BAB II

TINJAUAN PUSTAKA

2.1 Sistem Cerdas

Sistem cerdas adalah suatu sistem berbasis teknologi informasi yang dirancang untuk meniru kemampuan kognitif manusia dalam mengolah data, mengambil keputusan, dan menyelesaikan permasalahan secara otomatis, terutama pada situasi yang kompleks dan dinamis. Sistem ini memanfaatkan teknik kecerdasan buatan seperti machine learning dan pengenalan pola untuk menangkap informasi dari data masukan, melakukan proses pembelajaran, serta menghasilkan *output* yang relevan dan adaptif terhadap konteks permasalahan yang dihadapi. Dengan kemampuan tersebut, sistem cerdas mampu membuat keputusan secara otomatis tanpa memerlukan pengawasan manusia secara langsung dalam setiap langkahnya. Hal ini menjadikan sistem cerdas semakin banyak digunakan dalam berbagai domain aplikasi, seperti sistem rekomendasi, sistem pakar medis, deteksi keamanan, hingga aplikasi pertanian dan industri(Fauzan et al., 2025).

Penelitian dalam jurnal Indonesia juga menunjukkan bahwa pengembangan sistem cerdas telah meluas ke berbagai disiplin ilmu dengan fokus utama pada penerapan teknologi *AI* untuk meningkatkan efisiensi dan efektivitas proses sistem. Salah satu kajian menyatakan bahwa sistem cerdas tidak hanya sekadar menjalankan perintah terprogram, tetapi juga memiliki kemampuan untuk belajar dari data, mengenali pola, serta memproses informasi baru secara otomatis, sehingga mampu menghadirkan solusi yang lebih cepat dan akurat untuk permasalahan nyata. Hal ini memperkuat alasan pemanfaatan sistem cerdas dalam

berbagai inovasi teknologi modern, termasuk dalam upaya diagnosis berbasis citra digital seperti yang direncanakan dalam penelitian ini.

2.2 *Website*

Website adalah sekumpulan halaman informasi yang disimpan di internet dan dapat diakses oleh pengguna melalui jaringan global menggunakan *web browser* seperti *Chrome* atau *Firefox*. *Website* berfungsi sebagai media penyampaian informasi yang dapat diakses kapan saja dan dari mana saja oleh pengguna yang terhubung ke internet. Dalam konteks sistem informasi, *website* tidak hanya berperan sebagai penyaji konten statis tetapi juga mampu menyediakan layanan dinamis, seperti interaksi data, tampilan informasi *real-time*, dan integrasi dengan basis data untuk kebutuhan aplikasi modern berbasis *web*. Hal ini menunjukkan bahwa *web*site merupakan komponen fundamental dalam pembangunan sistem berbasis internet yang efektif dan efisien(Mahmud et al., 2023).

Penggunaan *web*site telah berkembang pesat dan menjadi media utama dalam berbagai aplikasi teknologi informasi karena sifatnya yang *multiplatform* dan tidak tergantung pada sistem operasi tertentu. *Website* memungkinkan pengembang mengintegrasikan berbagai fitur seperti formulir interaktif, manajemen data, dan sistem *log-in* yang mendukung layanan aplikasi secara langsung kepada pengguna tanpa perlu instalasi khusus. Dengan demikian, *website* berperan sebagai antarmuka utama antara sistem dan pengguna dalam berbagai domain aplikasi, termasuk sistem diagnosis berbasis citra seperti yang direncanakan dalam penelitian ini.

