

BAB II

KAJIAN LITERATUR

2.1 Sistem Informasi Rekomendasi

Sistem informasi rekomendasi merupakan bagian dari pengembangan sistem informasi yang berfungsi untuk memberikan saran atau rekomendasi kepada pengguna berdasarkan data, preferensi, maupun pola perilaku pengguna sebelumnya. Di era digital saat ini, perkembangan teknologi informasi yang sangat pesat menyebabkan terjadinya ledakan informasi (*information overload*), sehingga pengguna sering mengalami kesulitan dalam menentukan pilihan yang sesuai dengan kebutuhan mereka. Oleh karena itu, sistem rekomendasi hadir sebagai solusi untuk membantu proses pengambilan keputusan secara lebih efektif dan efisien. Secara umum, sistem merupakan sekumpulan elemen yang saling berhubungan, saling bergantung, dan bekerja sama untuk mencapai tujuan tertentu. Dalam konteks sistem informasi, sistem tidak hanya mencakup komponen teknologi, tetapi juga melibatkan manusia, prosedur, serta data yang saling terintegrasi. Sistem informasi sendiri berfungsi untuk mengumpulkan, mengolah, menyimpan, dan mendistribusikan informasi guna mendukung pengambilan keputusan dalam suatu organisasi [11].

Sistem rekomendasi adalah sistem yang dirancang untuk memprediksi preferensi atau penilaian (*rating*) seorang pengguna terhadap suatu item, kemudian memberikan saran item yang paling relevan. *Item* yang dimaksud dapat berupa produk, buku, film, layanan, maupun informasi lainnya. Sistem ini bekerja dengan memanfaatkan data interaksi antara pengguna dan item, seperti rating, riwayat pembelian, klik, atau pencarian. Dalam implementasinya, sistem rekomendasi memiliki beberapa komponen utama, yaitu [11]:

1. Matriks Interaksi.

Matriks interaksi merupakan representasi hubungan antara pengguna dan item dalam bentuk matriks berukuran $M \times N$, di mana M adalah jumlah pengguna dan N adalah jumlah item. Matriks ini berisi nilai rating atau interaksi pengguna terhadap item tertentu.

2. Fitur Pengguna dan Item.

Fitur pengguna dapat berupa data indikator maupun metadata yang menggambarkan karakteristik pengguna, sedangkan fitur item mencerminkan atribut yang dimiliki oleh suatu produk atau layanan.

3. Fungsi Representasi.

Fungsi ini bertugas merepresentasikan pengguna dan item ke dalam bentuk tertentu (misalnya vektor) agar dapat dihitung tingkat kesamaannya.

4. Fungsi Prediksi

Fungsi prediksi digunakan untuk memperkirakan rating atau tingkat kesukaan pengguna terhadap item yang belum pernah dinilai sebelumnya.

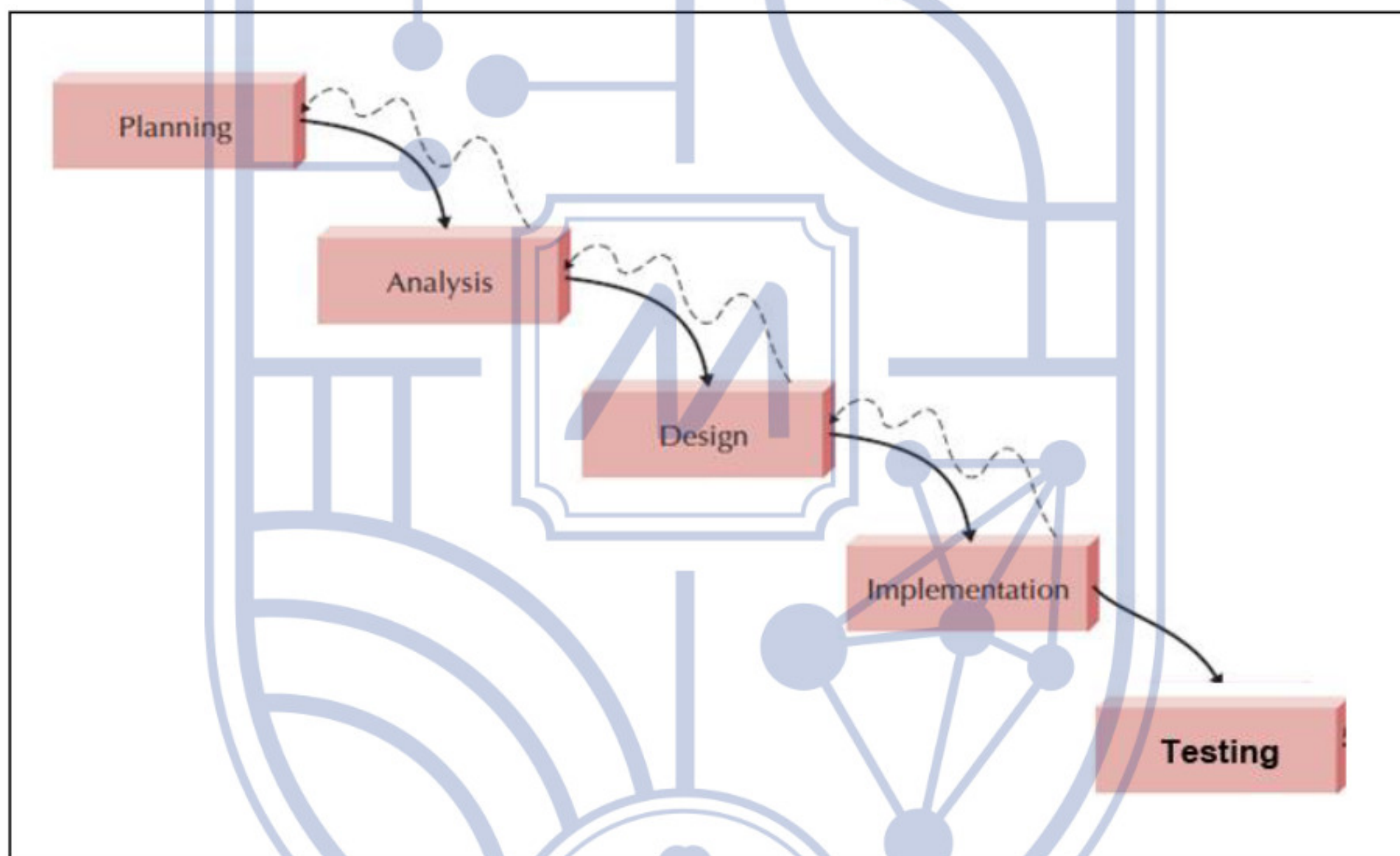
Pendekatan dalam sistem rekomendasi secara umum terbagi menjadi beberapa metode, antara lain [11]:

1. *Content-Based Filtering*, yaitu metode yang merekomendasikan item berdasarkan kesamaan atribut dengan item yang sebelumnya disukai oleh pengguna. Sistem membangun profil pengguna berdasarkan preferensi historisnya.
2. *Collaborative Filtering*, yaitu metode yang memberikan rekomendasi berdasarkan kesamaan perilaku atau preferensi antar pengguna. Jika dua pengguna memiliki pola rating yang mirip, maka item yang disukai oleh salah satu pengguna dapat direkomendasikan kepada pengguna lainnya.
3. *Hybrid Approach*, yaitu kombinasi dari beberapa metode untuk meningkatkan akurasi dan mengurangi kelemahan masing-masing pendekatan.

Dalam bidang pertanian, sistem rekomendasi juga mulai banyak dimanfaatkan untuk membantu petani dalam mengambil keputusan yang lebih tepat, khususnya dalam pengelolaan lahan dan pemupukan tanaman. Sistem rekomendasi pada pertanian dapat memberikan saran terkait jenis pupuk yang sesuai, dosis pemupukan, waktu pemberian pupuk, serta tindakan penanganan berdasarkan kondisi tanaman dan lahan. Rekomendasi tersebut biasanya dihasilkan melalui analisis data seperti kondisi tanah, jenis tanaman, gejala kekurangan unsur hara, maupun citra daun tanaman yang dianalisis menggunakan teknologi pengolahan citra dan *Machine Learning*. Pada tanaman kelapa sawit misalnya, sistem rekomendasi dapat digunakan untuk membantu mengidentifikasi gejala kekurangan unsur hara seperti nitrogen, fosfor, atau kalium melalui analisis citra daun. Berdasarkan hasil identifikasi tersebut, sistem kemudian dapat memberikan rekomendasi jenis pupuk dan tindakan pemupukan yang sesuai untuk mengatasi defisiensi unsur hara pada tanaman. Dengan adanya sistem rekomendasi di bidang pertanian, proses pengambilan keputusan oleh petani menjadi lebih cepat, akurat, dan berbasis data sehingga dapat meningkatkan produktivitas serta efisiensi penggunaan pupuk. Dalam mengembangkan sistem informasi rekomendasi diperlukan metodologi pengembangan sistem untuk menjamin sistem yang dibangun berjalan dengan sesuai. Salah satu metodologi pengembangan sistem yang umum digunakan adalah *Waterfall Model* [11].

2.2 Waterfall Model

Waterfall Model sering disebut sebagai metode air terjun. Dengan metode berbasis pengembangan air terjun, para analis dan pengguna melanjutkan secara berurutan dari satu fase ke fase berikutnya. Kunci kiriman untuk setiap fase biasanya sangat panjang (seringkali ratusan halaman) dan disajikan kepada sponsor proyek untuk mendapatkan persetujuan saat proyek berpindah dari fase ke fase. Setelah sponsor menyetujui pekerjaan yang dilakukan untuk suatu fase, fase tersebut berakhir dan selanjutnya dimulai. Metode ini disebut pengembangan air terjun karena bergerak maju dari fase ke fase dengan cara yang sama seperti air terjun. Gambar 2.1 berikut ini menunjukkan tahapan pengembangan sistem dengan metode *Waterfall Model* [12].



Gambar 2.1 Tahapan Metode *Waterfall Model* [11]

Berikut ini akan diuraikan tahapan-tahapan dari metode *Waterfall Model* yang ditunjukkan pada Gambar 2.1, yaitu [12]:

1. *Planning*/Perencanaan

Pada tahap ini dijelaskan secara singkat mengenai lingkungan atau organisasi yang menjadi lokasi penelitian, bidang kegiatan yang dijalankan, serta aktivitas operasional yang berkaitan dengan sistem yang akan dikembangkan. Penjelasan tersebut menjadi dasar untuk memahami kondisi operasional yang sedang berjalan dan digambarkan dengan *Flowchart*, sehingga dapat diperoleh pemahaman awal mengenai permasalahan yang dihadapi serta kebutuhan sistem yang akan dikembangkan.

2. *Analysis/Analisis*

Pada tahap analisis kebutuhan, dilakukan pengumpulan dan analisis kebutuhan sistem yang meliputi kebutuhan fungsional dan nonfungsional. Analisis ini bertujuan untuk memastikan bahwa sistem yang dikembangkan sesuai dengan kebutuhan pengguna. *Tools* yang digunakan terbagi menjadi 2 yaitu analisis kebutuhan fungsional dengan *Use Case Diagram* dan analisis kebutuhan non fungsional dengan PIECES yang meliputi aspek *Performance, Information, Economy, Control, Efficiency*, dan *Service*.

3. *Design/Desain*

Tahap ketiga ini biasanya mencakup kepentingan desain teknis, seperti bahasa pemrograman, lapisan data, layanan, dan sebagainya. Spesifikasi desain biasanya akan dibuat untuk menguraikan bagaimana logika bisnis yang tercakup dalam analisis akan diimplementasikan secara teknis. *Tools* yang digunakan menggunakan Visual Studio Code sebagai *text editor*.

4. *Implementation/Implementasi*

Dari tahap ketiga dan kedua didapatkan sebuah rancangan sistem secara teknis, kemudian akan dilakukan implementasi dengan membangun aplikasi menggunakan bahasa-bahasa pemrograman yang dipilih. Pada penelitian ini, bahasa pemrograman yang digunakan adalah bahasa pemrograman Python dengan *framework* Flask.

