

BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Masa Panen Tebu

Tebu merupakan tanaman yang mampu menghasilkan gula dan bioenergi, dengan masa tanam sekitar 12 bulan atau lebih. Pertumbuhan tebu berdasarkan usia tanam dapat dipantau menggunakan teknologi penginderaan jauh (Cahyono et al., 2022). Tanaman tebu merupakan tanaman perkebunan semusim yang hanya dipanen sekali, dan diklasifikasikan sebagai aset biologis menurut standar akuntansi. Aset biologis mengalami transformasi seperti pertumbuhan, degenerasi, produksi, dan prokreasi, yang menyebabkan perubahan kualitatif maupun kuantitatif pada tumbuhan tersebut (Nurlita et al., 2024).

Menurut (Wahyuni et al., 2022) produktivitas tebu dipengaruhi oleh beberapa faktor seperti morfologi lahan, varietas, hama atau penyakit, dan umur kematangan tebu. Umur kematangan terjadi saat tanaman memasuki bulan ke-9 pada fase pembuahan, dengan pertumbuhan tebu yang umumnya melalui empat fase, yaitu fase perkecambahan (3–45 hari), fase pertunasan (46 hari–4 bulan), fase pemanjangan batang (4–9 bulan), dan fase pembuahan (lebih dari 9 bulan). Menurut (Puspitasari et al., 2025) di Indonesia banyak yang mendukung kematangan tebu adalah faktor suhu dan kelembapan. Beberapa pabrik gula bahkan mengharuskan pemanenan tanaman yang belum masak untuk memenuhi kapasitas giling, sehingga

sebagian tanaman terpaksa dipanen saat persentase sukrosa masih kurang dari ideal, khususnya tebu yang dipanen pada awal musim giling.

Berdasarkan berbagai literatur, tanaman tebu merupakan komoditas penting yang menghasilkan gula dan bioenergi, dengan siklus tanam sekitar 12 bulan. Produktivitas tebu dipengaruhi oleh berbagai faktor seperti tipologi lahan, varietas, serangan hama/penyakit, serta umur kemasakan. Secara alami, proses kemasakan tebu memerlukan kondisi suhu dan kelembapan yang ideal, namun di Indonesia, seringkali tebu dipanen sebelum mencapai tingkat kemasakan optimal untuk memenuhi kebutuhan kapasitas giling pabrik. Pemantauan pertumbuhan dan kematangan tebu selama masa tanam dapat dilakukan menggunakan teknologi penginderaan jauh. Oleh karena itu, dalam rangka meningkatkan akurasi waktu panen dan kualitas hasil panen, diperlukan sistem diagnosa berbasis pengolahan citra, yang memanfaatkan karakteristik visual, seperti kondisi daun, untuk menentukan tingkat kemasakan tebu secara lebih objektif dan efisien.

2. Komoditas Tebu

Menurut (Heryanto et al., 2024), tebu (*Saccharum officinarum* Linn) merupakan bahan baku utama industri gula dan salah satu komoditas bahan pokok konsumsi masyarakat Indonesia, yang kebutuhannya terus meningkat seiring pertumbuhan penduduk. Secara geografis, tanaman tebu di Indonesia terkonsentrasi hanya di beberapa wilayah, yaitu Pulau Jawa, sebagian kecil Sumatera, dan Sulawesi. Menurut (Nuwa et al., 2022), tebu merupakan salah satu komoditas utama di sektor perkebunan yang berperan

penting dalam meningkatkan perekonomian serta kesejahteraan petani. Selain mendukung kegiatan usahatani, tebu juga menjadi bahan baku utama dalam menjaga keberlanjutan industri pengolahan gula pasir, yang merupakan salah satu kebutuhan pokok masyarakat.

Komoditas tebu dipandang sebagai agroindustri yang memiliki potensi besar dalam memberikan kontribusi terhadap pembangunan ekonomi, khususnya di negara-negara penghasil tebu. Industri gula yang menggunakan tebu sebagai bahan baku juga dianggap sebagai sumber pendapatan bagi petani tebu, pengusaha, dan pekerja di sektor industri gula (Novianti et al., 2021).

