

**PENERAPAN *LOCAL BINARY PATTERN* DAN *K-NEAREST  
NEIGHBOR* MENDETEKSI PENYAKIT PADA DAUN  
MANGGA**

**SKRIPSI**

**INTAN DWI NANDA**

**0701163097**



**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA  
MEDAN  
2021**

**PENERAPAN *LOCAL BINARY PATTERN* DAN *K-NEAREST  
NEIGHBOR* MENDETEKSI PENYAKIT PADA DAUN  
MANGGA**

**SKRIPSI**

*Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer*

**INTAN DWI NANDA**

**0701163097**



**PROGRAM STUDI ILMU KOMPUTER  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA  
MEDAN  
2021**

## PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp : -

Kepada Yth.,  
Dekan Fakultas Sains dan Teknologi  
UIN Sumatera Utara Medan

*Assalamu'alaikum Wr. Wb*

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengatakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama : Intan Dwi Nanda

Nomor Induk Mahasiswa : 0701163097

Program Study : Ilmu Komputer

Judul : Penerapan *Local Binary Pattern* Dan *K-Nearest Neighbor* Mendeteksi Penyakit Pada Daun Mangga

Dapat disetujui untuk segera di*Munaqasyahkan*. Atas perhatiannya kami ucapkan terimakasih.

Medan, 12 Agustus 2021 M  
1 Muharram 1443 H

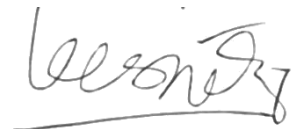
Komisi Pembimbing,

Pembimbing I,



Ilka Zufria, M.Kom.  
NIP. 198506042015031006

Pembimbing II,



Heri Santoso, M.Kom.  
NIB. 1100000114

## SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertandatangan di bawah ini:

Nama : Intan Dwi Nanda  
Nomor Induk Mahasiswa : 0701163097  
Program Studi : Ilmu Komputer  
Judul : Penerapan *Local Binary Pattern* Dan *K-Nearest Neighbor* Mendeteksi Penyakit Pada Daun Mangga

Dengan ini saya menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing dikutip oleh sumbernya. Apabila dikemudian hari ditemukan plagiarisme dalam skripsi ini, maka saya bersedia menerima sanksi pencabutan gelar akademik saya dan sanksi lainnya sesuai dengan ketentuan yang berlaku.

Medan, 12 Agustus 2021



Intan Dwi Nanda  
NIM. 0701163097



KEMENTERIAN AGAMA REPUBLIK INDONESIA UNIVERSITAS  
ISLAM NEGERI SUMATERA UTARA MEDAN FAKULTAS SAINS  
DAN TEKNOLOGI  
Jl. IAIN No. 1 Medan 20235  
Telp. (061) 6615683-6622925, Fax. (061) 6615683  
Url: <http://saintek.uinsu.ac.id>, E-mail: [saintek@uinsu.ac.id](mailto:saintek@uinsu.ac.id)

**PENGESAHAN SKRIPSI**

Nomor: B.167/ST/ST.V.2/PP.01.1/10/2021

Judul : Penerapan Local Binary Pattern dan K-Nearest Neighbor  
Mendeteksi Penyakit Pada Daun Mangga  
Nama : Intan Dwi Nanda  
Nomor Induk Mahasiswa : 0701163097  
Program Studi : Ilmu Komputer  
Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu Komputer Fakultas  
Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari/tanggal : Senin, 6 September 2021  
Tempat/media : Via Zoom Meeting

Tim Ujian Munaqasyah, b,  
Ketua,

Ilka Zufria, M.Kom  
NIP. 198506042015031006

Dewan Penguji,

Penguji I,

Ilka Zufria, M.Kom  
NIP. 198506042015031006

Penguji III,

Dr. Mhd. Furqan, S.Si., M.Comp.Sc.  
NIP. 198008062006041003

Penguji II,

Heri Santoso, M.Kom  
NIB. 1100000114

Penguji IV,

Sriani, S.Kom., M.Kom  
NIB. 1100000108

Mengesahkan,  
Dekan Fakultas Sains dan Teknologi  
UIN Sumatera Utara Medan,

Dr. Mhd. Syahnan, M.A.  
NIP. 196609051991031002

## ABSTRAK

Mangga merupakan salah satu jenis buah yang memiliki produksi tinggi dan banyak disukai oleh masyarakat. Produktivitas mangga berfluktuasi dari tahun ke tahun. Hal ini disebabkan fluktuasi luas panen, tanaman yang belum berproduksi secara optimal, gangguan iklim dan serangan berbagai hama dan penyakit yang menjadi faktor penghambat pertumbuhan dan produksi mangga di Indonesia. Identifikasi tersebut akan memakan waktu yang relatif lama dan menghasilkan berbagai penyakit pada daun mangga karena manusia memiliki keterbatasan visual dalam mengidentifikasi, tingkat kelelahan dan perbedaan pendapat tentang penyakit pada daun mangga. Proses pengenalan pola daun dapat dilakukan dengan cara mengenali ciri-ciri struktur daun seperti bentuk dan tekstur daun. Metode yang digunakan dalam penelitian ini adalah *Local Binary Pattern* yaitu algoritma yang dapat digunakan untuk mengklasifikasikan berdasarkan citra, sedangkan *k-nearest neighbor* digunakan untuk pendeteksian penyakit pada citra daun mangga. Setiap nilai bobot dari citra latih dan citra uji akan dibandingkan dengan meminimalkan nilai *Euclidean Distance*. Pada penelitian ini digunakan 4 jenis penyakit daun mangga. Berdasarkan hasil uji akurasi diperoleh nilai akurasi sebesar 87,5% untuk proses pendeteksian penyakit pada daun mangga sebanyak 16 data.

Kata Kunci: Daun, *Local Binary Pattern*, *K-Nearest Neighbor* (K-NN).

## **ABSTRACT**

Mango is one type of fruit that has a high production and is widely liked by the community. Mango productivity fluctuates from year to year. This is due to fluctuations in harvested area, plants that have not produced optimally, climate disturbances and attacks by various pests and diseases which are inhibiting factors for mango growth and production in Indonesia. This identification will take a relatively long time and produce various diseases on mango leaves because humans have visual limitations in identifying, fatigue levels and differences of opinion about diseases in mango leaves. The process of recognizing leaf patterns can be done by recognizing leaf structure characteristics such as leaf shape and texture. The method used in this study is Local Binary Pattern, which is an algorithm that can be used to classify based on image, while k-nearest neighbor is used for disease detection in mango leaf images. Each training image weight value and test image will be compared by minimizing the Euclidean Distance value. In this study, 4 types of mango leaf disease were used. Based on the results of the accuracy test, an accuracy value of 87.5% was obtained for the process of detecting disease in mango leaves as many as 16 data.

Keywords: Leaf , *Local Binary Pattern*, *K-Nearest Neighbors* (K-NN) .

## KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, hidayah, dan rahmat-Nya serta memberikan petunjuk dan kemudahan sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul **“Penerapan Lokal Binary Pattern dan K-Nearest Neighbor Mendeteksi Penyakit Pada Daun Mangga”**. Tidak lupa juga sholawat dan salam semoga selalu tercurah kepada uswatun hasanah kita, Nabi Muhammad SAW beserta keluarganya, sahabat dan pengikutnya termasuk kita semua yang selalu mengharapkan syafaatnya di hari akhir.

Penulis menyadari bahwa penyusunan proposal skripsi ini tidak lepas dari doa, perhatian, bantuan, bimbingan, motivasi dan dukungan dari berbagai pihak sehingga dengan keikhlasan dan kerendahan hati pada kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. Syahrin Harahap, MA, selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.
2. Bapak Dr. Mhd. Syahnan, M.A, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
3. Bapak Ilka Zufria, M.Kom selaku Ketua Jurusan Ilmu Komputer Fakultas Sains dan Teknologi, juga selaku Pembimbing Skripsi pertama yang telah berjasa dalam membantu penulis dalam memberikan ide, saran, kritik, dan bimbingan kepada penulis selama penulis mengerjakan proposal skripsi ini.
4. Bapak Rakhmat Kurniawan R,S.T., M.Kom selaku Sekretaris Jurusan Ilmu Komputer, juga selaku dosen pembimbing Akademik Fakultas Sains dan Teknologi.
5. Bapak Heri Santoso, M.Kom selaku pembimbing skripsi kedua yang telah berjasa membantu penulis dalam memberikan ide, saran, kritik, dan



bimbingan kepada penulis selama penulis mengerjakan proposal skripsi ini.

6. Seluruh staf pengajar dan pegawai Program Studi S1 Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
7. Terutama terima kasih kepada kedua orang tuaku tercinta yaitu Muhammad Nawawi Lubis dan Endang Tri Wahyuni Sitorus yang telah memberikan bantuan moril dan materil, semangat dan doa bagi penulis.
8. Teman-teman Ilmu Komputer 3 yang selalu memberikan dukungan dan arahan kepada penulis, harapan penulis untuk teman-teman ilmu komputer 3 pada masa pandemi ini tetap bersemangat dalam menjalankan aktivitas serta perjuangan untuk meraih gelar sarjananya.
9. Kepada abang, kakak, dan adik kandung penulis, Dewi Sukma Lubis, Zoan Antonio Vizi Lubis, Enggar Zaky Ramadhan Lubis, terima kasih untuk dukungan, doa dan semangat, serta bantuan baik moril maupun materil yang selalu diberikan kepada penulis.
10. Dan semua pihak yang telah membantu penulis namun tidak dapat disebutkan satu persatu.

Penulis menyadari sepenuhnya bahwa dalam pembuatan proposal skripsi ini masih jauh dari kesempurnaan, untuk itu kritik dan saran yang membangun sangat diharapkan dari para pembaca. Semoga hasil proposal skripsi ini dapat menjadi ibadah bagi penulis dan bermanfaat bagi para pembaca. Aamin Ya Rabbal'alamin.

Medan, 12 Agustus 2021

Hormat Saya,

Intan Dwi Nanda

## DAFTAR ISI

<b>ABSTRAK</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>KATA PENGANTAR</b> .....	<b>iii</b>
<b>DAFTAR ISI</b> .....	<b>v</b>
<b>DAFTAR GAMBAR</b> .....	<b>vii</b>
<b>DAFTAR TABEL</b> .....	<b>viii</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>ix</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah .....	3
1.4. Tujuan Penelitian .....	4
1.5. Manfaat penelitian .....	4
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1. Tanaman Mangga .....	5
2.2. Pengolahan Citra.....	8
2.2.1. Citra Biner .....	8
2.2.2. Citra RGB .....	8
2.2.3. Citra <i>Grayscale</i> .....	9
2.3. Ekstraksi Fitur.....	10
2.4. Format File Citra .....	11
2.4.1. Bitmap (bmp).....	11
2.5. <i>Local Binary Pattern</i> .....	11
2.6. Histogram Citra .....	14
2.7. Metode K-Nearest Neighbor (KNN) .....	16
2.8. Jarak <i>Euclidean</i> .....	17
2.9. Normalisasi Histogram .....	18
2.10. Perhitungan Akurasi .....	18
2.11. Matlab.....	18

2.12. <i>Flowchart</i> .....	19
2.13. Penelitian Terkait.....	20
<b>BAB III METODE PENELITIAN.....</b>	<b>25</b>
3.1. Tempat dan Waktu Penelitian.....	25
3.1.1. Tempat Penelitian .....	25
3.1.2. Waktu Penelitian .....	25
3.2. Bahan dan Alat Penelitian .....	26
3.2.1. Perangkat Keras.....	26
3.2.2. Perangkat Lunak.....	26
3.3. Prosedur Kerja .....	26
3.3.1. Teknik Pengumpulan Data .....	27
3.3.2. Analisis Kebutuhan .....	28
3.3.3. Perancangan.....	28
3.3.4. Pengujian .....	30
3.3.5. Penerapan/Penggunaan.....	30
<b>BAB IV HASIL &amp; PEMBAHASAN .....</b>	<b>31</b>
4.1. Pembahasan .....	31
4.1.1. Analisis Data .....	31
4.1.2. <i>Representasi Data</i> .....	31
4.1.3. Hasil Analisis .....	32
4.1.4. Perancangan Sistem.....	36
4.2. Hasil .....	39
4.2.1. Pengujian .....	39
4.2.2. Penerapan .....	48
<b>BAB V KESIMPULAN &amp; SARAN .....</b>	<b>49</b>
5.1. Kesimpulan .....	49
5.2. Saran .....	49
<b>DAFTAR PUSTAKA .....</b>	<b>50</b>
<b>LAMPIRAN-LAMPIRA</b>	

## DAFTAR GAMBAR

<b>Gambar</b>	<b>Judul Gambar</b>	<b>Halaman</b>
2.1.	Langkah & Perhitungan LBP .....	12
2.2.	Histogram Citra .....	15
2.3.	Tabulasi Histogram .....	15
3.1.	Tahap-tahap Prosedur Kerja .....	27
3.2.	Diagram Perencanaan .....	29
4.1.	Citra <i>Grayscale</i> .....	32
4.2.	Citra <i>Grayscale</i> .....	33
4.3.	Citra <i>Grayscale</i> .....	34
4.4.	Flowchart Sistem .....	37
4.5.	Flowchart Metode <i>K-Nearest Neighbord</i> .....	38
4.6.	Perancangan Antar Muka.....	39
4.7.	Tampilan Aplikasi.....	40
4.8.	Tampilan Input Citra.....	40
4.9.	Tampilan Citra <i>Grayscale</i> .....	41
4.10.	Tampilan Nilai Esktraksi & Histogram Citra .....	41
4.11.	Tampilan Nilai Data Uji & Data Latih .....	42
4.12.	Tampilan Hasil Deteksi Penyakit Cendawan Jelaga .....	42
4.13.	Tampilan Hasil Deteksi Penyakit Ulat Daun .....	43
4.14.	Tampilan Hasil Deteksi Penyakit Hama Kutu Putih.....	44
4.15.	Tampilan Hasil Deteksi Penyakit Hama Thrips Mangga .....	44

## DAFTAR TABEL

<b>Tabel</b>	<b>Judul Tabel</b>	<b>Halaman</b>
2.1.	Jenis Penyakit Pada Daun Mangga .....	6
2.2.	Perhitungan Histogram .....	16
2.3.	Simbol <i>Flowchart</i> .....	20
2.4.	Penelitian Terkait .....	21
3.1.	Waktu dan Jadwal Penelitian .....	25
4.1.	Matriks Citra .....	32
4.2.	Perhitungan <i>Local Binary Pattern</i> .....	33
4.3.	Perhitungan <i>Local Binary Pattern</i> .....	34
4.4.	Dataset .....	35
4.5.	Hasil Perhitungan Jarak <i>Eucliden</i> .....	36
4.6.	Pengujian Sistem .....	45
4.7.	Hasil Pengujian Sistem .....	46

## DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1.	Hasil uji citra daun mangga pada Penerapan <i>Local Binary Pattern</i> Dan <i>K-Nearest Neighbor</i> Mendeteksi Penyakit Pada Daun Mangga.
2.	Kode Program Matlab untuk Penerapan <i>Local Binary Pattern</i> Dan <i>K-Nearest Neighbor</i> Mendeteksi Penyakit Pada Daun Mangga.
3.	Kartu Bimbingan Skripsi
4.	Riwayat Hidup

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Tumbuhan adalah bagian terpenting dari kehidupan di bumi. Tumbuhan bermanfaat sebagai pemasok oksigen untuk bernafas, sebagai makanan, bahan bakar, obat-obatan, kosmetik dan masih banyak lagi. Proses pengelompokan tumbuhan dapat dilakukan dengan cara mengidentifikasi citra bentuk daun tumbuhan itu sendiri. Cara mengambil gambar daun dari tanaman ini, selanjutnya langkah dapat diambil untuk mengenali pola daun dengan mengenali ciri-ciri struktur daun seperti bentuk dan tekstur daun (Liantoni, 2015). Sebagaimana yang dijelaskan dalam surah An-Nahl: 11 yang berbunyi:

يُنْبِتُ لَكُمْ بِهِ الزَّرْعَ وَالزَّيْتُونَ وَالنَّخِيلَ وَالْأَعْنَابَ وَمِنْ كُلِّ الثَّمَرَاتِ إِنَّ فِي ذَلِكَ  
لَآيَةً لِّقَوْمٍ يَتَفَكَّرُونَ . ( ١١ )

Artinya: “Dia menumbuhkan kamu dengan air hujan itu tanaman; zaitun, kurma, anggur, dan segala jenis buah-buahan. Sesungguhnya pada yang demikian itu terdapat tanda-tanda (kekuasaan Allah) bagi kaum yang berpikir.”

