

**IDENTIFIKASI KEMATANGAN BUAH PEPAYA  
MENGGUNAKAN METODE LBP (*LOCAL*  
*BINARY PATTERN*) DAN K-NN  
(*K-NEAREST NEIGHBORS*)**

**Oleh  
SELA MARSELA ISMAIL  
T3116070**

**SKRIPSI  
Untuk memenuhi salah satu syarat ujian  
Guna memperoleh gelar sarjana**



**PROGRAM SARJANA  
TEKNIK INFORMATIKA  
UNIVERSITAS IHSAN GORONTALO  
GORONTALO  
2021/2022**

## **HALAMAN JUDUL**

# **IDENTIFIKASI KEMATANGAN BUAH PEPAYA MENGGUNAKAN METODE LBP (*LOCAL BINARY PATTERN*) DAN K-NN (*K-NEAREST NEIGHBORS*)**

**Oleh**  
**SELA MARSELIA ISMAIL**  
**T3116070**

**SKRIPSI**  
**Untuk memenuhi salah satu syarat ujian**  
**Guna memperoleh gelar sarjana**



**PROGRAM SARJANA  
TEKNIK INFORMATIKA  
UNIVERSITAS ICHSAN GORONTALO  
GORONTALO  
2021/2022**

**PENGESAHAN SKRIPSI**

**IDENTIFIKASI KEMATANGAN BUAH PEPAYA**

**MENGGUNAKAN METODE LBP (*LOCAL***

***BINARY PATTERN*) DAN K-NN**

**(*K-NEAREST NEIGHBORS*)**

Oleh

SELA MARSELIA ISMAIL

T3116070

**SKRIPSI**

Untuk memenuhi salah satu syarat ujian  
Guna memperoleh gelar sarjana  
Program Studi Teknik Informatika  
Ini telah disetujui oleh Tim Pembimbing  
Gorontalo, Juni 2021

Pembimbing Utama

Pembimbing Pendamping

---

Haditsa Annur, S.kom., M.Kom  
NIDN. 0908058403

---

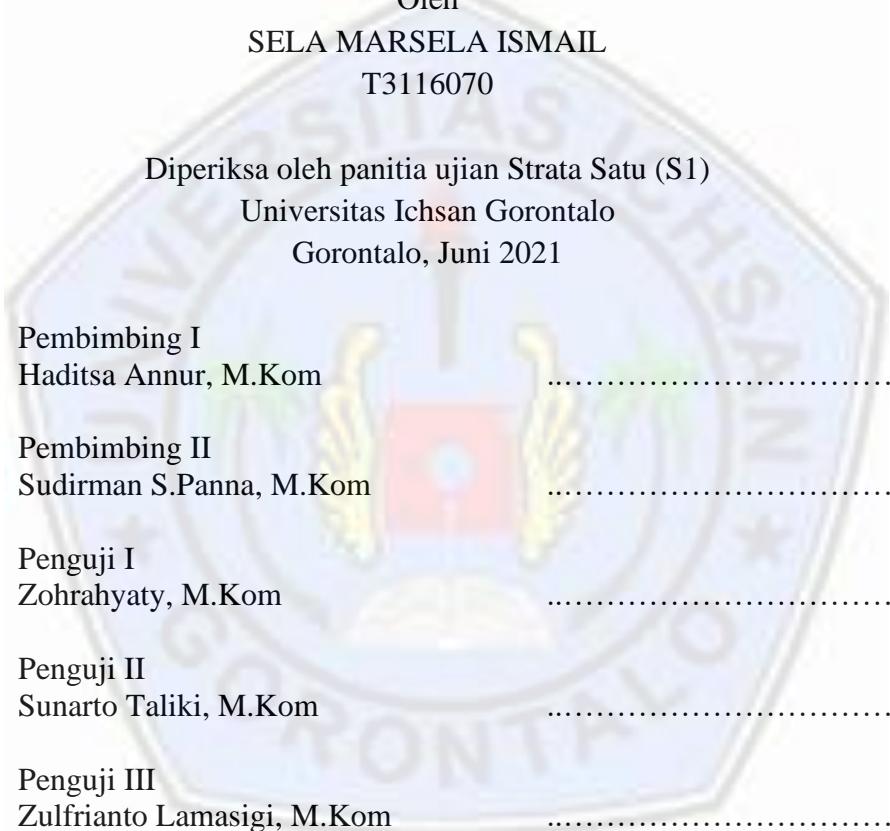
Sudirman S Panna, S.Kom., M.Kom  
NIDN. 0908017702

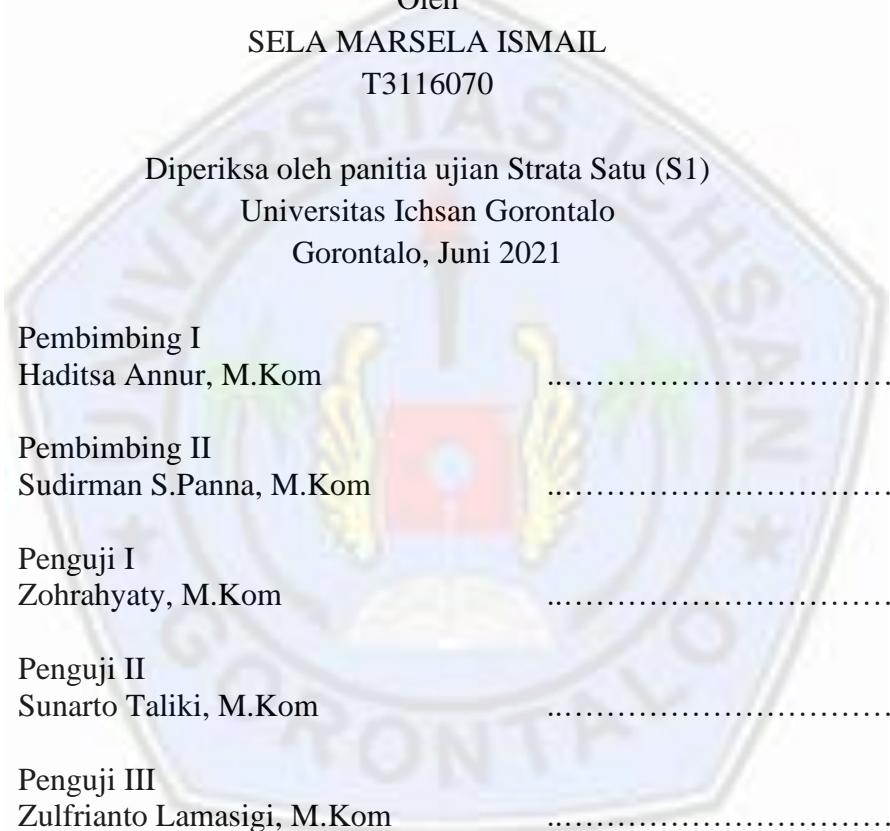
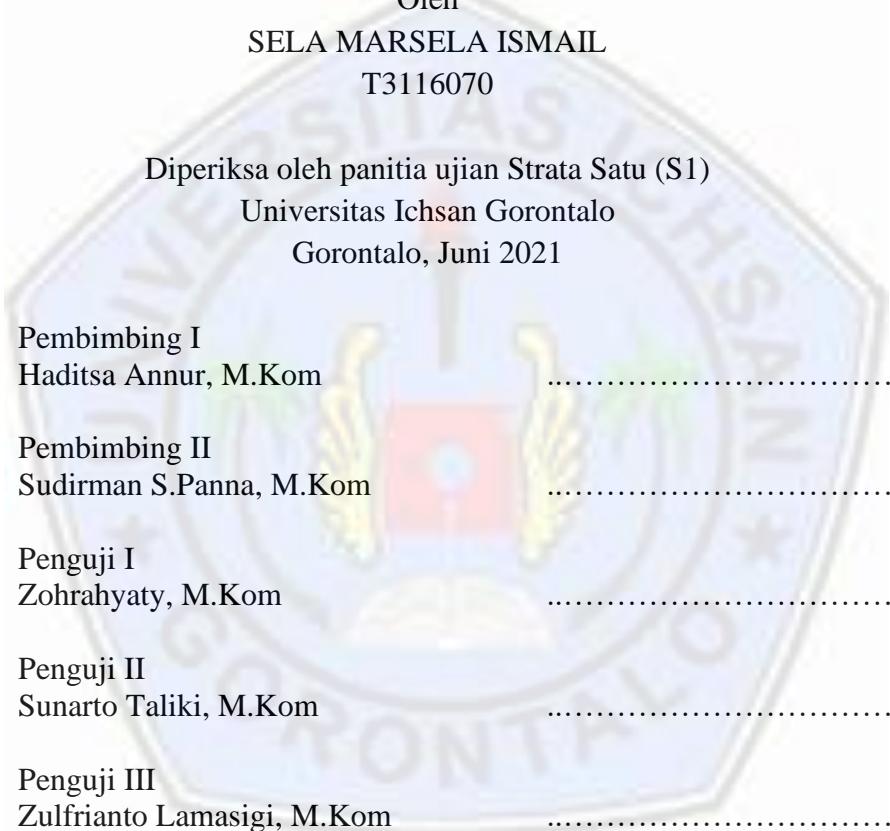
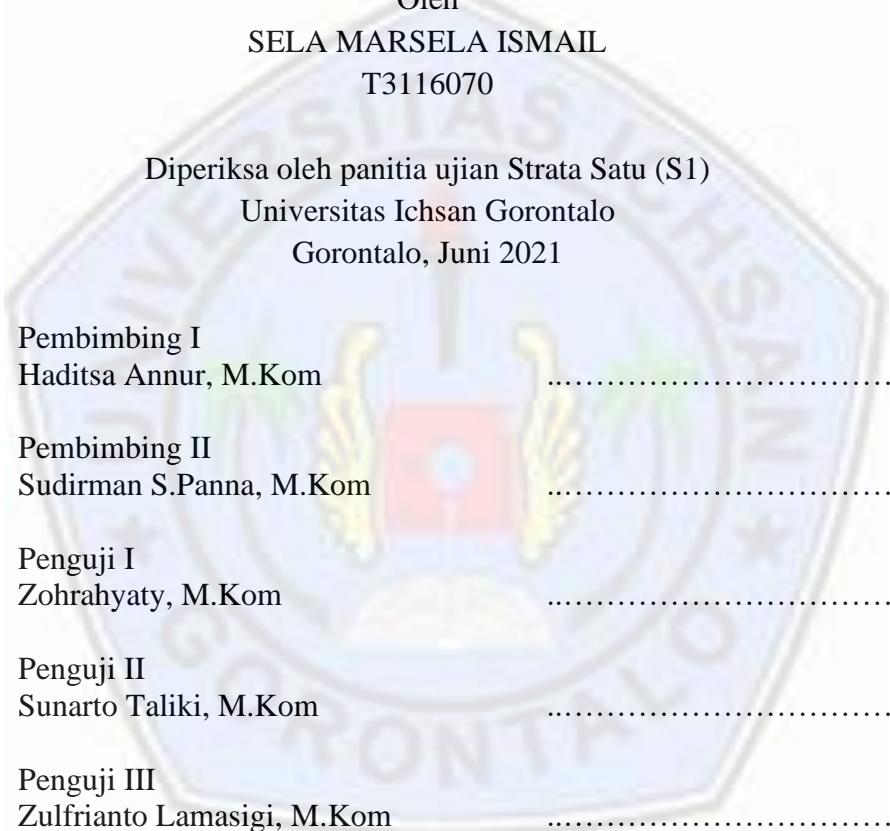
**PERSETUJUAN SKRIPSI**

**IDENTIFIKASI KEMATANGAN BUAH PEPAYA**  
**MENGGUNAKAN METODE LBP (*LOCAL***  
***BINARY PATTERN*) DAN K-NN**  
**(*K-NEAREST NEIGHBORS*)**

Oleh  
SELA MARSELIA ISMAIL  
T3116070

Diperiksa oleh panitia ujian Strata Satu (S1)  
Universitas Ichsan Gorontalo  
Gorontalo, Juni 2021

1. Pembimbing I  
Haditsa Annur, M.Kom .....  

2. Pembimbing II  
Sudirman S.Panna, M.Kom .....  

3. Penguji I  
Zohrahayaty, M.Kom .....  

4. Penguji II  
Sunarto Taliki, M.Kom .....  

5. Penguji III  
Zulfrianto Lamasigi, M.Kom .....  


**Dekan Fakultas**

**Zohrahayaty, M.Kom**  
**NIDN. 0912117702**

**Ketua Jurusan**

**Sudirman S. Panna M.Kom**  
**NIDN. 0908017702**

## **PERNYATAAN SKRIPSI**

Dengan ini saya menyatakan bahwa:

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam Karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasi orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naska dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, Juni 2021

Yang Membuat Pernyataan

Sela Marsela Ismail

## ***ABSTRACT***

### ***SELA MARSELA ISMAIL. T3116070. IDENTIFICATION OF PAPAYA RIPENESS BY USING LOCAL BINARY PATTERN (LBP) AND K-NEAREST NEIGHBOR (K-NN) METHODS***

*Papaya has the Latin name Carica papaya which belongs to the Caricaceae family. Papaya is a tropical fruit of horticultural commodities with high nutritional and economic value. Papaya California (Calina) is one of the superior varieties of papaya which is currently being favored and planted by many farmers because it is very profitable. The identification and detection of the ripeness of papaya are still done manually. Manual identification is strongly influenced by the subjectivity of the observer. The weakness is the difference in everyone's perception of the assessment of the level of ripeness of papaya fruit. Therefore, it is necessary for an application based on Image Processing Technology that allows to sort and observe agricultural and plantation products automatically. The determination of papaya ripeness detection can be supported by using the Local Binary Pattern and K-Nearest Neighbor feature extraction methods as the classification model. The results of the study show that the accuracy of the K-NN model for determining the maturity level of papaya fruit is 75% by using the test on the K-NN model experiment with the number of K-1. This study uses 45 testing data and 150 training data, namely ripe, almost ripe, and unripe categories with 50 data each. The application of the Local Binary Pattern and K-Nearest Neighbor methods in relation to the problems managed to get a fairly high accuracy value of 75% by using the Confusion Matrix test.*

***Keywords:*** *papaya, identification, Local Binary Pattern, K-Nearest Neighbor*



## ABSTRAK

### SELA MARSELA ISMAIL. T3116070. IDENTIFIKASI KEMATANGAN BUAH PEPAYA MENGGUNAKAN METODE LOCAL BINARY PATTERN (LBP) DAN K-NEAREST NEIGHBOR (K-NN).

Pepaya memiliki nama latin *Carica papaya*, termasuk dalam keluarga *Caricaceae*. Pepaya merupakan buah tropis komoditas hortikultura yang mempunyai nilai gizi dan ekonomi yang tinggi. Pepaya *California* (*Calina*) merupakan salah satu varietas unggul buah pepaya yang saat ini sedang digandrungi dan banyak dikebun oleh para petani karena sangat menjanjikan keuntungan. Mengidentifikasi atau mendeteksi kematangan buah pepaya masih dilakukan secara manual, Identifikasi secara manual sangat di pengaruhi oleh subjektifitas orang yang mengamati. Yang menjadi kelemahan adalah perbedaan persepsi setiap orang tentang penilaian terhadap tingkat kematangan buah pepaya. Maka dari itu dibutuhkan sebuah aplikasi yang berbasis Teknologi Pengolahan Citra yang memungkinkan untuk memilah dan mengamati produk pertanian dan perkebunan secara otomatis. Penentuan deteksi tingkat kematangan buah pepaya dapat diselesaikan dengan menggunakan metode ekstraksi fitur *Local Binary Pattern* dan *K-Nearest Neighbor* sebagai model Klasifikasinya. Hasil penelitian menunjukkan bahwa akurasi model K-NN untuk penentuan tingkat kematangan buah pepaya adalah sebesar 75% dengan menggunakan pengujian pada percobaan model K-NN dengan jumlah K-1. penelitian ini menggunakan 45 data testing dan 150 data training, yaitu kategori matang, mengkal dan mentah dengan masing-masing 50 data.

Dengan demikian, Penerapan metode *Local Binary Pattern* dan *K-Nearest Neighbor* pada masalah diatas berhasil mendapatkan nilai akurasi yang cukup tinggi yakni sebesar 75% dengan menggunakan pengujian *Confusion Matrix*.