Berdasarkan kutipan-kutipan di atas, dapat disimpulkan bahwa komoditas tebu memiliki peran penting dalam perekonomian Indonesia, baik sebagai bahan baku utama industri gula maupun sebagai komoditas bahan pokok yang terus meningkat kebutuhannya seiring pertumbuhan penduduk. Tebu tidak hanya mendukung kegiatan usahatani, tetapi juga menjadi sumber pendapatan utama bagi petani, pengusaha, dan pekerja di sektor industri gula. Oleh karena itu, tebu dipandang sebagai agroindustri yang berpotensi besar dalam berkontribusi pada pembangunan ekonomi, baik secara lokal maupun nasional. Keberlanjutan industri gula yang mengandalkan tebu sebagai bahan baku menjadi kunci dalam mendukung kestabilan ekonomi dan kesejahteraan petani di Indonesia. Selain itu, pengembangan teknologi pertanian modern dan peningkatan produktivitas tanaman tebu menjadi faktor strategis agar bisa bersaing secara global.

3. Pengolahan Citra

Pengolahan citra adalah metode untuk mengubah citra menjadi bentuk digital dan melakukan beberapa operasi padanya, untuk mendapatkan citra yang lebih baik atau untuk mengekstrak beberapa informasi yang berguna darinya. Pengolahan citra merupakan jenis pengolahan data yang pengambilan data masukannya berupa gambar, seperti frame video atau foto dan output bisa berupa gambar atau ciri yang terkait dengan gambar tersebut, (Yanu et al., 2022). Menurut (Laananila et al., 2023), tujuan utama pengolahan citra adalah agar citra lebih mudah diidentifikasi baik oleh manusia maupun mesin. Dalam prosesnya, sebuah citra ditransformasi menjadi citra lain untuk mencapai tujuan tersebut.

Menurut (Dijaya, 2023), pengolahan citra dibagi menjadi beberapa bagian atau segmen, dengan tujuan untuk mendefinisikan dan mengelompokkan objek atau area yang menarik dalam citra. Segmentasi ini digunakan dalam berbagai aplikasi, seperti pengenalan objek, pengukuran area, pemrosesan citra medis, dan analisis citra satelit.

Berdasarkan uraian yang telah disampaikan, pengolahan citra digital bertujuan untuk mengubah citra menjadi bentuk yang lebih mudah diinterpretasikan oleh manusia maupun mesin. Proses ini melibatkan transformasi citra untuk meningkatkan kualitas atau mengekstrak informasi penting. Salah satu teknik penting dalam pengolahan citra adalah segmentasi, yaitu membagi citra menjadi beberapa bagian untuk mengidentifikasi dan memisahkan objek yang relevan.

3.1 Metode *K-Nearest Neighbor (KNN)*

Pemodelan K-Nearest Neighbors (KNN) adalah teknik *machine learning* yang digunakan untuk membuat model klasifikasi maupun regresi. Pada proses ini, data pelatihan digunakan untuk menyimpan informasi yang akan digunakan kembali saat fase evaluasi, khususnya dalam modul evaluasi model, (Praneswara, 2023). *KNN* merupakan metode klasifikasi yang bekerja dengan mengidentifikasi objek-objek yang memiliki jarak paling dekat berdasarkan parameter k , yaitu jumlah tetangga terdekat yang diperhitungkan. Nilai k optimal diperoleh selama proses pelatihan melalui uji coba, dan perhitungan jarak dilakukan menggunakan rumus *Euclidean Distance*, (V. Putra et al., 2023). Metode ini dapat dimanfaatkan untuk menganalisis data berbasis perkiraan data, namun lebih sering diterapkan dalam klasifikasi karena keunggulannya dalam kecepatan *running time* saat digunakan dalam teknik *data mining*, (Adi & Wintarti, 2022).

Dapat disimpulkan bahwa *KNN* adalah metode klasifikasi sederhana namun efektif yang mengandalkan kedekatan data dalam ruang fitur, dan karena tidak memerlukan proses pelatihan kompleks, metode ini menjadi pilihan tepat untuk aplikasi klasifikasi cepat dan ringan di bidang *data mining* maupun *machine learning*.

4. *Website*

Website adalah halaman informasi yang disediakan melalui jalur internet sehingga dapat diakses dari mana saja selama terdapat koneksi internet. *Website* merupakan sekelompok halaman yang memuat informasi-

informasi dalam bentuk teks, gambar, video, dan lain-lain yang dapat diakses kapanpun dan di manapun dengan bantuan teknologi internet (Rahmi et al., 2023). Pengembangan sistem informasi atau proses pengembangan sistem (*system development*) didefinisikan sebagai aktivitas untuk menghasilkan sistem informasi berbasis komputer dalam rangka menyelesaikan persoalan organisasi atau memanfaatkan peluang yang ada. Sistem informasi berbasis *website* memiliki banyak manfaat dan dapat mempermudah kerja manusia, sehingga pengembangan sistem informasi berbasis web masih terus dilakukan dengan menggunakan model atau metode tertentu sesuai kebutuhan masing-masing pengembang. *Website* juga dapat diartikan sebagai halaman - halaman yang menyajikan berbagai macam informasi seperti teks, data, gambar, animasi, suara, video, atau gabungan semuanya, baik statis maupun dinamis, yang saling terhubung melalui jaringan hyperlink, (Lestari & Jaya, 2021).

