identifikasi penyakit

pada citra daun padi menggunakan operasi morfologi dan

rule-based classification

SKRIPSI

Diajukan untuk memenuhi Sebagian Syarat Guna Memperoleh

Gelar Sarjana Program Strata 1 (S.1)

Dalam Ilmu Teknologi Informasi

Oleh :

IQBAL GIVARI

NIM : 2008096055

PROGRAM STUDI S-1 TEKNOLOGI INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG

2024

KEMETRIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang Telp. 024-7601295

LEMBAR PENGESAHAN

Naskah skripsi berikut ini:

Judul : Identifikasi Penyakit Pada Citra Daun Padi Menggunakan Operasi Morfologi dan *Rule-Based Classification*

Nama : Iqbal Givari

NIM : 2008096055

Jurusan : Teknologi Informasi

Telah diujikan dalam Sidang Munaqosah oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang dan dapat diterima sebagai salah satu syarat memperoleh gerlar sarjana dalam bidang teknologi informasi.

Semarang,

DEWAN PENGUJI

Ketua Sidang / Penguji	Sekertaris Sidang / Penguji
Nur Cahyo Hendrowibowo, S.T, M,Kom.	Masy Ari Ulinuha, M.T.
NIP. 19731222200604 1 001	NIP. 19810812201101 1 007
Penguji Utama I	**Penguji Utama II**
Khotibul Umam, M.Kom.	Mokhamad Ikhl Musofa, M.Kom.
NIP. 19790827201101 1 007	NIP. 19880807201903 1 010
Pembimbing I	**Pembimbing II**
Masy Ari Ulinuha, M.T.	Adzhal Arwani Mahfudh, M.Kom.
NIP. 19810812201101 1 007	NIP. 19910703201903 1 006

KEMENTERIAN AGAMA FAKULTAS SAINS DAN TEKNOLOGI UIN WALISONGO SEMARANG REPUBLIK INDONESIA

PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama	: Iqbal Givari
NIM	: 2008096055
Jurusan	: Teknologi informasi

Menyatakan bahwa skripsi yang berjudul:

Identifikasi Penyakit Pada Citra Daun Padi Menggunakan Operasi Morfologi dan Metode *Rule-Based Classification*

Secara keseluruhan adalah penelitian/karya yang ditulis oleh saya sendiri, terkecuali bagian-bagian tertentu yang dirujuk sumbernya.

Semarang, 28 Juni 2024

t Pernyataan,

Iqbal Givari

NIM. 2008096055

NOTA DINAS

Semarang, 24 Mei 2024

Yth Ketua Program Studi Teknologi Informasi Fakultas Sains dan Teknologi UIN Walisongo Semarang

Assalaimu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan:

Judul : Identifikasi Penyakit pada Citra Daun Padi Menggunakan Operasi Morfologi dan Metode *Rule-Based Classification*
Nama : Iqbal Givari
NIM : 2008096055
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diajukan dalam Sidang Munaqosah

Wassalamu'alaikum. wr. wb.

Pembimbing I,

ADZHAL ARWANI MAHFUDH
NIP. 19910703 201903 1 006.

NOTA DINAS

Semarang, 24 Mei 2024

Yth. Ketua Program Studi Teknologi Informasi Fakultas Sains dan Teknologi UIN Walisongo Semarang

Assalaimu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan, dan koreksi naskah skripsi dengan:

Judul : **Identifikasi Penyakit pada Citra Daun Padi Menggunakan Operasi Morfologi dan Metode *Rule-Based Classification***
Nama : Iqbal Givari
NIM : 2008096055
Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diajukan dalam Sidang Munaqosah

Wassalamu'alaikum. wr. wb.

Pembimbing II,

MASY ARI ULINUHA
NIP. 19818012 201101 1007.

KATA PENGANTAR

Segala puja dan puji syukur atas kehadirat Allah *Subhanahu Wata'ala* penulis panjatkan karena atas berkat rahmat dan nikmat yang telah Ia karuniakan, penulis dapat menyelesaikan Tugas Akhir yang berjudul "Identifikasi Penyakit pada Citra Daun Padi Menggunakan Operasi Morfologi dan Metode *Rule-Based Classification*" ini. *Shalawat* serta salam penulis limpahkan kepada junjungan *Nabiallah* Muhammad *Shallallahu Alaihi Wassalam* yang telah membawa kita umat muslim dari zaman yang penuh kebodohan hingga zaman yang terang benerang akan ilmu pengetahuan dan teknologi seperti pada saat ini.

Penulis sadari bahwa banyak pihak yang telah memberi dukungan dan bantuan selama menyelesaikan program studi dan tugas akhir ini. Oleh sebab itu, segala rasa hormat dan ucapin terimakasih penulis sampaikan dan mendo'akan semoga Allah memberikan balasan yang terbaik kepada:

1. Orang tua penulis yang selalu memberikan semangat dan do'a tanpa kenal waktu dan tempat, dan satu-satunya tempat yang menjadi tempat untuk *'healing'* bagi penulis.
2. Bapak Masy Ari Ulinuha M.T. yang telah membantu serta membimbing penulis tahap demi tahap dalam menyelesaikan tugas akhir ini

3. Bapak Adzhal Arwani Mahfudh, S.Kom., M. Kom, selaku dosen pembimbing dalam menyelesaikan tugas akhir ini.
4. Bapak Dr. Khothibul Umam. S.T., M. Kom., selaku Kepala Program Studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang
5. Serta kepada para dosen-dosen Program Studi Teknologi Informasi UIN Walisongo lainnya, dan juga kepada para sahabat dan semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat penulis sebutkan satu persatu

Penulis menyadari bahwa tugas akhir yang diselesaikan ini masih jauh dari kata sempurna, dimana pastinya terdapat beberapa kesalahan dan kekurangan dalam penyusunannya. Oleh karena itu, penulis meminta maaf atas segala kesalahan dan kekurangan yang telah dilakukan.

Penulis harap laporan tugas akhir atau juga disebut sebagai skripsi ini dapat bermanfaat bagi para pembaca dan dapat dijadikan sebagai referensi untuk melakukan pengembangan selanjutnya ke arah yang lebih baik.

Semarang, 23 Mei 2024

Iqbal Givari

ABSTRAK

Pengolahan citra digital adalah bidang ilmu komputer yang berkaitan dengan pemrosesan gambar digital menggunakan berbagai algoritma dan teknik komputasi. Tujuan utama dari pengolahan citra itu sendiri ialah memanipulasi, menganalisis, atau bahkan meningkatkan informasi visual yang ada di dalam citra itu sendiri. Dalam konteks pertanian, khususnya pertanian yang melibatkan tumbuhan padi (*Oryza sativa*), pengolahan citra memainkan peran penting dalam mengidentifikasi gejala-gejala penyakit yang sedang diidap oleh tanaman. Padi sendiri merupakan salah satu tanaman budidaya terpenting dalam peradaban, namun juga sangat rentan terhadap penyakit yang dapat menyebabkan kegagalan produksi dan kerugian ekonomi bagi para petani.

Penyakit pada padi seringkali menunjukkan gejala visual seperti bintik-bintik, coretan, dan perubahan warna pada daun dan batangnya. Identifikasi penyakit tersebut dapat dilakukan secara konvensional yang dilakukan secara pengamatan langsung, namun teknologi digital juga memungkinkan untuk mendeteksi yang lebih akurat melalui analisis citra digital.

Penelitian ini bertujuan untuk membuat seubah algoritma atau program yang dapat membaca area dari daun padi yang terdampak oleh penyakit-penyakit tersebut. Teknik segmentasi citra dan ekstraksi fitur digunakan untuk memisahkan area yang terdampak penyakit dengan area yang tidak terdampak. *Rule-Based Classification* digunakan untuk mengklasifikasi termasuk penyakit apa area yang berhasil terbaca oleh teknik segmentasi citra dan ektraksi fitur tersebut. Hasil dari penelitian ini menunjukkan bahwa program yang dibuat memiliki hasil evaluasi yang menggunakan nilai metrik dari *confusion matrix*, yaitu nilai akurasi sebesar 0,86, nilai presisi sebesar 0,71, nilai *recall* sebesar 1, dan nilai *f1 score* sebesar 0,83.

Kata Kunci : Pengolahan Citra, Padi, Ekstraksi Fitur, Penyakit pada Padi, *Confusion Matrix, Rule-Based Classification*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

A. Latar Belakang

Image Processing adalah bidang pada Ilmu Komputer yang berkaitan dengan pemrosesan gambar atau citra digital menggunakan berbagai alrgoritma dan teknik komputasi. Tujuan utama dari *image processing* sendiri ialah untuk memanipulasi, menganalisis, atau meningkatkan informasi visual yang terkandung dalam citra (Dijaya, 2017). *Image Processing* sendiri memiliki banyak aplikasi dalam berbagai bidang (Iriyanto & Zaini, 2014), termasuk ke dalam bidang pertanian. Dalam penelitian ini bidang pertanian yang difokuskan ialah pertanian padi.

Padi (bahasa latin: *Oryza sativa)* merupakan salah satu tanaman budidaya terpenting dalam peradaban. Meskipun terutama mengacu pada jenis tanaman budidaya, padi juga digunakan untuk mengacu pada beberapa jenis dari marga (genus) yang sama, yang biasa disebut sebagai padi liar (Walascha et al., 2021). Dalam konteks 'tanaman budidaya terpenting dalam peradaban' padi merupakan salah satu dari tanaman yang paling sering ditanam untuk memenuhi kebutuhan pangan manusia hingga saat ini.

Namun sama dengan tumbuhan-tumbuhan lainnya, padi juga rentan terhadap penyakit. Penyakit-penyakit ini tak bisa dianggap remeh, dan dibutuhkan perawatan dan obat-obatan yang tepat agar dapat menjamin keberhasilan panen dari padi tersebut. Penyakit-penyakit ini dapat menyebabkan kegagalan produksi dan kerugian ekonomi terutama bagi para petani yang menanam tumbuhan ini sebagai mata pencaharian mereka (Bianome et al., 2020).

Penyakit ini sering menimbulkan gejala-gejala yang dapat dilihat secara langsung (visual) pada permukaan daun, seperti halnya bintik-bintik, coretan, dan perubahan warna pada beberapa bagian dari daun tersebut. Identifikasi penyakit-penyakit tersebut, biasanya dilakukan secara konvensional dengan memeriksa secara langsung daun-daun dari tanaman padi tersebut tanpa menggunakan alat apapun (Siregar & Sulardi, 2019). Sementara itu, tak dapat dipungkiri bahwa identifikasi tersebut juga dapat dilakukan menggunakan perangkat teknologi digital, yaitu pemeriksaan area kerusakan pada daun yang dilakukan berdasarkan citra digital (Anwar & Setyowibowo, 2021). Dalam hal ini perangkat teknologi ditantang untuk melokalisir gejala kerusakan pada area tertentu tersebut dengan benar.

Pada pengolahan citra digital, pemisahan area yang terdampak atau yang rusak karena gejala penyakit tersebut, perlu dipisahkan dengan area yang tidak terdampak oleh karena itu teknik melokalisir ini sangat diperlukan. Teknik dalam bidang pengolahan citra yang dibutuhkan dalam hal ini, ialah teknik segmentasi (Gong et al., 2019).

Segmentasi citra sampai saat ini masih menjadi perhatian yang cukup penting dalam hal pengolahan citra digital. Sebagai contoh tak hanya dalam penelitian ini, namun dalam beberapa hal seperti medis, teknik segmentasi citra ini sangat membantu dalam mendeteksi gangguan-gangguan medis dalam tubuh misalnya mendeteksi tumor atau bahkan untuk membaca organ-organ dalam tubuh manusia (Iriyanto & Zaini, 2014). Oleh karena itu, teknik pengelolaan citra segmentasi ini dipilih untuk membantu dalam melakukan penelitian ini.

Teknik segmentasi citra ini, merujuk pada partisi sebuah citra menjadi beberapa bagian yang didasarkan kemiripan ciri atau keseragaman yang dimiliki. Kegunaan dari teknik ini cukup penting khususnya dibidang analisis dan pengaplikasian pengolahan citra digital. Ada beberapa metode yang efektif untuk segmentasi citra dalam

mendeteksi kerusakan pada daun ini seperti *threshold, clustering, area* dan *model* (Gonzalez & Woods, 2008).

Selain teknik segmentasi citra, teknik ekstraksi fitur juga dapat digunakan, bahkan dapat digunakan secara bersamaan yaitu, setelah teknik segmentasi dilakukan. Teknik ini sangat berperan penting untuk mengidentifikasi fitur dari gejala penyakit yang terjadi pada daun padi, seperti halnya perubahan warna, bentuk kerusakan, dan lain sebagainya. Ekstraksi fitur sendiri merupakan proses mengidentifikasi dan mengekstrak informasi penting atau karakteristik khusus dari citra (Agustiani et al., 2022). Oleh karena itu, metode ini akan berperan sangat penting dalam penelitian ini, dalam rangka untuk mendeteksi dan mengidentifikasi gejala penyakit yang terjadi.

Selain itu, metode *rule-based classification* akan digunakan untuk mengklasifikasikan fitur yang berhasil dari hasil pengolahan citra-citra tersebut berdasarkan aturan-aturan yang akan dibuat. Aturan-aturan tersebut akan mengacu kepada beberapa referensi atau sumber yang menjelaskan tentang karakteristik dari penyakit-penyakit yang menyerang daun padi.

Namun, meskipun itu merupakan sebuah metode yang efektif akan tetapi belum menjamin akan kecocokan dan

keberhasilan dalam medeteksi area yang rusak dan gejala yang terlihat tersebut. Maka dari itu, ini merupakan sebuah tantangan yang akan dihadapi apabila tidak ada metode-metode pengolahan citra digital yang mampu digunakan untuk memecahkan kasus tersebut. Hal ini disebabkan citra memiliki perbedaan warna, tekstur, tingkat iluminasi dan *noise* yang berbeda (Lu, 2020).

B. Rumusan Masalah

Masalah-masalah yang telah diidentifikasi dalam melakukan penilitian ini adalah:

1) Bagaimana metode yang tepat untuk mengklasifikasikan penyakit yang terjadi pada citra daun padi?
2) Bagaimana mengukur peforma dari metode klasifikasi yang diusulkan?

C. Batasan Masalah

Dalam melaksanakan penelitian ini, adapun batasan-batasan masalah yang dihadapi adalah:

1) Data akan diambil di sebuah area pesawahan yang berada pada Kelurahan Karang Layung, Kec. Sukra, Kab. Indramayu, Jawa Barat.
2) Data yang didapat merupakan sebuah citra digital.

3) Gejala penyakit yang diidentifikasi hanya sampai ke beberapa gejala, seperti halnya bercak hitam yang menyebar, dan hawar berwarna kuning-kecoklatan yang memanjang.

D. Tujuan Penelitian

Penelitian ini bertujuan untuk memenuhi tugas akhir dari Program Studi Teknologi Informasi khususnya. Selain tujuan tersebut, adapun tujuan-tujuan lain dari penelitian ini ialah sebagai berikut.

1) Menemukan metode yang tepat agar dapat mengklasifikasikan penyakit yang muncul pada citra daun padi.
2) Mengetahui peforma dari metode yang diusulkan.

E. Manfaat Penelitian

Adapun manfaat-manfaat dari penelitian ini dibagi menjadi dua hal, yaitu manfaat secara teoritis dan manfaat secara praktis.

- Manfaat Teoritis

  Penelitian ini diharapkan menjadi sebuah referensi baru baik itu dalam penelitian dibidang Visi Komputer, maupun dalam penelitian dibidang Pertanian khususnya yang memusatkan tanaman padi sebagai objek penelitiannya.

- Manfaat Praktis

  Penelitian ini diharapkan dapat membantu dalam mendeteksi gejala-gejala penyakit yang timbul pada daun tanaman padi, yang juga dapat menuntun ke arah mengidentifikasi penyakit yang sedang diderita oleh tanaman padi. Baik itu kepada para petani tanaman padi itu sendiri, maupun kepada para pengembang.

BAB II

LANDASAN PUSTAKA

A. Kajian Teori

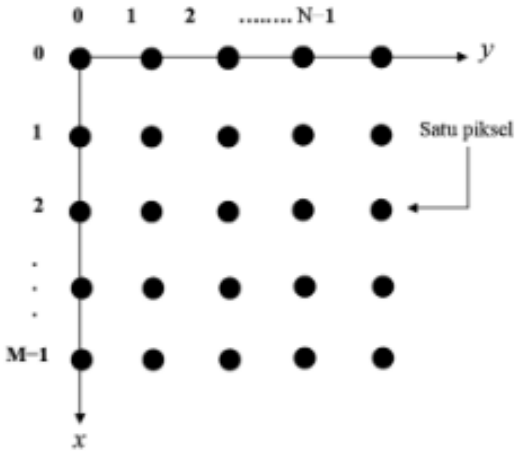
Dalam sebuah penelitian pasti diperlukan sebuah teori yang mana merujuk kepada apa yang sedang diteliti. Dikarenakan penelitian itu sendiri terkadang didasarkan kepada teori-teori yang sudah ada. Adapun kajian teori yang digunakan dalam penelitian ini ialah.

1. Citra Digital

Citra adalah suatu gambaran atau kemiripan dari suatu objek. Ada juga yang menyebutkan bahwasannya citra merupakan fungsi dari intesitas cahaya yang direpresentasikan dalam bidang dua dimensi. Berdasarkan bentuk sinyal penyusunnya, citra dapat digolongkan menjadi dua jenis, yaitu citra analog dan citra digital. Citra analog adalah citra yang dibentuk dari sinyal analog yang bersifat kontinyu, sedangkan citra digital adalah citra yang dibentuk dari sinyal digital yang bersifat diskrit (Gong et al., 2019).

Citra digital merupakan representasi dari fungsi intesitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra ini tersusun oleh sekumpulan piksel (*picture element*) yang memiliki koordinat *(x,y)* dan

amplitude *f(x,y)*. Koordinat *(x,y)* menunjukkan letak/posisi piksel dalam suatu citra, sedangkan amplitude *f(x,y)* menunjukkan nilai intensitas warna citra (Gonzalez & Woods, 2008).