**Kata Kunci:** Pepaya, Identifikasi, *Local Binary Pattern*, *K-Nearest Neighbor*.



## KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan usulan penelitian ini dengan Judul: "**Identifikasi Kematangan Buah Pepaya Menggunakan Metode LBP (Local Binary Pattern) dan K-NN (K-Nearest Neighbors)**", untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa usulan penelitian ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada:

1. Bapak Muhammad Ichsan Gaffar, SE, M.Ak, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi(YPIPT) Ichsan Gorontalo;
2. Bapak Dr.Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Bapak Jorry Karim, S.Kom, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman S.Panna, S.Kom, M.Kom, selaku Ketua Jurusan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo dan Pembimbing II;
5. Ibu Irma Surya Kumala Idris, S.Kom, M.Kom, selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Sudirman Melangi, S.Kom, M.Kom, selaku Wakil Dekan I Bidang Kemahasiswaan Fakultas Ilmu komputer Universitas Ichsan Gorontalo;
7. Ibu Haditsah Annur, S.Kom, M.Kom, selaku Pembimbing I;
8. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
9. Kedua Orang Tua tercinta, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;

10. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
11. Kepada semua pihak yang ikut membantu dalam penyelesaian skripsi ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah SWT melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, Juni 2021

Sela Marsela Ismail

## DAFTAR ISI

HALAMAN JUDUL.....	i
PENGESAHAN SKRIPSI .....	ii
PERSETUJUAN SKRIPSI .....	iii
PERNYATAAN SKRIPSI.....	iv
<i>ABSTRACT</i> .....	v
ABSTRAK .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN .....	1
1. 1.Latar Belakang .....	1
1. 2.Identifikasi Masalah .....	3
1. 3.Rumusan Masalah .....	3
1. 4.Tujuan Penelitian.....	4
1. 5.Manfaat Penelitian.....	4
1. 5. 1.    Manfaat Teoritis.....	4
1. 5. 2.    Manfaat Praktis .....	4
BAB II LANDASAN TEORI.....	5
2. 1.Tinjauan Studi .....	5
2. 2.Tinjauan Pustaka .....	7
2. 2. 1.    Buah Pepaya .....	7

2. 2. 2.	Pepaya <i>California</i> .....	7
2. 2. 3.	Kriteria Kematangan Buah Pepaya <i>California</i> .....	8
2. 2. 4.	Citra ( <i>Image</i> ).....	8
2. 2. 5.	Pengolahan Citra.....	9
2. 2. 6.	Ekstraksi Fitur.....	10
2. 2. 7.	Metode <i>Local Binary Pattern</i> (LBP) .....	10
2. 2. 8.	Metode K-NN ( <i>K- Nearest Neighbor</i> ).....	11
2. 2. 9.	Perangkat Lunak Pendukung .....	15
2. 2.10.	Evaluasi Model .....	16
2. 2. 11.	Pengujian Sistem.....	17
2. 3.	Kerangka Piki .....	19
BAB III METODE PENELITIAN .....		20
3. 1.	Jenis, Metode, Subjek, Objek, Waktu dan Lokasi Penelitian.....	20
3. 2.	Pengumpulan Data .....	20
3. 3.	Pemodelan .....	21
3. 3. 1.	Pra Pengolahan .....	22
3. 3. 2.	Ekstraksi Ciri .....	22
3. 3. 3.	Data Training .....	22
3. 3. 4.	Data Testing .....	22
3. 3. 5.	Hasil Klasifikasi.....	22
3. 3. 6.	Evaluasi Model .....	22
BAB IV HASIL PENELITIAN .....		23
4.1	Hasil Pengumpulan Data .....	23
4.2	Hasil Pemodelan.....	24
4.2.1	Pra-Pengolahan.....	24

4.2.2	Ekstraksi Citra .....	24
4.2.3	Klasifikasi.....	27
4.2.4	Evaluasi Model.....	33
	BAB V PEMBAHASAN .....	41
5.1	Pembahasan Sistem .....	41
5.2	Pembahasan Model.....	43
	BAB VI PENUTUP .....	45
6.1	KESIMPULAN .....	45
6.2	SARAN .....	45
	DAFTAR PUSTAKA .....	xvi
	LAMPIRAN	

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Buah Pepaya California .....	8
<b>Gambar 2. 2</b> Langkah Pertama LBP.....	10
<b>Gambar 2. 3</b> Langkah Kedua LBP .....	11
<b>Gambar 2. 4</b> Langkah Terakhir LBP .....	11
<b>Gambar 2. 5</b> Diagram Hasil Pengujian.....	15
<b>Gambar 2. 6</b> Kerangka Pikir.....	19
<b>Gambar 3. 1</b> Pemodelan .....	21
<b>Gambar 4. 2</b> Matrix 3x3 yang diambil pada pixel 6x6.....	26
<b>Gambar 4. 3</b> Perhitungan nilai decimal dari matrix 3x3 .....	26
<b>Gambar 5. 1</b> Tampilan Menu Utama .....	41
<b>Gambar 5. 2</b> Tampilan Proses Input Data Training.....	42
<b>Gambar 5. 3</b> Tampilan Proses Input Data testing .....	43
<b>Gambar 5. 4</b> Tampilan Hasil .....	43

## DAFTAR TABEL

<b>Tabel 2. 1</b> Penelitianan terkait.....	5
<b>Tabel 2. 2</b> Sample Data Training.....	13
<b>Tabel 2. 3</b> Sampel Data Testing .....	13
<b>Tabel 2. 4</b> Sampel Hasil Pengujian .....	14
<b>Tabel 2. 5</b> Hasil Pengujian .....	15
<b>Tabel 2. 6</b> Perangkat Lunak Pendukung.....	15
<b>Tabel 2. 7</b> Confusion Matrix .....	16
<b>Tabel 4. 1</b> Data Citra Buah Pepaya .....	23
<b>Tabel 4. 2</b> Contoh pixel 6x6 dari salah satu citra pepaya.....	25
<b>Tabel 4. 3</b> Nilai dari desimal di tempatkan pada pixel 6x6 yang baru .....	26
<b>Tabel 4. 4</b> Hasil perhitungan seluruh nilai pixel tengah.....	27
<b>Tabel 4. 5</b> Hasil Ekstraksi fitur dari 150 data trainning .....	27
<b>Tabel 4. 6</b> Hasil ekstraksi fitur salah satu citra data testing yang tidak diketahuikelasnya.....	28
<b>Tabel 4. 7</b> Perhitungan Jarak Euclidean Distance .....	29
<b>Tabel 4. 8</b> Hasil pengurutan nilai terkecil berdasarkan perhitungan jarak Euclidean distanc .....	30
<b>Tabel 4. 9</b> Jumlah k=1 .....	30
<b>Tabel 4. 10</b> Kelas data Uji dari jumlah k=1 .....	31
<b>Tabel 4. 11</b> Jumlah k=3 .....	31
<b>Tabel 4. 12</b> Kelas data uji dari jumlah k=3 .....	31
<b>Tabel 4. 13</b> Jumlah k=5 .....	31
<b>Tabel 4. 14</b> Kelas data uji dari jumlah k=5 .....	32
<b>Tabel 4. 15</b> Jumlah k=7 .....	32
<b>Tabel 4. 16</b> Kelas data uji dari jumlah k=7 .....	32
<b>Tabel 4. 17</b> Jumlah k=9 .....	33
<b>Tabel 4. 18</b> Kelas data uji dari jumlah k=9 .....	33
<b>Tabel 4. 19</b> Hasil Data Testing dari salah satu Nilai k (k=1) .....	34
<b>Tabel 4. 20</b> Hasil data uji k=1 .....	36
<b>Tabel 4. 21</b> Hasil data uji k=3 .....	37

<b>Tabel 4. 22</b> Hasil data uji k=5 .....	38
<b>Tabel 4. 23</b> Hasil data uji k=7 .....	39
<b>Tabel 4. 24</b> Hasil data uji k=9 .....	40
<b>Tabel 5. 1</b> Pembahasan Model .....	44

## **DAFTAR LAMPIRAN**

**Lampiran 1. 1 Kode Program**

**Lampiran 1. 2 Riwayat Hidup Peneliti**

**Lampiran 1. 3 Surat Rekomendasi Bebas Plagiasi**

**Lampiran 1. 4 Surat Keterangan Bebas Pustaka**

**Lampiran 1. 5 Surat Balasan Penelitian**

# **BAB I**

## **PENDAHULUAN**

### **1. 1. Latar Belakang**

Pepaya merupakan buah tropis komoditas hortikultura yang mempunyai nilai gizi dan ekonomi yang tinggi. Buah Pepaya memiliki kandungan vitamin C dan serat yang tinggi pula sehingga sangat baik dikonsumsi untuk melancarkan proses pencernaan tubuh[1]. Pepaya memiliki nama latin *Carica papaya*, termasuk dalam keluarga *Caricaceae*. Tanaman Pepaya merupakan tanaman buah yang berasal dari Amerika Tengah & Hindia Barat bahkan kawasan sekitar Meksiko & Costa Rica yang kemudian menyebar luas dan banyak ditanam diseluruh daerah tropis. Pepaya banyak tumbuh di daerah tropis maupun subtropis. Tanaman ini juga dapat ditanam di daerah- daerah basah atau kering serta di daerah-daerah dataran & pegunungan (sampai 1000 m dpl). Buah pepaya memiliki banyak jenis, diantaranya ada pepaya Bangkok, hawai dan jenis pepaya lokal seperti pepaya calina (*California*), carisya, cibinong yang merupakan pepaya asli dari Indonesia[2]. Pada penelitian ini jenis pepaya yang akan diidentifikasi adalah pepaya calina atau yang lebih dikenal dengan pepaya *California*, karena jenis pepaya ini paling banyak ditemukan di berbagai kota di Indonesia, termasuk kota Gorontalo.

Pepaya *California* (Calina) merupakan salah satu varietas unggul buah pepaya yang saat ini sedang digandrungi dan banyak dikebunkan oleh para petani karena sangat menjanjikan keuntungan. Pepaya *California* ini memiliki sifat dan keunggulan tersendiri yaitu buahnya tidak terlalu besar dengan bobot 0,8 – 1,5 kg/buah, rasanya manis, daging buah kenyal dan tebal. Proses panen buah papaya pada umumnya ditentukan oleh derajat warna kulit yang terlihat. Kematangan buah pepaya dapat dilihat dengan menyesuaikan warna buah yang telah matang dan belum matang. Kulit buah pepaya yang masih hijau tebal dan mulus serta memiliki tekstur yang agak keras menandakan buah pepaya yang belum matang. Pepaya mengkal di tandai dengan warna kulit yang masih hijau dan memiliki

guratan kuning pada kulit terutama dibagian ujung buah, sedangkan pepaya yang sudah matang ditandai dengan perubahan warna yang sudah mulai menguning tapi masih terlihat bercak hijau dengan tekstur buah yang sudah mulai lunak[3].

Mengidentifikasi atau mendeteksi kematangan buah pepaya masih dilakukan secara manual, Sayangnya masih terdapat kendala yang sering dihadapi dalam menentukan kematangan buah pepaya. Identifikasi secara manual sangat di pengaruhi oleh subjektifitas orang yang mengamati. Kelemahannya waktu yang dibutuhkan relatif lama jika identifikasi buah pepaya dilakukan dengan jumlah yang banyak. Selain itu subjektifitas lainnya tingkat kelelahan dan perbedaan persepsi tentang penilaian terhadap tingkat kematangan buah pepaya. Oleh karena itu diperlukan adanya metode yang memudahkan manusia khususnya masyarakat umum dalam mengidentifikasi tingkat kematangan buah pepaya secara otomatis.

Maka dari itu dibutuhkan sebuah aplikasi yang berbasis Teknologi Pengolahan Citra yang memungkinkan untuk memilah dan mengamati produk pertanian dan perkebunan secara otomatis. Seiring perkembangannya, metode-metode yang digunakan dalam teknologi pengolahan citra pun sudah cukup banyak. Salah satunya adalah metode LBP. Algoritma LBP (*local binary pattern*) merupakan metode transformasi sebuah citra menjadi sebuah susunan label integer yang menggambarkan kenampakan skala kecil dari suatu citra sebagai pertandingan nilai biner piksel pada pusat gambar dengan 8 nilai piksel disekelilingnya. Kelebihan metode ini yaitu mudah diimplementasikan dan tingkat komputerisasinya lebih rendah sehingga tidak membutuhkan waktu yang lama dalam ekstraksi fitur[4]. Pada penelitian ini untuk proses klasifikasi citra digunakan metode K-NN. K-NN (*K-Near Neighor*) merupakan sebuah metode klasifikasi terhadap objek berdasarkan data latih yang jaraknya paling dekat atau memiliki kemiripan ciri dengan objek tersebut. Metode K-NN mempunyai Kelebihan yaitu tahan terhadap data yang memiliki banyak noise dan lebih efektif dengan data latih yang besar dibandingkan dengan metode lain[5].

Beberapa hasil penelitian sebelumnya tentang identifikasi kematangan buah dengan menggunakan pengolahan citra seperti yang dilakukan oleh Sigit Sugiyanto dan Feri Wibowo tentang Klasifikasi tingkat kematangan buah pepaya

(*carica papaya l*) *California* (Calina IPB-9) dalam ruang warna HSV dan algoritma *K- Nearest Neighbors*, bersasarkan pengujian jumlah K tetangga 5 dan jumlah data citra uji sebanyak 12 data, didapatkan tingkat keakuratan algoritma KNN sebesar 83,34%[6]. Dan penelitian Novan Wijaya, Anugrah Ridwan tentang Klasifikasi Jenis Buah Apel Dengan Metode *K-Nearest Neighbors*, Data yang digunakan pada penelitian ini adalah 800 citra, yang terdiri dari 600 citra latih dan 200 citra uji, dan menghasilkan nilai akurasi sebesar 94%[5]. Adapun penelitian dari Diantarakita, Agus Wahyu Widodo dan Muh. Arif Rahman tentang Ekstraksi Ciri pada Klasifikasi Tipe Kulit Wajah Menggunakan Metode *Local Binary Pattern*, data yang digunakan pada penelitian ini sebanyak 112 citra wajah wanita yang diperoleh dengan mengambil data secara langsung di lapangan (data primer). Penelitian ini mendapatkan nilai akurasi yaitu sebesar 84,62%[7].

Adapun peneliti ingin mengajukan penelitian tentang “**Identifikasi Kematangan Buah Pepaya Menggunakan Metode LBP (Local Binary Pattern) dan K-NN (K-Nearest Neighbors)**” Diharapkan penelitian ini dapat memberikan kontribusi, berupa model LBP dan K-NN yang paling akurat untuk mendeteksi tingkat kematangan buah pepaya.

### **1. 2. Identifikasi Masalah**

Berdasarkan penjelasan diatas, maka identifikasi masalah dalam penelitian ini yaitu:

- 1) Terbatasnya fungsi identifikasi secara manual sehingga memberikan hasil yang tidak konsisten untuk mendeteksi tingkat kematangan buah pepaya
- 2) Perlunya metode komputerisasi untuk menentukan tingkat kematangan buah pepaya secara otomatis.

### **1. 3. Rumusan Masalah**

Berdasarkan identifikasi masalah yang telah diuraikan maka dapat ditentukan rumusan masalah dalam penelitian ini, yaitu:

Bagaimana kinerja dan efektifitas Sistem Identifikasi tingkat kematangan buah pepaya menggunakan metode LBP dan KNN?

#### **1. 4. Tujuan Penelitian**

Adapun yang menjadi tujuan pada penelitian ini adalah:

Untuk menetukan tingkat akurasi kematangan buah pepaya menggunakan metode LBP dan K-NN.

#### **1. 5. Manfaat Penelitian**

##### **1. 5. 1. Manfaat Teoritis**

Penelitian ini diharapkan dapat menjadi referensi pada penelitian yang berhubungan dengan mengidentifikasi kematangan buah pepaya dengan image processing menggunakan metode LBP dan K-NN.

##### **1. 5. 2. Manfaat Praktis**

Sebagai bahan kajian yang melibatkan suatu rancangan image processing untuk mengidentifikasi tingkat kematangan papaya.