5. *Testing/Pengujian*

Tahap terakhir adalah melakukan *testing* terhadap aplikasi yang telah dibangun dan diimplementasikan agar dapat diperbaiki *bug* ataupun *error* yang terdapat pada aplikasi. Pada penelitian ini, metode pengujian yang digunakan adalah dengan *Confusion Matrix*.

2.3 *Artificial Intelligence (AI)*

AI merupakan sistem yang dirancang untuk meniru kemampuan kognitif manusia dalam memecahkan masalah. Kesimpulannya, AI dalam penelitian ini berperan sebagai otak sistem yang belajar mengenali pola defisiensi nutrisi melalui data historis (*dataset*) tanpa perlu diprogram secara eksplisit untuk setiap aturan visualnya [8]. Tujuan utama AI adalah untuk mengembangkan sistem yang dapat melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia [13]. Beberapa tujuan utama AI meliputi [14]:

1. Pengolahan Bahasa Alami (*Natural Language Processing*).

AI bertujuan untuk mengembangkan sistem yang mampu memahami, menghasilkan, dan berinteraksi dengan bahasa manusia secara alami. Hal ini melibatkan kemampuan

sistem untuk memahami konteks, sintaksis, semantik, dan pragmatik dalam bahasa manusia.

2. Pembelajaran Mesin (*Machine Learning*).

AI bertujuan untuk mengembangkan algoritma dan teknik yang memungkinkan mesin untuk belajar dari data dan pengalaman, tanpa diprogram secara eksplisit. Pembelajaran mesin melibatkan identifikasi pola, membuat prediksi, dan mengoptimalkan kinerja sistem berdasarkan data yang diberikan.

3. Pengenalan Pola (*Pattern Recognition*).

AI bertujuan untuk mengembangkan sistem yang dapat mengenali pola dan fitur dalam data, seperti pengenalan wajah, pengenalan suara, dan pengenalan tulisan tangan. Ini melibatkan penggunaan algoritma dan teknik untuk mengidentifikasi dan mengklasifikasikan pola-pola yang ada dalam data.

4. Penalaran dan Pengambilan Keputusan (*Reasoning and Decision Making*).

AI bertujuan untuk mengembangkan sistem yang dapat menganalisis informasi, melakukan penalaran, dan membuat keputusan berdasarkan pemahaman konteks dan aturan yang ditentukan. Ini melibatkan penggunaan logika formal, inferensi, dan pemodelan pengetahuan untuk memungkinkan sistem AI untuk mengambil keputusan yang masuk akal.

Komponen utama dalam sistem AI meliputi [14]:

1. Basis Pengetahuan (*Knowledge Base*).

Tempat penyimpanan informasi dan pengetahuan yang digunakan oleh sistem AI. Basis pengetahuan dapat berisi fakta, aturan, konsep, dan model yang relevan dengan domain atau tugas yang sedang dihadapi.

2. Mesin Inferensi (*Inference Engine*).

Mesin inferensi bertanggung jawab untuk melakukan proses logika dan inferensi berdasarkan informasi yang ada dalam basis pengetahuan. Ini melibatkan penerapan aturan, penalaran, dan manipulasi pengetahuan untuk mencapai kesimpulan atau membuat keputusan.

3. Antarmuka Pengguna (*User Interface*).

Antarmuka pengguna memungkinkan interaksi antara pengguna manusia dan sistem AI. Ini dapat berupa antarmuka grafis, perintah suara, atau interaksi berbasis teks yang memungkinkan pengguna untuk memberikan input, menerima output, dan berkomunikasi dengan sistem AI.

4. Algoritma dan Teknik Kecerdasan Buatan (*AI Algorithms and Techniques*).

Ada berbagai algoritma dan teknik yang digunakan dalam AI, termasuk pembelajaran mesin, jaringan saraf tiruan, logika proposisional, pemrosesan bahasa alami, dan banyak lagi. Algoritma ini memungkinkan sistem AI untuk memproses data, mengenali pola, belajar dari pengalaman, dan membuat keputusan.

Ada beberapa jenis AI yang digunakan dalam berbagai konteks dan aplikasi. Dalam hal ini, kita akan menjelajahi beberapa jenis utama AI, termasuk sistem berbasis aturan, sistem pakar, pembelajaran mesin, jaringan saraf tiruan, dan AI yang terkait dengan robotika. Berikut adalah beberapa jenis AI yang umum [14]:

1. Sistem Berbasis Aturan.

Jenis AI ini menggunakan aturan-aturan yang ditentukan sebelumnya untuk menghasilkan output berdasarkan input yang diberikan. Sistem berbasis aturan terdiri dari himpunan aturan dan basis pengetahuan yang menggambarkan hubungan antara *input* dan *output*. Sistem ini bekerja dengan mencocokkan input dengan aturan yang sesuai untuk menghasilkan *output* yang diinginkan.

2. Sistem Pakar.

Sistem pakar adalah jenis AI yang menggabungkan pengetahuan ahli manusia dengan kemampuan komputasi untuk memecahkan masalah yang kompleks dalam domain tertentu. Sistem ini beroperasi dengan mengumpulkan pengetahuan dari para ahli manusia dan memanfaatkannya dalam membuat keputusan atau memberikan nasihat yang sesuai. Sistem pakar digunakan dalam bidang-bidang seperti kedokteran, keuangan, dan teknik.

3. Pembelajaran Mesin.

Pembelajaran mesin adalah cabang AI yang melibatkan pengembangan algoritma dan model statistik yang memungkinkan sistem untuk belajar dari data dan pengalaman. Dalam pembelajaran mesin, sistem menganalisis data untuk mengidentifikasi pola dan membuat prediksi atau keputusan berdasarkan informasi yang diberikan. Algoritma pembelajaran mesin dapat diklasifikasikan menjadi beberapa kategori, termasuk pembelajaran terawasi, pembelajaran tak terawasi, dan pembelajaran penguatan.

4. Jaringan Saraf Tiruan.

Jaringan saraf tiruan (*artificial neural networks*) adalah jenis AI yang terinspirasi oleh struktur dan fungsi jaringan saraf dalam otak manusia. Jaringan saraf tiruan terdiri dari banyak unit pemrosesan sederhana yang disebut neuron buatan, yang saling terhubung dan berinteraksi untuk memproses informasi. Jaringan saraf tiruan dapat digunakan

untuk tugas-tugas seperti pengenalan pola, pengenalan wajah, dan pemrosesan bahasa alami.

5. AI dalam Robotika.

AI juga digunakan dalam pengembangan robotika, di mana sistem AI dikombinasikan dengan fisik robot untuk menghasilkan perilaku yang cerdas dan adaptif. Robotika AI melibatkan penggunaan algoritma pengolahan sensorik, perencanaan gerakan, dan pembelajaran untuk memungkinkan robot berinteraksi dengan lingkungan sekitarnya dan melakukan tugas-tugas tertentu.

Oleh sebab itu, AI merupakan sistem yang dirancang untuk meniru kemampuan kognitif manusia dalam memecahkan masalah kompleks. Dalam bidang pertanian, AI berperan sebagai "otak" yang mampu mempelajari pola defisiensi nutrisi melalui data historis tanpa perlu diprogram secara eksplisit untuk setiap aturan visualnya. Implementasi AI memungkinkan otomatisasi pemantauan kesehatan tanaman yang sebelumnya bergantung sepenuhnya pada tenaga manusia [15][16].

2.4 Machine Learning

Machine Learning (ML) adalah cabang ilmu komputer yang berkembang dari bidang pengenalan pola dan teori pembelajaran komputer dalam kecerdasan buatan. ML mempelajari konstruksi dan analisis algoritma pembelajaran dan prediksi data. Algoritma ML membangun model berdasarkan *input* model untuk mendapatkan sebuah prediksi atau keputusan yang didasarkan dari data daripada mengikuti instruksi program statis secara ketat [17]. ML dikategorikan dalam 3 kategori besar yaitu [18]:

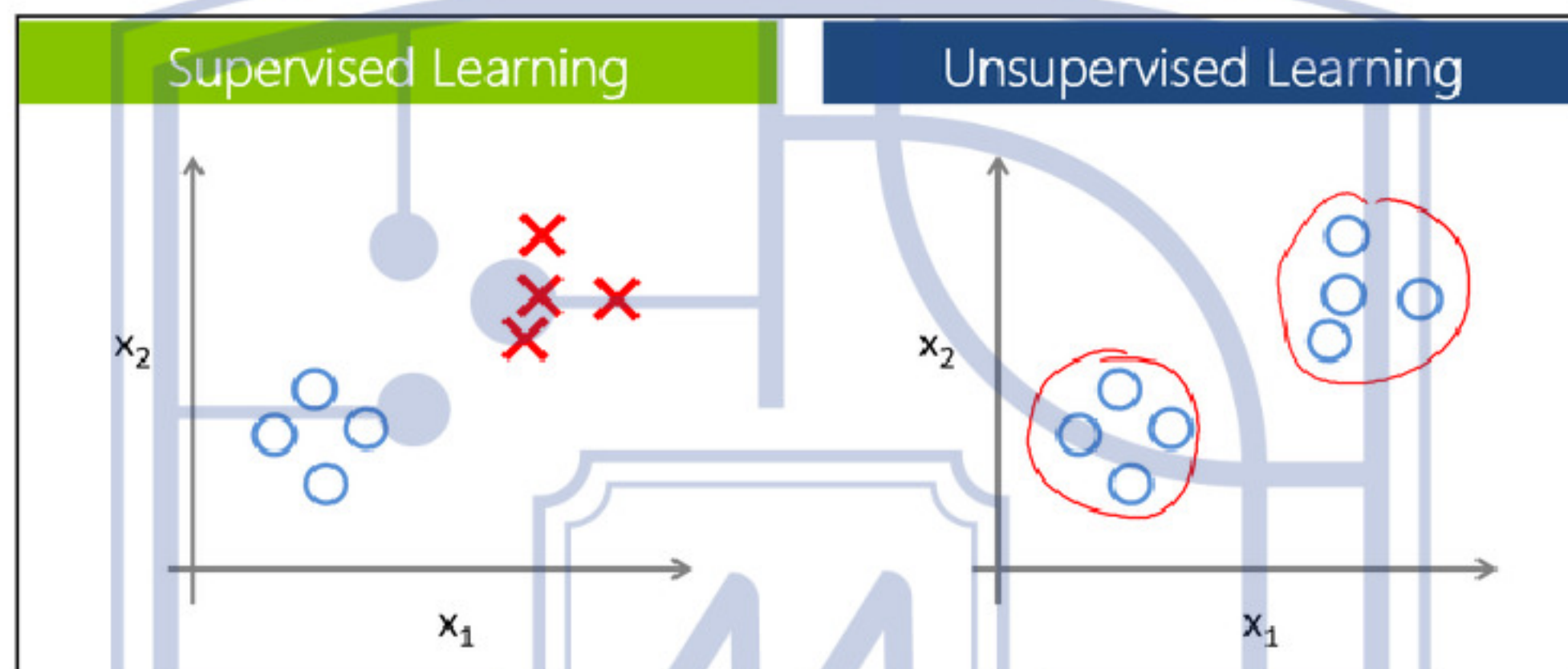
1. *Supervised Learning*.

Model atau algoritma disajikan dengan contoh masukan dan keluaran yang diinginkan kemudian dicari pola dan hubungan antara masukan dan keluaran tersebut. Tujuannya adalah untuk mempelajari aturan umum yang memetakan *input* ke *output*. Proses pelatihan berlanjut hingga model mencapai tingkat akurasi yang diinginkan pada data pelatihan. Beberapa contoh implementasi dari *Supervised Learning* di kehidupan nyata adalah:

- a. Klasifikasi gambar dimana dilakukan pelatihan dengan menggunakan gambar beserta label yang sesuai. Tujuannya adalah agar komputer dapat mengenali objek pada gambar yang baru di masa yang akan datang.
- b. Prediksi dimana komputer dilatih menggunakan data harga saham sebelumnya untuk menghasilkan prediksi harga di masa depan.

2. *Unsupervised Learning*.