Gambar 2.1 Sistem Koordinat yang dipergunakan untuk mewakili citra
(Sumber: pemogramanmatlab.com)

Artinya, sebuah citra digital dapat ditulis dalam bentuk matriks berikut :

$$f(x,y) = \begin{bmatrix} f(0,0) & \dots & \dots & f(0,M-1) \\ f(1,0) & \dots & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

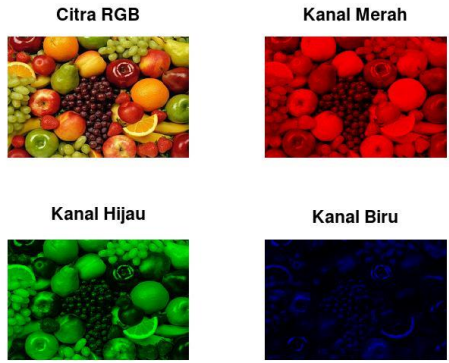
Berdasaran gambaran di atas, maka secara matematis citra digital dapat dituliskan sebagai fungsi

intensitas *f(x,y)*, di mana harga *x* (baris) dan *y* (kolom) merupakan koordinat posisi dan *f(x,y)* adalah nilai fungsi pada setiap titik (*x,y)* yang menyatakan besar intensitas citra atau tingkat keabuan atau dari piksel di titik tersebut (Gonzalez & Woods, 2008).

Pada umumnya, berdasarkan kombinasi warna pada piksel, citra dibagi menjadi tiga jenis yaitu citra *Red Green and Blue* (RGB), citra *grayscale*, dan citra biner. Citra RGB tersusun oleh tiga kanal warna, yaitu kanal merah, kanal hijau, dan kanal biru (Gong et al., 2019).

Masing-masing kanal warna memiliki nilai intensitas piksel dengan kedalaman bit sebesar 8-bit yang artinya memiliki variasi warna sebanyak 2^8 derajat warna (0 s.d 255) (Iriyanto & Zaini, 2014).

Pada kanal merah, warna merah sempurna direpresentasikan dengan nilai 255 dan hitam sempurna dengan nilai 0. Pada kanal hijau sempurna direpresentasikan dengan nilai 255 dan hitam sempurna dengan nilai 0. Begitu pula pada kanal biru, warna biru sempurna direpresentasikan dengan nilai 255 dan hitam sempurna dengan nilai 0 (Hermana et al., 2018).



Gambar 2.2 Representasi citra RGB dan kanal warnanya

Setiap piksel pada citra RGB, memiliki intensitas warna yang merupakan kombinasi dari tiga nilai intensitas pada kanal R, G, dan B. Sebagai contoh, suatu piksel yang memiliki nilai intensitas warna sebesar 255 pada kanal merah, 255 pada kanal hijau, dan 0 pada kanal biru akan menghasilkan warna kuning. Pada contoh lain, suatu piksel yang memiliki intensitas warna sebesar 255 pada kanal merah, 102 pada kanal hijau, dan 0 pada kanal biru akan menghasilkan warna oranye. Banyaknya kombinasi warna piksel yang mungkin pada citra RGB *truecolor* 24-bit adalah sebanyak *256 x 256 x 256 = 16.777.216* (Sinha, 2012).

Gambar 2.3 Nilai piksel RGB
(Sumber: pemogramanmatlab.com)

Jenis citra berikutnya adalah citra *grayscale*. Citra *grayscale* merupakan citra yang nilai intensitas pikselnya didasarkan pada derajat keabuan. Pada citra *grayscale* 8-bit, derajat warna hitam sampai dengan putih dibagi ke dalam 256 derajat keabuan dimana warna hitam sempurna direpresentasikan dengan nilai 0 dan putih sempurna dengan nilai 255. Citra biasa atau disebut juga sebagai citra RGB dapat dikonversikan menjadi citra *grayscale* sehingga dihasilkan hanya satu kanal warna (Vincent, 1993).

Persamaan yang umumnya digunakan untuk mengkonversi citra RGB *truecolor* 24-bit menjadi citra *grayscale* 8-bit adalah

$$Grayscale = (0.2989R) + (0.5870G) + (0.1140B)$$

Dimana:

Grayscale = nilai intensitas citra grayscale,

R = nilai intensitas piksel kanal merah

G = nilai intensitas piksel kanal hijau

B = nilai intensitas piksel kanal biru

Gambar 2.4 Citra hasil konversi RGB ke Grayscale

Jenis citra yang terakhir berdasarkan intensitas warnanya adalah citra biner. Citra biner adalah citra yang pikselnya memiliki kedalaman bit sebesar 1 bit, sehingga hanya memiliki dua nilai intensitas warna,

yaitu 0 (hitam) dan 1 (putih) (Dijaya, 2017). Citra *grayscale* dapat dikonversi menjadi citra biner melalui proses *thresholding*. Dalam proses *thresholding*, dibutuhkan suatu nilai *threshold* sebagai nilai pembatas konversi. Nilai intesitas piksel yang lebih besar atau sama dengan nilai *threshold* akan dikonversi menjadi 1. Sedangkan nilai intensitas piksel yang kurang dari nilai *threshold* akan dikonversi menjadi 0. Misalnya nilai *threshold* yang digunakan adalah 128, maka piksel yang memiliki nilai intensitas kurang dari 128 akan diubah menjadi 0 (hitam), dan yang lebih dari atau sama dengan 128 akan diubah menjadi 1 (putih) (Thanki & Kothari, 2019).

Gambar 2.5 Citra hasil konversi Grayscale ke biner

2. Segmentasi Citra

Segmentasi berarti membagi gambar atau bingkai video menjadi sejumlah bagian atau subset properti yang spesifik dan unik sesuai dengan prinsip, dan mengekstraksi target yang dipilih untuk tingkat analisis dan pemahaman yang lebih tinggi. Oleh karena itu, segmentasi citra merupakan dasar dari ekstraksi fitur, pengenalan dan pelacakan (Gonzales et al., 2009). Segmentasi gambar adalah salah satu bidang penglihatan tingkat rendah yang paling mendasar dan penting dalam visi komputer. Hal ini merupakan masalah yang klasik dan juga merupakan dasar-dasar analisis visual dalam pengenalan pola. Tidak ada metode yang umum untuk mensegmentasi gambar sejauh ini, atau kriteria obyektif untuk menilai apakah segmentasi tersebut berhasil (Gong et al., 2019).

Segmentasi gambar dapat dibangun di atas dua konsep dasar yakni, kemiripan dan diskontinuitas. Kemiripan piksel berarti bahwa piksel gambar dalam suatu wilayah tertentu memiliki beberapa karakteristik yang sama, seperti skala abu-abu pada piksel, atau tekstur yang dibentuk oleh susunan piksel. Diskontinuitas mengacu ke diskontinuitas beberapa

fitur piksel, seperti mutasi yang terjadi pada nilai skala abu-abu, mutasi warna dan mutasi struktur tekstur (Gong et al., 2019).

Berikut merupakan beberapa metode-metode yang dapat digunakan dalam melakukan segmentasi citra, diantaranya adalah:

- *Thresholding*
- *Active Contour*
- Deteksi Tepi
- *Region growing*
- Transformasi Hough
- *Clustering*

3. *Feature Extraction* (Ekstraksi Fitur)

Fitur pada gambar adalah salah satu atribut paling dasar yang digunakan untuk membedakan gambar yang berbeda. Fitur-fitur tersebut dapat berupa fitur alami yang dapat diidentifikasi oleh penglihatan, atau beberapa parameter buatan manusia tertentu selama proses pengukuran dan pemrosesan (Martinez-Murica et al., 2017). Ekstraksi fitur adalah proses pengukuran fitur atau atribut intrinsik, esensial dan penting dari objek penelitian dan mengkuantifikasi hasilnya, atau menguraikan dan melambangkan objek untuk

membentuk vektor fitur atau string simbol dan peta relasional (Nixon & Aguado, 2002). Fitur gambar yang umum biasanya meliputi warna, tekstur, bentuk, dan karakteristik hubungan spasial (Hong, 1991).

Fitur warna adalah fitur universal yang menggambarkan sifat permukaan objek yang sesuai dengan gambar atau wilayah gambar. Secara umum, fitur warna adalah karakteristik yang berdasar kepada nilai piksel, di mana semua piksel yang dimiliki oleh sebuah gambar atau wilayah gambar memiliki kontribusinya masing-masing. Karena warnanya tidak sensitif terhadap perubahan arah, ukuran pada gambar atau bahkan wilayah gambar, warna dari fitur warna tidak dapat menangkap karakteristik lokal objek dengan cukup baik (Gong et al., 2019). Fitur umum yang ada pada fitur warna meliputi histogram warna, set warna, momen warna, warna vektor koherensi, korelasi warna, dan sebagainya (Salau & Jain, 2019).

Fitur tekstur juga merupakan fitur universal, seperti fitur warna. Namun, tekstur hanya berisi karakteristik dari permukaan objek dan itu tidak dapat sepenuhnya mencerminkan atribut penting dari objek tersebut. Jadi, tidak mungkin untuk mendapatkan

konten gambar tingkat tinggi hanya dengan menggunakan fitur tekstur. Berbeda dengan fitur warna, fitur tekstur bukanlah karakteristik berdasarkan piksel, dan itu perlu dihitung dengan cara statistik di wilayah gambar yang mencakup banyak nilai piksel (Gong et al., 2019).

Metode ekstraksi fitur tekstur yang umum meliputi *gray level co-occurance matrix* (matriks *co-occurance* yang ada ditingkat warna abu-abu), fitur tekstur Tamura, *simultaneous auto-regresif* (auto-regresif simultan) model tekstur, transformasi wavelet dan sebagainya. *Gray level co-occurance matrix* (matriks *co-occurance* yang ada ditingkat warna abu-abu) difokuskan untuk menghitung empat parameter energi, inersia, entropi dan korelasi berdasarkan matriks *co-occurance* yang ada ditingkat warna keabuan. Tekstur Tamura fitur mengedepankan enam jenis atribut: kekasaran, kontras, orientasi, garis derajat gambar, keteraturan dan kekasaran yang didasarkan pada visual manusia studi psikologi persepsi tentang tekstur (Gong et al., 2019).

Ekstraksi fitur tekstur dari fungsi autokorelasi gambar (yaitu fungsi spektrum energi dari gambar)

mengekstrak parameter karakteristik seperti ketebalan dan arah tekstur melalui perhitungan fungsi spektrum energi. Model auto-regresif simultan (SAR) mengambil parameter sebagai fitur tekstur berdasarkan model konstruksi gambar. Metode yang umum digunakan adalah model medan acak seperti metode model medan acak Markov dan metode model medan acak Gibbs (Nixon & Aguado, 2002).

Biasanya ada dua jenis metode representasi fitur bentuk, yang pertama adalah fitur kontur, dan yang lainnya adalah fitur lokal. Fitur kontur sangat berfokus pada batas luar objek, sedangkan fitur lokal sangat terkait dengan seluruh area lokal (Salau & Jain, 2019). Metode deskripsi fitur bentuk secara umum adalah sebagai berikut:

a) Metode *Boundary Feature* (Fitur Batas)

Transformasi Hough biasanya berkaitan dengan identifikasi garis dalam gambar, tetapi kemudian transformasi Hough telah diperluas untuk mengidentifikasi posisi bentuk yang berubah-ubah, biasanya lingkaran atau elips. Transformasi Hough seperti yang digunakan secara universal saat ini ditemukan oleh Richard Duda dan Peter Hart pada

tahun 1972, yang menyebutnya sebagai "transformasi Hough umum" setelah paten tahun 1962 yang terkait dari Paul Hough. Transformasi ini dipopulerkan dalam komunitas visi komputer oleh Dana H. Ballard melalui artikel jurnal tahun 1981 berjudul "*Generalizing the Hough transform to detect arbitrary shapes*". Metode fitur batas mendapatkan parameter bentuk dari gambar dengan mendeskripsikan fitur batas. Diantaranya, metode deteksi garis paralel dengan Transformasi Hough dan Histogram arah tepi adalah metode klasik. Transformasi Hough adalah metode yang menghubungkan piksel marjinal dengan menggunakan global karakteristik ke batas tertutup lokal. Ide dasarnya adalah dualitas dari titik dan garis. Histogram arah tepi memperoleh batas gambar dengan kalkulus, kemudian membuat histogram ukuran dan arah batas dan cara yang biasa adalah dengan membuat matriks arah gradien skala abu-abu (Ballard, 1981).

b) Metode *Fourier Shape Descriptor* (Deskriptor Bentuk Fourier)

Ide dasar dari deskriptor bentuk Fourier adalah menggunakan transformasi Fourier dari batas objek sebagai deskripsi bentuk, hal ini mengubah masalah dua dimensi menjadi masalah satu dimensi dengan memanfaatkan penutupan batas-batas wilayah. Tiga jenis ekspresi bentuk diturunkan dari titik-titik batas, yaitu fungsi kelengkungan, jarak pusat dan fungsi koordinat kompleks (Gong et al., 2019).

c) Metode *Geometric Parameter* (Parameter Geometris)

Metode parameter geometris sebagian besar mencakup momen, luas, keliling, kebulatan, eksentrisitas, arah poros dan momen invarian aljabar daerah. Hal ini harus didasarkan pada pemrosesan gambar dan segmentasi gambar ketika mengacu pada ekstraksi parameter bentuk, dan keakuratan parameter akan dipengaruhi oleh hasil Segmentasi (Gong et al., 2019).

4. Analisis *Receiver Operating Characteristic* (ROC)

Grafik *reciever operating characteristic* (ROC) adalah teknik untuk memvisualisasikan, mengatur, dan memilih pengklasifikasi berdasarkan kinerjanya. Grafik ROC telah lama digunakan dalam teori deteksi

sinyal untuk menggambarkan pertukaran antara tingkat hit rate dan tingkat alarm palsu dari pengklasifikasi. Analisis ROC telah diperluas untuk digunakan dalam memvisualisasikan dan menganalisis perilaku sistem diagnostik (Fawcett, 2005).

Salah satu pengadopsi awal grafik ROC dalam pembelajaran mesin adalah Spackman (1989), yang mendemonstrasikan nilai kurva ROC dalam mengevaluasi dan membandingkan algoritme. Beberapa tahun terakhir ini telah terjadi peningkatan penggunaan grafik ROC dalam komunitas pembelajaran mesin, sebagian karena kesadaran bahwa akurasi klasifikasi sederhana sering kali merupakan metrik yang buruk untuk mengukur kinerja. Selain menjadi metode grafik kinerja yang secara umum berguna, metode ini memiliki sifat yang membuatnya sangat berguna untuk domain dengan distribusi kelas yang miring dan biaya kesalahan klasifikasi yang tidak sama. Karakteristik ini menjadi semakin penting karena penelitian terus berlanjut di bidang pembelajaran yang sensitif terhadap biaya dan pembelajaran dengan adanya kelas yang tidak seimbang (Fawcett: 2006).

Kurva ROC adalah plot dua dimensi yang mengilustrasikan seberapa baik sistem pengklasifikasi bekerja saat nilai batas diskriminasi diubah pada rentang variabel prediktor. Sumbu x atau variabel independen adalah tingkat positif palsu untuk tes prediktif. Sumbu y atau variabel dependen adalah tingkat positif yang benar untuk tes prediktif. Setiap titik dalam ruang ROC adalah pasangan data positif positif/salah positif untuk nilai batas diskriminasi dari tes prediktif (Nahm, 2021). Jika distribusi probabilitas untuk true positive dan false positive diketahui, kurva ROC dapat diplot dari fungsi distribusi kumulatif. Pada sebagian besar aplikasi nyata, sampel data akan menghasilkan satu titik dalam ruang ROC untuk setiap pilihan batas diskriminasi. Hasil yang sempurna adalah titik (0, 1) yang mengindikasikan 0% positif palsu dan 100% positif sejati. Pembuatan tingkat positif sejati dan positif palsu mengharuskan kita memiliki metode standar emas untuk mengidentifikasi kasus positif sejati dan negatif sejati (Yang & Berdine, 2017).

- *Confusion Matrix*

  Confusion matrix atau matriks kebingungan (juga dikenal sebagai matriks kesalahan) adalah

tabel kontingensi yang digunakan untuk menggambarkan kinerja sistem pengklasifikasian/klasifikasi, ketika kebenarannya diketahui (Yang & Berdine, 2017). Dalam sebuah matriks kebingungan, setiap kolom (atau baris) melaporkan angka-angka dalam kelas yang diprediksi, misalnya, jumlah penyakit yang diprediksi atau keadaan normal yang diprediksi, sementara setiap baris (atau kolom) melaporkan angka-angka dalam kelas yang sebenarnya, misalnya, jumlah penyakit yang sebenarnya atau keadaan normal yang sebenarnya (Ulinuha, 2022). Dalam tabel kontingensi 2×2 yang umum, ada empat angka yang dilaporkan:

1) *True Positive* (TP); juga disebut sensitivitas; ukuran proporsi positif, yang diprediksi dengan benar mengingat itu benar-benar positif)
2) False Negative (FN; ukuran proporsi negatif yang diprediksi, mengingat itu benar-benar positif)

3) False Positive (FP; ukuran proporsi positif yang diprediksi, mengingat itu benar-benar negatif), dan
4) True Negative (TN; juga disebut spesifisitas; ukuran proporsi negatif yang diprediksi, mengingat itu benar-benar negatif) (Yang & Berdine, 2017).

Dengan menggunakan nilai-nilai ini, dapat menghitung berbagai fungsi matriks evaluasi dari *confusion matrix* itu sendiri, seperti:

- Accuracy (Akurasi): $\frac{(TP + TN)}{(TP + TN + FP + FN)}$ (2.1)
- Precision (Presisi): $\frac{TP}{(TP + FP)}$ (2.2)
- Recall (Sensitivitas): $\frac{TP}{(TP + FN)}$ (2.3)
- F1 Score: $2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$ (2.4)

Adapun pengertian dari masing-masing fungsi metrik evaluasi diatas ialah :

a) Akurasi (*Accuracy*) adalah matriks evaluasi yang mengukur seberapa baik model membuat prediksi yang benar (Yang & Berdine, 2017). Dalam konteks pengolahan citra, akurasi memberikan gambaran mengenai seberapa akurat sistem yang

digunakan untuk memprediksi bagian-bagian dari suatu citra yang akan diolah.

b) Presisi (*Precision*) adalah matriks evaluasi yang mengukur seberapa baik model prediksi yang benar untuk kelas positif dari total prediksi positif yang dilakukan (Yang & Berdine, 2017). Dalam konteks pengolahan citra, presisi memberikan gambaran mengenai seberapa sering sistem yang digunakan memprediksi kelas positif yang benar, diantara semua prediksi positif yang dibuat oleh model.

c) *Recall* atau Sensitifitas adalah matriks evaluasi yang menggambarkan seberapa baik suatu model dalam mengidentifikasi kelas positif yang benar (Yang & Berdine, 2017). Dalam konteks pengolahan citra, *Recall* akan menunjukkan kemampuan sistem yang digunakan untuk membaca daerah positif yang sebenarnya.

d) *F1-Score* sendiri merupakan matriks evaluasi yang mencerminkan keseimbangan antara Presisi (*Precision*) dan Sensitivitas (*Recall*) (Yang & Berdine, 2017). Dalam konteks pengolahan citra, *F1 Score* memberikan gambaran mengenai

seberapa baik sistem yang digunakan dalam mengklasifikasi baik itu kelas positif maupun kelas negatif secara akurat.

