

BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

Teori adalah sebuah jaringan dari sejumlah ide yang menjelaskan hubungan antara dua variable atau lebih (Mindarti, 2016). Teori dasar adalah dasar atau prinsip-prinsip utama yang digunakan dalam suatu bidang pengetahuan atau disiplin ilmu tertentu.

2.1.1 AI (*Artificial Intelligence*)

AI (*Artificial Intelligence*) atau disebut juga kecerdasan buatan merupakan ilmu komputer yang fokus pada kemampuan mesin dalam mengimitasi kecerdasan perilaku manusia (Choi et al., 2024). Sejarah AI dimulai pada tahun 1950-an dan telah mengalami perkembangan yang signifikan sejak saat itu. AI telah ada selama lebih dari enam dekade dan telah mengalami musim dingin dan musim semi (Duan et al., 2019).

Sebagai bidang penelitian *interdisipliner*, *artificial intelligence* dalam pendidikan mengintegrasikan metode serta alat dari berbagai disiplin ilmu seperti ilmu komputer dan ilmu informasi yang berfungsi untuk mengatasi masalah pendidikan.

Kebanyakan AI sekarang diciptakan bukan hanya untuk membuktikan kecerdasan AI mampu menandingi manusia, malah berupaya untuk berfungsi jauh lebih baik. *Artificial Intelligence* (AI) memiliki kelebihan signifikan, antara lain meningkatkan efisiensi dan produktivitas, analisis data cepat dan akurat, kemampuan pembelajaran dan adaptasi, pemrosesan bahasa alami, pengenalan pola

dan penglihatan komputer, serta meningkatkan interaksi manusia-komputer. AI memberikan manfaat dalam berbagai bidang dengan kinerja yang lebih baik, pengambilan keputusan yang terinformasikan, dan interaksi yang lebih intuitif.

Kelemahan AI meliputi pemahaman konteks terbatas, keandalan yang terpengaruh oleh kesalahan atau bias, ketergantungan pada data pelatihan, kurangnya pengetahuan konteks luas, keterbatasan dalam kreativitas dan intuisi, serta perluasan isu-isu etika dan sosial. Dalam mengatasi kelemahan ini, perlu dipertimbangkan pengembangan AI yang lebih cermat, penggunaan data yang representatif, pendekatan yang lebih holistik, dan pemikiran etis yang matang untuk memastikan penggunaan AI yang bertanggung jawab dan optimal.

Perbedaan AI dengan ilmu lain ialah AI berfokus pada pengembangan kecerdasan buatan dan sistem yang bisa belajar dan beradaptasi. Robotika berkaitan dengan perancangan dan operasi robot fisik, sedangkan AI tidak terbatas pada robot. Pemrosesan Bahasa Alami melibatkan pemahaman dan penggunaan bahasa manusia oleh komputer. Pembelajaran Mesin adalah sub-bidang AI yang berfokus pada pengembangan algoritma untuk belajar dari data. AI juga mencakup konsep dan metode lain seperti pemrosesan bahasa alami dan logika. Kecerdasan Buatan Umum (AGI) adalah konsep kecerdasan buatan setara dengan kecerdasan manusia (Bunyamin, 2018).

Ilmu AI (*Artificial Intelligence*) mencakup berbagai aspek dan teknik yang mencerminkan beragam pendekatan untuk membangun kecerdasan buatan (Kanda Ruskandi, 2021). Beberapa bidang yang tercakup dalam ilmu AI meliputi: Pemrosesan Bahasa Alami (*Natural Language Processing*), Pembelajaran Mesin

(*machine learning*). Jaringan Saraf Tiruan (*Artificial Neural Networks*), Logika dan Pemikiran Penalaran (*Logic and Reasoning*), Penglihatan Komputer (*Computer Vision*), Robotika dan Kendali Otomatis (*Robotics and Autonomous Control*), Sistem Pemrosesan Pengetahuan (*Knowledge Representation and Reasoning*), Pengambilan Keputusan (*Decision Making*) dan *deep learning* yang terakhir.

2.1.2 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (JST) merupakan sistem pemrosesan informasi yang terinspirasi dari cara kerja otak manusia (Sutawinaya, Astawa, & Hariyanti, 2017). Konsep dasar JST diperkenalkan pertama kali oleh Warren McCulloch dan Walter Pitts pada tahun 1943. Mereka menciptakan model matematis untuk neuron sebagai unit pemroses informasi, di mana neuron-neuron ini saling terhubung melalui sinapsis.

2.1.3 Deep Learning

Deep learning adalah cabang ilmu *machine learning* berbasis Jaringan Saraf Tiruan (JST) atau bisa dikatakan sebagai perkembangan dari JST (Pratiwi, Cahyanti, & Lamsani, 2021). *Deep learning* adalah salah satu cabang dari *machine learning* menggunakan arsitektur jaringan saraf tiruan yang dalam (*deep neural networks*) dengan banyak lapisan (*layer*) untuk menghasilkan pemodelan yang lebih kompleks dan representasi yang lebih abstrak dari data.

Dalam *Deep learning*, pelatihan jaringan saraf dilakukan melalui proses iteratif yang disebut pembelajaran mendalam (*deep learning*) untuk mengoptimalkan bobot dan parameter jaringan. Teknik *Deep learning* telah memberikan kemajuan signifikan dalam pengenalan gambar, pemrosesan bahasa

alami, pengenalan suara, pengenalan tulisan tangan, dan berbagai aplikasi lainnya yang membutuhkan pemahaman dan analisis data yang kompleks.