Menurut (Syahputra et al., 2021), terdapat tiga jenis *website* yang dikembangkan, yaitu *website statis*, *website dinamis*, dan *website interaktif*. Berikut penjelasan website website tersebut:

- a) *Website statis* merupakan jenis *website* yang memiliki tampilan tetap dan tidak mengalami perubahan konten yang berarti, dengan perubahan yang biasanya hanya terjadi pada desain tampilan halaman. Contoh dari website statis adalah situs yang menampilkan profil suatu lembaga atau organisasi.

- b) *Website dinamis* adalah *website* yang mengalami pembaruan konten secara berkala sesuai dengan perkembangan kebutuhan informasi dan zaman. Website ini bersifat lebih interaktif dan umumnya dilengkapi fitur seperti kolom komentar serta layanan pesan (chatting). Contohnya dapat dilihat pada halaman pengumuman yang diperbarui secara rutin.
- c) *Website interaktif* adalah jenis *website* yang dirancang untuk memungkinkan interaksi antar pengguna. Website ini sering terintegrasi dengan platform media sosial seperti Facebook, Twitter, dan Instagram, serta menyediakan fasilitas komunikasi dua arah, misalnya dalam layanan pendaftaran anggota baru melalui tautan yang disediakan dalam situs utama.

4.1. *Framework*

Framework adalah sebuah kerangka kerja untuk membangun software yang bisa digunakan oleh *developer* untuk mengembangkan sebuah aplikasi berbasis *web*, *mobile*, maupun desktop. Menurut (Muthia et al., 2023) . Dengan adanya *framework*, membuat aktivitas penulisan program lebih efisien atau kita tidak perlu membuat fungsi dasar dari aplikasi *website* sehingga mempermudah dalam pembuatan *website*, (Nabila & Amnur, 2021).

Berdasarkan kutipan-kutipan tersebut, dapat disimpulkan bahwa *framework* merupakan sebuah kerangka kerja perangkat lunak yang dirancang untuk memudahkan pengembangan aplikasi dengan menyediakan struktur dasar dan fungsi-fungsi bawaan. Dengan adanya

framework, *developer* tidak perlu membangun semuanya dari nol, sehingga proses pembuatan aplikasi, khususnya berbasis web, menjadi lebih efisien dan terstruktur. Selain itu, penggunaan *framework* juga dapat meningkatkan konsistensi kode, mempercepat waktu pengembangan, serta mempermudah proses pemeliharaan aplikasi di masa depan.

4.2. *MySQL*

Menurut (Silalahi, 2022), *MySQL* adalah *DBMS* atau *database management system* yang menggunakan bahasa *SQL* sebagai bahasa penghubung antara perangkat lunak aplikasi dengan database pada sistem. *MySQL* merupakan salah satu *DBMS* yang sangat populer digunakan. (Arimbi et al., 2022), menambahkan bahwa *MySQL* memiliki dua jenis lisensi, yaitu *Free Software*, yang memungkinkan siapa saja untuk menggunakannya secara bebas, dan *Shareware*, yaitu perangkat lunak berlisensi dengan pembatasan tertentu dalam penggunaannya.

Berdasarkan dua kutipan tersebut, dapat disimpulkan bahwa *MySQL* merupakan *database management system* (*DBMS*) yang populer dan menggunakan bahasa *SQL* sebagai jembatan antara aplikasi dan basis data. Selain itu, *MySQL* memiliki dua jenis lisensi, yaitu *Free Software* yang bebas digunakan oleh siapa saja, dan *Shareware* yang penggunaannya dibatasi oleh hak kepemilikan.

4.3. *Python*

Menurut (Nazar, 2024), *Python* merupakan salah satu bahasa pemrograman yang populer digunakan oleh perusahaan besar maupun para *developer* untuk mengembangkan berbagai macam aplikasi berbasis *desktop*, *web*, dan *mobile*. Selain itu, *Python* juga banyak dipelajari di kalangan mahasiswa dan perguruan tinggi, terutama yang berbasis teknologi informasi. Sementara itu, (Alfarizi et al., 2023) menambahkan bahwa *Python* menjadi bahasa pemrograman yang unggul dalam bidang *Machine Learning* dan *Deep Learning*, karena memiliki sejumlah kelebihan seperti kemudahan penulisan sintaksis, dukungan komunitas yang besar, serta ketersediaan *library* yang lengkap dan populer.