Pada *Unsupervised Learning* data tidak diberikan label pada algoritma pembelajaran, dimana algoritma tersebut berusaha menemukan pola sebagai masukannya. Implementasi *Unsupervised Learning* dapat dimanfaatkan dalam mengelompokkan populasi yang memiliki perbedaan. *Unsupervised Learning* dapat mengelompokkan data tersendiri dan menemukan pola tersembunyi yang berada di dalam data. Perbedaan *Supervised Learning* dan *Unsupervised Learning* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Perbedaan *Supervised Learning* dan *Unsupervised Learning* [17]

Seperti yang dapat dilihat pada Gambar 2.2, data dalam *Supervised Learning* diberi label, sedangkan data dalam *Unsupervised Learning* tidak diberi label.

3. *Reinforcement Learning*.

Suatu bidang interdisipliner dalam pembelajaran mesin dan kontrol optimal yang berkaitan dengan bagaimana suatu agen cerdas dapat mengambil aksi di lingkungan yang dinamis dalam rangka untuk memaksimalkan penghargaan kumulatif. Dalam *Reinforcement Learning*, pengembang merancang metode untuk menghargai perilaku yang diinginkan dan menghukum perilaku negatif. Metode ini memberikan nilai positif pada tindakan yang diinginkan untuk mendorong agen dan nilai negatif pada perilaku yang tidak diinginkan. Metode ini memprogram agen untuk mencari imbalan keseluruhan jangka panjang dan maksimum untuk mencapai solusi yang optimal.

Seiring dengan perkembangan teknologi dan meningkatnya kompleksitas data, terutama data citra dan video yang berdimensi tinggi, metode *Machine Learning* konvensional termasuk pendekatan *Supervised Learning* yang digunakan dalam penelitian ini sering kali memerlukan proses ekstraksi fitur secara manual sebelum dilakukan klasifikasi. Untuk mengatasi keterbatasan tersebut, berkembanglah pendekatan yang lebih canggih yaitu *Deep Learning* yang mampu melakukan proses pembelajaran fitur secara

otomatis melalui arsitektur jaringan saraf berlapis atau *deep neural networks*. Pendekatan ini sangat efektif dalam menangani data visual dan menjadi dasar dalam pengembangan model klasifikasi citra daun kelapa sawit pada penelitian ini [4].

2.5 Deep Learning

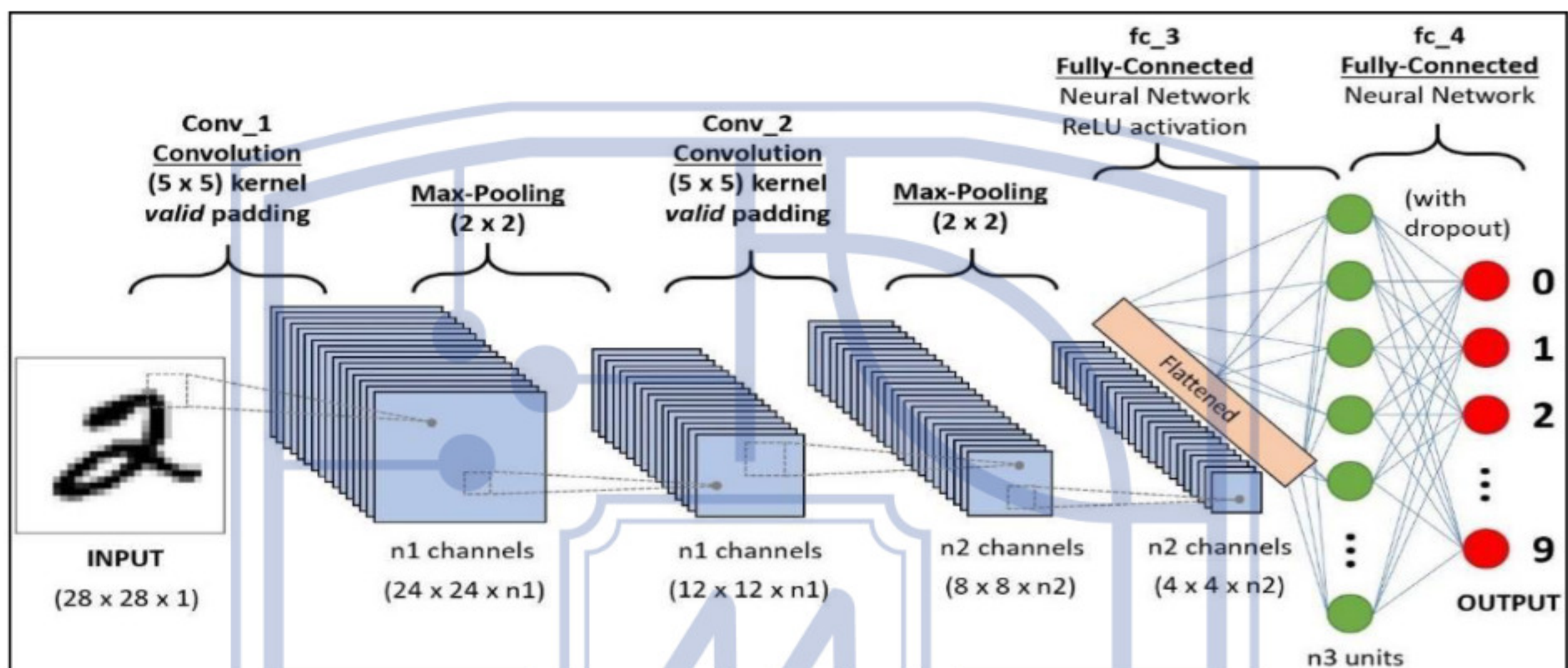
Deep Learning merupakan cabang dari *Machine Learning* yang menggunakan arsitektur jaringan saraf tiruan dengan banyak lapisan tersembunyi untuk mempelajari representasi data secara bertingkat. Pendekatan ini terinspirasi dari mekanisme kerja sistem saraf manusia dalam memproses informasi. Berbeda dengan metode *Machine Learning* konvensional yang umumnya memerlukan proses ekstraksi fitur secara manual, *Deep Learning* mampu melakukan pembelajaran fitur secara otomatis langsung dari data mentah [19]. Pada *Deep Learning*, setiap lapisan dalam jaringan memiliki peran dalam mengekstraksi pola dengan tingkat kompleksitas yang berbeda. Lapisan awal biasanya mengenali fitur dasar seperti tepi, garis, dan perbedaan warna, sedangkan lapisan yang lebih dalam mampu menangkap pola yang lebih kompleks seperti tekstur, bentuk, dan struktur objek. Proses pembelajaran dilakukan melalui penyesuaian bobot pada setiap neuron menggunakan algoritma optimasi untuk meminimalkan kesalahan prediksi [20].

Deep Learning sangat efektif dalam menangani data berdimensi tinggi seperti citra, audio, dan video. Dalam bidang pengolahan citra, arsitektur yang paling banyak digunakan adalah *Convolutional Neural Network* yang dirancang untuk mengekstraksi fitur spasial pada gambar secara efisien. Kemampuan ini menjadikan *Deep Learning* sebagai pendekatan yang sesuai untuk klasifikasi citra, termasuk dalam identifikasi pola defisiensi nutrisi pada daun kelapa sawit berdasarkan karakteristik warna dan teksturnya. Salah satu arsitektur *Deep Learning* yang secara khusus dirancang untuk mengolah data citra adalah *Convolutional Neural Network*. Arsitektur ini memiliki struktur jaringan yang mampu mempertahankan informasi spasial pada gambar melalui operasi konvolusi, *Pooling*, dan lapisan *Fully Connected* sehingga sangat efektif dalam tugas klasifikasi dan pengenalan pola visual. Oleh karena itu, penelitian ini memanfaatkan *Convolutional Neural Network* sebagai metode utama dalam proses identifikasi defisiensi nutrisi pada daun kelapa sawit [21].

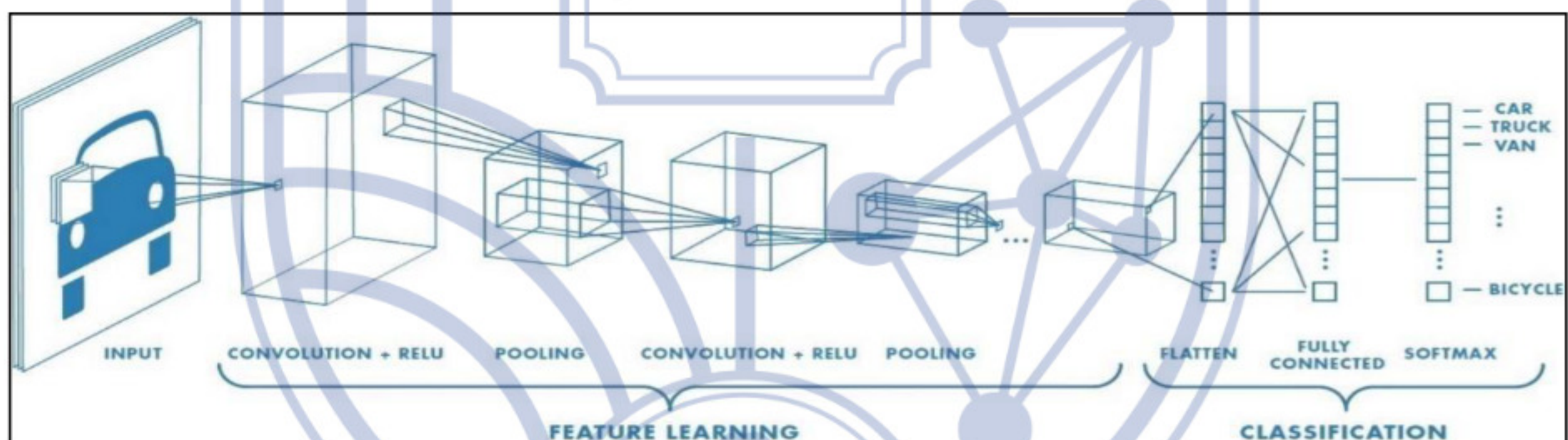
2.6 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah metode *Deep Learning* yang paling efektif untuk klasifikasi citra. CNN merupakan bagian dari *Deep Learning* yang membuat

gambar untuk dikenali dan diklasifikasi. Metode CNN adalah sebuah kelas pada *Neural Network* yang berjalan dalam bidang pemrosesan data-data yang memiliki topologi dalam bentuk *grid*, contohnya yaitu gambar. Metode CNN biasanya dapat digunakan dalam analisa dokumen, klasifikasi gambar dan video, pengenalan wajah, dan lain lain [22]. Secara garis besar, CNN terdiri dari *neuron* yang memiliki *weight*, bias dan *activation function*. Arsitektur *Convolutional Neural Network* dapat dilihat pada Gambar 2.3 [23].



Gambar 2.3 Arsitektur *Convolutional Neural Network* [22]



Gambar 2.4 Ilustrasi Arsitektur *Convolutional Neural Network* [22]

Dari Gambar 2.3 dan Gambar 2.4, dapat dilihat bahwa arsitektur CNN terbagi dalam dua bagian yaitu yang pertama *Feature Extraction layer* dan yang kedua yaitu *Fully Connected Layer*. Dalam *Feature Extraction Layer*, sebuah gambar akan dilakukan *encoding* untuk menjadi *Feature Map* yaitu hasil transformasi dari citra yang berisi informasi penting seperti pola, tepi, tekstur, dan bentuk objek dalam citra. Proses *encoding* menjadi *Feature Map* tersebut disebut juga *Feature Extraction*. *Feature Extraction Layer* ini kemudian dibagi menjadi dua yaitu *Convolution Layer* dan *Pooling Layer*. *Convolution Layer* merupakan bagian penting yang membedakan metode CNN dengan metode *Neural Network* lainnya dan *layer* ini juga merupakan bagian utama dari metode CNN. Pada *Convolution Layer*, citra yang dimasukkan akan diekstraksi ciri-cirinya sehingga dapat

dikenali. Proses konvolusi (*Convolution*) menggunakan fungsi matematika antara matriks citra dengan *filter* atau *kernel* yang terdapat dalam *layer* ini. *Layer* ini mempelajari ciri citra dengan menjaga hubungan antara setiap *pixel*. *Kernel* merupakan operator yang dipakai untuk memperoleh sebuah *array* dari citra. Cara kerja *kernel* yaitu dengan diterapkan dalam seluruh citra. Ukuran matriks pada *kernel* biasanya berukuran 3*3 atau 5*5 yang berisi nilai acak antara -1 dan 1. Pergerakan *kernel* pada citra ditentukan oleh parameter yang disebut *stride*, yaitu jumlah pergeseran kernel dalam satu langkah. Jika *stride* bernilai 1, maka *kernel* akan bergeser satu piksel setiap langkah, sedangkan jika lebih besar, pergeseran akan lebih cepat namun informasi yang diperoleh bisa berkurang. Pada Gambar 2.5 ditunjukkan proses konvolusi pada sebuah gambar [23].