Dalam ayat tersebut dikatakan bahwa Allah menumbuhkan dengan air hujan tumbuh-tumbuhan, sumber makananmu. Allah juga menanam zaitun, kurma dan anggur. Allah menumbuhkan segala macam buah. Sesungguhnya air hujan dan apa yang ditumbuhkannya mengandung tanda kekuasaan Allah bagi suatu kaum yang memikirkan tentang ciptaan-Nya, kemudian mereka menganggapnya sebagai bukti kebesaran Allah SWT.

Mangga merupakan salah satu jenis buah yang memiliki produksi tinggi dan banyak disukai oleh masyarakat. Produktivitas mangga berfluktuasi dari tahun ke tahun. Hal ini disebabkan fluktuasi luas panen, tanaman yang belum

berproduksi secara optimal, gangguan iklim dan serangan berbagai hama dan penyakit yang menjadi faktor penghambat pertumbuhan dan produksi mangga di Indonesia (Zuliyanti, 2017).

Proses pengenalan pola daun dapat dilakukan dengan mengenali ciri-ciri struktur daun seperti bentuk dan tekstur daun. Metode pengolahan citra masukan dengan memanfaatkan teknik pengolahan citra digital dilakukan untuk menganalisis karakteristik struktur daun. Perkembangan teknologi untuk teknik pengolahan citra juga berkembang pesat. Berbagai teknik telah dikembangkan untuk memudahkan pekerjaan manusia, baik sebagai pengolah gambar, analisis gambar maupun pengguna gambar untuk berbagai keperluan dan tujuan. Sering kali gambar yang digunakan tidak dalam kondisi ideal untuk dipelajari karena banyak gangguan, bisa berupa bayangan, foto atau gambar buram, ketidakjelasan tampilan objek sehingga dapat menimbulkan masalah dan mempengaruhi hasil interpolasi. dan akan mempengaruhi analisis dan perencanaan yang dilakukan, sehingga diperlukan berbagai teknik pengolahan. citra untuk mendapatkan citra yang ideal.

Pada penelitian ini penulis bermaksud untuk membuat sebuah sistem penelitian yang dapat mendeteksi penyakit pada daun mangga menggunakan algoritma *local binary pattern* dan metode *k-nearest neighbor* untuk menganalisis apakah kedua metode tersebut mampu mendeteksi penyakit pada daun mangga dengan mendapatkan akurasi yang lebih baik.

*Local Binary Pattern* didefinisikan sebagai nilai biner piksel di tengah gambar dengan 8 nilai piksel di sekitarnya. *Local Binary Pattern* merupakan algoritma yang dapat digunakan untuk mengklasifikasikan berdasarkan citra.



Metode *k-nearest neighbor* merupakan metode klasifikasi yang menentukan label (kelas) suatu objek baru berdasarkan mayoritas kelas dari jarak terdekat  $k$  pada kelompok data latih. Nilai  $k$  yang digunakan adalah 3 dan 5 digunakan dalam menggunakan metode *k-nearest neighbor*. Sedangkan perhitungan jarak menggunakan metode *Euclidean Distance*. *K-nearest neighbor* akan mengklasifikasikan citra uji ke dalam kelas dengan jumlah anggota terbanyak. Prinsip kerja  $k$ -nn adalah mencari jarak terpendek antara data yang akan dievaluasi dengan  $k$  tetangga terdekat pada data latih (Wijaya, 2019).

Penelitian ini dilakukan untuk menerapkan algoritma *local binary pattern* dan metode *K-Nearest Neighbor* dalam mendeteksi penyakit pada daun mangga, penelitian ini menggunakan format .bmp. dan juga menggunakan histogram untuk menggambarkan penyebaran nilai intensitas piksel suatu gambar. Kemudian gunakan jarak *Euclidean* untuk menghitung kemiripan 2 vektor pada daun mangga. Penelitian ini juga diharapkan mendapatkan hasil yang baik dalam mendeteksi penyakit pada daun mangga.

## 1.2. Rumusan Masalah

Adapun rumusan masalah berdasarkan latar belakang masalah diatas, yaitu sebagai berikut:

1. Bagaimana menerapkan *local binary pattern* dan metode *k-nearest neighbor* kedalam sistem yang dapat mendeteksi penyakit pada daun mangga ?
2. Bagaimana hasil penerapan *local binary pattern* dan metode *k-nearest neighbor* dalam mendeteksi penyakit pada daun mangga ?

## 1.3. Batasan Masalah

Adapun batasan – batasan masalah, yaitu sebagai berikut:

1. Metode yang digunakan adalah algoritma *local binary pattern* dan metode *k-nearest neighbor*.
2. Data penelitian pada penyakit daun mangga yang digunakan adalah 4 jenis penyakit daun mangga.
3. Penelitian ini hanya mendeteksi penyakit pada daun mangga.

4. Daun mangga yang digunakan adalah dari pohon mangga harum manis.
5. Data citra menggunakan file gambar dengan format \*. Bitmap (bmp).
6. Citra yang di uji menggunakan citra *Grayscale*.
7. Citra yang digunakan beresolusi 320 x 240 piksel.
8. Pengambilan dataset menggunakan smartphone Vivo Y93.

#### **1.4. Tujuan Penelitian**

Adapun tujuan dari rumusan masalah diatas yaitu, sebagai berikut:

1. Menerapkan algoritma *local binary pattern* dan metode *k-nearest neighbor* kedalam sistem yang dapat mengenali penyakit tanaman pada daun mangga.
2. Mengetahui gejala-gejala penyakit pada daun mangga.
3. Memperoleh hasil akurasi yang baik dari algoritma *local binary pattern* dan metode *k-nearest neighbor* dalam mengenali penyakit pada daun mangga.

#### **1.5. Manfaat Penelitian**

Adapun manfaat dan penelitian ini adalah sebagai berikut:

1. Memahami dan menerapkan algoritma *local binary pattern* dan metode *k-nearest neighbor* dalam mendeteksi penyakit pada daun mangga.
2. Mengenali jenis-jenis penyakit daun mangga berdasarkan gejala pada daun mangga.
3. Mengetahui pencegahan apa saja yang dilakukan untuk mencegah penyakit pada daun mangga.

## **BAB II**



### **TINJAUAN PUSTAKA**



#### **2.1. Tanaman Mangga**

Mangga merupakan tanaman buah berpotensi untuk dikembangkan karena memiliki tingkat keragaman genetik yang tinggi. Daun mangga memiliki variasi bentuk, ukuran dan warna daun yang menunjukkan keragaman genetik yang cukup luas. Struktur tulang daun merupakan salah satu ciri yang dapat membedakan daun dari berbagai jenis tumbuhan, karena urat daun memiliki keunikan tersendiri pada setiap jenis tumbuhan. Daun pohon mangga pada dasarnya memiliki berbagai macam bentuk daun mangga. Namun pada umumnya daunnya merupakan jenis daun tunggal dengan letak tersebar di seluruh cabang pohon. Tulang daun pohon mangga menyirip (Liantoni, 2015).

Daun terdiri dari dua bagian yaitu kaki daun dan badan daun. Tubuh daun bertulang dan berurat, di antara tulang dan urat ditutupi dengan daging daun. Daging daun terdiri dari jumlah sel yang tidak terbatas. Panjang tangkai daun bervariasi dari 1,25 – 12,5 cm, pangkal membesar dan pada sisi atas terdapat alur. Susunan daun pada batang biasanya  $\frac{3}{8}$ , tetapi semakin dekat ujung semakin rapat susunan daun pada batang biasanya  $\frac{3}{8}$ , tetapi semakin dekat ujung semakin mendekati lingkaran. Jenis tulang daun ini memiliki susunan seperti sirip ikan. Panjang tangkai daun bervariasi dari 1,25-12,5 cm. Fungsi daun adalah membuat makanan, pernapasan, dan penguapan. Daun merupakan salah satu bagian tumbuhan yang memiliki ciri khas sehingga dapat dijadikan sebagai acuan dalam pengolahan citra digital. Proses pengenalan daun dapat diperoleh berdasarkan deteksi penyakit pada daun mangga.

Berikut ini adalah Tabel 2.1. Data jenis penyakit pada daun mangga berdasarkan gejalanya (Raharjo A. A, 2017)

No.	Jenis penyakit	Ciri-ciri penyakit daun mangga	Gambar daun mangga
1.	Cendawan Jelaga	<ol style="list-style-type: none"> <li>1. Munculnya bintik-bintik hitam disekitar daun mangga.</li> <li>2. Daun mangga yang semula berwarna hijau berubah menjadi warna hitam.</li> <li>3. Pada musim kemarau, serangan jamur bisa menyebabkan daun menjadi kering dan mengering.</li> </ol>	
2.	Hama kutu putih	<ol style="list-style-type: none"> <li>1. Banyak kutu putih pada daun mangga.</li> <li>2. Hama kutu putih berbentuk oval, datar, tertutup lapisan tebal seperti lilin.</li> <li>3. Sering hinggap pada daun dan menghisap</li> </ol>	

		cairan sel daun.	
3.	Hama <i>striga</i> mangga	<ol style="list-style-type: none"> <li>1. Penyakit ini menyerang daun muda pada mangga.</li> <li>2. Adanya Bekas tusukan yang menjadi sumber penyakit.</li> <li>3. Daun mangga muda dan bunga mangga kelihatan seperti terbakar, warna coklat dan menggeling.</li> </ol>	
4.	Ulat daun ( <i>Handaulum Doloskallia Polibeta</i> ).	<ol style="list-style-type: none"> <li>1. Daun mangga sobek dan bergerigi bekas gigitan.</li> <li>2. Jika serangan berat, hanya tersisa tulang daun saja.</li> <li>3. Daun mangga yang tererang penyakit yaitu daun mangga muda dan tua.</li> </ol>	

## **2.2. Pengolahan Citra**

Arti kata mengolah menurut Kamus Besar Bahasa Indonesia (KBBI) adalah cara atau proses mencoba sesuatu agar berbeda atau menjadi lebih sempurna. Sedangkan menurut KBBI, citra berarti suatu bentuk atau citra, dalam hal ini adalah citra yang diperoleh dengan menggunakan sistem visual. Pengolahan citra secara keseluruhan berarti suatu cara untuk mencoba membuat suatu citra menjadi sebuah citra lain yang lebih sempurna atau diinginkan. Dengan kata lain, pengolahan citra adalah suatu proses memasukkan suatu citra dan menghasilkan keluaran berupa citra yang diinginkan (Sulistyanti dkk, 2016).

Citra menurut Webster berarti representasi, kemiripan atau tiruan dari suatu objek. Misalnya, foto apel mewakili identitas apel di depan kamera. Gambar bisa dalam bentuk fotografi, lukisan, atau gambar dan coretan yang terjadi di atas kertas, kanvas dan pada layar monitor juga dapat dikatakan bahwa gambar tersebut gelap-terang, redup-terang, atau variasi warna-warni di lapangan datar. Pengungkapan formalitas dengan angka yang mewakili variasi kecerahan atau intensitas warna dalam arah horizontal dan vertikal. Ada beberapa jenis citra, diantaranya (Putra, 2010) sebagai berikut;

### **2.2.1. Citra Biner**

Citra digital yang hanya memiliki dua kemungkinan nilai piksel, yaitu hitam dan putih. Gambar biner juga disebut sebagai gambar B&W (hitam putih) atau gambar monokrom. Hanya dibutuhkan 1 bit untuk merepresentasikan nilai setiap piksel dari citra biner. Gambar biner sering muncul sebagai akibat dari pemrosesan seperti segmentasi, floating, morfologi, atau dithering.

### **2.2.2. Citra RGB**

RGB merupakan model warna yang terdiri dari 3 warna yaitu merah (red), hijau (green), dan biru (blue), yang ditambahkan dengan berbagai cara untuk menghasilkan berbagai warna. Warna RGB adalah model warna yang didasarkan pada konsep peningkatan intensitas cahaya. Yang utama adalah

merah, hijau dan biru. Di sebuah ruangan di mana sama sekali tidak ada cahaya, ruangan itu benar-benar gelap. Tidak ada sinyal gelombang cahaya yang diserap oleh mata kita atau RGB (0, 0, 0). Jika kita menambahkan lampu merah ke ruangan, ruangan akan berubah warna menjadi merah, misalnya RGB(255, 0, 0), semua benda di dalam ruangan hanya bisa tampak merah. Jadi saat kita mengganti lampunya dengan warna hijau atau biru.

Seperti yang kita ketahui bahwa RGB atau Merah, Hijau, Biru adalah sistem pewarnaan untuk tampilan digital dan banyak digunakan untuk monitor komputer, video, layar handphone, dll. Sistem warna RGB terdiri dari 100% Merah, 100% Hijau dan 100% Biru yang menghasilkan 100% putih. Tidak ada warna hitam dalam RGB (Prabowo, 2018).

### **2.2.3. Citra *Grayscale***

Citra grayscale adalah citra digital yang hanya memiliki satu nilai kanal untuk setiap pikselnya, dengan kata lain nilai bagian MERAH = HIJAU = BIRU. Nilai ini digunakan untuk menunjukkan tingkat intensitas. Warna yang dimilikinya adalah hitam, abu-abu, dan putih. tingkat keabuan disini adalah warna keabuan dengan berbagai tingkatan dari hitam sampai mendekati putih. Citra *grayscale* berikut ini memiliki kedalaman warna 8 bit (256 kombinasi warna abu-abu) sebagai berikut:

#### a. Citra Warna (8 bit)

Setiap piksel dari suatu citra berwarna (8 bit) hanya diwakili oleh 98 bit dengan jumlah maksimal yang dapat digunakan adalah 256 warna. Ada dua jenis gambar 8-bit. Pertama, gambar berwarna 8-bit menggunakan 256 palet warna dengan masing-masing palet memiliki peta warna RGB tertentu.

b. Citra Warna (16 bit)

Gambar berwarna 16-bit biasanya disebut sebagai gambar highcolor. Setiap piksel diwakili oleh 2 byte memori (16 bit). Warna 16 bit memiliki 65.536 warna. Dalam pembentukan bit, nilai merah dan biru terjadi di 5 bit di kanan dan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra atau ekstra bit. Pemilik komponen hijau dengan seri 6-bit ini karena penglihatan manusia lebih sensitif terhadap warna hijau.

c. Citra Warna (24 bit)

Setiap piksel dari citra berwarna 24-bit diwakili oleh 24 bit dengan total 16.777.216 variasi warna. Variasi ini lebih dari cukup untuk memvisualisasikan semua warna yang dapat dilihat oleh mata manusia. Penglihatan manusia diyakini hanya mampu membedakan hingga 10 juta warna. Setiap titik informasi piksel (RGB) disimpan dalam 1 byte data. 8 bit pertama menyimpan nilai biru, kemudian diikuti oleh nilai hijau pada 8 bit kedua dan 8 bit terakhir berwarna merah.