Berdasarkan kedua kutipan tersebut, dapat disimpulkan bahwa *Python* adalah bahasa pemrograman yang populer dan banyak digunakan dalam pengembangan aplikasi, baik di kalangan perusahaan besar, *developer*, maupun mahasiswa. Keunggulannya dalam *Machine Learning* dan *Deep Learning* didorong oleh kemudahan sintaksis, dukungan komunitas besar, dan *library* yang lengkap.

4.4. *Streamlit*

Streamlit merupakan framework berbasis *Python* yang bersifat open-source. *Framework* ini dirancang untuk memudahkan *developer* dalam membangun program berbasis web yang interaktif di bidang *data science* dan *machine learning* (Putranto et al., 2023). Salah satu

keunggulan dari *Streamlit* adalah kemampuannya untuk membangun aplikasi web tanpa perlu pemahaman mendalam mengenai teknologi *web development* (Jauhari et al., 2024). Selain itu, modul *streamlit* juga merupakan pustaka *Python open-source* yang mendukung pembuatan dan pembagian aplikasi web kustom, khususnya untuk keperluan *machine learning* dan analisis data. Proses penggunaan dimulai dari normalisasi data, dilanjutkan dengan pengolahan dan visualisasi data dalam bentuk grafik (Pradika et al., 2023).

Ketiga kutipan tersebut menggambarkan bahwa *Streamlit* adalah sebuah *framework* yang sangat bermanfaat dalam pengembangan aplikasi web interaktif, khususnya di bidang *data science* dan *machine learning*. Dengan sifatnya yang *open-source* dan mudah digunakan, *Streamlit* memungkinkan pengguna untuk membangun aplikasi tanpa harus memiliki keahlian mendalam dalam pengembangan web, sekaligus memfasilitasi proses visualisasi data secara efisien.


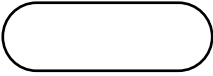


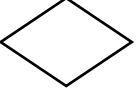

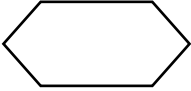


5. *Flowchart*

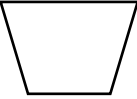


(G. T. R. Saputra, 2023), mendefinisikan diagram alur sebagai penggambaran berbasis grafis yang isinya berupa simbol-simbol tertentu yang menggambarkan urutan langkah demi langkah suatu proses dan interkoneksi antara berbagai proses di dalam suatu perangkat lunak. Menurut (Listyoningrum et al., 2023), yang menerangkan bahwa flowchart adalah gambaran grafik yang menjelaskan urutan suatu proses atau langkah-langkah secara sistematis untuk menjalankan sebuah program, dapat juga

diartikan sebagai suatu diagram yang menggunakan symbol - symbol grafis untuk menggambarkan alur dari suatu proses.

Berdasarkan kedua kutipan tersebut, dapat disimpulkan bahwa *flowchart* atau diagram alur merupakan representasi grafis yang menggunakan simbol-simbol tertentu untuk menggambarkan urutan langkah-langkah atau proses yang berjalan secara sistematis dalam suatu program atau perangkat lunak.

Tabel 2 1 Simbol *Flowchart*

Bagan	Nama	Fungsi
	<i>Flow</i>	Arah aliran program
	<i>Terminator</i>	Awal atau akhir program
	<i>Proces</i>	Menyatakan suatu proses
	<i>Input / Output</i>	Input / Output data
	<i>Decision</i>	Menyatakan suatu kondisi
	<i>Predefine Proces</i>	Menyatakan sub-program
	<i>Preparation</i>	Inisialisasi / pemberian nilai awal
	<i>Display</i>	Menyatakan alat output
	<i>Document</i>	Input / Output Berbentuk Dokumen

	<i>Manual Operation</i>	Proses Manual
	<i>On Page Connector</i>	Penghubung bagian dalam satu halaman
	<i>Off Page Connector</i>	Penghubung bagian berbeda halaman

Sumber : (L. Sari & Siregar, 2021)