131	162	232	84	91	207
104	-1	0	+1	237	109
243	-2	20	+2	135	26
185	-1	0	+1	61	225
157	124	25	14	102	108
5	155	116	218	232	249

Gambar 2.5 Contoh Operasi Matematis Proses Konvolusi [22]

Hasil konvolusi matriks citra dengan *filter* (*kernel*) juga dapat disebut sebagai *Feature Map*. Selanjutnya, *feature map* akan melalui fungsi aktivasi ReLU (*Rectified Linear Unit*). Secara matematis, ReLU dapat dituliskan sebagai [22]:

$$f(x) = \max(0, x) \dots \dots \dots (1)$$

Fungsi ReLU bekerja dengan mengubah semua nilai negatif menjadi nol dan mempertahankan nilai positif, seperti yang ditampilkan pada Gambar 2.6 [23].

Input			ReLU		
-249	-91	-37	0	0	0
250	-134	101	250	0	101
27	61	-153	27	61	0

Gambar 2.6 Contoh Penerapan Aktivasi ReLU [22]

Setelah melalui proses konvolusi dan aktivasi, *Feature Map* akan diproses pada *Pooling Layer*. *Pooling Layer* adalah lapisan dalam CNN yang berfungsi untuk melakukan reduksi dimensi (*downsampling*) pada *Feature Map* dengan cara merangkum informasi

penting dari area tertentu. Tujuan utama dari *pooling* adalah untuk mengurangi ukuran data, mempercepat proses komputasi, serta mengurangi risiko *overfitting* tanpa menghilangkan informasi penting dari citra. Terdapat beberapa jenis *pooling*, namun yang paling umum digunakan adalah *Max Pooling*. *Max Pooling* adalah metode *pooling* yang bekerja dengan cara mengambil nilai maksimum dari setiap area (*window*) tertentu pada *feature map*, misalnya berukuran 2×2 . Dari setiap area tersebut, hanya satu nilai terbesar yang dipertahankan, sedangkan nilai lainnya diabaikan. *Pooling Layer* dijalankan pada setiap *feature map* secara independen sehingga menghasilkan representasi data yang lebih ringkas namun tetap informatif [22].

Hasil dari proses *pooling* ini kemudian menjadi input untuk tahap selanjutnya, yaitu *Fully Connected Layer*. Pada tahap ini, *Feature Map* yang masih berbentuk multidimensi akan diubah terlebih dahulu menjadi vektor satu dimensi melalui proses yang disebut *flattening*. Proses ini bertujuan agar data dapat diproses oleh jaringan saraf tiruan (*Neural Network*) secara lebih optimal. Selanjutnya, data yang telah di-*flatten* akan diproses pada *Fully Connected Layer* untuk menggabungkan seluruh fitur yang telah diekstraksi sebelumnya menjadi suatu model klasifikasi. Pada tahap akhir dalam *Convolutional Neural Network* (CNN), digunakan fungsi aktivasi untuk mengubah output menjadi nilai yang dapat diinterpretasikan sebagai hasil klasifikasi. Pada tahap akhir, digunakan fungsi aktivasi seperti *Softmax* atau *Sigmoid* untuk menghasilkan *output* berupa kelas tertentu, misalnya klasifikasi wajah manusia, hewan seperti kucing atau anjing, dan sebagainya [22].

1. *Sigmoid*.

Fungsi *Sigmoid* merupakan fungsi aktivasi yang digunakan untuk kasus klasifikasi biner (dua kelas). Fungsi ini mengubah nilai input menjadi rentang antara 0 hingga 1, sehingga dapat diinterpretasikan sebagai probabilitas. Secara matematis, fungsi *Sigmoid* dapat dituliskan sebagai:

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \dots\dots\dots(2)$$

2. *Softmax*.

Fungsi *Softmax* digunakan untuk kasus klasifikasi multi-kelas (lebih dari dua kelas). Fungsi ini mengubah output menjadi bentuk probabilitas untuk setiap kelas, di mana total seluruh probabilitas adalah 1. Secara matematis, fungsi *Softmax* dapat dituliskan sebagai:

$$Softmax_i = \frac{e^{Fully_i}}{(\sum_{j=1}^{kelas} e^{Fully_j})} \dots\dots\dots(3)$$

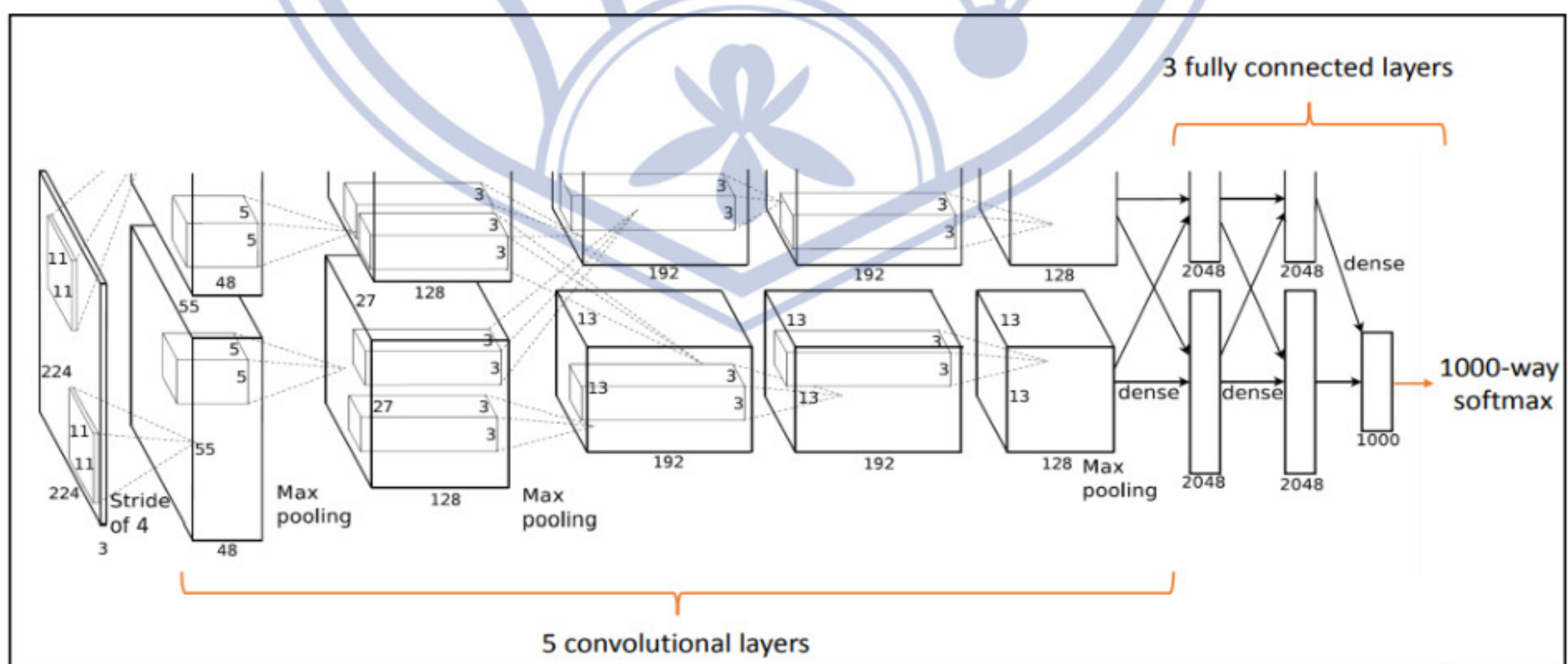
Secara singkat, dengan melalui serangkaian proses, CNN dapat melatih (*training*) dan menguji (*testing*) gambar agar dapat dikenali. Gambar yang dilatih akan diubah menjadi pecahan gambar yang lebih kecil. Kemudian gambar yang lebih kecil tersebut akan melalui *layer* yang akan menyimpan hasil dari gambar yang sudah diperkecil dalam serangkaian *array* untuk mengurangi ukuran gambar dan jumlah parameter ketika perhitungan dilakukan. Kemudian akan dilakukan prediksi untuk mengklasifikasi gambar tersebut. Metode CNN sering dipakai karena metode ini dapat mengekstraksi ciri-ciri penting dari sebuah gambar secara otomatis dan juga merupakan metode yang paling efisien dibandingkan metode *Neural Network* yang lain, utamanya dalam masalah kompleksitas dan juga memori. Namun metode ini juga memiliki kekurangan yaitu memerlukan banyak waktu melatih sebuah model, memerlukan banyak *data training* dan *data testing*, dan juga dapat terjadi *overfitting*. *Overfitting* terjadi ketika model tidak mampu melakukan generalisasi pada gambar karena banyaknya *noise* dalam gambar, model memiliki variasi yang tinggi, *data training* yang terlalu banyak ataupun model yang terlalu kompleks. [23].

2.7 AlexNet

AlexNet merupakan arsitektur *Convolutional Neural Network* (CNN) yang dikembangkan oleh Alex Krizhevsky, Ilya Sutskever, dan Geoffrey Hinton. Arsitektur ini diperkenalkan pada kompetisi *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC 2012) dan berhasil menurunkan tingkat kesalahan klasifikasi secara signifikan dibandingkan pendekatan konvensional pada saat itu. Keberhasilan tersebut menjadi titik balik dalam perkembangan *Deep Learning*, khususnya pada bidang *Computer Vision*, serta mendorong adopsi luas jaringan saraf dalam untuk berbagai permasalahan klasifikasi citra. Secara konseptual, AlexNet dibangun untuk mengatasi keterbatasan metode tradisional yang mengandalkan ekstraksi fitur manual, seperti SIFT (*Scale-Invariant Feature Transform*) dan HOG (*Histogram of Oriented Gradients*), yang kemudian dikombinasikan dengan pengklasifikasi seperti *Support Vector Machine* (SVM). Berbeda dengan pendekatan tersebut, AlexNet mengintegrasikan proses ekstraksi fitur dan klasifikasi ke dalam satu model terpadu yang dilatih secara end-to-end menggunakan algoritma *Backpropagation*. Dengan memanfaatkan *dataset ImageNet* yang berskala besar (lebih dari satu juta citra dan 1000 kelas), model ini mampu mempelajari representasi fitur yang bersifat hierarkis, mulai dari fitur tingkat rendah (tepi, sudut, tekstur) hingga fitur tingkat tinggi (bentuk dan bagian objek) [24].

Arsitektur AlexNet terdiri atas delapan lapisan utama, yaitu lima *Convolutional Layers* dan tiga *Fully Connected Layers*. Citra masukan berukuran 227×227 piksel dengan tiga kanal warna (RGB). Lapisan konvolusi berfungsi untuk mengekstraksi fitur melalui operasi konvolusi menggunakan sejumlah filter atau kernel yang dipelajari selama proses pelatihan. Setiap lapisan konvolusi diikuti oleh fungsi aktivasi non-linear, yaitu *Rectified Linear Unit* (ReLU), yang dirumuskan seperti persamaan 2.1. Penggunaan ReLU terbukti mempercepat konvergensi pelatihan dibandingkan fungsi aktivasi sigmoid atau tanh karena mampu mengurangi permasalahan vanishing gradient. Selain itu, pada beberapa lapisan awal diterapkan *Local Response Normalization* (LRN), yang bertujuan untuk meningkatkan kemampuan generalisasi model dengan melakukan normalisasi respons neuron berdasarkan aktivitas neuron di sekitarnya. Untuk mengurangi dimensi spasial dan kompleksitas komputasi, AlexNet menggunakan *max pooling* setelah beberapa lapisan konvolusi. Operasi ini mengambil nilai maksimum pada setiap wilayah kecil (*receptive field*) sehingga membantu mempertahankan fitur dominan sekaligus mengurangi sensitivitas terhadap translasi kecil pada citra [24].