Dalam pengolahan citra digital, matriks kebingungan (*confusion matrix*) adalah alat evaluasi yang penting dalam mengukur kinerja dari model sistem yang digunakan. Ini membantu untuk memahami seberapa baik atau buruk model sistem tersebut bekerja dalam mendeteksi piksel atau objek dalam citra (Gong et al., 2019). Selain itu, adapun kegunaan utama dari *confusion matrix* dalam pengolahan citra ialah:

1) Evaluasi Kinerja Model

   Memungkinkan analisis tentang seberapa baik model sistem mendeteksi dan mengidentifikasi kelas-kelas yang berbeda dalam citra.

2) Deteksi Kesalahan

   Memungkinkan identifikasi kesalahan yang dibuat oleh model.

3) Pemahaman Keseimbangan Kelas

   Berguna dalam memahami keseimbangan kelas antar dataset citra.

4) Optimasi Model

Dengan memahami dimana model melakukan kesalahan, hal tersebut dapat digunakan sebagai parameter untuk mengoptimalkan teknik yang digunakan selama pengolahan.

5) Validasi Model

Confusion Matrix membantu dalam memvalidasi model yang digunakan, memastikan bahwa model mampu mendeteksi dengan benar dan dapat diandalkan untuk penelitian atau penggunaan model kedepannya.

5. Deteksi Tepi (*Edge Detection*)

Deteksi tepi adalah proses dalam pengolahan citra digital yang bertujuan untuk mengidentifikasi titik-titik dalam citra dimana intensitas cahaya mengalami perubahan tajam atau diskontinuitas. Titik-titik ini biasanya menunjukkan batas objek dalam citra. Deteksi tepi berfungsi dengan mendeteksi perubahan yang signifikan di sekitar piksel gambar (Gonzales et al., 2009). Deteksi tepi sendiri memiliki beberapa metode yang menggunakan operator-operator tertentu sebagai algoritma pendeteksiannya. Adapaun metode-metode tersebut ialah:

a) Operator Sobel

Operator ini menggunakan dua kernel konvolusi 3x3 yang berbeda untuk mendeteksi tepi horizontal dan vertikal. Operator Sobel menghitung perkiraan gradien intensitas gambar unutk masing-masing piksel, yang kemudian digunakan untuk menemukan tepi. Operator ini juga memberikan efek perataan yang membantu mengurangi dampak dari noise dalam gambar (Gonzales & Woods, 2018)

b) Operator Prewitt

Mirip dengan Sobel, operator Prewitt juga menggunakan dua kernel konvolusi untuk mendeteksi tepi horizontal dan vertikal. Namun, kernel yang digunakan dalam operator Prewitt lebih sederhana dan berbasis integer, membuat lebih cepat dihitung akan tetapi kurang akurat dibandingkan dengan operator Sobel (Jain, 1989).

c) Operator Roberts Cross

Operator ini menggunakan dua kernel konvolusi 2x2 untuk menghitung gradien gambar. Metode ini cepat tetapi lebih sensitif terhadap noise dibandingkan metode lainnya karena tidak

melakukan perataan gambar sebelumnya (Pratt, 2001).

d) Laplacian of Gaussian (LoG)

Metode ini pertama-tama melakukan perataan gambar menggunakan filter Gaussian untuk mengurangi noise, kemudian menerapkan operator Laplacian untuk mendeteksi tepi. LoG sangat efektif untuk mendeteksi tepi di gambar dengan intensitas cahaya yang bervariasi dengan cepat. (Marr & Hildreth, 1980)

e) Deteksi Tepi Canny

Algoritma “Canny” dikembangkan oleh John Canny dan melibatkan beberapa tahap seperti penghalussan gambar dengan filter Gaussian, menghitung gradien intensitas, penerapan non-maximum suppression untuk menyingkirkan tanggapan palsu, dan pelacakan melalui histeresis untuk menemukan tepi yang sebenarnya. Metode ini dikenal karena keseimbangan baik antara deteksi, lokalitas, dan respons terhadap satu tepi dengan tepi lainya (Canny, 1986)

6. GNU Octave

GNU Octave adalah sebuah perangkat lunak sumber terbuka (*open source*) yang sering sekali digunakan untuk komputasi numerik. Octave sendiri menyediakan lingkungan yang mirip dengan perangkat lunak MATLAB dan juga dapat menjalankan sebagian besar skrip MATLAB tanpa harus memodifikasi skrip tersebut (Nagar, 2017). Octave sendiri digunakan secara luas dalam komunitas ilmiah dan teknik untuk tugas-tugas seperti analisis data, visualisasi data, dan pemrosesan sinyal.

- Fitur Utama Octave

a) Komputasi Berbasis Array

   Octave memungkinkan komputasi dengan *array* dan matriks, mirip dengan MATLAB, yang sangat berguna untuk perhitungna matematis dan teknik (Nagar, 2017).

b) Plotting dan Visualisasi Data

   Octave memiliki fungsi plotting yang kuat untuk visualisasi data dua dimensi dan tiga dimensi, juga dapat memudahkan analisis dan interpretasi data (Nagar, 2017).

c) Kompabilitas dengan aplikasi MATLAB

Octave memiliki kompabilitas yang cukup banyak dengan program-program dari aplikasi MATLAB, sehingga dapat menjalankan program-program dari MATLAB tanpa harus mengubah fungsi-fungsi yang ada. Hal ini menjadikan Octave sebagai alternatif yang lebih terjangkau dan fleksibel bagi pengguna MATLAB (Rogel-Salazar, 2014).

d) Lingkungan Pemograman Interaktif

Octave menyediakan antarmuka baris perintah yang interaktif serta GUI, yang memungkinkan pengguna untuk bereksperimen langsung dan dapat menguji kode yang para pengguna dapatkan secara langsung (Pajankar & Chandu, 2020)

7. Operasi Morfologi

Operasi morfologi adalah serangkaian teknik pemrosesan citra yang digunakan untuk menganalisis dan memproses bentuk serta struktur dalam gambar digital. Teknik ini didasarkan pada teori himpunan dan terutama diterapkan pada gambar biner (hitam-putih), meskipun juga dapat digunakan pada gambar berwarna atau skala abu-abu (*grayscale*). Operasi

morfologi digunakan dalam berbagai aplikasi, termasuk analisis bentuk, penyaringan gambar, segmentasi, dan pengenalan objek (Gonzales & Woods, 2018).

- Jenis-Jenis Operasi Morfologi

  Menurut Jean Serra (2012) didalam bukunya yang berjudul "*Mathematical Morphology and Its Application to Image and Signal Processing*", jenis-jenis operasi morfologi adalah sebagai berikut.

a) Dilasi (*Dilation*)

   Operasi dilasi (*dilation*) berfungsi untuk memperluas objek yang ada didalam gambar. Hal ini dilakukan dengan menambahkan piksel ke tepi objek tersebut dengan mengisi celah kecil dalam objek, menghubungkan objek yang berdekatan, dan memperluas fitur yang terdeteksi.

b) Erosi (*Erotion*)

   Operasi erosi (*erotion*) berfungsi untuk mengikis atau mengurangi ukuran objek didalam gambar. Hal ini dilakukan dengan menghilangkan piksel dari tepi suatu objek. Operasi ini juga dapat menghilangkan noise,

memisahkan objek yang berdekatan, dan memperkecil fitur yang ada.

c) *Opening*

Operasi ini merupakan kombinasi dari operasi erosi yang diikuti oleh operasi dilasi. Objek kecil yang terdeteksi atau noise akan dihilangkan sembari menjaga bentuk dan ukuran objek tetap dan lebih besar.

d) *Closing*

Operasi ini merupakan kombinasi dari operasi dilasi yang diikuti oleh operasi erosi. Dimana lubang-lubang kecil dalam objek akan diisi dan menghubungkan celah-celah kecil yang ada diantara objek-objek tersebut.

e) *Hit-or-Miss Transform*

Operasi ini berfungsi untuk mendeteksi pola-pola tertentu yang ada didalam gambar. Operasi ini akan melakukan pengenalan pola dan deteksi fitur-fitur tertentu yang terdeteksi pada citra binernya.

8. Rule-Based Classification

Rule-based Classification merupakan sebuah rule based classifier yang digunakan sebagai sekumpulan

dari rule IF-THEN untuk klasifikasi. Sebuah aturan IF-THEN dengan form. Proses data klasifikasi memiliki dua tahapan, yang pertama adalah Learning: dimana training data dianalisa dengan menggunakan sebuah algoritma klasifikasi (Suryani et al., 2019). Dan yang kedua adalah Classification: dimana pada tahap ini test data digunakan untuk mengestimasi ketepatan dari classification rules.

Jika keakuratan yang dikondisikan dan yang diperkirakan dapat diterima, rule tersebut dapat diaplikasikan pada klasifikasi lainya dari tuple data yang baru. Metode rule-based classification memiliki kelebihan jika diterapkan pada domain yang sederhana, maka rule-based mudah untuk diverifikasi dan divalidasi. (Subali & Fatichah, 2019).

Menurut Ling Liu dan M. Tamer Özsu (2009), jenis klasifikasi *Rule-Based* memiliki beberapa konsep utama, yaitu:

a) Struktur Aturan

   Antecedent: Bagian IF dari aturan, yang menentukan kondisi atau serangkaian kondisi pada atribut data.

Consequent: Bagian THEN dari aturan, yang menetapkan label kelas ketika kondisi antecedent terpenuhi.

b) Induksi Aturan

Proses menghasilkan aturan dari dataset disebut induksi aturan. Teknik seperti algoritma penutup berurutan (misalnya, CN2, RIPPER) sering digunakan untuk mengekstraksi aturan langsung dari data.

c) Pengurutan Aturan

Aturan dapat diurutkan berdasarkan kualitas atau tingkat kepercayaan mereka, membentuk daftar keputusan. Ketika instance baru perlu diklasifikasikan, aturan dievaluasi secara berurutan sampai satu terpenuhi.

d) Evaluasi dan Penyempurnaan

Setelah aturan diinduksi, mereka dievaluasi menggunakan dataset pengujian terpisah. Kesalahan klasifikasi dianalisis untuk menyempurnakan dan meningkatkan set aturan.

9. Penyakit yang Menyerang Daun Padi

Padi merupakan tanaman pangan utama bagi sebagian besar populasi di dunia, salah satunya ialah

negara kita sendiri Indonesia. Namun meskipun padi memiliki kekuatan adaptasi yang luar biasa, padi juga merupakan makhluk hidup yang rentan terkena penyakit bila tidak dirawat dengan baik. Pada dasarnya ada banyak sekali jenis-jenis penyakit yang dapat menyerang tanaman padi, terutama pada daunnya. Seperti halnya daun membengkak, distorsi atau deformasi, perubahan warna, dan lain sebagainya (Siregar & Sulardi, 2019). Adapun penyakit-penyakit tersebut ialah sebagai berikut.

a. Penyakit *Leaf Smut*

Penyakit *Leaf Smut* (Angus Daun) ini disebabkan oleh jamur *Entyloma oryzae*. Gejala utama yang terlihat adalah adanya bintik-bintik hitam kecil pada kedua sisi daun. Penyakit ini menyerang semua stadia pertanaman padi baik di musim hujan maupun di musim kemarau. Penyakit ini termasuk salah satu penyakit yang terbawa dari benih. Bentuk bintik-bintik hitam itu sendiri terkadang berbentuk seperti sebuah garis pendek, lingkaran, atau elips dengan ukuran panjang 0,5 – 5 mm dan lebar 0,5 – 1,5 mm. Penyakit ini biasanya dianggap sebagai penyakit minor, tetapi dapat menyebar dalam

kondisi lingkungan tertentu (Siregar & Sulardi, 2019).

Gambar 2.6 Penyakit Leaf Smut pada Daun Padi

b. Penyakit Hawar (Kresek)

Penyakit hawar, juga dikenal sebagai kresek, pada tanaman padi disebabkan oleh bakteri *Xanthomonas oryzae*. Patogen ini dapat menginfeksi tanaman padi pada semua fase pertumbuhan tanaman, mulai dari persemaian sampai menjelang panen. Umumnya penyebaran penyakit kresek ini berasal dari bekas jerami, singgang, benih padi yang terinfeksi bakteri dan juga gulma di sekitar lahan. Gejala diawali dengan timbulnya beracak abu-abu

(kekuningan), umumnya terjadi di tepi daun. Dalam perkembangannya, gejala akan meluas, dan membentuk hawar memanjang (*blight*), dan akhirnya akan membuat daun mengering (Siregar & Sulardi, 2019).

Gambar 2.7 Penyakit Hawar pada Daun Padi
(Sumber: plantix.com)

c. Penyakit Blast

Penyakit blast pada daun padi, yang juga dikenal sebagai bercak belah ketupat, adalah salah satu penyakit paling merusak pada tanaman padi. Penyakit ini disebabkan oleh jamur *Magnaporthe oryzae.* Gejala awal muncul sebagai bercak kelabu hingga hijau kebiruan yang berubah menjadi coklat dengan pusat abu-abu. Lesi ini bisa membentuk

pola berbentuk berlian atau mata ikan atau juga membentuk sebuah belah ketupat (Irwan & Mu'min, 2020).

Gambar 2.8 Penyakit Blast pada Daun Padi
(Sumber: nufarm.com)

d. Penyakit Brown Spot

Penyakit brown spot pada padi, juga dikenal sebagai bercak coklat, adalah penyakit yang disebabkan oleh jamur *Bipolaris oryzae* (dulu dikenal sebagai *Helminthosporium oryzae*). Penyakit ini adalah salah satu penyakit daun padi yang umum ditemukan di berbagai negara penghasil padi. Gejala utama adalah bercak-bercak kecil berbentuk bulat atau oval berwarna coklat dengan pusat berwarna abu-abu atau putih. Ukuran bercak biasanya 0,5-2 cm (Irwan & Mu'min, 2020).

Gambar 2.9 Penyakit Brown Spot pada Padi
(Sumber: doa.gov.lk)

B. Kajian Penelitian yang Relevan

Adapun kajian-kajian yang relevan dengan penelitian ini adalah :

1. Penelitian mengambil insipari dari Jurnal Teknologi dan Informasi Walisongo (*Walisongo Journal of Information Technology WJIT*) yang berjudul ”Steganografi Pada Digital Image Menggunakan Metode *Least Significant Bit Insertion*” oleh Siti Nur’aini. Jurnal ini mengandung sebuah cara lain dalam melaksanakan pengolahan citra digital, yaitu dengan metode *Least Significant Bit Insertion*. Beberapa kajian pustaka dan metode penelitian yang akan dilakukan mengambil referensi dari penelitian ini.

2. Penelitian lain yang relevan ialah sebuah disertasi yang berjudul ”Segmentasi Tulang Wajah Tiga Dimensi Berdasarkan Sudut Simpangan” oleh Masy Ari Ulinuha. Hampir seluruh metode yang digunakan dalam penelitian ini mengambil referensi dari disertasi tersebut, bedanya penelitian ini mengfokuskan kepada bidang pertanian khususnya pada tanaman padi dan akan menggunakan metode ekstraksi fitur, sedangkan disertasi ini difokuskan dalam mensegmentasi tulang wajah secara 3 dimensi menggunakan metode sudut simpangan.
3. Penelitian lain yang relevan adalah sebuah jurnal yang berjudul ”Klasifikasi Penyakit Daun Padi Menggunakan Metode *Random Forest* dan *Color Histogram”* yang ditulis oleh Safirah Agustiani, dkk. Jurnal ini berisi tentang penjelasan klasifikasi beberapa penyakit yang terjadi pada daun padi. Bedanya dengan penelitian yang akan dilakukan ialah jurnal tersebut menggunakan metode *Random Forest* dan *Color Histrogram* untuk melakukan pengklasifian. Dengan kata lain, penelitian tersebut hanya mengklasifikasi gejala penyakit yang terjadi

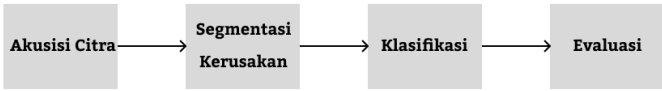
pada daun dari perbedaan warna. Sedangkan penelitian ialah identifikasi menggunakan metode ekstraksi fitur bentuk.

4. Penelitian berikutnya yang relevan ialah sebuah jurnal yang dipublish pada Jurnal Edukasi dan Penelitian Informatika (JEPIN) dengan judul ”Segmentasi Kerusakan Daun Padi pada Citra Digital” oleh Khoerul Anwar dan Sigit Setyowibowo. Jurnal ini menjelaskan tentang proses mensegmentasi kerusakan yang terjadi pada daun padi. Bedanya dengan penelitian yang akan dilakukan ialah jurnal ini hanya sampai kepada segmentasi kerusakan saja. Tidak menjelaskan penyakit apa yang sedang diderita oleh daun padi. Dan juga jenis kerusakan yang disegmentasi hanya satu bentuk saja. Sedangkan penelitian yang akan dilakukan, akan mencakup beberapa jenis kerusakan yang terjadi pada daun dan mengidentifikasi gejala penyakitnya.

BAB III
METODE PENELITIAN

Penelitian ini merupakan penelitian kuantitatif, karena data yang diambil dan yang akan diolah merupakan bilangan. Namun penelitian ini tidak menggunakan metode penelitian kuantitaif pada umumnya. Metode baru diusulkan untuk menyesuaikan dengan jenis penelitian yang akan dilakukan menyesuaikan dengan statitiska nilai piksel citra daun padi.

Adapun metode penelitian yang diusulkan dapat dilihat pada diagram metode penelitian pada gambar dibawah ini.



Gambar 3.1 Diagram Penelitian

A. Akusisi Citra

Data akan diambil melalui kamera digital dari *smartphone* Realme C25s yang memiliki ukuran hingga 48 MP. Data yang akan diambil tersebut merupakan citra digital yang akan berjumlah 50-100 data. Diketahui sebuah citra memiliki resolusi dengan persamaan *n x n* piksel, contoh: 1980 x 1080 piksel, 800 x 600 piksel, dan

seterusnya. Dalam penelitian ini, nilai *n* tersebut akan disesuaikan hingga tidak lebih dari 1000. Hal ini dikarenakan nilai dari *n* tersebut sangat mempengaruhi kinerja sistem yang akan dijalankan. Semakin besar nilai *n*, maka akan sistem akan semakin berat pula. Oleh karena itu, bila nilai *n* yang didapat dari sebuah data lebih dari 1000, maka data tersebut akan disesuaikan (*resize*). Nilai *n* pada resolusi skala yang disesuaikan akan berjarak sekitar 400-1000 piksel.

