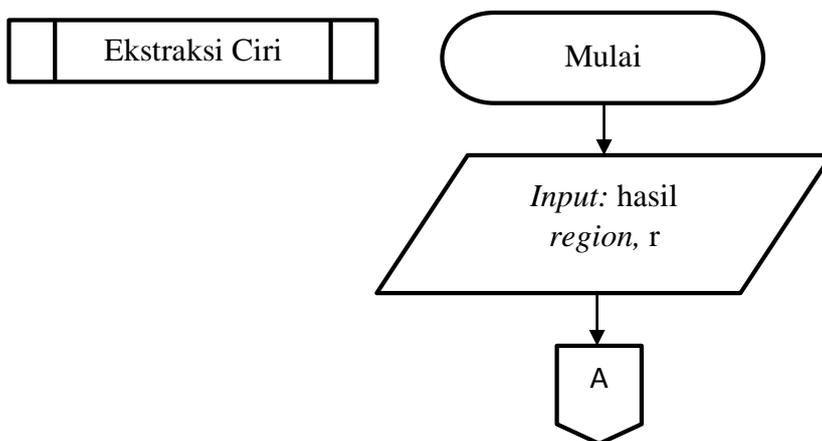


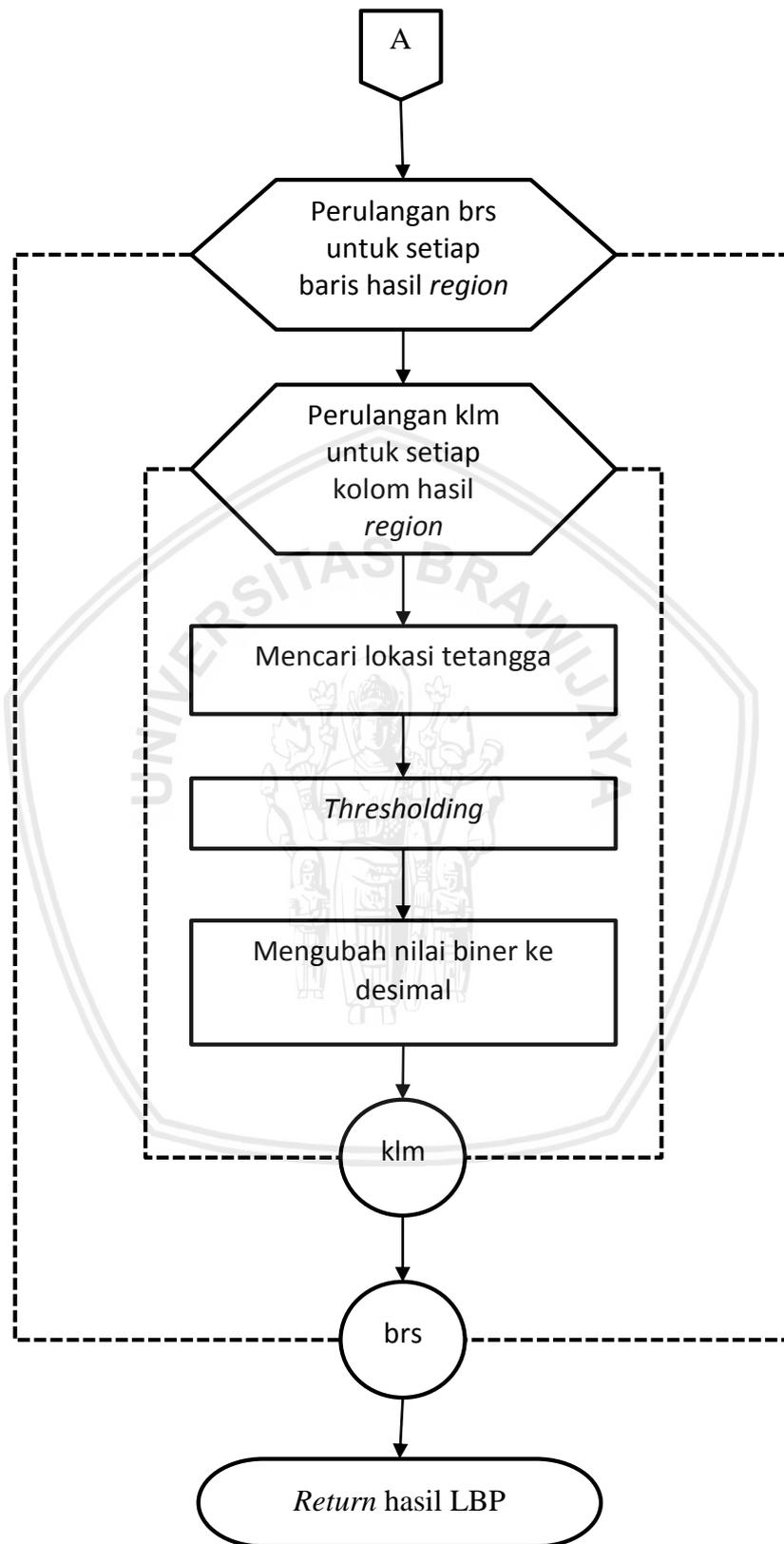
**Gambar 4.4 Regioning**

Dari Gambar 4.4 dapat dideskripsikan bahwa masukkan berupa citra telapak tangan yang telah dilakukan proses *grayscale* dan memasukkan *region* untuk mendapatkan nilai pembagi pada *height* dan *width* citra. Dilanjutkan dengan proses menghapus sisa piksel yang tidak terpakai. Setelah piksel yang terpakai dihapus maka akan dilakukan proses menghitung *window size* pada baris dan kolom yang akan dibagi. Ketika *window size* pada baris dan kolom sudah didapatkan maka masuk ke langkah berikutnya yaitu membagi piksel apabila proses sudah selesai maka nilai *region* akan dikembalikan.

### 4.1.3 Ekstraksi Ciri LBP

Ekstraksi dilakukan untuk mendapatkan ciri dari telapak tangan. Dimana untuk mendapatkan ciri tersebut menggunakan metode *Local Binary Pattern*, tahapan dari LBP ditunjukkan oleh Gambar 4.5.



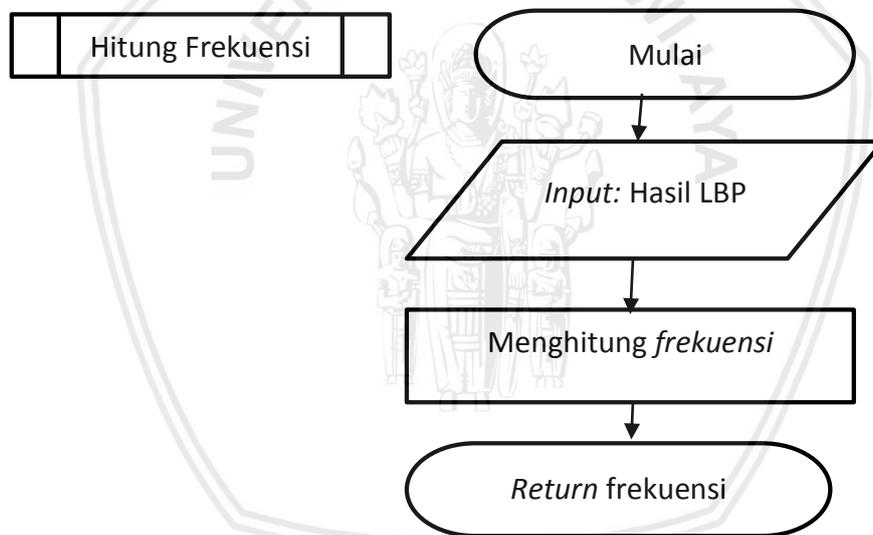


Gambar 4.5 Ekstraksi Ciri LBP

Dari Gambar 4.5 tersebut dapat dijelaskan bahwa masukkan berupa hasil dari proses *regioning* dan nilai  $r$  atau jarak ketetangga, yang kemudian akan dilakukan mencari lokasi dari tetangga, dilanjutkan dengan proses membandingkan antara nilai pusat dengan nilai tetangga atau proses *thresholding*. Pada proses ini mengubah nilai menjadi biner jika nilai tetangga lebih besar dari nilai pusat maka bernilai 1 dan apabila nilai tetangga lebih kecil dari nilai pusat maka akan bernilai 0. Setelah mendapat nilai biner tersebut maka akan dilakukan proses mengubah nilai biner ke desimal. Proses mencari lokasi tetangga sampai proses mengubah biner ke desimal diulangi terhadap seluruh piksel dan dikembalikan sebagai hasil LBP.