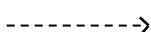
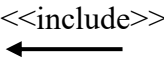
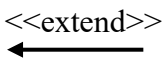
6. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) merupakan sebuah alat yang dimanfaatkan dalam mengembangkan sebuah sistem berorientasi objek dan juga merupakan alat bantu untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia TIK yang merupakan standar bahasa pemodelan umum dalam perancangan perangkat lunak dan pengembangan sistem (R. N. Sari & Rahayu, 2022). Menurut (Djohan et al., 2025), bagian *desain* perancangan perangkat lunak yang dituangkan menggunakan diagram *Unified Modeling Language (UML)*, yang memungkinkan pengembang untuk memvisualisasikan berbagai aspek sistem seperti fitur utama melalui *use case diagram*, alur kerja melalui *activity diagram*, interaksi antar objek melalui *sequence diagram*, serta struktur dan rancangan basis data melalui *entity relationship diagram (ERD)*. Dalam *UML* terdapat beberapa *class* yang menjadi kerangka penyusun didalamnya yang meliputi *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

a. *Use case diagram*

Menurut (Taufan et al., 2022), *use case diagram* merupakan dasar dalam merancang pemodelan perangkat lunak karena menyatakan interaksi yang terjadi antara pengguna dengan sistem secara visual serta mempresentasikan semua fungsi yang akan dilakukan oleh sistem. Adapun tabel mengenai *use case diagram* ini dapat dilihat pada tabel 2.2 seperti berikut.

Tabel 2.2 Simbol *use case diagram*






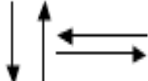
No	Simbol	Nama	Keterangan
1		<i>Aktor</i>	Mewakili peran orang
2		<i>Use Case</i>	Abstraksi antara sistem dan aktor
3		<i>Association</i>	Penghubung use case dengan aktor
4		<i>Generalisasi</i>	Spesialisasi aktor untuk berhubungan dengan use case
5		<i>Include</i>	Simbol yang menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya.
6		<i>Extend</i>	Simbol yang menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case

lainnya jika suatu kondisi terpenuhi.

Tabel 2.2 *Use Case Diagram* (Rasiban et al., 2024)

b. *Activity diagram*

Activity diagram digunakan untuk menampilkan tindakan dan sebagian dasar transisi yang dipicu oleh penyelesaian tindakan yang berasal dari sumber. *Activity diagram* mirip dengan *flowchart* yang menjelaskan proses yang terjadi antara *actor* dan sistem, (Lubis et al., 2023). Menurut (Nistrina & Sahidah, 2022), penggambaran berbagai alur pemrosesan data yang sedang dirancang dilakukan di *activity diagram*, yang akan menggambarkan proses berjalan dan membantu memahami proses sistem secara menyeluruh. Adapun simbol-simbol yang digunakan dalam *activity diagram* bisa dilihat pada gambar 2.1 berikut.


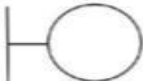
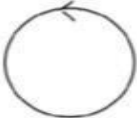
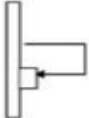


NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

Gambar 2.1 *Simbol Activity Diagram* (Rasiban et al., 2024)

c. *Sequence diagram*

Sequence diagram adalah suatu diagram yang menjelaskan hubungan antar objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi di antara objek-objek tersebut (Syabania & Rosmawarni, 2021).

Sedangkan menurut (Sumiati et al., 2021), *sequence diagram* dapat menjelaskan urutan waktu pemrosesan sistem, dan dimanfaatkan untuk menjelaskan keruntutan sistem per bagian objek. Adapun gambar mengenai simbol *sequence* dapat dilihat pada gambar 2.2 berikut.








Gambar	Nama	Keterangan
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya
	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

Gambar 2 2 Simbol *Sequence Diagram* (Rasiban et al., 2024)

d. *Class diagram*

Class diagram menjelaskan secara garis besar mengenai kelas-kelas perancangan sistem dari sudut pandang struktur sistem yang dapat memperjelas fungsi-fungsinya (T. Saputra et al., 2024). *Class diagram* digunakan untuk memodelkan kelas-kelas dalam sistem, atributnya, metodenya, dan hubungan antar kelas. Penerapan *class diagram* pada fase perancangan tujuannya untuk memperoleh model sistem yang jelas, terstruktur, dan mudah dipahami (Ramdany et al., 2024). Adapun simbol dan penjelasan mengenai *Class Diagram* dapat dipelajari pada Gambar 2.3 berikut.