2.3 Hama dan Penyakit Pada Tanaman Sawit

Hama dan penyakit pada tanaman kelapa sawit (*Elaeis guineensis*) merupakan salah satu faktor pembatas utama dalam budidaya sawit karena dapat menyebabkan penurunan pertumbuhan dan produktivitas tanaman secara signifikan. Hama yang umum ditemukan di perkebunan sawit antara lain kumbang tanduk (*Oryctes rhinoceros*), ulat api (*Setora nitens*), ulat kantong (*Metisa plana*), tikus (*Rattus spp.*), dan termite/serangga penggerek lainnya, yang menyerang berbagai bagian tanaman mulai dari pucuk hingga pelepah daun dan batang. Adapun penyakit yang sering dijumpai pada sawit meliputi busuk pangkal batang (*Ganoderma boninense*), bercak daun (*Curvularia sp.*), busuk buah, serta gangguan fisiologis lain yang sering berhubungan dengan kelembapan lingkungan dan kondisi tanah tertentu(Ridwan et al., 2025).

Gambar 2. 1 Hama Tanaman Sawit

Kelima hama dan penyakit tersebut dapat dikenali melalui gejala visual seperti kerusakan titik tumbuh akibat kumbang tanduk dan ulat, pengeringan daun oleh ulat api, kerusakan luas daun oleh ulat kantong, serta gejala bercak atau busuk pada daun dan buah akibat infeksi patogen jamur. Dampak serangan ini tidak hanya menurunkan kualitas tanaman tetapi juga berpotensi menurunkan

hasil panen dan produktivitas lahan secara menyeluruh apabila tidak dikendalikan secara efektif. Pemahaman mengenai jenis dan karakteristik serangan hama serta penyakit menjadi dasar penting dalam upaya deteksi dini dan strategi pengendalian yang tepat di lapangan(Lestari et al., 2025).

2.3.1 Defisiensi Pada Tanaman Sawit

Tanaman kelapa sawit (*Elaeis guineensis Jacq.*) merupakan komoditas utama perkebunan di Indonesia, namun kesehatan tanaman sering terpengaruh oleh berbagai jenis penyakit yang dapat menurunkan produktivitas. Salah satu penyakit yang sering dijumpai adalah penyakit bercak daun yang disebabkan oleh *Curvularia sp.*, ditandai oleh munculnya bercak berwarna kuning hingga coklat pada permukaan daun yang dapat menyebar secara cepat di pembibitan sawit, terutama pada fase awal pertumbuhan dan dipengaruhi oleh kondisi lingkungan seperti curah hujan dan kelembapan udara yang tinggi. Penelitian yang dilakukan menunjukkan bahwa penyakit bercak daun ini memiliki persentase serangan yang tinggi di beberapa lokasi pembibitan kelapa sawit di Indonesia dan menjadi salah satu ancaman utama terhadap pertumbuhan bibit, sehingga pemahaman mengenai ciri visual dan intensitas serangan penyakit sangat penting untuk strategi pengendalian dan diagnosis dini(Cameron et al., 2024).

2.3.2 Defisiensi Magnesium pada Tanaman Sawit

Defisiensi magnesium adalah kondisi kekurangan unsur hara magnesium yang dibutuhkan oleh tanaman sawit untuk menunjang proses fotosintesis dan metabolisme energi karena magnesium merupakan bagian dari molekul klorofil; apabila tanaman kekurangan Mg maka batang dan daun menunjukkan gejala seperti klorosis pada daun tua yang menguning di antara tulang daun, yang

mencerminkan gangguan fisiologis akibat rendahnya ketersediaan magnesium di dalam tanah dan penyerapan oleh tanaman. Di dalam studi *Jurnal Ilmiah Pertanian* dijelaskan bahwa magnesium termasuk salah satu unsur hara makro yang penting dan defisiensinya dapat dikenali secara visual pada daun kelapa sawit, sehingga pemahaman gejala defisiensi magnesium sangat penting untuk diagnosis nutrisi tanaman dan strategi pemupukan lebih lanjut guna mendukung pertumbuhan yang optimal(Satia et al., 2022).

