

**GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) UNTUK
DETEKSI PENYAKIT PADA TANAMAN TERONG
MENGUNAKAN METODE K-NEAREST
NEIGHBOR (KNN)**

Oleh

I GUSTI MADE TERESNA

T3117003

SKRIPSI

**Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar sarjana**



**PROGRAM SARJANA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2022**

HALAMAN PENGESAHAN

GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) UNTUK DETEKSI PENYAKIT PADA TANAMAN TERONG MENGUNAKAN METODE K-NEAREST NEIGHBOR (KNN)

**(Studi Kasus : Desa Boidu, Kecamatan Bulango Utara,
Kabupaten Bone Bolango, Gorontalo)**

Oleh

I GUSTI MADE TERESNA

T3117003

SKRIPSI

Untuk memenuhi salah satu syarat ujian guna memperoleh gelar sarjana Program
Studi Teknik Informatika, ini telah di setujui oleh Tim pembimbing.

Gorontalo, 01 September 2022

Pembimbing I



Zohrahayaty, S.Kom, M.Kom
NIDN. 0912117702

Pembimbing II



Irma Surya Kumala Idris, S.Kom, M.Kom
NIDN. 0921128801

PERSETUJUAN SKRIPSI

GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) UNTUK DETEKSI PENYAKIT PADA TANAMAN TERONG MENGUNAKAN METODE K-NEAREST NEIGHBOR (KNN)

Oleh

I GUSTI MADE TERESNA

T3117003

SKRIPSI

Di Periksa Oleh Panitia Ujian Strata Satu (S1)

Universitas Ichsan Gorontalo

Gorontalo, 01 September 2022

1. Ketua Penguji
Asmaul Husna, M.Kom
2. Anggota
Husdi, M.kom
3. Anggota
Zulfrianto Y. Lamasigi, M.Kom
4. Anggota
Zohrahayaty, M.Kom
5. Anggota
Irma Surya Kumala Idris, M.Kom



Mengetahui

Dekan Fakultas Ilmu Komputer


Jorry Karim, S.Kom, M.Kom
NIDN.0918077302

Ketua Program Studi


Sudirman S. Panna, M.Kom
NIDN.9909912852

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis saya (Skripsi) ini adalah asli dan belum pernah di ajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan Tinggi lainnya.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam karya tulis ini tidak dapat karya atau pendapat yang telah di publikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan tidak kebenaran dalam pernyataan ini, maka saya tidak bersedia menerima sanksi akademikberupapencabutan gelar yang telah di peroleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi ini.

Gorontalo, 30 Agustus, 2022

Yang Membuat Pernyataan,




IGUSTI MADE TERESNA

ABSTRACT

I GUSTI MADE TERESNA. T3170003. GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) FOR DISEASES DETECTION IN EGGPLANTS USING K-NEAREST NEIGHBOR (KNN) METHOD

This study aims at detecting diseases in eggplants. This study employs the Gray Level Co-occurrence Matrix (GLCM) method for leaf feature extraction and K-Nearest Neighbor (KNN) for eggplant disease classification/identification. The GLCM (Gray-Level Co-occurrence Matrix) method functions to analyze digital images, while K-Nearest Neighbor (KNN) classifies diseases in eggplants. This study has succeeded in collecting 104 test data. They are divided into 94 training data and 10 test data. The detection of eggplant disease in this study is divided into the classification of healthy plants, rotten fruit, bacterial wilt, anthracnose, leaf spot, seedlings, stem rot, and powdery mildew. The image data are processed into a grayscale image where the color texture is detected. Furthermore, after getting the image from the classification, it will be processed using Gray Level Co-occurrence Matrix (GLCM) and assisted with the detection of eggplant disease classification using K-Nearest Neighbor (KNN). The results of this study indicate that the method used is successful in detecting disease in eggplant plants. The percentage shows a 70% accuracy rate with a distance of 1. The value of $K=1$ $GLCM=0^{\circ}$. The percentage shows a 70% level of accuracy with a distance of 1. The value of $K=1$ $GLCM=135^{\circ}$. The more training data provided, the greater the possibility of the accuracy of the calculation of the training data. It means that GLCM and the KNN algorithm can detect diseases in eggplants.

Keywords: *eggplant disease detection, Gray Level Co-Occurrence Matrix, K-Nearest Neighbor*

ABSTRAK

I GUSTI MADE TERESNA. T3117003. GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) UNTUK DETEKSI PENYAKIT PADA TANAMAN TERONG MENGGUNAKAN METODE K-NEAREST NEIGHBOR (KNN)

Tujuan penelitian ini adalah mendeteksi penyakit pada tanaman terong. Penelitian ini digunakan metode *Gray Level Co-occurrence Matrix* (GLCM) untuk ekstraksi fitur daun dan *K-Nearest Neighbor* (KNN) untuk pengklasifikasian/identifikasi penyakit terong. Metode GLCM (*Gray-Level Cooccurrence Matrix*), yang digunakan untuk menganalisa citra digital sedangkan *K-Nearest Neighbor* (KNN) untuk mengklasifikasi penyakit pada tanaman terong. Adapun di dalam penelitian ini peneliti telah berhasil mengumpulkan 104 data test dimana terbagi atas 94 data training dan 10 data test. Adapun deteksi penyakit terong dalam penelitian ini dibagi ke dalam klasifikasi tanaman sehat, buah busuk, layu bakteri, antraknosa, bercak daun, rebai semai, busuk pangkal batang, dan jamur tepung. Adapun data citra tersebut telah diolah menjadi citra *grayscale* dimana kemudian akan dideteksi tekstur warna. Selanjutnya setelah mendapatkan citra dari klasifikasi tersebut maka akan diproses menggunakan *Gray Level Co-occurrence Matrix* (GLCM) dan dibantu deteksi klasifikasi penyakit terong dengan menggunakan *K-Nearest Neighbor* (KNN). Hasil penelitian ini menunjukkan bahwa metode yang digunakan berhasil dalam mendeteksi penyakit pada tanaman terong. Adapun persentase menunjukkan 70% tingkat akurasi dengan jarak 1 dan nilai $K=1$ GLCM=0°, serta persentase menunjukkan 70% tingkat akurasi dengan jarak 1 dan nilai $K=1$ GLCM=135°. Semakin banyak data latih yang diberikan maka akan semakin besar kemungkinan ketepatan perhitungan data latih. Dengan demikian penerapan GLCM dan algoritma KNN dapat mendeteksi penyakit pada tanaman terong.

Kata kunci: deteksi penyakit terong, *Gray Level Co-Occurrence Matrix*, *K-Nearest Neighbor*

KATA PENGANTAR

Puji tuhan atas rahmat dan karunia yang telah di limpahkan, akhirnya penulis dapat menyelesaikan penelitian yang berjudul **“GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) UNTUK DETEKSI PENYAKIT PADA TANAMAN TERONG MENGGUNAKAN METODE K-NEAREST NEIGHBOR (KNN)”** untuk memenuhi salah satu persyaratan penyusunan Skripsi Program Studi Teknik Informatika Fakultas ilmu Komputer Universitas Ikhsan Gorontalo.

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

1. Ibu Dr. Hj. Juriko Abdussamad, Selaku Ketua Yayasan Pengembangan Ilmu pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo.
2. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Bapak Jorry Karim, S.Kom, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Melangi, S.Kom, M.Kom, selaku wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Ibu Irma Surya Kumala Idris, S.Kom, M.Kom, selaku wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Sudirman S. Panna, S.Kom, M.Kom, selaku Ketua jurusan Teknik Informatika fakultas Ilmu Komputer universitas ichsan Gorontalo;
7. Ibu Zohrahayaty, S.Kom, M.Kom, selaku Pembimbing I;
8. Ibu Irma Surya Kumala Idris, S.Kom, M.Kom, Selaku Pembimbing II;
9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmun kepada penulis;
11. Kedua Orang Tua saya tercinta, atas segala kasih sayang, jerih payah dan do'a

restunya dalam membesarkan dan mendidik penulis;

12. Kepada semua pihak yang ikut membantu dalam penyelesaian proposal ini yang tak sempat penulis sebutkan satu-persatu.

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

Gorontalo, September 2022

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN	i
PERSETUJUAN SKRIPSI	ii
HALAMAN PERNYATAAN	iii
ABSTRACT	iv
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	3
1.3 Rumusan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.5.1 Manfaat Teoritis	4
1.5.2 Manfaat Praktis.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1. Tinjauan Studi.....	5
2.2.Tinjauan Pustaka.....	7
2.2.1 Deteksi	7
2.2.2 Penyakit Tanaman	7
2.2.3 Tanaman Terong.....	7
2.2.4 Penyakit Tanaman Terong.....	8
2.2.5 Pengolahan Citra Digital.....	15
2.2.6 Jenis Citra Digital	15
2.2.7 Proses Grayscale.....	17
2.2.8 Gray Level Co-occurrence Matrix.....	17
2.2.9 Metode K- Nearest Neighbor (K-NN).....	21

2.2.10 Penerapan Metode <i>K- Nearest Neighbor</i> (K-NN)	22
2.2.11 Confusion Matrix.....	23
2.2.12 Bahasa Pemograman Python	24
2.3 Kerangka Pikir	25
BAB III METODE PENELITIAN	26
3.1 Jenis Metode, Subjek, Objek, Waktu dan Lokasi Penelitian	26
3.2 Pengumpulan Data	26
3.3 Pemodelan.....	27
3.3.1 Pra Pengolahan Data.....	28
3.3.2 Ekstraksi Ciri	28
3.3.3 Data Training	28
3.3.4 Data Testing.....	29
3.3.5 Training Menggunakan KNN.....	29
3.3.6 Model Deteksi.....	29
3.3.7 Hasil Deteksi.....	30
3.3.8 Evaluasi	30
BAB IV HASIL PENELITIAN	30
4.1 Hasil Pengumpulan Data	31
4.1.1 Histogram Equalisation	33
4.2 Hasil Pemodelan	35
4.2.1 Pra Pengolahan	35
4.2.2 Ekstraksi Fitur	38
4.2.3 Klasifikasi.....	47
4.3 Evaluasi.....	56
4.3.1 Confusion Matrix	56
BAB V PEMBAHASAN	61
5.1 Pembahasan Model.....	61
5.2 Pembahasan Sistem	66
BAB VI PENUTUP	68
6.1 Kesimpulan.....	68
6.2 Saran	69

DAFTAR PUSTAKA	71
LAMPIRAN.....	74
1. Kode Program	
2. Matrix GLCM	
3. Tabel Penelitian	
4. Rekomendasi Penelitian.....	
5. Hasil Turnitin.....	
6. Keterangan Bebas Plagiasi	
7. Abstract.....	
8. Abstrak.....	
9. Keterangan Bebas Pustaka.....	
10. Biodata Mahasiswa.....	

DAFTAR GAMBAR

Gambar 2.1	Penyakit pada buah	9
Gambar 2.2	Penyakit layu pada daun.....	9
Gambar 2.3	Penyakit antraknosa pada daun	10
Gambar 2.4	Penyakit Bercak Daun.....	11
Gambar 2.5	Penyakit Rebai semai	12
Gambar 2.6	Penyakit busuk batang tanaman terong.....	13
Gambar 2.7	Penyakit Jamur Tepung.....	14
Gambar 2.8	Tingkat Grayscale	16
Gambar 2.9	Grayscale.....	17
Gambar 2.10	Piksel dengan berbagai sudut	18
Gambar 2.11	Contoh penentuan awal matriks GLCM	18
Gambar 2.12	Matriks framework menjadi matriks simetris	19
Gambar 2.13	Normalisasi matriks GLCM	19
Gambar 2.14	Kerangka Pikir	25
Gambar 3.1	Pemodelan.....	27
Gambar 5.1	Tampilan Menu Utama	66

DAFTAR TABEL

Table 2.1	Penelitian Terkait.....	5
Tabel 2.2	Penerapan Metode K- Nearest Neighbor (K-NN)	22
Tabel 2.3	K-NN (perhitungan kuadrat jarak data training dan data testing)	22
Tabel 2.4	K-NN (Hasil Perhitungan Berdasarkan Kelas).....	23
Tabel 2.5	K-NN(Klasifikasi Nearst Neighbor).....	23
Tabel 2.6	Tabel confusion matrix.	24
Tabel 4.1	Dataset Tanaman Terong.....	30
Tabel 4.2	Mengkonversi RGB ke <i>Greyscale</i>	33
Tabel 4.3	Citra Dengan Image 4x4	33
Tabel 4.4	Perhitungan Dengan Langkah dan Rumus.....	34
Tabel 4.5	Citra 4x4 Asli	34
Tabel 4.6	Data Citra 4x4 Setelah Dilakukan Histogram <i>Equalisation</i>	35
Tabel 4.7	Proses Perubahan Citra Asli Menjadi <i>Grayscale</i>	35
Tabel 4.8	Proses Normalisasi Citra	36
Tabel 4.9	Hasil Segmentasi Otsu	38
Tabel 4.10	Nilai Dari Suatu Citra	38
Tabel 4.11	Matrix Ternormalisasi.....	39
Tabel 4.12	Perhitungan Matrix <i>Contras</i>	40
Tabel 4.13	Perhitungan Matrix <i>Energy</i>	41
Tabel 4.14	Perhitungan Matrix <i>Homogenety</i>	43
Tabel 4.15	Perhitungan Matrix <i>Entropy</i>	45
Tabel 4.16	Hasil Perhitungan Data Training	47
Tabel 4.17	Hasil Perhitungan Data Testing	51
Tabel 4.18	Hasil Pengurutan dan Nilai Terkecil Berdasarkan Perhitungan Jarak dari Masing-masing Data Training ke Data Testing	56
Tabel 4.19	Hasil Jumlah K yang Sesuai Data.....	56
Tabel 4.20	Hasil Data Testing	57
Tabel 4.21	Consungision Matrix.....	59

Tabel 5.1	Hasil ujicoba 10 data testing dengan Glcm dan K yang Berbeda	61
Tabel 6.1	Rencana kegiatan Penelitian	74
Tabel 6.2	Matrix GLCM	

BAB I

PENDAHULUAN

1.1 Latar Belakang

Deteksi penyakit pada tanaman adalah salah satu hal terpenting dalam pemeliharaan dan perawatan tanaman.[1] Deteksi penyakit tanaman secara otomatis merupakan topik yang menjadi fokus utama para pemilik perkebunan. Berdasarkan pengamatan bahwa perkembangan teknologi visual dan produk digital ditunjang karena menggunakan pemrosesan gambar jika dilakukan dengan metode deteksi otomatis. Berdasarkan dari bentuk daun yang terkena penyakit, tekstur dan warna dapat dikategorikan masuk dalam pendeteksian. Bagaimanapun juga, deteksi otomatis ini masih banyak kekurangan dikarenakan sangat kompleksnya tanda dari penyakit tanaman itu sendiri.[2] Deteksi penyakit menggunakan komputer cukup direkomendasikan dikarenakan deteksi yang dihasilkan komputer dirasa cukup akurat. Dalam proses identifikasi hama dan penyakit tanaman terong tersebut dapat menggunakan teknologi komputasi dari pengolahan citra digital (*Digital Image Processing*).[3] Pada permasalahan tersebut peneliti melakukan suatu penelitian untuk menganalisis masalah penyakit terong. Maka solusi yang dibuat oleh peneliti yakni membuat sebuah sistem analisis yang mampu mendeteksi hama dan penyakit terong sebagai bentuk dari pengembangan teknologi digital (pengolahan citra) sehingga pendeteksian tidak lagi dilakukan secara manual.