Perbedaan antara *Deep learning* dan *Machine learning* terletak pada tingkat kompleksitas dan representasi data yang diolah. *Machine learning* menggunakan berbagai teknik dan algoritma untuk mengenali pola dalam data yang relatif sederhana dan pemodelan yang dangkal. Di sisi lain, *Deep learning* menggunakan jaringan saraf tiruan dengan banyak lapisan untuk mempelajari representasi yang hierarkis dan kompleks dari data yang lebih abstrak. *Deep learning* dapat menangani masalah yang lebih kompleks seperti pengenalan gambar, pemrosesan bahasa alami, dan pengenalan suara dengan kinerja yang lebih baik. Namun, *Deep learning* juga membutuhkan daya komputasi yang lebih besar dan waktu pelatihan yang lebih lama daripada *Machine learning* tradisional.

Manfaat utama *Deep learning* adalah analisis dan pembelajaran sejumlah besar data yang tidak diawasi, menjadikannya alat yang berharga untuk analisis data besar di mana data mentah sebagian besar tidak diberi label dan tidak dikategorikan (Alfarizi et al., 2023).

Beberapa algoritma *deep learning* yang sering digunakan antara lain CNN untuk pengenalan gambar, RNN untuk pemrosesan urutan data, GAN untuk menghasilkan data sintetis, *Deep Reinforcement Learning* untuk pembelajaran keputusan, DBN untuk klasifikasi pola, dan *Autoencoders* untuk pengurangan dimensi. Selain itu, SSD (*Single Shot Multibox Detector*) adalah algoritma deteksi objek yang menggunakan CNN untuk mendeteksi objek dengan cepat dan akurat, memungkinkan deteksi objek secara *real-time*.

2.1.4 *Object Detection*

Deteksi objek adalah teknik dalam *computer vision* yang digunakan untuk menentukan lokasi objek dalam gambar atau video (Dompeipen et al., 2021).



Gambar 2. 1 Contoh *Objek Detection*
(Sumber: Data Penelitian, 2025)

Dalam *Deep Learning*, tugas deteksi objek biasanya memanfaatkan *Convolutional Neural Network* (CNN) sebagai *backbone*, dan algoritma seperti R-CNN, YOLO v2, atau SSD digunakan untuk mendeteksi lokasi objek yang teridentifikasi oleh CNN. Deteksi objek bertujuan untuk memberikan *bounding box* di sekitar objek yang terdeteksi dalam gambar dan mengaitkan setiap *bounding box* dengan kategori objek yang sesuai.

2.1.5 *Citra Digital*

Citra digital adalah representasi numerik dari gambar yang terdiri dari piksel-piksel yang tersusun dalam *matriks* (Dompeipen et al., 2021). Setiap piksel memiliki nilai numerik yang menggambarkan kecerahan atau warna. Citra digital dapat dihasilkan melalui pemindaian gambar fisik atau perekaman menggunakan kamera digital. Kualitas dan resolusi citra ditentukan oleh jumlah piksel, di mana

semakin banyak piksel, semakin tinggi detail gambar. Proses perekaman dimulai dengan sensor cahaya yang mengubah cahaya dari objek menjadi sinyal listrik dan kemudian dikonversi menjadi data digital.

1. Akuisisi Citra (*Image Acquisition*)

Akuisisi citra adalah proses pengambilan gambar atau data visual dari berbagai sumber seperti kamera, sensor, atau perangkat lainnya. Tujuan dari akuisisi citra adalah untuk mendapatkan informasi visual yang dapat digunakan dalam berbagai aplikasi, termasuk pengolahan citra, analisis visual, pengenalan pola, dan banyak lagi. Proses akuisisi citra melibatkan pengambilan gambar secara digital dan mengkonversinya menjadi data yang dapat diolah dan dianalisis oleh komputer.

2. Pengolahan citra

Pengolahan citra adalah proses manipulasi dan analisis data citra menggunakan teknik-teknik komputer untuk mengubah, memperbaiki, atau mendapatkan informasi dari citra digital. Tujuan utama dari pengolahan citra adalah untuk meningkatkan kualitas, ekstraksi fitur, *segmentasi*, atau interpretasi citra agar mendukung pengambilan keputusan atau analisis lebih lanjut. Pengolahan citra digunakan dalam berbagai bidang, termasuk komputer visi, analisis medis, ilmu bumi, pemrosesan gambar dan video, dan banyak lagi. Teknik-teknik dalam pengolahan citra meliputi filtrasi, segmentasi, ekstraksi fitur, analisis tekstur, pengenalan pola, dan teknik lainnya yang digunakan untuk memproses citra secara digital.

a) *Image Normalization*

Normalisasi adalah proses dalam pemrosesan gambar yang mengubah rentang nilai intensitas piksel dalam gambar. Tujuannya adalah untuk mengubah gambar-gambar yang ingin dianalisis menjadi representasi yang lebih umum dan menunjukkan cara mengidentifikasinya dengan lebih baik (Sumijan, Widya Purnama, & Arlis, 2021).

b) Resize

Resize adalah proses dalam pemrosesan gambar yang dilakukan untuk mengubah ukuran gambar menjadi ukuran yang diinginkan. Hal ini dilakukan untuk memastikan konsistensi ukuran data dan memenuhi kebutuhan dari model atau algoritma yang akan digunakan untuk analisis atau prediksi (Sumijan et al., 2021). Dengan melakukan *resize*, gambar akan memiliki ukuran yang seragam sehingga memudahkan dalam pengolahan dan analisis lebih lanjut.