## 2.4. Ekstrasi Fitur

*Feature Extraction* (Ekstraksi Fitur) adalah praktik dari fitur atau suatu form yang nantinya nilai yang diperoleh akan dianalisa menggunakan proses selanjutnya. Ekstraksi fitur Ini dilakukan dengan menghitung jumlah titik atau piksel yang ditemui pada setiap pemeriksaan, di mana pemeriksaan dilakukan pada berbagai arah penelusuran, pengecekan koordinat kartesius dari citra digital yang dianalisis yaitu vertikal, horizontal, diagonal kanan dan diagonal kiri. Fitur adalah karakteristik unik dari suatu objek. Fitur dibagi menjadi dua, yaitu 'alami' adalah bagian dari gambar, seperti kecerahan dan tepi objek, Selanjutnya fitur 'buatan' adalah fitur yang diperoleh dengan operasi tertentu pada gambar, seperti histogram tingkat abu-abu. Sehingga ekstraksi ciri merupakan suatu proses memperoleh ciri-ciri yang membedakan suatu objek dengan objek lainnya (Putra, 2010)



#### a. Ciri Tekstur

Tekstur merupakan fitur citra yang digunakan untuk menentukan karakteristik suatu citra. Ini karena, tekstur mengandung informasi penting tentang komposisi struktur permukaan pada gambar.

### 2.3. Format File Citra

Ada beberapa format file gambar standar yang digunakan saat ini. Format ini digunakan dalam menyiapkan gambar dalam file. Setiap format memiliki karakteristik tersendiri (Putra, 2010). dibawah ini adalah bentuk format yang digunakan dalam penelitian ini yaitu sebagai berikut:

#### 2.3.1. Bitmap (.bmp)

Format .bmp adalah format penyimpanan terkompresi standar yang umumnya dapat digunakan untuk menyimpan gambar biner ke gambar berwarna. Format ini terdiri dari beberapa jenis, masing-masing jenis ditentukan oleh jumlah bit yang digunakan untuk menyimpan suatu nilai piksel.

### 2.4. Local Binary Pattern

*Local Binary Pattern* pertama kali diperkenalkan pada tahun 1992 oleh *Timo Ojala* dan *David Harwood* di *University of Maryland*. *Local Binary Pattern* didefinisikan sebagai rasio nilai biner piksel di tengah gambar dengan 8 nilai piksel di sekitarnya. *Local Binary Pattern* adalah deskriptor untuk mendefinisikan citra berdasarkan tekstur citra. Jadi, sebuah gambar berukuran 3x3, di mana nilai biner di tengah gambar dibandingkan dengan nilai di sekitarnya. Jika intensitas piksel pusat lebih besar dari biner pusat maka nilainya ditetapkan ke 1, jika kurang dari 0 (Purwati, tidak ada tahun).

Dengan 8 piksel di sekitarnya berarti ada  $2^8 = 256$  kemungkinan kombinasi kode *local binary pattern*.

$$LBP_{R,P} = \sum_{p=0}^{P-1} S(g_p - g_c) \cdot 2^p \quad (2.1)$$

$$S(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x \leq 0 \end{cases} \quad (2.2)$$

Keterangan Variabel :

$x_c$  dan  $y_c$  = koordinat pusat piksel ketetangaan

$p$  = circular sampling point

$P$  = banyaknya sampling point

$g_p$  = nilai keabuan dari  $p$

$g_c$  = nilai rata-rata piksel ketetangaan dan nilai pusat

$s(x)$  = thresholding

*Local binary pattern* memiliki label yang ditandai dengan P dan R. P mewakili jumlah piksel tetangga yang digunakan dalam perhitungan sedangkan R adalah jari-jari antara piksel titik pusat dan piksel tetangga. Berikut ini adalah contoh matriks 3 x 3 dalam perhitungan *local binary pattern* ( Hayaty, 2017).

Nilai Diagram 3x3	Hasil Threshold	Bobot LBP	Hasil hasil Threshold dengan Bobot LBP	Nilai Local Binary Pattern																																				
<table border="1"> <tr><td>251</td><td>252</td><td>254</td></tr> <tr><td>251</td><td>252</td><td>254</td></tr> <tr><td>254</td><td>252</td><td>254</td></tr> </table>	251	252	254	251	252	254	254	252	254	<table border="1"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td></td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	1	1		1	1	1	1	<table border="1"> <tr><td>1</td><td>2</td><td>4</td></tr> <tr><td>128</td><td></td><td>8</td></tr> <tr><td>64</td><td>32</td><td>16</td></tr> </table>	1	2	4	128		8	64	32	16	<table border="1"> <tr><td>1</td><td>2</td><td>4</td></tr> <tr><td>128</td><td></td><td>8</td></tr> <tr><td>64</td><td>32</td><td>16</td></tr> </table>	1	2	4	128		8	64	32	16	$\begin{aligned} \text{LBP} &= 1 + 2 + 4 + 8 + 16 \\ &\quad + 32 + 64 + 128 \\ &= 255 \end{aligned}$
251	252	254																																						
251	252	254																																						
254	252	254																																						
1	1	1																																						
1		1																																						
1	1	1																																						
1	2	4																																						
128		8																																						
64	32	16																																						
1	2	4																																						
128		8																																						
64	32	16																																						

**Gambar 2.1.** Langkah dan hasil Perhitungan LBP

(simki.unpkediri.ac.id)

Berdasarkan nilai ambang batas *local binary pattern*, maka nilai *local binary pattern* dapat diperoleh dengan mengubah nilai tetangga menjadi 1 atau 0. Nilai 1 akan diperoleh jika piksel tetangga memiliki nilai lebih besar atau sama dengan piksel tengah. Jika lebih kecil dari piksel tengah, maka piksel tetangga

diberi nilai 0. Untuk setiap nilai piksel baru dikalikan dengan  $2P$ .  $P$  memiliki nilai dari 0 sampai 7. Selanjutnya menghitung nilai *local binary pattern* untuk piksel tengah dimulai dari piksel di sekitarnya dengan cara searah jarum jam (searah jarum jam) atau berlawanan arah jarum jam (searah jarum jam). Nilai 0 sesuai dengan arah jarum jam nilainya bertambah 1. Kemudian hasil uji biner diubah menjadi desimal. Kemudian dapat divisualisasikan yang dapat disebut sebagai proses *thresholding*, yaitu mengumpulkan biner dan menyimpan nilai desimal pada keluaran *local binary pattern* yang diulang untuk setiap piksel pada citra masukan (Hataty, 2017).

Berikut ini adalah vektor *local binary pattern*, dalam bentuknya yang paling sederhana, dibuat dengan cara sebagai berikut:

- a. Bagilah jendela yang diperiksa menjadi sel (misalnya  $16 \times 16$  piksel untuk setiap sel).
- b. Untuk setiap piksel dalam sel, bandingkan piksel dengan masing-masing 8 tetangganya (di kiri atas, kiri tengah, kiri bawah, kanan atas, dll.). Ikuti piksel di sepanjang lingkaran, misalnya, Searah jarum jam atau berlawanan arah jarum jam.
- c. Pada langkah di atas, tetangga dianggap dapat diubah dengan memvariasikan jari-jari lingkaran di sekitar pixel,  $R$  dan quantisation dari ruang sudut  $P$ .
- d. Jika nilai pixel tengah lebih besar dari nilai tetangga, tulis "0". Kalau tidak, tulis "1". Ini memberikan angka biner 8 digit (yang biasanya dikonversi menjadi desimal untuk kenyamanan).
- e. Hitung histogram, di atas sel, dari frekuensi setiap "angka" yang terjadi (yaitu, masing-masing kombinasi piksel yang lebih kecil dan yang lebih besar dari pusat). Histogram ini dapat dilihat sebagai vektor fitur 256-dimensi.
- f. Opsional menormalkan histogram.
- g. Histogram gabungan (dinormalisasi) dari semua sel. Ini memberikan vektor fitur untuk seluruh jendela.

Vektor fitur sekarang dapat diproses menggunakan beberapa algoritma pembelajaran mesin untuk mengklasifikasikan gambar. Klasifikasi semacam itu sering digunakan untuk pengenalan wajah atau analisis tekstur.

## 2.5. Histogram Citra

Histogram adalah grafik yang menunjukkan frekuensi kemunculan setiap nilai gradasi warna. Jika digambarkan dalam titik koordinat kartesius, Sumbu X menunjukkan tingkat warna dan sumbu Y menunjukkan frekuensi kemunculan jumlah piksel (relatif) intensitas pada gambar (Zufria, 2018).

Histogram juga dapat memberi tahu Anda banyak hal tentang kecerahan dan kontras gambar. Oleh karena itu, histogram adalah alat yang berharga dalam pekerjaan pemrosesan gambar baik secara kualitatif maupun kuantitatif (Hermawati, F. A, 2013).

Misalkan citra digital memiliki derajat keabuan  $L$ , yaitu dari nilai 0 hingga  $L - 1$  (misalnya, pada citra dengan kuantisasi derajat keabuan 8-bit, nilai tingkat abu-abu dari 0 hingga 255). Secara matematis histogram citra dihitung dengan rumus:

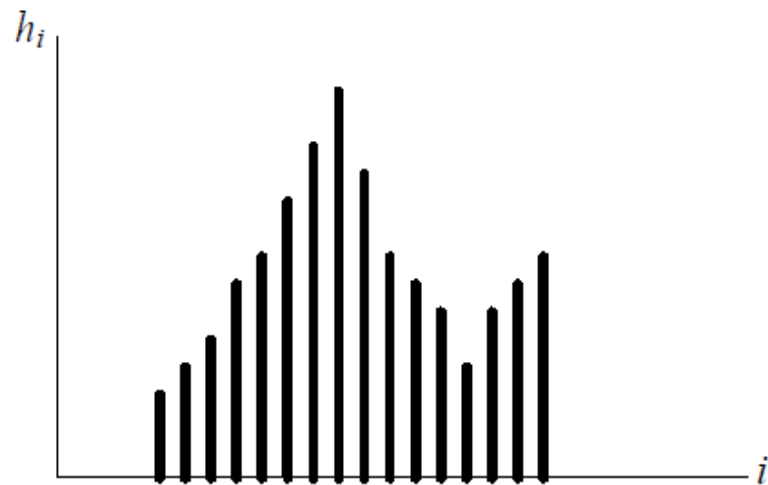
$$h_i = \frac{n_i}{n}, i = 0, 1, \dots, L-1 \quad (2.3)$$

Yang dalam hal ini :

$n_i$  = jumlah pixel yang memiliki derajat keabuan  $i$

$n$  = jumlah seluruh pixel di dalam citra

Plot  $h_i$  versus  $f_i$  disebut histogram. Gambar 2.2. merupakan contoh histogram citra. Secara grafis histogram ditampilkan dengan diagram batang. Perhatikan pada Gambar 2.2 di bawah ini nilai  $n_i$  telah dinormalisasi dengan membaginya dengan  $n$ . Nilai  $h_i$  berada pada rentang 0 sampai 1.



**Gambar 2.2.** Histogram Citra (Hermawati, F. A, 2013).

Sebagai contoh, misalkan matriks di bawah ini merepresentasikan citra digital berukuran 8 x 8 piksel dengan derajat keabuan dari 0 sampai 15 (ada 16 derajat keabuan) sebagai berikut:

$$\begin{bmatrix} 3 & 7 & 7 & 8 & 10 & 12 & 14 & 10 \\ 2 & 0 & 0 & 0 & 1 & 8 & 15 & 15 \\ 14 & 6 & 5 & 9 & 8 & 10 & 9 & 12 \\ 12 & 12 & 11 & 8 & 8 & 10 & 11 & 1 \\ 0 & 2 & 3 & 4 & 5 & 13 & 10 & 14 \\ 4 & 5 & 0 & 0 & 1 & 0 & 2 & 2 \\ 15 & 13 & 11 & 10 & 9 & 9 & 8 & 7 \\ 2 & 1 & 0 & 10 & 11 & 14 & 13 & 12 \end{bmatrix}$$

**Gambar 2.3.** Tabulasi histogram (Hermawati, F. A, 2013).

Tabulasi perhitungan histogram ditunjukkan pada Gambar 2.3. mudah untuk melihat bahwa semakin besar nilai  $n_i$ , semakin besar nilai  $h_i$ . Berikut ini adalah tabel perhitungan histogram sebagai berikut :

**Tabel 2.2.** Perhitungan Histogram (Hermawati, F. A, 2013)

$I$	$n_i$	$h_i - n_i/n (n = 64)$
0	8	0.125
1	4	0.0625
2	5	0.078125
3	2	0.03125
4	2	0.03125
5	3	0.046875
6	1	0.015625
7	3	0.046875
8	6	0.09375
9	3	0.046875
10	7	0.109375
11	4	0.0625
12	5	0.078125
13	3	0.046875
14	4	0.0625
15	3	0.046875

## 2.6. Metode K-Nearest Neighbor (KNN)

Metode *k-nearest neighbor* adalah metode yang menggunakan algoritma yang diawasi. Perbedaan antara pembelajaran terawasi dan pembelajaran tidak terawasi adalah pembelajaran terawasi bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang ada dengan data yang sudah ada.

Sedangkan pada *unsupervised learning*, data belum memiliki pola, dan tujuan *unsupervised learning* adalah untuk menemukan pola pada data. Tujuan dari metode *k-nearest neighbor* adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel pelatihan. Dimana hasil sampel uji baru diklasifikasikan berdasarkan kategori mayoritas pada *k-nearest neighbor* yang menggunakan klasifikasi tetangga sebagai nilai prediktif dari sampel uji baru. Jarak yang digunakan adalah *Euclidean Distance*.

Berikut ini adalah langkah-langkah untuk menghitung nilai k menggunakan metode *k-nearest neighbor*:

1. Tentukan parameter K (jumlah tetangga terdekat).
2. Menghitung jarak antara data yang akan dievaluasi dengan semua pelatihan jarak ini diperoleh dari data yang telah diekstraksi fitur menggunakan algoritma *local binary pattern* yaitu data pelatihan dikurangi data baru yang akan diuji yang disebut jarak *Euclidean*.
3. Kemudian mengurutkan objek-objek tersebut ke dalam kelompok-kelompok yang memiliki jarak *Euclidean* terkecil.
4. Kumpulkan kategori Y (Klasifikasi Tetangga Terdekat).
5. Dengan menggunakan kategori *Nearest Neighbor* yang merupakan mayoritas, dapat memprediksi nilai dari *query instance* yang dihitung.

## 2.7. Jarak *Euclidean* ( *Euclidean Distance* )

Jarak euclidean merupakan matriks yang paling sering digunakan untuk menghitung kemiripan 2 buah vektor. Jarak *Euclidean* menghitung akar kuadrat perbedaan antara 2 vektor (Amat, 2017).