#### 4.1.4 Hitung Frekuensi

Fitur yang digunakan yaitu histogram. Histogram yang digunakan yaitu 8-bit 256 bin atau skala 0-255. Untuk mendapatkan fitur tersebut maka dilakukan proses perhitungan frekuensi dari citra. Perhitungan frekuensi untuk mendapatkan fitur yang ingin digunakan yaitu ditunjukkan oleh Gambar 4.6.

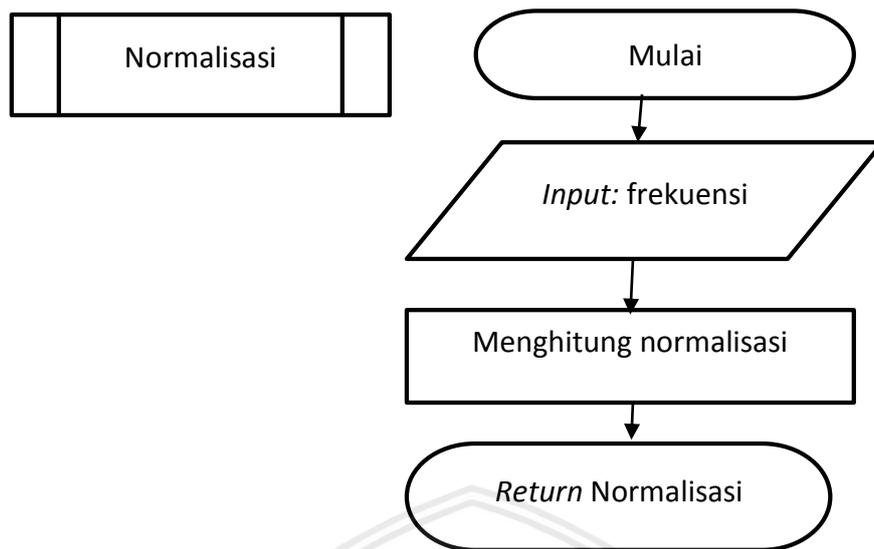


Gambar 4.6 Hitung Histogram

Dalam diagram alir tersebut dapat didefinisikan bahwa masukkan berupa hasil LBP. Kemudian dilakukan proses perhitungan frekuensi pada hasil LBP untuk mendapatkan nilai *histogram* dan dikembalikan sebagai frekuensi.

#### 4.1.5 Normalisasi

Normalisasi bertujuan agar tidak terjadi perbedaan nilai yang besar pada histogram. Normalisasi histogram ditunjukkan pada Gambar 4.7.

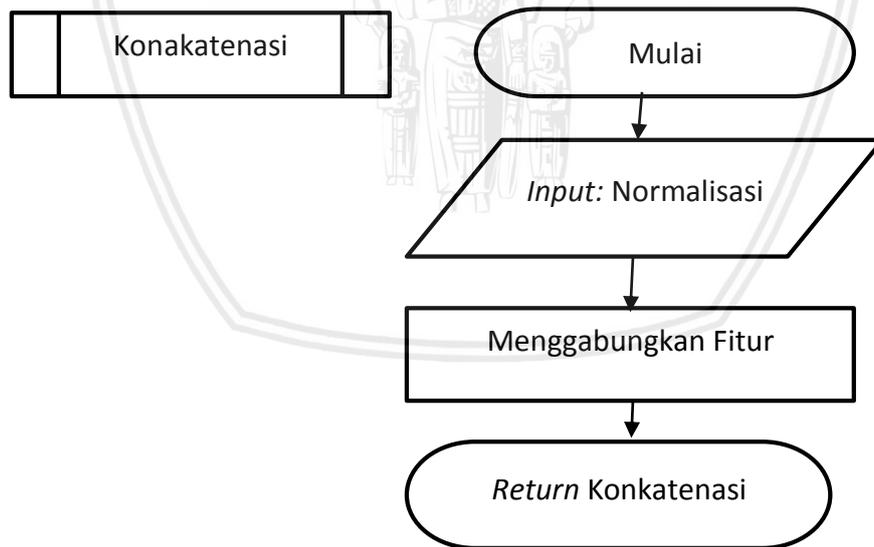


**Gambar 4.7 Normalisasi**

Pada Gambar 4.7 dapat dideskripsikan bahwa *input* atau masukan berupa frekuensi atau histogram dari citra yang kemudian dilakukan proses normalisasi menggunakan Persamaan 2.4 kemudian dikembalikan dengan nilai normalisasi.

#### 4.1.6 Konkatenasi

Proses konaktenasi ditunjukkan pada Gambar 4.8

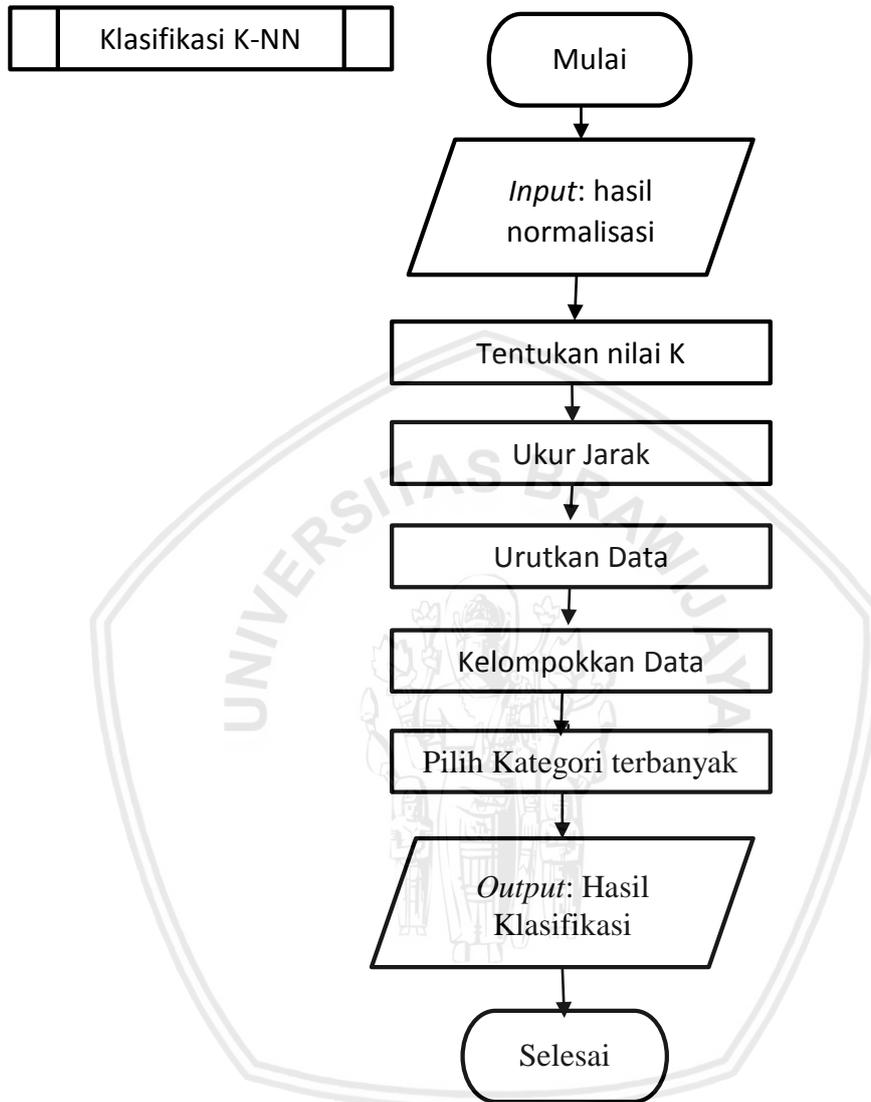


**Gambar 4.8 Konkatenasi**

Pada Gambar 4.8 menunjukkan bahwa input berupa hasil ekstraksi ciri yang telah dilakukan proses normalisasi pada setiap *region* kemudian dilakukan proses konkatenasi dengan menggabungkan seluruh fitur pada setiap *region*.