Dalam penelitian ini digunakan metode *Gray Level Co-occurrence Matrix* (GLCM) untuk ekstraksi fitur daun dan *K-Nearest Neighbor* (KNN) untuk pengklasifikasian/identifikasi penyakit terong. Metode GLCM (*Gray-Level Cooccurrence Matrix*), yang digunakan untuk menganalisa Citra digital. Teknik ini sangat kental dengan matematika, seiring dengan pesatnya komputasi dengan bantuan komputer menjadikan penemuan ini lebih bermanfaat dan banyak digunakan. Sehingga memudahkan para peneliti untuk mendapatkan nilai pada fitur ekstraksi dan menghasilkan tingkat akurasi yang lebih tinggi.[4]

Metode GLCM cukup banyak digunakan oleh peneliti untuk mendapatkan tingkat akurasi yang cukup dalam hal ekstraksi khususnya pengolahan citra, metode GLCM adalah metode yang di rasa cukup efektif dan fitur ekstraksi yang sering di gunakan dari sekian banyak fitur ekstraksi yang ada.[4]

Metode klasifikasi pada penelitian ini menggunakan *K- Nearest Neighbor* dengan teknik melaksanakan klasifikasi terhadap objek bersumber pada informasi pendidikan yang jaraknya sangat dekat dengan objek tersebut. Keluaran dari sistem berbentuk pengelompokkan penyakit pada tumbuhan terong yang menginfeksi tumbuhan serta buah terong itu sendiri dimana hendak dipecah jadi 7 kelas ialah Busuk buah, Layu bakteri, antraknosa, Bercak daun, rebai semai, Busuk Pangkal batang dan tepung. Sehingga dengan terdapatnya sistem ini, para petani bisa terbantu guna memberikan tindak lanjut apabila tumbuhan terong menampilkan tanda-tanda penyakit.[5]

Dengan demikian, metode *Gray Level Co-Occurrence Matrix* (GLCM) dan *K-Nearest Neighbor* (KNN) bisa di gunakan untuk membantu petani dalam mentukan suatu penyakit pada tumbuhan terong secara otomatis. Program yang hendak digunakan guna menciptakan keperluan ini iyalah PYTHON. Penelitian sebelumnya yang dilakukan oleh Padang Sukmandaru, 2019 dengan judul Identifikasi Penyakit Tanaman Tomat Berdasarkan Citra Daun Menggunakan Metode *Gray Level Co-Occurrence Matrix* (GLCM) Dan *K-Nearest Neighbor* (KNN).[6]

Penelitian ini melakukan analisis fitur *gray level co-occurence matrix* (GLCM) citra daun tomat, *gray level co-occurance matrix* (GLCM) merupakan salah satu metode untuk memperoleh tekstur kontras, energi, homogeneity, dan korelasi pada suatu gambar dengan menghitung probabilitas hubungan spasial antara dua piksel pada jarak dan arah tertentu, untuk fitur yang digunakan yaitu kontras, *angular second moment* (energi), *inverse different moment* (Homogenitas), dan korelasi.

K-Nearest Neighbor yaitu berfungsi untuk menghitung jarak terdekat antara data citra latih dan data citra uji untuk didapatkan hasil pembelajaran

maupun pengenalan dan dibandingkan dengan kondisi sebenarnya. Penelitian ini akan menghasilkan identifikasi penyakit tanaman tomat dengan kategori layu fusarium, bercak daun dan daun sehat. informasi sampel pada riset ini menggunakan 90 informasi latih. Pada riset identifikasi penyakit tumbuhan diperoleh hasil dari akurasi sistem dengan hasil menggapai 85%.[7]

Berdasarkan latar belakang di atas, maka dengan ini peneliti melakukan penelitian dengan judul: **“GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM) UNTUK DETEKSI PENYAKIT PADA TANAMAN TERONG MENGGUNAKAN METODE K-NEAREST NEIGHBOR (KNN)”**.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah di susun di atas maka dapat di tarik beberapa permasalahan yang timbul pada deteksi h penyakit pada tanaman terong itu sendiri antara lain:

1. Petani masih menggunakan cara manual untuk deteksi penyakit pada tanaman terong.
2. Belum adanya teknologi visual dan produk digital dalam hal deteksi penyakit pada tanaman terong.

1.3 Rumusan Masalah

Adapun masalah yang diteliti dalam penelitian ini adalah:

1. Bagaimana hasil dari metode GLCM dan *K-Nearest Neighbor* (KNN) Untuk bisa mendeteksi penyakit pada tanaman terong?
2. Bagaimana efektifitas dan kinerja dari metode GLCM dan *K-Nearest Neighbor* (KNN) untuk mengklasifikasi penyakit pada tanaman terong?

1.4 Tujuan Penelitian

1. Untuk mengetahui penyakit pada tanaman terong menggunakan metode GLCM dan *K-Nearest Neighbor* (KNN).
2. Untuk mendapatkan hasil klasifikasi dari deteksi penyakit pada tanaman terong dengan menggunakan metode GLCM dan *K-Nearest Neighbor* (KNN).

1.5 Manfaat Penelitian

1.5.1 Manfaat Teoritis

Penelitian ini di harapkan dapat memberikan ilmu pengetahuan di bidang teknologi komputer khususnya dalam ilmu pengetahuan *GLCM* dan *K-Nearest Neighbor* (KNN).

1.5.2 Manfaat Praktis

Hasil penelitian ini dapat di gunakan sebagai salah satu bahan dasar dalam penerapan metode *GLCM* dan *K-Nearest Neighbor* (KNN) dalam melakukan pendeteksian penyakit tanaman.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Studi

Ada beberapa penelitian yang terkait dengan metode *K-Nearest Neighbor* (KNN) seperti dibawah ini :

Table 2.1. Penelitian Terkait

NO	PENELITI	JUDUL	HASIL
1	Padang Sukmandaru ,2019 [6]	Penyakit Tanaman Tomat Berdasarkan Citra Daun Menggunakan Metode Gray Level Co-Occurrence Matri (GLCM) Dan K-Nearest Neighbor (KNN)	Penelitian ini akan menghasilkan identifikasi penyakit tanaman tomat dengan kategori layu fusarium, bercak daun dan daun sehat. Data sampel pada penelitian ini menggunakan 90 data latih. Pada Penelitian identifikasi penyakit tanaman diperoleh hasil dari akurasi sistem dengan hasil mencapai 85%.

2	Azizah Arif Paturrahman ,2020[7]	Analisis Pengenalan Pola Daun Bersumber pada Fitur <i>Canny Edge</i> Detection Dan Fitur GLCM Memaik Metode Klasi - fikasi <i>K-Nearest Neighbor</i> (K-NN)	Proses klasifikasi citra daun bisa dilakukan dengan cara mengidentifikasi citra bentuk daun yang bisa dilakukan dengan cara mengenali pola daun mengidentifikasi ciri struktural daun seperti bentuk serta tekstur daun tersebut. Klasifikasi citra daun ini berdasarkan fitur canny edge detection dan <i>Gray-Level Co-occurrence Matrix</i> (GLCM) menggunakan metode <i>K-Nearest Neighbor</i> (KKN) dengan jumlah data sebanyak 350 citra daun dengan 7 spesies yang berbeda yang diperoleh dari data penelitian sebelumnya.
3	Mhd.Furqan, Sriani,Lailan Sofinah Harahap, 2020 [8]	Klasifikasi Daun BugenvilMenggunakan GLCM Dan K-Nearest Neighbor (KNN).	Klasifikasi citra daun ini berdasarkan fitur <i>canny edge detection</i> dan <i>Gray-Level Co-occurrence Matrix</i> (GLCM) menggunakan metode <i>K-Nearest Neighbor</i> (KKN) dengan jumlah data sebanyak 350 citra daun dengan 7 spesies yang berbeda yang diperoleh dari data penelitian sebelumnya.

2.2. Tinjauan Pustaka

2.2.1 Deteksi

Deteksi adalah suatu proses untuk memeriksa atau melakukan pemeriksaan terhadap sesuatu dengan menggunakan cara dan teknik tertentu. Deteksi dapat digunakan untuk berbagai persoalan, misalnya dalam sistem pendeteksi suatu penyakit, dimana sistem mengidentifikasi masalah-masalah yang berhubungan dengan penyakit yang biasa disebut gejala. Tujuan dari deteksi adalah memecahkan suatu persoalan atau masalah dengan berbagai cara tergantung metode yang diterapkan sehingga menghasilkan sebuah solusi.[9]

2.2.2 penyakit Tanaman

penyakit adalah kelompok mikroorganisme yang aktifitasnya menimbulkan kerusakan pada tanaman sehingga mengakibatkan pertumbuhan dan perkembangan tanaman menjadi terganggu dan berdampak pada kerugian secara ekonomis. Berdasarkan organisme penyebabnya penyakit dapat disebabkan oleh jamur, bakteri, virus.[10]

penyakit tanaman perlu diketahui agar penanganan teratasi. Apalagi dalam budidaya berbagai tanaman, terkadang penyakit menjadi persoalan yang biasa terjadi, namun, kita dapat mengendalikannya agar tetap mendapatkan hasil yang maksimal dan terbaik. Serangan penyakit yang timbul di lapangan dapat dipengaruhi oleh lingkungan misalnya keadaan air, kesuburan tanah, suhu, kelembaban udara, penggunaan bibit unggul dan cara budidaya. [11]

2.2.3 Tanaman Terong

Tanaman terong (*Solanum melongena L*) termasuk salah satu tanaman sayur-sayuran. Di dalam kehidupan sehari-hari buah terong dapat digunakan sebagai sayur lodeh, opor, lalap segar ataupun lalap masak

karena cita rasanya yang enak, selain itu dapat juga dibuat terong asinan dan manisan. Dalam dunia kesehatan terong dikenal sebagai penurun kolesterol darah dan mengandung zat anti kanker.[12]

2.2.4 Penyakit Tanaman Terong

Dalam melakukan budidaya tanam terong baik terong ungu, terong lalap, terong belanda, terong putih ataupun terong hijau pastilah akan ada kendala yang dialami, salah satunya serangan hama dan penyakit.

Penyakit = Busuk buah, Layu bakteri, antraknosa, Bercak daun, rebai semai, Busuk Pangkal batang dan tepung.

Adapun penyakit tanaman terong adalah sebagai berikut : [14]

1. Penyakit Busuk buah.



Gambar 2.1 : Penyakit pada buah.

Busuk buah adalah Penyakit yang diakibatkan oleh jamur yang bernama *phytophthora Sp.* Gejala awal yang ditemui pada penyakit ini biasanya buah akan memiliki bercak cokelat kehitaman, bercak tersebut akan membesar keseluruh daging buah dan mengakibatkan buah lembek dan membusuk.

Pengendalian : Menanam varietas yang tahan, Ambil dan musnahkan buah yang terinfeksi, Semprotkan dengan fungisida *dithane*, *antracol*, *cozeb*, atau *Bion M*. [14]

2. Penyakit layu bakteri.



Gambar 2.2 : Penyakit layu pada daun.

Penyakit layu bakteri ini disebabkan oleh *patogen Ralstonia solanacearum* yang merupakan suatu bakteri yang dapat hidup, tumbuh, dan berkembang biak di dalam tanah. Munculnya penyakit ini sering terjadi ketika musim penghujan dikarenakan bakteri ini dapat berkembangbiak dengan cepat, bahkan bisa 2 kali lipat. Gejala yang timbul pada tanaman terong yang terserang penyakit ini yaitu mula-mula terjadinya layu dibagian pucuk daun kemudian akan layu seluruhnya.[14]

Pengendalian : Rotasi tanaman yang bukan inang, Menjaga area tanaman agar tidak lembab, Mencabut dan memusnahkan tanaman terserang lalu Semprotkan bakterisida pada pangkal batang. Bakterisida yang digunakan misalnya *agrimicin*, *agrept*, *bactoxyn* atau *plantomycin*.

3. Penyakit *Antraknosa*.



Gambar 2.3 : Penyakit *antraknosa* pada daun.

Gloesporium Melongena merupakan jenis jamur yang menyebabkan penyakit *antraknosa* pada pohon terong. Ciri-cirinya seperti permukaan daun yang warnanya berubah dan kemudian mulai menggulung.[14]

Pengendalian : Menanam *varietas* yang tahan, Mengatur jarak tanam, Membersihkan gulma yang bisa menjadi tanaman inang, Memusnahkan buah terinfeksi dan memangkas bagian tanaman yang terserang, Semprotkan dengan fungisida *antracol*, *cozeb*, *Bion M* atau *Dithane*.

4. Penyakit Bercak Daun.



Gambar 2.4 : Penyakit Bercak Daun.

Gejala awal serangan bercak daun pada tanaman terong ditandai dengan adanya bercak berwarna kehitaman pada permukaan daun. pada musim hujan akan cepat menyebar ke seluruh daun tanaman terong. Pada serangan yang sudah parah, bercak daun menyebabkan daun menjadi kuning seperti terbakar hingga akhirnya rontok. Penyakit bercak daun dapat menyebar dengan cepat pada kondisi curah hujan tinggi. Hal ini dikarenakan cendawan *cercospora* berkembang dengan cepat pada kondisi yang sangat lembab.[14]

Pengendalian : Mengatur jarak tanam, Menggunakan mulsa plastik jika menanam dimusim hujan, Menjaga kebersihan areal dari gulam agar area tanaman tidak lembab, Semprotkan fungisida starmyl, saromyl, score atau dithane.

5. Penyakit Rebai Semai.



Gambar 2.5 : Penyakit Rebai semai.

Rebai semai adalah penyakit yang menyerang terong pada saat sedang disemai. Penyakit ini mengakibatkan benih menjadi kekuningan lalu kemudian mati sehingga tidak mungkin untuk Anda menanamnya lagi.[14]

pengendalian : dapat dengan menggunakan bibit unggul, Menggunakan varietas yang tahan, Mencabut dan membuang tanaman terserang, Siram semaian seperlunya, jangan terlalu basah/lembab, Semprotkan fungisida antracol, dithane, Bion M atau Cozeb serta menjaga sanitasi lingkungan sekitar penyemaian.

6. Penyakit Busuk Pangkal batang .



Gambar 2.6 : Penyakit busuk batang tanaman terong.

Penyakit busuk batang pada tanaman terong disebabkan oleh cendawan *Phytophthora capsici* dan *Phytophthora infestans*. Pada musim penghujan dengan curah hujan yang sangat tinggi, dengan kelembaban yang tinggi intensitas serangan cendawan cepat meningkat dan berkembang biak dengan cepat. Gejala dan Penyebab Penyakit Busuk Batang terdapat bercak berwarna kecoklatan pada pangkal batang, cabang, maupun bagian pucuk tanaman. Bercak tersebut jika disentuh terasa lembek, kulit mudah terkelupas dan lama kelamaan bercak berubah warna menjadi kehitaman. Bagian tanaman yang terserang akan layu, menguning, kering, dan mati.[14]

Pengendalian : Menggunakan varietas yang tahan, Mengatur jarak tanam, Mencabut dan memusnahkan tanaman yang terserang, Menjaga kelembaban tetap stabil, Semprotkan *fungisida antracol, dithane, cozeb* atau *bionM*.

7. Penyakit Tepung atau Jamur tepung.



Gambar 2.7 : Penyakit Jamur Tepung.

Jamur tepung biasanya akan menyerang daun dan pangkal tanaman. Pada daun akan terlihat serbuk putih permukaan atas dan bawah daun layaknya tepung yang lama kelamaan akan berubah warna menjadi warna kuning dan merusak pohon terong. Penyakit embun tepung memang tidak akan berdampak langsung pada buah, namun penyakit ini mampu membunuh tanaman. Daun yang terserang perlahan akan berwarna kekuningan, hingga daun mati dan gugur.

Pengendalian : Hindari pengairan yang berlebih, Lakukan pengendalian gulma di lahan, Aplikasi fungisida dilakukan tiap 5 hari sekali jika kondisi lingkungan lembab dan basah, namun jika kondisi cuaca sedang kering, maka aplikasi fungisida dapat dilakukan dalam interval waktu 7-10 hari dengan menggunakan heksakonazol, triflumizol, miklobutanil (dengan masing masing takaran 2ml/l).[14]

2.2.5 Pengolahan Citra Digital

Pengolahan citra digital adalah proses yang bertujuan untuk memanipulasi dan menganalisis citra menggunakan bantuan komputer. Pengolahan citra digital dapat dikelompokkan dalam dua aspek kegiatan :

1. Memperbaiki kualitas gambar, agar dapat lebih mudah diinterpretasi oleh mata manusia.
2. Mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis.

Aspek kedua sangat erat hubungannya dengan ilmu pengenalan pola (*pattern recognition*) yang umumnya bertujuan mengenali suatu objek dengan cara mengekstrak informasi penting yang terdapat pada suatu citra.[15]

Proses pengambilan gambar dilakukan dengan Posisi vertikal yaitu sumbu kamera berada dalam posisi vertikal dengan area yang dipotret, dengan menggunakan prangkat kamera smartphone dengan resolusi bawaan agar tetap mempertahankan hasil yang alami yang nantinya akan di reziise kembali sesuai kebutuhan dalam proses pengolahan citra.

2.2.6 Jenis Citra Digital

Pada aplikasi pengolahan citra digital umumnya, citra digital dibagi menjadi 3, *RGB image*, *Grayscale image* dan *Binary image* :[16]

1. RGB

Citra berwarna atau citra RGB adalah citra yang menyediakan beberapa warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Masing-masing komponen menggunakan delapan bit yang nilainya berkisar antara antara 0 sampai dengan 255.[17]

2. *Grayscale*

Grayscale adalah suatu citra citra yang hanya memiliki warna tingkat keabuan. Gambar *grayscale* 8 bit memiliki 256 tingkat warna abu-abu mulai dari putih hingga hitam[18]. Tingkat *Grayscale* dapat di lihat pada gambar 2.8 :



Gambar 2.8 : Tingkat Grayscale.[19]

Untuk mendapatkan nilai *grayscale* dapat dilakukan dengan mengambil rata-rata

dari nilai R, G dan B. seperti di

bawah ini :

$$s = \frac{r + g + b}{3} \quad (1)$$

3. *Binary Image*

Binary image terdiri dari warna hitam atau putih, karena hanya ada dua warna untuk per piksel, maka hanya perlu 1 bit setiap piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255), sehingga sangat efisien dalam hal penyimpanan. atau gambar arsitektur. Binary image merupakan hasil pengolahan dari *blackand white image*[16].