B. Segmentasi Kerusakan Daun

Setelah citra-citra tersebut disesuaikan, maka segmentasi akan dilakukan. Sebelum melakukan segmentasi, citra-citra tersebut akan diolah terlebih dahulu, seperti halnya mengubah citra tersebut menjadi citra skala abu-abu (*grayscale*), mendeteksi tepi untuk mengubah citra tersebut menjadi citra biner, melakukan operasi morfologi untuk menutup bagian yang tidak dinginkan, dan melakukan operasi morfologi *filling* untuk mengisi bagian yang telah tersegmentasi.

C. Ekstraksi Fitur

Setelah segmentasi dilakukan, maka langkah berikutnya merupakan ekstraksi fitur. Ekstraksi fitur sendiri merupakan salah satu proses untuk memperoleh informasi

dari sebuah citra (Agustiani et al., 2022), oleh karena itu langkah ini akan dilakukan untuk melihat fitur apa yang dimiliki oleh citra hasil segmentasi tersebut, seperti halnya jumlah objek yang terdeteksi pada citra hasil segmentasi itu.

D. Klasifikasi

Tahap klasifikasi ini akan dilakukan menggunakan metode *Rule-Based Classification*, yang mana jumlah objek dari citra hasil segmentasi tersebut dihitung, lalu dicocokkan dengan beberapa referensi yang didapat dan dijadikan sebagai aturan untuk mengklasifikasi citra-citra yang sudah tersegmentasi.

E. Evaluasi

Evaluasi yang dilakukan ialah menghitung nilai hasil klasifikasi yang telah dilakukan berdasarkan nilai dari *receiver operatic characteristic* (ROC). Adapun nilai ROC yang dipilih sebagai metode untuk mengevaluasi hasil klasifikasi tersebut ialah nilai dari *confusion matrix* dua kelas. Nilai-nilai metrik yang akan digunakan dalam perhitungan evaluasi ini antara lain, nilai akurasi (*accuracy*), nilai presisi *(precision*), nilai *recall* (sensitifitas), dan nilai *f1* score (keseimbangan).

BAB IV
PEMBAHASAN

A. Akusisi Citra

Data yang diambil berasal dari area pesawahan yang ada di daerah Kelurahan Karang Layung, Sukra, Indramayu. Data ini hanya dapat diambil kita menjelang musim panen tiba. Hal ini dikarenakan gejala penyakit bisa dapat dilihat disaat menjelang musim panen tiba. Warna dari daun-daun padi pada musim tersebut akan terlihat memudar, ini dapat membuat gejala-gejala penyakit yang dituju dapat terlihat dengan jelas.

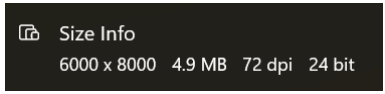
Daun-daun yang sudah menunjukkan gejala-gejala penyakit akan dipotong (diambil) dan dikumpulkan. Terdapat total 50 (lima puluh) daun yang berhasil dikumpulkan dan dilihat sesuai dengan kriteria gejala penyakit yang dipilih untuk diteliti.

Setelah daun-daun tersebut dikumpulkan maka hal yang selanjutnya dilakukan ialah memotretnya, karena data yang dibutuhkan dari penilitian ini merupakan citra digital. Alat yang digunakan untuk memotret daun-daun tersebut berupa sebuah *smartphone* dengan merek Realme C25s, yang memiliki ukuran kamera sebesar 48 mega piksel. Sebelum diambil citranya, daun yang memiliki gejala-gejala

penyakit tersebut akan ditaruh diatas sebuah kertas putih yang dijadikan sebagai *background* untuk data citra tersebut.

Pengambilan citra dilakukan dengan bantuan cahaya putih dari sebuah lampu belajar, juga sistem *flash* yang ada pada *smartphone*. Setelah seluruh citra dari daun-daun yang sudah dikumpulkan tersebut sudah diambil, maka metode berikutnya dapat dilakukan.

Sebelum citra-citra tersebut dimulai untuk diolah, ukuran dari citra-citra tersebut akan diperiksa terlebih dahulu apakah nilai dari masing-masing citra tersebut memiliki nilai piksel tidak lebih dari 1000x1000 piksel. Hal ini dapat diketahui dengan cara membuka detail citra tersebut yang tersimpan didalam *device*.

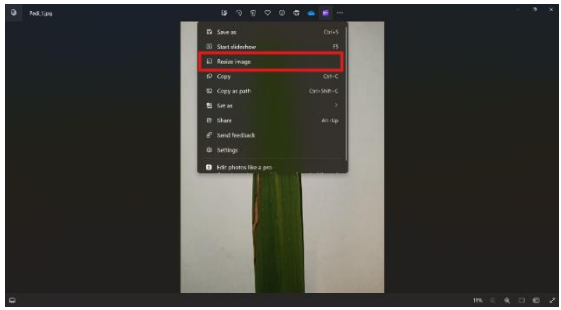


Gambar 4.1 Detail Size Citra yang Didapat

Berdasarkan detail yang ditemukan seperti gambar diatas, bahwasannya data citra yang didapat memiliki nilai piksel yang melibihi 1000x1000 piksel. Oleh karena itu, *resize* atau pengubahan nilai piksel dilakukan agar dapat mempermudah pengolahannya. *Resize* atau mengubah

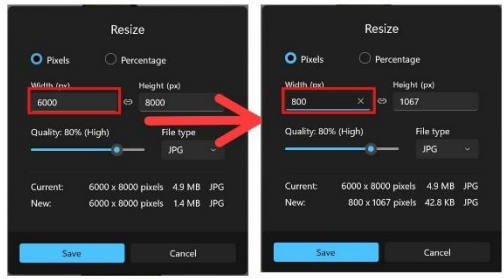
ukuran nilai piksel ini dilakukan secara manual menggunakan *tools* yang sudah tersedia dalam *device*.

Didalam *software* bawaan milik Windows 11, yaitu aplikasi "Photos", ada sebuah alat yang dapat digunakan untuk me-*resize* ukuran piksel dari sebuah citra.



Gambar 4.2 Opsi mengubah size citra dalam software 'Photos'

Ketika memilih opsi "*Resize image*" yang ada pada *software* tersebut, maka mengubah ukuran piksel dari suatu citra dapat dilakukan. Pada pilihan "*Resize image*" tersebut terdapat beberapa pilihan seperti mengubah ukuran lebar dan ukuran tinggi dari citra tersebut. Adapun ukuran yang diubah hanyalah ukuran lebar dari citra itu, menjadi 800 piksel.



Gambar 4.3 Resize citra yang didapat

Pengubahan ukuran citra ini (*resize*) dilakukan agar dalam proses pengolahan citra yang dijalankan tidaklah berat dan mudah. Pengubahan ukuran citra tidak berpengaruh besar pada pengolahan citra pada umumnya, karena pengubahan ukuran (*resize*) ini hanya memperkecil ukuran dari suatu citra tersebut.

Setelah melakukan *resize* maka data akan diseleksi dan diklasifikasi menjadi dua kelas, sesuai dengan jenis penyakit yang diduga dimiliki oleh masing-masing data tersebut secara manual. Terdapat total 22 data yang diduga mengidap penyakit hawar, dan total 28 data yang diduga mengidap penyakit *leaf smut*. Oleh karena itu, penyakit yang fokuskan pada penelitian ini hanyalah penyakit hawar daun dan angan daun (*leaf smut*) Adapun contoh dari citra-citra yang diambil dan diklasifikasikan secara manual tersebut ialah.

TABEL 4.1 Contoh Citra yang Diambil

Hawar				
Leaf Smut				

B. Segmentasi Kerusakan Daun

Dalam melakukan segmentasi kerusakan pada citra daun padi, ada beberapa tahap yang harus dilakukan. Adapun tahap-tahap yang akan dilakukan ialah:

1. Mengubah Citra Asli menjadi Citra *Grayscale*

   Untuk mengubah citra asli menjadi citra skala abu-abu (*grayscale*) fungsi yang digunakan didalam aplikasi GNU Octave ialah fungsi 'rgb2gray'. Fungsi ini akan secara otomatis mengubah seluruh warna dari citra asli menjadi abu-abu.

TABEL 4.2 Hasil Pengolahan

Citra Asli menjadi Citra *Grayscale*

Hawar				
Leaf Smut				

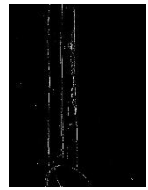
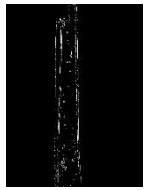
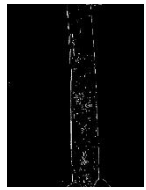
2. Deteksi Tepi menggunakan Operator 'Sobel'

Deteksi tepi adalah proses untuk menemukan tepi (boundary) dalam citra yang menunjukkan perubahan mendadak dalam intensitas piksel (Gonzales & Woods, 2018). Salah satu metode yang paling umum digunakan untuk deteksi tepi adalah operator Sobel.

Operator Sobel menggunakan filter konvolusi untuk mendeteksi tepi dengan menghitung perkiraan gradien intensitas citra. Operator ini terdiri dari dua kernel 3x3 yang diaplikasikan pada citra: satu untuk mendeteksi perubahan intensitas secara horizontal (Gx) dan satu lagi untuk perubahan intensitas secara vertikal (Gy) (Gonzales & Woods, 2018).

Pada aplikasi octave sendiri, fungsi yang digunakan untuk memanggil fungsi deteksi tepi yang menggunakan operator Sobel ialah "edges(image, 'Sobel')". Adapun hasil pendeteksian tepi menggunakan operasi Sobel pada citra *grayscale* tersebut, ialah sebagai berikut.

TABEL 4.3 Hasil Deteksi Tepi 'Sobel'

Hawar				
Leaf Smut				

3. Operasi Morfologi 1 (*Closing*)

Pada hasil pendeteksian tepi menggunakan operator "Sobel" tersebut dapat dilihat bahwa hasil pendeteksian juga mendeteksi bagian tepi dari daun padi pada citra tersebut. Karena dalam area fokus yang dipilih pada penelitian ini ialah hanya pada bagian daun padi yang

memiliki kerusakan, maka operasi lanjutan akan dilakukan untuk menutup tepian daun padi tersebut.

Adapun operasi yang dilakukan ialah operasi *morfologi*. Operasi *morfologi* adalah serangkaian teknik yang digunakan dalam pemrosesan citra untuk menganalisis dan memodifikasi struktur dan bentuk objek dalam gambar. Operasi ini memanfaatkan teori himpunan untuk melakukan berbagai manipulasi pada citra biner, meskipun juga dapat diterapkan pada citra berwarna atau skala abu-abu. Operasi morfologi membantu dalam identifikasi, analisis, dan pengolahan fitur geometris dari objek dalam gambar (Gonzales & Woods, 2018). Adapun fungsi yang digunakan untuk melakukan operasi morfologi untuk menutup bagian-bagian yang tidak diinginkan adalah seperti gambar.

```
function result_edges = ApplyMorfologi(edges)
  % Menampilkan hasil deteksi tepi
  figure, imshow(edges), title('Hasil deteksi tepi');

  % Initialize an empty mask for multiple areas
  combined_mask = false(size(edges));

  while true
      try
          % Get coordinates for the rectangle to define the area
          disp('Klik dua titik untuk menentukan persegi (sudut kiri atas dan kanan
bawah)');
          [x, y] = ginput(2);
          x = round(x);
          y = round(y);

          % Check if there are exactly 2 points
          if length(x) ~= 2 || length(y) ~= 2
              error('Harus memilih tepat dua titik.');
          end

          % Check if the coordinates are within the image bounds
          if any(x < 1 | y < 1 | x > size(edges, 2) | y > size(edges, 1))
              error('Koordinat berada di luar batas gambar. Silakan coba lagi.');
          end

          % Periksa apakah pengguna ingin berhenti
          if isempty(x) || isempty(y)
            disp('Pemberian mask dihentikan oleh pengguna.');
            break;
          end

          % Create a mask for the rectangular area
          mask = false(size(edges));
          mask(y(1):y(2), x(1):x(2)) = true;

          % Combine the current mask with the previous masks
          combined_mask = combined_mask | mask;

          % Apply morphological closing only within the rectangular area
          se = strel('rectangle', [5 5]); % Create a structuring element
          opened_edges = imopen(edges, se);

          % Combine the closed edges within the rectangle with the original edges
outside the rectangle
          final_edges = edges;
          final_edges(combined_mask) = opened_edges(combined_mask);

          % Fill holes only in the specific area defined by the mask
          filled_edges = imfill(final_edges, 'holes');

          % Combine the filled edges within the mask with the original edges
outside the mask
          result_edges = bsxfun(@times, edges, cast(filled_edges, 'like',
```

Gambar 4.4 Program untuk fungsi Operasi Morfologi

Adapun penjelasan untuk masing-masing fungsi dan *command* yang digunakan pada operasi morfologi yang digunakan ialah sebagai berikut.

- **function result_edges = ApplyMorfologi(edges)**
  Mendefinisikan fungsi dengan nama ApplyMorfologi yang menerima input edges dan mengembalikan result_edges.
- **figure, imshow(edges), title("Hasil deteksi tepi")**
  Menampilkan citra hasil deteksi tepi pada sebuah figure dengan judul "Hasil deteksi tepi".
- **combined_mask = false(size(edges))**
  Inisialisasi sebuah mask kosong dengan ukuran yang sama dengan citra edges.
- **while true**
  Memulai loop yang akan terus berjalan hingga dihentikan oleh kondisi di dalamnya.
- **disp('Klik dua titik untuk menentukan persegi (sudut kiri atas dan kanan bawah)')**
  Menampilkan pesan untuk meminta pengguna mengklik dua titik pada gambar.
- **[x, y] = ginput(2)**
  Mengambil koordinat dari dua titik yang diklik oleh pengguna.

- **x = round(x); y = round(y);**

  Membulatkan koordinat x dan y ke nilai integer terdekat.
- **if length(x) ~= 2 || length(y) ~= 2**

  Memeriksa apakah pengguna benar-benar mengklik dua titik.
- **error('Harus memilih tepat dua titik.')**

  Menampilkan pesan error dan menghentikan eksekusi jika pengguna tidak mengklik dua titik.
- **if any(x < 1 | y < 1 | x > size(edges, 2) | y > size(edges, 1))**

  Memeriksa apakah koordinat yang dipilih berada di luar batas gambar.
- **error('Koordinat berada di luar batas gambar. Silakan coba lagi.')**

  Menampilkan pesan error dan menghentikan eksekusi jika koordinat berada di luar batas gambar.
- **mask = false(size(edges))**

  Membuat mask baru dengan ukuran yang sama dengan citra edges, diinisialisasi dengan nilai false.
- **Mask(y(1):y(2), x(1):x(2)) = true**

Mengatur area persegi yang dipilih oleh pengguna pada mask menjadi true.

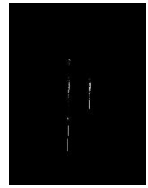
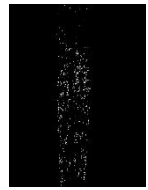
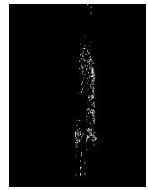
- **combined_mask = combined_mask | mask**;
  Menggabungkan mask saat ini dengan mask sebelumnya menggunakan operasi logika OR.
- **se = strel('rectangle', [5 5])**
  Membuat elemen struktural berbentuk persegi panjang dengan ukuran 5x5 piksel.
- **opened_edges = imopen(edges, se)**
  Menerapkan operasi morfologi 'imopen' (pembukaan) pada citra 'edges' dengan elemen struktural 'se'.
- **final_edges = edges**
  Menginisialisasi final_edges dengan nilai edges.
- **final_edges(combined_mask)=opened_edges(combined_mask)**
  Menggabungkan hasil opened_edges dengan 'edges' hanya pada area yang ditentukan oleh 'combined_mask'.
- **filled_edges = imfill(final_edges, 'holes')**
  Mengisi lubang dalam 'final_edges' hanya pada area yang ditentukan oleh 'combined_mask'.
- **result_edges = bsxfun(@times, edges, cast(filled_edges, 'like', edges))**

Menggabungkan hasil ‘filled_edges’ dengan ‘edges’ untuk menghasilkan ‘result_edges’.

- **figure, imshow(result_edges)**
  Menampilkan citra hasil operasi morfologi.
- **title('Edge-detected Image after Morphological Operation and Hole Filling in Multiple Areas')**:
  Memberikan judul pada gambar yang ditampilkan.
- **user_response = input('Apakah Anda masih mau memberi mask? (Yes/No): ', 's')**
  Meminta input dari pengguna apakah ingin menambahkan mask lagi.
- **if strcmpi(user_response, 'No')**
  Memeriksa apakah jawaban pengguna adalah 'No'.
- **Break**
  Menghentikan loop jika jawaban pengguna adalah 'No'.
- **end**
  Akhir dari loop dan fungsi.

Operasi morfologi yang dilakukan ini akan meminta *user* untuk memilih bagian mana yang ingin ditutup. Adapun hasil dari operasi morfologi *closing* ini ialah sebagai berikut.

Tabel 4.4 Citra Hasil Operasi Morfologi *Closing*

Hawar				
Leaf Smut				

4. Operasi Morfologi 2 (Dilation dan Filling)

Setelah melakukan operasi morfologi *closing*, operasi morfologi berikutnya ialah operasi morfologi *dilation* dan *filling* yang bertujuan untuk menyambungkan beberapa titik-titik objek lalu mengisinya dengan warna putih. Adapun program operasi morfologi yang dilakukan pada langkah berikutnya ialah.

```
% Deteksi objek dalam citra biner hasil
region_props = regionprops(results, 'BoundingBox');
if isempty(region_props)
  disp('Tidak ada objek yang ditemukan dalam citra.');
end

% Mengukur panjang objek (ambil objek pertama jika lebih dari satu)
bounding_box = region_props(1).BoundingBox;
object_length = max(bounding_box(3), bounding_box(4)); % Panjang objek

% Apply morphological closing only within the combined mask area
% Pilih elemen struktural berdasarkan panjang objek
if object_length > 10 % Sesuaikan nilai ini sesuai kebutuhan Anda
  se = strel('rectangle', [10 20]);
else
  se = strel('square', 5);
end
dilated_results = imdilate(results, se);
figure, imshow(dilated_results), title("Dilated Results");

# Apply filling
filled_results = imfill(dilated_results, 'holes');
figure, imshow(filled_results), title("Filled Holes");
```

Gambar 4.5 Program untuk operasi morfologi setelah closing

Adapun penjelasan fungsi dan *command* yang digunakan pada operasi morfologi diatas ialah:

- **results = combined_result;**

  Menginisialisasi variabel ‘results’ dengan nilai dari variabel ‘combined_result’

- **region_props = regionprops(results, 'BoundingBox');**

  Menggunakan fungsi ‘regionprops’ untuk mendeteksi properti objek dalam citra biner ‘results’. Properti yang diminta adalah ‘BoundingBox’, yang memberikan koordinat kotak pembatas untuk setiap objek yang terdeteksi.