#### 4.1.7 Klasifikasi K-NN

Klasifikasi K-NN memiliki lima proses utama. Tahapan klasifikasi *K-Nearest Neighbour* ditunjukkan oleh Gambar 4.9.



**Gambar 4.9** Klasifikasi K-NN

Masukkan berupa hasil normalisasi dari data *training* dan data *testing*. Nilai K ditentukan untuk mengambil k-hasil tertinggi. Pengukuran jarak dilakukan antara data *training* dan data *testing* untuk mendapatkan jarak terdekat. Setelah dilakukan pengukuran jarak maka hasil tersebut diurutkan dari jarak yang paling kecil hingga terbesar. Setelah itu kelompokkan data tersebut dengan jumlah k-data teratas. Nilai dari kategori terbanyak merupakan kelas dari data uji tersebut. Dan keluaran hasil klasifikasi berupa kategori.

## 4.2 Perhitungan Manual

Perhitungan manual dilakukan dengan tujuan untuk menjelaskan proses secara matematis dari algoritme yang digunakan. Perhitungan manual juga berfungsi sebagai acuan benar tidaknya algoritme.

Pada sub bab ini akan membahas proses perhitungan manual *pre-processing*, *local binary pattern*, klasifikasi K-NN. Proses perhitungan manualisasi penelitian ini dilakukan dengan cara membuat citra buatan berukuran 5 x 5 pixel yang merepresentasikan citra telapak tangan sebelum di lakukan proses *pre-processing*. Citra terdiri dari tiga warna yaitu RGB (*Red Green Blue*) yang masing-masing akan ditunjukkan oleh tabel 1.1, tabel 1.2, dan tabel 1.3.

**Tabel 4.1 Nilai dari Layer Red**

X, Y	0	1	2	3	4	5	6	7	8	9
0	220	210	204	211	208	204	207	205	210	201
1	212	217	217	200	191	192	193	194	195	201
2	209	211	211	201	220	210	204	211	300	203
3	215	208	208	202	212	217	217	200	191	204
4	208	211	211	203	209	211	211	201	215	198
5	206	210	210	204	215	208	208	202	200	196
6	193	199	199	205	208	211	211	203	198	195
7	201	193	211	206	206	210	210	204	199	194
8	201	211	195	198	193	199	199	205	189	201
9	201	203	203	196	201	193	211	206	200	210

**Tabel 4.2 Nilai dari Layer Green**

X, Y	0	1	2	3	4	5	6	7	8	9
0	173	162	152	155	151	148	153	149	153	143
1	163	160	136	137	138	139	140	141	152	164
2	162	161	165	141	151	144	155	147	166	165
3	168	162	166	142	155	141	144	145	165	155
4	161	163	168	143	162	161	165	141	168	144
5	154	164	173	144	168	162	166	142	169	141
6	136	165	172	145	161	163	168	143	170	151
7	140	155	171	146	154	164	173	144	171	138

**Tabel 4.2 Nilai dari Layer Red (lanjutan)**

X, Y	0	1	2	3	4	5	6	7	8	9
8	151	144	155	147	136	165	172	145	145	151
9	155	141	144	145	146	147	148	149	150	155

**Tabel 4.3 Nilai dari Layer Blue**

X, Y	0	1	2	3	4	5	6	7	8	9
0	155	148	138	38	134	125	129	136	123	147
1	146	151	147	148	149	150	141	139	135	145
2	146	143	120	120	155	140	133	134	129	132
3	150	140	121	121	146	139	117	122	128	131
4	141	139	122	122	146	134	121	136	149	136
5	133	134	123	123	150	122	128	123	151	119
6	117	122	124	124	141	136	132	130	121	117
7	121	136	148	125	133	127	146	151	147	115
8	128	123	151	133	122	143	146	143	120	125
9	132	130	119	132	141	145	150	140	121	126

#### 4.2.1 Grayscale

Dari ketiga tabel tersebut maka akan dilakukan proses perhitungan untuk mencari nilai keabuan atau *grayscale*. Apabila nilai *Red* = 35, *Green* = 67, dan *Blue* = 205 maka untuk mendapatkan nilai keabuan perhitungan digunakan dalam proses ini yaitu menggunakan Persamaan 2.1

$$\begin{aligned}
 \text{nilai}_{\text{grayscale}} &= (0.299 \times \text{Red}) + (0.587 \times \text{Green}) + (0.114 \times \text{Blue}) \\
 &= (0.299 \times 204) + (0.587 \times 152) + (0.114 \times 132) \\
 &= 175
 \end{aligned}$$

Perhitungan tersebut dilakukan pada tiap piksel. Hasil dari perhitungan seluruh piksel *grayscale* direpresentasikan pada Tabel 4.4.

**Tabel 4.4 Nilai dari Grayscale**

X, Y	0	1	2	3	4	5	6	7	8	9
0	185	174	165	158	166	162	166	164	166	160
1	175	176	161	157	155	156	155	156	162	172
2	174	173	173	156	172	163	167	164	201	172
3	180	173	173	157	171	163	162	158	168	166
4	172	174	175	158	174	172	173	158	179	159
5	167	174	178	159	180	171	174	157	176	154
6	150	170	174	160	172	174	176	159	172	160
7	156	164	180	161	167	173	180	162	176	152
8	163	161	166	160	151	172	177	162	155	162
9	166	158	158	158	161	160	167	165	161	168

**4.2.2 Regioning**

Proses regioning dilakukan dengan cara membagi matrix dari citra keabuan Tabel 4.4 menjadi sejumlah *n-height* dan *m-width*. Pembagian *region* ditunjukkan pada Tabel 4.5 dan Tabel 4.6.

**Tabel 4.5 height = 2, width = 2**

X, Y	0	1	2	3	4	5	6	7	8	9
0	185	174	165	158	166	162	166	164	166	160
1	175	176	161	157	155	156	155	156	162	172
2	174	173	173	156	172	163	167	164	201	172
3	180	173	173	157	171	163	162	158	168	166
4	172	174	175	158	174	172	173	158	179	159
5	167	174	178	159	180	171	174	157	176	154
6	150	170	174	160	172	174	176	159	172	160
7	156	164	180	161	167	173	180	162	176	152
8	163	161	166	160	151	172	177	162	155	162
9	166	158	158	158	161	160	167	165	161	168

Pada Tabel 4.5 terdapat 4 *region* dimana sub *region* memiliki ukuran yang sama, dan tidak ada piksel yang dibuang.



**Tabel 4.6 height = 3, width = 2**

X, Y	0	1	2	3	4	5	6	7	8	9
0	185	174	165	158	166	162	166	164	166	160
1	175	176	161	157	155	156	155	156	162	172
2	174	173	173	156	172	163	167	164	201	172
3	180	173	173	157	171	163	162	158	168	166
4	172	174	175	158	174	172	173	158	179	159
5	167	174	178	159	180	171	174	157	176	154
6	150	170	174	160	172	174	176	159	172	160
7	156	164	180	161	167	173	180	162	176	152
8	163	161	166	160	151	172	177	162	155	162
9	166	158	158	158	161	160	167	165	161	168

Pada Tabel 4.6 terdapat 6 *region* pada setiap sub *region* memiliki ukuran yang sama, sedangkan pada piksel yang berwarna merah maka akan dihilangkan.

### 4.2.3 Local Binary Pattern

Proses perhitungan *Local Binary Pattern* dilakukan tiap sub *region*, tahapan LBP dengan nilai  $r = 1$ ,  $p = 8$  dan sub *region* ditunjukkan oleh Gambar 4.10.