2.2.7 Proses Grayscale

Proses *grayscale* adalah proses dimana citra input yang terdiri dari tiga layer warna RGB dirubah hanya menjadi satu layer, dimana layer tersebut menyatakan tingkat keabuan disetiap titik pada citra. Proses ini diperlukan karena pada proses ekstraksi ciri GLCM memerlukan citra *grayscale* sebagai input-nya. Proses perubahan ini dilakukan dengan melakukan penjumlahan komponen RGB yang telah diberikan bobot masing-masing.[20]



Gambar Asli

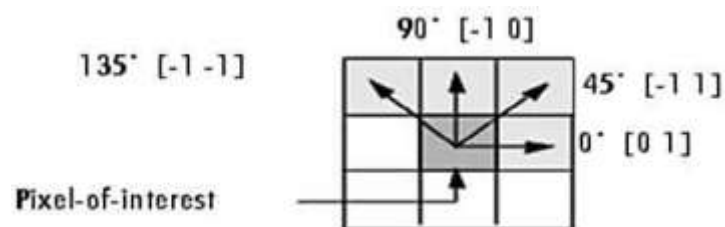
Gambar Grayscale

Gambar 2.9 : Grayscale.

2.2.8 Gray Level Co-occurrence Matrix

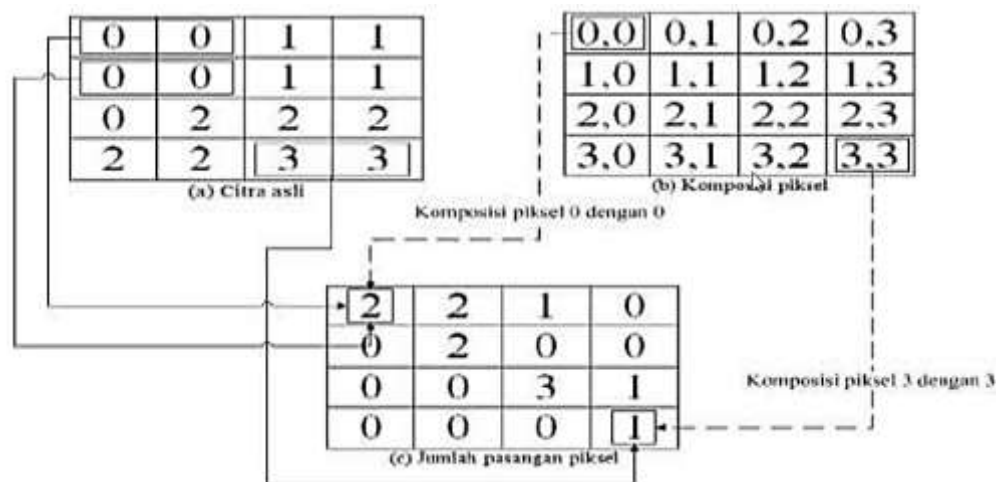
Metode *Gray Level Co-occurrence Matrix* (GLCM) adalah salah satu ekstraksi order kedua pada fitur statistic tekstur. Ekstraksi order kedua menunjukkan hubungan statistic antara 2 piksel. GLCM adalah sebuah matriks dengan jumlah baris dan kolom sebanding dengan jumlah *Gray Level* (G) dalam suatu citra. Metode GLCM menggunakan citra berkala keabuan (*Grayscale*). ciri diperoleh dari nilai piksel matrix yang mempunyai nilai tertentu dan membentuk suatu sudut pola . GLCM merepresentasikan hubungan dua piksel yang bertetangga dimana dua piksel yang berhubungan tersebut memiliki intensitas keabuan tertentu serta memiliki jarak dan arah tertentu diantara keduanya.

Jarak dinyatakan piksel dan arah dinyatakan sudut. Jarak dapat bernilai 1, 2, 3 dan seterusnya sedangkan arah dapat bernilai 0° , 45° , 90° , 135° dan seterusnya[21].



Gambar 2.10 : Piksel dengan berbagai sudut. (Eleyan & Demirel, 2011)

Sebagai ilustrasi, ketetanggaan piksel yang dapat dipilih ke arah timur (kanan). Salah satu untuk merepresentasikan hubungan ini yaitu berupa (1,0), yang menyatakan hubungan dua piksel bernilai 1 diikuti dengan piksel bernilai 0, sehingga jumlah kelompok piksel memenuhi hubungan tersebut dihitung[21].



Gambar 2.11 : Contoh penentuan awal matriks GLCM. (Kadir & Susanto, 2013)

Matriks pada Gambar 2.12 dinamakan matrix framework. Matriks ini kemudian diolah menjadi matriks dimetris dengan cara menambahkan

hasil transposnya[21](Fazrini, 2018). Seperti yang terlihat pada Gambar 2.12

$$\begin{bmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$



GLCM sebelum dinormalisasi

Gambar 2.12 : Matriks *framework* menjadi matriks simetris. (Kadir & Susanto, 2013)

Untuk menghilangkan ketergantungan pada ukuran citra, nilai-nilai elemen GLCM perlu dinormalisasikan sehingga jumlahnya bernilai 1. Dengan demikian, hasil normalisasi dari matriks GLCM pada Gambar 2.13

$$\begin{bmatrix} \frac{4}{24} & \frac{2}{24} & \frac{1}{24} & \frac{0}{24} \\ \frac{2}{24} & \frac{4}{24} & \frac{0}{24} & \frac{0}{24} \\ \frac{1}{24} & \frac{0}{24} & \frac{6}{24} & \frac{1}{24} \\ \frac{0}{24} & \frac{0}{24} & \frac{1}{24} & \frac{2}{24} \end{bmatrix}$$

Gambar 2.13 : Normalisasi matriks GLCM. (Kadir & Susanto, 2013)

Untuk mendapatkan fitur tekstur GLCM, hanya 14 besaran yang diusulkan untuk dipakai. Beberapa fitur yang akan dipakai nantinya adalah *contrast*, *homogeneity*, *entropy*, *energy* sebagai berikut :[21]

1. *Contrast* digunakan untuk mengukur frekuensi spasial dari citra dan perbedaan momen GLCM. Perbedaan yang dimaksudkan adalah perbedaan tinggi dan rendah pixel, Contrast merupakan ukuran keberadaan variasi atas keabuan pixel citra dihitung.

(2)

$$Contrast = \sum_{i_1} \sum_{i_2} (i_1 - i_2)^2 p(i_1, i_2)$$

2. *Homogeneity* digunakan untuk mengukur homogenitas yaitu ukuran kedekatan distribusi masing-masing elemen pada matriks GLCM ke matriks GLCM diagonal. Homogeneity dihitung dengan cara berikut :

(3)

$$Homogeneity = \sum_{i_1} \sum_{i_2} \frac{p(i_1, i_2)}{1 + |i_1 - i_2|}$$

3. *Entropy* adalah mengukur heterogenitas dan berkorelasi dengan standar deviasi. Penghitungan *Entropy* digunakan untuk mengukur kompleksitas (keacakan) citra, entropy akan bernilai tinggi ketika citra tidak seragam. Entropy dihitung dengan cara berikut :

(4)

$$Entropy = - \sum_{i_1} \sum_{i_2} p(i_1, i_2) \log p(i_1, i_2)$$

4. *Energy* adalah pengukuran intensitas keseragaman piksel. Sebuah keadaan homogen mengandung hanya sedikit gray level tetapi memiliki nilai piksel Pij yang tinggi, oleh karena itu jumlah dari pangkat Pij akan tinggi.

(5)

$$Energy = \sum_{i_1} \sum_{i_2} p^2(i_1, i_2)$$

2.2.9 Metode K- Nearest Neighbor (K-NN)

K-Nearest Neighbor (KNN) adalah suatu metode yang menggunakan algoritma supervised dimana hasil dari query instance yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN. Tujuan dari algoritma ini adalah mengklasifikasikan objek baru berdasarkan atribut dan training sample. Classifier tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik query, akan ditemukan sejumlah k objek atau (titik training) yang paling dekat dengan titik query[21].

Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k objek. Metode algoritma KNN sangatlah sederhana, bekerja berdasarkan jarak terpendek dari query instance ke training sample untuk menentukan KNN-nya. Training sample diproyeksikan ke ruang banyak, dimana masing-masing dimensi mempresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi training sample. Sebuah titik pada ruang ini ditandai sebagai c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat dari titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan Euclidean Distance yang dipresentasikan sebagai berikut :[21]

Dimana :

$$D(a, b) = \sqrt{\sum_{k=1}^d (a_k - b_k)^2}$$

k = variabel data

d = dimensi data

a_k = data pelatihan

b_k = data uji

2.2.10 Penerapan Metode *K- Nearest Neighbor* (K-NN)

Terdapat beberapa data yang berasal dari survey questioner tentang klasifikasi kualitas kertas tissue apakah baik atau jelek, dengan objek training dibawah ini menggunakan dua attribute yaitu daya tahan terhadap pengenalan bunga anggrek[21].

Tabel 2.2. K-NN (Klasifikasi Data training)[21]

X1	X2	Y = Klasifikasi
9	6	Tidak segar
4	5	Tidak segar
3	4	Segar
4	4	Segar

Terdapat data testing X1=5 dan x2=6

Penyelesaian :[21]

1. Tentukan Nilai K(Nilai K yang digunakan K=3)
2. Hitung jarak antara data baru dengan semua data training seperti table 2.3

Tabel 2.3. K-NN (perhitungan kuadrat jarak data training dan data testing)[21]

X1	X2	Kuadrat jarak dengan data baru (5,6)
9	6	$(9-5)^2 + (6-6)^2 = 16$
4	5	$(4-5)^2 + (5-6)^2 = 2$
3	4	$(3-5)^2 + (4-6)^2 = 8$
4	4	$(4-5)^2 + (4-6)^2 = 5$

3. Kemudian urutkan kuadrat jarak dari yang terkecil

Tabel 2.4 K-NN (Hasil Perhitungan Berdasarkan Kelas)[21]

X1	X2	Euclidean Distance	Jarak Terkecil	Tetangga Terdekat
9	6	$\sqrt{16} = 4$	4	TIDAK
4	5	$\sqrt{2} = 1.4$	1	YA
3	4	$\sqrt{8} = 2.8$	3	YA
4	4	$\sqrt{5} = 2.2$	2	YA

4. Mengumpulkan kategori Y(klasifikasi nearest neighbor)[21]

Tabel 2.5 K-NN(Klasifikasi Nearst Neighbor)[21]

X1	X2	Euclidean Distance	Jarak Terkecil	Tetangga Terdekat	Y
9	6	$\sqrt{16} = 4$	4	TIDAK	segar
4	5	$\sqrt{2} = 1.4$	1	YA	Tidak segar
3	4	$\sqrt{8} = 2.8$	3	YA	segar
4	4	$\sqrt{5} = 2.2$	2	YA	segar

5. Berdasarkan hasil dari tabel 3.7 dimana nilai $K=3$, kita memiliki 3 segar dan 1 tidak segar, maka dapat kita simpulkan bahwa hasil klasifikasi data testing dengan nilai $X1=5$ dan $X2=6$ termasuk kategori Segar[21].

2.2.11 Confusion Matrix

Confusion matrix merupakan suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi. *Confusion matrix* memberikan keputusan yang diperoleh penilaian Performance deteksi berdasarkan objek dengan benar atau salah (*gurunescu*). *Confusion matrix* berisi informasi actual dan prediksi pada sistem deteksi.

Table 2.6. Tabel confusion matrix.

	Actual	Actual
	Positive	Negative
<i>Predicted positive</i>	<i>TP</i>	<i>FP</i>
<i>Predicted Negative</i>	<i>FN</i>	<i>TN</i>
Recall	=	$TP/(TP+FN)$
Precision	=	$TP/(TP+FP)$
True Positive rate	=	$TP/(TP+FN)$
False Negative rate	=	$FP/(FP+TN)$

Keterangan :

- True positive (TP) = proporsi positif dalam dataset yang dideteksi positif.
- True negative (TN) = proporsi negative dalam dataset yang dideteksi negativ.
- False positive (FP) = proporsi negativ dalam dataset yang dideteksi positif.
- False negative (FN) = proporsi negativ dalam dataset yang dideteksi negativ.

2.3. Kerangka Pikir



Gambar 2.14 : Kerangka Pikir

BAB III

METODE PENELITIAN

3.1 Jenis Metode, Subjek, Objek, Waktu dan Lokasi Penelitian

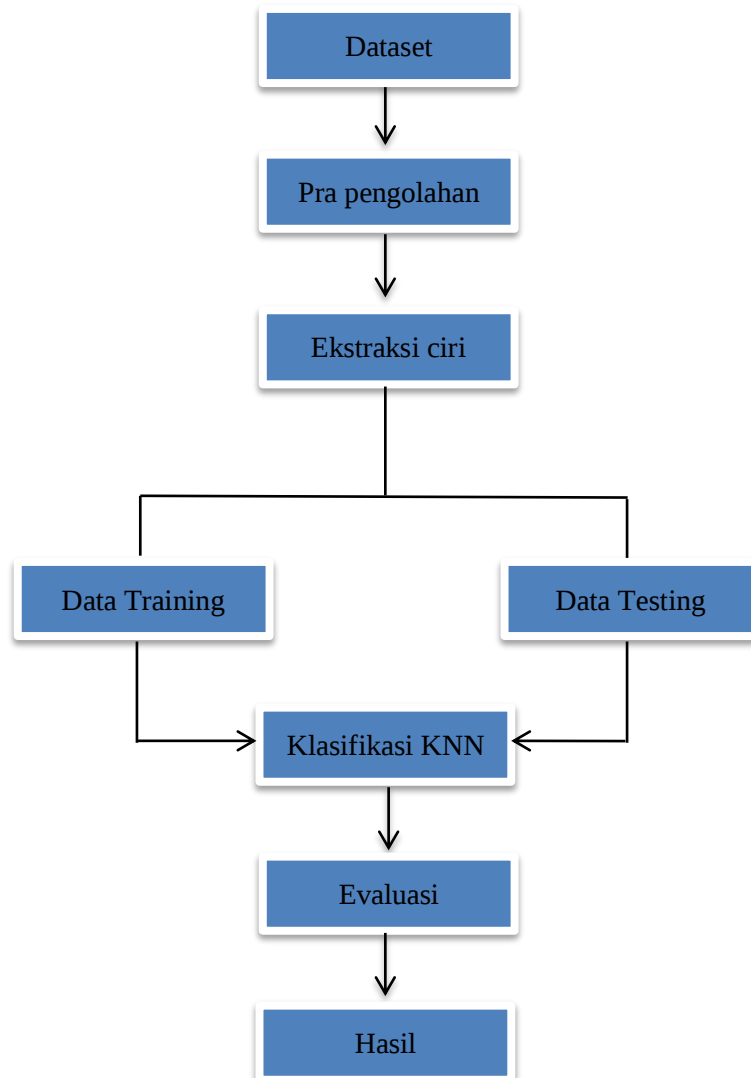
Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian terapan. Dipandang dari jenis informasi yang di olah, maka penelitian ini merupakan penelitian kuantitatif. Dipandang dari perlakuan terhadap data, maka penelitian ini merupakan merupakan konfirmatori.

Penelitian ini merupakan metode penelitian *eksperimen* dan *studi kasus*. Dengan demikian jenis penelitian ini adalah penelitian *eksperimental* dan *studi kasus*. Subjek penelitian ini adalah deteksi penyakit pada tanaman terong. Adapun Tabel rencana/waktu kegiatan penelitian di lampirkan setelah Daftar Pustaka.

3.2 Pengumpulan Data

Data primer penelitian ini adalah penyakit pada tanaman terong yang merupakan data observasi dan dokumentasi. Citra yang dikumpulkan dibagi menjadi dua dataset yaitu: data latih dan data uji. Data latih yang digunakan sebagai pembandingan terhadap data uji pada proses pendeteksian hama dan penyakit pada tanaman terong. Sedangkan data uji adalah data digunakan untuk mengetahui hasil deteksi penyakit pada tanaman terong yang didapatkan melalui proses deteksi.

3.3 Pemodelan



Gambar 3.1 : Pemodelan

3.3.1 Pra Pengolahan Data

Pada tahap ini, data yang telah ada dilakukan praproses terlebih dahulu. Praproses dilakukan secara manual dengan mengubah warna latar pada citra buah menjadi putih. Proses untuk mengubah warna dilakukan dengan memisahkan citra buah dari latarnya dan memindahkan citra tersebut ke sebuah latar putih berukuran 318*318 pixel. Selanjutnya, dilakukan normalisasi histogram ekualisasi pada setiap gambar untuk membuat kesamaan nilai thresholding masing-masing gambar. Setelah setiap gambar telah dinormalisasi kemudian dilakukan teknik otsu threshold untuk mendeteksi hama dan penyakit yang ada. Pada proyek ini dilakukan threshold bawah, yaitu mengambil warna yang mendekati nilai 0 pada channel green untuk mendeteksi penyakit blackspot. Selain itu dilakukan *threshold* atas yaitu mengambil warna yang mendekati nilai 255 pada channel green dan yellow untuk mendeteksi penyakit *frogeye*.