## **BAB II**

### **LANDASAN TEORI**

#### **2. 1. Tinjauan Studi**

Penelitian tentang sistem mendeteksi kematangan buah dan metode yang digunakan telah dilakukan oleh beberapa peneliti sebelumnya. Berikut beberapa penelitian terkait:

**Tabel 2. 1 Penelitian terkait**

<b>NO</b>	<b>PENELITI</b>	<b>JUDUL/TAHUN</b>	<b>METODE</b>	<b>HASIL</b>
1	Sigit Sugiyanto dan Feri Wibowo[6]	Klasifikasi tingkat kematangan buah pepaya ( <i>carica papaya l</i> ) <i>California</i> (Calina IPB-9) dalam ruang warna HSV dan algoritma <i>K-Nears Neighbors/201 5</i>	Menggunakan metode <i>K-Nears Neighbor</i>	Bersasarkan Pengujian dengan jumlah K tetangga 3 dan jumlah data citra uji sebanyak 12 data, didapatkan tingkat keakuratan algoritma KNN sebesar 75%, dengan data output yang tidak sesuai dengan target sejumlah 3 data dan yang sesuai target sejumlah 9 data. pengujian jumlah K tetangga 5 dan jumlah data citra uji sebanyak 12 data, didapatkan tingkat keakuratan algoritma KNN sebesar 83,34%

2	Novan Wijaya dan Anugrah Ridwan[5].	Klasifikasi Jenis Buah Apel Dengan Metode K-Nearest Neighbor/2019	Menggunakan Metode Algoritma K-Nearest Neighbor	Berdasarkan penelitian yang telah dilakukan tentang klasifikasi buah apel, Data yang digunakan pada penelitian ini adalah 800 citra, yang terdiri dari 600 citra latih dan 200 citra uji, dan menghasilkan nilai akurasi sebesar 94%
3	Diantara kita, Agus Wahyu Widodo, dan Muh. Arif Rahman[7]	Ekstraksi Ciri pada Klasifikasi Tipe Kulit Wajah Menggunakan Metode Local Binary Pattern/2019	Menggunakan Metode <i>Local Binary Pattern</i>	Penelitian ini menggunakan data sebanyak 112 citra wajah wanita yang diperoleh dengan mengambil data secara langsung di lapangan (data primer). Penelitian ini mendapatkan hasil akurasi tertinggi yaitu sebesar 84,62% dengan jarak ketetanggaan ( $R$ ) = 1 dan kombinasi dari 3 fitur/ciri statistik tekstur yaitu kombinasi fitur/ciri mean, skewness dan energi.

## 2. 2. Tinjauan Pustaka

### 2. 2. 1. Buah Pepaya

Pepaya merupakan jenis tanaman perdu yang mempunyai tinggi 2-10 meter. Pepaya juga termasuk jenis tanaman tropis basah yang mampu tumbuh subur di daerah yang memiliki ketinggian 0 m - 1500 m di atas permukaan laut. Pepaya juga termasuk salah satu tanaman yang kaya akan manfaat, tidak hanya buah nya, hampir semua bagian papaya memiliki manfaat masing – masing. Mulai dari Akarnya yang dapat digunakan sebagai obat obatan, daunnya dapat dimanfaatkan untuk lalapan menambah nafsu makan, sumber vitamin A, dan bisa digunakan untuk mengobati penyakit seperti malaria, demam berdarah, kejang perut, dan sakit panas. batangnya juga bisa diambil untuk pakan ternak, bunganya dapat dimanfaatkan sebagai sayuran dan bunga hias, bijinya bermanfaat untuk mengurangi berat badan, obat cacing, dan mengeluarkan keringat bagi penderita masuk angin, getahnya bermanfaat untuk melunakkan daging, digunakan sebagai bahan baku industri farmasi, dan bahan kosmetik. Buah Pepaya bisa dinikmati dengan berbagai cara, dimakan langsung, dibuat Jus, dibuat untuk campuran saus tomat atau diolah bersama bahan lain[8].

### 2. 2. 2. Pepaya *California*

Pepaya *California* (*Calina*) merupakan salah satu varietas unggul buah pepaya yang saat ini sedang digandrungi dan banyak dikebunkan oleh para petani karena sangat menjanjikan keuntungan. Pepaya *California* ini memiliki sifat dan keunggulan tersendiri yaitu buahnya tidak terlalu besar dengan bobot 0,8 – 1,5 kg/buah, rasanya manis, daging buah kenyal dan tebal. berkulit hijau tebal dan mulus, berbentuk lonjong, buah matang berwarna kuning. Tanaman ini memang gampang tumbuh, dengan media tanah berhumus campur pasir, cukup sinar matahari dan drainase baik, tanaman pepaya dapat tumbuh subur. Pohonnya dapat berbuah hingga umur mencapai empat tahun. Dalam satu bulan bisa dipanen sampai empat kali. Sekali panen setiap pohon papaya *California* dapat menghasilkan 2 hingga 3 buah dengan sekali panen setiap minggu bisa mencapai berkisar 1,9 hingga 3,6 ton per hektar[3].



**Gambar 2. 1 Buah Pepaya *California***

### **2. 2. 3. Kriteria Kematangan Buah Pepaya *California***

Berikut ini kriteria tingkat kematangan buah pepaya *California* berdasarkan warna dan tekstur kulit buah[3]:

1. Pepaya Mentah, Kulit buah masih hijau penuh dan mulus serta memiliki tekstur yang agak keras.
2. Pepaya Mengkal, ditandai dengan perubahan warna yang mulai menguning kehijauan.
3. Pepaya Matang, ditandai dengan warna kuning kemerahan atau oranye, dengan tekstur buah yang sudah mulai lunak.

### **2. 2. 4. Citra (*Image*)**

Citra atau *image* adalah representasi, kemiripan, atau imitasi dari suatu objek. Citra merupakan komponen multimedia yang memegang peranan yang sangat penting sebagai bentuk informasi visual. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optic berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu penyimpanan[9]. Secara umum terdapat 3 jenis citra yaitu[9]:

### 1. Citra Biner

Citra biner adalah citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih (0 dan 1). Nilai 0 menyatakan warna hitam dan 1 menyatakan warna putih.

### 2. Citra Keabuan

Citra jenis ini menangani gradasi warna hitam dan putih, yang menghasilkan efek warna abu-abu. Citra skala keabuan memberi kemungkinan warna yang lebih banyak daripada citra biner, karena ada nilai-nilai lain di antara nilai minimum (hitam) dan nilai maksimumnya (putih).

### 3. Citra Berwarna

Citra Berwana merupakan citra yang menyajikan warna dalam bentuk RGB ( R = Merah ), ( G = Hijau ), ( B = Biru ). Setiap komponennya menggunakan 8 bit, nilainya berkisar dari ( 0 – 255 ). Dengan demikian warna yang dapat disajikan mencapai  $255 \times 255 \times 255$  warna.

## 2. 2. 5. Pengolahan Citra

Pengolahan citra (*image processing*) merupakan suatu sistem di mana proses dilakukan dengan masukan berupa citra (*image*) dan hasilnya juga berupa citra (*image*). Pengolahan citra digital mempelajari tentang teknik-teknik mengolah citra. Citra adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Digital mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Secara sistematis citra merupakan fungsi kontinyu (*continue*) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan computer digital, maka suatu citra harus dipresentasikan secara numeric dengan nilai-nilai diskrit. Representasi dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra[10].

## 2. 2. 6. Ekstraksi Fitur

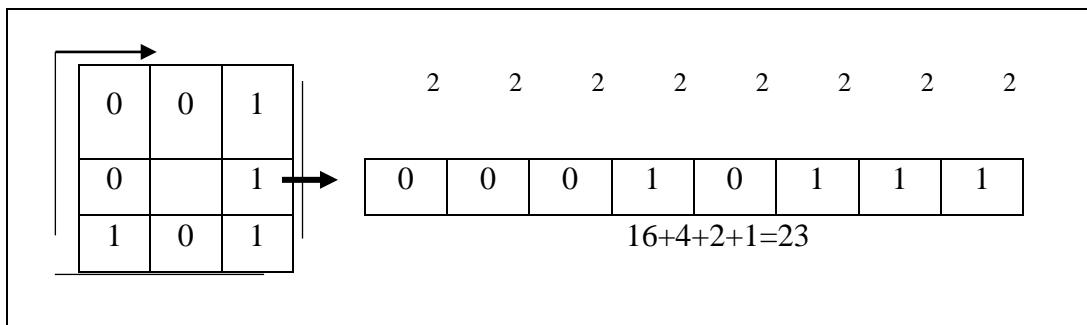
Ekstraksi fitur merupakan tahapan mengekstrak informasi yang terkandung dalam suatu objek citra digital yang nantinya nilai yang didapatkan akan dianalisis untuk proses selanjutnya. Fitur adalah karakteristik dari suatu objek yang akan berguna untuk proses klasifikasi. Ekstraksi Fitur dilakukan dengan cara menghitung jumlah titik atau pixel yang ada pada citra[11].

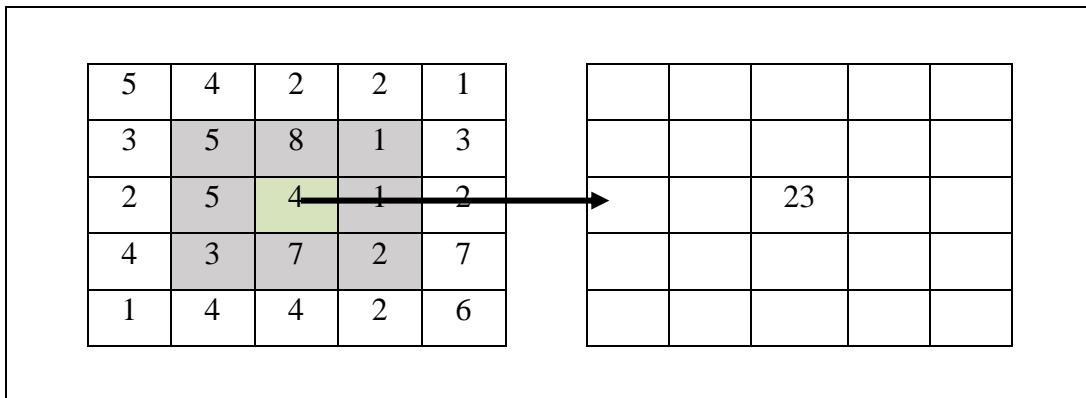
## 2. 2. 7. Metode *Local Binary Pattern* (LBP)

Metode LBP pertama kali diperkenalkan oleh Timo Ojala dan David Harwood pada tahun 1992. *Local Binary Pattern* merupakan sebuah metode yang digunakan untuk mendapatkan fitur tekstur sebuah citra. Proses dari metode ini bekerja dengan delapan piksel tetangga, menggunakan nilai piksel pusatnya sebagai threshold untuk dibandingkan dengan tetangga dari piksel tersebut[12]. LBP menggunakan 8 pixel pada sebuah blok pixel 3x3, formula dasar dari operator ini tidak menempatkan batasan pada ukuran dari ketetanggaan atau pada jumlah titik sampling (banyaknya pixel tetangga)[12].

<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>5</td><td>8</td><td>1</td></tr> <tr><td>5</td><td>4</td><td>1</td></tr> <tr><td>3</td><td>7</td><td>2</td></tr> </table>	5	8	1	5	4	1	3	7	2	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td></td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> </table>	0	0	1	0		1	1	0	1
5	8	1																	
5	4	1																	
3	7	2																	
0	0	1																	
0		1																	
1	0	1																	

Gambar 2. 2 Langkah Pertama LBP



**Gambar 2. 3 Langkah Kedua LBP****Gambar 2. 4 Langkah Terakhir LBP**

Perhitungan metode *Local Binary Pattern* dapat di selesaikan dengan persamaan sebagai berikut:

$$LBP_{P,R} = \sum_{p=1}^{P-1} s(g_p - g_c) 2^p$$

$$S(z) = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0 \end{cases}$$

Keterangan :

- P : Banyaknya pixel tetangga
- R : Nilai Jarak/Radius
- $g_c$  : Nilai dari pixel x dan y
- $g_p$  : Nilai pixel tetangga
- $x_c y_c$  : Koordinat Pusat

## 2. 2. 8. Metode K-NN (*K- Nearest Neighbor*)

Algoritma K-NN merupakan sebuah metode untuk melakukan klasifikasi obyek berdasarkan data pembelajaran yang jaraknya paling dekat dengan obyek tersebut. Prinsip kerja K-NN adalah mencari jarak terdekat antara data yang akan dievaluasi dengan  $k$  tetangga (*neighbour*) terdekatnya dalam data pelatihan[5]. Jika D adalah sekumpulan data pelatihan maka ketika data uji d disajikan,

algoritma akan menghitung jarak antara setiap data dalam D dengan data uji d, Kemudian k buah data dalam D yang memiliki jarak terdekat dengan d diambil. Himpunan k merupakan k-nearest neighbor. Selanjutnya kategori data uji d ditentukan berdasarkan label mayoritas kategori dalam himpunan k-tetangga terdekat[13].

Metode K-NN melakukan proses pencocokan/pengenalan berdasarkan jumlah tetangga terdekat untuk menentukan kelasnya. Adapun penerapan algoritma *K-Nearest Neighbor* terdiri langkah-langkah berikut[14]:

- 1) Menentukan parameter K (jumlah tetangga paling dekat)
- 2) Menghitung kuadrat jarak euclid (*query instance*) masing-masing obyek terhadap data sampel yang diberikan.

Rumus *Euclidean Distance*:

$$d_i = \sqrt{\sum_{i=1}^p (x_{2i} - x_{1i})^2}$$

Keterangan:

- d : Jarak *Euclidean*
- $x_1$  : Sampel data
- $x_2$  : Data Uji
- i : Variable
- p : Dimensi Data

- 3) Kemudian mengurutkan objek-objek tersebut kedalam kelompok yang mempunyai jarak *Euclidean* terkecil.
- 4) Mengumpulkan kategori Y (Klasifikasi *nearest neighbor*)
- 5) Dengan menggunakan kategori nearest neighbor yang paling mayoritas maka diprediksi nilai query instan yang telah dihitung.

#### 2.2.8.1. Contoh Penerapan Metode K-NN

Berikut merupakan contoh penerapan metode K-NN dari studi kasus Klasifikasi jenis rempah-rempah berdasarkan fitur warna RGB dan tekstur dengan

proses pengujian klasifikasi menggunakan K-NN. Metode pengukuran jarak yang digunakan adalah Euclidean Distance. Secara keseluruhan data training yang digunakan adalah 30 untuk masing-masing jenis rempah-rempah sehingga total keseluruhan berjumlah 120 data[15].

**Tabel 2. 2** Sample Data Training

Nama	Red	Green	Blue	Energy	Homogeneity
Jahe 1	243.947	238.967	231.500	0,75108	0,98817
Jahe 2	250.102	247.829	244.378	0,84991	0,98974
Kencur 1	251.516	249.977	248.058	0,93425	0,99570
Kencur 2	247.648	245.583	243.968	0,90663	0,99389
Kunyit 1	216.296	207.255	200.842	0,56372	0,98040
Kunyit 2	198.486	184.850	176.708	0,34896	0,95906
Lengkuas 1	230.962	221.110	213.744	0,66720	0,99041
Lengkuas 2	248.632	244.912	240.438	0,86982	0,99466

**Tabel 2. 3** Sampel Data Testing

Nama	Red	Green	Blue	Energy	Homogeneity
Jahe 1	240.915	235.502	228.249	0,74449	0,98763
Jahe 6	249.537	246.454	240.995	0,80756	0,98988
Jahe 8	250.110	247.861	244.917	0,89477	0,99255
Kencur 1	248.707	246.5	244.557	0,90695	0,9934
Kencur 5	251.905	251.059	250.285	0,96216	0,99771
Kencur 6	250.997	249.764	248.898	0,94807	0,99628
Kunyit 3	244.339	241.190	237.806	0,83032	0,99002
Kunyit 5	247.306	244.755	242.526	0,86992	0,99196
Kunyit 8	247.402	244.262	241.149	0,8629	0,99367
Lengkuas 1	229.378	219.924	211.976	0,64473	0,98368
Lengkuas 2	237.728	230.507	223.552	0,73537	0,98993
Lengkuas 3	247.495	244.549	241.371	0,87983	0,99575

Setelah diperoleh data yang digunakan untuk pengujian kemudian dilakukan proses klasifikasi menggunakan Algoritma *K-Nearest Neighbour* dengan langkah-langkah sebagai berikut[15]:

- 1) Menentukan parameter k (jumlah tetangga paling dekat). Parameter k (jumlah tetangga paling dekat untuk melakukan klasifikasi jenis rempah-rempah diatas adalah menggunakan nilai k=1, k=3, k=5.
- 2) Menghitung kuadrat jarak *euclidean* masing-masing objek terhadap data sample yang diberikan.