185	174	165	158	166
175	176	161	157	155
174	173	173	156	172
180	173	173	157	171
172	174	175	158	174

**Gambar 4.10 Sub Region**

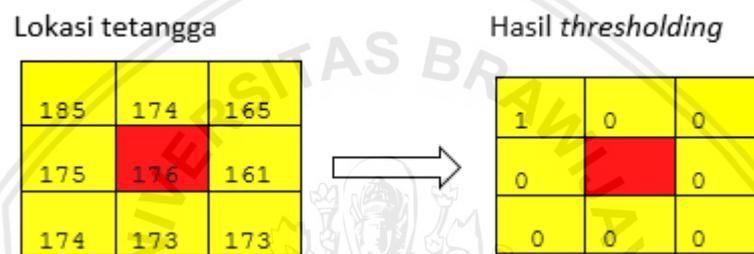
1. Melakukan proses pencarian lokasi tetangga dimulai dari nilai pusat dengan koordinat(1,1) ditunjukkan pada Gambar 4.11.



185	174	165	158	166
175	176	161	157	155
174	173	173	156	172
180	173	173	157	171
172	174	175	158	174

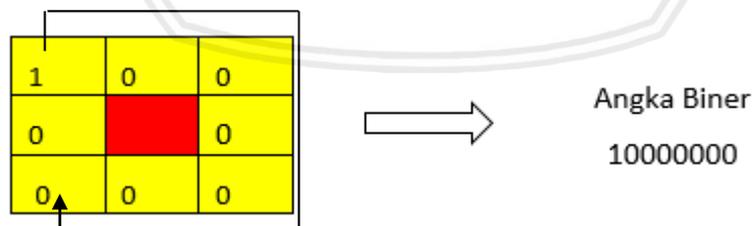
**Gambar 4.11 Lokasi Tetangga**

- Melakukan proses *thresholding* terhadap nilai tetangga dan mengubah nilai menjadi 1 apabila nilai tetangga lebih besar dari nilai pusat, dan sebaliknya bernilai 0 jika nilai tetangga lebih kecil dari nilai pusat yang ditunjukkan pada Gambar 4.12.



**Gambar 4.12 Thresholding**

- Setelah mendapatkan hasil dari *thresholding* maka dilakukan proses pengubah biner ke desimal secara *clockwise* atau searah jarum. Setelah mendapatkan hasil biner ke desimal maka nilai desimal tersebut dimasukkan ke dalam koordinat (1,1) yang ditunjukkan oleh Gambar 4.13.



**Gambar 4.13 Penyusunan biner**

$$\begin{aligned} \text{Nilai Desimal} &= (2^7 \times 1) + (2^6 \times 0) + (2^5 \times 0) + (2^4 \times 0) + (2^3 \times 0) + (2^2 \times 0) + (2^1 \times 0) + (2^0 \times 0) \\ &= 128 \end{aligned}$$

Nilai desimal dari hasil proses *thresholding* pada koordinat (1,1) adalah 128. Kemudian pusat akan bergeser ke koordinat (1,2) dst. Dan tahapan 1-3 tersebut dilakukan sampai semua piksel sudah dilakukan proses *thresholding*. Hasil dari proses *Local Binary Pattern* ditunjukkan pada Gambar 4.14.



0	0	0	0	0
0	128	199	235	0
0	195	128	223	0
0	141	6	191	0
0	0	0	0	0

Gambar 4.14 Hasil LBP

#### 4.2.4 Hitung Frekuensi

Ketika hasil dari *Local Binary Pattern* sudah didapat maka langkah berikutnya adalah menghitung frekuensi dari histogram. Hasil dari frekuensi histogram dari Gambar 4.13 ditunjukkan dengan Tabel 4.9

Tabel 4.7 Frekuensi Histogram

Fitur	Frekuensi
0	16
6	1
128	2
141	1
191	1
195	1
199	1
223	1
235	1

Fitur histogram 0-255 yang tidak ada diatas akan diisi dengan frekuensi 0.

#### 4.2.5 Normalisasi

Ketika nilai frekuensi sudah didapatkan maka akan dilakukan normalisasi dengan menggunakan Persamaan 2.4.

$$X_{baru} = \left(\frac{16}{25}\right) = 0.64$$

**Tabel 4.8 Normalisasi**

Fitur	Frekuensi
0	0.64
6	0.04
128	0.08
141	0.04
191	0.04
195	0.04
199	0.04
223	0.04
235	0.04

Proses normalisasi dilakukan pada semua fitur histogram yaitu dari 0-255 untuk setiap *region* nya. Yang kemudian dilakukan konkatenasi dari sub *region* satu dengan yang lainnya.

#### 4.2.6 Konkatenasi

Proses konkatenasi menggabungkan hasil normalisasi pada setiap *region* contohnya terdapat 4 *region* yaitu sub *region* 1,2,3,4 maka fitur yang digabungkan juga sesuai urutan yaitu fitur pada *region* 1, kemudian digabung fitur pada *region* 2, kemudian digabung dengan fitur pada *region* 3, dan terakhir digabung dengan fitur pada *region* 4, jadi total fitur setelah dilakukan proses konkatenasi yaitu sebanyak  $256 \times 4 \text{ region} = 1024$  fitur baru yang terbentuk pada sebuah citra. Yang ditunjukkan pada Gambar 4.15.

Fitur sub region 1	Fitur sub region 2	Fitur sub region 3	Fitur sub region 4
--------------------	--------------------	--------------------	--------------------

**Gambar 4.15 Hasil Konkatenasi**

#### 4.2.7 K-NN

Setelah mendapatkan nilai normalisasi dari histogram maka akan dilakukan proses klasifikasi yaitu menggunakan  $K=NN$ . Pada proses manualisasi K-NN jarak yang digunakan yaitu *Euclidean*. Langkah untuk pengklasifikasian K-NN adalah sebagai berikut:

1. Menentukan nilai K yaitu 5

2. Mengukur jarak data uji dengan data latih dengan menggunakan Euclidean, setiap fitur yang akan diukur yaitu menggunakan rumus Persamaan 2.5 . Maka perhitungannya sebagai berikut:

$$d(x,y) = \sqrt{(0.64 - 0.2)^2 + (0.0 - 0.3)^2 + \dots + (0.11 - 0.1)^2}$$

$$= 0.041$$

Hasil perhitungan jarak ditunjukkan pada Tabel 4.5

**Tabel 4.9 Hasil Perhitungan Jarak**

Data Latih	d(x,y)	Kelas
1	0.0525	1
2	0.06	1
3	0.041	1
4	0.2	2
5	0.3	2
6	0.21	2

3. Setelah melakukan pengukuran jarak makalangkah berikutnya yaitu mengurutkan hasil dari pengukuran jarak dari terkecil hingga terbesar. Hasil pengurutan ditunjukkan oleh Tabel 4.6.

**Tabel 4.10 Hasil Pengurutan Jarak**

Data Latih	d(x,y)	Kelas
3	0.041	1
1	0.0525	1
2	0.06	1
4	0.2	2
6	0.21	2
5	0.3	2

4. Setelah diurutkan yaitu mengambil nilai 5 teratas dan mengelompokkan sesuai kategori atau kelas. Nilai 5 teratas ditunjukkan oleh Tabel 4.7

**Tabel 4.11 Hasil Pengambilan 5 Teratas**

Data Latih	$d(x,y)$	Kelas
3	0.041	1
1	0.0525	1
2	0.06	1
4	0.2	2
6	0.21	2

5. Langkah terakhir yaitu memilih kategori atau kelas terbanyak yaitu kelas 1. Maka hasil dari klasifikasi adalah kelas 1.



## BAB 5 IMPLEMENTASI

Pada bab implementasi, akan dibahas tentang implementasi alur untuk menemukan solusi atau algoritme dari metode yang telah dirancang pada Bab Perancangan. Pada bab ini juga akan dijelaskan mengenai lingkungan implementasi, batasan algoritme dan implementasi algoritme. Dari Pada subbagian lingkungan implementasi, dijelaskan mengenai spesifikasi kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam penelitian. Pada subbagian batasan implementasi menjelaskan batasan selama penelitian. Pada subbagian implementasi algoritme menjelaskan kode program dari metode yang digunakan dalam penelitian, yaitu *Regioning* dan *Local Binary Pattern*.