SIMBOL CLASS DIAGRAM

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Gambar 2 3 Simbol Class Diagram (Rasiban et al., 2024)

B. Kajian Empiris

Sebagai pijakan utama dalam merancang sistem ini, peneliti merujuk pada berbagai studi terdahulu yang relevan, guna memperkuat dasar teori dan arah pengembangan sistem, khususnya yang berkaitan dengan pengolahan citra digital dan algoritma klasifikasi. Penelitian oleh (Syaputra, 2024), menjadi salah satu referensi penting. Dalam studi ini, ia membandingkan efektivitas algoritma *K-Nearest Neighbors* (KNN), *Multilayer Perceptron* (MLP), dan *Support Vector Machine* (SVM) dalam mengklasifikasikan penyakit daun tebu. Penelitian ini memanfaatkan model deep learning VGG16 untuk ekstraksi fitur citra, dan menggunakan aplikasi Orange sebagai alat bantu data mining. Dengan total 8.200 citra daun yang dibagi dalam data pelatihan dan pengujian, hasilnya menunjukkan bahwa algoritma MLP unggul dengan akurasi mencapai 97,4% di pelatihan dan 94,5% saat pengujian. Temuan ini menegaskan potensi besar pendekatan kecerdasan buatan dalam membangun sistem klasifikasi yang canggih, responsif, dan akurat.

Selanjutnya, penelitian yang dilakukan oleh (Sunandar & Sutopo, 2024), menyajikan pendekatan berbeda namun sejalan, dengan mengembangkan sistem klasifikasi penyakit tanaman padi berbasis pengolahan citra daun. Fokus mereka tertuju pada deteksi penyakit umum seperti *Bacterial Leaf Blight*, *Brown Spot*, dan *Leaf Smut*, dengan memanfaatkan data visual daun sebagai sumber utama informasi. Citra daun diunduh dari UCI Machine Learning Repository, lalu diproses melalui tahapan pra-pemrosesan dan pelabelan. Hasil penelitian menekankan pentingnya bentuk visual pada daun sebagai indikator

penyakit, serta efektivitas kecerdasan buatan dalam mengenali dan mengelompokkan pola-pola visual secara otomatis. Studi ini memperkuat gagasan bahwa data visual dapat menjadi fondasi yang kuat dalam pengembangan sistem deteksi penyakit tanaman yang efisien.

Penelitian oleh (Mustofa et al., 2022) juga memberikan kontribusi besar dalam ranah ini. Mereka mengusulkan metode kombinasi antara ekstraksi fitur tekstur *Gray Level Co-occurrence Matrix* (GLCM) dan algoritma KNN untuk mengidentifikasi penyakit pada daun apel. Permasalahan utama yang mereka angkat adalah deteksi dini terhadap penyakit seperti *Apple Scab*, *Black Rot*, dan *Cedar Apple Rust* yang berpotensi menurunkan hasil panen. Dengan menggunakan dataset citra daun apel, sistem yang dikembangkan mampu mencapai tingkat akurasi yang tinggi. Hasil ini menunjukkan bahwa pendekatan gabungan antara teknik pengolahan citra dan klasifikasi berbasis AI mampu memberikan solusi nyata dalam meningkatkan produktivitas pertanian melalui deteksi penyakit yang cepat dan akurat.