Gambar 2. 2 Defisiensi Magnesium

2.3.3 Defisiensi Mangan pada Tanaman

Defisiensi mangan (Mn) adalah kondisi kekurangan unsur hara mikro mangan yang penting dalam proses fotosintesis dan metabolisme protein tanaman, sehingga kekurangannya akan memicu terganggunya fungsi enzim dan produksi klorofil, serta munculnya gejala klorosis interveinal (menguning di antara tulang daun) pada daun muda, pola yang sering dipakai untuk mengidentifikasi kebutuhan hara mikro tanaman. Dalam konteks tanaman perkebunan seperti kelapa sawit, gejala visual akibat kekurangan Mn dapat diamati sebagai peredupan warna hijau daun diikuti timbulnya bercak kekuningan, yang menunjukkan gangguan fisiologis di daun karena sistem metabolisme tanaman tidak sehat.

Penelitian yang memanfaatkan pendekatan visual dan analisis kadar hara mikro menyatakan bahwa kekurangan unsur Mn serta unsur mikro lainnya pada tanaman dapat diidentifikasi melalui perubahan pigmentasi daun dan berdampak negatif terhadap pertumbuhan, sehingga pemahaman ciri-ciri defisiensi mangan menjadi penting dalam diagnosis kesehatan tanaman secara umum(Khodijah et al., 2024).

Gambar 2. 3 Defisiensi Mangan

2.3.4 Hawar Rachis

Hawar rachis merupakan salah satu penyakit pada tanaman kelapa sawit yang menyerang bagian rachis, yaitu tulang utama pelepah daun yang menjadi penopang anak-anak daun (*leaflet*). Penyakit ini umumnya ditandai dengan munculnya bercak atau perubahan warna kecokelatan hingga kehitaman pada bagian rachis, yang kemudian dapat meluas dan menyebabkan jaringan mengering atau membusuk. Dalam kondisi yang parah, serangan hawar rachis dapat mengganggu distribusi nutrisi ke helaian daun sehingga daun menjadi layu, menguning, dan akhirnya mengering (Evizal & Prasmatiwi, 2022).

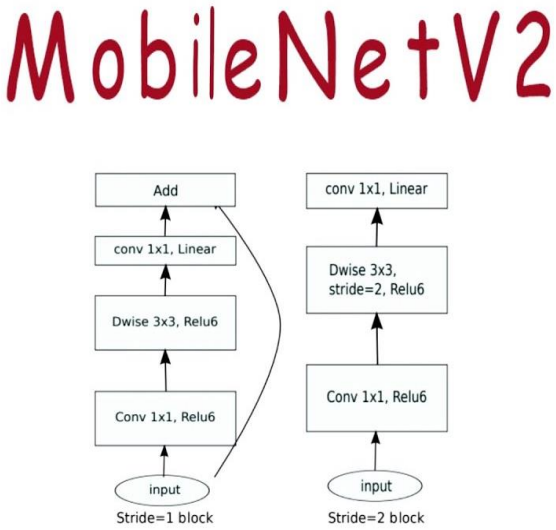
Gambar 2. 4 Hawar Rachis

Secara umum, hawar rachis disebabkan oleh infeksi patogen, terutama dari kelompok jamur yang berkembang pada kondisi lingkungan lembap dengan sirkulasi udara yang kurang baik. Penyebaran penyakit dapat terjadi melalui percikan air hujan, angin, maupun alat pertanian yang terkontaminasi. Dampak dari serangan hawar rachis tidak hanya menurunkan kualitas daun, tetapi juga dapat mengurangi kemampuan fotosintesis tanaman sehingga berpotensi menurunkan produktivitas tandan buah kelapa sawit apabila tidak ditangani dengan baik. Oleh karena itu, deteksi dini melalui pengamatan visual maupun sistem berbasis kecerdasan buatan sangat penting untuk mencegah penyebaran yang lebih luas.

2.4 *MobileNetV2*

MobileNetV2 adalah salah satu varian arsitektur *Convolutional Neural Network (CNN)* yang dirancang khusus untuk efisiensi komputasi dan ukuran model ringan, sehingga sangat cocok digunakan pada aplikasi yang memerlukan pemrosesan citra cepat tanpa sumber daya perangkat yang tinggi. *MobileNetV2* mengimplementasikan *depthwise separable convolution* dan *inverted residual*

with linear bottleneck, yang membantu menurunkan jumlah parameter dan operasi komputasi sambil mempertahankan kemampuan ekstraksi fitur visual yang baik. Pendekatan ini membuat *MobileNetV2* banyak dipilih dalam penelitian aplikatif yang memerlukan klasifikasi citra otomatis dengan responsivitas tinggi, terutama pada sistem berbasis *web* atau perangkat terbatas(Santosa et al., 2025).