c) Image Augmentation

Image augmentation adalah teknik dalam pemrosesan gambar yang digunakan untuk menciptakan variasi data baru dari data asli. Hal ini dilakukan dengan melakukan transformasi atau manipulasi pada gambar, seperti *flip*, rotasi, perbesaran, pergeseran, pencerahan, dan pemotongan. Tujuannya adalah untuk meningkatkan variasi data latih, sehingga model dapat belajar lebih baik dan lebih general dalam mengenali pola atau fitur pada gambar. Dengan mengaplikasikan *augmentasi* data, kita dapat mencegah *overfitting* dan meningkatkan kinerja model ketika diterapkan pada data yang belum pernah dilihat sebelumnya.

2.1.6 Kalori

Kalori adalah satuan pengukuran energi. Kalori juga dapat merujuk pada jumlah energi yang terkandung dalam makanan atau minuman yang dikonsumsi. Dalam tubuh manusia, kalori berperan penting sebagai sumber energi untuk menjalankan berbagai fungsi tubuh seperti bernapas, bergerak, dan mempertahankan suhu tubuh. Meremehkan kalori yang lebih besar oleh orang yang kelebihan berat badan adalah konsekuensi dari kecenderungan mereka untuk mengonsumsi makanan yang lebih besar. Meremehkan kalori terkait dengan ukuran makanan, bukan ukuran tubuh.

Terdapat tiga jenis makronutrien, yaitu karbohidrat, protein, dan lemak. Setiap gram karbohidrat mengandung 4 kalori, setiap gram protein mengandung 4 kalori, dan setiap gram lemak mengandung 9 kalori. Karbohidrat, protein dan lemak masing-masing memberikan energi dalam jumlah yang berbeda (Yunita & Nur'aini, 2018).

2.1.7 Aplikasi *Web*

Aplikasi merujuk pada perangkat lunak yang dirancang dan dikembangkan untuk menjalankan tugas-tugas spesifik atau memberikan fungsi tertentu kepada pengguna. Aplikasi dapat diinstal dan dijalankan pada perangkat komputer, smartphone, tablet, atau perangkat lainnya. Mereka dirancang untuk memenuhi kebutuhan dan keinginan pengguna dalam berbagai bidang, seperti komunikasi, produktivitas, hiburan, pendidikan, bisnis, dan banyak lagi.

Aplikasi *web* adalah perangkat lunak yang diakses melalui browser *web* dan berjalan di platform internet. Mereka tidak perlu diunduh dan diinstal secara khusus

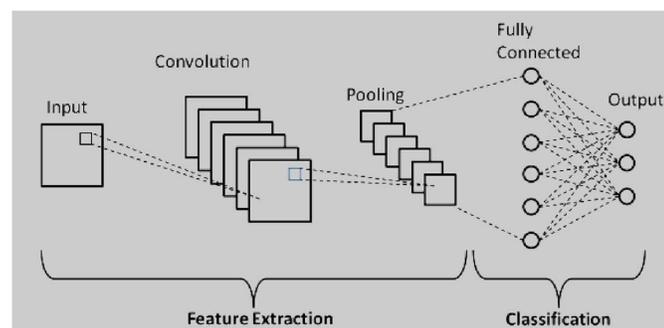
pada perangkat pengguna, melainkan dapat diakses melalui URL atau tautan yang diberikan. Aplikasi *web* dapat digunakan melalui berbagai perangkat, termasuk komputer, laptop, *smartphone*, dan tablet.

2.1.8 FatSecret Indonesia

FatSecret Indonesia adalah platform daring yang menyediakan informasi tentang diet, nutrisi, dan program penurunan berat badan. Platform ini juga memungkinkan pengguna untuk membuat profil, memantau aktivitas, makanan, dan berat badan. FatSecret menyediakan data nutrisi lengkap, termasuk informasi kalori dari berbagai makanan di Indonesia, untuk membantu individu mencapai tujuan diet mereka.

2.1.9 CNN (*Convolutional Neural Network*)

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan yang dirancang untuk mengenali pola dalam data gambar. CNN menggunakan filter atau kernel untuk mengekstraksi fitur dari gambar, memungkinkan model untuk mengenali objek, tekstur, atau pola secara otomatis tanpa memerlukan fitur yang dibuat secara manual. Berikut adalah Arsitektur utama dalam CNN:

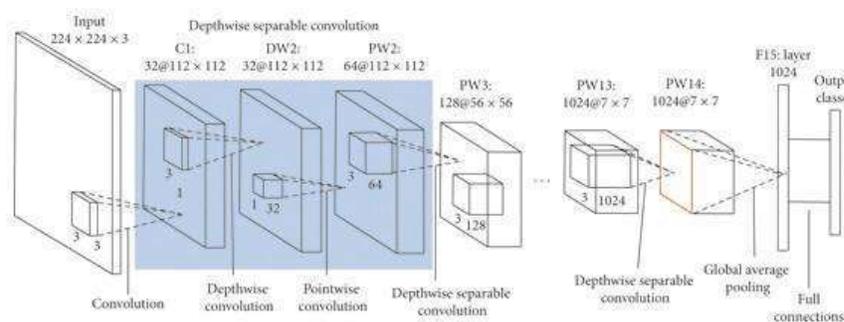


Gambar 2. 2 arsitektur jaringan CNN
(Sumber: (Phung & Rhee, 2019))