3.3.2 Ekstraksi Ciri

Pada proses ekstraksi ciri data input akan dibaca dan kemudian diolah dalam dua proses yang berlainan. Proses ekstraksi fitur tekstur menggunakan metode GLCM (*Gray Level Co-Occurrence Matrix*). Masukan dari proses ini adalah citra penyakit tanaman terong. Semua citra diambil pada kondisi pencahayaan yang terang. Output dari proses ini adalah matriks $N \times 1$ dimana N adalah jumlah vektor ciri.

3.3.3 Data Training

Data training ini merupakan kumpulan data yang telah terekstak cirinya yang selanjutnya akan dilatih menggunakan algoritma *K- Nearest Neighbor* (K-NN). Jumlah data training yang digunakan berupa 10% dari sampel daun, buah, dan batang tanaman terong dengan deteksi penyakit yang berbeda. Data training ini berupa hasil dari ekstraksi GLCM (*Gray Level Co-Occurrence Matrix*) terhadap penyakit daun terong.

3.3.4 Data Testing

Data testing merupakan data yang telah terekstak cirinya, untuk menguji data yang telah dilatih, jumlah data training yang digunakan berupa 90% dari sampel tanaman terong dengan deteksi penyakit yang berbeda. data testing merupakan ciri dari hasil ekstraksi GLCM (*Gray Level Co-Occurrence Matrix*) terhadap penyakit tanaman terong.

3.3.5 Model Deteksi

Model deteksi merupakan tahapan yang terdiri dari *input*, *hidden layer* dan *output* dengan kombinasi bobot terbaik yang dihasilkan dari proses pelatihan data training menggunakan Metode *K- Nearest Neighbor* (K-NN). Model ini akan digunakan untuk pendeteksian dan *testing* berdasarkan *output* yang dihasilkan tersebut.

3.3.6 Hasil Deteksi

Hasil deteksi merupakan output, pada data testing yang didapatkan dari proses deteksi yang menggunakan Metode *K- Nearest Neighbor* (K-NN) berdasarkan model yang diperoleh dari data training.

3.3.7 Evaluasi

Proses evaluasi bertujuan untuk mengetahui kinerja dari metode GLCM (*Gray Level Co-Occurrence Matrix*) terhadap penyakit tanaman terong yang digunakan. Proses evaluasi dilakukan pada seluruh data *testing* kemudian hasil *output* yang akan dipetakan kedalam *Confusion Matrix* untuk dihitung nilai akurasinya.

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Dalam penelitian ini peneliti ingin mendeteksi penyakit pada terong dengan menggunakan metode GLCM (*Grey Level Co-onccurrence Matrix*) dan algoritma KNN(*K-Nearst Neighbor*). Seperti yang telah peneliti tuangkan pada tahapan metode penelitian dimana dataset gambar yang digunakan dalam penelitian ini diambil dari proses observasi dan dokumentasi. Adapun dalam pengumpulan data tersebut telah berhasil peneliti kumpulkan sebanyak 104 citra. Seluruh data citra tersebut telah peneliti bagi ke dalam 94 data gambar untuk data training dan 10 data gambar untuk data testing. Adapun secara terperinci yang keseluruhan objek terdiri atas 10 objek yakni citra daun sehat, citra buah sehat, citra batang sehat, citra buah busuk, citra layu bakteri, citra antraknosa, citra bercak daun, citra rebai semai, citra busuk pangkal batang, dan citra jamur tepung dimana semua objek gambar dengan format file gambar jpg. Berikut sampel gambar yang berhasil dilakukan :

Tabel 4.1 Dataset Tanaman Terong

Keterangan	Terong Sehat	
1. Buah Terong Sehat		
2. Daun Terong Sehat		

3. Batang Terong Sehat		
Keterangan	Penyakit Terong	
4. Buah Terong Busuk		
5. Penyakit Layu Bakteri		
6. Penyakit Antraknosa		
7. Penyakit Bercak Daun		

8. Penyakit Rebai Semai		
9. Penyakit Batang		
10. Jamur Tepung		
11. dst	-----	-----

Di dalam bab hasil dan pembahasan ini, peneliti telah berhasil mengumpulkan sebanyak 104 dataset. Dari keseluruhan dataset di dalam penelitian ini, telah peneliti bagi ke dalam rasio 90% :10% sehingga peneliti mendapatkan 94 data untuk training dan 10 data untuk test. Adapun pada bab ini peneliti akan menampilkan hasil perhitungan dari representasi data test tersebut.

Tabel 4.2 Mengkonversi RGB ke Grayscale

Daftar Gambar	Nilai R, G, B	Hasil
	$239R + 240G + 239B$	239,33
	$156R + 143G + 144B$	147,66
	$205R + 193G + 197B$	198,33

4.1.1 HISTOGRAM EQUALIZATION

Histogram Equalization adalah teknik pemrosesan gambar komputer yang digunakan untuk meningkatkan kontras dalam gambar. menyelesaikan ini secara efektif menyebarkan nilai intensitas yang paling sering, yaitu merentangkan rentang intensitas gambar. Metode ini biasanya meningkatkan kontras global gambar ketika data yang dapat digunakan diwakili oleh nilai kontras dekat. Ini memungkinkan area dengan kontras lokal yang lebih rendah untuk mendapatkan kontras yang lebih tinggi. Dari hasil penurunan rumus perataan Histogram secara matematis didapat tranformasi sebagai berikut:

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j)$$

$$= \sum_{j=0}^k \frac{n_j}{n}$$

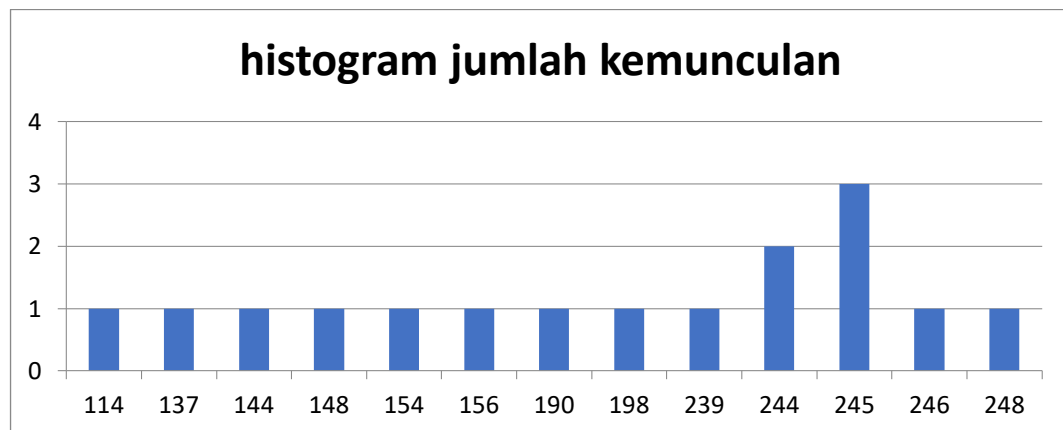
Sebagai Contoh didapat sebuah citra dengan image 4x4

Tabel 4.3Citra Dengan Image 4x4

239	148	198	245
246	154	144	245
248	156	114	244
244	137	190	245

Dengan Histogram sebagai berikut :

Jml. Kemunculan



Setelah didapat histogram asli dari citra diatas, maka dilakukan perhitungan dengan langkah dan rumus pada tabel dibawah :

Tabel 4.4 Perhitungan Dengan Langkah Dan Rumus

Derajat Keabuan (j)	114	137	144	148	154	156	190	198	239	244	245	246	248
Kemunculan	1	1	1	1	1	1	1	1	1	2	3	1	1
$s = \sum_{j=0}^k \frac{n_j}{n}$	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	1/16	2/16	3/16	1/16	1/16

Hasil dari perhitungan Perataan Histogram diatas telah menunjukan sejumlah data citra baru seperti pada tabel diatas, dan disimpulkan seperti berikut :

Tabel 4.5 Citra 4x4 Asli

239	148	198	245
246	154	144	245
248	156	114	244
244	137	190	245



Tabel 4.6 Data Citra 4x4 Setelah Dilakukan Histogram Equalisation

0,0625	0,0625	0,0625	0,1875
0,0625	0,0625	0,0625	0,1875
0,0625	0,0625	0,0625	0,125
0,125	0,0625	0,0625	0,1875

4. 2. Hasil Pemodelan

4. 2. 1. Pra Pengolahan

Pra-pengolahan citra (image pre-processing), yaitu proses paling awal dalam pengolahan citra sebelum proses utama dilakukan. Pada tahap ini citra tanaman terong yang sudah ada dikonversi agar diperoleh data citra tanaman terong yang sesuai kebutuhan. Tahap ini berfungsi untuk menormalisasi citra tanaman terong dari permasalahan luminasi yang terlalu gelap atau terlalu terang sehingga dapat meningkatkan performansi dari sistem pengenalan tanaman terong. Pra-pengolahan dalam penelitian ini dilakukan dengan 4 tahap, yaitu:

1. Pengubahan citra warna ke *grayscale*

Pra-pengolahan pertama yang akan dilakukan adalah merubah citra *training* atau citra *testing* yang awalnya berbentuk citra dari RGB (red, green, blue) menjadi citra bentuk grayscale, perubahan ini dilakukan karena citra grayscale memiliki persamaan yang sederhana dan mampu mengurangi kebutuhan memory dimana nilai warna putih diwakili dengan angka 255 dan nilai warna hitam diwakili dengan angka 0. Berikut gambar proses perubahan citra warna ke grayscale

Tabel 4.7 Proses Perubahan Citra Asli Menjadi *Grayscale*

Perbedaan	<i>Images Original</i>	<i>Images Grayscale</i>
-----------	------------------------	-------------------------

BUAH TERONG SEHAT		
BUAH TERONG BUSUK		

2. Normalisasi citra

Setelah citra asli di konversi ke citra abu-abu maka pra-pengolahan selanjutnya adalah normalisasi citra dengan histogram ekualisasi. Histogram ekualisasi adalah sebuah proses yang mengubah distribusi nilai derajat keabuan pada sebuah citra sehingga menjadi seragam. Tujuan dari histogram equalization adalah untuk memperoleh penyebaran histogram yang merata sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama. Berikut gambar normalisasi citra abu-abu dengan histogram ekualisasi.

Tabel 4.8 Proses Normalisasi Citra

Perbedaan	Citra <i>Grayscale</i>	Citra <i>Histogram</i>
-----------	------------------------	------------------------

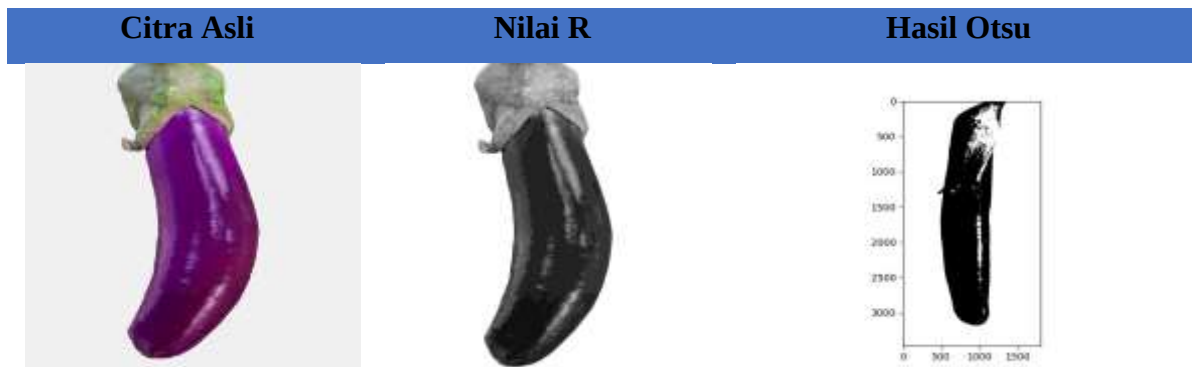
BUAH TERONG SEHAT		
BUAH TERONG BUSUK		

3. Segmentasi Citra

Segmentasi citra merupakan bagian dari proses pengolahan citra. Segmentasi citra (*image segmentation*) mempunyai arti membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel-piksel tetangganya, kemudian hasil dari proses segmentasi ini akan digunakan untuk proses lebih lanjut [8].

4. Metode Otsu *Thresholding*

Metode Otsu merupakan salah satu metode untuk segmentasi citra digital dengan menggunakan nilai ambang secara otomatis, yakni mengubah citra digital warna abu-abu menjadi hitam putih berdasarkan perbandingan nilai ambang dengan nilai warna piksel citra digital. Metode Otsu *Thresholding* diperkenalkan pertama kali oleh Nobuyuki Otsu, dalam jurnal ilmiahnya yang berjudul “A *Threshold Selection Method from Grayscale Histogram*” pada tahun 1979 [9].

Tabel 4.9 Hasil Segmentasi Otsu

4. 2. 2. Ekstraksi Fitur

Fitur ekstraksi Grey Level Co-occurrence Matrix (GLCM) merupakan matrix yang menggambarkan frekuensi munculnya pasangan dua pixel dengan intensitas tertentu dalam jarak d dan orientasi arah sudut 0° , 90° , 135° , atau 45° tertentu dalam citra.

Contoh: Jarak 1 dan orientasi arah sudut 0°

Nilai dari Gambar 4.10 berupa matrix 10×10 Yang sebenarnya matrix 104×104 dan di ambil dari 1 contoh gambar 1 dataset buah terong sehat sama seperti gambar pada Tabel 4.9 yang disederhanakan unruk memudahkan perhitungan :

Tabel 4.10 Nilai dari suatu citra

I/J	0	1	2	3	4	5	6	7	8	9
0	240	243	219	132	153	175	238	241	240	240
1	240	242	225	129	129	145	229	241	240	240
2	240	241	237	142	82	121	216	243	240	240
3	240	240	244	190	69	105	175	246	240	240
4	240	240	243	219	78	74	117	241	240	240
5	240	240	242	225	89	68	95	232	241	240
6	240	240	244	214	79	58	108	240	241	240
7	240	240	246	169	47	70	168	246	240	240
8	240	241	240	90	41	134	238	241	240	240
9	240	240	242	165	151	235	242	240	240	240

242	0	0	0	0	0	0	0	0	0	0	0	0
243	0	0	0	0	0	0	0	0	0	0	0	0
244	0	0	0	0	0	0	0	0	0	0	0	0
246	0	0	0	0	0	0	0	0	0	0	0	0

Setelah itu nilai kontras, homogeneity, entropy dan energy di hitung secara manual seperti di bawah ini:

a. Menghitung nilai *Contras*

Contras =

Tabel 4.12 Perhitungan Matrix kontras

$((41-90)^2 (0,00609756))$	+	$(41-134)^2 (0,00609756)$	+	$(41-190)^2 (0,00609756)$	+
$(47-70)^2 (0,00609756)$	+	$(47-169)^2 (0,00609756)$	+	$(58-79)^2 (0,00609756)$	+
$(58-108)^2 (0,00609756)$	+	$(68-89)^2 (0,00609756)$	+	$(68-95)^2 (0,00609756)$	+
$(69-105)^2 (0,00609756)$	+	$(69-190)^2 (0,00609756)$	+	$(70-47)^2 (0,00609756)$	+
$(70-168)^2 (0,00609756)$	+	$(74-78)^2 (0,00609756)$	+	$(74-117)^2 (0,00609756)$	+
$(78-74)^2 (0,00609756)$	+	$(78-219)^2 (0,00609756)$	+	$(79-58)^2 (0,00609756)$	+
$(79-214)^2 (0,00609756)$	+	$(82-121)^2 (0,00609756)$	+	$(82-142)^2 (0,00609756)$	+
$(89-68)^2 (0,00609756)$	+	$(89-225)^2 (0,00609756)$	+	$(90-41)^2 (0,00609756)$	+
$(90-240)^2 (0,006097561)$	+	$(95-68)^2 (0,00609756)$	+	$(95-232)^2 (0,00609756)$	+
$(105-69)^2 (0,00609756)$	+	$(105-175)^2 (0,00609756)$	+	$(108-58)^2 (0,00609756)$	+
$(108-240)^2 (0,006097561)$	+	$(117-74)^2 (0,00609756)$	+	$(117-241)^2 (0,00609756)$	+
$(121-82)^2 (0,00609756)$	+	$(121-216)^2 (0,00609756)$	+	$(129-129)^2 (0,00609756)$	+
$(129-145)^2 (0,00609756)$	+	$(129-225)^2 (0,00609756)$	+	$(132-153)^2 (0,00609756)$	+
$(132-219)^2 (0,00609756)$	+	$(134-41)^2 (0,00609756)$	+	$(134-238)^2 (0,00609756)$	+
$(142-82)^2 (0,00609756)$	+	$(142-237)^2 (0,00609756)$	+	$(145-121)^2 (0,00609756)$	+
$(145-229)^2 (0,00609756)$	+	$(151-165)^2 (0,00609756)$	+	$(151-235)^2 (0,00609756)$	+
$(153-132)^2 (0,00609756)$	+	$(153-175)^2 (0,00609756)$	+	$(165-151)^2 (0,00609756)$	+
$(165242)^2 (0,006097561)$	+	$(168-70)^2 (0,00609756)$	+	$(168-246)^2 (0,00609756)$	+
$(169-47)^2 (0,00609756)$	+	$(169-246)^2 (0,00609756)$	+	$(175-105)^2 (0,00609756)$	+
$(175-153)^2 (0,00609756)$	+	$(175-238)^2 (0,00609756)$	+	$(175-246)^2 (0,00609756)$	+
$(190-69)^2 (0,00609756)$	+	$(190-244)^2 (0,00609756)$	+	$(214-79)^2 (0,00609756)$	+
$(214-244)^2 (0,00609756)$	+	$(216-121)^2 (0,00609756)$	+	$(216-243)^2 (0,00609756)$	+