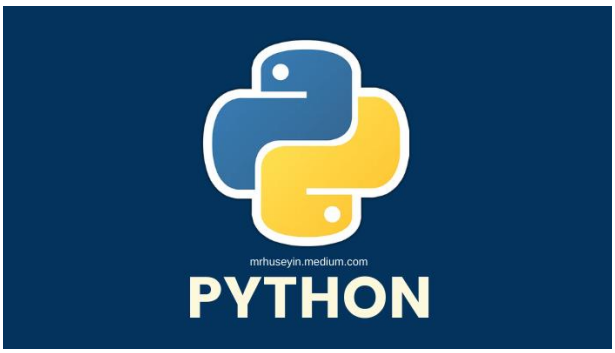


Gambar 2. 5 *MobileNetV2*

Beberapa penelitian di Indonesia telah menunjukkan efektivitas penggunaan *MobileNetV2* dalam tugas klasifikasi citra dengan hasil yang memuaskan. Misalnya, studi perbandingan antara *MobileNetV2* dan model CNN lainnya untuk klasifikasi tingkat kematangan tomat menemukan bahwa arsitektur *MobileNetV2* memberikan hasil klasifikasi yang kompetitif, menunjukkan bahwa model ini memiliki kemampuan representasi fitur yang kuat sekaligus efisien dalam komputasi. Temuan semacam ini mendukung penggunaan *MobileNetV2* sebagai basis model dalam sistem diagnosis otomatis gambar tanaman yang direncanakan dalam penelitian ini(Ristanti, 2024).

2.5 Bahasa Pemrograman *Python*

Bahasa pemrograman *Python* merupakan bahasa tingkat tinggi yang banyak digunakan dalam pengembangan aplikasi perangkat lunak, kecerdasan buatan (*AI*), dan pemrosesan data. *Python* memiliki sintaks yang sederhana, mudah dipahami, serta mendukung paradigma pemrograman berorientasi objek, prosedural, dan fungsional, sehingga memudahkan pengembang dalam membangun sistem kompleks secara efisien. Di samping itu, *Python* didukung oleh ekosistem pustaka (*libraries*) yang sangat kaya, seperti NumPy, Pandas, TensorFlow, dan Scikit-Learn, yang secara luas digunakan dalam penelitian dan pengembangan sistem berbasis *machine learning* maupun *deep learning* untuk berbagai aplikasi, termasuk pengolahan citra dan sistem diagnosis cerdas (Alfarizi & Al-farish, 2023).



Gambar 2. 6 *Phyton*

Dalam pengembangan aplikasi berbasis *web*, *Python* juga berperan penting dalam sisi *backend* karena dapat mengintegrasikan proses logika aplikasi, manipulasi data, dan eksekusi model kecerdasan buatan sebelum hasilnya ditampilkan ke antarmuka pengguna. Framework *Python* seperti Flask dan Django memungkinkan pembangunan aplikasi *web* yang dinamis dan responsif dengan kemampuan untuk menangani permintaan (*requests*) dari klien serta memproses

data secara efisien. Hal ini menjadikan *Python* sebagai pilihan utama dalam banyak penelitian sistem informasi dan aplikasi berbasis *web* yang membutuhkan kapabilitas pemrosesan data dan interaksi *real-time* antara pengguna dan sistem(Fahriza, 2024).