Rumus dari *Euclidean Distance* sebagai berikut :

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}$$

Keterangan :

$d_{ij}$  = nilai / besarnya jarak

$n$  = panjang vektor

$X_{ik}$  = vektor fitur masukan

$X_{jk}$  = vektor fitur perbandingan

Semakin kecil nilai  $d_{ij}$ , semakin mirip kedua vektor tersebut dicocokkan. Sebaliknya, semakin besar nilai  $d_{ij}$ , semakin berbeda dua vektor yang dicocokkan.

## 2.8. Normalisasi Histogram

Normalisasi histogram dilakukan agar tidak terdapat perbedaan nilai yang signifikan antara fitur yang satu dengan fitur histogram yang lain. Normalisasi matematis ditunjukkan oleh Persamaan 2.4 sebagai berikut: (Retnoningrum, 2019):

$$X_{\text{baru}} = \left( \frac{X_{\text{lama}}}{\text{Sum}x} \right) \quad (2.4)$$

Keterangan :

$X_{\text{baru}}$  = nilai normalisasi

$X_{\text{lama}}$  = nilai  $x$  lama yang belum dilakukan proses normalisasi

$\text{Sum } x$  = merupakan jumlah dari nilai  $x$ .

## 2.9. Perhitungan Akurasi

Perhitungan akurasi ini digunakan untuk menentukan tingkat keberhasilan tes. Tingkat keberhasilan dapat ditentukan dengan membagi total data yang benar dengan jumlah total data yang diuji dikalikan 100. Berikut ini adalah perhitungan akurasi matematis, yaitu sebagai berikut (Retnoningrum, 2019):

$$\text{Akurasi} = \frac{\sum \text{Benar}}{\sum \text{Data Uji}} \times 100\%$$

## 2.10. Matlab

Matlab adalah program untuk analisis dan komputasi numerik yang merupakan bahasa pemrograman matematika tingkat lanjut dibangun di atas premis menggunakan sifat dan bentuk matriks. Awalnya, program ini adalah antarmuka untuk kumpulan rutinitas numerik untuk proyek LINPACK dan EISPACK dan dikembangkan menggunakan bahasa Fortran. Tapi sekarang Program ini adalah produk komersial dari perusahaan Mathworks, Inc. yang sedang dalam pengembangan lebih lanjut menggunakan bahasa C++ dan Assembler. Matlab telah berkembang menjadi lingkungan pemrograman yang



canggih dan berisi fungsi bawaan untuk melakukan pemrosesan sinyal, aljabar linier, dan perhitungan matematis lainnya. Matlab juga berisi toolbox yang berisi fungsi tambahan untuk aplikasi tertentu. Matlab adalah bahasa pemrograman tingkat tinggi berdasarkan matriks yang sering digunakan untuk teknik komputasi dan digunakan untuk memecahkan masalah yang melibatkan operasi matematika, elemen matriks, optimisasi, perkiraan, dll.). Selain itu, Matlab juga banyak digunakan untuk (Laksono dan Afrianita, 2016):

- a. Matematika dan komputasi
- b. Pengembangan dan algoritma
- c. Pemrograman pemodelan, simulasi dan pembuatan prototipe
- d. Analisa numerik dan statistik
- e. Pengembangan aplikasi teknik

### **2.11. Flowchart**

*Flowchart* adalah diagram alir yang menjelaskan logika program yang dibuat dalam bentuk simbol atau bagan yang memiliki fungsi/makna masing-masing (Fauziah, 2016). *Flowchart* dibuat pada tahap desain untuk memudahkan pemecahan masalah komputasi yang memerlukan evaluasi lebih lanjut (Zufria, 2020). Sebagai pedoman dalam membuat flowchart, ada beberapa petunjuk yang harus diperhatikan, sebagai berikut:







- a. Flowchart digambar dari atas ke bawah dan dari kiri ke kanan.
- b. Kegiatan yang dijelaskan harus didefinisikan dengan hati-hati dan definisi ini harus dapat dimengerti oleh pembaca.
- c. Kapan aktivitas dimulai dan diakhiri harus didefinisikan dengan jelas.
- d. Setiap langkah kegiatan harus dijelaskan dengan menggunakan deskripsi kata kerja, misalnya melakukan penggandaan diri.
- e. Setiap langkah kegiatan harus dalam urutan yang benar.
- f. Ruang lingkup dan jangkauan kegiatan yang sedang dijelaskan harus dieksplorasi dengan hati-hati. Cabang yang dipotong aktivitas yang digambar pada flowchart yang sama. Simbol konektor harus digunakan




dan cabang ditempatkan pada halaman terpisah atau dihilangkan seluruhnya dari cabang yang tidak terkait dengan sistem.

- g. Gunakan simbol *flowchart* yang standar.

Berikut ini adalah simbol *flowchart* yang sering digunakan adalah simbol *flowchart* standar yang dikeluarkan oleh *ANSI* dan *ISO*. Tabel 2.6 merupakan beberapa simbol *flowchart* yang sering digunakan.

**Tabel 2.3.** Simbol-Simbol Flowchart (Fauziah, 2016)

No.	Nama	Simbol	Fungsi
1.	<i>Terminator</i>		Permulaan/akhir program.
2.	Garis Alir		Arah aliran program.
3.	<i>Preparation</i>		Proses inisialisasi/pemberian harga awal.
4.	Proses		Proses perhitungan/proses pengolahan data.
5.	<i>Input/Output data</i>		Proses <i>input/output</i> data, parameter, informasi.
6.	<i>Predefined process</i> (sub program)		Permulaan sub program/proses menjalankan sub program.
7.	<i>Decision</i>		Perbandingan pernyataan

			penyeleksian data yang memberikan pilihan untuk langkah selanjutnya.
8.	<i>On page connector</i>		Penghubung bagian-bagian <i>flowchart</i> yang berada pada satu halaman.
9.	<i>Off page connector</i>		Penghubung bagian-bagian <i>flowchart</i> yang berada pada halaman berbeda.

## 2.12. Penelitian Terkait

Berikut ini adalah Tabel 2.4. penelitian terlebih dahulu yaitu sebagai berikut

No	Judul dan Tahun	Penulis	Ringkasan
1.	Ekstraksi ciri pada telapak tangan dengan metode <i>local binary pattern</i> (LBP). 2019.	Dwi Retnoningrum, Agus Wahyu Widodo, Muh. Arif Rahman.	Penelitian ini dilakukan karena telapak tangan memiliki fitur unik yang berbeda pada setiap individu. Selain fitur unik, luas permukaan telapak tangan menjadi salah satu pertimbangan penulis dalam menentukan objek penelitian. Luas permukaan telapak tangan lebih besar jika dibandingkan dengan luas permukaan salah satu jari. Salah satu metode yang dapat dimanfaatkan dalam proses identifikasi adalah metode ekstraksi ciri LBP

			<p>(<i>Local Binary Pattern</i>) yang menerapkan jarak ketetanggaan dan jumlah tetangga yang dibandingkan. Dimulai dengan tahapan Pre-processing yaitu tahap persiapan citra berwarna yang akan dirubah menjadi citra keabuan kemudian dilanjutkan dengan proses regioning atau proses pembagian citra menjadi beberapa region. Dilanjutkan dengan tahap ekstraksi ciri dengan metode LBP. Hasil akurasi tertinggi yang diperoleh dari penelitian ini sebesar 92,31% dengan jarak ketetanggaan bernilai 2, jumlah tetangga yang dibandingkan = 8, jumlah region sebesar 16 dan jumlah pembagian height = 4 dan width = 4.</p>
2.	<p>Penerapan algoritma <i>local binary pattern</i> untuk pengenalan pola sidik jari. 2017</p>	<p>Nurul Hayary, Martaleli Bettiza, Eko Imam Pratama.</p>	<p>Penelitian ini menggunakan penerapan algoritma <i>Local Binary Pattern</i> (LBP) dan algoritma Manhattan Distance telah berhasil dibangun dengan bahasa pemrograman java. Dari pengujian yang dilakukan terhadap sistem yang dibangun terbukti dengan</p>

			<p>pelatihan dan pengujiannya. Dimana akurasi didapat dari data uji dengan pola sidik jari menggunakan mendapat akurasi 61.54%. Tebal atau tipisnya tinta yang digunakan saat akuisisi citra mempengaruhi nilai akurasi sistem.</p>
3.	<p>Implementasi metode <i>local binary pattern</i> untuk pengenalan pola huruf hiragana dan katakana pada smartphone 2017</p>	<p>Rabiuldien Amat, Jayanti Yusmah Sari, Ika Purwanti Ningrum.</p>	<p>Penelitian ini tentang Pengenalan Pola Huruf Hiragana dan Katakana dengan metode <i>Local Binary Pattern</i> dan memiliki keakuratan pengenalan yang cukup baik yaitu sebesar 81,1%, akan tetapi dalam mengenali perbedaan pola hurufnya kurang signifikan. Akurasi tersebut juga dipengaruhi oleh adanya perbedaan karakteristik huruf yang diinputkan oleh user berupa ukuran dan rotasi. Hal ini menunjukkan bahwa aplikasi ini kurang mampu mengatasi adanya perbedaan ukuran dan rotasi huruf. Kelebihan dari LBP yaitu mudah diimplementasikan dan merupakan metode ekstraksi fitur yang proses komputasi yang rendah.</p>

Penelitian ini dilakukan untuk menerapkan algoritma *local binary pattern* dan metode *k-nearest neighbor* (k-nn) dalam mendeteksi penyakit pada daun mangga, penelitian ini menggunakan format .bmp. dan juga menggunakan histogram untuk menggambarkan penyebaran nilai intensitas piksel suatu gambar. Kemudian gunakan jarak *Euclidean* untuk menghitung kemiripan 2 vektor pada daun mangga. Penelitian ini juga diharapkan mendapatkan hasil yang baik dalam mendeteksi penyakit pada daun mangga.

**BAB III**  
**METODE PENELITIAN**

**3.1. Tempat dan Waktu Penelitian**

**3.1.1. Tempat Penelitian**

Tempat dan waktu penelitian ini dilaksanakan di Perumahan Taman Permata Jalan Utama II Desa kolam, Sumatera Utara.

**3.1.2. Waktu Dan Jadwal Pelaksanaan Penelitian**

Waktu dan jadwal pelaksanaan penelitian ini dilaksanakan pada semester genap tahun ajaran 2020/2021 dengan tabel sebagai berikut ini:

**Tabel 3.1.** Waktu dan jadwal pelaksanaan penelitian

No	Kegiatan	Bulan 2021			
		Mei	Juni	Juli	Agustus
1.	Tahap persiapan penelitian				
	a. Penyusunan dan pengajuan judul				
	b. Pengajuan proposal				
	c. Perijinan penelitian				
2.	Tahap Pelaksanaan				
	a. Pengumpulan data				
	b. Analisis data				
3.	Tahap penyusunan laporan				



## **3.2. Bahan dan Alat Penelitian**

### **3.2.1. Perangkat Keras**

Perangkat keras yang digunakan dalam pengembangan sistem ini diperlukan sebagai berikut:

1. Laptop

Merek : Del

Type : Intel® Core(TM) i3 CPU M 350 @ 2.27GHz 2.26 GHz

RAM : 1.00 GB

Processor : System type 32-bit Operating System.

2. Smartphone

Merk : Vivo

Type : Y93

Kamera : 13 MP

CPU : 1,95 GHz Snapdragon 439 Octa-core

RAM : 3.00 GB

Internal Memory : 32 GB

### **3.2.2. Perangkat Lunak**

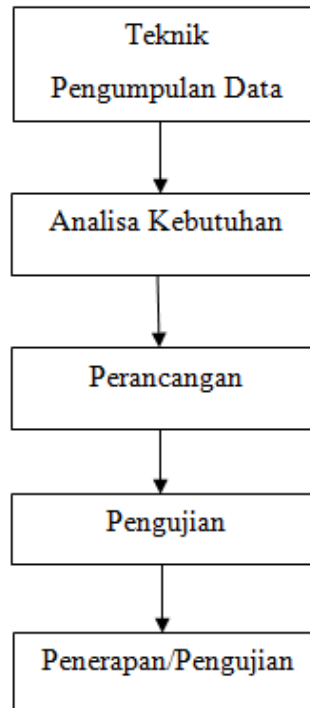
Perangkat lunak yang digunakan dalam pengembangan sistem ini diperlukan sebagai berikut:

1. Operating System Windows 7.0 Pro 32 bit.

2. *Matlab R2015b*

## **3.3. Prosedur Kerja**

Prosedur kerja dalam penelitian ini dilakukan melalui tahapan sebagai berikut:



**Gambar 3.1.** Tahap-Tahap Prosedur Kerja

### 3.3.1. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Library Research* (Penelitian Pustaka).

Penelitian ini dilakukan oleh penulis dengan mencari jurnal dan ebook, mempelajari dan mengumpulkan referensi dan landasan teori yang diambil dari berbagai artikel dan jurnal di internet. Penulis mencari beberapa buku yang terdapat di perpustakaan online IPUSDA, Sumatera Utara, Medan dan perpustakaan lainnya. Berikut ini adalah buku terkait penelitian yang dilakukan oleh penulis yaitu buku yang berjudul Pemrograman Matlab, 2016, Heru Dibyo Laksono, MT dan Reri Afrianita, MT, Hama dan Penyakit Tanaman (Mengenali dan Mengatasinya), 2017, Argohartono Arie Raharjo, , Logika dan Algoritma Aplikatif (dengan C++, C# & Java), 2016, Fauziah, S.Kom., MMSI Pengolahan citra digital, 2010, Darma Putra, 2010. Pengolahan Citra Digital, Yogyakarta: CV Andi Offset., Hermawati, F. A.,

2013. Pengolahan Citra Digital, Yogyakarta: CV. Andi Offset. Sulistyanti, S.R., dkk. 2016. Pengolahan Citra Dasar dan Contoh Penerapannya, Yogyakarta: Andi Offset.

## 2. Studi *Literatural*

Studi kepustakaan adalah serangkaian kegiatan yang berkaitan dengan metode pengumpulan daftar pustaka, membaca dan mencatat, serta mengolah bahan penelitian atau menemukan referensi teori yang relevan dengan kasus atau masalah yang ditemukan.