Sebagai contoh perhitungan diambil data testing yang pertama yaitu data testing Jahe1, lalu dikalikan terhadap seluruh data training yang ada[15]:

$$d = \sqrt{(243.947 - 240.915)^2 + (238.967 - 235.502)^2 + (231.500 - 228.249)^2 + (0,75108 - 0,74449)^2 + (0,98817 - 0,98763)^2} = 5636,33303$$

Jadi jarak *Euclidean Distance* yang diperoleh adalah sebesar 5636,33303. Data testing jahe1 tersebut dihitung *Euclidean Distance* nya keseluruhan data training yang ada, kemudian dirangking dengan mengambil 5 neighbor yang terdekat, hasil dari perkalian data testing jahe1 terhadap 120 data training dapat dilihat pada tabel berikut ini[15]:

**Tabel 2. 4** Sampel Hasil Pengujian

Ranking	<i>Euclidean Distance</i>	Nama Kelas
1	1784,19422	Jahe
2	1872,70873	Lengkuas
3	1938,29461	Lengkuas
4	2042,767731	Lengkuas
5	2174,98	Jahe

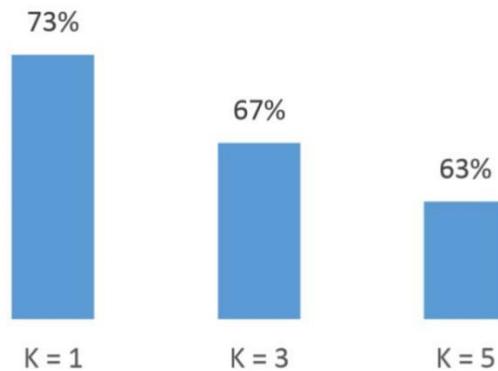
Dari hasil pengujian diatas dapat dilihat bahwa jika menggunakan nilai K=1 maka proses klasifikasi benar, jika menggunakan nilai K=3 maka proses

klasifikasi salah, dan jika menggunakan nilai K=5 maka proses klasifikasi juga salah. Pada penelitian ini dilakukan 30 kali pengujian dengan menggunakan data training masing-masing jenis rempah-rempah terhadap 120 data training, hasil dari keseluruhan pengujian dapat dilihat pada table 2.5 [15].

**Tabel 2. 5** Hasil Pengujian

Nilai K	Jumlah Kebenaran	Persentasi Kebenaran
1	23/30	73%
3	20/30	67%
5	19/30	63%

Untuk lebih jelasnya perbandingan hasil pengujian dapat dilihat pada diagram seperti yang ditunjukkan pada gambar 2.9 [15].



**Gambar 2. 5** Diagram Hasil Pengujian

## 2. 2. 9. Perangkat Lunak Pendukung

**Tabel 2. 6** Perangkat Lunak Pendukung

No	Perangkat Lunak Pendukung	Kegunaan
1.	Mendeley Dekstop	Untuk membuat daftar pustaka secara otomatis.
2.	Phyton	Bahasa pemograman untuk membuat sebuah program

## 2. 2.10. Evaluasi Model

Evaluasi Model merupakan kegiatan pengumpulan data yang berkaitan dengan objek yang dievaluasi sebagai bahan bagi pengambilan keputusan dalam menentukan tindak lanjut suatu program. Evaluasi model dalam penelitian ini dilakukan untuk menguji performansi dari sistem yang dibangun. Beberapa cara untuk melakukan evaluasi adalah dengan menghitung akurasi (*accuracy*), presisi (*precision*), dan *recall* dari hasil analisis sistem. Metode yang digunakan pada penelitian ini adalah *Confusion Matrix*.

### 2. 2.10.1 Confusion matrix

*Confusion matrix* merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu proses klasifikasi. Pada dasarnya *confusion matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya[16]. Terdapat empat istilah sebagai representasi hasil proses klasifikasi sebagai berikut (M.Fadly Rahman)[17].

**Tabel 2. 7 Confusion Matrix**

Kelas	Klasifikasi Positif	Klasifikasi Negatif
Positif	TP ( <i>True Positive</i> )	FN ( <i>False Negative</i> )
Negatif	FP ( <i>False Positive</i> )	TN ( <i>True Negative</i> )

Keterangan:

- 1) *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar oleh sistem,
- 2) *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif oleh sistem,
- 3) *True Positive* (TP) merupakan data positif yang terdeteksi benar oleh sistem.
- 4) *False Negative* (FN) merupakan data negatif yang terdeteksi salah oleh sistem.

Berikut persamaan model *confusion matrix*[17]:

1. Nilai Akurasi (*accuracy*) menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Nilai akurasi merupakan perbandingan antara data yang terkласifikasi benar dengan keseluruhan data. Persamaan Akurasi sebagai berikut.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

2. Nilai Presisi (*precision*) menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Persamaan Presisi sebagai berikut.

$$\text{Precision} = \frac{TP}{FP + TP} * 100\%$$

3. Nilai Sensitivity (*recall*) menunjukkan berapa persen data kategori positif yang terkласifikasi dengan benar oleh system. Persamaan Recall sebagai berikut.

$$\text{Recall} = \frac{TP}{FN + TP} * 100\%$$

## 2. 2. 11. Pengujian Sistem

Pengujian sistem adalah sebuah pengujian yang diperlukan untuk memastikan sebuah sistem yang sudah atau sedang dibuat dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Pengujian itu sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak yang ada dan merupakan suatu bagian yang tidak bisa terpisah dari siklus hidup pengembangan software seperti halnya, desain analisis, dan pengkodean. Pengembang atau penguji sistem harus menyiapkan sesi khusus untuk menguji program atau sistem yang sudah dibuat agar kesalahan maupun kekurangan bisa dideteksi sejak awal dan bisa dikoreksi secepatnya[18].

### 2.2.11.1 Black Box

Black-Box Testing merupakan pengujian yang sangat berfokus pada spesifikasinya dan fungsional dari perangkat lunak, tester dapat mendefinisikan suatu kumpulan dan kondisi input dan dapat melakukan pengetesan pada spesifikasi fungsional program[19].

Black Box Testing cenderung untuk menemukan hal-hal berikut:

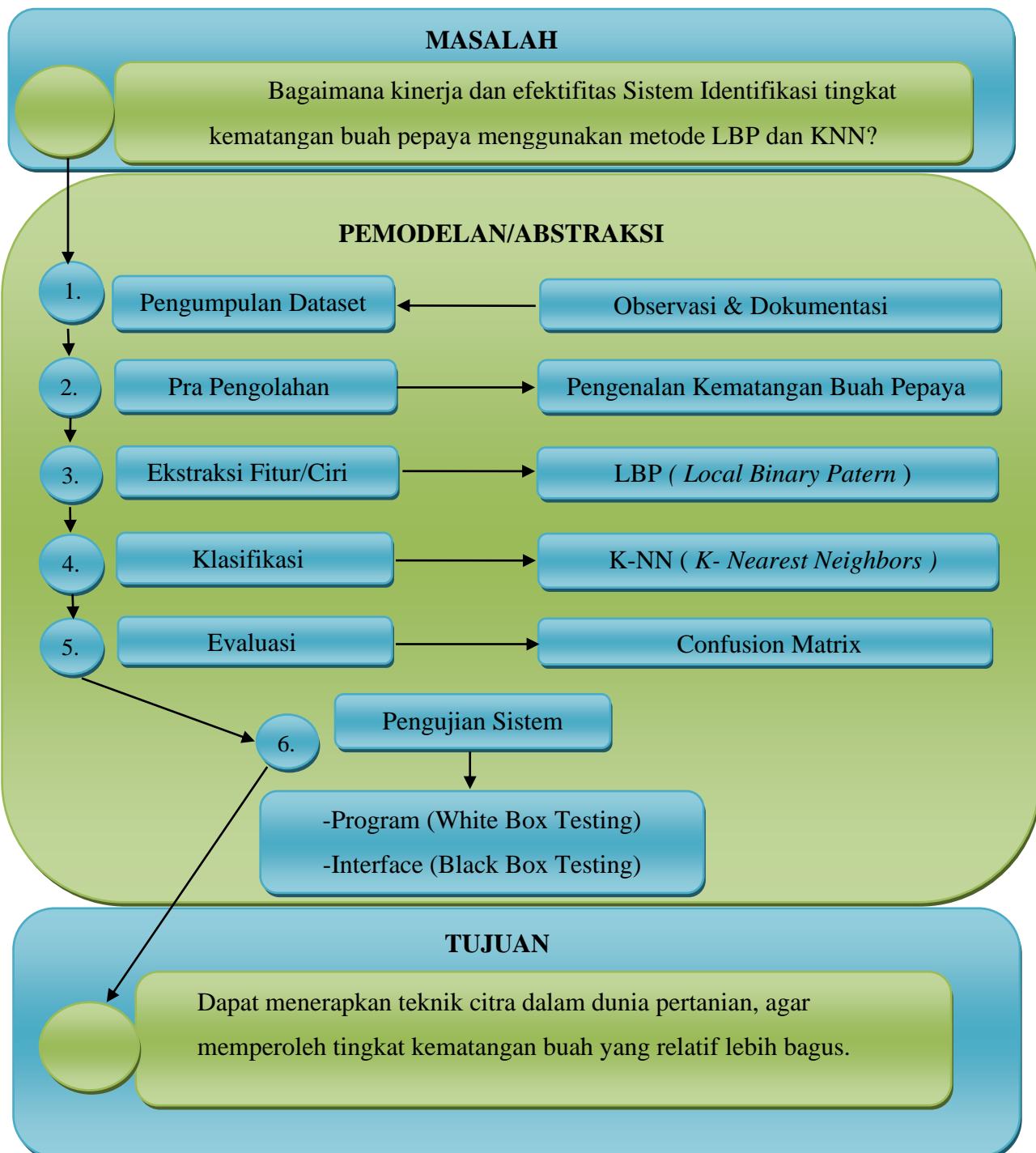
- 1) Kesalahan inisialisasi dan terminasi.
- 2) Kesalahan antarmuka (interface errors).
- 3) Kesalahan pada struktur data dan akses basis data.
- 4) Kesalahan performansi (performance errors).
- 5) Fungsi yang tidak benar atau tidak ada.

### 2.2.11.2 White Box

Pengujian white box adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan white box testing merupakan petunjuk untuk mendapatkan program yang benar secara 100%[18].

*White Box Testing* juga adalah salah satu cara untuk menguji suatu aplikasi atau *software* dengan cara melihat sebuah modul untuk dapat meneliti dan menganalisa kode program yang di buat jika mempunyai kesalahan atau tidak. Kalau modul yang telah dan sudah di hasilkan berupa output yang tidak sesuai dengan yang di harapkan maka akan dikompilasi ulang dan di cek kembali kode-kode tersebut hingga sesuai dengan yang bisa diharapkan[19].

## 2. 3. Kerangka Piki



**Gambar 2. 6** Kerangka Pikir

## **BAB III**

### **METODE PENELITIAN**

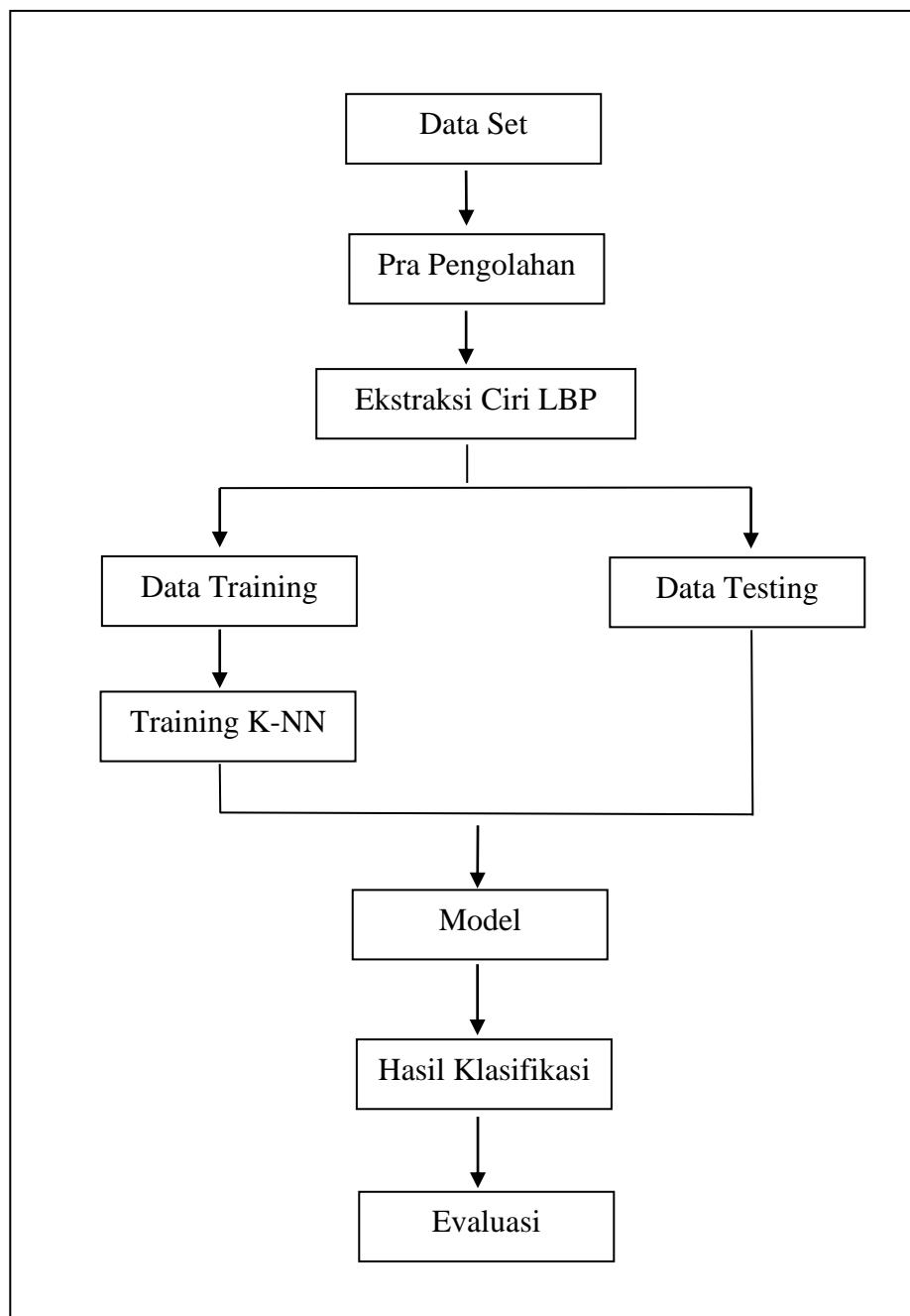
#### **3. 1. Jenis, Metode, Subjek, Objek, Waktu dan Lokasi Penelitian**

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan. Dipandang dari jenis informasi yang diolah, maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data, maka penelitian ini merupakan penelitian konfirmatori. Penelitian ini menggunakan metode penelitian *eksperimental*. Dengan demikian ini adalah penelitian *eksperimental*. Subjek penelitian ini adalah Identifikasi pada objek Buah Pepaya.

#### **3. 2. Pengumpulan Data**

Data primer penelitian ini adalah citra Buah Pepaya *California*, dimana peneliti mengambil gambar secara langsung menggunakan kamera Smartphone Vivo Z1 Pro dengan resolusi 16 MP. Jumlah data yang digunakan yaitu 195 file dalam format JPG dengan tingkat kematangan (Mentah, Mengkal, Matang).

### 3.3. Pemodelan



**Gambar 3.1** Pemodelan

### **3. 3. 1. Pra Pengolahan**

Sebelum data diolah, terlebih dahulu dilakukan *histogram equalization*. Hal ini dilakukan karena *Histogram equalization* adalah proses yang mengubah distribusi nilai derajat keabuan pada sebuah citra sehingga menjadi seragam (uniform).

### **3. 3. 2. Ekstraksi Ciri**

Ekstrasi ciri berfungsi sebagai pendekripsi ciri dari suatu citra. Ciri yang dapat digunakan untuk membedakan objek satu dengan objek lainnya, diantaranya adalah ciri bentuk, ukuran, ciri geometri, ciri tekstur, dan warna. Pada penelitian ini menggunakan ekstrasi ciri tekstur *Local Binary Pattern* (LBP).

### **3. 3. 3. Data Training**

Data latih digunakan untuk menguji data baru dari hasil ekstraksi fitur *Local Binary Pattern* (LBP) terhadap tingkat kematangan buah papaya yang belum pernah ada, yang kemudian diproses oleh algoritma klasifikasi *K-Nearest Neighbor* (K-NN).