Bagian akhir arsitektur terdiri dari tiga lapisan *fully connected*. Dua lapisan pertama masing-masing memiliki 4096 neuron, sedangkan lapisan terakhir menggunakan fungsi aktivasi softmax untuk menghasilkan distribusi probabilitas terhadap 1000 kelas pada *dataset* ImageNet. Struktur ini memungkinkan model melakukan klasifikasi multi-kelas secara efektif. Arsitektur lengkap dan alur pemrosesan pada model ini dapat dilihat pada Gambar 2.6, yang menggambarkan susunan berurutan lapisan konvolusi, normalisasi, pooling, serta lapisan *Fully Connected* hingga keluaran klasifikasi [24].



Gambar 2.7 Arsitektur AlexNet [23]

Dalam rangka mengurangi risiko *overfitting*, AlexNet menerapkan beberapa teknik regularisasi. Salah satu teknik utama adalah dropout pada lapisan *Fully Connected*, di mana neuron dinonaktifkan secara acak selama pelatihan untuk mencegah ketergantungan berlebihan antar neuron. Selain itu, dilakukan data augmentation berupa translasi acak, refleksi horizontal, dan variasi intensitas warna untuk meningkatkan keberagaman data latih tanpa perlu menambah *dataset* secara eksplisit. Dari sisi komputasi, pelatihan AlexNet dilakukan menggunakan GPU, yang pada saat itu merupakan pendekatan inovatif dalam pelatihan jaringan saraf berskala besar. Model ini memiliki sekitar 60 juta parameter dan memerlukan sumber daya komputasi yang signifikan. Meskipun demikian, kontribusinya sangat besar dalam membuka jalan bagi pengembangan arsitektur CNN yang lebih dalam dan efisien, seperti *VGGNet*, *GoogLeNet (Inception)*, dan *ResNet*. Secara keseluruhan, AlexNet dapat dipandang sebagai tonggak penting dalam sejarah perkembangan deep learning. Arsitektur ini tidak hanya menunjukkan bahwa jaringan saraf dalam mampu mengungguli metode konvensional dalam klasifikasi citra skala besar, tetapi juga memperkenalkan praktik-praktik penting seperti penggunaan ReLU, dropout, data augmentation, serta pelatihan berbasis GPU yang hingga kini tetap menjadi fondasi dalam penelitian dan implementasi model pembelajaran mendalam [24].

2.8 Transfer Learning

Transfer Learning merupakan metode dalam *Deep Learning* yang memanfaatkan pengetahuan yang telah dipelajari oleh suatu model pada tugas sebelumnya untuk digunakan pada tugas yang berbeda tetapi masih memiliki keterkaitan. Pada metode ini, model tidak dilatih dari awal (*training from scratch*), melainkan menggunakan bobot (*weights*) yang telah diperoleh dari proses pelatihan sebelumnya pada *dataset* berskala besar. Pendekatan ini banyak digunakan pada bidang *Computer Vision* karena mampu meningkatkan performa model serta mengurangi kebutuhan data pelatihan yang besar [25].

Dalam klasifikasi citra, model CNN umumnya dilatih menggunakan *dataset* besar seperti *ImageNet* yang berisi jutaan gambar dari berbagai kategori objek. Selama proses pelatihan tersebut, model mempelajari berbagai fitur visual dasar seperti tepi (*edges*), tekstur, pola, bentuk, dan karakteristik visual lainnya. Fitur-fitur yang telah dipelajari tersebut dapat digunakan kembali untuk menyelesaikan tugas klasifikasi pada *dataset* yang berbeda [25].

Penggunaan *Transfer Learning* memberikan beberapa keuntungan, antara lain mengurangi waktu pelatihan, mempercepat konvergensi model, meningkatkan kemampuan

generalisasi, serta mengatasi keterbatasan jumlah data pelatihan. Oleh karena itu, metode ini sangat sesuai diterapkan pada penelitian yang menggunakan *dataset* citra daun kelapa sawit dengan jumlah data yang relatif terbatas dibandingkan *dataset* skala besar yang digunakan untuk melatih model dari awal [25].

2.9 Fine-Tuning

Fine-Tuning merupakan teknik lanjutan dari *Transfer Learning* yang dilakukan dengan melatih kembali sebagian atau seluruh lapisan model yang telah memiliki bobot hasil pelatihan sebelumnya. Tujuan utama *Fine-Tuning* adalah menyesuaikan representasi fitur yang telah dipelajari model agar lebih sesuai dengan karakteristik *dataset* baru yang digunakan [26].

Pada tahap awal *Transfer Learning*, model biasanya menggunakan bobot pretrained yang diperoleh dari *dataset* besar seperti *ImageNet*. Meskipun fitur-fitur umum yang dipelajari model dapat digunakan pada berbagai tugas klasifikasi citra, karakteristik citra daun kelapa sawit memiliki pola visual yang berbeda dengan objek-objek yang terdapat pada *dataset ImageNet*. Oleh karena itu, diperlukan proses *Fine-Tuning* agar model mampu mempelajari fitur yang lebih spesifik terhadap gejala defisiensi nutrisi pada daun [26].

Proses *Fine-Tuning* dilakukan dengan membuka sebagian lapisan model untuk dilatih kembali menggunakan *dataset* target. Lapisan awal CNN umumnya mempertahankan fitur-fitur dasar seperti garis, sudut, dan tekstur karena fitur tersebut bersifat umum pada berbagai jenis citra. Sementara itu, lapisan yang lebih dalam akan disesuaikan melalui proses pelatihan ulang karena lapisan tersebut bertanggung jawab dalam mengenali fitur yang lebih kompleks dan spesifik terhadap objek yang diklasifikasikan [26].

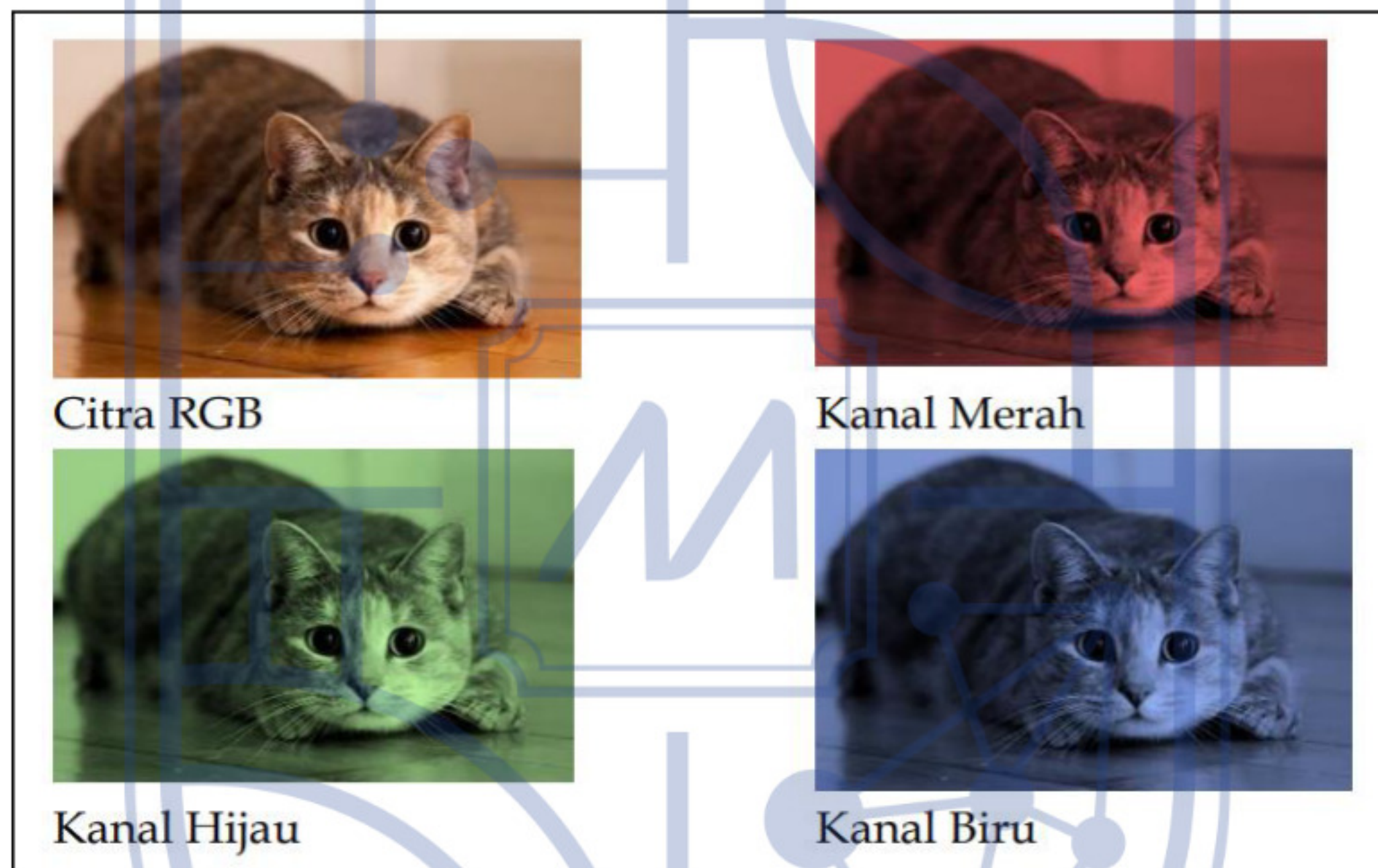
Keunggulan *Fine-Tuning* dibandingkan pelatihan dari awal adalah kebutuhan data yang lebih sedikit, waktu pelatihan yang lebih cepat, serta kemampuan model yang lebih baik dalam beradaptasi dengan *dataset* baru. Selain itu, *Fine-Tuning* dapat membantu meningkatkan akurasi klasifikasi karena model tidak hanya memanfaatkan pengetahuan awal yang telah dimiliki, tetapi juga mempelajari karakteristik khusus dari data yang digunakan dalam penelitian [26].

2.10 Citra Digital

Citra digital atau digital *image* adalah gambar yang terdiri dari elemen gambar, juga dikenal sebagai piksel, masing-masing dengan jumlah representasi numerik yang terbatas dan diskrit untuk intensitas atau tingkat keabuannya [27]. Kombinasi warna pada sebuah piksel dalam citra dapat diuraikan menjadi tiga bagian antara lain [28]:

1. Citra RGB

Citra RGB ini disusun oleh tiga warna utama yaitu warna merah, hijau dan biru. Pada warna ini, masing-masing memiliki nilai intensitas pada pikselnya yang masing-masing memiliki 8-bit atau 2^8 atau 256 variasi warna.



Gambar 2.8 Representasi Citra RGB dan Kanal Warna Penyusunnya

Kombinasi warna dalam sebuah piksel yang mungkin dapat terbentuk pada citra RGB yaitu sebanyak $256^3 = 16.777.216$. Pada Gambar 2.8 dapat dilihat contoh-contoh kombinasi warna yang dapat terbentuk dengan nilai RGB yang berbeda.

Yellow R = 255 G = 255 B = 0	Orange R = 255 G = 102 B = 0	Green R = 0 G = 255 B = 0
Cyan R = 0 G = 255 B = 255	Violet R = 204 G = 102 B = 204	White R = 255 G = 255 B = 255
Black R = 0 G = 0 B = 0	Turquoise R = 102 G = 255 B = 204	Brown R = 153 G = 102 B = 51

Gambar 2.9 Representasi piksel dengan kombinasi warna R, G, dan B [25]

2. Citra *Grayscale*

Citra *grayscale* adalah citra yang dihitung nilai intensitas pada pikselnya berdasarkan tingkat keabuan. Dalam sebuah citra *grayscale* yang memiliki nilai 8-bit, tingkat keabuan tersebut dibagi menjadi 256 tingkat keabuan dimana nilai 255 merupakan putih total dan nilai 0 merupakan hitam total. Sebuah citra RGB dapat dibuat menjadi satu kanal warna dengan cara mengkonversikan citra tersebut menjadi citra *grayscale*. Cara mengkonversi citra RGB menjadi citra *grayscale* yaitu dengan menggunakan formula:

$$Grayscale = 0,2989 * R + 0,5870 * G + 0,1140 * B \dots\dots\dots (4)$$

dimana:

- Grayscale* merupakan nilai intensitas citra *grayscale*.
- R merupakan nilai intensitas merah pada piksel.
- G adalah nilai intensitas hijau pada piksel.
- B adalah nilai intensitas biru pada piksel.