### 3.3.2. Analisis Kebutuhan

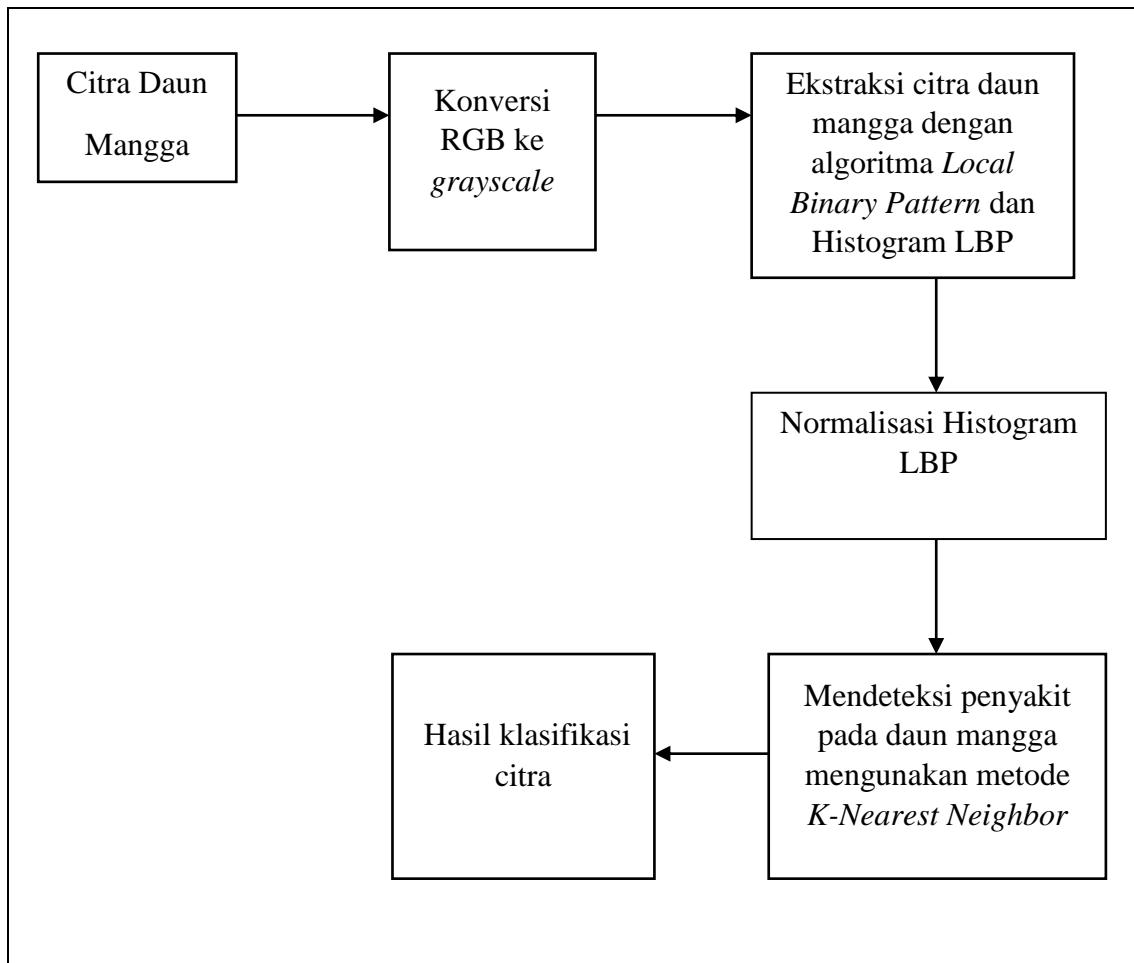
Tahapan analisis kebutuhan merupakan tahapan proses untuk mendapatkan segala kebutuhan informasi digunakan untuk mengidentifikasi masalah yang muncul dalam suatu penelitian untuk membangun suatu sistem guna mendukungnya berjalan dengan baik. Persyaratan ini meliputi perangkat keras dan perangkat lunak.

Pendeteksian penyakit pada daun mangga menggunakan algoritma *local binary pattern* dan metode *k-nearest neighbor* dipilih karena sebelumnya belum ada yang meneliti/mendeteksi penyakit psada daun mangga menggunakan algoritma *local binary pattern* dan metode *k-nearest neighbor*.

### 3.3.3. Perancangan

Desain adalah langkah pertama dalam fase pengembangan rekayasa produk atau sistem dimana proses penerapan berbagai teknik dan prinsip yang bertujuan untuk mendefinisikan perangkat, proses atau sistem yang memungkinkan realisasi fisik. This phase is the technical core of the software engineering process.

Diagram perencanaan algoritma *local binary pattern* dan metode *k-nearest neighbor* adalah sebagai berikut:



**Gambar 3.2.** Diagram perencanaan mendeteksi penyakit pada daun mangga

Gambar diatas merupakan sistem atau jalur pendeteksian penyakit pada daun mangga menggunakan algoritma *local binary pattern* dan metode *k-nearest neighbor*. Langkah pertama adalah menginput citra daun mangga yang sakit, kemudian mengubahnya menjadi citra *grayscale*, kemudian mengekstrak citra daun mangga yang sakit menggunakan algoritma *local binary pattern* dan muncul lah hasil histogram, kemudian melakukan normalisasi histogram. Selanjutnya untuk mendeteksi jenis penyakit pada daun mangga menggunakan metode *k-nearest neighbor*, maka didapatkan hasil citra penyakit pada daun mangga tersebut.

#### **3.3.4. Pengujian**

Pengujian sistem dimaksudkan untuk mengetahui apakah sistem mendeteksi penyakit pada daun mangga berjalan dengan baik atau tidak. Folder data untuk identifikasi citra dibagi menjadi data latih dan data uji. Pengujian dilakukan pada citra daun mangga menggunakan format \*bitmap.

#### **3.3.5. Penerapan/ Penggunaan**

Aplikasi/penggunaan sistem ini adalah untuk dapat mendeteksi penyakit pada daun mangga. Dengan menggunakan pengenalan bentuk digital dari objek daun dapat dilakukan pengolahan citra digital, sehingga memungkinkan mesin atau komputer untuk mengenali citra penyakit pada daun mangga seperti penglihatan manusia dan dapat menentukan langkah pengobatan selanjutnya.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Pembahasan

Adapun Beberapa tahapan yang akan dibahas dalam penelitian ini yaitu analisis data, representasi data, hasil analisis data dan perancangan sebagai berikut.

##### 4.1.1. Analisis Data

Pada tahapan analisis, data yang dibutuhkan dalam penelitian ini adalah berupa citra daun mangga, kemudian diolah menggunakan pengolahan citra yaitu menggunakan algoritma *local binary pattern* dan metode *k-nearest neighbor* untuk mendeteksi penyakit pada daun mangga. Pada tahap analisis data dilakukan penentuan citra masukan dan citra keluaran serta perancangan tampilan. Kemudian mengimplementasikan aplikasi menggunakan algoritma *local binary pattern* dan metode *k-nearest neighbor* untuk mendeteksi penyakit pada daun mangga menggunakan Matlab 2015b. Ada beberapa langkah dalam program ini, yaitu:

- a. Menginput citra daun mangga.
- b. Citra daun mangga yang di input diubah menjadi citra *grayscale*.
- c. Citra uji *local binary pattern* dan histogram *local binary pattern*.
- d. Normalisasi histogram *local binary pattern*,
- e. Kemudian mendeteksi jenis penyakit terhadap citra daun mangga menggunakan metode *k-nearest neighbor*.

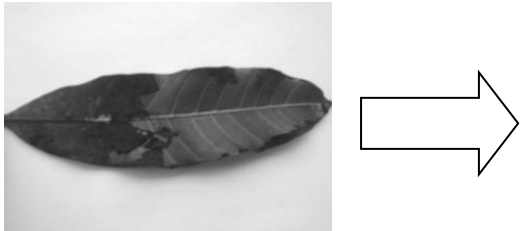
##### 4.1.2. Representasi Data

Adapun dataset yang digunakan untuk keperluan hitungan manual adalah citra daun mangga sebanyak 2 citra yang diklasifikasi dengan jenis

penyakit cendawan jelaga (A), penyakit ulat daun mangga (B), yang digunakan sebagai variabel data latih dengan ukuran pixel 3x3 pixel.

Berikut ini adalah dataset berupa citra daun mangga pengujian klasifikasi penyakit pada daun mangga menggunakan 40 citra daun yaitu citra data latih 24 dan 16 citra data uji. Pengujian klasifikasi penyakit pada daun mangga menggunakan 2 jenis penyakit (Furqan, 2020). Data yang digunakan berupa citra daun mangga dan akan dilakukan proses penerapan *local binary pattern* untuk mendeteksi penyakit pada daun mangga, untuk pengujian sampel penulis menggunakan citra daun mangga dengan ukuran dari 3x3 piksel. Gambar 4.1. merupakan contoh nilai intensitas piksel dari citra grayscale dengan ukuran citra 3 x 3 piksel.

**Tabel 4.1.** Matriks Citra



247	247	247
0	0	73
7	7	8

**Gambar 4.1.** Citra *Grayscale*

Contoh gambar diatas adalah gambar daun mangga yang memiliki nilai pada setiap pikselnya, gambar tersebut bertipe *grayscale* yang memiliki rentang warna 0 – 255. Contoh gambar terdiri dari 3 baris dan 3 kolom yang akan digunakan untuk melakukan proses aplikasi penerapan *local binary pattern* untuk mendeteksi penyakit pada daun mangga.

#### 4.1.3. Hasil Analisis Data

Dalam proses penerapan metode *thresholding* menggunakan algoritma *local binary pattern* dalam mengekstraksi citra daun mangga, penulis

menggunakan sampel citra grayscale dengan ukuran gambar 3 x 3 piksel dengan kedalaman 8 bit dengan menggunakan format \*.bmp sebagai berikut:

Langkah-langkah dalam menentukan nilai *local binary pattern* adalah sebagai berikut:

1. Jika nilai  $T$  lebih tinggi dari nilai pixel maka bernilai 1.
2. Jika nilai  $T$  lebih rendah dari nilai pixel maka bernilai 0.

Berikut ini adalah langkah dan perhitungan local binary pattern sebagai berikut:

- a. Citra daun1



**Gambar 4.2.** Citra Grayscale

Dengan metode *threshold* ini maka diketahui nilai  $T$  adalah 82

**Tabel 4.2.** Perhitungan LBP

Matriks Diagonal 3x3			Hasil Threshold			Bobot LBP			Hasil Kali Treshold dan Bobot LBP			Nilai Local Binary Pattern
7	247	164	0	1	1	8	16	32	0	16	32	LBP = 4 + 16 + 32 + 64 + 128 = 244
82	82	82	1		1	4		64	4		64	
7	7	164	0	0	1	2	1	128	0	0	128	



## b. Citra daun2



**Gambar 4.3.** Citra *Grayscale*

Dengan metode *threshold* ini maka diketahui nilai  $T$  adalah 247

**Tabel 4.3.** Perhitungan LBP

Matriks Diagonal 3x3	Hasil Threshold	Bobot LBP	Hasil Kali Treshold dan Bobot LBP	Nilai Local Binary Pattern
247 7 7	1 0 0	8 0 0	8 0 0	LBP = 8 + 64 = 72
91 247 247	0 1	0 64	0 64	
155 164 164	0 0 0	0 0 0	0 0 0	

Selanjutnya langkah-langkah proses pendeteksian penyakit pada citra daun mangga dilakukan dengan menghitung jarak *Euclidean* dari nilai ekstraksi ciri *local binary pattern* menggunakan metode *k-nearest neighbor*. sebagai berikut:

## 1. Dataset

Tabel 4.7. merupakan tabel dataset dari 4 penyakit pada daun mangga. Dari 2 penyakit daun mangga tersebut akan diklasifikasi menggunakan metode *k-nearest neighbor* yang terdiri dari penyakit hama kutu putih dan penyakit ulat daun.

**Tabel 4.4.** Dataset

No	Dataset	Citra Hasil Ekstraksi LBP
1	Daun1	244
2	Daun2	72

1. Tentukan jumlah penyakit daun mangga yang akan dibentuk.
  - a. Penyakit cendawan jelaga
  - b. Penyakit ulat daun
2. Tetapkan nilai  $k$  yaitu sebagai berikut:
 

$K= 1$

$K= 3$
3. Alokasikan semua data/objek ke jarak terdekat. Berikut adalah hasil pengalokasian data dengan jarak terdekat. Hasil jarak terdekat didapatkan dari perhitungan dengan rumus :

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}$$

$$\begin{aligned} \text{Jarak 1} &= \sqrt{(40 - 244)^2} \\ &= \sqrt{(-204)^2} \\ &= \sqrt{41616} \\ &= 204 \end{aligned}$$

$$\begin{aligned} \text{Jarak 2} &= \sqrt{(40 - 72)^2} \\ &= \sqrt{(-32)^2} \\ &= \sqrt{1024} \\ &= 32 \end{aligned}$$

Setelah menghitung jarak *Euclidean*, hasilnya adalah sebagai berikut:

**Tabel 4.5.** Hasil Perhitungan Jarak *Eucludien*

No	Dataset	Jarak Terdekat
1	Daun1	204
2	Daun2	32

Berdasarkan hasil deteksi didapatkan  $K = 3$  atau 3 tetangga dengan jarak terdekat yaitu penyakit daun mangga B dengan deteksi hama Ulat Daun Mangga, sehingga data citra daun mangga yang diolah dimasukkan pada jenis penyakit “Ulat Daun Mangga”.

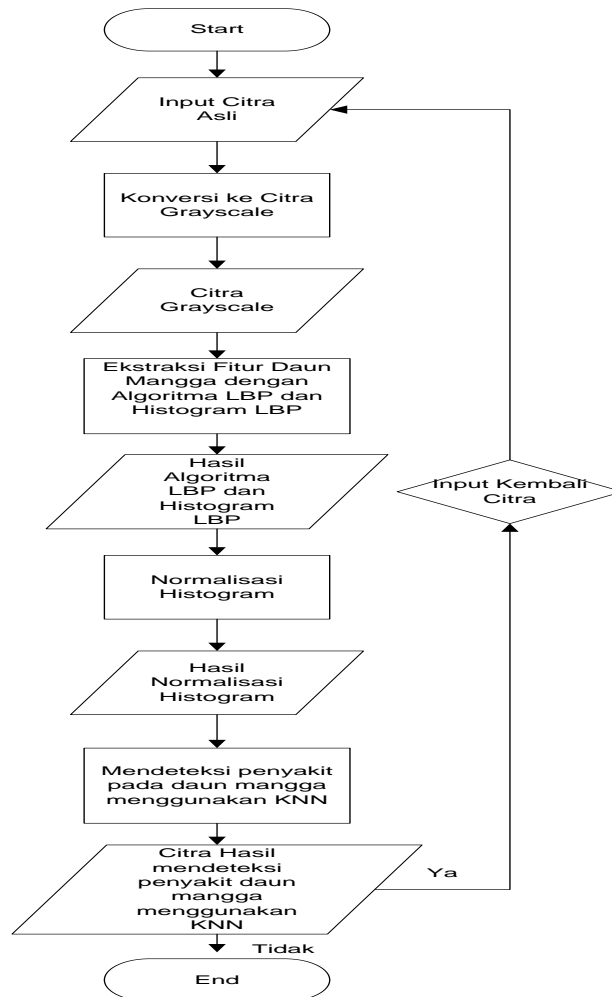
#### 4.1.4. Perancangan Sistem

Dalam penelitian ini, untuk merancang sistem yang digunakan sebagai gambaran umum dari sistem yang akan dibangun, Berikut ini adalah tampilan rancangan sistem pengujian penerapan *local binary pattern* dan *k-nearest neighbor* untuk mendeteksi penyakit pada daun mangga, sebagai berikut :

##### 1. *Flowchart*

Flowchart adalah bagan atau gambar yang menunjukkan alur sebuah proses dan hubungan suatu program. Diagram alir diperlukan untuk menjelaskan alur program yang dibuat dalam bentuk grafik agar orang lain dapat memahami alur yang telah dibuat. Desain diagram alir yang akan digunakan untuk mendeteksi penyakit pada daun mangga ditunjukkan pada gambar di bawah ini:

##### a. Perancangan *Flowchart* Sistem



**Gambar 4.4.** *Flowchart* Sistem

Gambar di atas adalah diagram alir dari *flowchart* sistem dalam mengimplementasikan *local binary pattern* dan *k-nearest neighbor* dalam mendeteksi penyakit pada daun mangga. Dimulai dari menginput citra asli, kemudian mengubahnya menjadi citra *grayscale*, kemudian dilakukan proses ekstraksi menggunakan *local binary pattern*, kemudian muncul histogram citra *local binary pattern*, kemudian histogram *local binary pattern*

dinormalisasi. Setelah itu, deteksi penyakit pada daun mangga menggunakan metode *k-nearest neighbor*, maka proses metode *k-nearest neighbor* selesai.