### **3. 3. 4. Data Testing**

Data uji bertujuan untuk mengetahui tingkat kematangan buah papaya baru yang tidak dikenali sebelumnya. Selain itu data uji juga digunakan untuk mengukur sejauh mana algoritma K-NN berhasil melakukan klasifikasi.

### **3. 3. 5. Hasil Klasifikasi**

Hasil klasifikasi merupakan hasil output yang didapatkan pada data uji dari proses klasifikasi yang menggunakan *K-Nearest Neighbor* (K-NN) berdasarkan model yang di dapatkan dari data latih. Atau hasil yang didapatkan dari proses klasifikasi yang dilakukan model.

### **3. 3. 6. Evaluasi Model**

Model yang telah di hasilkan kemudian dievaluasi dengan menggunakan *confusion matrix*. Untuk mengetahui tingkat akurasinya.

## BAB IV

### HASIL PENELITIAN

#### 4.1 Hasil Pengumpulan Data

Dataset diambil dengan menggunakan kamera smartphone Vivo Z1 Pro dengan resolusi kamera 16 MP. Data set yang digunakan pada penelitian ini adalah data real yang didapatkan langsung dari lapangan. Data set ini terdiri dari 195 data dan terbagi menjadi 150 data training dan 45 data testing.

**Tabel 4. 1** Data Citra Buah Pepaya

Pepaya Mentah	Pepaya Mengkal	Pepaya Matang

## 4.2 Hasil Pemodelan

### 4.2.1 Pra-Pengolahan

Pra-pengolahan citra (Image processing) yaitu proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Prosedur ini bertujuan untuk memperbaiki data citra yang mengalami distorsi atau kesalahan kedalam bentuk aslinya. Pra Pengolahan dilakukan dengan beberapa tahapan yaitu :

1. Pengubahan citra warna (RGB) ke citra keabuan (Grayscale)

Pra-Pengolahan proses awal yang mengubah citra training atau testing yang awalnya citra dari RGB menjadi citra grayscale, perubahan ini dilakukan karena citra grayscale memiliki persamaan yang sederhana dan mampu mengurangi kebutuhan memory dimana nilai warna putih diwakili dengan angka 255 dan nilai warna hitam diwakili dengan angka 0.

2. Segmentasi Citra

Segmentasi citra mempunyai arti membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan yang tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel-piksel tetangganya. Tahapan ini bertujuan untuk mempartisi citra menjadi bagian-bagian pokok yang mengandung informasi penting, misalnya memisahkan objek dan latar belakang. Hasil dari proses segementasi citra akan digunakan untuk proses tingkal tinggi lebih lanjut yang dapat dilakukan terhadap suatu citra, misalnya proses klasifikasi citra dan proses identifikasi objek.

### 4.2.2 Ekstraksi Citra

Proses ekstraksi fitur ini dilakukan untuk mengetahui setiap cirri dari sebuah tekstur dari kulit buah pepaya yang kemudian akan dirubah kedalam kode biner agar mudah dikenali oleh system dengan menggunakan salah satu metode ekstraksi fitur yaitu metode LBP dan dari hasil ekstraksi fitur kemudian akan menghasilkan sekumpulan data dari setiap tekstur kulit buah pepaya. Berikut contoh perhitungan matrix 3x3 dengan ukuran pixel 6x6.

**Tabel 4. 2** Contoh pixel 6x6 dari salah satu citra pepaya

x,y	0	1	2	3	4	5
0	187	188	189	189	191	191
1	186	188	188	189	189	191
2	186	187	188	188	189	190
3	185	187	187	188	189	190
4	184	185	186	187	188	189
5	184	185	186	187	188	189

Berikut Langkah-langkah melakukan perhitungan manual metode LBP:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P=1} s(g_p - g_c)2^p$$

1) Langkah Pertama

Ambil matrix 3x3 untuk melakukan perhitungan dengan mengambil nilai tengah untuk menjadi nilai threshold dengan cara membandingkan setiap pixel yang berada disekelilingnya jika nilai threshold lebih besar dengan nilai tetangganya maka diberi nilai 0 jika lebih kecil atau sama dengan nilai threshold maka diberi nilai 1. Seperti yang terlihat pada gambar berikut:

x,y	0	1	2	3	4	5
0	187	188	189	189	191	191
1	186	188	188	189	189	191
2	186	187	188	188	189	190
3	185	187	187	188	189	190
4	184	185	186	187	188	189
5	184	185	186	187	188	189

187	188	189
186	188	188
186	187	188

0	1	1
0		1
0	0	1

**Gambar 4. 1** Matrix 3x3 yang diambil pada pixel 6x6

## 2) Langkah kedua

Jika sudah didapatkan nilai binernya kemudian nilai biner tersebut disusun searah jarum jam dan dihitung berapa jumlah desimalnya seperti yang terlihat pada gambar berikut:

0	1	1
0		1
0	0	1

2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
1	1	1	1	0	0	0	0

$128+64+32+16=240$

**Gambar 4. 2** Perhitungan nilai decimal dari matrix 3x3

## 3) Langkah Ketiga

Ketika nilai decimal yang sudah didapatkan dari hasil perhitungan biner tersebut, kemudian ditempatkan pada pixel yang baru untuk menggantikan nilai pixel sebelumnya yang dijadikan sebagai nilai tengah atau thereshold seperti yang terlihat pada gambar berikut :

**Tabel 4. 3** Nilai dari desimal ditempatkan pada pixel 6x6 yang baru

x,y	0	1	2	3	4	5
0	187	188	189	189	191	191
1	186	240	188	189	189	191
2	186	187	188	188	189	190
3	185	187	187	188	189	190
4	184	185	186	187	188	189
5	184	185	186	187	188	189

Lakukan perhitungan yang sama sampai semua nilai yang berada ditengah pixel berubah seperti pada tabel berikut ini :

**Tabel 4. 4** Hasil perhitungan seluruh nilai pixel tengah

x,y	0	1	2	3	4	5
0	187	188	189	189	191	191
1	186	240	251	241	251	191
2	186	248	241	251	249	190
3	185	224	243	241	240	190
4	184	249	249	249	249	189
5	184	185	186	187	188	189

#### 4.2.3 Klasifikasi

Pada tahapan ini hasil dari ekstraksi fitur data testing dan data uji pada pembahasan sebelumnya diklasifikasikan berdasarkan kelasnya masing-masing dengan jumlah  $k=1$ ,  $k=3$ ,  $k=5$ ,  $k=7$ , dan  $k=9$ , nilai dari jumlah  $k$  tersebut diambil berdasarkan nilai angka ganjil agar dapat menghindari hasil klasifikasi dengan jumlah data yang seimbang. Nilai  $k$  tersebut juga dapat menentukan bagaimana kinerja algoritma K-NN, dalam melakukan identifikasi kematangan buah pepaya, seperti yang terlihat pada tabel dibawah ini:

**Tabel 4. 5** Hasil Ekstraksi fitur dari 150 data training

Data	X1	X2	X3	X4	...	X8	X9	X10	Y
0	0.204	0.197	0.065	0.061	...	0.051	0.058	0.167	Matang
1	0.180	0.156	0.081	0.061	...	0.067	0.056	0.232	Matang
2	0.132	0.138	0.057	0.055	...	0.052	0.077	0.320	Matang
...	...	...	...	...	...	...	...	...	
50	0.163	0.154	0.058	0.058	...	0.067	0.060	0.226	Mengkal
51	0.204	0.189	0.073	0.073	...	0.035	0.043	0.238	Mengkal
52	0.182	0.159	0.076	0.085	...	0.032	0.058	0.276	Mengkal
...	...	...	...	...	...	...	...	...	
100	0.237	0.205	0.025	0.018	...	0.035	0.059	0.368	Mentah
101	0.221	0.249	0.047	0.038	...	0.032	0.045	0.322	Mentah

...	...	...	...	...	...	...	...	...	...
149	0.147	0.187	0.077	0.052	...	0.054	0.054	0.276	Mentah

**Tabel 4. 6** Hasil ekstraksi fitur salah satu citra data testing yang tidak diketahui kelasnya

Data	X1	X2	X3	X4	...	X8	X9	X10	Y
1	0.150	0.116	0.032	0.027	...	0.062	0.101	0.431	?

Untuk mengetahui kelas dari data testing pada tabel 4.6 diatas maka peneliti akan melakukan perhitungan manual dengan rumus perhitungan jarak *Euclidean Distance* seperti dibawah ini:

$$d = \sqrt{\sum_{i=1}^P (x_{2i} - x_{1i})^2}$$

Penyelesaian:

$$d = \sqrt{(0,204 - 0,150)^2 + (0,197 - 0,116)^2 + (0,065 - 0,032)^2 + (0,061 - 0,027)^2 + (0,074 - 0,020)^2 + (0,054 - 0,020)^2 + (0,063 - 0,035)^2 + (0,051 - 0,062)^2 + (0,058 - 0,101)^2 + (0,167 - 0,431)^2}$$

$$d = \sqrt{(0,002916) + (0,006561) + (0,001089) + (0,001156) + (0,002916) + (0,001156) + (0,000784) + (0,000121) + (0,001849) + (0,069696)}$$

$$d = \sqrt{0,088244}$$

$$= 0,2970589167$$

Kemudian dari proses perhitungan jarak yang telah dilakukan maka nilai yang didapatkan dari perhitungan jarak antara data training dengan data testing tersebut akan disusun berdasarkan nilai pada masing-masing kelas seperti pada tabel 4.7 sebagai berikut:

**Tabel 4. 7** Perhitungan Jarak Euclidean Distance

Data	Perhitungan Jarak Euclidean Distance	Nilai	Kelas
0	$d = \sqrt{(0,204 - 0,150)^2 + (0,197 - 0,116)^2 + (0,065 - 0,032)^2 + (0,061 - 0,027)^2 + (0,074 - 0,020)^2 + (0,054 - 0,020)^2 + (0,063 - 0,035)^2 + (0,051 - 0,062)^2 + (0,058 - 0,101)^2 + (0,167 - 0,431)^2}$	0,2970589167	Matang
1	$d = \sqrt{(0,180 - 0,150)^2 + (0,156 - 0,116)^2 + (0,081 - 0,032)^2 + (0,061 - 0,027)^2 + (0,050 - 0,020)^2 + (0,054 - 0,020)^2 + (0,057 - 0,035)^2 + (0,067 - 0,062)^2 + (0,056 - 0,101)^2 + (0,232 - 0,431)^2}$	0,2241606567	Matang
2	$d = \sqrt{(0,132 - 0,150)^2 + (0,138 - 0,116)^2 + (0,057 - 0,032)^2 + (0,055 - 0,027)^2 + (0,047 - 0,020)^2 + (0,054 - 0,020)^2 + (0,061 - 0,035)^2 + (0,052 - 0,062)^2 + (0,077 - 0,101)^2 + (0,320 - 0,431)^2}$	0,1333229163	Matang
...	...	...	...
50	$d = \sqrt{(0,163 - 0,150)^2 + (0,154 - 0,116)^2 + (0,058 - 0,032)^2 + (0,058 - 0,027)^2 + (0,065 - 0,020)^2 + (0,085 - 0,020)^2 + (0,058 - 0,035)^2 + (0,067 - 0,062)^2 + (0,060 - 0,101)^2 + (0,226 - 0,431)^2}$	0,2318620279	Mengkal
51	$d = \sqrt{(0,204 - 0,150)^2 + (0,189 - 0,116)^2 + (0,073 - 0,032)^2 + (0,073 - 0,027)^2 + (0,053 - 0,020)^2 + (0,043 - 0,020)^2 + (0,043 - 0,035)^2 + (0,035 - 0,062)^2 + (0,043 - 0,101)^2 + (0,238 - 0,431)^2}$	0,2346614583	Mengkal
52	$d = \sqrt{(0,182 - 0,150)^2 + (0,159 - 0,116)^2 + (0,076 - 0,032)^2 + (0,085 - 0,027)^2 + (0,056 - 0,020)^2 + (0,043 - 0,020)^2 + (0,027 - 0,035)^2 + (0,032 - 0,062)^2 + (0,058 - 0,101)^2 + (0,276 - 0,431)^2}$	0,1919279695	Mengkal
...	...	...	...
100	$d = \sqrt{(0,237 - 0,150)^2 + (0,205 - 0,116)^2 + (0,025 - 0,032)^2 + (0,018 - 0,027)^2 + (0,022 - 0,020)^2 + (0,018 - 0,020)^2 + (0,010 - 0,035)^2 + (0,035 - 0,062)^2 + (0,059 - 0,101)^2 + (0,368 - 0,431)^2}$	0,1507149628	Mentah
101	$d = \sqrt{(0,221 - 0,150)^2 + (0,249 - 0,116)^2 + (0,047 - 0,032)^2 + (0,038 - 0,027)^2 + (0,015 - 0,020)^2 + (0,013 - 0,020)^2 + (0,012 - 0,035)^2 + (0,032 - 0,062)^2 + (0,045 - 0,101)^2 + (0,322 - 0,431)^2}$	0,1926447508	Mentah
...	...	...	...
149	$d = \sqrt{(0,147 - 0,150)^2 + (0,187 - 0,116)^2 + (0,077 - 0,032)^2 + (0,052 - 0,027)^2 + (0,053 - 0,020)^2 + (0,047 - 0,020)^2 + (0,048 - 0,035)^2 + (0,054 - 0,062)^2 + (0,054 - 0,101)^2 + (0,276 - 0,431)^2}$	0,189697127	Mentah

Setelah nilai jarak dari masing-masing data latih ke data uji didapatkan maka akan disusun berdasarkan nilai terkecil berikut.

**Tabel 4. 8** Hasil pengurutan nilai terkecil berdasarkan perhitungan jarak *Euclidean distance*

Data Ke	Jarak Ke Masing-Masing Data Latih	Kelas
77	0,0626099034	Mengkal
19	0,0998999499	Matang
2	0,1333229163	Matang
100	0,1507149628	Mentah
47	0,1521019395	Matang
95	0,1590471628	Mengkal
61	0,1605988792	Mengkal
149	0,1896971127	Mentah
52	0,1919279695	Mengkal
101	0,1926447508	Mentah
...	...	...

Dari hasil pengurutan yang terdapat pada tabel 4.8, maka yang akan diambil adalah nilai berdasarkan jumlah k yang telah ditentukan sebelumnya:

**Tabel 4. 9** Jumlah k=1

Data Ke	Perhitungan Jarak Euclidean Distance	Kelas
77	0,0626099034	Mengkal

Berdasarkan tabel 4.9, dari data uji jumlah k=1, dapat dilihat bahwa kelas yang muncul adalah “Mengkal” dengan 1 kelas maengkal seperti tabel dibawah ini :

**Tabel 4. 10** Kelas data Uji dari jumlah k=1

No	Kelas	Jumlah Data
1	Matang	0
2	Mengkal	1
3	Mentah	0

**Tabel 4. 11** Jumlah k=3

Data Ke	Perhitungan Jarak Euclidean Distance	Kelas
77	0,0626099034	Mengkal
19	0,0998999499	Matang
2	0,1333229163	Matang

Berdasarkan tabel 4.11, dari data uji jumlah k=3, dapat dilihat bahwa kelas yang paling banyak muncul adalah “Matang”, dengan 2 kelas matang dan 1 kelas mengkal seperti tabel dibawah ini :

**Tabel 4. 12** Kelas data uji dari jumlah k=3

No	Kelas	Jumlah Data
1	Matang	2
2	Mengkal	1
3	Mentah	0

**Tabel 4. 13** Jumlah k=5

Data Ke	Perhitungan Jarak Euclidean Distance	Kelas
77	0,0626099034	Mengkal
19	0,0998999499	Matang
2	0,1333229163	Matang
100	0,1507149628	Mentah
47	0,1521019395	Matang

Berdasarkan tabel 4.13, dari data uji jumlah k=5, dapat dilihat bahwa kelas yang paling banyak muncul adalah “Matang“, dengan 3 kelas matang, 1 kelas mengkal dan 1 kelas mentah seperti tabel dibawah ini :

**Tabel 4. 14** Kelas data uji dari jumlah k=5

No	Kelas	Jumlah Data
1	Matang	3
2	Mengkal	1
3	Mentah	1

**Tabel 4. 15** Jumlah k=7

Data Ke	Perhitungan Jarak Euclidean Distance	Kelas
77	0,0626099034	Mengkal
19	0,0998999499	Matang
2	0,1333229163	Matang
100	0,1507149628	Mentah
47	0,1521019395	Matang
95	0,1590471628	Mengkal
61	0,1605988792	Mengkal

Berdasarkan tabel 4.15, dari data uji jumlah k=7, dapat dilihat bahwa kelas yang paling banyak muncul adalah “Matang dan Mengkal“, dengan 3 kelas matang, 3 kelas mengkal dan 1 kelas mentah seperti tabel dibawah ini:

**Tabel 4. 16** Kelas data uji dari jumlah k=7

No	Kelas	Jumlah Data
1	Matang	3
2	Mengkal	3
3	Mentah	1

**Tabel 4. 17** Jumlah k=9

Data Ke	Perhitungan Jarak Euclidean Distance	Kelas
77	0,0626099034	Mengkal
19	0,0998999499	Matang
2	0,1333229163	Matang
100	0,1507149628	Mentah
47	0,1521019395	Matang
95	0,1590471628	Mengkal
61	0,1605988792	Mengkal
149	0,1896971127	Mentah
52	0,1919279695	Mengkal

Berdasarkan tabel 4.17, dari data uji jumlah k=9, dapat dilihat bahwa kelas yang paling banyak muncul adalah “Mengkal”, dengan 4 kelas mengkal, 3 kelas matang dan 2 kelas mentah seperti tabel dibawah ini:

**Tabel 4. 18** Kelas data uji dari jumlah k=9

No	Kelas	Jumlah Data
1	Matang	3
2	Mengkal	4
3	Mentah	2

Dari penentuan jumlah k=1, k=3, k=5, k=7 dan k=9, maka dapat dinyatakan bahwa kelas yang paling mayoritas/mirip dengan data uji adalah “Matang”.