- **if isempty(region_props)**

  Memeriksa apakah tidak ada objek yang terdeteksi dalam citra.
- **disp('Tidak ada objek yang ditemukan dalam citra.');**

  Menampilkan pesan bahwa tidak ada objek yang ditemukan dalam citra jika kondisi 'isempty(region_props) terpenuhi'.
- **bounding_box = region_props(1).BoundingBox;**:

  Mengambil properti 'BoundingBox' dari objek pertama yang terdeteksi. 'BoundingBox' adalah vektor yang berisi [x, y, width, height].
- **object_length = max(bounding_box(3), bounding_box(4));**

  Menghitung panjang objek sebagai nilai maksimum antara lebar ('bounding_box'(3)) dan tinggi ('bounding_box'(4)) dari kotak pembatas.
- **if object_length > 10**

  Memeriksa apakah panjang objek lebih dari 10 piksel. Hal ini karena gejala umum penyakit hawar (kresek) ialah memiliki bercak hawar (*blight*) yang **memanjang** (Siregar & Sulardi, 2019).
- **se = strel('rectangle', [10 20]);**

Jika panjang objek lebih dari 10 piksel, elemen struktural berbentuk persegi panjang dengan ukuran 10x20 piksel dipilih.

- **Else**

  Jika panjang objek tidak lebih dari 10 piksel. Hal ini karena gejala umum penyakit *leaf smut* ialah memiliki bintik-bintik hitam dengan ukuran **panjang 0,5 – 5 mm** dan **lebar 0,5 – 1,5 mm** (Siregar & Sulardi, 2019).

- **se = strel('square', 5);**

  Jika panjang objek tidak lebih dari 10 piksel, elemen struktural berbentuk persegi dengan ukuran 5x5 piksel dipilih.

- **dilated_results = imdilate(results, se);**

  Menerapkan operasi morfologi 'dilate' (penggelembungan) pada citra 'results' menggunakan elemen struktural 'se'.

- **figure, imshow(dilated_results), title("Dilated Results");**

  Menampilkan hasil operasi 'dilate' dalam sebuah figure dengan judul "Dilated Results".

- **filled_results = imfill(dilated_results, 'holes');**:

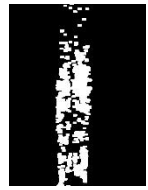
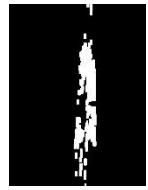
  Mengisi lubang dalam citra hasil 'dilated_results'.

- **figure, imshow(filled_results), title("Filled Holes");**:

Menampilkan hasil operasi pengisian lubang dalam sebuah figure dengan judul "Filled Holes".

Setelah melakukan operasi morfologi diatas, citra hasil akhir akan berbentuk objek-objek putih sesuai dengan jenis gejala penyakit yang diidentifikasi.

Tabel 4.5 Citra Hasil Akhir

Hawar				
Leaf Smut				

5. Melakukan Segmentasi kepada Seluruh Data

Setelah program dirasa cocok untuk digunakan untuk mengsegmentasi kedua jenis penyakit, hal berikutnya ialah memasukkan seluruh data kedalam program yang telah dibuat. Dengan sedikit modifikasi pada program utama, dikarenakan program utama hanya dapat membaca 1 citra

saja, program tersebut dapat digunakan untuk membaca seluruh data. Adapun program utama yang telah dimodifikasi tersebut seperti gambar 4.7 dibawah ini.

```
pkg load image;

% Tentukan folder tempat file gambar berada
folder_path = 'C:\Users\lgiva\OneDrive\Documents\Skripsi\Experiment\1.
Hawar\Dataset'; % Ganti dengan path ke folder Anda
output_folder = 'C:\Users\lgiva\OneDrive\Documents\Skripsi\Experiment\1.
Hawar\Dataset\Hasil'; % Ganti dengan path ke folder output
log_file_path = fullfile(output_folder, 'processed_files.log');

% Membaca daftar file yang sudah diproses
processed_files = {};
if exist(log_file_path, 'file')
    fid = fopen(log_file_path, 'r');
    if fid ~= -1
        processed_files = textscan(fid, '%s');
        fclose(fid);
        if ~isempty(processed_files)
            processed_files = processed_files{1}; % textscan returns a cell array,
get the first cell
        else
            processed_files = {}; % Set to empty if log file is empty
        end
    end
end

% Membuka log file untuk menambahkan file yang diproses
log_file = fopen(log_file_path, 'a');

% Dapatkan daftar semua file .jpg dalam folder
image_files = dir(fullfile(folder_path, '*.jpg'));

% Iterasi melalui setiap file gambar dalam folder
for i = 1:length(image_files)
    % Dapatkan nama file dan path lengkap
    file_name = image_files(i).name;
    file_path = fullfile(folder_path, file_name);

    % Lewatkan file yang sudah diproses
    if any(strcmp(processed_files, file_name))
        continue;
    end

    % Membaca citra
    img = imread(file_path);

    % Merotasi citra agar terlihat vertikal (bila diperlukan)
    # img = rot90(img, -1);

    % Mengubah citra ke grayscale
    GrImg = rgb2gray(img);

    % Mendeteksi kerusakan menggunakan deteksi tepi
    edges = edge(GrImg, 'Sobel'); % Menggunakan metode "Sobel"

    # Menampilkan hasil deteksi tepi
    figure, imshow(edges), title ({"Hasil deteksi tepi - ", file_name});

    % Memanggil fungsi untuk mengisi background dengan warna hitam
    masking_result = ApplyMorfologi(edges);

    % Tampilkan hasil akhir setelah pengisian background
    figure, imshow(masking_result);
    title(['Hasil akhir setelah pemberian masking - ', file_name]);

    % Deteksi objek dalam citra biner hasil
    region_props = regionprops(masking_result, 'BoundingBox');
    if isempty(region_props)
      disp('Tidak ada objek yang ditemukan dalam citra.');
    end

    % Mengukur panjang objek (ambil objek pertama jika lebih dari satu)
    bounding_box = region_props(1).BoundingBox;
    object_length = max(bounding_box(3), bounding_box(4)); % Panjang objek

    % Pilih elemen struktural berdasarkan panjang objek
    if object_length > 5 % Sesuaikan nilai ini sesuai kebutuhan Anda
      se = strel('rectangle', [10 20]);
    else
      se = strel('square', 5);
    end
    dilated_results = imdilate(masking_result, se);
    figure, imshow(dilated_results), title(["Dilated Results - ", file_name]);

    # Apply filling
    filled_results = imfill(dilated_results, 'holes');
    figure, imshow(filled_results), title(["Filled Holes - ", file_name]);

    % Simpan hasil ke dalam file
    [~, name, ext] = fileparts(file_name);
    output_file_path = fullfile(output_folder, [name, '_processed', ext]);
    imwrite(filled_results, output_file_path);
    disp(['Hasil disimpan ke ', output_file_path]);
end

% Menutup log file
fclose(log_file);
```

Gambar 4.6 Program untuk membaca seluruh data

Adapun penjelasan untuk masing-masing fungsi dan *command* yang digunakan dalam program diatas terbagi menjadi beberapa bagian, yaitu sebagai berikut.

- Bagian 1 : Inisialisasi dan Persiapan

Gambar 4.7 Bagian 1 : Inisialisasi dan Persiapan

pkg load image :

Memuat paket image di Octave yang menyediakan fungsi untuk pemrosesan citra.

folder_path :

Menentukan path ke folder yang berisi file gambar yang akan diproses. Untuk folder yang berisi citra *Leaf Smut* menggunakan jalur "\2. Leaf Smut\Dataset".

output_folder :

Menentukan path ke folder tempat menyimpan hasil pemrosesan gambar. Untuk folder yang berisi citra *Leaf Smut* menggunakan jalur "\2. Leaf Smut\Dataset\Hasil".

log_file_path:

Menentukan path ke file log yang digunakan untuk mencatat file yang sudah diproses.

- Bagian 2 : Membaca Daftar File ang sudah Diproses

  Bagian ini dilakukan untuk mengantisipasi adanya *program failure* atau kesalahan-kesalahan lain yang mengakibatkan aplikasi Octave tidak berfungsi atau *force-close* (tertutup secara paksa).

```
% Membaca daftar file yang sudah diproses
processed_files = {};
if exist(log_file_path, 'file')
    fid = fopen(log_file_path, 'r');
    if fid ~= -1
        processed_files = textscan(fid, '%s');
        fclose(fid);
        if ~isempty(processed_files)
            processed_files = processed_files{1}; % textscan returns a cell array, get the
first cell
        else
            processed_files = {}; % Set to empty if log file is empty
        end
    end
end
```

Gambar 4.8 Bagian 2 : Membaca Daftar File yang sudah Diproses

processed_files = {}; :

Inisialisasi variable ‘processed_files’ sebagai array kosong.

if exist(log_file_path, 'file') :

Memeriksa apakah file log ada.

fid = fopen(log_file_path, 'r'); :

Membuka file log untuk dibaca.

processed_files = textscan(fid, '%s'); :

Membaca file log dan menyimpan daftar file yang sudah diproses ke dalam 'processed_files'.

fclose(fid); : Menutup file log.

if ~isempty(processed_files) :

Memeriksa apakah 'processed_files' tidak kosong.

processed_files = processed_files{1}; :

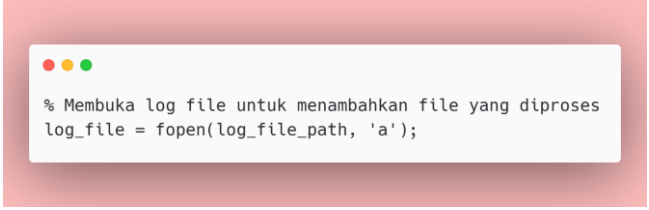
Mengambil cell array pertama dari hasil textscan.

else processed_files = {}; :

Mengatur processed_files menjadi array kosong jika file log kosong.

- Bagian 3 : Membuka Log File untuk Menambahkan File yang sudah Diproses

  Bagian ini merupakan lanjutan dari bagian 2.

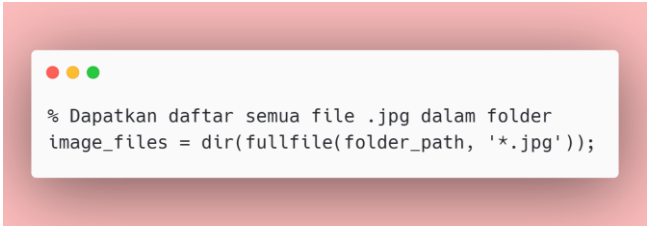


Gambar 4.9 Bagian 3 : Membuka Log File

log_file = fopen(log_file_path, 'a'); :

Membuka file log dalam mode append untuk menambahkan file yang baru diproses.

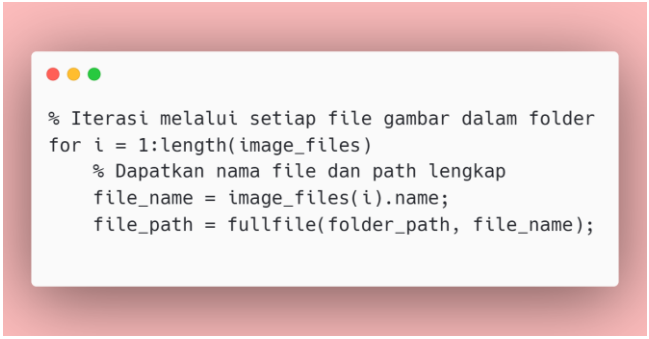
- Bagian 4 : Memproses Setiap File Gambar dalam Folder



Gambar 4.10 Bagian 4 : Membaca Directory

image_files = dir(fullfile(folder_path, '*.jpg')); :

Mendapatkan daftar semua file ‘.jpg’ dalam folder yang ditentukan.



Gambar 4.11 Iterasi Setiap File Gambar dalam Folder

for i = 1(image_files) :

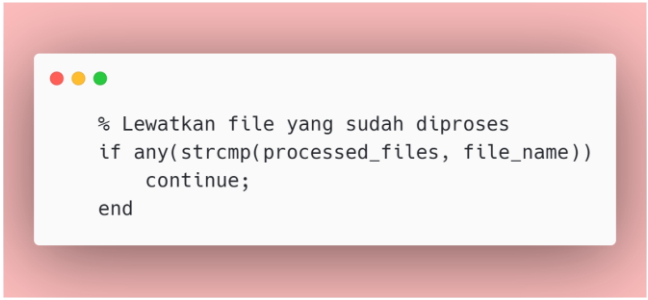
Iterasi melalui setiap file gambar dalam folder.

file_name = image_files(i).name; :

Mendapatkan nama file dari daftar file gambar.

file_path = fullfile(folder_path, file_name); :

Membuat path lengkap ke file gambar.



```
% Lewatkan file yang sudah diproses
if any(strcmp(processed_files, file_name))
    continue;
end
```

Gambar 4.12 Memeriksa File yang sudah Diproses

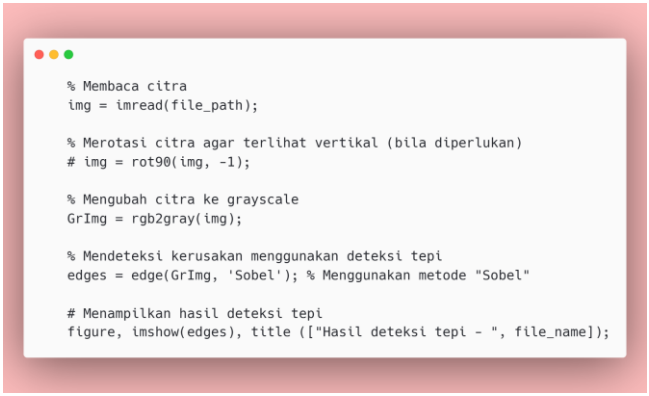
if any(strcmp(processed_files, file_name)) :

Memeriksa apakah file sudah diproses dengan membandingkan nama file dengan daftar file yang sudah diproses.

continue; :

Lewatkan file jika sudah diproses.

- Bagian 5 : Membaca dan Memproses Citra



```
% Membaca citra
img = imread(file_path);

% Merotasi citra agar terlihat vertikal (bila diperlukan)
# img = rot90(img, -1);

% Mengubah citra ke grayscale
GrImg = rgb2gray(img);

% Mendeteksi kerusakan menggunakan deteksi tepi
edges = edge(GrImg, 'Sobel'); % Menggunakan metode "Sobel"

# Menampilkan hasil deteksi tepi
figure, imshow(edges), title (["Hasil deteksi tepi - ", file_name]);
```

Gambar 4.13 Bagian 5 : Membaca dan Memproses Citra

img = imread(file_path); :

Membaca citra dari file.

 # img = rot90(img, -1); :

Mengomentari rotasi citra. Jika diperlukan, kode ini bisa digunakan untuk merotasi citra agar terlihat vertikal. Hilangkan lambang '#' bila memang diperlukan

GrImg = rgb2gray(img); :

Mengubah citra ke grayscale.

edges = edge(GrImg, 'Sobel'); :

Mendeteksi tepi citra menggunakan metode Sobel.

figure, imshow(edges), title (["Hasil deteksi tepi - ", file_name]); :

Menampilkan hasil deteksi tepi dengan judul yang mencakup nama file.

- Bagian 6 : Memanggil Fungsi 'ApplyMorfologi' untuk Menutup Bagian yang Tidak Ingin Terbaca

```
% Memanggil fungsi untuk mengisi background dengan warna hitam
masking_result = ApplyMorfologi(edges);

% Tampilkan hasil akhir setelah pengisian background
figure, imshow(masking_result);
title(['Hasil akhir setelah pemberian masking - ', file_name]);
```

Gambar 4.14 Bagian 6 : Memanggil Fungsi 'ApplyMorfologi'

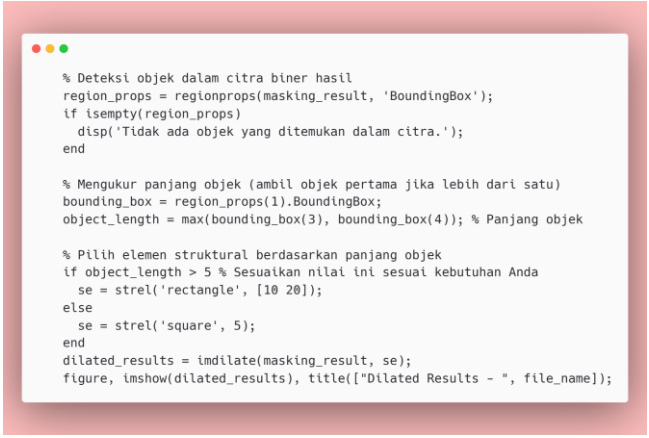
masking_result = ApplyMorfologi(edges); :

Memanggil fungsi ApplyMorfologi untuk memproses citra dan menutup bagian yang tidak dinginkan

figure, imshow(masking_result); title(['Hasil akhir setelah pemberian masking - ', file_name]); :

Menampilkan hasil akhir setelah pemberian masking dengan judul yang mencakup nama file.

- Bagian 7 : Deteksi dan Pengolahan Objek dalam Citra



```
% Deteksi objek dalam citra biner hasil
region_props = regionprops(masking_result, 'BoundingBox');
if isempty(region_props)
  disp('Tidak ada objek yang ditemukan dalam citra.');
end

% Mengukur panjang objek (ambil objek pertama jika lebih dari satu)
bounding_box = region_props(1).BoundingBox;
object_length = max(bounding_box(3), bounding_box(4)); % Panjang objek

% Pilih elemen struktural berdasarkan panjang objek
if object_length > 5 % Sesuaikan nilai ini sesuai kebutuhan Anda
  se = strel('rectangle', [10 20]);
else
  se = strel('square', 5);
end
dilated_results = imdilate(masking_result, se);
figure, imshow(dilated_results), title(["Dilated Results - ", file_name]);
```

Gambar 4.15 Bagian 7 : Deteksi dan Pengolahan Objek

region_props = regionprops(masking_result, 'BoundingBox');:

Menggunakan fungsi regionprops untuk mendeteksi objek dalam citra biner hasil dan mendapatkan properti BoundingBox dari setiap objek.

if isempty(region_props) :

Memeriksa apakah tidak ada objek yang terdeteksi.

disp('Tidak ada objek yang ditemukan dalam citra.');:

Menampilkan pesan bahwa tidak ada objek yang ditemukan.

bounding_box = region_props(1).BoundingBox; :

Mengambil bounding box dari objek pertama yang terdeteksi.

object_length = max(bounding_box(3), bounding_box(4));:

Mengukur panjang objek sebagai nilai maksimum antara lebar dan tinggi bounding box.

if object_length > 5 :

Memeriksa apakah panjang objek lebih dari 5 piksel.

se = strel('rectangle', [10 20]);:

Jika panjang objek lebih dari 5 piksel, elemen struktural berbentuk persegi panjang dengan ukuran 10x20 piksel dipilih.