Pada Gambar 2.9 terdapat hasil konversi Gambar 2.7 yang merupakan citra RGB menjadi citra *grayscale*.



Gambar 2.10 Citra Hasil Konversi RGB Menjadi *Grayscale* [25]

3. Citra Biner

Citra biner ini hanya memiliki dua nilai intensitas dan nilai bit sebesar 1. Oleh karena itu citra ini memiliki dua nilai intensitas yaitu warna hitam ataupun 0 dan warna putih ataupun 1. Citra biner didapatkan melalui konversi citra *grayscale* yang dinamakan proses *thresholding*. Dalam proses ini, akan diambil sebuah nilai *threshold* yang akan dijadikan sebagai acuan untuk proses konversi. Jika nilai intensitas sebuah piksel sama dengan atau lebih besar dari nilai *threshold*, maka nilai intensitas piksel tersebut akan dikonversi menjadi 1. Dan sebaliknya jika nilai intensitas piksel kurang dari nilai *threshold*, maka nilai intensitas piksel tersebut akan dikonversi menjadi 0. Sebagai contoh, 128 akan ditetapkan sebagai nilai *threshold*, jika piksel memiliki nilai intensitas sama dengan atau lebih dari 128, maka akan dikonversikan menjadi 1 atau putih. Dan sebaliknya jika piksel tersebut memiliki nilai intensitas kurang dari 128, maka akan dikonversikan menjadi 0 atau hitam. Pada Gambar 2.10 akan ditunjukkan Gambar 2.9 yang telah dikonversikan menjadi citra biner.



Gambar 2.11 Citra Hasil Konversi RGB Menjadi Biner [25]

Citra digital biasanya merupakan data mentah yang kemudian dianalisis dan disempurnakan untuk mengekstraksi informasi yang bermakna atau meningkatkan kualitas

visual. Melalui pengolahan citra digital, berbagai penyesuaian dapat dilakukan, seperti peningkatan kecerahan, kontras, atau ketajaman gambar, atau menyoroti fitur-fitur tertentu dalam citra. Hal ini membuka peluang untuk berbagai aplikasi, seperti pengenalan objek, pengenalan wajah, hingga diagnosa medis [28].

2.11 Pengolahan Citra Digital

Pengolahan citra digital (*digital image processing*) mempelajari cara sebuah citra dibentuk, dianalisis, dan diolah sedemikian rupa agar menghasilkan data-data atau informasi yang dapat dipahami secara manusiawi. Salah satu kegunaan pengolahan citra yaitu untuk memperbaiki kesalahan sinyal pada gambar atau piksel yang rusak yang disebabkan oleh adanya gangguan pada saat pembuatan gambar tersebut. Pengolahan citra digunakan untuk mempermudah manusia untuk melihat suatu gambar yang kabur agar terlihat lebih manusiawi dan gampang dilihat. Pengolahan citra dikelompokkan menjadi 3 bagian yaitu: *Image Enhancement*, *Image Restoration*, dan *Image Compression* [28]. *Image Enhancement* atau peningkatan kualitas gambar merupakan tahap dimana kontras pada gambar diperbaiki agar tingkat keabuaannya berada pada titik yang normal sehingga lebih gampang untuk diinterpretasikan manusia [29]. Dalam proses *image enhancement* terdapat beberapa teknik yaitu [28]:

1. Operasi titik

Operasi titik dalam peningkatan kualitas gambar dilakukan dengan cara menggantikan data-data dari citra dengan data-data yang baru secara manual agar sesuai dengan hasil yang diinginkan.

2. Operasi spasial

Operasi spasial dalam peningkatan kualitas gambar dilakukan dengan cara menggantikan nilai sebuah piksel yang akan diganti berdasarkan nilai-nilai median piksel lain yang disekitarnya sehingga diperoleh sebuah nilai piksel baru.

3. Operasi transformasi

Operasi transformasi dalam peningkatan kualitas gambar digunakan dengan cara memindahkan suatu citra asal ke tempat yang baru sesuai dengan keinginan.

Pada saat ini, teknologi pengolahan citra digital banyak digunakan dan dimanfaatkan dalam AI ataupun dalam kehidupan sehari-hari, sebagai contoh beberapa aplikasi pengolahan citra digital yaitu [28]:

1. Aplikasi deteksi, klasifikasi dan identifikasi objek.

Aplikasi ini digunakan untuk dalam pendeteksi, pengklasifikasi dan pengidentifikasi objek, seperti aplikasi yang dipakai untuk mendeteksi objek seperti mobil atau sepeda .

2. *Aplikasi pengenalan pola.*

Aplikasi ini merupakan aplikasi yang digunakan untuk mengklasifikasi objek berdasarkan ciri-ciri yang dimiliki citra ke dalam beberapa kategori atau kelas.

3. *Machine vision*

Aplikasi ini mengembangkan secara lanjut aplikasi pengenalan pola. Mesin ini menggunakan kamera untuk mengambil gambar dari sebuah objek dengan tujuan untuk menganalisa dan mendeskripsikan objek yang diambil kamera.

4. *Character recognition (OCR)*

OCR juga mengembangkan aplikasi pengenalan pola yang digunakan untuk mengenali teks atau karakter dari suatu citra. OCR banyak dimanfaatkan dalam bidang informasi dan otomatisasi dengan menggunakan *scanner* ,sumber cahaya ,dan detektor.

5. *Computer aided diagnosis*

Computer aided diagnosis biasanya digunakan untuk mendiagnosa atau mengambil keputusan untuk membantu dokter

6. *Speech recognition*

Speech recognition atau pengenalan pola suara manusia digunakan agar komputer dapat mengenali bahasa manusia. Aplikasi ini biasa diterapkan dalam industri untuk meningkatkan efisiensi.

7. *Face recognition*

Face recognition atau pengenalan wajah merupakan aplikasi yang digunakan untuk mengenali wajah manusia. Aplikasi ini biasanya digunakan dalam bidang keamanan.

8. *Biometrics*

Biometrics dipakai untuk mengidentifikasi sebuah karakteristik unik yang dimiliki setiap individual.

9. *Image Database Retrieval*

Image database retrieval merupakan sistem komputer untuk menelusuri, mencari, dan mengambil gambar dari *database* besar gambar digital. Metode yang paling umum digunakan adalah menambahkan metadata ke teks, kata kunci, dan sebagainya.

10. *Data Mining*

Data mining adalah proses yang digunakan oleh perusahaan untuk mengubah data mentah menjadi informasi yang berguna. Dengan menggunakan perangkat lunak untuk mencari pola dalam kumpulan data yang besar, bisnis dapat mempelajari lebih lanjut

tentang pelanggan mereka untuk mengembangkan strategi pemasaran yang lebih efektif, meningkatkan penjualan, dan menurunkan biaya.

Dengan kemampuan untuk memproses dan menganalisis fitur dalam citra, diperlukan teknik pendukung yang dapat meningkatkan variasi data pelatihan agar model mampu belajar secara lebih optimal. Salah satu teknik yang digunakan adalah data augmentation, yaitu proses memperbanyak data secara artifisial melalui transformasi tertentu seperti rotasi, flipping, scaling, dan perubahan tingkat kecerahan pada citra, sehingga model dapat lebih robust terhadap variasi kondisi data dan meningkatkan kemampuan generalisasi pada proses klasifikasi [28].

2.12 Augmentasi Data

Augmentasi data (*data augmentation*) merupakan teknik yang digunakan untuk memperbanyak variasi data pelatihan secara artifisial tanpa menambah data baru secara langsung. Teknik ini dilakukan dengan berbagai transformasi pada data citra yang sudah ada sehingga model dapat belajar dari variasi bentuk data yang lebih beragam. Tujuan utama dari augmentasi data adalah untuk meningkatkan kemampuan generalisasi model serta mengurangi risiko *overfitting* pada proses pelatihan, terutama ketika jumlah data terbatas dan tidak seimbang [30].

Dalam penerapannya pada citra digital, augmentasi data dilakukan melalui beberapa teknik transformasi, antara lain [30]:

1. *Rotation* (rotasi), yaitu memutar gambar pada sudut tertentu untuk menghasilkan variasi orientasi objek.
2. *Flipping* (pembalikan), yaitu membalik gambar secara horizontal atau vertikal.
3. *Scaling* (penskalaan), yaitu memperbesar atau memperkecil ukuran objek dalam citra.
4. *Translation* (pergeseran), yaitu menggeser posisi objek ke arah tertentu tanpa mengubah bentuk utama.
5. *Cropping*, yaitu mengambil sebagian area dari citra untuk menghasilkan fokus pada bagian tertentu.
6. Perubahan intensitas warna, seperti *brightness*, *contrast*, dan *saturation* untuk mensimulasikan kondisi pencahayaan yang berbeda.

Teknik-teknik tersebut bertujuan untuk mensimulasikan kondisi nyata yang mungkin terjadi di lapangan, sehingga model tidak hanya mengenali pola dalam kondisi ideal, tetapi juga pada kondisi yang bervariasi. Hal ini sangat penting terutama pada data citra daun yang dapat memiliki perbedaan bentuk, posisi, maupun pencahayaan. Dengan demikian,

pengolahan dan analisis citra tersebut menjadi bagian penting dalam bidang *Computer Vision* yang berfokus pada pemahaman dan ekstraksi informasi dari data visual [30].