2.6 MongoDB

MongoDB merupakan salah satu jenis basis data NoSQL (*Not Only* SQL) yang menggunakan model penyimpanan data berbasis dokumen (*document-oriented*), sehingga tidak lagi bergantung pada struktur tabel relasional seperti pada database tradisional. Data dalam MongoDB disimpan dalam bentuk dokumen JSON (*JavaScript Object Notation*) atau BSON (*Binary JSON*) yang memungkinkan penyesuaian skema secara fleksibel sesuai kebutuhan aplikasi. Struktur ini memudahkan pengembangan aplikasi yang memerlukan skalabilitas tinggi dan performa responsif dalam pengolahan data besar, terutama ketika struktur data sering berubah atau tidak seragam. Studi pemanfaatan MongoDB dalam sistem informasi akademik menunjukkan bahwa penggunaan basis data ini mampu meningkatkan efisiensi pengambilan data serta fleksibilitas struktur data dibandingkan database relasional(Syahwali, 2025).

Gambar 2. 7 MongoDB

Selain fleksibilitas struktur data, MongoDB juga mendukung performa yang baik dalam pengelolaan data tidak terstruktur yang berkembang pesat pada aplikasi *modern*. Hal ini membuat MongoDB sangat relevan bagi sistem informasi kontemporer yang memerlukan integrasi cepat antara aplikasi *web* dan server database, termasuk pada pengembangan sistem cerdas berbasis *web*. Dalam konteks pengembangan teknologi informasi, pemilihan basis data NoSQL seperti MongoDB bergantung pada kebutuhan spesifik aplikasi, terutama bila data yang harus dikelola bersifat *dynamic schema* atau beragam formatnya.

2.7 *Unified Modeling Language* (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan untuk mendeskripsikan, merancang, dan memvisualisasikan struktur serta perilaku sistem perangkat lunak. UML membantu pengembang dalam menganalisis kebutuhan sistem dan menyusun desainnya melalui berbagai diagram, seperti *use case*, *class*, *sequence*, dan *Activity diagrams*. Hal ini memudahkan komunikasi antar tim pengembang serta memberikan gambaran yang jelas tentang komponen dan hubungan dalam sistem. Penelitian terbaru menunjukkan bahwa penerapan UML pada perancangan sistem informasi dapat memperjelas kebutuhan fungsional sistem dan alur proses kerja secara terstruktur, sehingga meningkatkan pemahaman tim terhadap spesifikasi sistem yang akan dikembangkan(Setiaji et al., 2024).

Gambar 2. 8 UML

Beberapa studi juga menyebut bahwa penggunaan UML dalam tahap perancangan sistem informasi tidak hanya membantu proses dokumentasi, tetapi juga mempermudah tahap implementasi dan pengujian karena diagram UML mampu menggambarkan interaksi antara komponen sistem secara visual. Dengan demikian, UML dapat berfungsi sebagai alat bantu utama dalam metode pengembangan sistem berorientasi objek maupun terstruktur, yang sangat penting dalam pengembangan aplikasi berbasis *web*, termasuk sistem diagnosis cerdas seperti yang direncanakan dalam penelitian ini (Susanto & Cahyono, 2024).

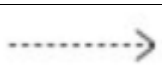
UML terdiri atas beberapa jenis diagram yang masing-masing memiliki fungsi berbeda dalam menggambarkan aspek struktural maupun perilaku sistem. Diagram UML dikelompokkan menjadi diagram struktural dan diagram perilaku, yang digunakan secara saling melengkapi dalam proses perancangan sistem berbasis objek.

2.7.1 *Use case* Diagram

Use case Diagram merupakan salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang berfungsi untuk menggambarkan interaksi fungsional antara pengguna sistem (*aktor*) dengan sistem itu sendiri, sehingga mempermudah dalam mendefinisikan ruang lingkup dan kebutuhan fungsional

dari sistem yang akan dikembangkan. Diagram ini menyajikan gambaran secara visual mengenai fungsi-fungsi utama yang dapat dilakukan oleh aktor dalam sistem serta bagaimana hubungan antar aktor dengan proses-proses tersebut, sehingga menjadi dasar penting dalam proses analisis dan desain sistem informasi. *Use case* Diagram sangat berguna dalam menyusun kebutuhan fungsional sistem serta memfasilitasi komunikasi antara tim pengembang dengan pemangku kepentingan lainnya, karena memberikan gambaran jelas mengenai layanan sistem dari sudut pandang pengguna (Sihombing & Juledi, 2024). Berikut iniadalah simbol diagram *use case* yang sering digunakan.