1. *Convolutional Layers*: Lapisan utama dalam CNN yang melakukan konvolusi pada gambar untuk mengekstraksi fitur seperti tepi, sudut, atau tekstur.
2. *Pooling Layers*: Lapisan yang digunakan untuk mereduksi dimensi dan kompleksitas data, meningkatkan efisiensi komputasi, dan mencegah *overfitting*. Teknik yang umum digunakan adalah *max pooling*.
3. *Fully Connected Layers*: Lapisan akhir CNN yang menghubungkan semua neuron dari lapisan sebelumnya untuk menghasilkan output, seperti klasifikasi atau prediksi.

2.1.10 *Mobile-Net*

Mobile-Net adalah arsitektur *Convolutional Neural Network* (CNN) yang dirancang untuk efisiensi komputasi pada perangkat dengan sumber daya terbatas, seperti ponsel pintar dan perangkat *embedded*.



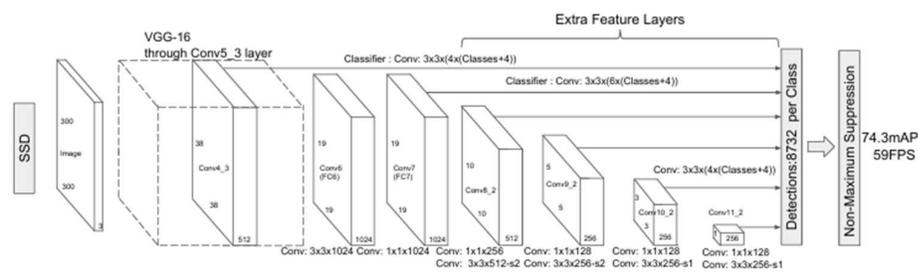
Gambar 2. 3 Arsitektur *Mobile-Net*
(Sumber: (Setyanto et al., 2021))

Dengan menggunakan teknik *Depthwise Separable Convolutions*, *Mobile-Net* membagi konvolusi menjadi dua Langkah *Depthwise Convolution* untuk filter pada setiap saluran gambar dan *Pointwise Convolution* untuk menggabungkan hasilnya

menggunakan filter 1x1. Hal ini memungkinkan pengurangan jumlah parameter dan komputasi yang diperlukan, menjadikannya lebih ringan dan cepat dibandingkan dengan CNN tradisional. *Mobile-Net* sering digunakan dalam aplikasi deteksi objek dan klasifikasi gambar, karena kemampuannya dalam menjaga kecepatan dan akurasi meskipun pada perangkat dengan keterbatasan sumber daya. Keunggulannya menjadikannya populer dalam sistem pemantauan otomatis, perangkat seluler, dan aplikasi pengenalan gambar makanan yang memerlukan pemrosesan cepat dan efisien.

2.1.11 SSD (*Single Shot Multibox Detector*)

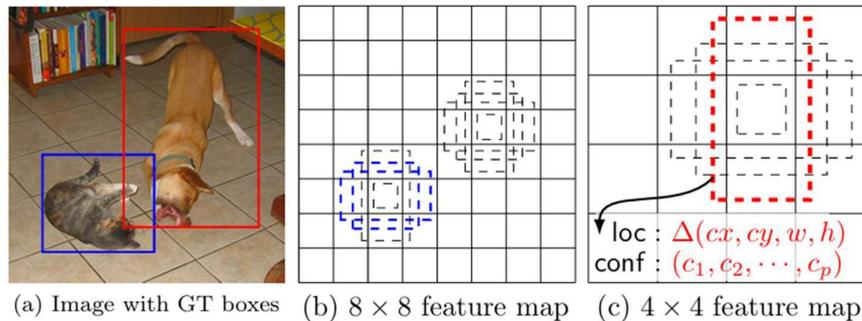
SSD (*Single Shot Detector*) adalah algoritma deteksi objek yang terkenal karena efisiensinya. SSD bekerja dengan cara lebih sederhana dibandingkan algoritma lain seperti R-CNN, yang memerlukan dua tahap pembacaan gambar.



Gambar 2. 4 Arsitektur SSD (*Single Shot Multibox Detector*)
(Sumber: (Liu et al., 2016))

SSD tidak melalui tahap pembuatan proposal atau *feature resampling*, sehingga membuatnya lebih cepat. Dalam SSD, seluruh proses deteksi dilakukan dalam satu langkah menggunakan satu jaringan. Arsitektur CNN seperti *MobileNet* atau VGG-16 digunakan untuk mengekstraksi fitur dari gambar. Setelah fitur diekstraksi, algoritma menghasilkan *feature layer* dengan dimensi $M \times N$ dan P

channel, seperti 8x8 atau 4x4. Proses selanjutnya melibatkan konvolusi 3x3 pada *feature layer* tersebut, yang menghasilkan beberapa *bounding box* dengan ukuran dan rasio aspek yang bervariasi untuk setiap lokasi, memungkinkan deteksi objek secara lebih akurat dan cepat.