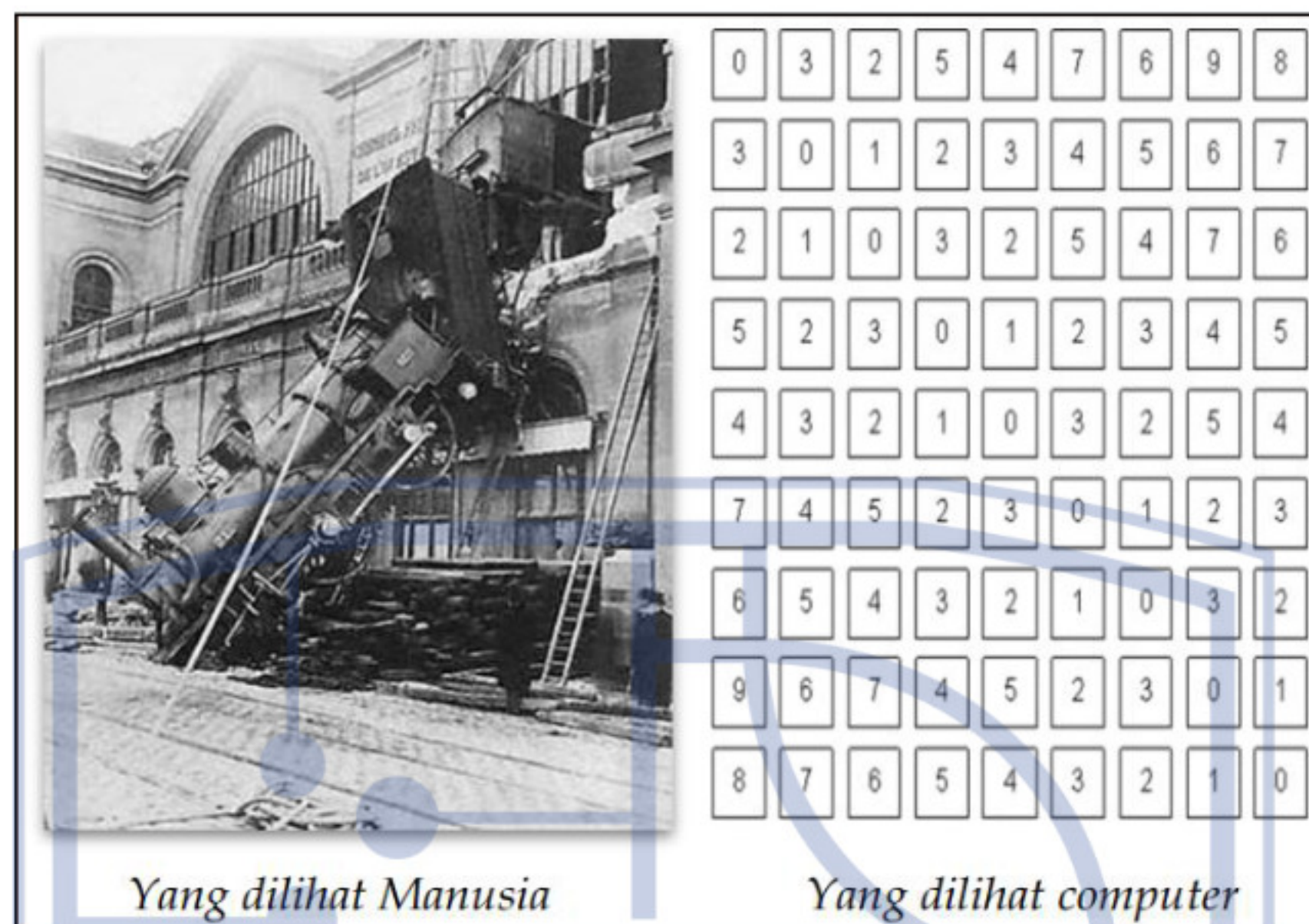
2.13 *Computer Vision*

Computer Vision adalah bidang yang memungkinkan komputer "melihat" dan menginterpretasikan konten visual. Kesimpulannya, teknologi ini mengubah piksel warna (RGB) pada daun kelapa sawit menjadi data numerik yang dapat diproses oleh algoritma untuk menentukan kondisi kesehatan tanaman berdasarkan degradasi warna dan tekstur daun [3]. Dari perspektif teknik, *computer vision* dapat digunakan untuk melakukan tugas-tugas yang dilakukan oleh penglihatan manusia secara otomatis. Tugas *Computer Vision* yaitu meliputi metode untuk memperoleh, memproses, menganalisis dan memahami gambar digital, serta ekstraksi data dimensi tinggi dari dunia nyata untuk menghasilkan informasi numerik atau simbolis, misalnya, dalam bentuk keputusan [28].

Sensor ini melampaui mata manusia karena mereka dapat melihat hal-hal yang tidak terlihat oleh manusia tetapi terbatas dalam aspek lain. Komputer dapat diprogram untuk memahami cukup banyak dari gambar yang diambil, seringkali hal-hal yang tidak dilihat atau dipedulikan manusia. Namun, komputer masih jauh dari kemampuan otak manusia. Kesenjangan menjadi semakin kecil seiring kemajuan teknologi dan sains. Aplikasi *Computer Vision* sangat luas dan mencakup semua aspek kegiatan sosial dan ekonomi, pengenalan pola, penilaian kualitas tidak rusak untuk berbagai jenis produk, deteksi objek 3D, pertanian, cuaca, perawatan kesehatan, identifikasi, pengawasan, keselamatan publik, pembantu mengemudi, kendaraan *self-driving*, dll. Komputer sudah dapat memperkirakan cacat dan sifat produk/objek seperti warna, bentuk, ukuran, permukaan, kontaminasi, dan mereka dapat mengidentifikasi orang dan melacaknya. Batas ke mana visi komputer dapat berkembang tidak dapat ditentukan, tetapi perkembangan saat ini menunjukkan bahwasannya batasannya hanya diberikan oleh imajinasi manusia dan segera komputer bahkan mungkin melihat lebih baik daripada mata manusia [28].

Transformasi gambar secara visual biasanya membutuhkan input yang memiliki ciri-ciri gambar agar dapat berinteraksi dengan proses pemikiran sistem agar tindakan yang sesuai dapat dieksekusi. Transformasi gambar ini melibatkan data gambar yang diperoleh melalui model yang dibangun dengan menggunakan campuran dari ilmu-ilmu lain, contohnya yaitu : geometri, statistika, dan juga algoritma komputer. *Computer Vision* terbagi menjadi beberapa sub bagian yaitu deteksi peristiwa, perkiraan gambar 3D,

perkiraan gerakan, pengenalan objek dan juga pemulihan gambar. Gambar 2.11 merupakan ilustrasi cara komputer melihat sebuah citra [28].



Gambar 2.12 Ilustrasi Cara Komputer Melihat Citra [25]

Salah satu bidang yang erat kaitannya dengan *Computer Vision* adalah kecerdasan buatan, yang merupakan upaya untuk menciptakan mesin yang dapat meniru fungsi kognitif manusia, termasuk dalam memahami dan memproses informasi visual [28].

2.14 Ekstraksi Fitur pada Daun

Ekstraksi fitur merupakan salah satu tahapan penting dalam proses pengolahan citra digital yang bertujuan untuk memperoleh informasi penting dari sebuah citra sehingga dapat digunakan dalam proses analisis, pengenalan pola, maupun klasifikasi. Pada tahap ini, data citra yang awalnya berupa kumpulan piksel diubah menjadi sekumpulan fitur yang mampu merepresentasikan karakteristik utama dari objek yang dianalisis. Dengan melakukan ekstraksi fitur, dimensi data citra dapat disederhanakan tanpa menghilangkan informasi penting yang diperlukan dalam proses pengolahan selanjutnya. Dalam penelitian yang berkaitan dengan analisis kondisi daun tanaman, ekstraksi fitur dilakukan untuk mengidentifikasi karakteristik visual yang muncul akibat perubahan fisiologis pada daun. Perubahan tersebut dapat disebabkan oleh berbagai faktor, seperti kekurangan unsur hara, serangan penyakit, maupun kondisi lingkungan. Oleh karena itu, ekstraksi fitur menjadi tahap penting untuk memperoleh parameter yang dapat digunakan oleh sistem komputer dalam membedakan kondisi daun yang sehat dan daun yang mengalami gangguan [31].

Secara umum, fitur yang sering digunakan dalam analisis citra daun meliputi fitur warna, fitur tekstur, dan fitur bentuk. Ketiga jenis fitur tersebut dapat memberikan

informasi yang saling melengkapi dalam menggambarkan karakteristik objek pada citra daun [31].

1. Fitur Warna (*Color Feature*)

Fitur warna merupakan salah satu fitur yang paling banyak digunakan dalam analisis citra daun karena perubahan warna daun sering menjadi indikator utama kondisi kesehatan tanaman. Daun yang mengalami defisiensi unsur hara umumnya menunjukkan perubahan warna tertentu, seperti hijau pucat, kekuningan, atau kecokelatan. Dengan mengekstraksi informasi warna dari citra daun, sistem dapat mendeteksi pola perubahan warna tersebut secara otomatis. Ekstraksi fitur warna biasanya dilakukan dengan menggunakan model ruang warna tertentu, seperti RGB (*Red, Green, Blue*) atau HSV (*Hue, Saturation, Value*). Model RGB merepresentasikan warna sebagai kombinasi tiga komponen utama, yaitu merah, hijau, dan biru. Sementara itu, model HSV memisahkan komponen warna menjadi hue (jenis warna), saturation (tingkat kejenuhan warna), dan value (tingkat kecerahan). Penggunaan ruang warna HSV sering dianggap lebih efektif dalam analisis citra karena lebih mendekati cara manusia dalam memersepsikan warna. Informasi warna dari citra biasanya diekstraksi dalam bentuk nilai rata-rata warna, histogram warna, maupun distribusi intensitas warna. Nilai-nilai tersebut kemudian digunakan sebagai fitur numerik yang dapat diproses oleh algoritma klasifikasi.

2. Fitur Tekstur (*Texture Feature*)

Selain warna, fitur tekstur juga memiliki peran penting dalam analisis citra daun. Tekstur menggambarkan pola distribusi intensitas piksel pada suatu citra yang mencerminkan karakteristik permukaan objek. Pada citra daun, tekstur dapat menunjukkan adanya pola bercak, kekasaran permukaan, atau perubahan struktur jaringan daun yang sering muncul akibat penyakit tanaman atau defisiensi unsur hara. Salah satu metode yang banyak digunakan untuk mengekstraksi fitur tekstur adalah *Gray Level Co-occurrence Matrix* (GLCM). Metode ini diperkenalkan oleh Haralick dan digunakan untuk menganalisis hubungan spasial antara piksel pada citra. GLCM bekerja dengan menghitung seberapa sering pasangan piksel dengan nilai intensitas tertentu muncul pada jarak dan arah tertentu dalam citra. Dari matriks tersebut kemudian dapat dihitung beberapa parameter tekstur, seperti *contrast*, *correlation*, *energy*, dan *homogeneity*, yang digunakan untuk menggambarkan karakteristik tekstur citra. Parameter *contrast* menggambarkan tingkat perbedaan intensitas antara piksel yang berdekatan, sedangkan *correlation* menunjukkan tingkat keterkaitan antara piksel

dalam citra. Parameter *energy* menunjukkan tingkat keseragaman distribusi intensitas piksel, sementara *homogeneity* menggambarkan kedekatan distribusi elemen matriks terhadap diagonal utama. Kombinasi parameter tersebut dapat memberikan gambaran yang cukup baik mengenai pola tekstur pada citra daun.

3. Fitur Bentuk (*Shape Feature*)

Fitur bentuk digunakan untuk menggambarkan karakteristik geometris objek pada citra. Dalam analisis citra daun, fitur bentuk dapat digunakan untuk mengetahui ukuran dan struktur daun yang mungkin mengalami perubahan akibat gangguan pertumbuhan. Defisiensi unsur hara tertentu dapat menyebabkan daun mengalami deformasi, perubahan ukuran, atau pertumbuhan yang tidak normal. Beberapa parameter yang umum digunakan dalam ekstraksi fitur bentuk antara lain luas area (*area*), keliling objek (*perimeter*), rasio aspek (*aspect ratio*), dan tingkat kebulatan objek (*roundness*). Parameter tersebut diperoleh melalui proses segmentasi citra yang bertujuan untuk memisahkan objek daun dari latar belakang sehingga bentuk daun dapat dianalisis secara lebih akurat. Dengan menggabungkan berbagai jenis fitur seperti warna, tekstur, dan bentuk, sistem pengolahan citra dapat memperoleh informasi yang lebih lengkap mengenai karakteristik daun tanaman. Informasi tersebut kemudian digunakan sebagai masukan dalam proses klasifikasi menggunakan algoritma tertentu, sehingga sistem dapat mengidentifikasi kondisi daun tanaman secara otomatis dan lebih akurat.

2.15 Confusion Matrix

Confusion Matrix adalah salah satu cara yang sering digunakan pada proses evaluasi model data mining klasifikasi dengan memprediksi kebenaran objek. *Confusion Matrix* digambarkan dengan tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan. Berikut ini akan ditunjukkan tabel *Confusion Matrix* pada Tabel 2.1 [32].

Tabel 2.1 *Confusion Matrix* Untuk lasifikasi 2 Kelas

<i>Correct Classification</i>	Classified as	
	<i>Predicted “+”</i>	<i>Predicted “-“</i>
<i>Actual “+”</i>	<i>True Positives</i>	<i>False Negatives</i>
<i>Actual “-“</i>	<i>False Positives</i>	<i>True Negatives</i>

Berdasarkan tabel *Confusion Matrix* di atas [32]:

1. *True Positives* (TP) adalah jumlah *record* data positif yang diklasifikasikan sebagai nilai positif.
2. *False Positives* (FP) adalah jumlah *record* data negatif yang diklasifikasikan sebagai nilai positif.
3. *False Negatives* (FN) adalah jumlah *record* data positif yang diklasifikasikan sebagai nilai negatif.
4. *True Negatives* (TN) adalah jumlah *record* data negatif yang diklasifikasikan sebagai nilai *negative*.