Tabel 2. 1 Simbol *Diagram Use case*

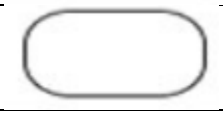
NO	SIMBOL (BENTUK)	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang dimainkan pengguna ketika berinteraksi dengan *use case*.
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen lain yang bergantung padanya.
3		Generalization	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek induk (*ancestor*).
4		Include	Menspesifikasikan bahwa suatu *use case* selalu menyertakan *use case* lainnya secara eksplisit.
5		Extend	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada kondisi tertentu.
6		Association	Menggambarkan hubungan antara aktor dengan *use case* atau antar objek dalam sistem.
7		System	Menspesifikasikan batasan sistem yang memuat seluruh *use case*.

			
8		*Use case*	Mendeskripsikan fungsi atau layanan sistem yang menghasilkan keluaran bagi aktor.
9		Collaboration	Menunjukkan interaksi antar elemen sistem yang bekerja sama untuk mencapai tujuan tertentu.
10		Note	Digunakan untuk memberikan keterangan tambahan atau catatan pada elemen diagram.

2.7.2 *Activity* Diagram

Activity Diagram merupakan salah satu diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk memodelkan alur aktivitas atau proses kerja dalam suatu sistem secara berurutan dari awal hingga akhir. Diagram ini memanfaatkan simbol-simbol seperti *Activity*, *Action*, *Initial Node*, *Activity Final Node*, dan *Fork Node* untuk menggambarkan urutan proses, percabangan alur, serta aktivitas paralel, sehingga membantu pengembang dalam memahami proses bisnis sistem secara sistematis dan terstruktur. *Activity* Diagram sangat berguna dalam tahap analisis dan perancangan sistem karena mampu memberikan visualisasi yang jelas terhadap alur kerja sistem sebelum tahap implementasi dilakukan(Rahmalisa et al., 2022).

Tabel 2. 2 Simbol *Activity Diagram*

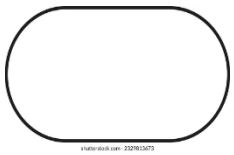
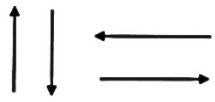
NO	SIMBOL (BENTUK)	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan aktivitas atau proses yang dilakukan dalam sistem.
2		*Action*	Menunjukkan state dari sistem yang mencerminkan pelaksanaan suatu aksi.
3		Initial Node	Menunjukkan titik awal suatu aktivitas atau proses.
4		*Activity* Final Node	Menunjukkan akhir dari suatu aktivitas atau proses.
5		Fork Node	Menunjukkan satu alur yang bercabang menjadi beberapa alur secara paralel.

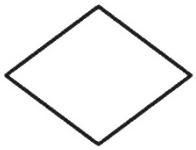
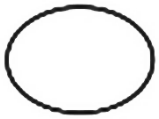
2.8 *Flowchart*

Flowchart merupakan diagram yang menggambarkan alur proses atau langkah-langkah logis suatu sistem atau program secara visual melalui simbol-simbol standar seperti panah, kotak proses, dan belah ketupat untuk keputusan. *Flowchart* digunakan selama analisis dan perancangan sistem untuk mempermudah pemahaman langkah kerja, identifikasi alur data dan proses, serta membantu pengembang dalam menyusun logika fungsi sistem secara sistematis. Diagram ini sering dipakai dalam dokumentasi proses dan membantu pengembang serta tim lain melihat struktur proses secara keseluruhan sebelum implementasi kode program dimulai, sehingga meminimalkan kesalahan logika pada tahap pengembangan(Listyoningrum et al., 2023).