Gambar 2. 5 *Feature Map SSD*
(Sumber: (Liu et al., 2016))

Contoh bagaimana SSD memanfaatkan peta fitur multi-skala dan kotak batas default untuk mendeteksi objek dengan skala dan rasio aspek yang berbeda. Objek anjing pada gambar berikut cocok dengan satu kotak default (ditandai dengan warna merah) pada lapisan peta fitur berukuran 4×4 , namun tidak terdapat kotak default yang sesuai pada peta fitur dengan resolusi lebih tinggi, yaitu 8×8 . Sementara itu, objek kucing yang lebih kecil hanya terdeteksi oleh lapisan peta fitur 8×8 melalui dua kotak default (ditandai dengan warna biru).

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 600
Fast YOLO	52.7	155	1	98	448 × 448
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8732	300 × 300
SSD512	76.8	19	1	24564	512 × 512
SSD300	74.3	59	8	8732	300 × 300
SSD512	76.8	22	8	24564	512 × 512

Gambar 2. 6 Perbandingan metode SSD
(Sumber: (Liu et al., 2016))

Model SSD300 mampu mencapai performa sebesar 74.3% mAP (*Mean Average Precision*) dengan kecepatan 59 FPS (*Frame per Second*), sedangkan SSD500 mencapai performa 76.9% mAP pada kecepatan 19 FPS. Sebagai perbandingan, algoritma lain seperti Faster R-CNN menghasilkan 73.2% mAP pada 7 FPS, dan YOLOv1 mencapai 66.4% mAP dengan kecepatan 21 FPS.

2.1.12 *TensorFlow*

TensorFlow adalah pustaka perangkat lunak untuk *machine learning* dan *deep neural networks*, dikembangkan oleh Google Brain. *TensorFlow* memungkinkan definisi, perhitungan, dan optimisasi ekspresi matematis dengan *array* multi dimensi secara efisien. Keunggulannya termasuk dukungan penggunaan GPU dan skalabilitas tinggi pada mesin serta data besar. Banyak digunakan oleh perusahaan untuk aplikasi *machine learning* dan *deep learning*, memudahkan riset dalam pengembangan teknologi terkait.

TensorFlow Lite adalah versi ringan dari *TensorFlow* yang dirancang untuk perangkat mobile atau perangkat dengan sumber daya terbatas.

2.1.13 *Confusion Matrix*

Confusion matrix adalah sebuah tabel yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan prediksi model dengan label sebenarnya (Permadi & Gumilang, 2024). Tabel 2.1 ini memberikan gambaran rinci tentang bagaimana model bekerja pada setiap kelas data.

Tabel 2. 1 *Confusion Matrix*

	A	B
A	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
B	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>
FN	FN_A	FN_B

(Sumber: (SATRIA et al., 2024))

Keterangan:

1. *True Positive (TP)*: Jumlah prediksi benar untuk kelas A.
2. *True Negative (TN)*: Jumlah prediksi benar untuk kelas B.
3. *False Positive (FP)*: Jumlah prediksi salah ketika kelas negatif diprediksi sebagai positif.
4. *False Negative (FN)*: Objek yang ada di ground truth tetapi tidak terdeteksi oleh model (tidak ada *bounding box* dengan $IoU \geq threshold$).

Dari *confusion matrix*, kita dapat menghitung metrik evaluasi seperti:

1. *Precision*: Proporsi prediksi positif yang benar.

$$Precision = \frac{TP}{TP + FP} \quad \text{Rumus 2. 1 Precision}$$

2. *Recall (Sensitivity)*: Proporsi aktual positif yang teridentifikasi dengan benar.

$$Recall = \frac{TP}{TP + FN} \quad \text{Rumus 2. 2 Recall}$$

3. *F1-Score*: Rata-rata harmonis dari *Precision* dan *Recall*.

$$F1 - Score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \quad \text{Rumus 2. 3 F1-Score}$$

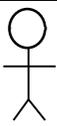
2.1.14 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Putra, 2018).

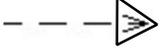
1. *Use case*

Use case diagram menjelaskan manfaat dari aplikasi jika dilihat dari sudut pandang orang yang berada di luar sistem (aktor) (Putra, 2018).

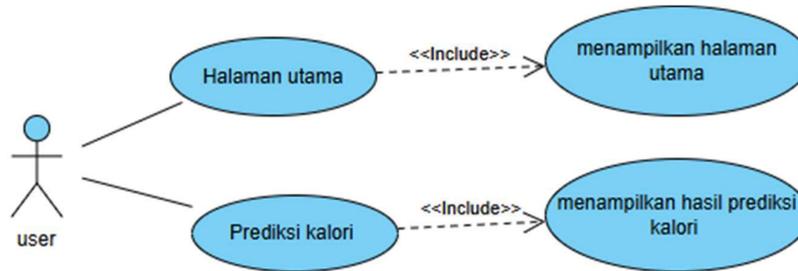
Tabel 2. 2 simbol *use case*

No	Simbol	Nama simbol	Deskripsi
1		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja diawal frase nama use case
2		Aktor	Orang, proses, sistem lai yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor gambar orang tapi aktor belum tentu merupakan orang
3		Asosiasi/ <i>Assosiation</i>	Komunikasi antara aktor dan usecase yang berpartisipasi pada <i>use case</i> dan memiliki interaksi dengan aktor
4		<i>Ekstend / extend</i>	Relasi <i>usecase</i> tambahan kesebuah usecase, dimana usecase yang ditambahkan dapat berdiri sendiri walaupun tanpa usecase tambahan
5		Generalisasi	Hubungan generalisasi dengan spesialisasi (umum-khusus) antara dua buah usecase dimana fungsi yang satu merupakan fungsi yang lebih umum dari lainnya