Nilai yang dihasilkan melalui metode *Confusion Matrix* adalah berupa evaluasi sebagai berikut [32]:

1. *Accuracy*, persentase jumlah *record* data yang diklasifikasikan (prediksi) secara benar oleh algoritma.
Rumus: $(TP + TN) / \text{Total data} = \text{Accuracy}$(5)
2. *Precision*, persentase rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif.
Rumus: $(TP) / (TP + FP) = \text{Precision}$(6)
3. *Recall*, persentase rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.
Rumus: $(TP) / (TP + FN) = \text{Recall}$(7)
4. *Misclassification (Error) Rate*, persentase jumlah *record* data yang diklasifikasikan (prediksi secara salah oleh algoritma).
Rumus : $(FP + FN) / \text{Total data} = \text{Misclassification Rate}$(8)

2.16 Kelapa Sawit dan Identifikasi Defisiensi

Kelapa sawit (*Elaeis guineensis*) merupakan tanaman perkebunan yang membutuhkan unsur hara makro dan mikro dalam jumlah cukup untuk mendukung pertumbuhan dan produksi. Ketidakseimbangan atau kekurangan unsur hara akan menimbulkan gangguan fisiologis yang dapat diamati melalui perubahan visual pada daun. Identifikasi defisiensi nutrisi pada penelitian ini difokuskan pada lima unsur hara, yaitu Nitrogen (N), Kalium (K), Magnesium (Mg), dan Boron (B) [1].

1. Nitrogen berperan dalam pembentukan klorofil dan pertumbuhan vegetatif tanaman. Defisiensi Nitrogen ditandai dengan klorosis merata pada daun, yaitu perubahan warna hijau menjadi hijau pucat atau kuning secara menyeluruh. Gejala umumnya muncul pada daun tua karena Nitrogen bersifat mobil di dalam jaringan tanaman.



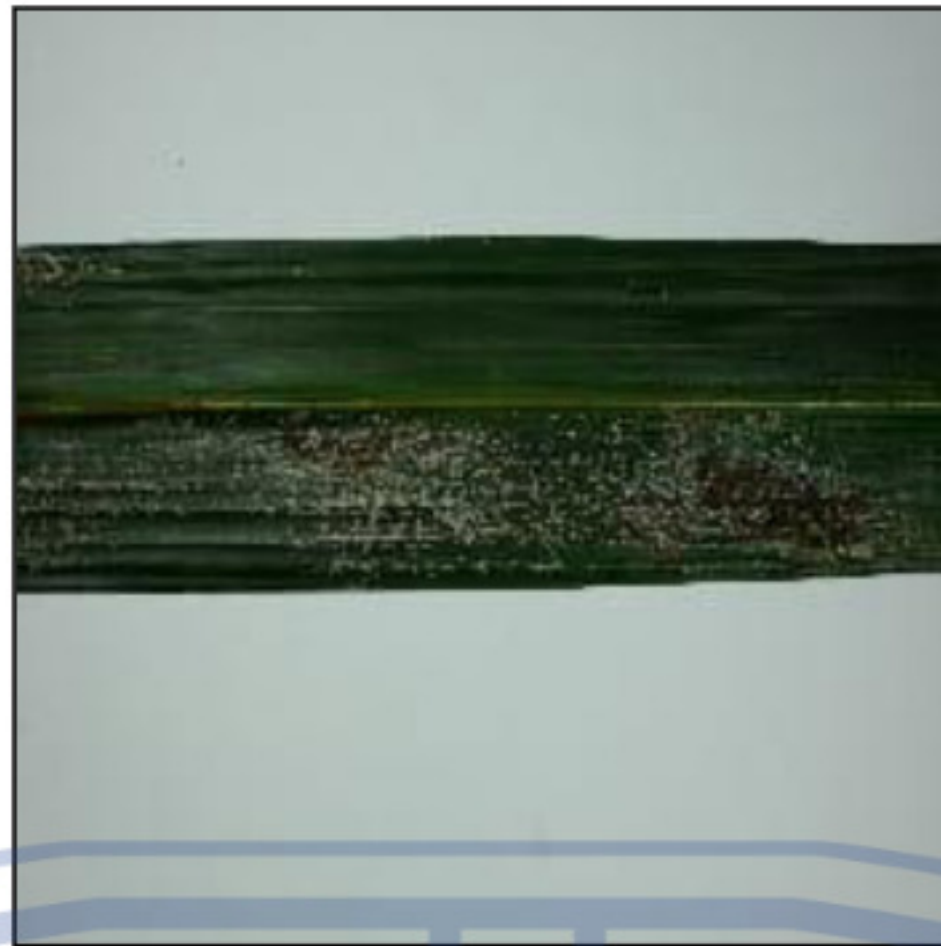
Gambar 2.13 Contoh Gejala Defisiensi Nitrogen pada Daun Kelapa Sawit

2. Kalium berfungsi dalam regulasi keseimbangan air dan aktivitas enzim. Defisiensi Kalium menyebabkan munculnya bercak berwarna oranye atau cokelat pada daun yang kemudian berkembang menjadi nekrosis, terutama pada bagian tepi daun. Pola bercak ini menjadi indikator visual utama kekurangan Kalium.



Gambar 2.14 Contoh Gejala Defisiensi Kalium pada Daun Kelapa Sawit

3. Magnesium merupakan komponen inti klorofil dan berperan dalam proses fotosintesis. Defisiensi Magnesium ditandai dengan klorosis interveinal, yaitu perubahan warna kuning di antara tulang daun sementara tulang daun tetap hijau. Gejala biasanya terlihat jelas pada daun tua dan pada bagian yang terpapar sinar matahari.



Gambar 2.15 Contoh Gejala Defisiensi Magnesium pada Daun Kelapa Sawit

4. Boron termasuk unsur mikro yang berperan dalam proses pembelahan sel serta pertumbuhan jaringan meristem pada tanaman. Defisiensi boron pada kelapa sawit dapat menyebabkan pertumbuhan daun tidak normal, daun menjadi kaku atau keriting, serta muncul deformasi pada pucuk tanaman. Selain itu, pada beberapa kasus daun dapat mengalami perubahan warna menjadi hijau pucat hingga kekuningan, dan helaian daun tampak tidak berkembang secara sempurna.



Gambar 2.16 Contoh Gejala Defisiensi Boron pada Daun Kelapa Sawit

Dalam konteks pengolahan citra digital, keempat jenis defisiensi tersebut memiliki karakteristik warna, tekstur, dan pola distribusi gejala yang berbeda. Pendekatan kecerdasan buatan berbasis *Convolutional Neural Network* dapat digunakan untuk mengidentifikasi perbedaan pola visual tersebut secara otomatis. Model melakukan ekstraksi fitur dari citra daun untuk membedakan klorosis merata akibat Nitrogen, bercak oranye dan nekrosis akibat Kalium, klorosis interveinal akibat Magnesium, serta deformasi daun akibat Boron.

2.17 Rekomendasi Pemupukan

Rekomendasi pemupukan merupakan suatu pendekatan dalam menentukan jenis dan dosis pupuk yang tepat berdasarkan kondisi tanaman, khususnya gejala yang muncul pada daun. Gejala pada daun sering digunakan sebagai indikator awal untuk mengetahui kekurangan atau kelebihan unsur hara pada tanaman, sehingga dapat dilakukan tindakan pemupukan yang sesuai. Tanaman membutuhkan unsur hara makro dan mikro untuk menunjang pertumbuhan dan perkembangan. Berikut beberapa gejala kekurangan unsur hara yang umum terjadi yang disesuaikan pada penelitian ini serta rekomendasi pemupukannya [33]:

1. Kekurangan Nitrogen (N).

Pemberian pupuk Urea atau ZA (*Zwavelzure Ammoniak*) dengan teknik pemupukan *circle* (cangkul-sebar-tutup). Selain itu, perlu dilakukan pengendalian tanaman kompetitor (gulma) dan perbaikan drainase lahan

2. Kekurangan Kalium (K).

Pemberian pupuk KCl atau MOP (*Muriate of Potash*). Penting untuk memperhatikan pH tanah dan teknik aplikasi karena unsur ini sangat mudah tercuci oleh air (*leaching*).

3. Kekurangan Magnesium (Mg).

Pemberian pupuk Kieserit (MgSO_4) atau Dolomit ($\text{MgCa}(\text{CO}_3)_2$). Pengaplikasiannya juga harus mempertimbangkan keseimbangan (rasio) antara kation K, Ca, dan Mg.

4. Kekurangan Boron (B).

Berikan pupuk borat seperti Borax atau pupuk yang mengandung unsur boron sesuai dosis rekomendasi. Aplikasi harus dilakukan secara hati-hati karena kebutuhan boron relatif kecil

2.18 Penelitian Terdahulu

Berikut ini diuraikan beberapa penelitian terdahulu yang berkaitan dengan penelitian yang dilakukan pada Tabel 2.2.

Tabel 2.2 Ringkasan Penelitian Terdahulu

No.	Nama Peneliti	Tahun	Judul Penelitian	Hasil Penelitian
1	Itin Riati, Yuhandri, dan Gunadi W Nurcahyo	2024	Penerapan <i>Convolutional Neural Network</i> untuk Mengidentifikasi Penyakit Tanaman	Hasil penelitian menunjukkan bahwa CNN dapat secara efektif mendeteksi berbagai jenis hama dan penyakit pada tanaman kelapa sawit dengan akurasi lebih dari 90%. Penelitian ini

No.	Nama Peneliti	Tahun	Judul Penelitian	Hasil Penelitian
			Kelapa Sawit	menunjukkan potensi besar dalam pertanian modern dan dapat memfasilitasi praktik pertanian yang lebih berkelanjutan dan efisien [34].
2	Arif Fadllullah dan Sri Erdina	2025	Identifikasi Penyakit Bercak Daun Kelapa Sawit Menggunakan Algoritma CNN dengan Arsitektur VGG19 Berbasis Citra Digital	Hasil pelatihan model sebanyak 200 epoch menunjukkan akurasi sebesar 0,93 pada data latih dan 0,98 pada data validasi. Kemudian evaluasi model terhadap data uji menghasilkan nilai <i>precision</i> , <i>recall</i> , dan <i>F1-score</i> masing-masing sebesar 0,94, 0,81, dan 0,87 untuk kelas “Karat Daun”, 0,84, 0,95, dan 0,89 untuk kelas “Daun Sehat”, serta 1,00 untuk kelas “Curvularia”. Hasil pengujian menunjukkan performa yang konsisten, sehingga sistem usulan dinilai efektif dalam melakukan klasifikasi penyakit bercak daun kelapa sawit. Sistem usulan yang dikembangkan berpotensi digunakan sebagai alat bantu deteksi dini penyakit bercak daun guna mendukung peningkatan produktivitas tanaman kelapa sawit [35].
3	Tobyanto Putra Mandiri, Benedictus Budhi Dharmawan, Ferreyla Setara Ibn, dan Meida Cahyo Untoro	2025	Identifikasi Penyakit pada Daun Kelapa Sawit dengan Pendekatan CNN AlexNet	Hasil penelitian menunjukkan bahwa model <i>Convolutional Neural Network</i> (CNN) dengan arsitektur AlexNet mampu mengidentifikasi penyakit pada daun kelapa sawit menggunakan <i>dataset</i> sebanyak 263 citra yang terdiri dari beberapa kelas penyakit. Model yang dilatih memperoleh akurasi pelatihan sebesar 97%, akurasi validasi sebesar 80%, dan akurasi akhir sebesar 79%, sehingga menunjukkan bahwa model memiliki kinerja yang cukup baik dalam proses klasifikasi penyakit daun kelapa sawit [7].
4	Nanang Riyadi, Erni	2026	Penerapan <i>Deep Learning</i> Berbasis	Hasil evaluasi menunjukkan akurasi rata-rata 88% dengan nilai

No.	Nama Peneliti	Tahun	Judul Penelitian	Hasil Penelitian
	Rouza, Fajar Maftukhan, Sukma Zahara, Theresia Panjaitan, Citra Dea Wulandari, dan Rendi Saputra		CNN untuk Klasifikasi Otomatis Penyakit Daun Kelapa Sawit	<i>precision</i> , <i>recall</i> , dan <i>F1-score</i> yang stabil. Kelas Karat Daun dan Serangan Ulat menjadi kategori dengan performa terbaik karena memiliki gejala visual yang tegas, sedangkan defisiensi kalium dan magnesium menunjukkan tingkat kesalahan lebih tinggi akibat kemiripan ciri. Temuan ini mengonfirmasi bahwa CNN efektif sebagai perangkat deteksi berbasis citra dan berpotensi dikembangkan lebih lanjut melalui peningkatan kualitas <i>dataset</i> , arsitektur model, serta integrasi dengan sensor lapangan untuk mendukung praktik budidaya kelapa sawit yang lebih presisi dan berkelanjutan [36].