Selain sebagai alat bantu desain, *flowchart* juga dapat digunakan dalam penyusunan algoritma untuk *debugging*, perencanaan proses, dan evaluasi prosedur kerja dalam sistem informasi berbasis *web* maupun *standalone*. Keunggulan *flowchart* adalah kemampuannya untuk menyajikan informasi kompleks dalam bentuk yang mudah dipahami, terutama untuk tim yang terlibat dalam pengembangan sistem yang memerlukan visualisasi langkah proses lengkap, seperti sistem diagnosis otomatis(Rahmadan & Gunawan, 2024). Berikut ini adalah beberapa simbol *flowchart* yang sering digunakan.

Tabel 2. 3 Simbol *Flowchart*

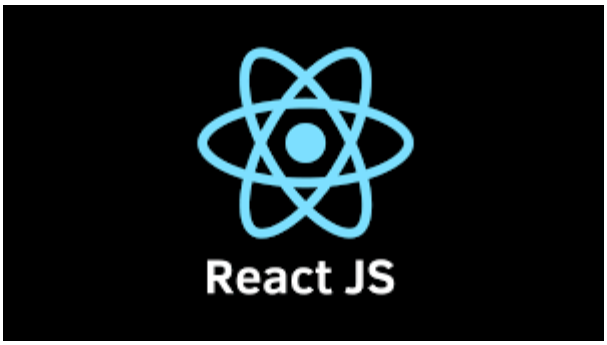
NO	SIMBOL	NAMA	FUNGSI
1		Terminator	Menandai titik awal (start) atau akhir (end) dari sebuah alur program.
2		Process	Menunjukkan kegiatan pemrosesan data, perhitungan, atau operasi tertentu.
3		*Input / Output*	Digunakan untuk proses memasukkan data atau menampilkan hasil data
4		*Prepation*	Digunakan untuk menyiapkan suatu variabel atau tempat penyimpanan suatu pengolahan data atau pemberian awal.
5		*Flow Line*	Menunjukkan arah aliran proses dari satu symbol ke symbol lainnya.

6		*Decision*	Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.
7		*Connector on-page*	simbol untuk menunjukkan hubungan simbol dalam *flowchart* sebagai pengganti garis untuk menyederhanakan bentuk saat simbol yang akan dihubungkan jaraknya berjauhan dan rumit jika dihubungkan dengan garis tapi masih di halaman yang sama.
8		*Predifine Proses*	Simbol untuk pelaksanaan suatuu bagian (sub-program) atau *procedure*.
9		*Off-Page Connector*	Simbol untuk keluar-masuk atau penyambungan proses dalam lembar kerja yang berbeda
10		Dokumen	Simbol yang menyatakan bahwa *input* berasal dari dokumen dalam bentuk fisik, atau *output* yang perlu dicetak.

2.9 *ReactJS*

ReactJS adalah *library JavaScript* yang digunakan untuk membangun antarmuka pengguna (*frontend*) pada aplikasi *web* modern dengan cara yang lebih efisien dan modular. *ReactJS* mendukung konsep *component-based architecture*, di mana setiap bagian antarmuka *web* dibagi menjadi komponen-komponen kecil

yang dapat digunakan ulang dan dikombinasikan untuk membentuk tampilan aplikasi yang kompleks. Dengan pendekatan ini, pengembang dapat menciptakan tampilan *web* yang dinamis dan responsif tanpa perlu memuat ulang seluruh halaman sebuah karakteristik penting dalam pengembangan *Single Page Application* (SPA) yang interaktif dan cepat. Studi pengembangan *ReactJS* pada aplikasi persediaan barang menunjukkan bahwa penggunaan *ReactJS* mampu membantu menciptakan antarmuka yang lebih efisien dalam menangani interaksi pengguna serta mempercepat proses pemutakhiran tampilan halaman *web*(Bastian et al., 2023).