### 5.1 Batasan Implementasi

Batasan implementasi diperlukan sebagai batasan dalam melakukan implementasi sehingga dapat sesuai dengan perancangan yang telah di buat dan sebagai sarana untuk memperjelas ruang lingkupnya. Batasan implementasi dalam penelitian ini adalah:

1. Teknik yang digunakan untuk mengekstraksi ciri yaitu menggunakan *regioning*, ekstraksi ciri *Local Binary Pattern* dan klasifikasi K-NN
2. Citra di-convert ke skala *grayscale* menggunakan bantuan *library* dari *OpenCv*.
3. Untuk mengelola array citra yang berjumlah banyak, penelitian ini memanfaatkan *library* dari *numpy*.
4. *Library Os* dimanfaatkan untuk mengakses *path* dan mengambil isi *file*.
5. Untuk menyimpan daftar citra dalam *folder* yang hendak diambil dan diproses yaitu menggunakan *library* dari *Cvutils*.
6. Klasifikasi K-NN memanfaatkan *library* dari *Sklearn*

### 5.2 Implementasi Algoritme

Algoritme yang diimplementasikan ke dalam program yaitu seperti yang telah dijelaskna pada perancangan antara lain yaitu ekstraksi fitur *Local Binary Pattern* dan menggunakan metode klasifikasi K-NN.

#### 5.2.1 Pre-Processing

Implementasi *Pre-Processing* di tunjukkan pada Gambar 5.1

<i>Pre-Processing</i>	
1	<code>def preprocessing(citra):</code>
2	<code>    citra = cv.cvtColor(citra, cv.COLOR_BGR2GRAY)</code>
3	<code>    return citra</code>

**Gambar 5.1 Source Code Pre-Processing**

Penjelasan :

Pada *method* ini berfungsi untuk mengubah citra berwarna menjadi citra keabuan dengan memanfaatkan *library OpenCv*.

## 5.2.2 Algoritme Regioning

Implementasi *Regioning* di tunjukkan pada Gambar 5.2

<i>Regioning</i>	
1	<b>def</b> region(image,divide):
2	
3	h, w = image.shape
4	# menghapus sisa
5	<b>if</b> h%divide[0] !=0:
6	sisa = h%divide[0]
7	h -= sisa
8	
9	<b>if</b> w%divide[1] !=0:
10	sisa = h%divide[0]
11	w -= sisa
12	
13	windowsize_brs = int(h/divide[0])
14	windowsize_klm = int(w/divide[1])
15	
16	# Inisialisasi array numpy
17	array_reg = []
18	# Cropping window
19	<b>for</b> x <b>in</b> range (int(h/windowsize_brs)):
20	<b>for</b> y <b>in</b> range (int(w/windowsize_klm)):
21	array_reg.append(image[x * windowsize_brs:
22	(x+1) * windowsize_brs, y * windowsize_klm: (y+1) *
23	windowsize_klm])
24	
25	<b>return</b> array_reg

**Gambar 5.2 Source Code Regioning**

Pembahasan :

1. Baris 3 adalah proses deklarasi dan inialisasi h dan w dari image shape.
2. Baris 5-7 merupakan proses menghapus sisa piksel yang tidak digunakan pada h.
3. Baris 9-11 adalah proses menghapus sis piksel yang tidak dibutuhkan pada w.
4. Baris 13-14 deklarasi dan inialisasi varibel windowsize\_brs danwindowsize\_klm.
5. Baris 17 merupakan deklarasi dari array\_reg[]
6. Baris 19-23 adalah proses membagi citra menjadi *region-region*.
7. Baris 25 mengembalikan nilai array\_reg[]

## 5.2.3 Algoritme Local Binary Pattern

Pada proses ekstraksi fitur *Local Binary Pattern* terdapat beberapa proses di dalamnya yaitu mengambil cek poin atau nilai tetangga yang berjumlah delapan dari piksel yang ingin diolah, merubah nilai pada piksel tetangga menjadi nilai biner, menyusun menjadi deret bilangan biner, ubah deret bilangan biner menjadi angka desimal. Implementasi *Local Binary Pattern* ditunjukkan pada Gambar 5.3.

Ekstraksi Fitur *Local Binary Pattern*

```

1  def lbp(array_reg):
2      array_LBP = np.zeros_like(array_reg)
3      baris = len(array_reg)
4      kolom = len(array_reg[0])
5
6      r = 1
7      p = 8
8      for brs in range(r, baris-r):
9          for klm in range(r, kolom-r):
10             biner = []
11
12             nilai = 1 if array_reg[brs][klm] <
13 array_reg[brs - r][klm - r] else 0
14             biner.append(nilai)
15
16             nilai = 1 if array_reg[brs][klm] <
17 array_reg[brs - r][klm] else 0
18             biner.append(nilai)
19
20             nilai = 1 if array_reg[brs][klm] <
21 array_reg[brs - r][klm + r] else 0
22             biner.append(nilai)
23
24             nilai = 1 if array_reg[brs][klm] <
25 array_reg[brs][klm + r] else 0
26             biner.append(nilai)
27
28             nilai = 1 if array_reg[brs][klm] <
29 array_reg[brs + r][klm + r] else 0
30             biner.append(nilai)
31
32             nilai = 1 if array_reg[brs][klm] <
33 array_reg[brs + r][klm] else 0
34             biner.append(nilai)
35
36             nilai = 1 if array_reg[brs][klm] <
37 array_reg[brs + r][klm - r] else 0
38             biner.append(nilai)
39
40             nilai = 1 if array_reg[brs][klm] <
41 array_reg[brs][klm - r] else 0
42             biner.append(nilai)
43
44             pangkat = p-1
45             hasilbiner = 0
46
47             for i in range(p):
48                 hasilbiner += pow(2, pangkat) * biner[i]
49                 pangkat -= 1
50
51             array_LBP[brs][klm] = hasilbiner
52     return array_LBP

```

Gambar 5.3 Source Code *Local Binary Pattern*

Pembahasan :

- Baris 1-52 merupakan *method* untuk menghitung LBP dimana :
  1. Baris 2 adalah pendeklarasian dan inialisasi array\_LBP yang berukuran sama dengan array citra yang sudah dilakukan proses *regioning* dengan isi array yang nilai 0.
  2. Baris 3-4 adalah deklarasi dan inialisasi variabel baris dan kolom yang sesuai dengan ukuran citra yang telah dilakukan proses *regioning*.
  3. Baris 6-7 merupakan deklarasi dan inialisasi r atau jarak tetangga dan p sebagai jumlah cek poin atau jumlah tetangga.
  4. Baris 8-42 merupakan proses pencarian ketetanggan, membandingkan piksel saat ini serta memberikan nilai 1 jika nilai tetangga yang dibandingkan bernilai lebih besar dari nilai piksel pusat dan sebaliknya apabila nilai tetangga yang dibandingkan lebih kecil dari nilai pusat maka akan diberi nilai 0.
  5. Baris 47-49 merupakan proses mengubah nilai biner menjadi nilai *decimal*.
  6. Baris 51 adalah proses memasukkan nilai *decimal* yang telah didapatkan ke dalam array\_LBP yang telah dibuat sebelumnya.
  7. Baris 52 mengembalikan array\_LBP.

### 5.2.4 Hitung Frekuensi

Implementasi hitung frekuensi ditunjukkan pada Gambar 5.4

Hitung Frekuensi	
1	<b>def</b> hitungFreq(array_LBP):
2	array_LBP_1d = array_LBP.ravel()
3	array_freq = np.bincount(array_LBP_1d)
4	<b>return</b> array_freq

Gambar 5.4 Source Code hitung frekuensi

Penjelasan :

1. Baris 2 yaitu proses mengubah *array* LBP 2 dimensi menjadi 1 dimensi
2. Baris 3 adalah proses menghitung histogram yang dilakukan dengan cara menghitung *frekuensi* dari 0-255 .
3. Baris 4 berfungsi untuk mengembalikan nilai histogram.