Else :

Jika panjang objek tidak lebih dari 5 piksel.

se = strel('square', 5); :

Jika panjang objek tidak lebih dari 5 piksel, elemen struktural berbentuk persegi dengan ukuran 5x5 piksel dipilih.

dilated_results = imdilate(masking_result, se);:

Menerapkan operasi morfologi dilate (penggelembungan) pada citra hasil masking menggunakan elemen struktural yang dipilih.

figure, imshow(dilated_results), title(["Dilated Results - ", file_name]);

Menampilkan hasil operasi dilate dengan judul yang mencakup nama file

- Bagian 8 : Mengisi Lubang dalam Citra dan Menyimpan Hasil

```
# Apply filling
filled_results = imfill(dilated_results, 'holes');
figure, imshow(filled_results), title(["Filled Holes - ", file_name]);

% Simpan hasil ke dalam file
[~, name, ext] = fileparts(file_name);
output_file_path = fullfile(output_folder, [name, '_processed', ext]);
imwrite(filled_results, output_file_path);
disp(['Hasil disimpan ke ', output_file_path]);
```

Gambar 4.16 Bagian 8 : Mengisi Lubang dan Menyimpan

filled_results = imfill(dilated_results, 'holes'); :

Menggunakan fungsi imfill untuk mengisi lubang dalam citra hasil operasi dilate.

figure, imshow(filled_results), title(["Filled Holes - ", file_name]); :

Menampilkan hasil pengisian lubang dengan judul yang mencakup nama file.

[~, name, ext] = fileparts(file_name); :

Memisahkan nama file, ekstensi, dan path dari file_name.

output_file_path = fullfile(output_folder, [name, '_processed', ext]); :

Membuat path lengkap untuk menyimpan file hasil dengan nama yang sudah dimodifikasi.

imwrite(filled_results, output_file_path); :

Menyimpan citra hasil ke dalam file dengan path yang sudah ditentukan.

disp(['Hasil disimpan ke ', output_file_path]); :

Menampilkan pesan bahwa hasil telah disimpan ke path yang ditentukan.

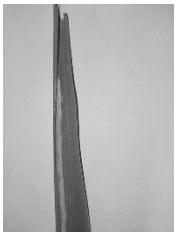
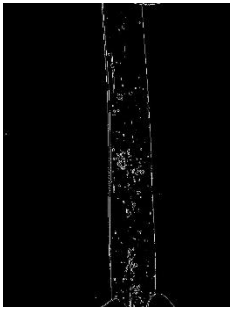
fclose(log_file); :

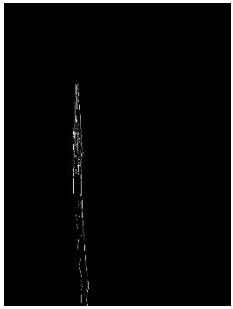
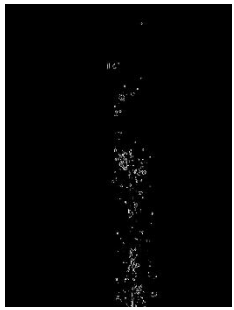
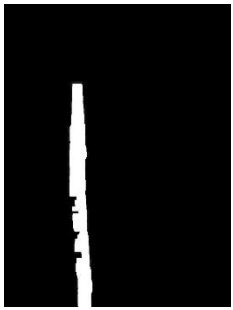
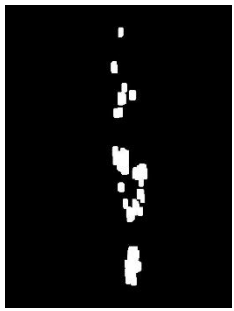
Menutup file log setelah semua file gambar dalam folder diproses.

Berdasarkan langkah-langkah diatas, adapun bila langkah-langkah tersebut ditabel untuk membandingkan

hasil dari setiap langkah maka akan menjadi sebagai berikut.

Tabel 4.6 Perbandingan Hasil pada Setiap Tahap

	Hawar	*Leaf Spot*
Citra Asli		
Citra *Grayscale*		
Deteksi Tepi 'Sobel'		

Operasi Morfologi 1		
Operasi Morfologi 2 (Hasil Akhir)		

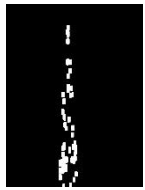
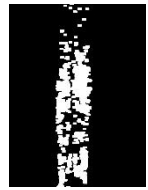
Setelah seluruh citra berhasil terbaca dan tersegmentasi oleh program maka langkah selanjutnya dapat dilakukan, yaitu ekstraksi fitur.

C. Ekstraksi Fitur

Berdasarkan hasil akhir dari Segmentasi Kerusakan Daun, dapat terlihat beberapa objek yang memiliki bentuk dan didefinisikan sebagai ‘objek putih’ dalam citra hasil akhir tersebut. Berdasarkan hasil akhir citra, bahwasannya

penyakit hawar memiliki 1 buah objek putih yang memanjang, dan penyakit *Leaf Smut* memiliki beberapa objek putih yang berbentuk bulat kecil dan jumlahnya banyak.

Tabel 4.7 Jumlah Objek yang Terdeteksi pada Contoh Citra Hasil Akhir

Hawar				
	1	1	2	1
Leaf Smut				
	12	9	22	9

Perbedaan jumlah bentuk yang dihasilkan oleh objek-objek tersebutlah yang akan diekstraksi untuk menentukan variabel yang akan digunakan pada tahap berikutnya, yaitu Identifikasi. Adapun program yang dijalankan untuk

mengekstraksi fitur jumlah objek putih tersebut adalah sebagai berikut.

```
pkg load image;

% Tentukan folder tempat file gambar hasil berada
output_folder = 'C:/Users/igiva/OneDrive/Documents/Skripsi/Experiment/1. Hawar/Dataset/Hasil';
% Ganti dengan path ke folder output

% Dapatkan daftar semua file .jpg dalam folder output
image_files = dir(fullfile(output_folder, '*.jpg'));

% Inisialisasi array untuk menyimpan hasil
results = cell(length(image_files) + 1, 3);
results(1, :) = {'File Name', 'Jumlah Objek', 'Klasifikasi'};

% Iterasi melalui setiap file gambar dalam folder output
for i = 1:length(image_files)
    % Dapatkan nama file dan path lengkap
    file_name = image_files(i).name;
    file_path = fullfile(output_folder, file_name);

    % Membaca citra
    img = imread(file_path);

    % Panggil fungsi untuk menghitung dan mengklasifikasikan objek
    [jumlah, klasifikasi] = ApplyCounting(img);

    % Simpan hasil ke dalam array
    results{i + 1, 1} = file_name;
    results{i + 1, 2} = jumlah;

    % Tampilkan hasil
    fprintf('Jumlah objek pada citra: %s', file_name, jumlah, klasifikasi);
end
```

Gambar 4.17 Program untuk Ekstraksi Fitur

Adapun penjelasan penggunaan *command* dan fungsi-fungsi pada program adalah sebagai berikut

pkg load image:

Memuat paket image di Octave yang menyediakan fungsi untuk pemrosesan citra.

output_folder:

Menentukan path ke folder tempat menyimpan hasil pemrosesan gambar. Gunakan *path* '/2. Leaf Smut/Dataset/Hasil' untuk folder yang berisi citra *Leaf Smut.*

image_files = dir(fullfile(output_folder, '*.jpg')); :

Mendapatkan daftar semua file .jpg dalam folder output.

results = cell(length(image_files) + 1, 3);:

Membuat array sel berukuran (jumlah file gambar + 1) x 3 untuk menyimpan hasil.

results(1, :) = {'File Name', 'Jumlah Objek', 'Klasifikasi'};:

Mengisi baris pertama array results dengan header: 'File Name', 'Jumlah Objek', dan 'Klasifikasi'.

for i = 1(image_files):

Iterasi melalui setiap file gambar dalam folder output.

file_name = image_files(i).name; :

Mendapatkan nama file dari daftar file gambar.

file_path = fullfile(output_folder, file_name); :

Membuat path lengkap ke file gambar.

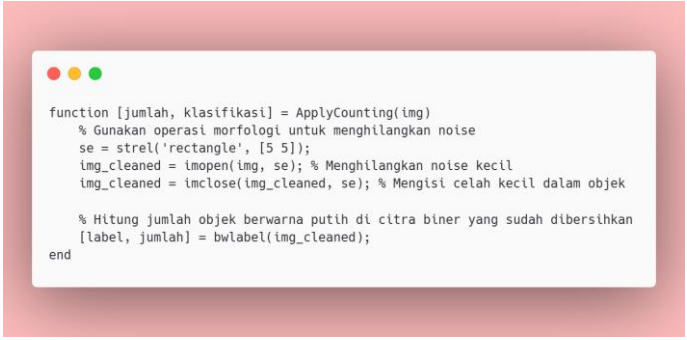
img = imread(file_path); :

Membaca citra dari file.

[jumlah, klasifikasi] = ApplyCounting(img); :

Memanggil fungsi ApplyCounting untuk menghitung jumlah objek dan mengklasifikasikannya dalam citra img.

Untuk fungsi ApplyCounting itu sendiri dapat dilihat pada gambar 4.18 dibawah ini.



```
function [jumlah, klasifikasi] = ApplyCounting(img)
    % Gunakan operasi morfologi untuk menghilangkan noise
    se = strel('rectangle', [5 5]);
    img_cleaned = imopen(img, se); % Menghilangkan noise kecil
    img_cleaned = imclose(img_cleaned, se); % Mengisi celah kecil dalam objek

    % Hitung jumlah objek berwarna putih di citra biner yang sudah dibersihkan
    [label, jumlah] = bwlabel(img_cleaned);
end
```

Gambar 4.18 Fungsi ApplyCounting

function [jumlah, klasifikasi] = ApplyCounting(img):
Mendefinisikan fungsi 'ApplyCounting' yang menerima input 'img' dan mengembalikan dua output: 'jumlah dan klasifikasi'.

se = strel('rectangle', [5 5]);:
Membuat elemen struktural berbentuk persegi panjang dengan ukuran 5x5 piksel yang akan digunakan untuk operasi morfologi.

img_cleaned = imopen(img, se); :

Menerapkan operasi morfologi ‘imopen’ pada citra ‘img’ menggunakan elemen struktural ‘se’. Operasi ‘imopen’ terdiri dari operasi erosi diikuti oleh dilasi, yang bertujuan untuk menghilangkan noise kecil.

img_cleaned = imclose(img_cleaned, se); :

Menerapkan operasi morfologi ‘imclose’ pada citra ‘img_cleaned’ menggunakan elemen struktural ‘se’. Operasi ‘imclose’ terdiri dari operasi dilasi diikuti oleh erosi, yang bertujuan untuk mengisi celah kecil dalam objek.

[label, jumlah] = bwlabel(img_cleaned); :

Menggunakan fungsi ‘bwlabel’ untuk memberi label pada objek terhubung di citra biner ‘img_cleaned’. Fungsi ini mengembalikan matriks label ‘label’ dan jumlah objek terhubung ‘jumlah’.

Adapun hasil keseluruhan perhitungan jumlah objek pada citra biner hasil akhir segmentasi adalah sebagai berikut.

Tabel 4.8 Hasil Perhitungan Jumlah Objek pada Citra Biner

Hawar		*Leaf Smut*	
Nama Citra	Jumlah Objek	Nama Citra	Jumlah Objek
Hawar1	1	LSmut1	12
Hawar2	1	LSmut2	9

Hawar3	3	LSmut3	22
Hawar4	1	LSmut4	9
Hawar5	3	LSmut5	5
Hawar6	2	LSmut6	2
Hawar7	1	LSmut7	31
Hawar8	1	LSmut8	12
Hawar9	2	LSmut9	9
Hawar10	1	LSmut10	15
Hawar11	1	LSmut11	6
Hawar12	2	LSmut12	1
Hawar13	2	LSmut13	8
Hawar14	2	LSmut14	6
Hawar15	2	LSmut15	4
Hawar16	2	LSmut16	3
Hawar17	1	LSmut17	2
Hawar18	1	LSmut18	10
Hawar19	1	LSmut19	2
Hawar20	1	LSmut20	3
Hawar21	2	LSmut21	4
Hawar22	2	LSmut22	1
		LSmut23	9
		LSmut24	5
		LSmut25	6
		LSmut26	5

	LSmut27	4
	LSmut28	7

Berdasarkan tabel diatas, diketahui bahwa penyakit *leaf smut* memiliki jumlah bentuk yang lebih banyak dibandingkan penyakit hawar. Setelah seluruh jumlah bentuk yang didefinisikan oleh objek putih sudah terhitung dan/atau terekstraksi maka langkah berikutnya ialah mengidentifikasi dan mengklasifikasi gejala penyakit berdasarkan fitur yang sudah terekstraksi.

D. Klasifikasi

Berdasarkan fitur yang telah berhasil terekstraksi, yaitu total jumlah bentuk yang didefinisikan oleh objek putih tersebut dapat disimpulkan bahwa jumlah bentuk yang dimiliki oleh penyakit *leaf smut* jauh lebih banyak dibanding dengan penyakit hawar. Dalam metode *Real-basec Classification,* hal ini dapat menjadi sebuah aturan dalam klasifikasi yang mana akan menjadi:

"*Jika jumlah objek melibihi jumlah objek milik kelas Hawar, maka termasuk kedalam kelas LSmut*"

Namun, sebelum yakin untuk menjadikan aturan tersebut sebagai aturan untuk mengklasifikasi seluruh data, hal yang perlu dilakukan sebelumnya adalah

memastikan apakah ada referensi yang menjelaskan karakteristik dari kedua penyakit tersebut.

Adapun sumber-sumber referensi yang memiliki penjelasan terkait karakteristik dari kedua penyakit tersebut adalah:

1) Buku Budidaya Tanaman Padi (2019) karya M. Siregar dan Sulardi.
2) Website 'rice-diseases.irri.org' yang ditulis oleh beberapa tokoh Agrikultur dan Pertanian dari seluruh dunia
3) Artikel jurnal yang berjudul '*Major Diseases of Rice and their management*' (2023) yang ditulis oleh Deepak Mourya dan Amir Kumar Mourya.
4) Buku Saku Penyakit Padi (2020) yang disusun oleh Cahyadi Irwan dan Sendy Sofyan Mu'min
5) Artikel jurnal yang berjudul '*Diagnosis of Bacterial Leaf Blight, Brown Spots, and Leaf Smut Rice Plant Diseases using Light GBM*' yang ditulis oleh G.R.I.L Jayasooriya dan Samantha Mathara Arachchi

Berdasarkan sumber-sumber referensi tersebut, tidak dijelaskan secara spesifik mengenai jumlah gejala penyakit yang nampak pada daun. Namun dari referensi-referensi diatas memiliki satu kesamaan dalam menjelaskan

karakteristik kedua penyakit tersebut, yaitu dampak dari penyakit hawar dapat terlihat dengan melihat sebuah bercak yang membentuk hawar daun yang berwarna kuning dan **memanjang** pada daun. Sedangkan gejala *leaf spot* ialah bintik-bintik yang yang **menyebar** ke seluruh daun.

Kata 'menyebar' dan 'memanjang' memang tidak bisa didefinisikan sebagai satuan jumlah, namun merupakan sebuah indikasi dari jumlah gejala penyakit tersebut. Dalam ilmu kalkulus, ketika sedang mempertimbangkan integral Riemann atau Lebesgue pada satu titik, maka hasil dari titik tersebut adalah nol. Hal ini dikarenakan panjang atau ukuran dari satu titik adalah nol. Yang mana hal ini menunjukkan bahwa meskipun mengintegrasikan (memanjang) nilai fungsi pada satu titik, titik tersebut tidak berubah dari esensi dasarnya sebagai satu titik tunggal (Royden & Fitzpatrick, 1968).

Secara logika dan matematika, penyebaran satu titik menjadi beberapa titik dapat dipahami sebagai transformasi atau proses yang mengubah satu entitas tunggal menjadi beberapa entitas yang terpisah. Dalam konteks ini, satu titik yang tersebar ke beberapa tempat memang tidak lagi dianggap sebagai satu titik saja,

melainkan beberapa titik yang terdistribusi (Guardiano, Srivastava, & al., 2023).

Berdasarkan kedua hal tersebut, didapatkan sebuah logika yang mana titik yang ‘menyebar‘ dapat dipastikan memiliki lebih dari 1, sedangkan titik yang ‘memanjang’ masih dapat diidentifikasi sebagai 1. Oleh karena itu, dengan menggunakan logika tersebut terdapat aturan baru, yaitu:

“*Jumlah objek hawar lebih sedikit dibanding jumlah objek LSmut*.”

Aturan tersebut didukung dengan jumlah hasil ektraksi fitur yang terbaca oleh program yang dijalankan, yang mana jumlah objek bentuk milik penyakit *leaf smut* jauh lebih banyak dibanding penyakit hawar, dengan kata lain untuk mengklasifikasikan data tersebut menjadi dua kelas dibutuhkan nilai N yang mana nilai N akan menjadi:

“Jika jumlah objek <= N maka kelas = Hawar

Jika jumlah objek > N maka kelas = LSmut”

Pada penelitian ini, nilai N yang akan digunakan ialah jumlah objek tertinggi yang dimiliki oleh kelas hawar, yaitu 3.

Kemudian, untuk memastikan bila kelas penyakit hawar benar-benar memiliki jumlah objek maksimal 3, hal yang

dilakukan ialah mencari sebuah dataset yang berasal dari internet (*kegel, mendeley*, dan lain sebagainya) dan mengambil beberapa data dari dataset tersebut untuk diolah. Data yang diambil merupakan dataset kelas hawar yang berjumlah 15 data, untuk membuktikkan apakah berdasarkan data yang sudah diklasifikasikan secara manual dan disebar dalam bentuk dataset tersebut, benar memiliki jumlah objek tertinggi sebesar 15.