b. *Flowchart k- nearest neighbor.*

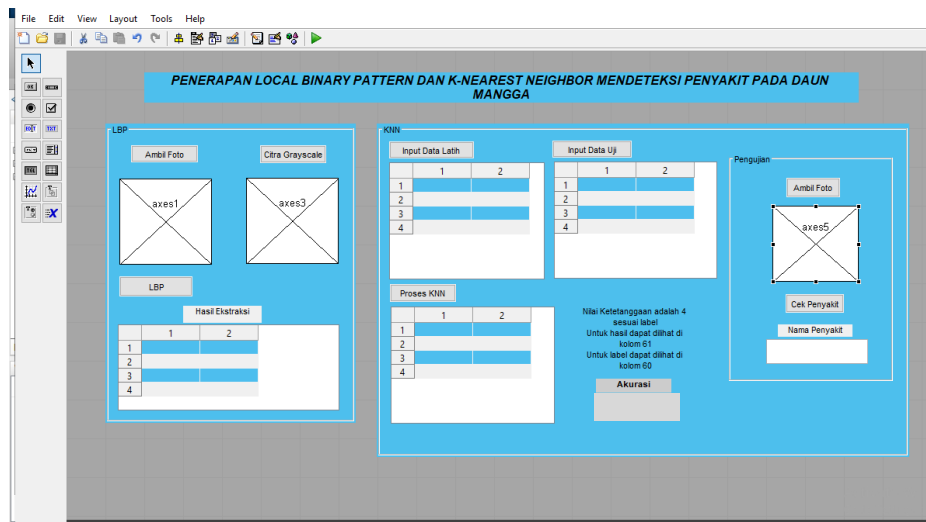


**Gambar 4.5.** *Flowchart K- Nearest Neighbor.*

## 2. Perancangan Antar Muka

Sistem ini dirancang menggunakan pemrograman Matlab. Perancangan ini bertujuan untuk memudahkan pemakai (user) dalam menggunakan sistem yang telah dibuat. desain *interface* ini akan mempengaruhi spesifikasi komputer yang digunakan, agar dapat berjalan dengan baik, spesifikasi *hardware* harus sesuai. Bentuk perancangan antarmuka ini digunakan untuk menginput citra daun

mangga yang akan diolah menggunakan *local binary pattern* dan *k-nearest neighbor*.



**Gambar 4.6.** Perancangan Antar Muka

## 4.2. Hasil

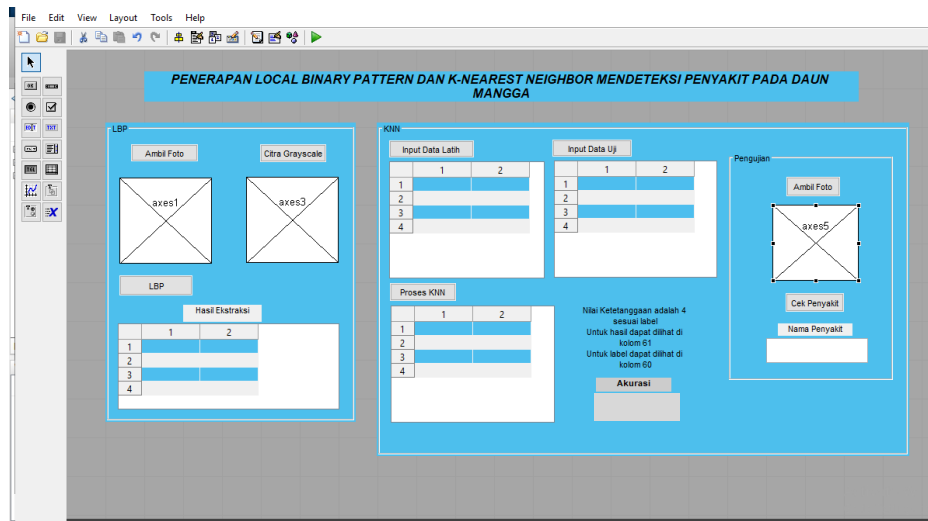
Ada beberapa tahapan yang akan dibahas mengenai hasil yang diperoleh dalam penelitian ini yaitu pengujian dan aplikasi sebagai berikut.

### 4.2.1. Pengujian

Setelah merancang dan membuat sistem, selanjutnya dilakukan pengujian. Pengujian bertujuan untuk melihat sejauh mana sistem yang telah dibangun sesuai dengan yang diharapkan, contoh hasil penerapan *local binary pattern* dan *k-nearest neighbor* mendeteksi cendawan jelaga, ulat daun, kutu putih dan hama thrips mangga pada daun mangga dapat dilihat sebagai berikut ini :

#### 1. Tampilan awal aplikasi

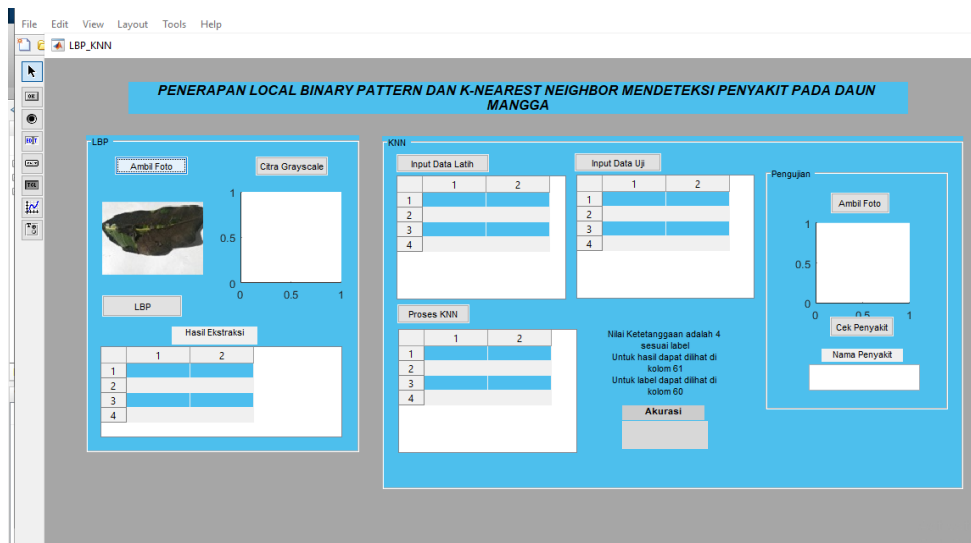
Tampilan dibawah ini adalah tampilan aplikasi penerapan *local binary pattern* dan *k-nearest neighbor* mendeteksi penyakit pada daun mangga



Gambar 4.7. Tampilan Aplikasi

### 1. Tampilan input citra

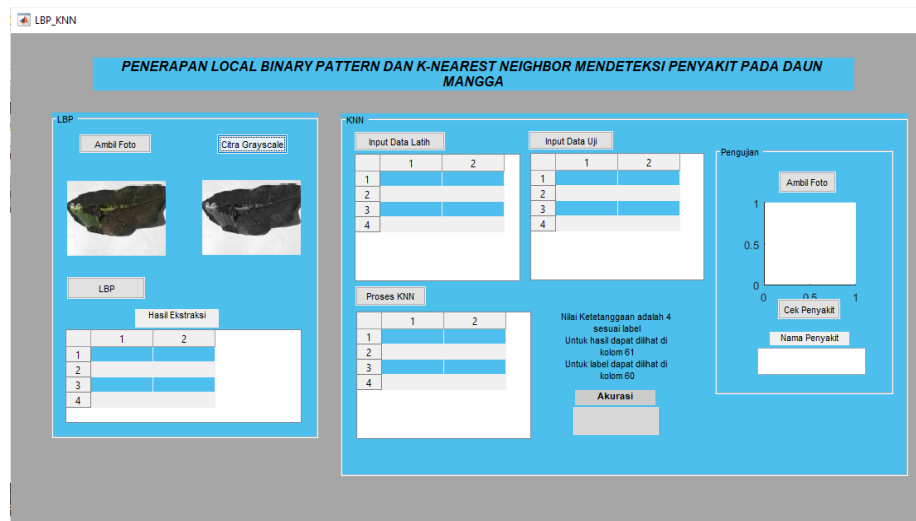
Setelah itu, klik button ambil gambar untuk mengambil gambar citra daun mangga, setelah itu klik open, seperti gambar dibawah ini:



Gambar 4.8. Tampilan Input Citra

### 2. Tampilan citra asli setelah di ubah ke citra grayscale

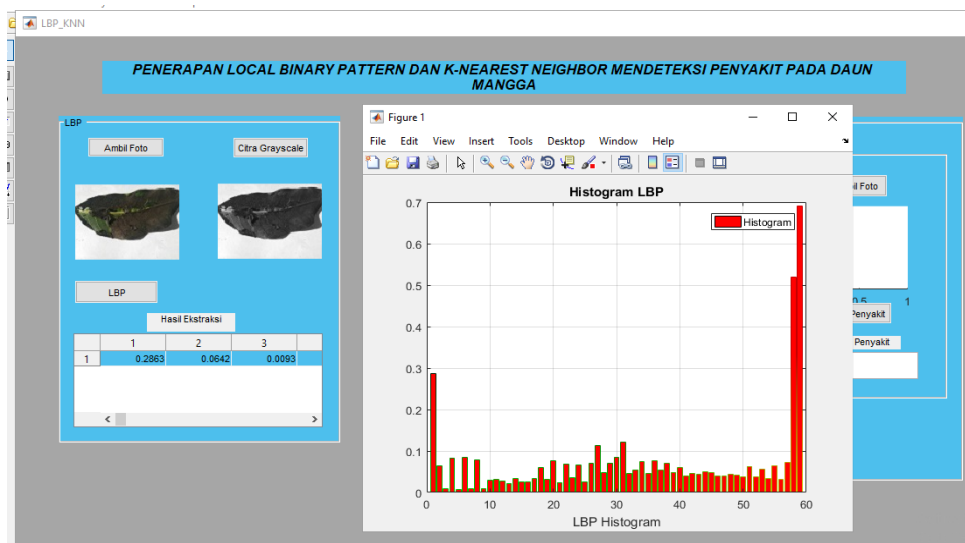
Setelah mengklik button citra grayscale, citra akan otomatis berubah menjadi citra grayscale, seperti gambar dibawah ini:



**Gambar 4.9.** Tampilan Citra *Grayscale*

3. Tampilan nilai ekstraksi *local binary pattern* dan histogram lbp

Setelah gambar daun mangga berubah menjadi citra *grayscale*, berikutnya klik *button* citra *local binary pattern*, hasilnya seperti gambar dibawah ini:

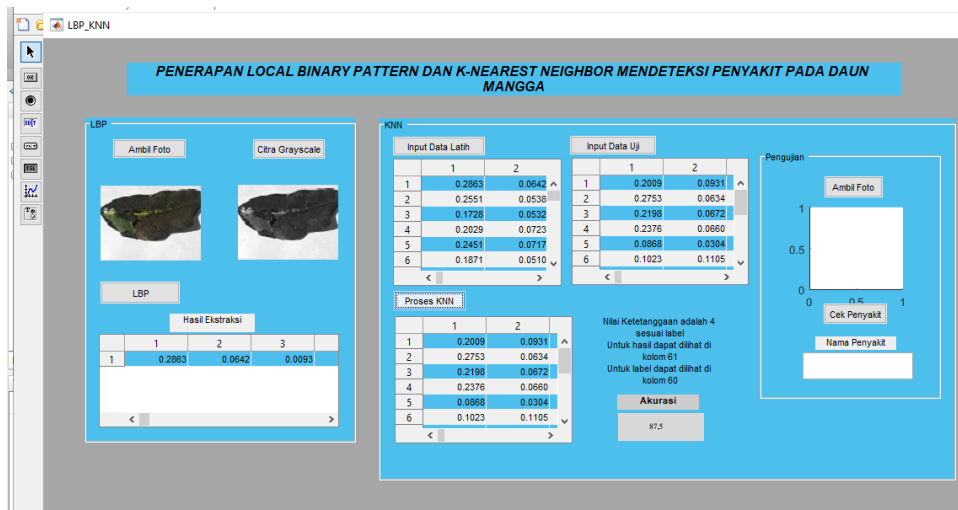


**Gambar 4.10.** Tampilan nilai ekstraksi dan histogram *local binary pattern*



4. Tampilan hasil perhitungan data uji, data latih dan k-nn

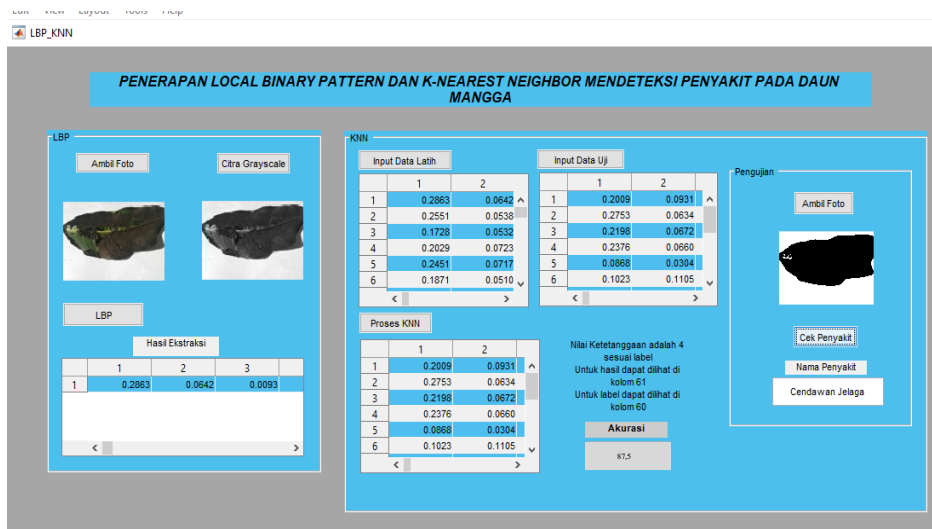
Setelah citra *local binary pattern* muncul, klik *button* data uji dan data latih untuk melihat hasil perhitungan citra daun mangga, seperti gambar dibawah ini:



**Gambar 4.11.** Hasil perhitungan data uji, data latih dan k-nn

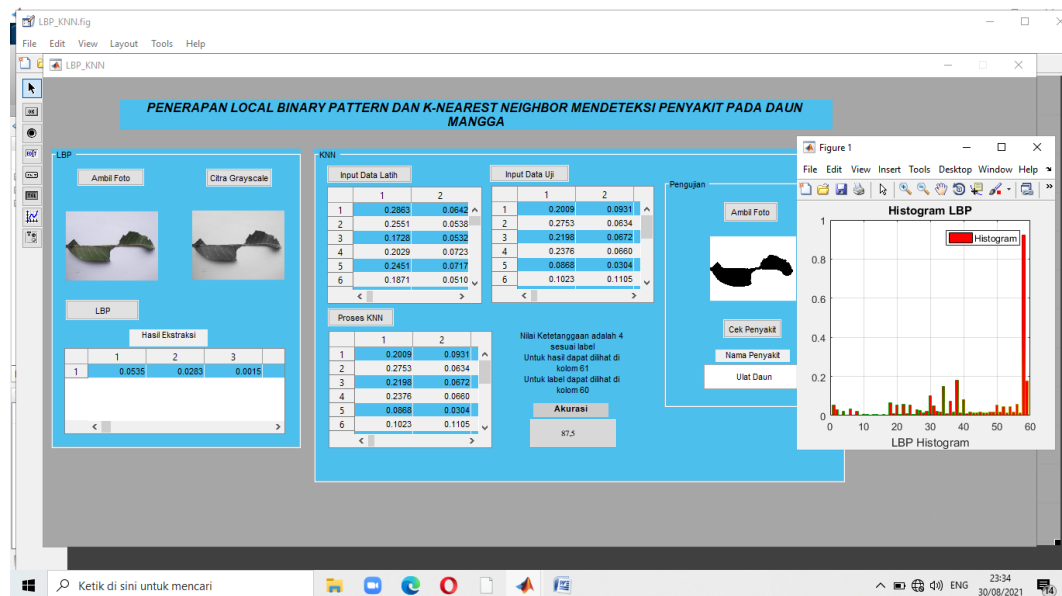
5. Tampilan deteksi penyakit daun mangga menggunakan metode *k-nearest neighbor*.