### 5.2.5 Normalisasi

Implementasi normalisasi *Histogram* di tunjukkan pada Gambar 5.5

Normalisasi	
1	<b>def</b> normalisasi(array_freq):
2	array_norm = array_freq / sum(array_freq)
3	<b>return</b> array_norm

### Gambar 5.5 Source Code normalisasi

Penjelasan :

1. Baris 2 yaitu proses normalisasi histogram.
2. Baris 3 berfungsi untuk mengembalikan nilai normalisasi histogram.

### 5.2.6 Konkatenasi

Implementasi konkatenasi ditunjukkan pada Gambar 5.6

Konkatenasi	
1	histogram.extend(array_norm)

### Gambar 5.6 Source Code Konkatenasi

Penjelasan :

1. Untuk melakukan konkatenasi fitur.

### 5.2.7 Klasifikasi K-NN

Implementasi klasifikasi K-NN di tunjukkan pada Gambar 5.7

Klasifikasi K-NN	
1	<b>def</b> knn(histo_latih, label_latih, histogram, kelas):
2	ambil = knn(n_neighbors=5)
3	ambil.fit(histo_latih,label_latih)
4	
5	truth = "Sesuai" <b>if</b>
6	str(kelas).casefold()==(ambil.predict([histogram]))[0].casefo
7	ld() <b>else</b> "Tidak Sesuai"
8	hasil = "Kelas asli : " + str(kelas) + " <b>hasil klasifikasi</b>
9	: " + str((ambil.predict([histogram]))[0]) + "\t" + truth
10	return hasil

### Gambar 5.7 Source Code Klasifikasi K-NN

Penjelasan :

1. Baris 2 yaitu menentukan nilai k = 5
2. Baris 3 untuk melabeli histo\_latih dengan label\_latih.
3. Baris 5 digunakan untuk mencocokkan hasil klasifikasi apakah bernilai "sesuai" atau "tidak sesuai" dengan label asli data uji.
4. Baris 8 untuk menuliskan hasil.
5. Mengembalikan nilai hasil

## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini berisi pengujian data yang telah diperoleh, setelah itu akan dilakukan proses analisis pada telapak tangan dengan menggunakan metode *Local Binary Pattern* telah dirancang pada bab 4 dan dilakukan implementasi pada bab 5.

### 6.1 Pengujian

Pada subbab ini akan dibahas tentang deskripsi penjelasan sesuai dengan rancangan pengujian. Skenario pengujian pada penelitian ini dijelaskan sebagai berikut:

1. Pengujian untuk mendapatkan jumlah *region* dan nilai pembagian *height* dan *width* terbaik.
2. Pengujian untuk mendapatkan nilai *r* atau jarak tetangga yang terbaik.

#### 6.1.1 Pengujian Jumlah *Region* dan Nilai *Height* dan *Width*

Pada pengujian pertama ini bertujuan mencari jumlah *region* dan nilai *height* dan *width* terbaik untuk mengekstraksi ciri pada telapak tangan. Pada pengujian ini menggunakan citra berukuran seragam yaitu 640 x 480 piksel. Citra dimasukkan setelah dilakukan proses *grayscale*. Setelah itu citra *grayscale* tersebut akan dibagi menjadi beberapa *region*. Jumlah *region* merupakan hasil dari perkalian nilai *height* dan *width*. Pengujian pada nilai *height* dan *width* yaitu dilakukan dengan cara mengkombinasikan nilai dari *height* dan *width* sehingga berjumlah *n-region* citra telapak tangan. Jumlah *region* dan nilai *height* dan *width* yang ditunjukkan pada Tabel 3.1

Citra yang digunakan dalam pengujian ini yaitu sebanyak 208 buah citra telapak tangan yang telah dipotong untuk pengujian. Jumlah data latih sejumlah 130 buah dan data uji sebanyak 78 buah.

Hasil dari pengujian ini merupakan jumlah *region* dan nilai *height* dan *width* terbaik yang dapat menjelaskan pengaruh *region* pada Ekstraksi Ciri Pada Telapak Tangan Dengan Metode *Local Binary Pattern* (LBP). Hasil dari pengujian ini ditunjukkan pada Tabel 6.2.

#### 6.1.2 Pengujian Jarak Tetangga

Pada pengujian kali ini akan menguji pengaruh parameter *r* atau jarak tetangga dari LBP. Parameter jarak tetangga yang akan diuji yaitu 1, 2, dan 3 dan jumlah cek poin 8. Dimana setiap *r* akan dilakukan percobaan 16 kali dengan jumlah *region* 1, 2, 3, 4, 6, 8, 9, 12, 16 dan menggunakan kombinasi *height* dan *width* seperti yang ditunjukkan oleh Tabel 3.1.

Hasil dari pengujian ini ditujukan untuk mencari parameter jarak tetangga terbaik dengan cek poin yang berjumlah 8 dari metode *Local Binary Pattern*. Hasil dari pengujian direpresentasikan oleh Tabel 6.1, Tabel 6.2 dan Tabel 6.3.

## 6.2 Hasil Pengujian dan Analisis

Hasil pengujian ini akan menampilkan hasil dari skenario pengujian yang telah dilakukan. Hasil pengujian dengan nilai jarak tetangga ( $R$ ) = 1 dan jumlah cek poin ( $P$ ) = 8 ditunjukkan oleh Tabel 6.2, hasil pengujian dengan nilai jarak tetangga ( $R$ ) = 2 dan jumlah cek poin ( $P$ ) = 8 ditunjukkan oleh Tabel 6.3, hasil pengujian dengan nilai jarak tetangga = 3 dan jumlah cek poin = 8 ditunjukkan oleh Tabel 6.4.

**Tabel 6.1 Hasil Pengujian  $R = 1$  ,  $P = 8$**

Jumlah Region	Height	Width	R = 1 , P = 8		
			Jumlah Salah	Jumlah Benar	Akurasi
1	1	1	15	63	80,77%
2	1	2	15	63	80,77%
3	1	3	13	65	83,33%
4	1	4	12	66	84,62%
2	2	1	13	65	83,33%
4	2	2	12	66	84,62%
6	2	3	9	69	88,46%
8	2	4	9	69	88,46%
3	3	1	10	68	87,18%
6	3	2	9	69	88,46%
9	3	3	8	70	89,74%
12	3	4	7	71	91,03%
4	4	1	10	68	87,18%
8	4	2	8	70	89,74%
12	4	3	8	70	89,74%
16	4	4	7	71	91,03%

Pada pengujian jarak tetangga ( $R$ )= 1 dan jumlah cek poin ( $P$ ) = 8 mendapatkan hasil sebagai berikut:

- *Height* = 1 : terjadi peningkatan hasil akurasi pada setiap *width*.
- *Height* = 2 : terjadi peningkatan hasil akurasi pada *width* 1, 2 dan 3, namun konvergen pada *width* = 4.
- *Height* = 3 : terjadi peningkatan hasil akurasi pada setiap *width*.
- *Height* = 4 : terjadi terjadi penurunan hasil akurasi pada *width* = 2 dan 3, namun terjadi peningkatan kembali pada *width* = 4.

Akurasi tertinggi terdapat pada jumlah *region* = 12 dengan jumlah pembagian pada *height* = 3 dan *width* = 4 serta jumlah *region* = 16 dengan jumlah pembagian pada *height* = 4 dan *width* = 4 dengan akurasi sebesar 91,03%.

**Tabel 6.2 Hasil Pengujian R = 2 , P = 8**

Jumlah Region	Height	Width	R = 2 , P = 8		
			Jumlah Salah	Jumlah Benar	Akurasi
1	1	1	14	64	82,05%
2	1	2	10	68	87,18%
3	1	3	9	69	88,46%
4	1	4	9	69	88,46%
2	2	1	11	67	85,90%
4	2	2	8	70	89,74%
6	2	3	7	71	91,03%
8	2	4	9	69	88,46%
3	3	1	10	68	87,12%
6	3	2	9	69	88,46%
9	3	3	9	69	88,46%
12	3	4	10	68	87,12%
4	4	1	7	71	91,03%
8	4	2	10	68	87,12%
12	4	3	7	71	91,03%
16	4	4	6	72	92,31%

Pada pengujian jarak tetangga (R) = 2 dan jumlah cek poin (P) = 8 mendapatkan hasil sebagai berikut:

- *Height* = 1 : terjadi peningkatan hasil akurasi pada *width* = 1, 2, 3 namun konvergen pada *width* = 4.
- *Height* = 2 : terjadi peningkatan hasil akurasi pada *width* 1, 2 dan 3, akan tetapi terjadi penurunan pada *width* = 4.
- *Height* = 3 : terjadi peningkatan hasil akurasi pada *width* = 1, 2 namun konvergen pada *width* = 3 kemudian mengalami peningkatan pada *width* = 4.
- *Height* = 4 : terjadi terjadi penurunan hasil akurasi pada *width* = 1, 2, namun terjadi peningkatan kembali pada *width* = 3 dan 4.