Adapun hasil dari perhitungan jumlah objek setelah data yang berasal dari dataset tersebut diolah ialah sebagai berikut.

TABEL 4.9 Jumlah Objek yang Terdeteksi pada Citra Milik Dataset

Hawar	
Nama Citra	Jumlah Objek
Blight1	1
Blight2	1
Blight3	1
Blight4	1
Blight5	1
Blight6	1
Blight7	1
Blight8	1
Blight9	2

Blight10	2
Blight11	2
Blight12	3
Blight13	3
Blight14	3
Blight15	2

Berdasarkan jumlah objek yang sudah dihitung dari citra yang berasal dari dataset tersebut, terlihat bahwa jumlah objek terbanyak yang terdeteksi pada kelas hawar ialah 3 (Blight12, Blight13, dan Blight14). Oleh karena itu, nilai N yang akan digunakan untuk pengklasifikasian ini ialah 3.

Setelah nilai N ditentukan maka aturan baru dapat ditetapkan untuk mengidentifikasi dan mengklasifikasi seluruh data tersebut kedalam dua kelas, yaitu:

"Jika jumlah objek <= 3 maka kelas = hawar

Jika jumlah objek > 3 maka kelas = leaf smut"

Dengan aturan tersebut maka klasifikasi dan identifikasi dapat dilakuakan menggunakan fungsi:

```
if jumlah > 3
   klasifikasi = 'L.Smut';
elseif jumlah <= 3
   klasifikasi = 'Hawar';
```

```
end
```

Adapun hasil dari klasifikasi berdasarkan aturan yang dipakai tersebut adalah sebagai berikut

Tabel 4.10 Hasil Klasifikasi

Actual Class	Prediction	Actual Class	Prediction
Hawar1	Hawar	LSmut1	LSmut
Hawar2	Hawar	LSmut2	LSmut
Hawar3	Hawar	LSmut3	LSmut
Hawar4	Hawar	LSmut4	LSmut
Hawar5	Hawar	LSmut5	LSmut
Hawar6	Hawar	LSmut6	Hawar
Hawar7	Hawar	LSmut7	LSmut
Hawar8	Hawar	LSmut8	LSmut
Hawar9	Hawar	LSmut9	LSmut
Hawar10	Hawar	LSmut10	LSmut
Hawar11	Hawar	LSmut11	LSmut
Hawar12	Hawar	LSmut12	Hawar
Hawar13	Hawar	LSmut13	LSmut
Hawar14	Hawar	LSmut14	LSmut
Hawar15	Hawar	LSmut15	LSmut
Hawar16	Hawar	LSmut16	Hawar
Hawar17	Hawar	LSmut17	Hawar
Hawar18	Hawar	LSmut18	LSmut
Hawar19	Hawar	LSmut19	Hawar

Hawar20	Hawar	LSmut20	Hawar
Hawar21	Hawar	LSmut21	LSmut
Hawar22	Hawar	LSmut22	Hawar
		LSmut23	LSmut
		LSmut24	LSmut
		LSmut25	LSmut
		LSmut26	LSmut
		LSmut27	LSmut
		LSmut28	LSmut

Berdasarkan hasil klasifikasi dan identifikasi pada tabel 4.7 diatas maka didapatkan nilai hasil prediksi dari kedua kelas dengan membandingkan jumlah data *actual class* dengan data hasil prediksi dan dikali 100 untuk mendapatkan persentase hasil, seperti persamaan ini bawah ini.

$$Hasil\ prediksi\ hawar : \frac{22}{22}\ x\ 100 = 100\%$$

$$Hasil\ prediksi\ LSmut : \frac{21}{28}\ x\ 100 = 75\%$$

Berdasarkan hasil persamaan diatas, hasil prediksi untuk kelas hawar sebesar 100%, sedangkan hasil prediksi untuk kelas LSmut sebesar 75%.

E. Evaluasi

Metode yang akan digunakan untuk mengevaluasi hasil dari program yang dijalankan ini ialah metode *receiver*

operating characteristic (ROC), yang mana menggunakan perhitungan matrix kebingungan (*confusion matrix*).

Untuk membuat confusion matrix berdasarkan data klasifikasi dan identifikasi pada tahapan sebelumnya, penyakit hawar akan dianggap sebagai kelas positif (P) dan penyakit *leaf smut* dianggap sebagai kelas negatif (N). Sebelum memulai untuk menghitung nilai dari *confusion matrix* berdasarkan data diatas, nilai *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN) akan ditentukan. Adapun untuk menentukan nilai metriks dari *confusion matrix* tersebut adalah sebagai berikut.

- Informasi yang didapat :
  a) Total data penyakit hawar = 22
  b) Total data penyakit *leaf smut* = 28
  c) Semua data penyakit hawar yang berhasil teridentifikasi sebagai hawar (*True Positive*, TP) = 22
  d) Ada 7 data penyakit *leaf smut* yang teridentifikasi sebagai hawar (*False Positive*, FP) = 7
- Perhitungan lainnya :
  a) **False Negative (FN)** : Jumlah penyakit hawar yang seharusnya terklasifikasi sebagai penyakit hawar,

tetapi terklasifikasi sebagai penyakit *leaf smut*. Berdasarkan hasil klasifikasi pada tahap sebelumnya, berarti nilai FN = 0.

b) **True Negative (TN)** : Jumlah penyakit *leaf smut* yang berhasil teridentifikasi sebagai penyakit *leaf smut*. Berdasarkan hasil klasifikasi dan identifikasi, maka nilai TN adalah, TN = 28 – 7 = 21.

Berdasarkan data yang telah dikumpulkan diatas, maka diketahui :

TP = 22

TN = 21

FP = 7

FN = 0

Berdasarkan nilai metriks yang didapat, maka bentuk tabel *confusion matrix*-nya. Bentuk tabel *confusion matrix* yang digunakan adalah sebagai berikut.

Tabel 4.11 Format Tabel *Confusion Matrix*

	Actual Class I	Actual Class II
Prediction I	TP	FP
Prediction II	FN	TN

Berdasarkan format tabel *confusion matrix* tersebut, maka tabel yang didapat dengan menggunakan hasil data yang diperoleh ialah.

Tabel 4.12 Tabel *Confusion Matrix*

	Actual Positive (Hawar)	Actual Negative (*Leaf Smut*)
Predicted Positive (Hawar)	22	7
Predicted Negative (*Leaf Smut*)	0	21

Berdasarkan data tabel 4.8 diatas, dapat digunakan untuk menentukan nilai-nilai evaluasi metrik dari *confusion matrix*. Adapun metrik evaluasi dari *confusion matrix* menggunakan seluruh data yang diperoleh ialah sebagai berikut.

- Akurasi

$$\boldsymbol{Akurasi} = \frac{TP+TN}{Jumlah\ seluruh\ data} \quad (4.1)$$

$$= \frac{22+21}{50} = \frac{43}{50} = 0{,}86$$

- Presisi

$$\boldsymbol{Presisi} = \frac{TP}{TP+FP} \quad (4.2)$$

$$= \frac{22}{22+7} = \frac{22}{31} \approx 0{,}71$$

- Recall

$$\boldsymbol{Recall} = \frac{TP}{TP+FN} \tag{4.3}$$

$$= \frac{22}{22+0} = 1$$

- F1 Score

$$\boldsymbol{F1\ Score} = 2 \times \frac{(Presisi \times Recall)}{Presisi+Recall} \tag{4.4}$$

$$= 2 \times \frac{(0{,}71 \times 1)}{0{,}71+ 1}$$

$$= 2 \times \frac{0{,}71}{1{,}71} \approx 0{,}83$$

Dari hasil evaluasi metrik dari *confusion matrix* ini, dapat dilihat bahwa model ini memiliki performa yang sangat baik dalam mendeteksi kelas positif (penyakit hawar). *Precision* menunjukkan bahwa sebagian besar prediksi positif adalah benar, sedangkan *recall* menunjukkan bahwa model menangkap semua kasus positif yang sebenarnya. F1 Score memberikan gambaran keseimbangan yang baik antara precision dan recall.

BAB V
PENUTUP

A. Kesimpulan

Berdasarkan hasil dari penelitian yang telah dilakukan, adapun kesimpulan yang dapat diambil dari hasil-hasil setiap metode yang dilakukan ialah sebagai berikut:

1. Data citra padi yang berjumlah 50 data citra, dan didapat dari mengambil langsung pada area pesawahan di Kelurahan Karang Layung, Kec. Sukra, Indramayu, jawa Barat, mengacu kepada sumber dan referensi yang ada, data tersebut diduga memiliki 2 jenis penyakit, yaiu hawar daun (kresek) dan angan daun (*leaf smut*).
2. Dari total 50 data, 22 data diantaranya termasuk kedalam jenis penyakit hawar daun (kresek) dan 28 diantaranya termasuk kedalam penyakit angan daun (*leaf smut*).
3. Teknik Segmentasi yang digunakan dan berhasil membaca bagian kerusakan, ialah teknik deteksi tepi menggunakan operator Sobel.
4. Berdasarkan hasil dari ekstraksi fitur, jumlah objek yang tersegmentasi milik penyakit hawar jauh lebih sedikit dibanding dengan milik penyakit angan daun (*leaf smut*).

5. Metode yang digunakan untuk mengklasifikasi seluruh penyakit dan memprediksi apakah hasil klasifikasi manual benar-benar termasuk kedalam masing-masing penyakit ialah metode *rule-based classification*.
6. Berdasarkan pengukuran kinerja model, diperoleh nilai akurasi sebesar 0,86, presisi sebesar 0,71, *recall* sebesar 1, dan *F1 score* sebesar 0,83, yang menunjukkan bahwa model memiliki performa yang cukup baik, dengan recall yang sangat tinggi namun presisi yang lebih rendah, terutama dalam memprediksi kelas *Leaf Smut*.

B. Saran

Berdasarkan kesimpulan dari penelitian ini, terdapat beberapa saran yang dapat dipertimbangkan untuk penelitian ke depannya. Adapun saran-saran tersebut untuk penelitian kedepannya ialah:

1. Peningkatan teknik segmentasi mungkin diperlukan. Meskipun metode deteksi tepi Sobel efektif, eksplorasi teknik segmentasi lain seperti segmentasi berbasis warna, model *deep learning*, atau teknik *Watershed* dapat memberikan hasil yang lebih akurat dalam memisahkan kelas Hawar dan Leaf Smut.

2. Perlu ada peningkatan dalam aspek presisi model, terutama untuk kelas Leaf Smut yang saat ini hanya memiliki tingkat keberhasilan yang tak terlalu tinggi dibanding dengan kelas Hawar. Pendekatan yang bisa diambil termasuk menambah data pelatihan yang lebih bervariasi, menggunakan metode augmentasi data, atau menerapkan algoritma *machine learning* yang lebih kompleks seperti model *ensemble* untuk meningkatkan presisi, bila ingin menggunakan *machine learning*.
3. Mengingat karakteristik bentuk yang berbeda antara Hawar dan Leaf Smut (memanjang dan menyebar), lebih banyak fitur bentuk dapat dieksplorasi dan dimasukkan dalam proses klasifikasi. Fitur seperti eksentrisitas, kekompakan, atau aspek rasio dapat digunakan untuk membedakan kedua kelas dengan lebih jelas.
4. Berikutnya, program yang dijalankan masih mengandalkan beberapa proses secara manual, khususnya dalam *masking* area yang tidak dinginkan. Mungkin dengan mengubah metode segmentasi seperti yang telah disebutkan dapat mengurangi proses yang harus dijalankan secara manual, dan dapat

sepenuhnya dijalankan menggunakan model tanpa adanya bantuan manual.

5. Terakhir, kurangnya informasi dan referensi-referensi yang menyebutkan secara spesifik apa saja karakteristik jelas yang membedakan satu penyakit dengan penyakit lainya membuat hasil dari identifikasi masih dipertanyakan akan kebenarannya. Bekerja sama dengan seseorang yang ahli dalam bidang pertanian mungkin bisa mengurangi bahkan menghilangkan keraguan apakah hasil identifikasi dari model benar-benar termasuk kedalam penyakit yang bersangkutan.

DAFTAR PUSTAKA

Agustiani, S., Arifin, Y. T., Junaidi, A., Wildah, S. K., & Ali Mustopa. (2022). Klasifikasi Penyakit Daun Padi Menggunakan Random Forest dan Color Histogram. *Jurnal Komputasi Ilmu Komputer Unila*, *Vol. 10*(No. 1), 65–74.

Anwar, K., & Setyowibowo, S. (2021). Segmentasi Kerusakan Daun Padi pada Citra Digital Berdasarkan Hue, Saturation, Value, dan Grayscale. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, *Vol. 7*(No. 1), 39–43.

Ballard, D. H. (1981). Generalizing The Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, *Vol. 13*(No. 2), 111–118.

Bianome, R. M., Sina, D. R., & Nabuasa, Y. Y. (2020). Diagnosa Hama dan Penyakit pada Tanaman Padi Menggunakan Metode Naive Bayes dan K-Nearest Neighbor. *Jurnal Komputer & Informatika (J-ICON)*, *Vol. 8*(No. 2), 156–158.

Dijaya, R. (2017). *Buku Ajar Pengolahan Citra Digital* (M. Tanzil & M. D.KW, Eds.; Cetakan Pertama). UMSIDA Press.

Fawcett, T. (2005). An Introduction to ROC Analysis. *Pattern Recognition Letters*, *Vol. 27*, 861–863.

Gong, S., Liu, C., Zhong, Y. J. B., & Dong, Y. L. H. (2019). *Advanced Image and Video Processing Using MATLAB* (Vol. 12). Spinger International Publishing AG.

Gonzales, R. C., Woods, R. E., & Eddins, S. L. (2009). *Digital Image Processing Using MATLAB* (2nd ed.). Gatesmark Publishing.

Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing* (3rd ed.). Pearson Education.

Hermana, A. N., Zulkarnain, A., & Riadi, Y. A. (2018). Implementasi Pengolahan Model Warna RGB pada Aplikasi Identifikasi Warna. *MIND Journal*, *Vol. 3*(No. 1), 38–40.

Hong, Q. H. (1991). *3D Feature Extraction from a Single 2D Image*. University of Reading.

Iriyanto, S. Y., & Zaini, T. M. (2014). *Pengolahan Citra Digital* (IKAPI, Ed.). Anugra Utama Raharja (AURA).

Lu, S. (2020). Analysis of Digital Image Representation in Public Art Creation in the New Period. *Journal of Physics: Conference Series*, 2–3.

Martinez-Murica, F. J., Ramirez, J., & Gorriz, J. M. (2017). Feature Extraction. In *Wiley Encyclopedia of Electrical and Electronics Engineering* (pp. 3–9). University of Granada.

Nahm, F. S. (2021). Receiver Operating Characteristic Curve: Overview and Practical Use for Clinicians. *Korean Journal of Anesthesiology*, 2–4.

Nixon, M. S., & Aguado, A. S. (2002). *Feature Extraction & Image Processing for Computer Vision* (4th ed.). Replika Press.

Salau, A. O., & Jain, S. (2019). Feature Extraction: A Survey of the Types, Techniques, Applications. *IEEE International Conference on Signal Processing and Communication*, 158–162.

Sinha, P. K. (2012). *Image Acquisition and Preprocessing for Machine Vision Systems*. SPIE.

Siregar, M., & Sulardi. (2019). *Budidaya Tanaman Padi* (1st ed.). Fakultas Ekonomi Pembangunan Panca Budi.

Thanki, R. M., & Kothari, A. M. (2019). *Digital Image Processing using SCILAB*. Spinger International Publishing AG.

Ulinuha, M. A. (2022). *Segmentasi Tulang Wajah Tiga Dimensi Berbasis Sudut SImpangan* [Disertasi]. Institut Teknologi Sepuluh Novermber.

Vincent, L. (1993). Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. *IEEE Transactions on Image Processing*, *Vol. 2*(No. 2), 176–178.

Walascha, A., Febriana, A., Saputri, D., Haryanti, D. S. N., Tsania, R., Sanjaya, Y., & Priyanti. (2021). Review Artikel: Inventarisasi Jenis Penyakit yang Menyerang Daun Tanaman Padi (Oryza sativa L.). *Inovasi Riset Biologi Dalam Pendidikan Dan Pengembangan Sumber Daya Lokal*, 472–474.

Yang, S., & Berdine, G. (2017). The Receiver Operating Characteristic (ROC) Curve. *The Southwest Repiratory and Critical Care Chronicles*, *Vol. 5*(No. 19), 34–36.

Aggarwal, C. C. (20114). *Data Classification: Algorithms and Applications.* Florida: CRC Press.

Canny, J. (1986). *A Computational Approach to Edge Detection.* California: IEE Transactions on Pattern Analysis and Machine Intelligence.

Cohen, W., Furnkranz, J., Wilmer, G., & al., e. (1998). Rule-Based Classification on Machine Learning. *International Conference on Machine Learning.* IEEE

Gonzales, R. C., & Woods, R. E. (2018). *Digital Image Processing.* Upper Saddle River: Prentince Hall.

Guardiano, F., Srivastava, M., & al., e. (2023). *Multiple Point Statistics: A Review.* Berlin: Springer.

Irwan, C., & Mu'min, S. S. (2020). *Buku Saku Penyakit Padi.* Karawang: Balai Besar Peramalan Organisme Pengganggu Tumbuhan.

Jain, A. K. (1989). *Fundamentals of Digital image Processing.* Prentice Hall.

Liu, L., & Özsu, M. T. (2009). *Encyclopedia of Database Systems.* Chicago: Springer.

Marr, D., & Hildreth, E. (1980). *Theory of Edge Detection.* London: London Biological Sciences.

Mew, T. (2024, June 23). *Rice Diseases Online Resource - Contents*. Retrieved from IRRI Rice Diseases Online Resource: https://rice-diseases.irri.org/contents

Nagar, S. (2017). *Introduction to Octave: For Engineers and Scientists.* Berkeley: Apress Berkeley, CA.

Pajankar, A., & Chandu, S. (2020). *GNU Octave by Example: A Fast and Practical Approach to Learning GNU Octave.* Berkeley: Apress Berkeley, CA.

Pratt, W. K. (2001). *Digital Image Processing: PIKS Scientific Inside.* Wiley.

Rogel-Salazar, J. (2014). *Essential MATLAB and Octave.* Oxfordshire: Routledge.

Royden, H. L., & Fitzpatrick, P. M. (1968). *Real Analysis.* New York: Macmillan Publishing Co., Inc.