klik *button* hasil untuk mendeteksi jenis penyakit terhadap daun mangga yaitu seperti gambar dibawah ini:



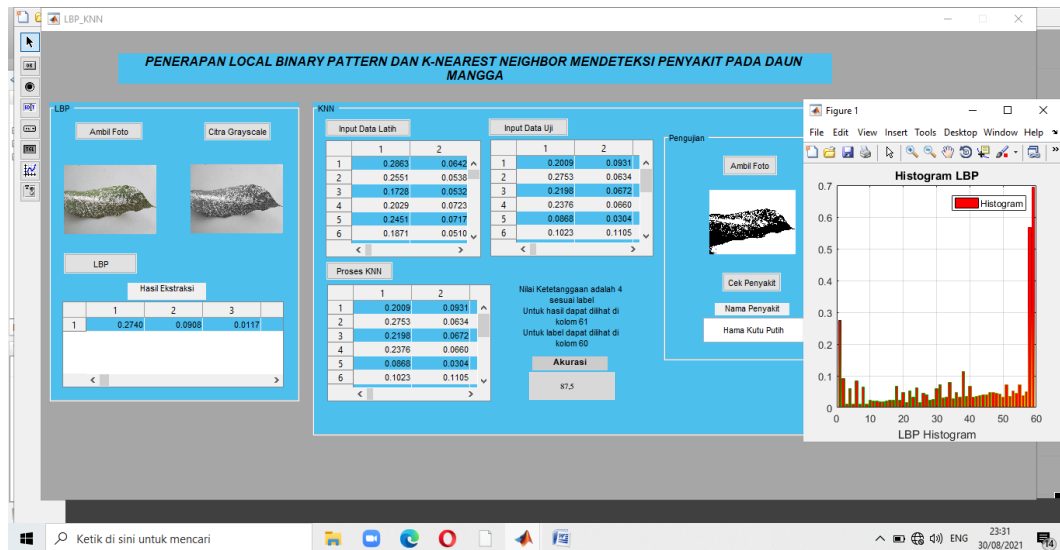
**Gambar 4.12.** Tampilan deteksi penyakit daun mangga menggunakan metode *k-nearest neighbor*

Selanjutnya adalah hasil proses dalam melakukan deteksi penyakit ulat daun, hama kutu putih dan hama trips mangga pada daun mangga berdasarkan citra daun mangga yaitu sebagai berikut:



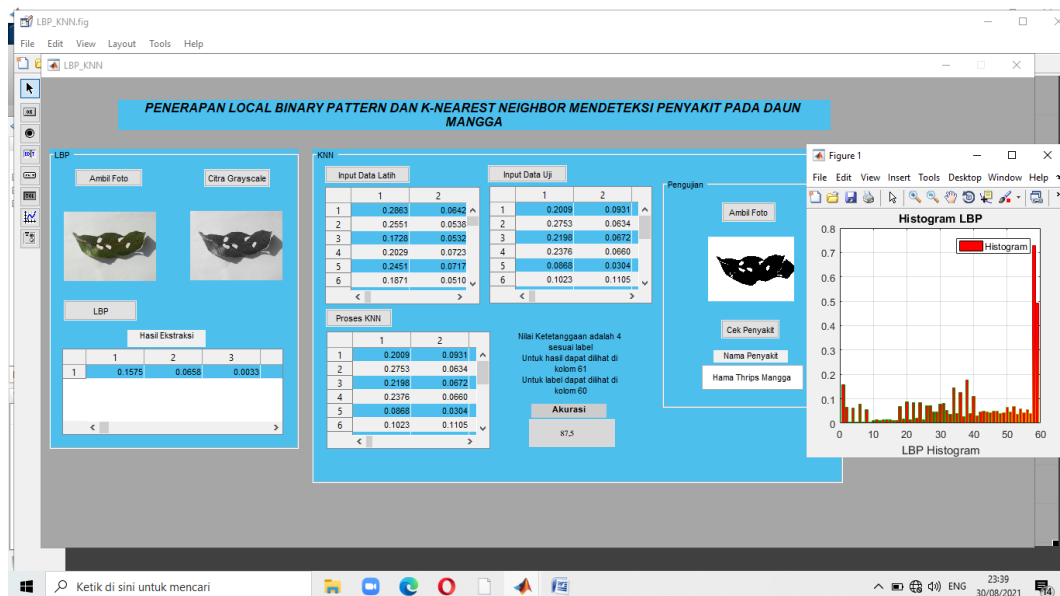
**Gambar 4.13.** Hasil deteksi penyakit ulat daun

Bedasarkan pada gambar diatas, proses deteksi penyakit pada daun mangga berhasil diterapkan pada sistem aplikasi, dimana dengan citra uji daun mangga berhasil dideteksi dengan penyakit "ulat daun", sehingga hasil deteksi sistem benar.



**Gambar 4.14.** Hasil deteksi penyakit hama kutu putih

Bedasarkan pada gambar diatas, proses deteksi penyakit pada daun mangga berhasil diterapkan pada sistem aplikasi, dimana dengan citra uji daun mangga berhasil dideteksi dengan penyakit “hama kutu putih”, sehingga hasil deteksi sistem benar.



**Gambar 4.15.** Hasil deteksi penyakit hama thrips mangga

Bedasarkan pada gambar diatas, proses deteksi penyakit pada daun mangga berhasil diterapkan pada sistem aplikasi, dimana dengan citra uji daun mangga berhasil dideteksi dengan penyakit “hama thrips mangga”, sehingga hasil deteksi sistem benar.

Berikut ini adalah tabel Tabel 4.6. hasil deteksi penyakit pada daun mangga menggunakan pengujian sistem dapat dilihat sebagai berikut :

**Tabel 4.6.** Pengujian Sistem

No.	Pengujian Sistem	Sesuai/Tidak Sesuai
1	Perancangan Antar Muka	Sesuai
2	Tampilan Aplikasi	Sesuai
3	Tampilan Input Citra	Sesuai
4	Tampilan Citra <i>Grayscale</i>	Sesuai
5	Tampilan Nilai Ekstraksi LBP dan Histogram LBP	Sesuai
6	Tampilan Hasil Perhitungan Data Latih dan Data Uji	Sesuai
7	Tampilan Hasil Perhitungan K-NN	Sesuai
8	Tampilan Deteksi Penyakit Cendawan Jelaga Pada Daun Mangga Menggunakan Metode K-NN	Sesuai
9	Tampilan Deteksi Penyakit Ulat Daun Pada Daun Mangga Menggunakan Metode K-NN	Sesuai
10	Tampilan Deteksi Penyakit Hama Kutu Putih Pada Daun Mangga Menggunakan Metode K-NN	Sesuai
11	Tampilan Deteksi Penyakit Hama Thrips Pada Daun Mangga Menggunakan Metode K-NN	Sesuai

## 6. Hasil Pengujian

Bedasarkan dari hasil pengujian klasifikasi menggunakan pengujian sistem dengan 16 citra uji yang terdiri dari 4 jenis penyakit pada daun mangga didapati hasil sebagai berikut:

**Tabel 4.7.** Hasil Uji Sistem

No.	Nama Citra	Jenis Penyakit	Hasil Deteksi Penyakit	Keterangan
1	Ind1	Cendawan Jelaga	Cendawan Jelaga	Benar
2	Ind2	Cendawan Jelaga	Cendawan Jelaga	Benar
3	Ind3	Cendawan Jelaga	Hama Trips Mangga	Salah
4	Ind4	Cendawan Jelaga	Cendawan Jelaga	Benar
5	2nd1	Ulat Daun	Ulat Daun	Benar
6	2nd2	Ulat Daun	Ulat Daun	Benar
7	2nd3	Ulat Daun	Ulat Daun	Benar
8	2nd4	Ulat Daun	Ulat Daun	Benar
9	3nd1	Hama Kutu Putih	Hama Kutu Putih	Benar
10	3nd2	Hama Kutu Putih	Hama Kutu Putih	Benar

11	3nd3	Hama Kutu Putih	Cendawan Jelaga	Salah
12	3nd4	Hama Kutu Putih	Hama Kutu Putih	Benar
13	4nd1	Hama Thrips Mangga	Hama Thrips Mangga	Benar
14	4nd2	Hama Thrips Mangga	Hama Thrips Mangga	Benar
15	4nd3	Hama Thrips Mangga	Hama Thrips Mangga	Benar
16	4nd4	Hama Thrips Mangga	Hama Thrips Mangga	Benar

Bedasarkan pada tabel hasil pengujian di atas, didapati citra uji yang berhasil dideteksi dengan benar dan citra uji yang dideteksi dengan salah. Adapun selanjutnya menghitung tingkat akurasi berdasarkan citra uji yang dipakai. Adapun rumusnya sebagai berikut:

$$\text{Akurasi} = \frac{14}{16} \times 100\% = 87,5\%$$

Bedasarkan dari hasil uji akurasi, didapatkan nilai akurasi sebesar 87,5% untuk proses deteksi penyakit pada daun mangga berdasarkan citra daun mangga sebanyak 16 data.

#### **4.2.2. Penerapan**

Penggunaan sistem ini digunakan untuk mengetahui penyakit pada daun mangga. Dengan menggunakan aplikasi *local binary pattern* dan *k-nearest neighbor* untuk mendeteksi penyakit pada daun mangga, berguna untuk membantu siswa dan siswa melihat penyakit apa saja yang menyerang daun mangga. Penerapan sistem ini dapat diterapkan oleh pelajar dan mahasiswa yang ingin mengetahui penyakit pada daun mangga. Dimana sistem ini akan memudahkan mahasiswa dan pelajar dalam menganalisa penyakit pada daun mangga dapat diketahui dengan cara yang mudah dan dapat meminimalisir kesalahan dalam mendeteksi penyakit pada daun mangga.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Berdasarkan penelitian yang dilakukan penulis dalam Penerapan *local binary pattern* dan *k-nearest neighbors* mendeteksi penyakit pada daun mangga dapat disimpulkan sebagai berikut.

1. Dari hasil penerapan *local binary pattern* dan *k-nearest neighbors* mendeteksi penyakit pada daun mangga dapat dilakukan dalam melakukan proses Ekstraksi berdasarkan tekstur dari citra yang akan digunakan pada proses knn.
2. Dalam hal ini *local binary pattern* berperan mengeluarkan nilai yang terkandung dalam citra, kemudian nilai tersebut diproses menggunakan metode *k-nearest neighbors* sehingga dapat mendeteksi penyakit pada daun.

#### 5.2. Saran

Adapun saran penulis dalam penelitian ini sebagai berikut.

1. Penulis menyadari banyak kesalahan dalam penulisan maupun pembuatan program aplikasi.
2. Perlu ditambahkan data uji dan data latih untuk mendapatkan hasil deteksi pada daun mangga berupa akurasi yang lebih optimal.
3. Penulis menyadari dalam pengambilan dataset daun mangga masih kurang akurat, sehingga penulis menyarankan agar pada penelitian selanjutnya pengambilan dataset pada daun mangga lebih akurat lagi.



## DAFTAR PUSTAKA

- Amat, R., Sari, J. Y., Nigrum, I. K., 2017, Implementasi Metode Local Binary Patterns Untuk Pengenalan Pola Huruf Hiragana dan Katakana Pada Smartphone. *Jurnal Ilmiah teknologi Informasi*, **Vol. 15**, Nomor 2, Juli 2017, hlm. 162-172.
- Fauziah., 2016. Aplikatif Logika dan Algoritma dengan C++, C# dan Java. Yogyakarta: Teknosain.
- Furqon, M., Sriani., Harahap, L.S., 2020, Klasifikasi Daun Bungenvil Menggunakan Gray Level Co-Occurrence Matrix dan K-Nearest Neighbor. *Jurnal CorelIT*, Vol.6.
- Hayaty, N., Bettiza, M., Pratama, E. I., 2017, Penerapan Algoritma Local Binary Pattern untuk Pengenalan Pola Sidik Jari. *Jurnal Hasil Penelitian dan Industri Terapan*, **Vol. 06**, No. 02, Oktober 2017, hlm. 74-79. <http://simki.unpkediri.ac.id/mahasiswa/fileartikel/2019/14.1.03.02.0121.pdf> f. Diakses 15 oktober 2017.
- Hermawati, F. A., 2013. Pengolahan Citra Digital. Yogyakarta: Andi Offset
- Laksono, H. D., dan Afrianita, Reri., 2016. Pemograman Matlab. Yogyakarta: Teknosain.
- Liantoni, F., 2015. Deteksi Tepi Citra Daun Mangga Menggunakan Algoritma Ant Colony Optimization. Seminar Nasional Sains dan Teknologi Terapan III 2015 Institut Teknologi Adhi Tama Surabaya.
- Muwardi, F., Fadlil, A., 2017. Sistem Pengenalan Bunga Berbasis Pengolahan Citra Dan Klasifikasi Jarak. *Jurnal Ilmu Teknik Elektro Komputer dan Informatika (JITEKI)*, **Vol. 3**, No.2, Desember 2017.
- Prabowo, D. A., Abdullah, D., Manik, A., 2018, Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking. *Jurnal Pseudocode*, **Vol. 5**, No. 2, September 2018.