$(219-78)^2 (0,00609756)$	+	$(219-132)^2 (0,00609756)$	+	$(219-243)^2 (0,01219512)$	+
$(225-89)^2 (0,00609756)$	+	$(225-129)^2 (0,00609756)$	+	$(225-242)^2 (0,012195122)$	+
$(229-145)^2 (0,00609756)$	+	$(229-241)^2 (0,00609756)$	+	$(232-95)^2 (0,00609756)$	+
$(232-240)^2 (0,00609756)$	+	$(235-151)^2 (0,00609756)$	+	$(235-242)^2 (0,00609756)$	+
$(237-142)^2 (0,00609756)$	+	$(237-241)^2 (0,00609756)$	+	$(238-134)^2 (0,00609756)$	+
$(238-175)^2 (0,00609756)$	+	$(238-241)^2 (0,01219512)$	+	$(240-90)^2 (0,00609756)$	+
$(240-108)^2 (0,00609756)$	+	$(240-240)^2 (0,091463415)$	+	$(240-241)^2 (0,06097561)$	+
$(240-242)^2 (0,024390244)$	+	$(240-243)^2 (0,01829268)$	+	$(240-244)^2 (0,01219512)$	+
$(240-246)^2 (0,01829268)$	+	$(241-117)^2 (0,00609756)$	+	$(241-229)^2 (0,00609756)$	+
$(241-232)^2 (0,00609756)$	+	$(241-237)^2 (0,00609756)$	+	$(241-238)^2 (0,01219512)$	+
$(241-240)^2 (0,06097561)$	+	$(242-165)^2 (0,00609756)$	+	$(242-225)^2 (0,01219512)$	+
$(242+235)^2 (0,00609756)$	+	$(242+240)^2 (0,024390244)$	+	$(243+216)^2 (0,0060975)$	+
$(243-219)^2 (0,01219512)$	+	$(243-240)^2 (0,018292683)$	+	$(244-190)^2 (0,00609756)$	+
$(244-214)^2 (0,00609756)$	+	$(244-240)^2 (0,012195122)$	+	$(246-168)^2 (0,012195122)$	+
$(246-169)^2 (0,00609756)$	+	$(246-175)^2 (0,00609756)$	+	$(246-240)^2 (0,018292683)$	
Contras = 21.749085455693013					

b. Menghitung nilai *Energy*

Energy =

Tabel 4.13 Tabel Perhitungan Matrix Energy

$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+
$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+
$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+
$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+
$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+
$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+
$((0,006097561)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+	$((0,00609756)^2)$	+

c. Menghitung nilai *Homogeneity*

Homogeneity =

Tabel 4.14 Perhitungan Matrix Homogeneity

$((0,00609756) / 1 + (41-90))$	+	$((0,00609756) / 1 + (41-190))$	+
$((0,00609756) / 1 + (47-70))$	+	$((0,00609756) / 1 + (47-169))$	+
$((0,00609756) / 1 + (58-79))$	+	$((0,00609756) / 1 + (58-108))$	+
$((0,00609756) / 1 + (68-89))$	+	$((0,00609756) / 1 + (68-95))$	+
$((0,00609756) / 1 + (69-105))$	+	$((0,00609756) / 1 + (69-190))$	+
$((0,00609756) / 1 + (70-47))$	+	$((0,00609756) / 1 + (70-168))$	+
$((0,00609756) / 1 + (74-78))$	+	$((0,00609756) / 1 + (74-117))$	+
$((0,00609756) / 1 + (78-74))$	+	$((0,00609756) / 1 + (78-219))$	+
$((0,00609756) / 1 + (79-58))$	+	$((0,00609756) / 1 + (79-214))$	+
$((0,00609756) / 1 + (82-121))$	+	$((0,00609756) / 1 + (82-142))$	+
$((0,00609756) / 1 + (89-68))$	+	$((0,00609756) / 1 + (89-225))$	+
$((0,00609756) / 1 + (90-41))$	+	$((0,006097561) / 1 + (90-240))$	+
$((0,00609756) / 1 + (95-68))$	+	$((0,00609756) / 1 + (95-232))$	+
$((0,00609756) / 1 + (105-69))$	+	$((0,00609756) / 1 + (105-175))$	+
$((0,00609756) / 1 + (108-58))$	+	$((0,006097561) / 1 + (108-240))$	+
$((0,00609756) / 1 + (117-74))$	+	$((0,00609756) / 1 + (117-241))$	+
$((0,00609756) / 1 + (121-82))$	+	$((0,00609756) / 1 + (121-216))$	+
$((0,00609756) / 1 + (129-129))$	+	$((0,00609756) / 1 + (129-145))$	+
$((0,00609756) / 1 + (129-225))$	+	$((0,00609756) / 1 + (132-153))$	+
$((0,00609756) / 1 + (132-219))$	+	$((0,00609756) / 1 + (134-41))$	+
$((0,00609756) / 1 + (134-238))$	+	$((0,00609756) / 1 + (142-82))$	+

$((0,00609756)/1 + (142-237))$	+	$((0,00609756)/1 + (145-121))$	+
$((0,00609756)/1 + (145-229))$	+	$((0,00609756)/1 + (151-165))$	+
$((0,00609756)/1 + (151-235))$	+	$((0,00609756)/1 + (153-132))$	+
$((0,00609756)/1 + (153-175))$	+	$((0,00609756)/1 + (165-151))$	+
$((0,006097561)/1 + (165-242))$	+	$((0,00609756)/1 + (168-70))$	+
$((0,00609756)/1 + (168-246))$	+	$((0,00609756)/1 + (169-47))$	+
$((0,00609756)/1 + (169-246))$	+	$((0,00609756)/1 + (175-105))$	+
$((0,00609756)/1 + (175-153))$	+	$((0,00609756)/1 + (175-238))$	+
$((0,00609756)/1 + (175-246))$	+	$((0,00609756)/1 + (190-69))$	+
$((0,00609756)/1 + (190-244))$	+	$((0,00609756)/1 + (214-79))$	+
$((0,00609756)/1 + (214-244))$	+	$((0,00609756)/1 + (216-121))$	+
$((0,00609756)/1 + (216-243))$	+	$((0,00609756)/1 + (219-78))$	+
$((0,00609756)/1 + (219-132))$	+	$((0,01219512)/1 + (219-243))$	+
$((0,00609756)/1 + (225-89))$	+	$((0,00609756)/1 + (225-129))$	+
$((0,012195122)/1 + (225-242))$	+	$((0,00609756)/1 + (229-145))$	+
$((0,00609756)/1 + (229-241))$	+	$((0,00609756)/1 + (232-95))$	+
$((0,00609756)/1 + (232-240))$	+	$((0,00609756)/1 + (235-151))$	+
$((0,00609756)/1 + (235-242))$	+	$((0,00609756)/1 + (237-142))$	+
$((0,00609756)/1 + (237-241))$	+	$((0,00609756)/1 + (238-134))$	+
$((0,00609756)/1 + (238-175))$	+	$((0,01219512)/1 + (238-241))$	+
$((0,00609756)/1 + (240-90))$	+	$((0,00609756)/1 + (240-108))$	+
$((0,091463415)/1 + (240-240))$	+	$((0,06097561)/1 + (240-241))$	+
$((0,024390244)/1 + (240-242))$	+	$((0,01829268)/1 + (240-243))$	+
$((0,01219512)/1 + (240-244))$	+	$((0,01829268)/1 + (240-246))$	+

[illegible]

$(0,00609756 \cdot \log 0,00609756) + (0,01219512 \cdot \log 0,01219512)$	+
$(0,00609756 \cdot \log 0,00609756) + (0,00609756 \cdot \log 0,00609756)$	+
$(0,091463415 \cdot \log 0,091463415) + (0,06097561 \cdot \log 0,06097561)$	+
$(0,024390244 \cdot \log 0,024390244) + (0,01829268 \cdot \log 0,01829268)$	+
$(0,01829268 \cdot \log 0,01829268) + (0,01219512 \cdot \log 0,01219512)$	+
$(0,06097561 \cdot \log 0,06097561) + (0,06097561 \cdot \log 0,06097561)$	+
$(0,06097561 \cdot \log 0,06097561) + (0,06097561 \cdot \log 0,06097561)$	+
$(0,01219512 \cdot \log 0,01219512) + (0,06097561 \cdot \log 0,06097561)$	+
$(0,06097561 \cdot \log 0,06097561) + (0,01219512 \cdot \log 0,01219512)$	+
$(0,06097561 \cdot \log 0,06097561) + (0,024390244 \cdot \log 0,024390244)$	+
$(0,06097561 \cdot \log 0,06097561) + (0,01219512 \cdot \log 0,01219512)$	+
$(0,018292683 \cdot \log 0,018292683) + (0,06097561 \cdot \log 0,06097561)$	+
$(0,06097561 \cdot \log 0,06097561) + (0,012195122 \cdot \log 0,012195122)$	+
$(0,012195122 \cdot \log 0,012195122) + (0,06097561 \cdot \log 0,06097561)$	+
$(0,06097561 \cdot \log 0,06097561) + (0,018292683 \cdot \log 0,018292683)$	
Entropy = 2.6194496180209876	

4. 2. 3. Klasifikasi

Klasifikasi pada data yang di gunakan yaitu berupa 94 data training dan 10 data testing, sehingga ketika di akumulasi keseluruhan data adalah berjumlah 104. Masing-masing gambar tersebut memiliki nilai kontras, homogeneity, entropy, dan energy dengan perhitungan seperti yang telah peneliti jabarkan di atas. Adapun nilai-nilai tersebut dapat peneliti jabarkan keumumannya seperti tabel di bawah ini :

Tabel 4.16 Hasil Perhitungan Data Training

No.	Data Training	Con(X1)	Ene(X2)	Hom(X3)	Ent(X4)	Y
-----	---------------	---------	---------	---------	---------	---

0		21.7490854	0.72801189	0.84061847	2.6194496	BUAH SEHAT
1		21.9641992	0.70963036	0.81763944	2.8432143	BUAH SEHAT
2		30.2559686	0.67707774	0.80777390	3.0374986	BUAH SEHAT
3		31.2835902	0.53705316	0.64513808	4.1230187	DAUN SEHAT
4		10.6563662	0.67988500	0.83112720	2.8019364	BATANG SEHAT

5		105.754092	0.49345121	0.61800362	4.9140225	BUSUK BUAH
6		88.2725508	0.70375441	0.77699165	3.2357522	BUSUK BUAH
7		68.2720805	0.69297277	0.76081079	3.2369949	BUSUK BUAH
8		54.9083333	0.72512901	0.78159023	3.0140065	LAYU BAKTERI

9		77.5588220	0.44892668	0.55005456	5.2860918	ANTRAK NOSA
10		91.1732300	0.41087222	0.50949430	5.6650069	BERCAK DAUN
11		60.5610581	0.40794928	0.53652663	5.6638500	REBAI SEMAI
12		77.9387539	0.70816315	0.76237723	3.2251883	BATANG BUSUK
13		58.4113559	0.28873434	0.47787627	6.4225300	JAMUR TEPUNG

0		19.1701279	0.67831740	0.80919705	3.0657291	?
	Gambar Testing					

Pada tabel 4.14 ini yaitu hasil nilai dari 10 dataset tersebut di dapatkan dengan jarak 1 dan arah 0^0 .

Tabel 4.17 Hasil Perhitungan Data Testing

No.	Data Testing	X1	X2	X3	X4	Y
0	 Gambar Testing	19.170127	0.67831740	0.80919705	3.0657291	BUAH SEHAT
1		137.12135	0.37726033	0.60703256	4.8161967	DAUN SEHAT
2		52.656792	0.76528795,	0.82237449	2.3885275	BATANG SEHAT

3		65.169269	0.67026060	0.77163232	3.2639634	BUAH BUSUK
4		42.337856	0.70757127	0.78383285	3.0512440	LAYU BAKTERI
5		72.613706	0.52967717	0.61684009	4.8492053	ANTRAK NOSA
6		128.77369	0.37333583	0.47486795,	6.1538777 6	BERCAK DAUN
7		49.178880	0.36638564	0.53246361	5.8446273 1	REBAI SEMAI

8		76.603716	0.72255098	0.78505167	2.9744882 4	BATANG BUSUK
9		80.609181	0.33236526	0.44731499	6.2044568	JAMUR TEPUNG

Berikut ini adalah tahapan untuk menghitung nilai Euclidean distance dan menentukan nilai k. tahapan-tahapan tersebut seperti di bawah ini :

1. Tentukan parameter k
2. Hitung jarak antara data yang akan dievaluasi dengan semua pelatihan
3. Urutkan jarak yang terbentuk (urut naik)
4. Tentukan jarak terdekat sampai urutan k
5. Pasangkan kelas yang bersesuaian
6. Cari jumlah kelas dari tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi.

Euclidean Distance

$$1. d = \sqrt{(21.7490854 - 19.1701279)^2 + (0.72801189 - 0.67831740)^2 + (0.84061847 - 0.80919705)^2 + (2.6194496 - 3.0657291)^2}$$

$$d = \sqrt{6.6507 + 0.0024 + 0.0012 + 0.1991}$$

$$= 2.617$$

$$2. d = \sqrt{(21.9641992 - 19.1701279)^2 + (0.70963036 - 0.67831740)^2 + (0.81763944 - 0.80919705)^2 + (2.8432143 - 3.0657291)^2}$$

$$d = \sqrt{7.7807 + 0.00098 + 0.00007 + 0.059} \\ = 2.80$$

$$3. d = \sqrt{(30.2559686 - 19.1701279)^2 + (0.67707774 - 0.67831740)^2 + (0.80777390 - 0.80919705)^2 + (3.0374986 - 3.0657291)^2}$$

$$d = \sqrt{122.2897 + 0.000001 + 0.000002 + 0.0008} \\ = 11.06$$

$$4. d = \sqrt{(31.2835902 - 19.1701279)^2 + (0.53705316 - 0.67831740)^2 + (0.64513808 - 0.80919705)^2 + (4.1230187 - 3.0657291)^2}$$

$$d = \sqrt{146.734 + 0.01996 + 0.0269 + 1.1178} \\ = 12.16$$

$$5. d = \sqrt{(10.6563662 - 19.1701279)^2 + (0.67988500 - 0.67831740)^2 + (0.83112720 - 0.80919705)^2 + (2.8019364 - 3.0657291)^2}$$

$$d = \sqrt{72.4847 + 0.0000023 + 0.0004 + 0.0695} \\ = 8.52$$

$$6. d = \sqrt{(105.754092 - 19.1701279)^2 + (0.49345121 - 0.67831740)^2 + (0.61800362 - 0.80919705)^2 + (4.9140225 - 3.0657291)^2}$$

$$d = \sqrt{86.5839 + 0.0341 + 0.03651 + 3.4162} \\ = 9.49$$

$$7. d = \sqrt{(88.2725508 - 19.1701279)^2 + (0.70375441 - 0.67831740)^2 + (0.77699165 - 0.80919705)^2 + (3.2357522 - 3.0657291)^2}$$

$$d = \sqrt{4,775.14 + 0.00064 + 0.001 + 0.0289} \\ = 69.10$$

$$8. d = \sqrt{(68.2720805 - 19.1701279)^2 + (0.69297277 - 0.67831740)^2 + (0.76081079 - 0.80919705)^2 + (3.2369949 - 3.0657291)^2}$$

$$d = \sqrt{2,411.0064 + 0.0002 + 0.0002 + 0.0293} \\ = 49.10$$

$$9. d = \sqrt{(3.2369949 - 19.1701279)^2 + (0.72512901 - 0.67831740)^2 + (0.78159023 - 0.80919705)^2 + (3.0140065 - 3.0657291)^2}$$

$$d = \sqrt{253.86 + 0.0021 + 0.00076 + 0.0027}$$

$$= 15.93$$

$$10. d = \sqrt{(77.5588220 - 19.1701279)^2 + (0.44892668 - 0.67831740)^2 + (0.55005456 - 0.80919705)^2 + (5.2860918 - 3.0657291)^2}$$

$$d = \sqrt{134.8228 + 0.0526 + 0.0671 + 4.9297}$$

$$= 11.83$$

$$11. d = \sqrt{(91.1732300 - 19.1701279)^2 + (0.41087222 - 0.67831740)^2 + (0.50949430 - 0.80919705)^2 + (5.6650069 - 3.0657291)^2}$$

$$d = \sqrt{5,184.44 + 0.072 + 0.0898 + 6.7563}$$

$$= 72.05$$

$$12. d = \sqrt{(60.5610581 - 19.1701279)^2 + (0.40794928 - 0.67831740)^2 + (0.53652663 - 0.80919705)^2 + (5.6638500 - 3.0657291)^2}$$

$$d = \sqrt{1,713.1321 + 0.0731 + 0.0743 + 6.750}$$

$$= 41.47$$

$$13. d = \sqrt{(77.9387539 - 19.1701279)^2 + (0.70816315 - 0.67831740)^2 + (0.76237723 - 0.80919705)^2 + (3.2251883 - 3.0657291)^2}$$

$$d = \sqrt{3,452.7376 + 0.0008 + 0.0021 + 0.025}$$

$$= 58.76$$

$$14. d = \sqrt{(58.4113559 - 19.1701279)^2 + (0.28873434 - 0.67831740)^2 + (0.47787627 - 0.80919705)^2 + (6.4225300 - 3.0657291)^2}$$

$$d = \sqrt{1539.87 + 0.1517 + 0.1097 + 11.2681}$$

$$= 39.38$$

Tabel 4.18 Hasil Pengurutan dan Nilai Terkecil Berdasarkan Perhitungan Jarak dari Masing-masing Data Training ke Data Testing