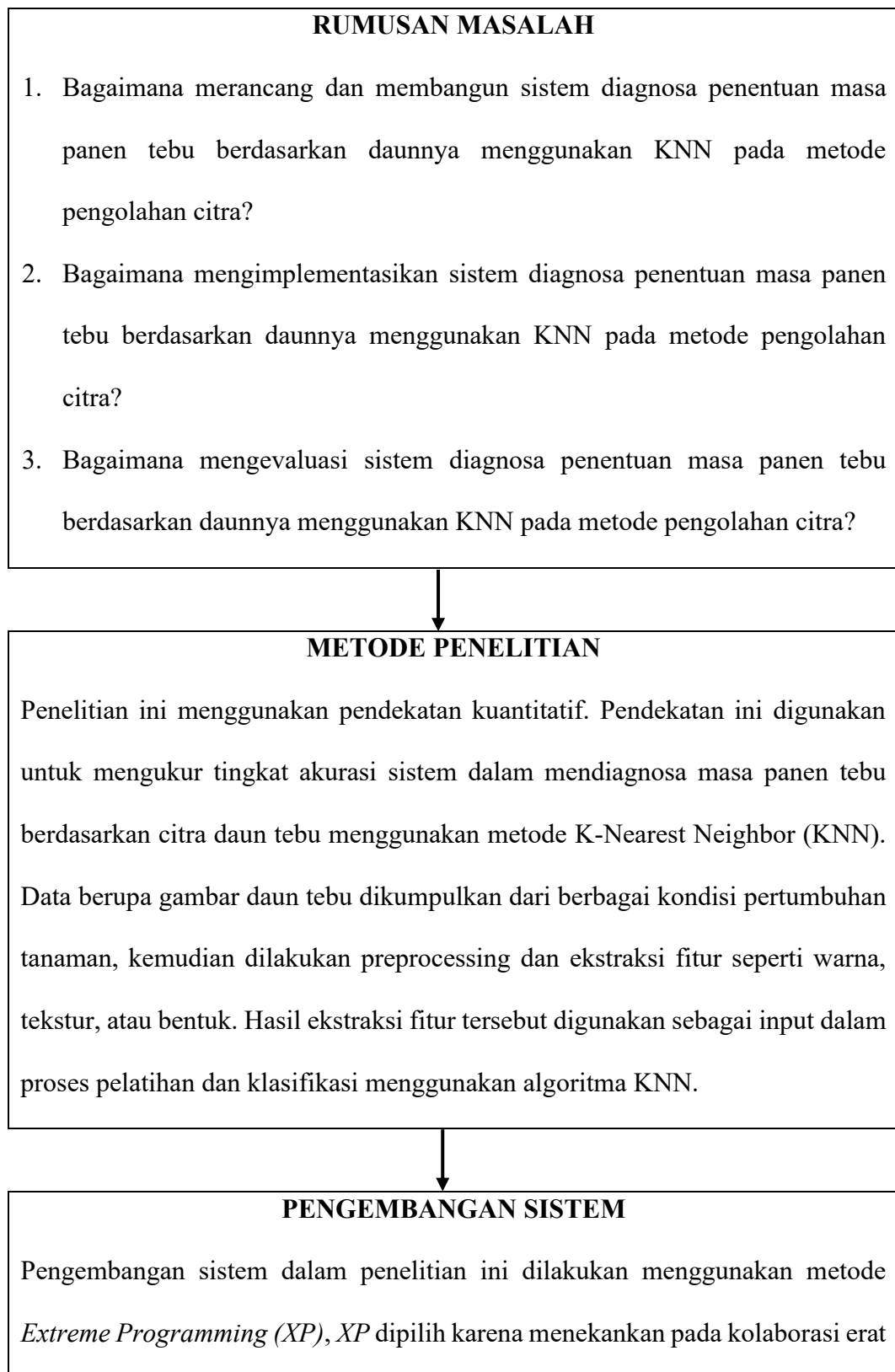
Sementara itu, (Luna-Benoso et al., 2021), menyajikan penelitian yang menarik dengan pendekatan analisis warna dan pengenalan pola berbasis citra digital untuk mendeteksi penyakit pada daun tomat. Mereka memfokuskan penelitian pada tiga jenis penyakit utama: *late blight*, *tomato mosaic virus*, dan *Septoria leaf spot*. Prosesnya terdiri dari segmentasi citra daun, ekstraksi ciri menggunakan momen warna dan GLCM, serta klasifikasi melalui gabungan algoritma KNN, SVM, dan MLP. Menggunakan dataset PlantVillage yang berisi 960 gambar daun tomat, sistem ini mampu membedakan daun sehat dan