Gambar 2. 9 ReactJS

Selain itu, *ReactJS* memungkinkan *frontend web* untuk berkomunikasi secara efektif dengan *backend* melalui API (*Application Programming Interface*), sehingga data dari server dapat ditampilkan secara *real-time* sesuai kebutuhan pengguna. Integrasi *ReactJS* dengan *backend* seperti *Python* (misalnya menggunakan Flask atau Django) sering dipilih dalam pengembangan sistem informasi *modern* karena *ReactJS* dapat menyajikan pengalaman pengguna yang lebih baik tanpa gangguan *reload* pada setiap aksi. Hal ini mendukung pemanfaatan *ReactJS* dalam berbagai aplikasi berbasis *web*, termasuk sistem diagnosis hama dan penyakit tanaman sawit, di mana *ReactJS* digunakan untuk

menampilkan hasil diagnosis dengan cepat dan interaktif kepada pengguna setelah data diproses oleh *backend*.

2.10 Penelitian Terdahulu

Penelitian terdahulu disusun untuk mengetahui posisi penelitian yang akan dilakukan dengan membandingkannya terhadap penelitian-penelitian sebelumnya yang relevan, khususnya yang berkaitan dengan sistem cerdas, klasifikasi citra tanaman, *deep learning*, serta implementasi berbasis *web*.

Tabel 2. 4 Penelitian Terdahulu

No	Peneliti & Tahun	Judul Penelitian	Metode / Teknologi	Objek Penelitian	Hasil Penelitian
1	Pratama dkk. (2022)	Sistem Deteksi Penyakit Daun Tanaman Menggunakan CNN	CNN	Daun tanaman pangan	Model CNN mampu mengklasifikasikan penyakit daun dengan tingkat akurasi yang baik berdasarkan citra daun
2	Siregar & Putra (2022)	Klasifikasi Penyakit Tanaman Berbasis Citra Digital	*Deep learning*	Daun tanaman perkebunan	Sistem berhasil mengenali jenis penyakit berdasarkan pola visual daun
3	Rahman dkk. (2023)	Penerapan *MobileNetV2* untuk Klasifikasi Citra Tanaman	*MobileNetV2*	Citra daun tanaman	*MobileNetV2* memberikan performa efisien dengan akurasi tinggi dan waktu komputasi rendah
4	Hidayat dkk. (2023)	Implementasi Sistem Deteksi Penyakit Tanaman Berbasis *Web*	CNN + *Web*	Tanaman hortikultura	Sistem berbasis *web* memudahkan pengguna dalam mengakses hasil diagnosis
5	Nugroho	Sistem	CNN	Daun	Sistem mampu

	dkk. (2024)	Diagnosis Penyakit Tanaman Menggunakan *Deep learning*		tanaman pertanian	memberikan hasil diagnosis otomatis berbasis citra
6	Kurniawan dkk. (2024)	Aplikasi Klasifikasi Citra Daun Menggunakan Model CNN Ringan	MobileNet	Daun tanaman	Model ringan cocok untuk aplikasi dengan keterbatasan sumber daya
7	Lestari dkk. (2024)	Pengembangan Sistem Klasifikasi Penyakit Tanaman Berbasis *Web*	*Web + AI*	Tanaman pertanian	Integrasi *AI* dan *web* meningkatkan kemudahan penggunaan sistem
8	Putri & Wibowo (2024)	Identifikasi Penyakit Tanaman Menggunakan *Deep learning*	CNN	Daun tanaman	Sistem mampu mengidentifikasi penyakit dengan tingkat akurasi tinggi
9	Saputra dkk. (2025)	Sistem Cerdas Diagnosis Penyakit Tanaman Berbasis Citra	*Deep learning*	Tanaman perkebunan	Pendekatan berbasis citra efektif untuk deteksi dini penyakit tanaman
10	Ramadhan dkk. (2025)	Implementasi *MobileNetV2* pada Sistem Deteksi Penyakit Tanaman	*MobileNetV2*	Daun tanaman	*MobileNetV2* cocok untuk sistem diagnosis *real-time*