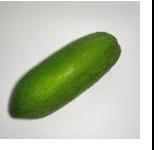
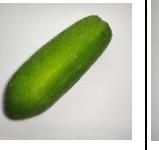
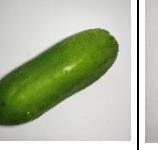
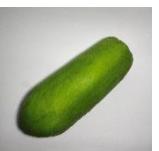
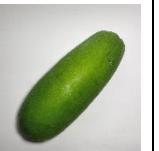
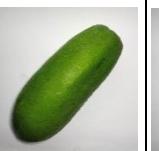
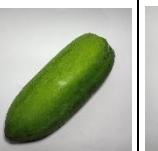
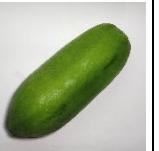
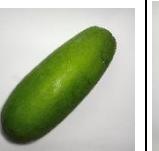
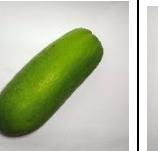
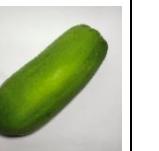
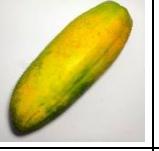
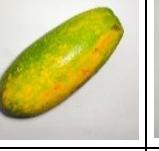
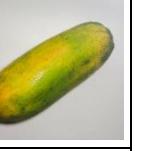
#### 4.2.4 Evaluasi Model

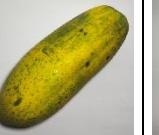
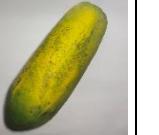
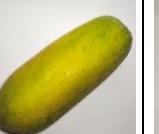
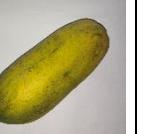
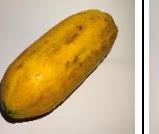
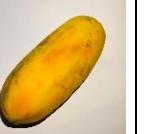
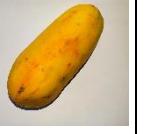
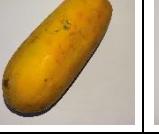
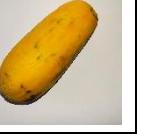
##### 4.2.4.1 Confusion Matrix

Pada penelitian ini peneliti menggunakan *confusion matrix*s sebagai metode untuk mengukur hasil akurasi dari metode klasifikasi yang telah diujikan

sebelumnya. *Confusion matrix* dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah classifier tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari True-positive dan True-negative memberikan informasi ketika classifier dalam melakukan klasifikasi data bernilai benar, sedangkan False-Posotive dan False-negative memberikan informasi ketika classifier salah dalam melakukan klasifikasi data. Berikut merupakan sebuah tabel dari hasil perhitungan 30 data uji dengan jumlah  $k=1$ ,  $k=3$ ,  $k=5$ ,  $k=7$  dan  $k=9$  sebagai berikut:

**Tabel 4. 19** Hasil Data Testing dari salah satu Nilai k ( $k=1$ )

Mentah					
Klasifikasi	Mentah	Mentah	Mentah	Mentah	Mentah
Hipotesis	Mentah	Mengkal	Mentah	Mentah	Mentah
Mentah					
Klasifikasi	Mentah	Mentah	Mentah	Mentah	Mentah
Hipotesis	Mentah	Mentah	Mentah	Mentah	Mengkal
Mentah					
Klasifikasi	Mentah	Mentah	Mentah	Mentah	Mentah
Hipotesis	Mengkal	Mentah	Mentah	Mentah	Mentah
Mengkal					
Klasifikasi	Mengkal	Mengkal	Mengkal	Mengkal	Mengkal
Hipotesis	Mentah	Mengkal	Mentah	Matang	Mengkal

Mengkal					
Klasifikasi	Mengkal	Mengkal	Mengkal	Mengkal	Mengkal
Hipotesis	Mengkal	Mengkal	Mengkal	Mengkal	Mengkal
Mengkal					
Klasifikasi	Mengkal	Mengkal	Mengkal	Mengkal	Mengkal
Hipotesis	Mengkal	Mengkal	Matang	Mengkal	Mengkal
Matang					
Klasifikasi	Matang	Matang	Matang	Matang	Matang
Hipotesis	Matang	Matang	Matang	Matang	Matang
Matang					
Klasifikasi	Matang	Matang	Matang	Matang	Matang
Hipotesis	Mengkal	Matang	Matang	Matang	Mengkal
Matang					
Klasifikasi	Matang	Matang	Matang	Matang	Matang
Hipotesis	Mengkal	Matang	Matang	Mengkal	Matang

 Data uji yang benar

 Data uji yang salah

**Tabel 4. 20** Hasil data uji k=1

Kelas	Matang	Mengkal	Mentah
Matang	11	4	0
Mengkal	2	11	2
Mentah	0	3	12

Keterangan:

Berdasarkan hasil hipotesis dari 45 data testing citra pepaya diatas diketahui bahwa kategori pepaya matang yang di deteksi matang (benar) oleh sistem adalah 11 citra, pepaya matang yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 4 citra, pepaya matang yang dideteksi mentah (salah/tidak sesuai) oleh sistem 0 citra atau tidak ada. Berikut kategori pepaya mengkal yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 2 citra, pepaya mengkal yang dideteksi mengkal (benar) oleh sistem adalah 11 citra, pepaya mengkal yang dideteksi mentah (salah/tidak sesuai) adalah 2 citra. Selanjutnya pepaya mentah yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 0 citra atau tidak ada, pepaya mentah yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 3 citra, pepaya mentah yang dideteksi mentah (benar) oleh sistem adalah 12 citra. Dari keseluruhan hasil diatas di hasilkan : Jumlah data yang diklasifikasikan Benar adalah 34 citra dan Jumlah data yang diklasifikasikan Salah adalah 11 citra.

Berikut perhitungsn nilai akurasinya :

$$Akurasi = \frac{\text{Total Data Uji yang Benar}}{\text{Total Keseluruhan Data Uji}} \times 100\%$$

$$Akurasi = \frac{34}{45} \times 100\% = 75\%$$

Jadi, hasil akurasi data uji K-1 adalah 75%.

**Tabel 4. 21** Hasil data uji k=3

Kelas	Matang	Mengkal	Mentah
Matang	9	6	0
Mengkal	1	12	2
Mentah	1	2	12

Keterangan:

Berdasarkan hasil hipotesis dari 45 data testing citra pepaya diatas diketahui bahwa kategori pepaya matang yang di deteksi matang (benar) oleh sistem adalah 9 citra, pepaya matang yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 6 citra, pepaya matang yang dideteksi mentah (salah/tidak sesuai) oleh sistem 0 citra atau tidak ada. Berikut kategori pepaya mengkal yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 1 citra, pepaya mengkal yang dideteksi mengkal (benar) oleh sistem adalah 12 citra, pepaya mengkal yang dideteksi mentah (salah/tidak sesuai) adalah 2 citra. Selanjutnya pepaya mentah yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 1 citra atau tidak ada, pepaya mentah yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 2 citra, pepaya mentah yang dideteksi mentah (benar) oleh sistem adalah 12 citra. Dari keseluruhan hasil diatas di hasilkan : Jumlah data yang diklasifikasikan Benar adalah 33 citra dan Jumlah data yang diklasifikasikan Salah adalah 12 citra.

Berikut perhitungsn nilai akurasinya :

$$Akurasi = \frac{\text{Total Data Uji yang Benar}}{\text{Total Keseluruhan Data Uji}} \times 100\%$$

$$Akurasi = \frac{33}{45} \times 100\% = 73\%$$

Jadi, hasil akurasi data uji K-3 adalah 73%.

**Tabel 4. 22** Hasil data uji k=5

Kelas	Matang	Mengkal	Mentah
Matang	9	6	0
Mengkal	1	12	2
Mentah	1	2	12

Keterangan:

Berdasarkan hasil hipotesis dari 45 data testing citra pepaya diatas diketahui bahwa kategori pepaya matang yang di deteksi matang (benar) oleh sistem adalah 9 citra, pepaya matang yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 6 citra, pepaya matang yang dideteksi mentah (salah/tidak sesuai) oleh sistem 0 citra atau tidak ada. Berikut kategori pepaya mengkal yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 1 citra, pepaya mengkal yang dideteksi mengkal (benar) oleh sistem adalah 12 citra, pepaya mengkal yang dideteksi mentah (salah/tidak sesuai) adalah 2 citra. Selanjutnya pepaya mentah yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 1 citra atau tidak ada, pepaya mentah yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 2 citra, pepaya mentah yang dideteksi mentah (benar) oleh sistem adalah 12 citra. Dari keseluruhan hasil diatas di hasilkan : Jumlah data yang diklasifikasikan Benar adalah 33 citra dan Jumlah data yang diklasifikasikan Salah adalah 12 citra.

Berikut perhitungsn nilai akurasinya :

$$Akurasi = \frac{\text{Total Data Uji yang Benar}}{\text{Total Keseluruhan Data Uji}} \times 100\%$$

$$Akurasi = \frac{33}{45} \times 100\% = 73\%$$

Jadi, hasil akurasi data uji K-5 adalah 73%.

**Tabel 4. 23** Hasil data uji k=7

Kelas	Matang	Mengkal	Mentah
Matang	7	8	0
Mengkal	2	11	2
Mentah	1	1	13

Keterangan:

Berdasarkan hasil hipotesis dari 45 data testing citra pepaya diatas diketahui bahwa kategori pepaya matang yang di deteksi matang (benar) oleh sistem adalah 7 citra, pepaya matang yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 8 citra, pepaya matang yang dideteksi mentah (salah/tidak sesuai) oleh sistem 0 citra atau tidak ada. Berikut kategori pepaya mengkal yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 2 citra, pepaya mengkal yang dideteksi mengkal (benar) oleh sistem adalah 11 citra, pepaya mengkal yang dideteksi mentah (salah/tidak sesuai) adalah 2 citra. Selanjutnya pepaya mentah yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 1 citra atau tidak ada, pepaya mentah yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 1 citra, pepaya mentah yang dideteksi mentah (benar) oleh sistem adalah 13 citra. Dari keseluruhan hasil diatas di hasilkan : Jumlah data yang diklasifikasikan Benar adalah 31 citra dan Jumlah data yang diklasifikasikan Salah adalah 14 citra.

Berikut perhitungsn nilai akurasinya :

$$Akurasi = \frac{\text{Total Data Uji yang Benar}}{\text{Total Keseluruhan Data Uji}} \times 100\%$$

$$Akurasi = \frac{31}{45} \times 100\% = 68\%$$

Jadi, hasil akurasi data uji K-7 adalah 68%.

**Tabel 4. 24** Hasil data uji k=9

Kelas	Matang	Mengkal	Mentah
Matang	7	8	0
Mengkal	2	11	2
Mentah	0	2	13

Keterangan:

Berdasarkan hasil hipotesis dari 45 data testing citra pepaya diatas diketahui bahwa kategori pepaya matang yang di deteksi matang (benar) oleh sistem adalah 7 citra, pepaya matang yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 8 citra, pepaya matang yang dideteksi mentah (salah/tidak sesuai) oleh sistem 0 citra atau tidak ada. Berikut kategori pepaya mengkal yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 2 citra, pepaya mengkal yang dideteksi mengkal (benar) oleh sistem adalah 11 citra, pepaya mengkal yang dideteksi mentah (salah/tidak sesuai) adalah 2 citra. Selanjutnya pepaya mentah yang dideteksi matang (salah/tidak sesuai) oleh sistem adalah 0 citra atau tidak ada, pepaya mentah yang di deteksi mengkal (salah/tidak sesuai) oleh sistem adalah 2 citra, pepaya mentah yang dideteksi mentah (benar) oleh sistem adalah 13 citra. Dari keseluruhan hasil diatas di hasilkan : Jumlah data yang diklasifikasikan Benar adalah 31 citra dan Jumlah data yang diklasifikasikan Salah adalah 14 citra.

Berikut perhitungsn nilai akurasinya :

$$Akurasi = \frac{\text{Total Data Uji yang Benar}}{\text{Total Keseluruhan Data Uji}} \times 100\%$$

$$Akurasi = \frac{31}{45} \times 100\% = 68\%$$

Jadi, hasil akurasi data uji K-9 adalah 68%.

## BAB V

### PEMBAHASAN

#### 5.1 Pembahasan Sistem

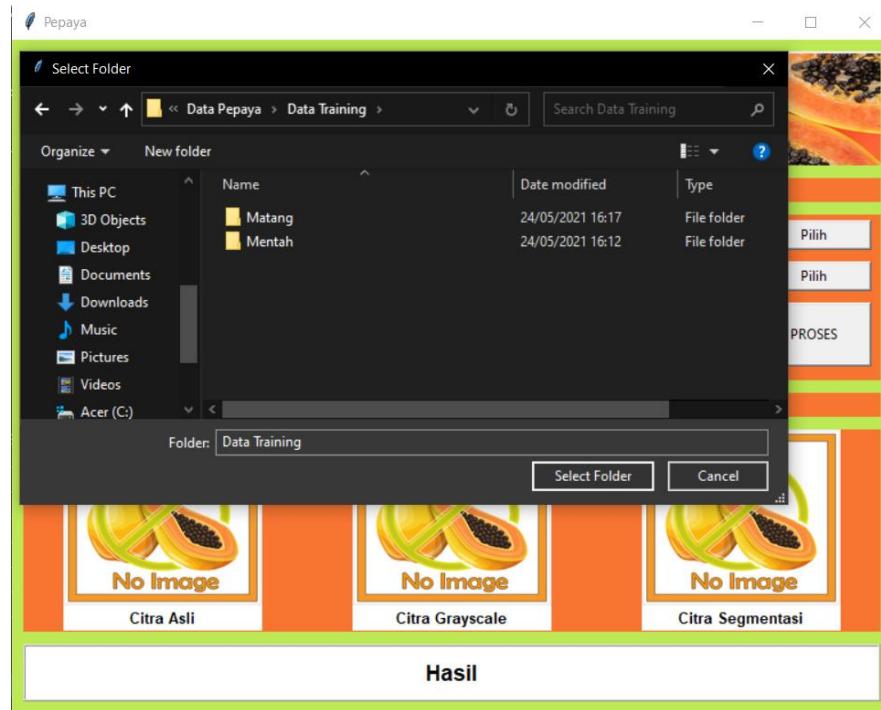
##### 1. Tampilan Menu Utama

Form menu utama merupakan form utama yang muncul saat program dijalankan. Pada form menu utama menampilkan menu – menu yang ada pada sistem yaitu menu input training, menu input testing, dan proses.