DATA	EUCLIDEAN	KATEGORI
0	2.617	BUAH SEHAT
1	2.80	BUAH SEHAT
4	8.52	BATANG SEHAT
5	9.49	BUAH BUSUK
2	11.06	BUAH SEHAT
9	11.83	ANTRAKNOSA
3	12.16	DAUN SEHAT
8	15.93	LAYU BAKTERI
13	39.38	JAMUR TEPUNG
11	41,47	REBAI SEMAI
7	49,10	BUAH BUSUK
12	58,76	BATANG BUSUK
6	69.10	BUAH BUSUK
10	72.05	BERCAK DAUN

Tabel 4.19 Hasil Jumlah K yang Sesuai Data

DATA	EUCLIDEAN	KATEGORI
0	2.617	BUAH SEHAT
1	2.80	BUAH SEHAT
4	8.52	BATANG SEHAT

Jumlah K = 3 Sesuai Yang Ditentukan Sebelumnya

KATEGORI	JUMLAH DATA
BUAH SEHAT	2
BATANG SEHAT	1

Tetangga Paling Banyak Muncul = BUAH SEHAT

4. 3.Evaluasi

4. 3. 1. Confusion Matrix

Pada penelitian ini menggunakan confusion Matrix sebagai metode. Dalam perhitungan akurasi pada penerapan deteksi penyakittanaman terong menggunakan tekstur warna. Evaluasi kinerja deteksi penyakit tanaman terong di dasarkan dari jumlah pengujian objek.

Tabel 4.20 Hasil Data Testing

No	Data Testing	DATA AKTUAL	HASIL KLASIFIKASI
0		BUAH SEHAT	TANAMAN SEHAT
1		DAUN SEHAT	TANAMAN SEHAT
2		BATANG SEHAT	TANAMAN SEHAT
3		BUAH BUSUK	BUAH BUSUK
4		LAYU BAKTERI	LAYU BAKTERI

5		ANTRAKNOSA	ANTRAKNOSA
6		BERCAK DAUN	BUAH BUSUK
7		REBAI SEMAI	TANAMAN SEHAT
8		BATANG BUSUK	BATANG BUSUK
9		JAMUR TEPUNG	LAYU BAKTERI

Berdasarkan hasil data uji di atas maka peneliti menemukan bahwa jumlah klasifikasi penyakit terong yang benar adalah 7 sedangkan jumlah klasifikasi penyakit terong yang salah adalah 3. Adapun tingkat akurasi ini dapat dilihat melalui tabel Confusion Matrix di bawah ini.

TABEL 4.21 confusion Matrix

N=10 Glem 0°, jarak 1, k=1	Aktual : Positif (1)	Aktual : Negatif (0)
Prediksi : Akurat (1)	True Positive : 5	False Positive : 1
Prediksi : Akurat (0)	False Negative : 2	True Negative : 2
N=10 Glem 0°, jarak 1, k=3	Aktual : Positif (1)	Aktual : Negatif (0)
Prediksi : Akurat (1)	True Positive : 4	False Positive : 0
Prediksi : Akurat (0)	False Negative : 3	True Negative : 3
N=10 Glem 135°, jarak 1, k=1	Aktual : Positif (1)	Aktual : Negatif (0)
Prediksi : Akurat (1)	True Positive : 5	False Positive : 1
Prediksi : Akurat (0)	False Negative : 2	True Negative : 2
N=10 Glem 135°, jarak 1, k=3	Aktual : Positif (1)	Aktual : Negatif (0)
Prediksi : Akurat (1)	True Positive : 4	False Positive : 0
Prediksi : Akurat (0)	False Negative : 3	True Negative : 3

True Positive (TP)

Merupakan data positif yang diprediksi benar. Contohnya, data terong menderita penyakit buah busuk dan dari uji data test yang dilakukan memprediksi data terong tersebut memiliki penyakit buah busuk yang sama.

True Negative (TN)

Merupakan data negatif yang diprediksi benar. Contohnya, data terong tidak memiliki penyakit dan dari data test yang dilakukan menunjukkan bahwa data tersebut merupakan tanaman yang sehat.

False Postive (FP) — Type I Error

Merupakan data negatif namun diprediksi sebagai data positif. Contohnya, data uji test tersebut merupakan tanaman sehat namun dari hasil pengujian memprediksi data test tersebut menderita penyakit layu bakteri atau yang lainnya.

False Negative (FN) — Type II Error

Merupakan data positif namun diprediksi sebagai data negatif. Contohnya, data test tersebut memiliki penyakit terong namun dari hasil prediksi menunjukkan data tersebut tidak memiliki penyakit atau memiliki penyakit namun dengan jenis yang berbeda. Sehingga melihat dari tabel confusion matrix di atas maka dilakukan perhitungan akurasi sebagai berikut:

GlcM 0° & GlcM 135° jarak 1, k=1	GlcM 0° & GlcM 135° jarak 1, k=3
$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} (100\%)$ $= \frac{5+2}{5+2+1+2} (100\%)$ $= \frac{7}{10} (100\%) = 70\%$	$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} (100\%)$ $= \frac{4+3}{4+3+0+3} (100\%)$ $= \frac{7}{10} (100\%) = 70\%$

BAB V PEMBAHASAN

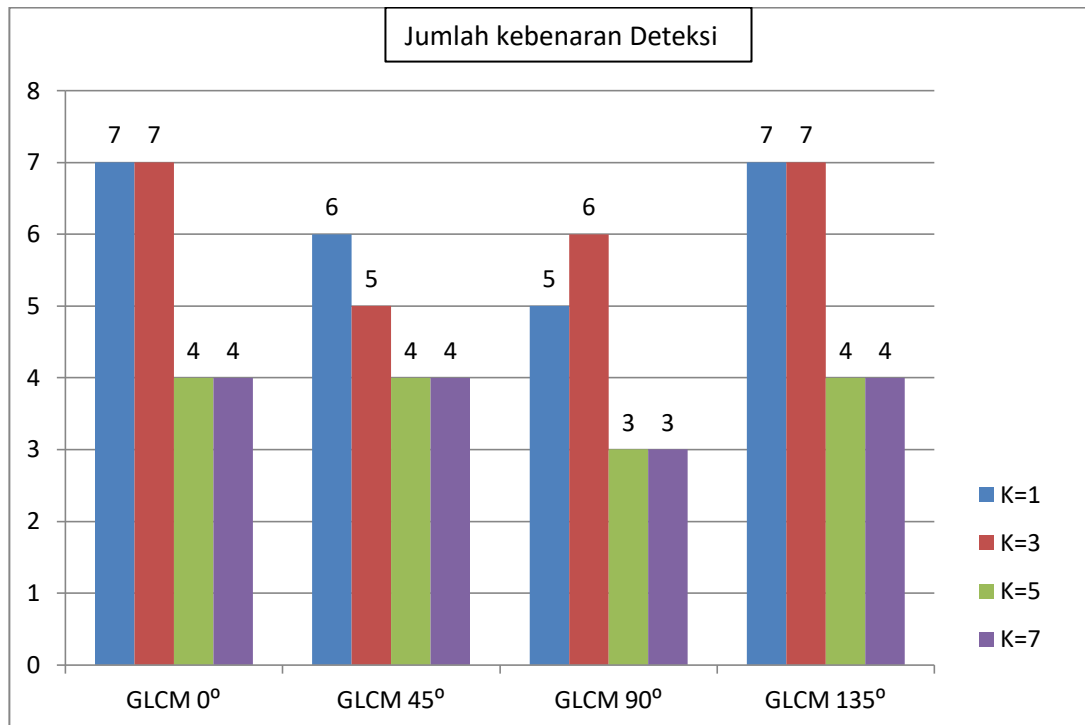
5.1. Pembahasan Model

Tabel 5.1 Hasil ujicoba 10 data testing dengan Glcm dan K yang berbeda

Catatan: Y= Benar, X= Salah

GLCM 0° Jarak = 1	K=1	K=3	K=5	K=7	GLCM 45° Jarak = 1	K=1	K=3	K=5	K=7
Data 0	X	Y	Y	Y	Data 0	Y	Y	Y	Y
Data 1	Y	Y	Y	Y	Data 1	Y	Y	Y	Y
Data 2	Y	Y	X	Y	Data 2	Y	Y	Y	Y
Data 3	Y	Y	Y	X	Data 3	X	X	X	X
Data 4	Y	Y	X	X	Data 4	Y	X	X	X
Data 5	X	X	X	X	Data 5	X	X	X	X
Data 6	Y	Y	Y	Y	Data 6	X	X	X	X
Data 7	X	X	X	X	Data 7	Y	Y	Y	Y
Data 8	Y	Y	X	Y	Data 8	Y	Y	X	X
Data 9	Y	X	X	X	Data 9	X	X	X	X
Total Y	7	7	4	5	Total Y	6	5	4	4
Total X	3	3	6	5	Total X	4	5	6	6
GLCM 90° Jarak = 1	K=1	K=3	K=5	K=7	GLCM 135° Jarak = 1	K=1	K=3	K=5	K=7
Data 0	Y	Y	Y	Y	Data 0	X	Y	Y	Y
Data 1	Y	Y	Y	Y	Data 1	Y	Y	Y	Y
Data 2	Y	Y	X	Y	Data 2	Y	Y	X	X
Data 3	Y	Y	X	X	Data 3	Y	Y	Y	X
Data 4	X	X	X	X	Data 4	Y	Y	X	X
Data 5	Y	Y	X	X	Data 5	X	X	X	X
Data 6	X	X	Y	X	Data 6	Y	Y	Y	Y
Data 7	X	Y	X	X	Data 7	X	X	X	X
Data 8	X	X	X	X	Data 8	Y	Y	X	Y
Data 9	X	X	X	X	Data 9	Y	X	X	X

Total Y	5	6	3	3	Total Y	7	7	4	4
Total X	5	4	7	7	Total X	3	3	6	6



GlcM 0° & Glcm 135° jarak 1, k=1

Accuracy=

$$\frac{TP+TN}{TP+TN+FP+FN} (100\%)$$

$$= \frac{5 + 2}{5 + 2 + 1 + 2} (100\%)$$

$$= \frac{7}{10} (100\%) = 70\%$$

GlcM 0° & Glcm 135° jarak 1, k=3

Accuracy=

$$\frac{TP+TN}{TP+TN+FP+FN} (100\%)$$

$$= \frac{4 + 3}{4 + 3 + 0 + 3} (100\%)$$

$$= \frac{7}{10} (100\%) = 70\%$$

Hasil Probabilitas tertinggi terjadi pada Glcm 0°, jarak 1, k=1 mendapatkan hasil sebesar 70%, dan Glcm 135° jarak 1, k=1 dengan hasil 70% Glcm 0° jarak1, k=3 dengan hasil 70% dan Glcm 135° jarak 1, k=3 dengan hasil 70%.

Hasil evaluasi menggunakan Confungtion matrix, metapkan bahwa hasil terbaik dengan tingkat probabilitas kebenaran tertinggi terletak pada Glcm 0°, jarak 1, k=1 mendapatkan hasil sebesar 70%, dan Glcm 135° jarak 1, k=1 dengan hasil 70% .

Glcm 0° & Glcm 135° jarak 1, k=1
$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} (100\%)$ $= \frac{5+2}{5+2+1+2} (100\%)$ $= \frac{7}{10} (100\%) = 70\%$

Pada percobaan Berdasarkan tabel di atas menggunakan perhitungan confungtion Matrix maka hasil deteksi sebagai berikut :

1. GLCM 0°

- GLCM 0° menggunakan jarak=1 dan nilai K=1 mendapatkan hasil = 70%
- GLCM 0° menggunakan jarak=1 dan nilai K=3 mendapatkan hasil = 70%
- GLCM 0° menggunakan jarak=1 dan nilai K=5 mendapatkan hasil = 40%
- GLCM 0° menggunakan jarak=1 dan nilai K=7 mendapatkan hasil = 50%

2. GLCM 45°

- GLCM 45° menggunakan jarak=1 dan nilai K=1 mendapatkan hasil = 60%
- GLCM 45° menggunakan jarak=1 dan nilai K=3 mendapatkan hasil = 50%
- GLCM 45° menggunakan jarak=1 dan nilai K=5 mendapatkan hasil =40%

- d. GLCM 45° menggunakan jarak=1 dan nilai K=7 mendapatkan hasil =40%
- 3. GLCM 90°
 - a. GLCM 90° menggunakan jarak=1 dan nilai K=1 mendapatkan hasil =50%
 - b. GLCM 90° menggunakan jarak=1 dan nilai K=3 mendapatkan hasil =60%
 - c. GLCM 90° menggunakan jarak=1 dan nilai K=5 mendapatkan hasil =30%
 - d. GLCM 90° menggunakan jarak=1 dan nilai K=7 mendapatkan hasil =30%
- 4. GLCM 135°
 - a. GLCM 135° menggunakan jarak=1 dan nilai K=1 mendapatkan hasil =70%
 - b. GLCM 135° menggunakan jarak=1 dan nilai K=3 mendapatkan hasil =70%
 - c. GLCM 135° menggunakan jarak=1 dan nilai K=5 mendapatkan hasil =40%
 - d. GLCM 135° menggunakan jarak=1 dan nilai K=7 mendapatkan hasil =40%

Point di atas berdasarkan hasil deteksi dengan data training = 94 dan data testing 10. Pada tahap pengujian metode peneliti mendapatkan kendala di dalam pengumpulan sampel untuk data training, dimana pada beberapa kategori penyakit terong tidak memungkinkan untuk menemukan data lebih dikarenakan cuaca hujan pada saat pengambilan sampel. Hal ini membuat kategori penyakit seperti jamur tepung, rebai semai, antraknosa dan bercak daun sedikit lebih sulit untuk di dapatkan.

Adapun dalam penelitian ini penggunaan sampel penyakit terong telah penulis klasifikasikan juga berdasarkan tanaman sehat. Hal ini dimaksudkan agar ketika pengambilan testing maka tanaman sehat tidak akan diklasifikasikan pada penyakit terong. Adapun kendala selanjutnya dalam pengolahan citra pada penelitian ini adalah harus memperhatikan dari kontras dan posisi dari data

sampel, karenan ada beberapa sampel data yang tidak biasa dikenali atau tidak terdeteksi.

Lebih lanjut lagi kontribusi dari arah dan jarak dari GLCM menyumbang kontribusi yang cukup berpengaruh dalam mengenai penyakit terong. Sehingga hasil yang didapatkan berdasarkan pengujian aplikasi adalah mayoritas derajat dengan probabilitas terbesar terletak pada $k=1$ dan $K=3$ jarak 1 $Glc_m=0^\circ$, serta pada $k=1$ dan $K=3$ jarak 1 sudut 135° .

berdasarkan hasil evaluasi Confusion Matrix maka di dapatkan probabilitas kebenaran terbesar terletak pada $K=1$, jarak=1, $Glc_m=0^\circ$ dan $K=1$, jarak=1, $Glc_m=135^\circ$ dengan Hasil evaluasi yang sama yaitu 70%.