- Purwati, R., Ariyanto, G., Pengenalan Wajah Manusia Berbasis Algoritma Local Binary Pattern. *Jurnal Emitor*, **Vol. 17**, No.02. <http://journals.ums.ac.id/index.php/emitor/article/download/>.
- Putra, D., 2010. Pengolahan Citra Digital. Yogyakarta: Andi Offset
- Raharjo, A. A., 2017. Hama dan Penyakit Tanaman. Depok: PT. Trubus Swadaya
- Retnoningrum, D., Widodo, A. W., Rahman, M. A., 2019, Ekstraksi Ciri Telapak Tangan Dengan Metode Local Binary Pattern (LBP). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, **Vol. 3**, Maret 2019, hlm. 2611-2618.
- Rahayu,P.A.,H., Pawening, E. R., Klasifikasi Jenis Mangga Berdasarkan Bentuk dan Tekstur Daun Menggunakan Metode K-Nearest Neighbor. *Jurnal Prosiding SENTIA 2016-Politeknik Negeri Malang*, **Vol. 08**, 2016.
- Sulistiyanti, S. R., dkk., 2016. Pengolahan Citra Dasar dan Contoh Penerapannya. Yogyakarta: Andi Offset
- Wijaya, N.,Ridwan, A.,2019, Klasifikasi Jenis Buah Apel Dengan Metode K-Nearest Neighbors. *Jurnal SISFOKOM*, **Vol. 08**, Nomor 01, Maret 2019.
- Zufria, I., Furqan, M., 2020, Pemanfaatan Machine Learning Dengan Algoritma Support Vector Machine, Polynomial Regression Dan Bayesian Ridge regression Untuk Memprediksi Kasus covid-19 Di indonesia. *Jurnal Laporan penelitian Program Studi*, 04 November 2020.
- Zufria, I., Hasugian, H, A.,2018, Perancangan Sistem Restorasi Citra Dengan Metode Image Inpainting. *Jurnal Ilmu Komputer dan Informatika*, **Vol.03**, Nomor 01, November 2018.
- Zuliyanti, A., Nurmuharani, D., 2017. Hama dan Penyakit Dominan Tanaman Mangga (*Mangifera Indical I.*). Makalah. <http://repository.usu.ac.id/handle/123456789/66585.pdf>. Diakses 2017

## LAMPIRAN-LAMPIRAN

### LAMPIRAN 1

Hasil uji citra daun mangga pada Penerapan *Local Binary Pattern dan K-Nearest Neighbor* Mendeteksi Penyakit Pada Daun Mangga sebagai berikut :

No.	Nama Citra	Jenis Penyakit	Hasil Deteksi Penyakit	Keterangan
1	Ind1	Cendawan Jelaga	Cendawan Jelaga	Benar
2	Ind2	Cendawan Jelaga	Cendawan Jelaga	Benar
3	Ind3	Cendawan Jelaga	Hama Trips Mangga	Salah
4	Ind4	Cendawan Jelaga	Cendawan Jelaga	Benar
5	2nd1	Ulat Daun	Ulat Daun	Benar
6	2nd2	Ulat Daun	Ulat Daun	Benar
7	2nd3	Ulat Daun	Ulat Daun	Benar
8	2nd4	Ulat Daun	Ulat Daun	Benar
9	3nd1	Hama Kutu Putih	Hama Kutu Putih	Benar
10	3nd2	Hama Kutu Putih	Hama Kutu Putih	Benar

11	3nd3	Hama Kutu Putih	Cendawan Jelaga	Salah
12	3nd4	Hama Kutu Putih	Hama Kutu Putih	Benar
13	4nd1	Hama Thrips Mangga	Hama Thrips Mangga	Benar
14	4nd2	Hama Thrips Mangga	Hama Thrips Mangga	Benar
15	4nd3	Hama Thrips Mangga	Hama Thrips Mangga	Benar
16	4nd4	Hama Thrips Mangga	Hama Thrips Mangga	Benar

## LAMPIRAN 2

Kode program matlab untuk penerapan *local binary pattern* dan *k-nearest neighbor* mendeteksi penyakit pada daun mangga:

```
function varargout = LBP_KNN(varargin)
% LBP_KNN MATLAB code for LBP_KNN.fig
%   LBP_KNN, by itself, creates a new LBP_KNN or raises the
existing
%   singleton*.
%
%   H = LBP_KNN returns the handle to a new LBP_KNN or the
handle to
%   the existing singleton*.
%
%   LBP_KNN('CALLBACK',hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in LBP_KNN.M with the given input
arguments.
%
%   LBP_KNN('Property','Value',...) creates a new LBP_KNN or
raises the
%   existing singleton*. Starting from the left, property
value pairs are
%   applied to the GUI before LBP_KNN_OpeningFcn gets called.
An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to LBP_KNN_OpeningFcn via
varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help LBP_KNN

% Last Modified by GUIDE v2.5 21-Aug-2021 04:05:58

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @LBP_KNN_OpeningFcn, ...
                  'gui_OutputFcn',  @LBP_KNN_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```

        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end
% End initialization code - DO NOT EDIT

% --- Executes just before LBP_KNN is made visible.
function LBP_KNN_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to LBP_KNN (see VARARGIN)

% Choose default command line output for LBP_KNN
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
movegui(hObject, 'center');

% UIWAIT makes LBP_KNN wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = LBP_KNN_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in train.
function train_Callback(hObject, eventdata, handles)
% hObject    handle to train (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.output = hObject;
[file,path] = uigetfile({'*.xls;*.xlsx','Excel Files'}, 'DATA
TRAIN');
filename = strcat(path,file);
data = xlsread(filename);
set(handles.uitable1, 'Data', data)

% --- Executes on button press in testing.

```

```

function testing_Callback(hObject, eventdata, handles)
% hObject      handle to testing (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
handles.output = hObject;
[file,path] = uigetfile({'*.xls;*.xlsx','Excel Files'},'DATA
TEST');
filename = strcat(path,file);
data = xlsread(filename);
set(handles.uitable2,'Data',data)

% --- Executes on button press in knn.
function knn_Callback(hObject, eventdata, handles)
% hObject      handle to knn (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
x=xlsread('DATA TRAIN.xlsx');
latih=x;
group=latih(:,60);
latih = [latih(:,1) latih(:,2) latih(:,3) latih(:,4) latih(:,5)
latih(:,6) latih(:,7) latih(:,8) latih(:,9) latih(:,10)
latih(:,11) latih(:,12) latih(:,13) latih(:,14) latih(:,15)
latih(:,16) latih(:,17) latih(:,18) latih(:,19) latih(:,20)
latih(:,21) latih(:,22) latih(:,23) latih(:,24) latih(:,25)
latih(:,26) latih(:,27) latih(:,28) latih(:,29) latih(:,30)
latih(:,31) latih(:,32) latih(:,33) latih(:,34) latih(:,35)
latih(:,36) latih(:,37) latih(:,38) latih(:,39) latih(:,40)
latih(:,41) latih(:,42) latih(:,43) latih(:,44) latih(:,45)
latih(:,46) latih(:,47) latih(:,48) latih(:,49) latih(:,50)
latih(:,51) latih(:,52) latih(:,53) latih(:,54) latih(:,55)
latih(:,56) latih(:,57) latih(:,58) latih(:,59)];

for i = 1 : 16
    y=xlsread('DATA TEST.xlsx');
    sampel = y;
    test = [sampel(:,1) sampel(:,2) sampel(:,3) sampel(:,4)
sampel(:,5) sampel(:,6) sampel(:,7) sampel(:,8) sampel(:,9)
sampel(:,10) sampel(:,11) sampel(:,12) sampel(:,13) sampel(:,14)
sampel(:,15) sampel(:,16) sampel(:,17) sampel(:,18) sampel(:,19)
sampel(:,20) sampel(:,21) sampel(:,22) sampel(:,23) sampel(:,24)
sampel(:,25) sampel(:,26) sampel(:,27) sampel(:,28) sampel(:,29)
sampel(:,30) sampel(:,31) sampel(:,32) sampel(:,33) sampel(:,34)
sampel(:,35) sampel(:,36) sampel(:,37) sampel(:,38) sampel(:,39)
sampel(:,40) sampel(:,41) sampel(:,42) sampel(:,43) sampel(:,44)
sampel(:,45) sampel(:,46) sampel(:,47) sampel(:,48) sampel(:,49)
sampel(:,50) sampel(:,51) sampel(:,52) sampel(:,53) sampel(:,54)
sampel(:,55) sampel(:,56) sampel(:,57) sampel(:,58) sampel(:,59)];
    hasil=knnclassify(test,latih,group);
end
hasil = [sampel(:,1) sampel(:,2) sampel(:,3) sampel(:,4)
sampel(:,5) sampel(:,6) sampel(:,7) sampel(:,8) sampel(:,9)
sampel(:,10) sampel(:,11) sampel(:,12) sampel(:,13) sampel(:,14)
sampel(:,15) sampel(:,16) sampel(:,17) sampel(:,18) sampel(:,19)
sampel(:,20) sampel(:,21) sampel(:,22) sampel(:,23) sampel(:,24)
sampel(:,25) sampel(:,26) sampel(:,27) sampel(:,28) sampel(:,29)

```

```

sampel(:,30) sampel(:,31) sampel(:,32) sampel(:,33) sampel(:,34)
sampel(:,35) sampel(:,36) sampel(:,37) sampel(:,38) sampel(:,39)
sampel(:,40) sampel(:,41) sampel(:,42) sampel(:,43) sampel(:,44)
sampel(:,45) sampel(:,46) sampel(:,47) sampel(:,48) sampel(:,49)
sampel(:,50) sampel(:,51) sampel(:,52) sampel(:,53) sampel(:,54)
sampel(:,55) sampel(:,56) sampel(:,57) sampel(:,58) sampel(:,59)
sampel(:,60) hasil];
set(handles.uitable4,'Data',hasil)

set(handles.edit1,'String','75%')

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in ambilfoto.
function ambilfoto_Callback(hObject, eventdata, handles)
% hObject    handle to ambilfoto (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename,pathname] = uigetfile({'*.jpg*'});
    Img = imread(fullfile(pathname,filename));
    axes(handles.axes1)
    cla('reset')
    imshow(Img)
handles.Img = Img;
guidata(hObject,handles);

```



```

% --- Executes on button press in rgb.
function rgb_Callback(hObject, eventdata, handles)
% hObject      handle to rgb (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
Img = handles.Img;
Img = rgb2gray(Img);
axes(handles.axes3)
cla('reset')

% --- Executes on button press in lbp.
function lbp_Callback(hObject, eventdata, handles)
% hObject      handle to lbp (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
Img = handles.Img2;
Img = rgb2gray(Img);
LbpImg = extractLBPFeatures(Img, 'Upright', true);
figure
b = bar(LbpImg);
b(1).FaceColor = 'r';
grid on
title('Histogram LBP')
xlabel('LBP Histogram')

% --- Executes on button press in inputuji.
function inputuji_Callback(hObject, eventdata, handles)
% hObject      handle to inputuji (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile({'*.jpg*'});
Img = imread(fullfile(pathname, filename));
axes(handles.axes4)
cla('reset')
imshow(Img)
handles.Img3 = Img;

% --- Executes on button press in uji.
function uji_Callback(hObject, eventdata, handles)
% hObject      handle to uji (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
Img3=handles.Img3;
Img3=rgb2gray(Img3);
LbpImg = extractLBPFeatures(Img3, 'Upright', false);
lbp = mean(LbpImg);

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)

```

---

```

% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename,pathname] = uigetfile({'*.jpg*'});
    Img = imread(fullfile(pathname,filename));
    axes(handles.axes5)
    cla('reset')
    imshow(Img)

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
I = handles.Img3;
J = I(:,:,1);
K = im2bw(J,.6);
L = imcomplement(K);
str = strel('disk',5);
M = imclose(L,str);
N = imfill(M,'holes');
O = bwareaopen(N,1000);
stats = regionprops(O,'Area','Perimeter','Eccentricity');
area = stats.Area;
perimeter = stats.Perimeter;
metric = 4*pi*area/(perimeter^2);
eccentricity = stats.Eccentricity;

    input = [metric;eccentricity];

    load net1
    output = round(sim(net,input));

    axes(handles.axes5)
    imshow(K)

    if output == 1
        kelas = 'Cendawan Jelaga';
    elseif output == 2
        kelas = 'Di gigit Ulat';
    elseif output == 3
        kelas = 'Hama Kutu Putih';
    elseif output == 4
        kelas = 'Hama Trips Mangga';
    end

    set(handles.edit4,'String',kelas)

```

```

function edit4_Callback(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of
edit4 as a double


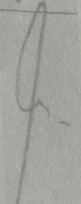

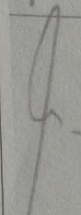
% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called


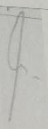

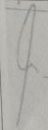

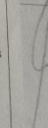
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


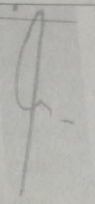

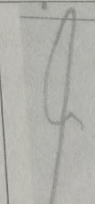

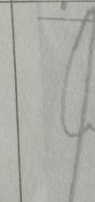
```

**KARTU BIMBINGAN SKRIPSI**  
**PROGRAM STUDI ILMU KOMPUTER**  
**FAKULTAS SAINS DAN TEKNOLOGI**

Nama : Intan Dwi Nanda	Pembimbing I : Ilka Zufria, M.Kom.
Nim : 0701163097	Pembimbing II : Heri Santoso, M.Kom
Prog. Studi : Ilmu Komputer	SK Pembimbing :
<b>Judul Skripsi:</b> <b>“ Penerapan <i>Local Binary Pattern</i> Dan <i>K-Nearest Neighbor</i> Mendeteksi Penyakit Pada Daun Mangga”</b>	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	22-06-2020	1. Bab I Pendahuluan (Rumusan Masalah, Manfaat Penelitian, Batasan Masalah, Tujuan Penelitian)		10-06-2020	1. Bab I pendahuluan	
II	14-07-2020	1. Bab II Tinjauan Pustaka (Tabel Jenis Penyakit)		12-06-2020	1. Bab II Tinjauan Pustaka	

III	4-08-2020	1. Bab Tinjauan Pustaka (Penelitian Terkait)	II		15-06-2020	1. Bab III Flowchart Sistem	
VI	12-08-2020	1. Bab Metode Penelitian (Tabel waktu & jadwal penelitian)	III		17-06-2020	1. Pengecekan Proposal Skripsi	
V	14-08-2020	1. Bab III Metode Penelitian (Flowchart, Tahapan perancangan)			17-06-2020	1. ACC Proposal Skripsi	

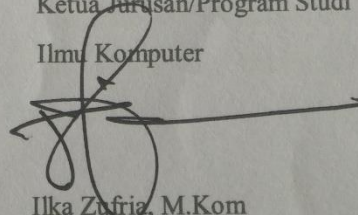
VI	27-08-2021	<ol style="list-style-type: none"> <li>1. Perubahan Judul</li> <li>2. Daftar Gambar</li> <li>3. Daftar Tabel</li> <li>4. Bab 4 Hasil dan Pembahasan</li> <li>5. Daftar Pustaka</li> <li>6. Flowchart kmn</li> <li>7. Tabel Penyakit Tabel Pengujian Sistem</li> </ol>		23-08-2021	1. Bab 4 Analisis	
VII	30-08-2021	<ol style="list-style-type: none"> <li>1. Bab 3 dan 4 Format Tabel dan Gambar</li> <li>2. Daftar Tabel dan Daftar Gambar</li> </ol>		24-08-2021	<ol style="list-style-type: none"> <li>1. Bab 5</li> <li>2. Kesimpulan</li> </ol>	
VIII	01-09-2021	Acc Sidang Munaqasah		27-08-2021	Acc Sidang Munaqasah	

Medan, 20 Desember 2021

An. Dekan

Ketua Jurusan/Program Studi

Ilmu Komputer



Ilka Zufri, M.Kom

NIP. 198506042015031006

**DAFTAR RIWAYAT HIDUP**  
**(*CURICULUM VITAE*)**



Nama : Intan Dwi Nanda  
Nim : 0701163097  
Tempat Tanggal Lahir : Peladangan, 4 Juni 1998  
Alamat : Perumahan Taman Permata Blok D.11 Desa Kolam  
Kelurahan : Desa Kolam  
Kecamatan : Percut Sei Tuan  
Kabupaten : Deli Serdang  
Agama : Islam  
Status Menikah : Belum Menikah  
No Hp : 08887762202  
Nama Orang Tua  
    Ayah : M. Nawawi Lubis  
    Ibu : Endang Tri Wahyuni Sitorus  
Pendidikan Formal  
    2003 – 2009 : SDN 097342 Pondok 32 Bandar Betsy  
    2009 – 2013 : SMP PAB 2 Helvetia  
    2013 – 2015 : SMK BM Apipsu Medan  
    2016 – 2020 : Universitas Islam Negeri Sumatera Utara