Akurasi tertinggi terdapat pada jumlah *region* = 16 dengan jumlah pembagian pada *height* = 4 dan *width* = 4 dengan akurasi sebesar 92,31%.

**Tabel 6.3 Hasil Pengujian R = 3 , P = 8**

Jumlah Region	Height	Width	R = 3 , P = 8		
			Jumlah Salah	Jumlah Benar	Akurasi
1	1	1	12	66	84,62%

Tabel 6.3 Hasil Pengujian R = 3 , P = 8(lanjutan)

Jumlah <i>Region</i>	<i>Height</i>	<i>Width</i>	R = 3 , P = 8		
			Jumlah Salah	Jumlah Benar	Akurasi
2	1	2	9	69	88,46%
3	1	3	10	68	87,18%
4	1	4	10	68	87,18%
2	2	1	13	15	83,33%
4	2	2	12	66	84,62%
6	2	3	10	68	87,18%
8	2	4	9	69	88,46%
3	3	1	11	67	85,90%
6	3	2	11	67	85,90%
9	3	3	10	68	87,18%
12	3	4	11	67	85,90%
4	4	1	11	67	85,90%
8	4	2	11	67	85,90%
12	4	3	9	69	88,46%
16	4	4	9	69	88,46%

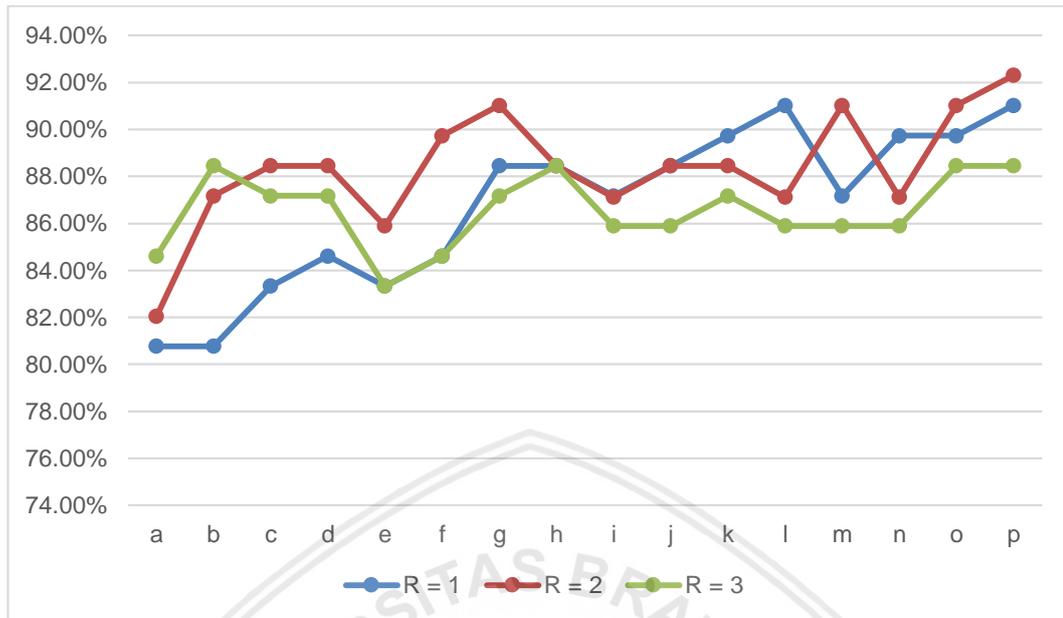
Pada pengujian jarak tetangga (R) = 3 dan jumlah cek poin (P) = 8 mendapatkan hasil sebagai berikut:

- *Height* = 1 : terjadi peningkatan hasil akurasi pada *width* = 1, 2, akan tetapi terjadi penurunan pada *width* = 3, dan konvergen pada *width* = 4.
- *Height* = 2 : terjadi peningkatan hasil akurasi pada setiap *width*.
- *Height* = 3 : konvergen pada *width* = 2, dan taerjadi peningkatan pada *width* = 3, namun terjadi peningkatan akurasi pada *width* = 4.
- *Height* = 4 : konvergen pada *width* = 2, dan taerjadi peningkatan akurasi pada *width* = 3, namun konvergen pada *width* = 4.

Akurasi tertinggi yang didapatkan yaitu sebesar 88,46% yang terdapat pada beberapa jumlah *region* yang berbeda antara lain yaitu:

1. Jumlah *region* = 2 dengan jumlah pembagian pada *height* = 1 dan *width* = 2.
2. Jumlah *region* = 8 dengan jumlah pembagian pada *height* = 2 dan *width* = 4.
3. Jumlah *region* = 12 dengan jumlah pembagian pada *height* = 4 dan *width* = 3.
4. Jumlah *region* = 16 dengan jumlah pembagian pada *height* = 4 dan *width* = 4.

Dari percobaan tersebut ditunjukkan oleh grafik yang ditunjukkan Gambar 6.1



**Gambar 6.1 Grafik Analisis Pengujian**

Keterangan dari Gambar 6.1 dijelaskan oleh Tabel 6.5.

**Tabel 6.4 Keterangan**

Kode	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
<i>Height</i>	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
<i>Width</i>	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4

Berdasarkan gambar 6.1 dapat dideskripsikan bahwa hasil terbaik pada kode:

1. Kode a memiliki akurasi terbaik pada jarak tetangga = 3 dan jumlah cek poin = 8 yaitu sebesar 84,62%.
2. Kode b memiliki akurasi terbaik pada jarak tetangga = 3 dan jumlah cek poin = 8 yaitu sebesar 88,46%.
3. Kode c memiliki akurasi terbaik pada jarak tetangga = 2 dan jumlah cek poin = 8 yaitu sebesar 88,46%.
4. Kode d memiliki akurasi terbaik pada jarak tetangga = 2 dan jumlah cek poin = 8 yaitu sebesar 88,46%.
5. Kode e memiliki akurasi terbaik pada jarak tetangga = 2 dan jumlah cek poin = 8 yaitu sebesar 85,90%.
6. Kode f memiliki akurasi terbaik pada jarak tetangga = 2 dan jumlah cek poin = 8 yaitu sebesar 89,74%.

7. Kode g memiliki akurasi terbaik pada jarak tetangga = 2 dan jumlah cek poin = 8 yaitu sebesar 91,03%.
8. Kode h memiliki akurasi yang sama pada ketiga jarak dan jumlah cek poin = 8 yaitu sebesar 88,46%.
9. Kode i memiliki akurasi terbaik pada jarak tetangga = 2 dan jumlah cek poin = 8 yaitu sebesar 87,12%.
10. Kode j memiliki akurasi terbaik pada jarak tetangga = 1, 2 dan jumlah cek poin = 8 yaitu sebesar 88,46%.
11. Kode k memiliki akurasi terbaik pada jarak tetangga = 1 dan jumlah cek poin = 8 yaitu sebesar 88,46%.
12. Kode l memiliki akurasi terbaik pada jarak tetangga = 2 dan jumlah cek poin = 8 yaitu sebesar 89,74%.
13. Kode m memiliki akurasi terbaik pada jarak tetangga = 2 dan jumlah cek poin = 8 yaitu sebesar 91,03%.
14. Kode n memiliki akurasi terbaik pada jarak tetangga = 1 dan jumlah cek poin = 8 yaitu sebesar 89,74%.
15. Kode o memiliki akurasi terbaik pada jarak tetangga = 2, 3 dan jumlah cek poin = 8 yaitu sebesar 88,46%.
16. Kode c memiliki akurasi terbaik pada jarak tetangga = 2 dan jumlah cek poin = 8 yaitu sebesar 92,31%.