5. 2. Pembahasan Sistem

1. Tampilan Menu Utama



Gambar 5.1Tampilan Menu Utama

Tampilan pertama saat program dijalankan

2. Tampilan Browse Data Training

Halaman ini adalah tampilan jika pengguna menekan tombol browse untuk data training.

3. Tampilan Browse Data Testing

Halaman ini adalah tampilan jika pengguna menekan tombol browse untuk data testing.

Tampilan ini akan muncul saat pengguna memilih/mengklik *browser* pada *ImageTesting*, dan pengguna akan memilih folder yang tersimpan gambar.

4. Tampilan Setelah Pemilihan Folder *Traning* dan data *Testing*

Tampilan ini akan memunculkan nama folder yang telah dipilih pengguna.

5. Tampilan Pemasukan Nilai GLCM

Tampilan ini akan muncul saat pengguna telah memasukan nilai dan memilih perhitungan pada GLCM.

6. Tampilan Pemasukan Nilai K

Tampilan ini akan muncul saat pengguna telah memasukan nilai K.

7. Tampilan Hasil Kualitas Penyakit Terong

Tampilan ini akan muncul saat pengguna mengklik/memilih tombol tau button proses setelah menginput data training, data testing, arah dan jarak GLCM serta nilai k. Tampilan ini adalah tampilan hasil pemrosesan.

8. Tampilan Hasil Klasifikasi Dan Nilai Akurasi

Tampilan ini akan muncul bersamaan dengan tampilan hasil deteksi penyakit terong.

BAB VI

PENUTUP

6. 1. Kesimpulan

Melihat hasil penelitian yang di lakukan serta pembahasan yang telah di uraikan maka dapat ditarik suatu kesimpulan bahwa :

1. hasil penerapan GLCM dan K-NN untuk proses deteksi penyakit pada terong. Menggunakan tekstur warna dalam tingkat akurasinya mendapatkan nilai akurasi kebenaran terbesar dalam penelitian ini adalah $K=1$, jarak=1, $Glcm=0^{\circ}$ dan $K=1$, jarak=1, $Glcm=135^{\circ}$ dengan Hasil evaluasi yang sama yaitu 70%. Hasil yang di dapatkan dari pengujian metode untuk mendeteksi penyakit tanaman terong dengan akurasi yang diperoleh di rasa cukup baik dikarenakan probabilitas kebenaran tertinggi lebih dari 50% yaitu hasil mendeteksi lebih banyak benar di bandingkan hasil deteksi yang salah menggunakan $K=1$, jarak=1, $Glcm=0^{\circ}$ dan $K=1$, jarak=1, $Glcm=135^{\circ}$ dengan Hasil 70% , dari jumlah dataset 104, berupa 94 data training dan 10 data testing.
2. Hasil penerapan metode GLCM dan K-NN pada penelitian ini mendapatkan hasil akurasi dimana :
 1. GLCM 0°
 - a. GLCM 0° menggunakan jarak=1 dan nilai $K=1$ mendapatkan hasil = 70%
 - b. GLCM 0° menggunakan jarak=1 dan nilai $K=3$ mendapatkan hasil = 70%
 - c. GLCM 0° menggunakan jarak=1 dan nilai $K=5$ mendapatkan hasil = 40%
 - d. GLCM 0° menggunakan jarak=1 dan nilai $K=7$ mendapatkan hasil = 50%

2. GLCM 45°

- a. GLCM 45° menggunakan jarak=1 dan nilai K=1 mendapatkan hasil = 60%
- b. GLCM 45° menggunakan jarak=1 dan nilai K=3 mendapatkan hasil = 50%
- c. GLCM 45° menggunakan jarak=1 dan nilai K=5 mendapatkan hasil =40%
- d. GLCM 45° menggunakan jarak=1 dan nilai K=7 mendapatkan hasil =40%

3. GLCM 90°

- a. GLCM 90° menggunakan jarak=1 dan nilai K=1 mendapatkan hasil =50%
- b. GLCM 90° menggunakan jarak=1 dan nilai K=3 mendapatkan hasil =60%
- c. GLCM 90° menggunakan jarak=1 dan nilai K=5 mendapatkan hasil =30%
- d. GLCM 90° menggunakan jarak=1 dan nilai K=7 mendapatkan hasil =30%

4. GLCM 135°

- a. GLCM 135° menggunakan jarak=1 dan nilai K=1 mendapatkan hasil =70%
- b. GLCM 135° menggunakan jarak=1 dan nilai K=3 mendapatkan hasil =70%
- c. GLCM 135° menggunakan jarak=1 dan nilai K=5 mendapatkan hasil =40%
- d. GLCM 135° menggunakan jarak=1 dan nilai K=7 mendapatkan hasil =40%

6.2 Saran

Setelah melakukan Penelitian deteksi penyakit terong, ada beberapa saran yang perlu di perhatikan untuk mencapai tujuan yang di harapkan, yaitu

sebagai berikut:

1. Penulis berharap untuk penelitian selanjutnya agar mendapatkan hasil akurasi yang lebih baik.
2. Penulis berharap agar kualitas dataset berikutnya menjadi lebih baik lagi.
3. Penulis berharap agar penelitian selanjutnya memiliki peralatan yang memadai dan mumpuni untuk pengumpulan dataset sehingga hasilnya lebih maksimal.
4. Penulis berharap penelitian selanjutnya memiliki objek penelitian secara pribadi ataupun milik sanak saudara sehingga lebih leluasa menggunakan dan melakukan penelitian pada objek tersebut.
5. Penulis berharap untuk penelitian selanjutnya agar bisa memberikan klasifikasi pada penyakit terong maupun terong yang sehat serta dapat berkontribusi dalam kemajuan industri khususnya sektor pertanian melalui pengembangan penelitian ini.

DAFTAR PUSTAKA

- [1] A. Tsany and R. Dzaky, “Deteksi Penyakit Tanaman Cabai Menggunakan Metode *Convolutional Neural Network*,” vol. 8, no. 2, pp. 3039–3055, 2021.
- [2] Y. Agtiyan, “Sistem Pakar Diagnosa Penyakit Tanaman Terong Menggunakan Algoritma *Dempster-Shafer*,” pp.1–12, 2020.
- [3] E. F. S. Ndruru, “Sistem Pakar Mendiagnosis Hama dan Penyakit Pada Tanaman Kelapa Sawit,” vol. 5, no. 2, pp. 47–51, 2014.
- [4] I. P. Sari, “Perancangan dan Simulasi Deteksi Penyakit Tanaman Jagung Berbasis Pengolahan Citra Digital Menggunakan Metode *Color Moments* dan *GLCM*,” *Semin. Nas. Inov. Dan Apl. Teknol. Di Ind.*, pp. 215–220, 2016.
- [5] Pertanianku, "7 Penyakit Tanaman Terong yang Biasa Menyerang", October 4, 2018
- [6] P. Sukmandaru, “Identifikasi Penyakit Tanaman Tomat Berdasarkan Citra Daun Menggunakan Metode *Gray Level Co-Occurrence Matrix* (GLCM) Dan *K-Nearest Neighbor* (KNN),” 2019.
- [7] A. A. Paturrahman, “Fitur *Canny Edge Detection* Dan Fitur *GlcM* Menggunakan Metode Klasifikasi *K-Nearest*,” 2020.
- [8] L. S. H. Mhd. Furqan, Sriani, “Klasifikasi Daun Bugenvil Menggunakan *Gray Level Co-Occurrence Matrix* dan *K-Nearest Neighbor*,” *J. CoreIT*, vol.6, no. 1, pp. 22–29, 2020.
- [9] Mustika Mentari, Y. A. Sari, and R. K. Dewi, “Deteksi Kanker Kulit Melanoma dengan *Linear Discriminant Analysis-fazzy k-Nearest Neighbour Lp-Norm*,” 2016.
- [10] D. Buchori, “Riset Aksi Partisipatif dalam Pengendalian Hama Terpadu Tanaman Hortikultura: Model Pengembangan Kemandirian Petani dalam Praktek Pertanian

Ramah Lingkungan,” no. April, 2007.

- [11] Husnul abdi, “Hama dan Penyakit Tanaman serta Cara Mengatasinya yang Perlu Diketahui,” 2019.
- [12] E. K. Wowor, “Persentase Serangan *Leucinodes orbonalis* Gueene (*Lepidoptera*; *Crambidae*) Pada Buah Terong Di Kelurahan Wailan Dan Kakaskasen Dua Kecamatan Tomohon Utara,” no. April 2016, pp. 1–11.
- [13] R. P. NIM.10110448, “Pembangunan aplikasi deteksi dan tracking warna virtual drawing menggunakan algoritma color filtering,” 2015.
- [14] Mitalom, “Jenis dan Gejala Serangan Penyakit pada Tanaman Terong”, 29 July 2015
- [15] E. Miyamoto and T. M. Jr., “*Fast Calculation Of Haralick Texture Features Human Computer Interaction Institute Department of Electrical and Computer Engineering Carnegie Mellon University*,” *Hum. Comput. Interact. Inst. Dep. Electr. Comput. Eng. Carnegie Mellon Univ. Pittsburgh PA*, vol. 15213, pp. 1–6, 2011.
- [16] S. in E. Science, “*Technogenic Activity of Man and Local Sources of Environmental Pollution*,” 1981.
- [17] Y. Fernando, “Klasifikasi Jenis Daging Berdasarkan Analisis Citra Tekstur *Gray Level Co-Occurrence Matrices* (*Glcmm*) Dan Warna,” *Semin. Nas. Sains dan Teknol. 2017*, no. Fakultas Teknik Universitas Muhammadiyah Jakarta, pp. 1–7, 2017.
- [18] E. D. Handoyo, “Perancangan Mini Image Editor Versi 1.0 sebagai Aplikasi Penunjang Mata Kuliah Digital Image Processing,” *J. Inform.*, vol. 2, no. 2, pp. 145–154, 2006.
- [19] R. Santi, “Mengubah Citra Berwarna Menjadi Gray-Scale dan Citra Biner,”

None, vol. 16, no. 1, p. 243035, 2011.

- [20] A. W. Sanjaya, “Deteksi Penyakit Kulit Menggunakan Analisis Fitur Warna Dan Tekstur Dengan Metode Color Moment, *Gray Level Cooccurrence Matrix*, Dan Jaringan Saraf Tiruan *Backpropagation*,” 2011.
- [21] NIKOLAUS WESLI ROBERTO SITUMORANG, “PENERAPAN METODE K-NEAREST NEIGHBOR DALAM IDENTIFIKASI KESEGERAN IKAN,” 2019.
- [22] E. Alvansga, “*Texture Recognition Using Glcm Method*,” 2019.
- [23] Mahasiswa, “Pengujian Dengan Confusion Matrix,” 2018.
- [24] Lolita Valda Claudia, ”5 Bahan Alami yang Ampuh Membasmi Tungau Laba-laba,” 2022.
- [25] E. F. S. Ndruru, “Sistem Pakar Mendiagnosis Hama dan Penyakit Pada Tanaman Kelapa Sawit,” vol. 5, no. 2, pp. 47–51, 2014.
- [26] Try Sony(G64130020), Gishella Erdyaning (G64130040), Amaliya Sukma Ragil Pristiyanto (G64130044), Muhammad Rizqi (G64130091), and Dita Utami Putri (G64130092), “Deteksi Penyakit Daun Menggunakan *Artificial Neural Network* (ANN),” 2010

LAMPIRAN

Tabel 6.1. Rencana Kegiatan Penelitian.

No	Rencana Kegiatan	Bulan & Tahun 2021-2022								
		Mar	Apr	Mei	Jun	Jul	Ags		Jan	Feb
1	Pengajuan Judul									
2	Observasi									
3	Wawancara Serta Pengumpulan Data									
4	Penyusunan Proposal									
5	Wawancara Lanjutan									
6	Revisi Proposal									
7	Ujian Proposal									

Tabel 6.2 Matrix GLCM

I/J	41	47	58	68	69	70	74	78	79	82	89	90	95	105	108	117	121	129	132	134	142	145	151	153	165	168	169	175	190	214	216	219	225	229	232	235	237	238	240	241	242	243	244	246	
41	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
47	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
79	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
89	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
90	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
95	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
105	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
108	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
117	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
121	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
132	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
134	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
142	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
145	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

151	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0		
153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
165	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
168	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
169	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
175	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
190	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
214	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
216	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
219	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	
225	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	
229	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
232	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
235	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
237	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
238	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	
240	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	15	10	4	3	2	3		
241	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	2	10	0	0	0	0	0	0		
242	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	1	0	0	4	0	0	0	0	0	0		
243	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	3	0	0	0	0	0	0		
244	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	
246	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0		0	0	0	0	0	0	0	3	0	0	0	0	0	

KODE PROGRAM

```
import tkinter as tk
import tkinter.filedialog as fdialog
import tkinter.ttk as ttk
from PIL import Image, ImageTk
#from sklearn.preprocessing import OrdinalEncoder
import dataTraining
import cv2

class Aplication(tk.Frame):
    def __init__(self, parent):
        self.parent = parent
        self.frames()
        self.widget()

    def frames(self):
        #frame judul
        self.FrameAtas = tk.Frame(self.parent, width=1058,
height=150, background="#79f7c5", relief='raised')
        self.FrameAtas.grid(row=0, column=0, columnspan=3,
padx=4, pady=10, sticky="NSEW")
        self.FrameAtas.grid_propagate(0)

        #frame input gambar
        self.FrameKiri1 = tk.LabelFrame(self.parent, height=120,
background="#16fa9f", text="Input Gambar", font="Times",
fg="black")
        self.FrameKiri1.grid(row=1, column=0, padx=10,
sticky="NSEW")
        self.FrameKiri1.grid_propagate(0)

        #glcm
        self.FrameKiri2 = tk.LabelFrame(self.parent, height=120,
background="#16fa9f", text="Gray Level Co-Occurrence Matrix",
font="Times", fg="black")
        self.FrameKiri2.grid(row=2, column=0, padx=10, pady=10,
sticky="NSEW")
        self.FrameKiri2.grid_propagate(0)

        #knn
```

```

self.FrameKiri3 = tk.LabelFrame(self.parent, height=100,
background="#16fa9f", text="K - Nearest Neighbor", font="Times",
fg="black")
    self.FrameKiri3.grid(row=3, column=0, padx=10,
sticky="NSEW")
    self.FrameKiri3.grid_propagate(0)

    #proses
    self.FrameKiri4 = tk.Frame(self.parent, height=100,
background="#16fa9f")
    self.FrameKiri4.grid(row=5, column=0, padx=10, pady=6,
sticky="NSEW")
    self.FrameKiri4.grid_propagate(0)

    #hasil gambar
    self.FrameKanan5 = tk.LabelFrame(self.parent,
background="#16fa9f", font="Times", fg="black")
    self.FrameKanan5.grid(row=1, column=1, rowspan=2,
columnspan=2, padx=10, pady=2, sticky="NSEW")
    self.FrameKanan5.grid_propagate(0)

    #hasil akhir
    self.FrameKanan6 = tk.LabelFrame(self.parent,
background="#16fa9f", font="Times", fg="black", text="Hasil")
    self.FrameKanan6.grid(row=3, column=1, rowspan=3,
columnspan=2, padx=10, pady=9, sticky="NSEW")
    self.FrameKanan6.grid_propagate(0)

    #frame tambahan
    self.FrameKanan7 = tk.LabelFrame(self.parent,
background="#16fa9f", font="Times", fg="black",
text="Penanganan")
    self.FrameKanan7.grid(row=5, column=1, rowspan=3,
columnspan=2, padx=10, pady=9, sticky="NSEW")
    self.FrameKanan7.grid_propagate(0)

    #kaki
    self.FrameBawah = tk.LabelFrame(self.parent, height=30,
width=1020, background="#16fa9f")
    self.FrameBawah.grid(row=6, column=0, columnspan=3,
padx=10, pady=10, sticky="NSEW")
    self.FrameBawah.grid_propagate(0)

    def widget(self):