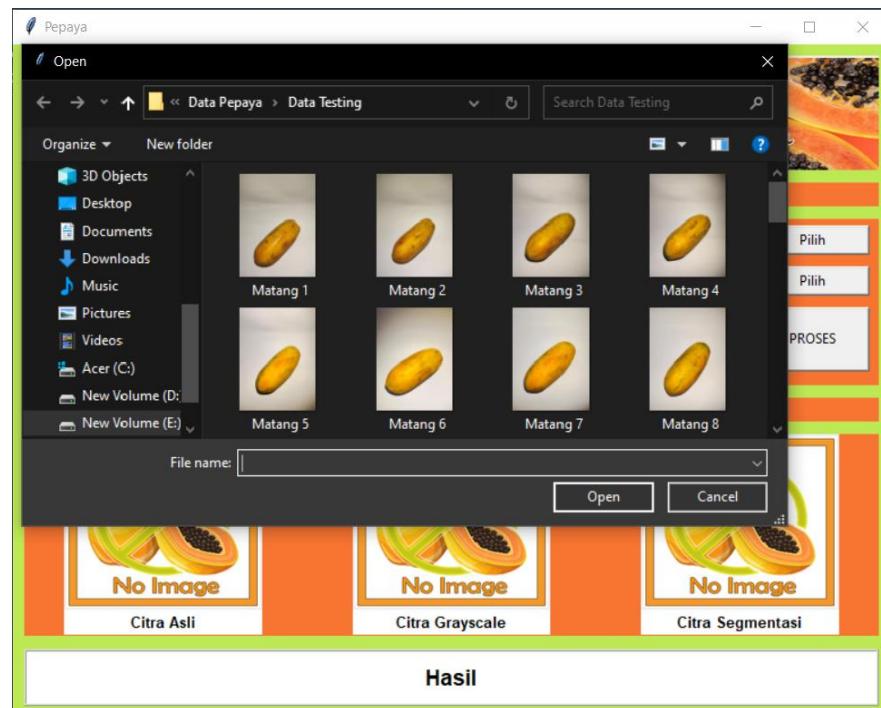
**Gambar 5. 1** Tampilan Menu Utama

## 2. Tampilan Proses Input Data Training



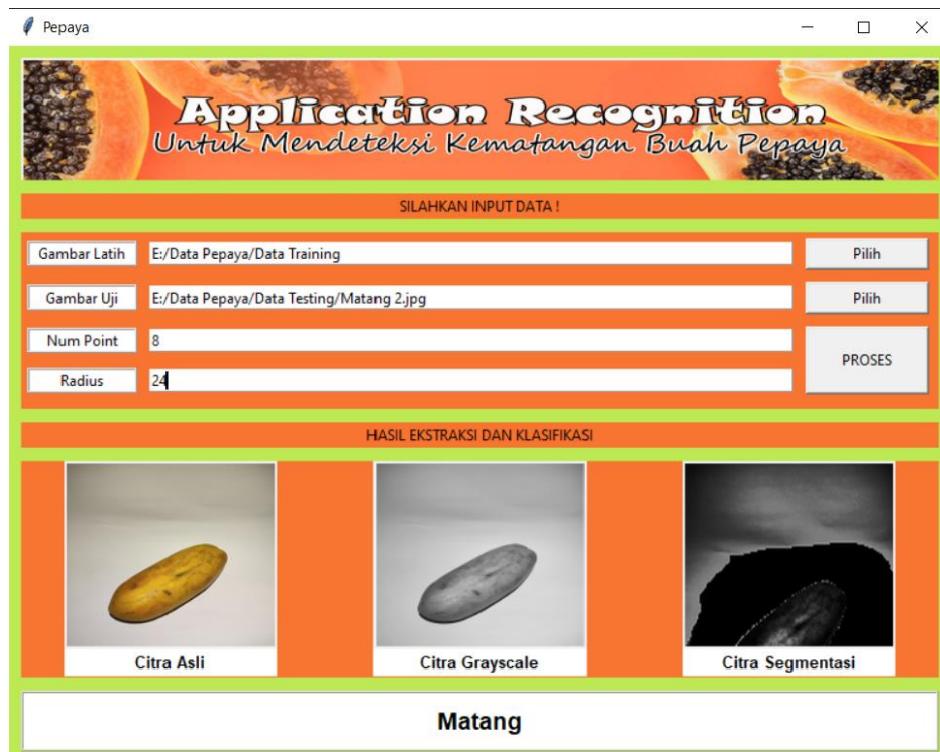
**Gambar 5. 2** Tampilan Proses Input Data Training

## 3. Tampilan Proses Input Data testing



**Gambar 5. 3 Tampilan Proses Input Data testing**

#### 4. Tampilan Hasil



**Gambar 5. 4 Tampilan Hasil**

## 5.2 Pembahasan Model

Data yang digunakan dalam penelitian ini terdiri dari data real peneliti yang di dapat dengan mengambil gambar secara langsung dengan menggunakan kamera smarthphone VIVO Z1 Pro dengan resolusi kamera 16 MP. Data set ini terdiri dari 195 data dan terbagi menjadi 150 data training dan 45 data testing.

Berdasarkan hasil penelitian data yang di ambil terdiri dari 3 kategori yaitu Matang, Mengkal dan Mentah. Selanjutnya citra buah Pepaya di ekstrasi ciri menggunakan *Local Binary Pattern*. Setelah di dapatkan ciri dari citra buah pepaya selanjutnya dilakukan proses klasifikasi menggunakan *K-Nearest Neighbor*. Pada tahap ini citra pada buah pepaya dijadikan sebagai inputan untuk mencari nilai akurasi terbaik pada percobaan model K-NN.

**Tabel 5. 1** Pembahasan Model

Nilai K	Hasil Akurasi	Pembahasan
K=1	75%	Penelitian ini dilakukan dengan menggunakan data citra papaya sebanyak 195 citra, dengan data testing 45 dan data training 150 dengan masing – masing 50 citra mentah, 50 citra mengkal dan 50 citra matang. Pada percobaan pertama model K-NN dengan jumlah nilai k=1 mendapatkan nilai akurasi tertinggi yaitu 75%. Selanjutnya percobaan model K-NN dengan jumlah nilai k=3 dan k=5 mendapatkan nilai akurasi yang sama yakni 73%. Sedangkan pada percobaan model K-NN dengan jumlah nilai k=7 dan k=9 hanya mendapatkan nilai akurasi 68%.
K=3	73%	
K=5	73%	
K=7	68%	
K=9	68%	

Pada tahap Confusion Matrix terdapat kendala pengenalan tingkat kematangan, beberapa data keluar dengan hasil yang tidak sesuai. Penambahan jumlah nilai K sangat berpengaruh untuk akurasi tingkat kematangan buah pepaya. Dilihat dari perhitungan hasil akurasi di atas bahwa, semakin banyak nilai K semakin rendah nilai akurasi yang di peroleh. Dari hasil perhitungan akurasi untuk pengujian Klasifikasi mendapatkan nilai akurasi tertinggi 75% pada percobaan model K-NN K-1. Pada penelitian ini penggunaan metode K-NN cukup baik untuk tahap klasifikasi.

## **BAB VI**

### **PENUTUP**

#### **6.1 KESIMPULAN**

Berdasarkan hasil penelitian yang telah dilakukan dan pembahasan yang telah diuraikan sebelumnya maka dapat ditarik kesimpulan bahwa:

Hasil penerapan ekstraksi fitur LBP (*Local Binary Pattern*) dan klasifikasi K-NN (*K-Nearest Neighbor*) yang diimplementasikan pada proses identifikasi kematangan buah pepaya dengan diukur menggunakan *confusion matrix* menghasilkan nilai akurasi sebesar 75% pada percobaan model K-NN K-1. Penambahan jumlah nilai K sangat berpengaruh untuk akurasi tingkat kematangan buah pepaya. Dilihat dari perhitungan hasil akurasi di atas bahwa, semakin banyak jumlah nilai K semakin rendah nilai akurasi yang di peroleh.

#### **6.2 SARAN**

1. Penulis berharap pada penelitian selanjutnya agar memperoleh hasil akurasi yang lebih baik.
2. Penulis berharap penelitian selanjutnya dapat menggunakan sample data yang lebih banyak agar hasil akurasi lebih tinggi.

## DAFTAR PUSTAKA

- [1] E. Syaefulloh, H. Purwadaria, and S. Suroso, “Identifikasi Tingkat Ketuaan Dan Kematangan Pepaya (Carica papaya L.) IPB 1 Dengan Pengolahan Citra Digital Dan Jaringan Syaraf Tiruan,” *Agritech J. Fak. Teknol. Pertan. UGM*, vol. 27, no. 2, pp. 75–81, 2007, doi: 10.22146/agritech.9496.
- [2] Maghfirah, T. . Johan, and Z. Yunizar, “Identifikasi Tahap Kematangan Buah Pepaya Berdasarkan Warna Menggunakan Metode Fuzzy,” *TIKA*, vol. 4, no. 1, pp. 28–37, 2019.
- [3] T. Mirani, E. Sinduningrum, and A. R. Dzikrillah, “Rancang Bangun Sistem Pengenalan Citra untuk Tingkat Kematangan Buah Pepaya California Berdasarkan Warna Berbasis Android,” *Semin. Nas. TEKNOKA*, vol. 3, no. 3, pp. 21–27, 2018.
- [4] K. Mujib, A. Hidayatno, and T. Prakoso, “Pengenalan Wajah Menggunakan Local Binary Pattern (Lbp) Dan Support Vector Machine (Svm),” *Transient*, vol. 7, no. 1, 2018, doi: 10.14710/transient.7.1.123-130.
- [5] N. Wijaya and A. Ridwan, “Klasifikasi Jenis Buah Apel Dengan Metode K-nearest Neighbors,” *Sisfokom*, vol. 08, no. 1, pp. 74–78, 2019.
- [6] S. Sugiyanto and F. Wibowo, “Klasifikasi Tingkat Kematangan Buah Pepaya (Carica Papaya L) California (Callina-IPB 9) Dalam Ruang Warna HSV Dan Algoritma K-Nears Neighbors,” *Pros. Senat.*, vol. 1, no. November, pp. 335–341, 2015.
- [7] D. Kita, A. W. Widodo, and M. A. Rahman, “Ekstraksi Ciri pada Klasifikasi Tipe Kulit Wajah Menggunakan Metode Local Binary Pattern,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 8, pp. 7938–7945, 2019.
- [8] D. Sartika Siagian, S. Herlina, W. Margi Sidoretno, F. Kedokteran dan Ilmu Kesehatan, and U. Abdurrah, “Kandungan Vitamin a Pada Buah Pepaya Hijau: Solusi Meningkatkan Produksi Asi,” *PSNKH*, pp. 129–134, 2019.
- [9] C. Iswahyudi, “Prototype Aplikasi Untuk Mengukur Kematangan Buah

- Apel,” *J. Teknol.*, vol. 3, no. 2, pp. 107–112, 2010.
- [10] R. Kusumanto and A. N. Tompunu, “Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB,” *Semin. Nas. Teknol. Inf. Komun. Terap.*, vol. 17, no. C, pp. 329–332, 2011, doi: 10.1016/S0166-1116(08)71924-1.
- [11] R. Munarto, E. Permata, and R. Salsabilla, “Klasifikasi Kualitas Biji Jagung Manis Berdasarkan Fitur Warna Menggunakan Fuzzy Logic,” *Simp. Nas. RAPI XIII - 2014 FT UMS*, pp. 5–12, 2014.
- [12] R. Purwati, G. Ariyanto, S. V. Machine, H. Equalization, and S. V. Machine, “Pengenalan Wajah Manusia Berbasis Algoritma Local Binary Pattern,” *J. Emit.*, vol. 17, no. 02, pp. 29–38, 2018.
- [13] R. N. Whidhiasih, N. A. Wahyuni, and Supriyanto, “Klasifikasi Buah Belimbing Berdasarkan Citra Red-Green-Blue Menggunakan Knn Dan Lda,” *PIKSEL (Penelitian Ilmu Komput. Sist. Embed. dan Logic)*, vol. 1, no. 1, pp. 29–35, 2013.
- [14] Arwansyah and J. S. Arie, “Implementasi Algoritma KNN Dalam Memprediksi Curah Hujan dan Temperatur Untuk Tanaman Padi,” *Prospektus Semin. Ilm. Sist. Inf. DAN Teknol. Inf.*, vol. 8, no. 1, pp. 11–20, 2019.
- [15] Kaharuddin, Kusrini, and E. T. Luthfi, “Klasifikasi Jenis Rempah-Rempah Berdasarkan Fitur Warna RGB Dan Tekstur Menggunakan Algoritma K-Nearst Neighbor,” *J. Inf. Interaktif*, vol. 4, no. 1, pp. 17–22, 2019.
- [16] P. Mayadewi and E. Rosely, “Prediksi Nilai Proyek Akhir Mahasiswa Menggunakan Algoritma Klasifikasi Data Mining,” *Semin. Nas. Sist. Inf. Indones.*, pp. 330–334, 2015.
- [17] M. F. Rahman, D. Alamsah, M. I. Darmawidjadja, and I. Nurma, “Klasifikasi Untuk Diagnosa Diabetes Menggunakan Metode Bayesian Regularization Neural Network (RBNN),” *J. Inform.*, vol. 11, no. 1, pp. 36–45, 2017, doi: 10.26555/jifo.v11i1.a5452.
- [18] T. Hidayat and M. Muttaqin, “Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis,” *J. Tek.*

- Inform. UNIS JUTIS*, vol. 6, no. 1, pp. 25–29, 2018, [Online]. Available: [www.ccsenet.org/cis](http://www.ccsenet.org/cis).
- [19] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, “Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analisis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN),” *JITTER J. Ilm. Teknol. Inf. Terap.*, vol. I, no. 3, pp. 31–36, 2015.

## **LAMPIRAN**

### **Lampiran 1. 1 Kode Program**

#### **1. Pembuatan Form**

```
import tkinter as tk
import tkinter.filedialog as fdialog
import tkinter.ttk as ttk
from PIL import Image, ImageTk
import LBPKN as lt1

class formpepaya(tk.Frame):
    def __init__(self, parent):
        self.parent = parent
        self.frame()
        self.widget()
        self.hasil()

    def frame(self):
        #frame judul
        self.frame1 = tk.Frame(self.parent, width=750, height=100, borderwidth=0,
        background="#F87431", relief='raised')
        self.frame1.grid(row=0, column=0, padx=10, pady=10, columnspan=2,
        sticky="NSEW")
        self.frame1.grid_propagate(0)

        #frame text
        self.frame2 = tk.Frame(self.parent, width=750, height=30, borderwidth=0,
        background="#F87431", relief='raised')
```

```
    self.frame2.grid(row=1,  column=0,  padx=10,  pady=1,  columnspan=2,
sticky="NSEW")
    self.frame2.grid_propagate(0)
#frame penginputan
    self.frame3 = tk.Frame(self.parent, width=750, height=145, borderwidth=0,
background="#F87431", relief='raised')
    self.frame3.grid(row=2,  column=0,  padx=10,  pady=10,  columnspan=2,
sticky="NSEW")
    self.frame3.grid_propagate(0)
#frame text
    self.frame4 = tk.Frame(self.parent, width=750, height=30, borderwidth=0,
background="#F87431", relief='raised')
    self.frame4.grid(row=3,  column=0,  padx=10,  pady=1,  columnspan=2,
sticky="NSEW")
    self.frame4.grid_propagate(0)
#frame hasil
    self.frame5 = tk.Frame(self.parent, width=750, height=177, borderwidth=0,
background="#F87431", relief='raised')
    self.frame5.grid(row=4,  column=0,  padx=10,  pady=10,  columnspan=2,
sticky="NSEW")
    self.frame5.grid_propagate(0)
#frame text hasil
    self.frame6 = tk.Frame(self.parent, width=750, height=30, borderwidth=0,
background="#F87431", relief='raised')
    self.frame6.grid(row=5,  column=0,  padx=10,  pady=1,  columnspan=2,
sticky="NSEW")
    self.frame6.grid_propagate(0)

def widget(self):
    #label silahkan input
```

```

    self.labeltext1 = tk.Label(self.frame2, text="SILAHKAN INPUT DATA !",
background="#F87431", anchor="center")
        self.labeltext1.grid()
        self.labeltext1.pack()

#input data
#folder training
    self.labeldatatrain = tk.Label(self.frame3, text='Gambar Latih', relief='ridge',
width=12, bg="white")
        self.labeldatatrain.grid(row=0, column=0, padx=5, pady=5, sticky="W")

#input
    value_datatraining = tk.StringVar(self.frame3, value="")
    self.inputdatatrain = tk.Entry(self.frame3, textvariable=value_datatraining,
relief='ridge', width=87)
        self.inputdatatrain.grid(row=0, column=1, padx=5, pady=5, sticky="W")

#button
    self.buttondatatrain = tk.Button(self.frame3, text='Pilih', width=13,
command=lambda:value_datatraining.set(fdialog.askdirectory()))
        self.buttondatatrain.grid(row=0, column=2, padx=5, pady=5, sticky="W")

#label
    self.labelfdatatest = tk.Label(self.frame3, text='Gambar Uji', relief='ridge',
width=12, bg="white")
        self.labelfdatatest.grid(row=1, column=0, padx=5, pady=5, sticky="W")

#input
    value_datatesting = tk.StringVar(self.frame3, value="")
    self.inputdatabtest = tk.Entry(self.frame3, textvariable=value_datatesting,
relief='ridge', width=87)
        self.inputdatabtest.grid(row=1, column=1, padx=5, pady=5, sticky="W")

#button