Dari grafik hasil akurasi pada Gambar 6.1, terdapat kecenderungan kenaikan nilai akurasi dengan perlakuan *width* berbeda namun *height* yang sama. Misalnya pada pengujian nilai  $r=2$  dan *height* bernilai 1, nilai akurasi meningkat dari 82,05%, 87,18%, 88,46%, dan 88,46%. Hasil ini didapatkan dari pengubahan nilai *width* dari 1, 2, 3, dan 4. Hal ini berlaku kepada hampir semua nilai *height* meskipun ada nilai yang mengalami konvergen.

Dari seluruh pengujian hasil terbaik terdapat pada parameter jarak tetangga = 2, jumlah cek poin = 8 yaitu terdiri dari 16 *region* dimana pembagian nilai *height* = 4 dan *width* 4 yaitu sebesar 92,31%.

## BAB 7 KESIMPULAN DAN SARAN

Pada bab ini akan membahas tentang kesimpulan dan saran. Kesimpulan yang diambil dari hasil penelitian. Saran yang diberikan dapat digunakan sebagai referensi untuk penelitian berikutnya agar lebih baik lagi.

### 7.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis dapat ditarik kesimpulan bahwa :

1. Semakin banyak jumlah *region* dan pembagian citra telapak tangan yang tepat dengan menggunakan metode *Local Binary Pattern* maka akan menghasilkan hasil yang optimal. Hal ini terjadi karena semakin banyak jumlah *region* maka semakin banyak fitur lokal yang terbentuk. Hasil akurasi terbaik yaitu 92,31% dengan jumlah *region* sebanyak 16.
2. Parameter jarak tetangga dari metode *Local Binary Pattern* mempengaruhi hasil akurasi. Ketika jarak tetangga bernilai 1 maka hasil yang didapatkan akan lebih stabil dan meningkat, ketika jarak tetangga bernilai 2 maka akurasi lebih bagus akan tetapi ketika pembagian *height* dan *width* dari citra tidak tepat maka akurasi menurun, untuk jarak tetangga yang bernilai 3 cocok untuk pembagian citra dengan nilai *height* = 1 dan *width* = 2, serta *height* = 2 dan *width* = 4 ataupun citra yang tidak dilakukan pembagian. Tingkat akurasi tertinggi yaitu pada jarak tetangga = 2 dengan pembagian citra menjadi 16 *region*, *height* = 4 dan *width* = 4 yaitu sebesar 92,31%.

### 7.2 Saran

Berdasarkan hasil dan kesimpulan yang didapat maka penelitian ini dapat disempurnakan dan dikembangkan. Saran yang dapat diberikan oleh peneliti untuk penelitian berikutnya yaitu:

1. Menggunakan *Region Of Interest* (ROI) untuk melakukan proses *cropping* manual.
2. Menggunakan telapak tangan kanan dan kiri sebagai data latih.
3. Menggunakan telapak tangan yang memiliki *rotasi* yang berbeda.
4. Mengoptimasi parameter jumlah cek poin pada LBP.
5. Menerapkan proses *regioning* atau pembagian citra setelah dilakukan proses ekstraksi ciri LBP.

## DAFTAR PUSTAKA

- Abdesselam, A., 2013. Improving Local Binary Patterns Techniques by Using Edge Information. *Lecture Notes on Software Engineering*, [online] 1(4), pp.360–363. Available at: <<http://www.lnse.org/show-31-94-1.html>>.
- Achsani, F.N., Atmaja, R.D. and Purnamasari, R., 2015. Deteksi Adanya Cacat Pada Kayu Menggunakan Metode Local Binary Pattern. 2(1), pp.298–305.
- Adi, Saputra, Tjokorda, A.B.W., 2015. Pengenalan Ekspresi Wajah Menggunakan Local Binary Pattern ( LBP ). (1103081084).
- Ahmed, Z.S., 2015. Palmprint Recognition with Statistical , Wavelet and Local Feature Extraction Methods. (February).
- Ali, M.M.H., Mahale, V.H., Yannawar, P. and Gaikwad, A.T., 2016. Study Of edge Detection Detection Methods Based On Palmprint Lines. *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, (October 2017), pp.1334–1338.
- Amyarto, N., Sari, Y.A. and Cahyawihandika, R., 2018. Pengenalan Emosi Berdasarkan Ekspresi Mikro Menggunakan Metode Local Binary Pattern. 2(10), pp.3230–3238.
- Basya, I.M., Wirayuda, T.A.B., Rachmawati, E. and Kunci, K., 2012. Sistem identifikasi biometrik telapak tangan manusia menggunakan metode haar wavelet dan support vector machine (svm).
- Bhattacharyya, D., Ranjan, R., a, F.A. and Choi, M., 2009. Biometric Authentication : A Review. *International Journal of Service, Science and technology*, 2(3), pp.13–28.
- Bustami, 2014. Penerapan Algoritma Naive Bayes Untuk Mengklasifikasi Data Nasabah Asuransi. *Jurnal Informatika*, 8(1), pp.884–898.
- Dharavath, K., Talukdar, F. a. and Laskar, R.H., 2013. Study on biometric authentication systems, challenges and future trends: A review. *2013 IEEE International Conference on Computational Intelligence and Computing Research*, [online] (July 2014), pp.1–7. Available at: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6724278>>.
- Han, D., Guo, Z. and Zhang, D., 2008. Multispectral palmprint recognition using wavelet-based image fusion. *International Conference on Signal Processing Proceedings, ICSP*, pp.2074–2077.
- Ibadurrahman, H., Wirayuda, T.A.B. and Hevanie, F., 2014. BINARY PATTERN ( LBP ) DAN PROPAGASI BALIK LEVENBERG MARQUARDT.
- Jain, A.K., Nandakumar, K. and Nagar, A., 2008. Biometric template security. *Eurasip Journal on Advances in Signal Processing*, 2008.

- Manegopale, P., 2014. A Survey on Palmprint Recognition. 3(2), pp.9085–9094.
- Mujib, K., Hidayatno, A. and Prakoso, T., 2018. PENGENALAN WAJAH MENGGUNAKAN LOCAL BINARY PATTERN ( LBP ) DAN SUPPORT VECTOR MACHINE ( SVM ).
- N, J.. and Pato . Millett, L.I., 2010. *Biometric Recognition : Challenges and Opportunities*. [online] Available at: <[http://www.nap.edu/catalog.php?record\\_id=12720%0Awww.nap.edu%0Awww.national-academies.org](http://www.nap.edu/catalog.php?record_id=12720%0Awww.nap.edu%0Awww.national-academies.org)>.
- Ojala, T., Pietikäinen, M. and Harwood, D., 1996. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1), pp.51–59.
- Ojala, T., Pietikäinen, M. and Maenpää, T., 2002. Multiresolution Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *Proceedings - International Conference on Image Processing, ICIP*, 24(7), pp.1852–1855.
- Purnamasari, I., Sutojo, T., Informatika, J., Dian, U. and Semarang, N., 2017. Pengenalan Ciri Garis Telapak Tangan Menggunakan Ekstraksi Fitur ( GLCM ) Dan Metode K-NN. 10(2), pp.221–229.
- Sari, J.Y. and Saputra, R.A., 2017. Pengenalan Finger Vein Menggunakan Local Line Binary Pattern dan Learning Vector Quantization. (December).
- Turiyanto, M.D., Purwanto, D. and Dikairono, R., 2014. Penerapan Teknik Pengenalan Wajah Berbasis Fitur Local Binary Pattern pada Robot Pengantar Makanan. pp.1–6.
- Wibawa, P.A., Wirayuda, T.A.B. and Hevanie, F., 2014. Sistem Biometrik Berbasis Pola Telapak Tangan Menggunakan Metode Local Binary Pattern Dan Probabilistic Neural Network.
- Zhang, D., Kong, W.K., You, J. and Wong, M., 2003. Online palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9), pp.1041–1050.