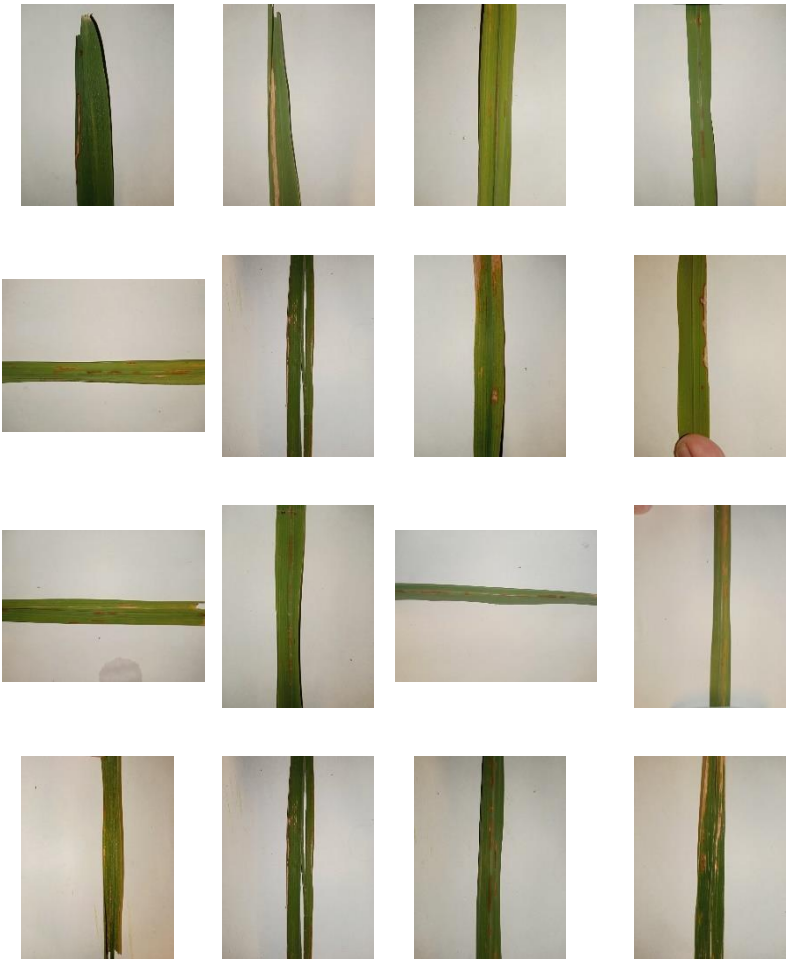
Serra, J. (2012). *Mathematical Morphology and Its Applications to Image and Signal Processing.* Chicago: Springer. dan N-Gram Stemming Untuk Mengenali Stemmer Bahasa Bali. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(2), 219.

Suryani, P. S., Linawati, L., & Saputra, K. O. (2019). Penggunaan Metode Naive Bayes Classifier Pada Analisis Sentimen Facebook Berbahasa Indonesia. *Majalah Ilmiah Teknologi Elektro*, 18(1), 145.

Liu, L., & Özsu, M. T. (2009). *Encyclopedia of Database Systems.* Chicago: Springer.

LAMPIRAN

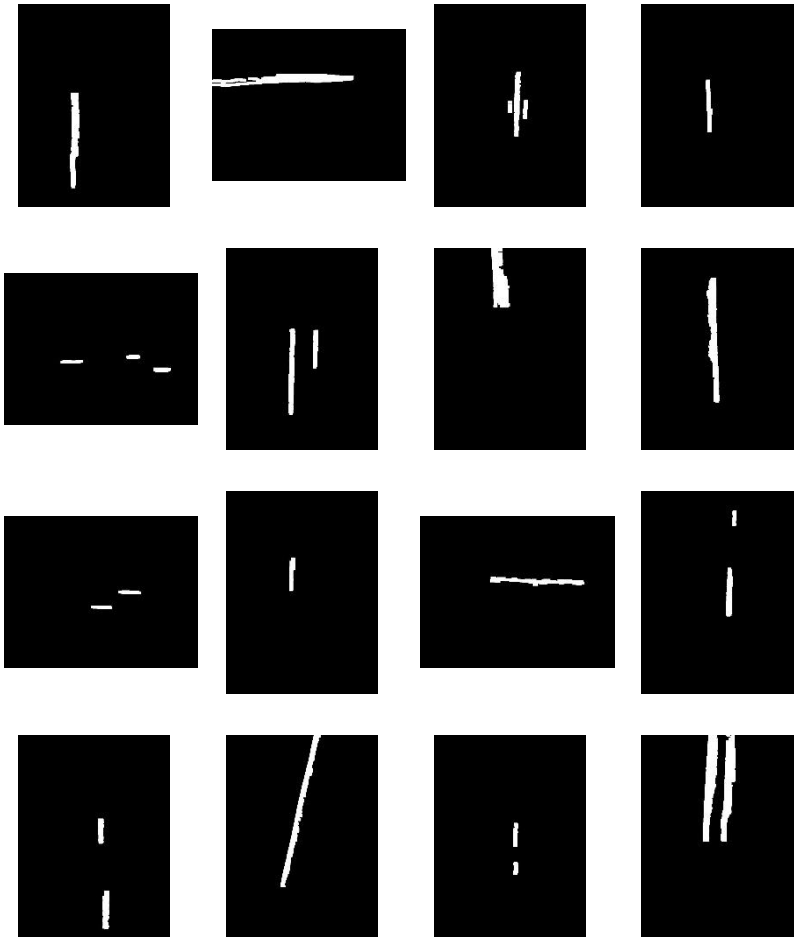
Lampiran 1. Citra Asli Hawar

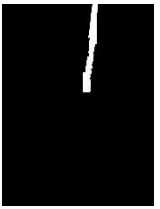
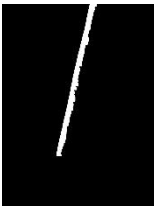
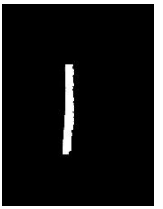


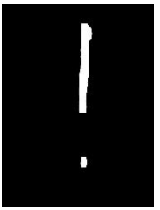
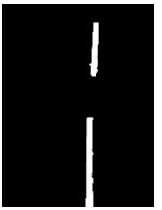




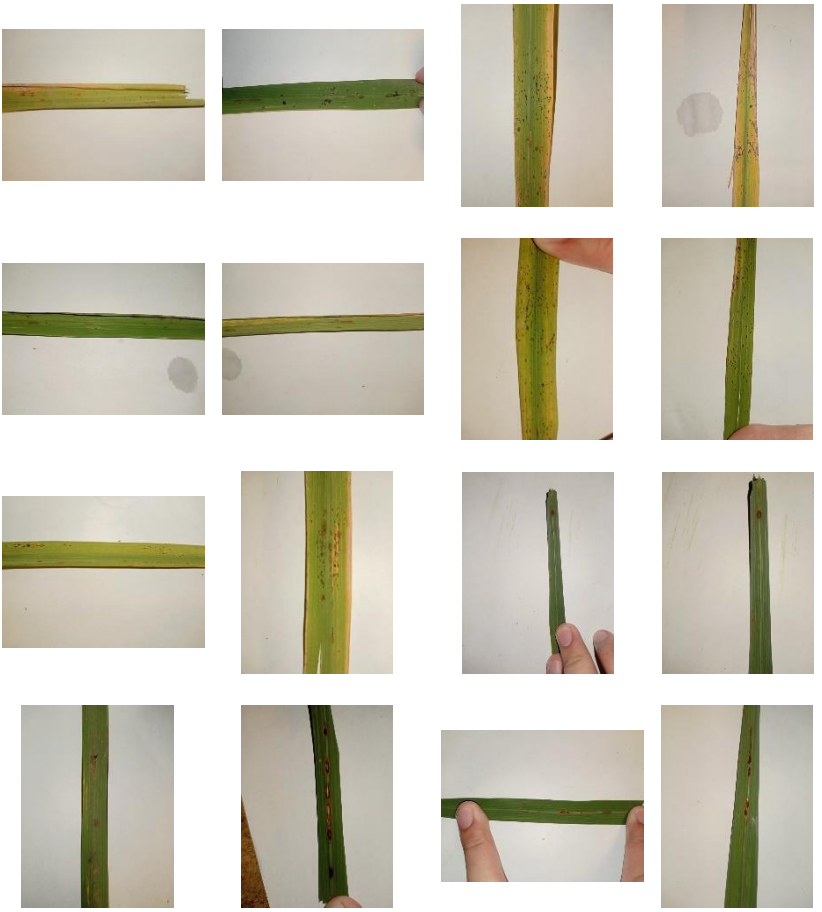

Lampiran 2. Citra Hasil Akhir Hawar



Lampiran 3. Citra Asli *Leaf Smut*





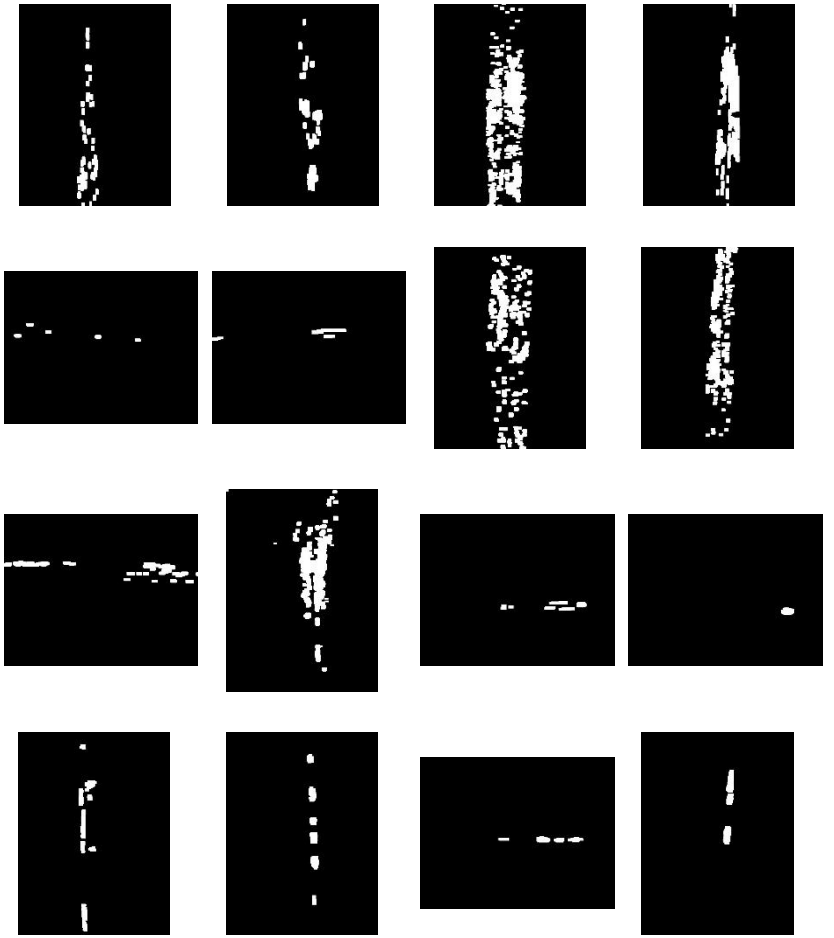


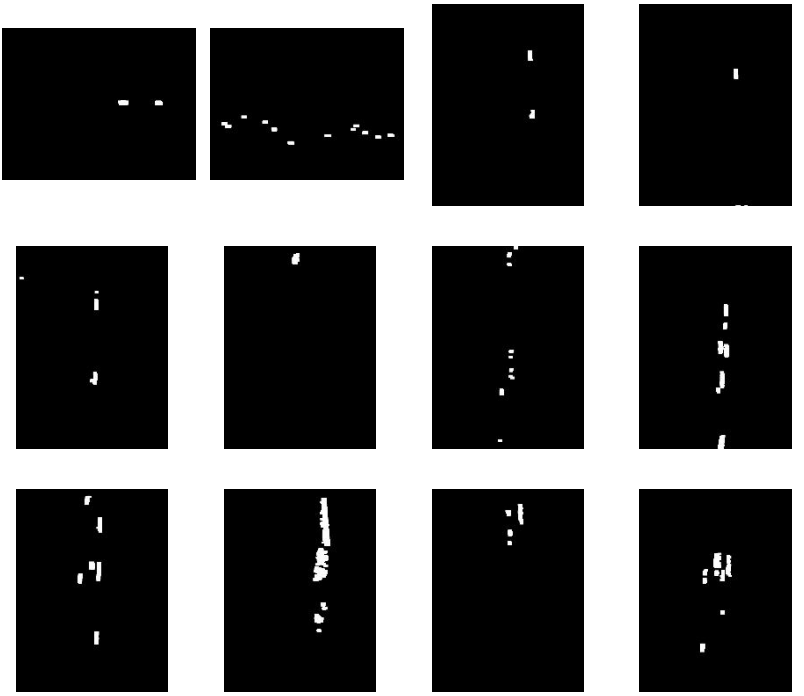




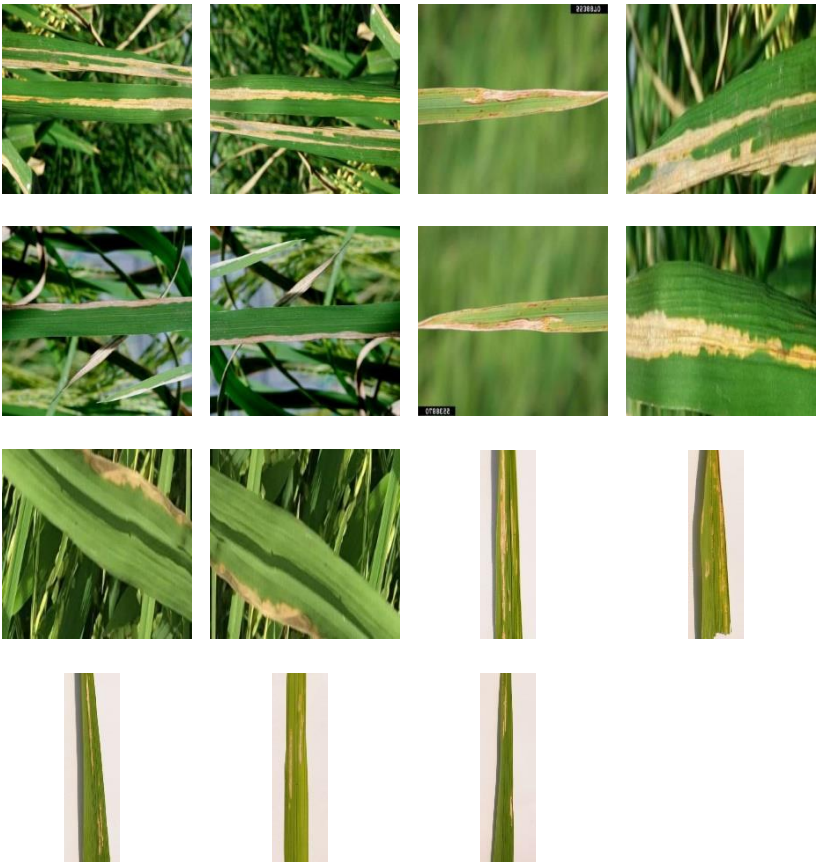



Lampiran 4. Citra Akhir Hasil *Leaf Smut*

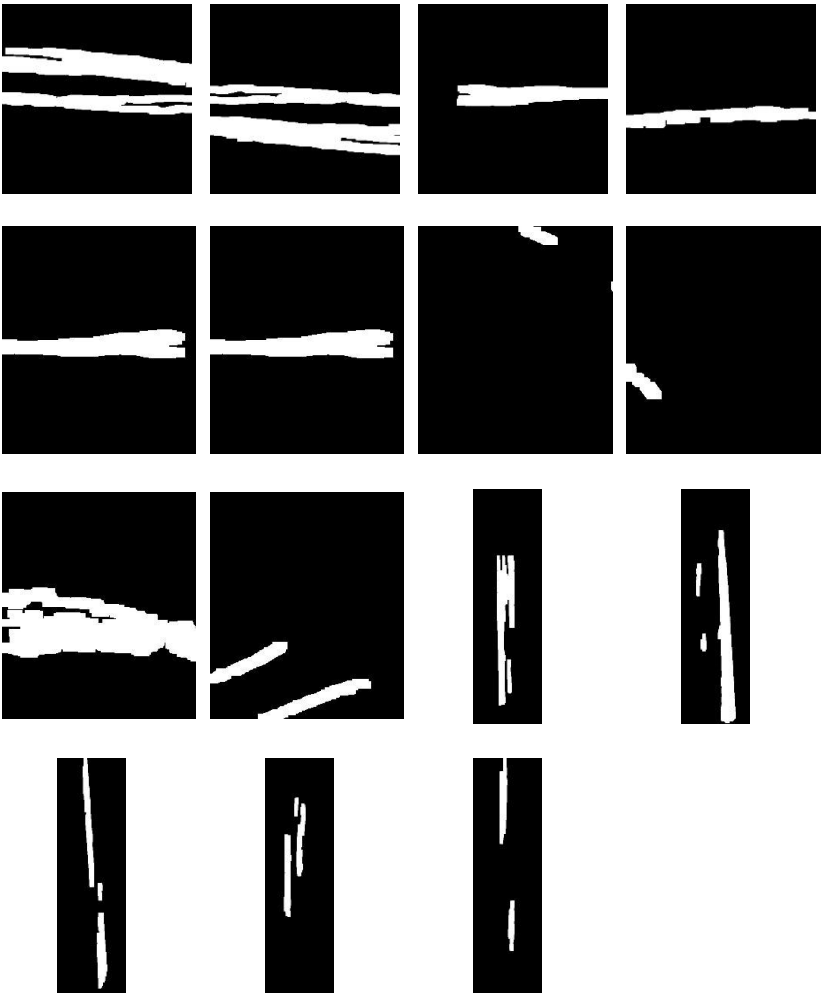




Lampiran 5. Citra Dataset Hawar



Lampiran 6. Citra Akhir Dataset Hawar



Lampiran 8. Riwayat Hidup

RIWAYAT HIDUP

A. Identitas Diri

1.	Nama	:	Iqbal Givari
2.	NIM	:	2008096055
3.	Alamat Rumah	:	Jl. Raya Pantura Kel. Karang Layung, RT 005 RW 001 Kec. Sukra, Kab Indramayu, Jawa Barat
4.	HP	:	081280803453
5.	Email	:	i.givari32@gmail.com

B. Riwayat Pendidikan

1. TK Islam Al-Aqobah
2. SD Negeri Sumuradem III
3. Madrasah Diniyah Asy-Syanawiyah
4. Pon-Pes Darussalam Eretan Kulon
5. SMA Negeri 4 Cirebon

Semarang, 24 Juni 2024

Iqbal Givari

NIM : 2008096055