```

```

    self.buttondatatesting = tk.Button(self.frame3, text='Pilih', width=13,
command=lambda:value_datatesting.set(fdialog.askopenfilename()))
    self.buttondatatesting.grid(row=1, column=2, padx=5, pady=5, sticky="W")

#label
    self.labelnumpoint = tk.Label(self.frame3, text='Num Point', relief='ridge',
width=12, bg="white")
    self.labelnumpoint.grid(row=2, column=0,padx=5, pady=5, sticky="W")

#combobox
    value_textboxnumpoint = tk.StringVar(self.frame3, value="")
    self.textboxnumpoint = tk.Entry(self.frame3,
textvariable=value_textboxnumpoint, relief='ridge', width=87)
    self.textboxnumpoint.grid(row=2, column=1, columnspan=2, padx=5,
pady=7, sticky="W")

#label
    self.labelradius = tk.Label(self.frame3, text='Radius', relief='ridge',
width=12, bg="white")
    self.labelradius.grid(row=3, column=0, padx=5, pady=5, sticky="W")

#combobox2
    value_textboxradius = tk.StringVar(self.frame3, value="")
    self.textboxradius = tk.Entry(self.frame3, textvariable=value_textboxradius,
relief='ridge', width=87)
    self.textboxradius.grid(row=3, column=1, columnspan=2, padx=5, pady=7,
sticky="W")

#buttonproses
    self.buttonproses = tk.Button(self.frame3, text='PROSES', width=13,
command=self.proses)

```

```

    self.buttonproses.grid(row=2, column=2, columnspan=3, padx=5, pady=5,
    rowspan=2, sticky="NSEW")
#label hasil ekstraksi
    self.labeltext2 = tk.Label(self.frame4, text="HASIL EKSTRAKSI DAN
KLASIFIKASI", background="#F87431", anchor="center")
    self.labeltext2.grid()
    self.labeltext2.pack()

#label hasil
    self.labeltext3 = tk.Label(self.frame6, text="Hasil", background="white",
relief="ridge", height=2, width=62, font="arial 14 bold", anchor="center")
    self.labeltext3.grid()
    self.labeltext3.pack()

#panel gambar
def hasil(self):

    img = Image.open("noimagepapaya.jpg")
    img = img.resize((170, 150), Image.ANTIALIAS)
    img = ImageTk.PhotoImage(img)

    self.panel1 = tk.Label(self.frame5, image=img)
    self.panel1.image_names = img
    self.panel1.grid(row=1, padx=35, column=0, rowspan=5, sticky="NSEW")
    self.labelpanel1 = tk.Label(self.frame5, text='Citra Asli', font="arial 10
bold", bg="white")
    self.labelpanel1.grid(row=9,      padx=35,      rowspan=2,      column=0,
sticky="NSEW")

    self.panel2 = tk.Label(self.frame5, image=img)
    self.panel2.image_names = img
    self.panel2.grid(row=1, padx=44, column=1, rowspan=5, sticky="NSEW")

```

```

self.labelpanel2 = tk.Label(self.frame5, text='Citra Grayscale', font="arial 10
bold", bg="white")
    self.labelpanel2.grid(row=9,      padx=44,           rowspan=2,      column=1,
sticky="NSEW")

self.panel3 = tk.Label(self.frame5, image=img)
self.panel3.image_names = img
self.panel3.grid(row=1, padx=35, column=2, rowspan=5, sticky="NSEW")
self.labelpanel3 = tk.Label(self.frame5, text='Citra Segmentasi', font="arial
10 bold", bg="white")
    self.labelpanel3.grid(row=9,      padx=35,           rowspan=2,      column=2,
sticky="NSEW")

banner = Image.open("papaya1.jpg")
banner = banner.resize((750, 100), Image.ANTIALIAS)
banner = ImageTk.PhotoImage(banner)

self.panel4 = tk.Label(self.frame1, image=banner)
self.panel4.image_names = banner
self.panel4.grid(row=0, column=0, rowspan=5, sticky="NSEW")

def proses(self):
    datatraining = self.inputdatatrain.get()
    datatesting = self.inputdatatest.get()
    numpoint = self.textboxnumpoint.get()
    radius = self.textboxradius.get()
    a = lt1.imageprocessing()
    result, grayscale, hasil_segment = a.predicts(datatesting, datatraining,
numpoint, radius)
    print(result)

```

```
gambarasli = Image.open(datatesting)
gambarasli = gambarasli.resize((170, 150), Image.ANTIALIAS)
gambarasli = ImageTk.PhotoImage(gambarasli)
self.panel1.configure(image = gambarasli)
self.panel1.image_name = gambarasli

gambargray = Image.fromarray(grayscale)
gambargray = gambargray.resize((170, 150), Image.ANTIALIAS)
gambargray = ImageTk.PhotoImage(gambargray)
self.panel2.configure(image = gambargray)
self.panel2.image_name = gambargray

gambarsegment = Image.fromarray(hasil_segment)
gambarsegment = gambarsegment.resize((170, 150), Image.ANTIALIAS)
gambarsegment = ImageTk.PhotoImage(gambarsegment)
self.panel3.configure(image = gambarsegment)
self.panel3.image_name = gambarsegment

self.labeltext3.config(text=result[0])

def main():
    root = tk.Tk()
    root.geometry('770x590+50+50')
    root.configure(background='#BCE954')
    app = formpepaya(root)
    app.parent.title("Pepaya")
    root.mainloop()
if __name__=='__main__':
    main()
```

## 2. Image Processing

```
import cv2
from imutils import paths
import os
import numpy as np
from skimage import feature
from sklearn import neighbors

from matplotlib import pyplot as plt

class imageprocessing:

    def imPaths(self, imfolder):
        patharray = []
        for item in paths.list_images(imfolder):
            patharray.append(item)
        return patharray

    def preprocessing_1(self, img):
        grayscale = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        grayscale = cv2.resize(grayscale, (150, 150),
                             interpolation=cv2.INTER_AREA)
        equalize = cv2.equalizeHist(grayscale)
        blur = cv2.GaussianBlur(equalize, (5,5), 0)
        ret, alpha = cv2.threshold(np.uint8(blur), 50,255,
                                  cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)

        alpha[alpha==255] = 1
        hasil_segment = alpha * equalize
```

```

cnts      = cv2.findContours(hasil_segment,      cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

cnts = cnts[0] if len(cnts) == 2 else cnts[1]

cnts = sorted(cnts, key=cv2.contourArea, reverse=True)

for c in cnts:

    x,y,w,h = cv2.boundingRect(c)

    ROI = hasil_segment[y:y+h, x:x+w]

    break

return ROI, grayscale


def lbpfitur(self, img, num_point, radius):

    lbp_fitur = []

    # img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    lbp = feature.local_binary_pattern(img, 24, 8,
                                         method="uniform")

    (hist, _) = np.histogram(lbp, bins=np.arange(32), density=True)

    (hist, _) = np.histogram(lbp.ravel(), bins=np.arange(0, int(num_point) + 3),
                            range=(0, int(num_point) + 2))

    hist = hist.astype('float')

    hist /= (hist.sum() + 1e-7)

    values = np.array(hist).tolist()

    for i in range(len(values)):

        lbp_fitur.append(values[i])

    return lbp_fitur


def getClass(self, imfolder):

    label = []

    impath = self.imPaths(imfolder)

    for item in impath:

        label.append(os.path.split(os.path.dirname(item))[-1])

    return label

```

```

def create_datalatih(self, imfolder, num_point, radius):
    data_latih = []
    impath = self.imPaths(imfolder)
    for item in impath:
        img = cv2.imread(item)
        hasil_preprocessing, grayscale = self.preprocessing_1(img)
        hasil_ekstraksi = self.lbpfitur(hasil_preprocessing, num_point, radius)
        data_latih.append(hasil_ekstraksi)
    return data_latih

def create_datatesting(self, pathimg, num_point, radius):
    data_testing = []
    img = cv2.imread(pathimg)
    hasil_preprocessing, grayscale = self.preprocessing_1(img)
    plt.imshow(hasil_preprocessing, cmap='gray')
    plt.show()
    hasil_ekstraksi = self.lbpfitur(hasil_preprocessing, num_point, radius)
    data_testing.append(hasil_ekstraksi)
    return data_testing, grayscale, hasil_preprocessing

def model_knn(self, imfolder, num_point, radius):
    x = self.create_datalatih(imfolder, num_point, radius)
    y = self.getClass(imfolder)
    clf = neighbors.KNeighborsClassifier(n_neighbors=1, weights='distance',
                                         algorithm='auto', metric='euclidean')
    clf.fit(x,y)
    return clf

def predicts(self, imtest, imfolder, num_point, radius):

```

```
    data_test,    grayscale,    hasil_segment    =    self.create_datatesting(imtest,
num_point, radius)

    model = self.model_knn(imfolder, num_point, radius)
    hasil_klasifikasi = model.predict(data_test)
    return hasil_klasifikasi, grayscale, hasil_segment

# if __name__=='__main__':
#     folder_train = "D:/Data Nanas"
#     folder_test = "D:/Data_test/datest3.jpg"
#     print(predicts(folder_test, folder_train))
```

## **Lampiran 1. 2 Riwayat Hidup Peneliti**



Nama	:	Sela Marsela Ismail
NIM	:	T3116070
Tempat/Tanggal Lahir	:	Paguyaman, 21 Maret 1996
Email	:	selamarsela0104@gmail.com

### **Riwayat Pendidikan :**

1. Tahun 2008, menyelesaikan Pendidikan di Sekolah Dasar Negeri 6 Paguyaman, Kabupaten Boalemo, Provinsi Gorontalo
2. Tahun 2011, menyelesaikan Pendidikan di Sekolah Menengah Pertama Negeri 1 Paguyaman, Kabupaten Boalemo, Provinsi Gorontalo
3. Tahun 2014, menyelesaikan Pendidikan di Sekolah Menengah Atas Negeri 1 Paguyaman, Kabupaten Boalemo, Provinsi Gorontalo
4. Tahun 2016, telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo.

## Lampiran 1. 3 Surat Rekomendasi Bebas Plagiasi



### KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN UNIVERSITAS IHSAN (UNISAN) GORONTALO

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001  
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

#### **SURAT REKOMENDASI BEBAS PLAGIASI**

No. 0914/UNISAN-G/S-BP/VI/2021

Yang bertanda tangan di bawah ini:

Nama : Sunarto Taliki, M.Kom  
NIDN : 0906058301  
Unit Kerja : Pustikom, Universitas Ihsan Gorontalo

Dengan ini Menyatakan bahwa :

Nama Mahasiswa : SELA MARSELA ISMAIL  
NIM : T3116070  
Program Studi : Teknik Informatika (S1)  
Fakultas : Fakultas Ilmu Komputer  
Judul Skripsi : IDENTIFIKASI KEMATANGAN BUAH PEPAYA MENGGUNAKAN METODE LOCAL BINARY PATTERN (LBP) DAN K-NEAREST NEIGHBOR (K-NN)

Sesuai dengan hasil pengecekan tingkat kemiripan skripsi melalui aplikasi Turnitin untuk judul skripsi di atas diperoleh hasil Similarity sebesar 19%, berdasarkan SK Rektor No. 237/UNISAN-G/SK/IX/2019 tentang Panduan Pencegahan dan Penanggulangan Plagiarisme, bahwa batas kemiripan skripsi maksimal 35% dan sesuai dengan Surat Pernyataan dari kedua Pembimbing yang bersangkutan menyatakan bahwa isi softcopy skripsi yang diolah di Turnitin SAMA ISINYA dengan Skripsi Aslinya serta format penulisannya sudah sesuai dengan Buku Panduan Penulisan Skripsi, untuk itu skripsi tersebut di atas dinyatakan BEBAS PLAGIASI dan layak untuk diujikank.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 22 Juni 2021  
Tim Verifikasi,



**Sunarto Taliki, M.Kom**  
NIDN. 0906058301

Tembusan :

1. Dekan
2. Ketua Program Studi
3. Pembimbing I dan Pembimbing II
4. Yang bersangkutan
5. Arsip

T3116070 SELA MARSELA ISMAIL

## IDENTIFIKASI KEMATANGTAN BUAH PEPAYA MENGGUNAKAN...

## Sources Overview

19%

OVERALL SIMILARITY

1	e-jurnal.janabatra.ac.id INTERNET	3%
2	es.scribd.com INTERNET	1%
3	Zufrianto Y Lamasagi, Maryam Hasan, Yulianti Lasena, "Local Binary Pattern untuk Pengenalan Jenis Daun Tanaman Obat menggunakan...	1%
4	repositorydynamika.ac.id INTERNET	1%
5	repository.usd.ac.id INTERNET	<1%
6	www.scribd.com INTERNET	<1%
7	ej.unsoed.ac.id INTERNET	<1%
8	eprints.uns.ac.id INTERNET	<1%
9	garuda.ristekbrin.go.id INTERNET	<1%
10	ejournal.amikia.ac.id INTERNET	<1%
11	sinta.unud.ac.id INTERNET	<1%
12	repository.uin-suska.ac.id INTERNET	<1%
13	ejournal.unisi.ac.id INTERNET	<1%
14	e-jurnal.ippmunsera.org INTERNET	<1%
15	eprints.undip.ac.id INTERNET	<1%
16	jurnalatmukhur.ac.id INTERNET	<1%
17	repo.darmajaya.ac.id INTERNET	<1%
18	www.zeliti.com INTERNET	<1%
19	id.123dok.com INTERNET	<1%

## Lampiran 1. 4 Surat Keterangan Bebas Pustaka



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN  
UNIVERSITAS IHSAN GORONTALO  
UPT. PERPUSTAKAAN PUSAT  
SURAT KEPUTUSAN MENDIKS RI NO. 84/D/0/2001  
Jln. Achmad Nadjamuddin No.17 Telp. (0435) 829975 Fax. (0435) 829976 Gorontalo

### SURAT KETERANGAN BEBAS PUSTAKA

No.007/perpus\_fikom/VI/2021

Perpustakaan Rabu, 16 Juni 2021 Fakultas Ilmu komputer (FIKOM) Universitas Ihsan  
Gorontalo dengan ini menerangkan bahwa :

Nama	:	Sela Marsela Ismail
Nim	:	T3116070
No anggota	:	M202122

Terhitung sejak tanggal 16 Juni 2021, dinyatakan telah bebas dari pinjaman buku dan  
koleksi lainnya diperpustakaan Fakultas Ilmu komputer.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.

Gorontalo, 16 Juni 2021  
Kepala Perpustakaan  
Fakultas Ilmu Komputer

Apriyanto Alhamad, M.Kom  
NIDN 09240486

## Lampiran 1. 5 Surat Balasan Penelitian



**PEMERINTAH KABUPATEN BOALEMO  
KECAMATAN PAGUYAMAN  
DESA TANGKOBU**

*Jalan Trans Sulawesi Dusun Taruna Desa Tangkobu Kecamatan Paguyaman*

**SURAT – KETERANGAN**

Nomor :140/ 744 /DTkb-V/2021

Kepala Desa Tangkobu menerangkan dengan benar kepada :

Nama : Sela Marsela Ismail  
Tempat tanggal lahir : Paguyaman, 21 Maret 1996  
NIM : T3116070  
Jenis Kelamin : Perempuan  
Agama : Islam  
Pekerjaan : Mahasiswa  
Fakultas : Ilmu Komputer  
Program Study : Teknik Informatika  
Judul Penelitian : IDENTIFIKASI KEMATANGAN BUAH PEPAYA MENGGUNAKAN  
: METODE LOCAL BINARY PATTERN ( LBP ) DAN K- NEAREST  
: NEIGHBOR ( K- NN ).

Bahwa nama yang bersangkutan di atas benar-benar telah melaksanakan pengambilan Data dalam rangka penyusunan **Proposal atau Skripsi**.

Demikian Surat keterangan ini dibuat dengan benar untuk keperluan dan digunakan sebagaimana mestinya.-

Tangkobu, 31 Mei 2021

Kepala Desa Tangkobu