```

```

#frame
#logo
self.PathTestingImg = 'terongbg.jpg'
self.imT = Image.open(self.PathTestingImg)
self.imT = ImageTk.PhotoImage(self.imT)
self.label1 = tk.Label(self.FrameAtas, width=1058,
height=120, bg="green", image=self.imT)
self.label1.pack(side="top")

#frame 1
#label data training
self.label1 = tk.Label(self.FrameKiri1, text="Image
Training", font="Times", relief="ridge", width=12,
background="#b0f5d9")
self.label1.grid(row=3, column=0, sticky="NSEW", padx=8,
pady=5)

#textbox data training
self.nilai = tk.StringVar(self.FrameKiri1, value="")
self.txtrain = tk.Entry(self.FrameKiri1, textvariable =
self.nilai, width=23)
self.txtrain.grid(row=3, column=1, sticky="NSEW", pady=5)

#button browse training
self.path2 = tk.Button(self.FrameKiri1, text="Browse",
font="Times", width=6, background="#038a54",
command=lambda:self.nilai.set(fdialog.askdirectory()))
self.path2.grid(row=3, column=5, padx=5)

#label data testing
self.label2 = tk.Label(self.FrameKiri1, text="Image
Testing", font="Times", relief="ridge", background="#b0f5d9")
self.label2.grid(row=4, column=0, sticky="NSEW", padx=8,
pady=10)

#textbox data testing
self.nilai1 = tk.StringVar(self.FrameKiri1, value="")
self.txtest = tk.Entry(self.FrameKiri1, textvariable =
self.nilai1)
self.txtest.grid(row=4, column=1, sticky="NSEW", pady=10)

#button browse testing
self.path1 = tk.Button(self.FrameKiri1, text="Browse",
font="Times", background="#038a54",
command=lambda:self.nilai1.set(fdialog.askopenfilename()))

```

```

        self.path1.grid(row=4, column=5, sticky="NSEW", padx=5,
pady=5)

        #frame 2
        #COMBOBOX GLCM
        arah = tk.IntVar(self.FrameKiri2)
        self.label = tk.Label(self.FrameKiri2, text="GLCM",
width=12, font="Times", relief="ridge", background="#b0f5d9")
        self.label.grid(row=5, column=0, padx=8 , pady=10)

        #arah GLCM
        arah = tk.StringVar(self.FrameKiri2, value="")
        self.cbarahglcm = ttk.Combobox(self.FrameKiri2,
textvariable = arah, width=10)
        self.cbarahglcm['value'] = (0, 45, 90, 135)
        self.cbarahglcm.current(0)
        self.cbarahglcm.grid(row=5, column=1, sticky="NSEW",
pady=10)

        #jarak GLCM
        self.label1 = tk.Label(self.FrameKiri2, text="Jarak",
font="Times", relief="ridge", width=12, background="#b0f5d9")
        self.label1.grid(row=6, column=0, sticky="NSEW", padx=8,
pady=5)

        jarak = tk.StringVar(self.FrameKiri2, value="")
        self.tbjarak = tk.Entry(self. FrameKiri2, textvariable =
jarak, width=30)
        self.tbjarak.grid(row=6, column=1, sticky="NSEW",
padx=8, pady=5 )

        #frame 3
        #klasifikasi KNN
        #label iterasi
        self.label = tk.Label(self.FrameKiri3, text="Nilai K",
font="Times", relief="ridge", width=12, background="#b0f5d9")
        self.label.grid(row=0, column=0, sticky="NSEW", padx=8,
pady=5)

        #textbox iterasi
        nilai2 = tk.StringVar(self.FrameKiri3, value="")
        self.textboxKnn = tk.Entry(self.FrameKiri3, textvariable
= nilai2, width=30)

```

```

        self.textboxKnn.grid(row=0, column=1, sticky="NSEW",
pady=5)

        #frame 4
        #button proses
        self.path2 = tk.Button(self.FrameKiri4, width=34,
height=4, text="PROSES",
font="Times", background="#038a54", command=self.Proses)
        self.path2.grid(row=5, column=0, padx=10, pady=6,
sticky="NSEW")

        #frame 5
        #output
        #label hasil Image Asli
        self.label = tk.Label(self.FrameKanan5, text="Image
Testing", font="Times", background="#79f7c5", fg="black")
        self.label.grid(row=0, column=0, padx=60, pady=17)

        #images Hasil Testing
        self.noimage1 = 'no-image.png'
        self.noimageTest1 = Image.open(self.noimage1)
        resize = self.noimageTest1.resize([120, 120])
        self.noimageTest1 = ImageTk.PhotoImage(resize)
        self.labelTest1 = tk.Label(self.FrameKanan5, bg = "gray",
image=self.noimageTest1)
        self.labelTest1.grid(row=1, column=0, padx=30, pady=1,
sticky="W")

        #label Grayscale
        self.label = tk.Label(self.FrameKanan5, text="Grayscale",
font="Times", background="#79f7c5", fg="black")
        self.label.grid(row=0, column=1, padx=60, pady=17)
        #images Hasil Grayscale
        self.noimage2 = 'no-image.png'
        self.noimageTest2 = Image.open(self.noimage2)
        resize2 = self.noimageTest2.resize([120, 120])
        self.noimageTest2 = ImageTk.PhotoImage(resize2)
        self.labelTest2 = tk.Label(self.FrameKanan5, bg= "gray",
image=self.noimageTest2)
        self.labelTest2.grid(row=1, column=1, padx=30, pady=1,
sticky="N")

        #label Segmentasi

```

```

        self.label = tk.Label(self.FrameKanan5,
text="Segmentasi", font="Times", background="#79f7c5",
fg="black")
        self.label.grid(row=0, column=2, padx=60, pady=17)
        #images Hasil Segmentasi
        self.noimage3 = 'no-image.png'
        self.noimageTest3 = Image.open(self.noimage3)
        resize3 = self.noimageTest3.resize([120, 120])
        self.noimageTest3 = ImageTk.PhotoImage(resize3)
        self.labelTest3 = tk.Label(self.FrameKanan5, bg = "gray",
image=self.noimageTest3)
        self.labelTest3.grid(row=1, column=2, padx=30, pady=1,
sticky="S")

        #frame 6
        #label keterangan hasil output
        self.label12 = tk.Label(self.FrameKanan6, text="NILAI
TINGKAT AKURASI", font="Times", background="#038a54")
        self.label12.grid(row=0, column=0)

        self.text13 = tk.Text(self.FrameKanan6, height=4,
width=55, font="Times")
        self.text13.grid(row=0, column=2, padx=3, pady=3)
        self.desc = '''
'''
        self.text13.insert(tk.END, self.desc)

        self.text14 = tk.Text(self.FrameKanan7, height=4,
width=80, font="Times")
        self.text14.grid(row=2, column=0)
        self.desc1 = '''
'''
        self.text14.insert(tk.END, self.desc1)

        #frame 7
        #label kaki
        self.label = tk.Label(self.FrameBawah, text="DETEKSI
PENYAKIT PADA TANAMAN TERONG MENGGUNAKAN METODE GLCM DAN
ALGORITMA KNN", font="Times", fg="black")
        self.label.grid(row=2, column=2, sticky="NSEW", padx=150,
pady=1)
        self.label.config(background="#16fa9f")
        self.FrameBawah.grid_propagate(0)

```

```

def Proses(self):
    self.text13.delete("1.0","end")
    self.text14.delete("1.0","end")
    #get image training
    folder_train = self.txtrain.get()
    #get image testing
    folder_test = self.txtest.get()
    #glc
    GLCM= self.cbarahglcm.get()
    jarak = self.tbjarak.get()

    #textbox iterasi
    K = self.textboxKnn.get()

    #hasil output akhir
    a = dataTraining.analisiscitra()
    hasil, img_segmen, gray, =
a.proses_rekognisi(folder_train, folder_test, GLCM, jarak,K)

    if hasil[0] == '1_busuk_buah':
        self.text13.insert(tk.END, "Busuk buah adalah
Penyakit yang diakibatkan oleh jamur yang bernama phytophthora Sp.
Gejala awal yang ditemui pada penyakit ini biasanya buah akan
memiliki bercak coklat kehitaman, bercak tersebut akan membesar
keseluruh daging buah dan mengakibatkan buah lembek dan
membusuk.")
        self.text14.insert(tk.END, "Menanam varietas yang
tahan, Ambil dan musnahkan buah yang terinfeksi, Semprotkan
dengan fungisida dithane, antracol, cozeb, atau Bion M")
    elif hasil[0] == '2_layu_bakteri':
        self.text13.insert(tk.END, "Penyakit Layu Bakteri
merupakan suatu bakteri yang dapat hidup, tumbuh, dan berkembang
biak di dalam tanah. Munculnya penyakit ini sering terjadi ketika
musim penghujan dikarenakan bakteri ini dapat berkembangbiak
dengan cepat, bahkan bisa 2 kali lipat. Gejala yang timbul pada
tanaman terong yang terserang penyakit ini yaitu mula-mula
terjadinya layu dibagian pucuk daun kemudian akan layu
seluruhnya")
        self.text14.insert(tk.END, "Rotasi tanaman yang bukan
inang, Menjaga area tanaman agar tidak lembab, Mencabut dan
memusnahkan tanaman terserang lalu Semprotkan bakterisida pada
pangkai batang. Bakterisida yang digunakan misalnya agrimicin,
agrept, bactoxyn atau plantomycin.")
    elif hasil[0] == '3_antraknosa':

```

```

        self.text13.insert(tk.END, "Gloesporium Melongena merupakan jenis jamur yang menyebabkan penyakit antraknosa pada pohon terong. Ciri-cirinya seperti permukaan daun yang warnanya berubah dan kemudian mulai menggulung.")
        self.text14.insert(tk.END, "Menanam varietas yang tahan, Mengatur jarak tanam, Membersihkan gulma yang bisa menjadi tanaman inang, Memusnahkan buah terinfeksi dan memangkas bagian tanaman yang terserang, Semprotkan dengan fungisida antracol, cozeb, Bion M atau Dithane.")
        elif hasil[0] == '4_bercak_daun':
            self.text13.insert(tk.END, "Gejala awal serangan bercak daun pada tanaman terong ditandai dengan adanya bercak berwarna kehitaman pada permukaan daun. pada musim hujan akan cepat menyebar ke seluruh daun tanaman terong. Pada serangan yang sudah parah, bercak daun menyebabkan daun menjadi kuning seperti terbakar hingga akhirnya rontok. Penyakit bercak daun dapat menyebar dengan cepat pada kondisi curah hujan tinggi. Hal ini dikarenakan cendawan cercospora berkembang dengan cepat pada kondisi yang sangat lembab.")
            self.text14.insert(tk.END, "Mengatur jarak tanam, Menggunakan mulsa plastik jika menanam dimusim hujan, Menjaga kebersihan areal dari gulam agar area tanaman tidak lembab, Semprotkan fungisida starmyl, saromyl, score atau dithane.dengan fungisida antracol, cozeb, Bion M atau Dithane.")
        elif hasil[0] == '5_rebai_semai':
            self.text13.insert(tk.END, "Rebai semai adalah penyakit yang menyerang terong pada saat sedang disemai. Penyakit ini mengakibatkan benih menjadi kekuningan lalu kemudian mati sehingga tidak mungkin untuk Anda menanamnya lagi.")
            self.text14.insert(tk.END, "dapat dengan menggunakan bibit unggul, Menggunakan varietas yang tahan, Mencabut dan membuang tanaman terserang, Siram semaian seperlunya, jangan terlalu basah/lembab, Semprotkan fungisida antracol, dithane, Bion M atau Cozeb serta menjaga sanitasi lingkungan sekitar penyemaian.")
        elif hasil[0] == '6_penyakit_batang':
            self.text13.insert(tk.END, "Penyakit busuk batang pada tanaman terong disebabkan oleh cendawan Phytophthora capsici dan Phytophthora infestans. Pada musim penghujan dengan curah hujan yang sangat tinggi, dengan kelembaban yang tinggi intensitas serangan cendawan cepat meningkat dan berkembang biak dengan cepat. Gejala dan Penyebab Penyakit Busuk Batang terdapat bercak berwarna kecoklatan pada pangkal batang, cabang, maupun bagian pucuk tanaman. Bercak tersebut jika disentuh terasa lembek, kulit mudah terkelupas dan lama kelamaan bercak berubah

```

warna menjadi kehitaman. Bagian tanaman yang terserang akan layu, menguning, kering, dan mati.")

```
self.text14.insert(tk.END, "Menggunakan varietas yang tahan, Mengatur jarak tanam, Mencabut dan memusnahkan tanaman yang terserang, Menjaga kelembaban tetap stabil, Semprotkan fungisida antracol, dithane, cozeb atau bionM.")
```

```
elif hasil[0] == '7_jamur_tepung':  
    self.text13.insert(tk.END, "Jamur tepung biasanya akan menyerang daun dan pangkal tanaman. Pada daun akan terlihat serbuk putih permukaan atas dan bawah daun layaknya tepung yang lama kelamaan akan berubah warna menjadi warna kuning dan merusak pohon terong. Penyakit embun tepung memang tidak akan berdampak langsung pada buah, namun penyakit ini mampu membunuh tanaman. Daun yang terserang perlahan akan berwarna kekuningan, hingga daun mati dan gugur.")
```

```
self.text14.insert(tk.END, "Hindari pengairan yang berlebih, Lakukan pengendalian gulma di lahan, Aplikasi fungisida dilakukan tiap 5 hari sekali jika kondisi lingkungan lembab dan basah, namun jika kondisi cuaca sedang kering, maka aplikasi fungisida dapat dilakukan dalam interval waktu 7-10 hari dengan menggunakan heksakonazol, triflumizol, miklobutanil (dengan masing masing takaran 2ml/l).")
```

```
else:  
    self.text13.insert(tk.END, "tanaman anda sehat walafiat")
```

```
self.text14.insert(tk.END, "dibutuhkan pupuk dan air secukupnya untuk tetap sehat")
```

```
self.label12.configure(text = hasil)
```

```
#hasil gambar
```

```
#hasil image 1
```

```
imgTest = Image.open(self.txttest.get())  
imgTest = imgTest.resize((165, 165), Image.ANTIALIAS)  
imgTest = ImageTk.PhotoImage(imgTest)  
self.labelTest1.configure(image=imgTest)  
self.labelTest1.image_names = imgTest
```

```
#hasil image 2
```

```
imgGray = Image.fromarray(gray)  
imgGray = imgGray.resize((165, 165), Image.ANTIALIAS)  
imgGray = ImageTk.PhotoImage(imgGray)  
self.labelTest2.configure(image=imgGray)  
self.labelTest2.image_names = imgGray
```

```

        #hasil image 3
        imgsegmen = cv2.cvtColor(img_segmen, cv2.COLOR_BGR2RGB)
        imgsegmen = Image.fromarray(imgsegmen)
        imgsegmen = imgsegmen.resize((165,165), Image.ANTIALIAS)
        imgsegmen = ImageTk.PhotoImage(imgsegmen)
        self.labelTest3.configure(image=imgsegmen)
        self.labelTest3.image_names = imgsegmen

def main():
    root = tk.Tk()
    root.geometry('1070x670+130+10')
    root.configure(background='#79f7c5')
    app = Aplication(root)
    app.parent.title("Deteksi Penyakit Terong")
    root.mainloop()

if __name__=='__main__':
    main()

```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
SK MENDIKNAS NOMOR 84/D/O/2001

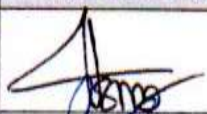
JL. Achmad Nadjamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

Berita Acara Perbaikan/Revisi Ujian SKRIPSI

Pada hari ini, Selasa 30 Agustus 2022, Pukul 13.00-15.00 Wita. Telah dilaksanakan Ujian SKRIPSI mahasiswa Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Nama : I Gusti Made Teresna
Nim : T3117003
Pembimbing I : Zohrahayaty, M.Kom
Pembimbing II : Irma Surya Kumala I, M.Kom
Judul SKRIPSI : Gray Level Co-Occurrence Matrix (GLCM) Untuk Deteksi Penyakit Pada Tanaman Terong Menggunakan K-Nearest Neighbor (KNN)

Oleh Komite Seminar sebagai berikut :

No	Komite Seminar	Status	Tanda Tangan
1	Asmaul Husna, M.Kom	Ketua	
2	Husdi, M.Kom	Anggota	
3	Zulfrianto Y. Lamasigi, M.Kom	Anggota	
4	Zohrahayaty, M.Kom	Anggota	
5	Irma Surya Kumala I, M.Kom	Anggota	

DAFTAR RIWAYAT HIDUP



Nama : I GUSTI MADE TERESNA
Tempat,Tgl Lahir : Werdhi Agung, 22 juni 1998
Pekerjaan : Mahasiswa
Email : guztiteresna@gmail.com

Daftar Riwayat Hidup :

1. Tahun 2011, menyelesaikan Pendidikan di Sekolah Dasar Negeri 1 Werdhi Agung, Kecamatan Dumoga Tengah, Kabupaten Bolaang Mongondow, Provinsi Sulawesi Utara.
2. Tahun 2013, menyelesaikan Pendidikan di Sekolah Menengah Pertama Negeri 1 Dumoga, Kecamatan Dumoga Tengah, Kabupaten Bolaang Mongondow, Provinsi Sulawesi Utara.
3. Tahun 2016, menyelesaikan Pendidikan di Sekolah Menengah Kejuruan Negeri 3 Gorontalo, Kecamatan Kota Tengah, Kota Gorontalo.
4. Tahun 2017, telah diterima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo.