Lanjutan Tabel 2.2

6		Menggunakan / <i>include/ uses</i>	Relasi usecase tambahan kesebuah usecase dimana <i>use case</i> yang ditambahkan memerlukan usecase ini untuk menjalankan fungsinya
---	---	---------------------------------------	--

(Sumber: (Purnasari et al., 2022))

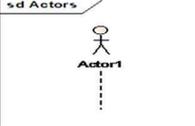
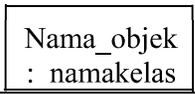


Gambar 2. 7 Contoh Diagram *Use Case*
(Sumber: Data Penelitian, 2024)

2. *Sequence diagram*

Sequence diagram menjelaskan interaksi antara objek di dalam dan di sekitar sistem berupa pesan-pesan yang tersusun dalam urutan waktu, yaitu urutan kejadian yang dilakukan oleh seorang aktor dalam menjalankan sistem.

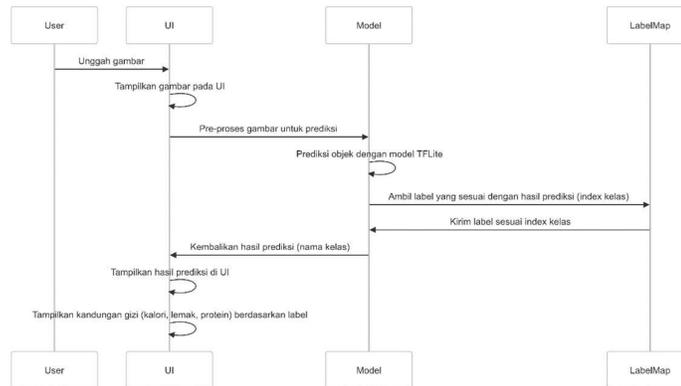
Tabel 2. 3 simbol *sequence diagram*

No	Gambar	Nama	Deskripsi
1		Actor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.
2		LifeLine	Objek entity, antarmuka yang saling berinteraksi.
3		Objek	Menyatakan objek yang berinteraksi oleh pesan.
4		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi
5		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi tentang aktifitas yang terjadi

Lanjutan Tabel 2.3

6	I:keluaran ----->	Pesan tipe return	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
7	I: masukan ----->	Pesan tipe send	Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8	I:nama_metode() ----->	Pesan tipe call	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.

(Sumber: (Purnasari et al., 2022))



Gambar 2. 8 contoh gambar *sequence* diagram
(Sumber: Data Penelitian, 2025)

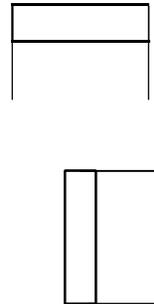
3. Activity Diagram

Diagram *Activity* gambar aktivitas user terhadap semua menu yang dibuat dalam sistem, berikut salah satu aktivitas dosen terhadap sistem yang berhubungan dengan menu Pendidikan dosen.

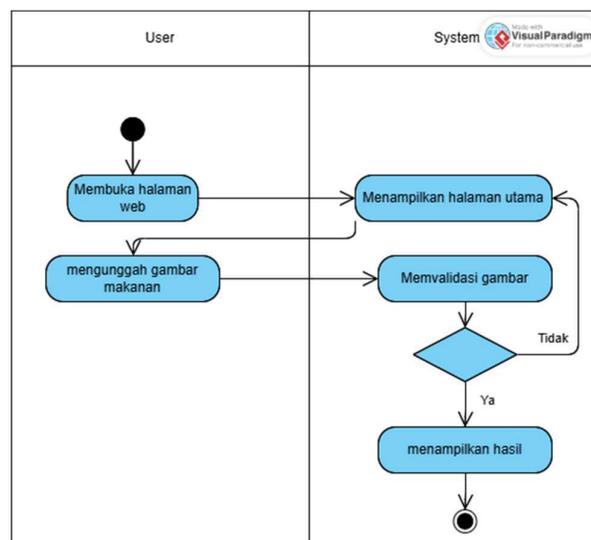
Tabel 2. 4 simbol *Activity* diagram

no	simbol	Nama simbol	Deskripsi
1	●	Status awal	Status awal aktifitas sistem, sebuah diagram aktifitas memiliki sebuah status awal

Lanjutan Tabel 2.4

2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3		Percabangan <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu
4		Pengabungan <i>join</i>	Asosiasi pengabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5		Status akhir	Tatus akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6		<i>swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi

(Sumber: (Purnasari et al., 2022))



Gambar 2. 9 Contoh *Diagram Activity*
(Sumber: Data penelitian, 2025)

2.2. Penelitian Terdahulu

Berikut adalah penelitian terdahulu yang menjadi referensi dalam penelitian ini:

1. Penelitian "Deteksi Objek pada Citra Makanan Sebagai Rekomendasi Diet Menggunakan Metode *Mask R-CNN*" bertujuan untuk mendeteksi dan mengklasifikasikan objek makanan guna mendukung sistem rekomendasi diet. Metode yang digunakan adalah *Mask R-CNN*, yang unggul dalam segmentasi objek dengan tingkat akurasi yang lebih baik dibandingkan metode konvensional. Dataset citra makanan menjalani proses *preprocessing* seperti *resizing* dan anotasi sebelum pelatihan model dilakukan. Hasil penelitian menunjukkan bahwa model yang dikembangkan mampu mendeteksi beberapa jenis makanan dengan akurasi 72%. Namun, terdapat kendala seperti variasi tampilan makanan, keterbatasan dataset, dan kebutuhan sumber daya komputasi tinggi. Oleh karena itu, pengembangan lebih lanjut disarankan dengan meningkatkan jumlah dataset, menerapkan teknik augmentasi data, serta mengeksplorasi algoritma lain seperti YOLO atau SSD untuk meningkatkan akurasi dan efisiensi system (Ratri Enggar Pawening et al., 2023).
2. Penelitian "*FoodTracker: A Real-time Food Detection Mobile Application by Deep Convolutional Neural Networks*" mengembangkan aplikasi mobile untuk mendeteksi makanan secara *real-time* dan memberikan informasi nutrisi. Mengatasi keterbatasan pencatatan makanan manual, penelitian ini menggunakan *MobileNet* sebagai backbone dan *YOLOv2* untuk deteksi objek. Hasilnya menunjukkan akurasi mean *average precision (mAP)* hampir 80%

pada dataset *UECFood100* dan *UECFood256*. Kendala utama adalah variasi jenis makanan, keterbatasan *dataset*, dan optimalisasi komputasi untuk perangkat *mobile*. Pengembangan lebih lanjut diperlukan dengan memperluas dataset dan meningkatkan efisiensi model agar lebih ringan dan cepat (Sun et al., 2019).

3. Penelitian "Deteksi Kalori Makanan Tradisional Indonesia Menggunakan Metode *Single Shot Multibox Detector (SSD)*" bertujuan mengembangkan sistem otomatis untuk mendeteksi makanan tradisional Indonesia dan memperkirakan jumlah kalorinya menggunakan teknologi pengolahan citra digital. Dengan metode SSD, model dilatih menggunakan *dataset* yang dibagi menjadi 90% training, 10% validasi, dan 10% testing, menghasilkan *mean average precision (mAP)* 65.09% dan total loss 0.1670681. Kendala utama meliputi variasi bentuk dan penyajian makanan serta keterbatasan dataset. Pengembangan lebih lanjut diperlukan dengan menambah dataset dan mengeksplorasi metode lain seperti Faster R-CNN atau YOLO untuk meningkatkan akurasi sistem deteksi (Ratri Enggar Pawening et al., 2023).
4. Penelitian "*ConciseCarNet: Convolutional Neural Network for Parking Space Classification*" mengembangkan model *deep learning* berbasis CNN untuk mendeteksi tempat parkir kosong secara otomatis. Mengatasi keterbatasan sensor fisik, penelitian ini menggunakan *computer vision* dengan arsitektur *ConciseCarNet*, yang lebih efisien dan akurat. Dengan metode *preprocessing*, augmentasi data, dan pelatihan model, hasil evaluasi menunjukkan akurasi 97,8% dan *F1-score* 96%. Kendala utama meliputi variasi pencahayaan, sudut

kamera, dan kebutuhan daya komputasi tinggi. Pengembangan lebih lanjut diperlukan untuk optimalisasi model agar lebih ringan dan dapat diterapkan pada sistem parkir cerdas berbasis *edge computing* (Ramli et al., 2024).

5. Penelitian "*Catbreedsnet: An Android Application for Cat Breed Classification Using Convolutional Neural Networks*" mengembangkan aplikasi Android untuk klasifikasi ras kucing menggunakan CNN dan *transfer learning*. Dengan dataset gambar dari 13 ras kucing, model *MobileNetV2*, *VGG16*, dan *InceptionV3* diuji untuk mendapatkan performa terbaik. Hasilnya, *MobileNetV2* menunjukkan akurasi tertinggi sebesar 82%, yang kemudian diterapkan dalam aplikasi *Catbreedsnet*. Aplikasi ini memungkinkan pengguna mengambil foto kucing dan mengidentifikasi rasnya secara otomatis. Kendala utama meliputi keterbatasan dataset, variasi pencahayaan, dan optimasi model untuk perangkat *mobile*. Untuk pengembangan lebih lanjut, disarankan peningkatan dataset, optimasi model agar lebih ringan, dan penambahan fitur deteksi penyakit berdasarkan ras kucing (Ramadhan et al., 2023).
6. Penelitian "*Comparison of EfficientNetB0 and EfficientNetB7 Models in Classifying Malaria Based on Blood Cells*" membandingkan performa dua model *deep learning*, *EfficientNetB0* dan *EfficientNetB7*, dalam klasifikasi sel darah yang terinfeksi malaria. Dengan dataset 27.558 gambar dari *Kaggle*, penelitian ini menerapkan *preprocessing* seperti normalisasi dan augmentasi data. Hasil evaluasi menunjukkan *EfficientNetB7* mencapai akurasi 99%, lebih unggul dibandingkan *EfficientNetB0* yang mencapai 95%. Kendala utama meliputi keterbatasan dataset dan kebutuhan daya komputasi tinggi untuk

model kompleks. Untuk pengembangan lebih lanjut, disarankan optimasi model agar lebih ringan serta integrasi dengan aplikasi AI untuk membantu diagnosis malaria secara otomatis (Kiki et al., 2024).