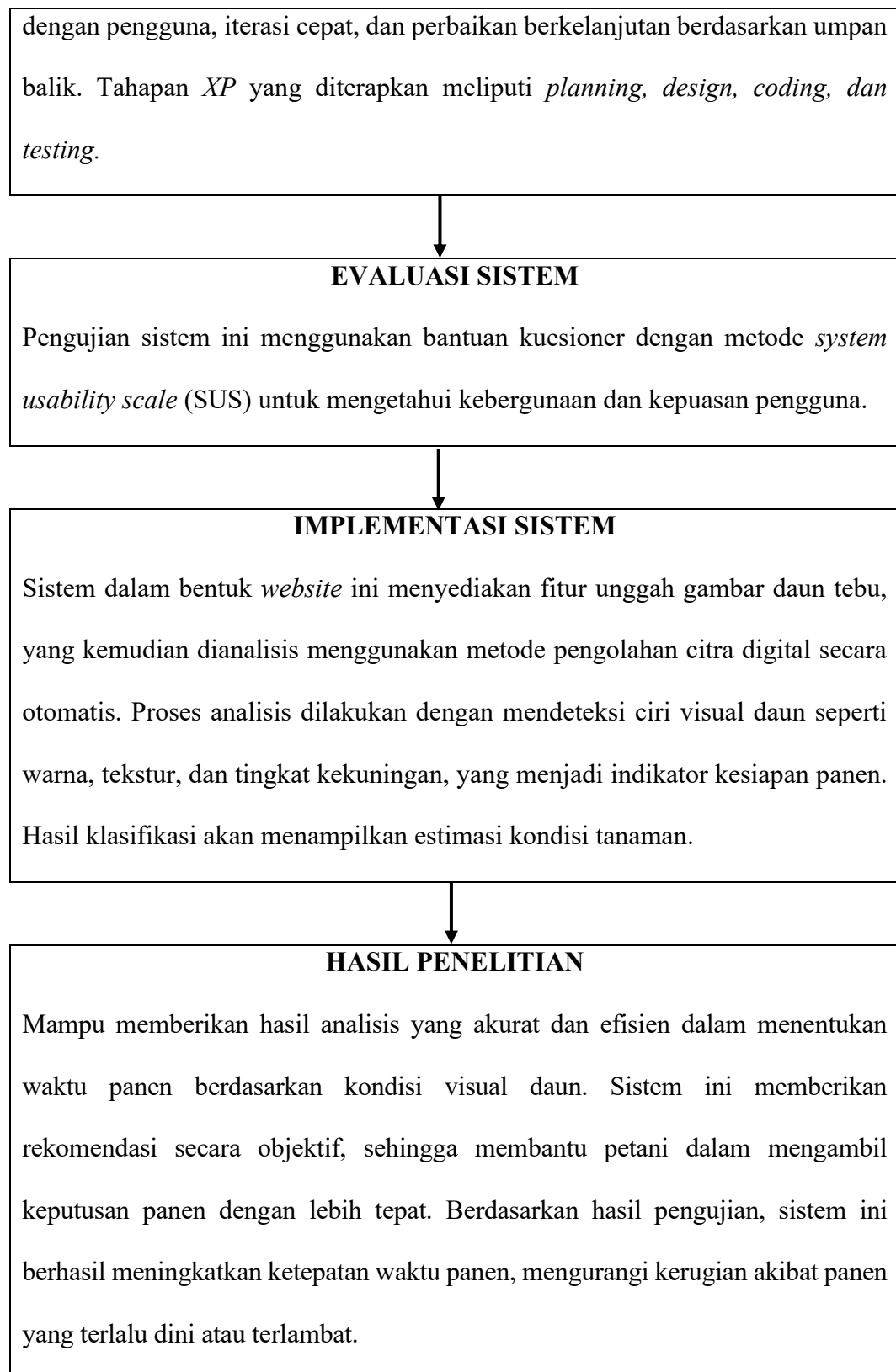
terinfeksi, sekaligus mengidentifikasi jenis penyakitnya. Studi ini membuktikan bahwa perubahan warna dan pola visual pada daun dapat menjadi indikator kuat untuk mendukung diagnosis otomatis dalam dunia pertanian modern.

Akhirnya, penelitian oleh (Arianto & Yunitasari, 2023) menyoroti perbandingan efektivitas antara algoritma *Convolutional Neural Network* (CNN) dan *Random Forest* dalam mengklasifikasikan penyakit daun pada tanaman kedelai. Mereka memanfaatkan framework Flask dan bahasa pemrograman Python, serta menerapkan metode pengembangan *Extreme Programming* untuk membangun sistem. Dataset yang digunakan berisi 1.400 citra daun kedelai yang terbagi dalam tujuh kategori penyakit, dan dilatih menggunakan 30 epoch serta augmentasi gambar sebagai bagian dari tahap preprocessing. Hasil akhir menunjukkan CNN mampu mencapai akurasi sempurna 100%, jauh mengungguli Random Forest yang hanya memperoleh 78%. Temuan ini menegaskan bahwa CNN sangat andal untuk sistem klasifikasi berbasis citra dan layak dijadikan model dalam pengembangannya.

Secara keseluruhan, kelima penelitian ini memperlihatkan arah yang konsisten dalam penerapan teknologi *Artificial Intelligence* dan pengolahan citra digital untuk mengidentifikasi penyakit tanaman melalui daun. Baik melalui CNN, MLP, SVM, maupun KNN, semuanya menekankan pentingnya fitur visual seperti GLCM, VGG16, dan warna sebagai komponen inti. Pendekatan-pendekatan ini membuka jalan bagi terciptanya sistem otomatis deteksi penyakit yang cerdas, cepat, dan dapat diandalkan dalam mendukung keberlanjutan pertanian masa depan.

C. Kerangka Berfikir





Gambar 2 4 Kerangka Berfikir