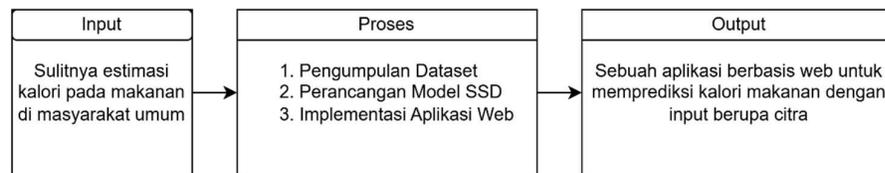
7. Penelitian "Penggunaan CNN untuk Menentukan Jumlah Kalori pada Sayuran dan Buah Menggunakan *Image Processing*" mengembangkan sistem berbasis CNN untuk mengklasifikasikan sayuran dan buah serta memperkirakan jumlah kalorinya menggunakan image processing. Dengan dataset gambar yang telah diproses melalui normalisasi dan augmentasi, model dilatih untuk mengekstraksi fitur makanan dan mengaitkannya dengan nilai kalori berdasarkan referensi nutrisi. Hasilnya menunjukkan akurasi klasifikasi 99%, dengan sistem mampu memberikan estimasi kalori berdasarkan jenis makanan. Kendala utama adalah variasi pencahayaan, keterbatasan dataset, dan kesulitan dalam menentukan ukuran makanan secara akurat. Untuk pengembangan lebih lanjut, diperlukan integrasi teknologi depth estimation guna meningkatkan akurasi estimasi kalori serta perluasan dataset agar model lebih robust (Ardiansyah et al., 2025).
8. Penelitian ini berjudul "Deteksi Ikan Segar Secara Realtime dengan YOLOv4 menggunakan *Metode Convolutional Neural Network*", yang bertujuan untuk mengembangkan sistem deteksi kesegaran ikan secara real-time guna membantu konsumen memilih ikan berkualitas. Permasalahan utama adalah sulitnya membedakan ikan segar dari yang sudah tidak layak konsumsi, serta potensi pedagang menjual ikan yang tidak segar. Metode yang digunakan dalam penelitian ini adalah kombinasi YOLOv4 untuk deteksi objek dan

Convolutional Neural Network (CNN) untuk klasifikasi. Data yang digunakan mencakup 118 citra untuk pelatihan dan 13 citra untuk pengujian, dengan total 6.000 *epoch* pelatihan. Hasil pengujian menunjukkan performa sistem yang tinggi dengan *Mean Average Precision* (MAP) sebesar 93,75%, *presisi* 100%, *recall* 93%, dan *f-score* 96%, serta rata-rata nilai *Intersection over Union* (IoU) sebesar 74,17%. Kendala utama yang mungkin dihadapi dalam implementasi sistem ini mencakup keterbatasan data pelatihan, variasi pencahayaan saat pengambilan gambar, dan kebutuhan daya komputasi yang tinggi untuk mendukung model real-time. Secara keseluruhan, penelitian ini berhasil membuktikan bahwa kombinasi YOLOv4 dan CNN mampu mendeteksi serta mengklasifikasikan kesegaran ikan dengan akurasi tinggi, sehingga dapat diterapkan dalam berbagai aplikasi berbasis visi computer (Gunawan et al., 2023).

9. Penelitian "*Official logo recognition based on multilayer convolutional neural network model*" mengembangkan model CNN berlapis untuk mengenali logo resmi secara otomatis. Tantangan utama adalah variasi bentuk, warna, dan orientasi logo. Dengan menggunakan Multilayer CNN, model ini dilatih pada dataset logo resmi dari pemerintahan Irak dan menunjukkan akurasi tinggi sebesar 99,16%. Model ini efektif dalam klasifikasi logo dan dapat diterapkan dalam sistem verifikasi merek dan keamanan dokumen. Kendala utama meliputi keterbatasan dataset, variasi kompleks logo, dan kebutuhan daya komputasi tinggi (Abdullah et al., 2022) .

2.3. Kerangka Pemikiran

Kerangka pemikiran adalah suatu pola pikir atau konsep yang menjadi dasar dalam menjalankan penelitian berikut kerangka pemikiran Penulis dalam Penelitian.



Gambar 2. 10 Gambar Kerangka Pemikiran
(Sumber: Data Penelitian, 2025)

Flowchart pada gambar 2.10 ini menjelaskan alur pengembangan aplikasi web untuk prediksi kalori makanan:

1. *Input*: Sulitnya estimasi kalori pada makanan di masyarakat umum, banyak individu mengalami kesulitan dalam memperkirakan jumlah kalori yang masuk ke dalam tubuh, yang dapat menyebabkan ketidakseimbangan antara asupan dan kebutuhan energi.
2. *Proses*: Proses pengembangan sistem prediksi kalori berbasis web mencakup pengumpulan dataset, perancangan model SSD, dan implementasi aplikasi web. Dataset gambar makanan dikumpulkan untuk melatih model deteksi objek. Model SSD digunakan karena kemampuannya mendeteksi makanan secara real-time dengan akurasi tinggi. Terakhir, model yang telah dilatih diintegrasikan ke dalam aplikasi web, memungkinkan pengguna mengunggah gambar makanan dan memperoleh estimasi kalori secara otomatis.

3. *Output*: Sebuah aplikasi berbasis web yang memungkinkan pengguna mengunggah citra makanan untuk memperoleh estimasi kalori secara otomatis. Sistem ini menggunakan model deteksi objek berbasis SSD (*Single Shot MultiBox Detector*) untuk mengenali jenis makanan, lalu menghitung perkiraan kalori berdasarkan hasil deteksi.