

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Teori Dasar

2.1.1 Laptop

Laptop adalah perangkat komputer portabel yang berukuran kecil dan ringan, dirancang untuk mudah dibawa dan digunakan kapan saja dan dimana saja. Bobot laptop umumnya berkisar antara 1 hingga 6 kg, tergantung pada ukuran dan bahan yang digunakan. Di era modern ini, laptop memegang peranan penting dalam mendukung berbagai aktivitas sehari-hari. Perangkat ini memungkinkan pengguna menyelesaikan berbagai jenis pekerjaan, mulai dari tugas-tugas sederhana hingga proyek-proyek kompleks, dengan mengandalkan teknologi canggih yang terintegrasi di dalamnya. Sebagai salah satu contoh utama dalam teknologi informasi, laptop terus mengalami perkembangan untuk memenuhi kebutuhan pengguna dan menyesuaikan dengan kemajuan zaman (Yassirullah Apta Saputra, 2021).

2.1.2 Website

Website atau yang biasa disebut situs web adalah kumpulan halaman dalam domain yang berisi berbagai informasi yang dapat diakses oleh pengguna Internet melalui mesin pencari. Jenis informasi ini biasanya termasuk gambar, ilustrasi, video, dan konten lainnya (Immanuel Rui Costa et al., 2022). Kemudian menurut (Finsensia Riti, 2023) *Website* merupakan serangkaian halaman web yang dirancang khusus sebagai media informasi terkait kepentingan tertentu, semisal pada instansi, organisasi, bisnis, dan masih banyak lagi. Dengan demikian, *website*

merupakan sistem server internet yang mendukung dokumen dengan format *HTML* (*Hyper Text Markup Language*).

2.1.3 Python

Python adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan dikembangkan oleh Guido van Rossum. Sebagai bahasa lintas fungsi yang ditafsirkan, *Python* memiliki banyak fitur yang memudahkan pengolahan kumpulan data kompleks. Fleksibilitasnya menjadikannya populer di kalangan ilmuwan dan analis data, karena memungkinkan pembuatan model data, sistematisasi data, pengembangan algoritma untuk *Machine Learning*, layanan web, dan penerapan penambahan data untuk menyelesaikan berbagai tugas dengan cepat. *Python* menekankan keterbacaan kode, sehingga sintaksnya mudah dipahami, menjadikannya pilihan yang baik baik untuk pemula maupun bagi mereka yang sudah berpengalaman dalam bahasa pemrograman lain. Selain itu, *Python* dilengkapi dengan berbagai perpustakaan siap pakai yang mendukung berbagai aplikasi, mulai dari pemrograman statistik hingga pembelajaran mendalam (Junaidi et al., 2023).

2.1.4 Database

Database adalah sekumpulan data yang memiliki keterkaitan satu sama lain, disimpan dengan pengaturan redundansi untuk memastikan efisiensi penggunaannya dalam berbagai aplikasi. Data ini dikelola secara terpisah dari perangkat lunak yang mengaksesnya, memungkinkan pengelolaan seperti penambahan, pembaruan, serta pengambilan data dengan cara yang terstruktur dan terkendali. Basis data juga berfungsi sebagai sistem penyimpanan digital yang

dirancang untuk memenuhi kebutuhan informasi organisasi secara efisien (Riyan Dirgantara et al., 2023).

2.1.5 Chatbot

Chatbot adalah sebuah program komputer yang dirancang untuk mensimulasikan percakapan dengan pengguna manusia, baik melalui pesan teks, suara maupun visual. *Chatbot* memanfaatkan teknologi kecerdasan buatan untuk memahami dan merespons input yang diberikan oleh pengguna. *Chatbot* dapat ditemukan dalam berbagai aplikasi, mulai dari layanan pelanggan dan pembelian produk. *Chatbot* dirancang untuk memberikan respon yang cepat dan akurat, yang pada akhirnya meningkatkan pengalaman pengguna (Harahap & Fitria, 2020).

Chatbot dapat dikategorikan menjadi dua jenis, yaitu:

1. *Chatbot* Berbasis Aturan (*Rule-Based*): *Chatbot* jenis ini bekerja berdasarkan sekumpulan aturan yang sudah ditentukan sebelumnya. Interaksi dengan pengguna dibatasi oleh respons yang sudah diprogram, sehingga kemampuannya terbatas dan tidak bisa menangani pertanyaan atau pernyataan yang tidak sesuai dengan aturan yang telah ditentukan. (Soyusiawaty & Ganda Putra, 2023).
2. *Chatbot* Berbasis Kecerdasan Buatan (*AI-Based*): *Chatbot* ini menggunakan teknologi kecerdasan buatan, untuk memahami dan merespons bahasa manusia secara lebih alami. *Chatbot* berbasis kecerdasan buatan mampu belajar dari interaksi sebelumnya dan memberikan respons yang lebih akurat (Suryana et al., 2022).

Konsep *chatbot* pertama kali diperkenalkan oleh Alan Turing pada tahun 1950 melalui *Turing Test*, yang bertujuan untuk mengevaluasi kemampuan mesin dalam meniru kemampuan manusia. *Chatbot* pertama, *ELIZA* yang dikembangkan oleh Joseph Weizenbaum dibuat pada tahun 1966 dan dirancang untuk meniru percakapan layaknya seorang psikiater. Meskipun kemampuan *ELIZA* terbatas, *ELIZA* menjadi inspirasi bagi pengembangan *chatbot* di masa depan.

Pada tahun 1972, muncul *PARRY* yang memiliki kemampuan lebih baik dibandingkan *ELIZA*. *PARRY* dirancang untuk mensimulasikan perilaku individu dengan skizofrenia paranoid dan dianggap sebagai program yang lebih serius dan maju. *PARRY* diuji dalam konteks Turing Test, di mana para psikiater mengalami kesulitan dalam membedakan antara pasien nyata dan *PARRY*.

Perkembangan selanjutnya terjadi pada tahun 1988 dengan hadirnya *Jabberwacky*, yang menggunakan pola percakapan untuk merespons dengan lebih alami. *Jabberwacky* mampu menjawab berdasarkan konteks diskusi sebelumnya, meskipun masih memiliki keterbatasan dalam menangani banyak pengguna sekaligus. Pada tahun 1991, istilah *chatterbot* mulai dikenal luas, ketika *chatbot* mulai berinteraksi dengan pengguna dalam dunia virtual.

Di tahun 1995, *ALICE* (Artificial Linguistic Internet Computer Entity) diluncurkan sebagai *chatbot* online pertama yang terinspirasi dari *ELIZA*, dengan pengetahuan yang lebih luas dan kemampuan untuk melakukan percakapan yang lebih kompleks. Kemajuan teknologi *chatbot* berlanjut dengan pengembangan *SmarterChild* pada tahun 2001, yang merupakan *chatbot* pertama yang dapat

membantu pengguna dalam tugas sehari-hari seperti mengambil informasi tentang cuaca dan berita (Adamopoulou & Moussiades, 2020). Seiring berjalannya waktu, muncul asisten suara pribadi pintar seperti Siri, Google Assistant, dan Amazon Alexa. Asisten-asisten ini tidak hanya memahami perintah suara tetapi juga dapat melakukan tugas-tugas otomatisasi rumah dan menjawab pertanyaan pengguna secara lebih cerdas. Dengan demikian, perkembangan chatbot telah menunjukkan kemajuan signifikan dalam interaksi manusia-komputer serta pemahaman bahasa alami.

2.1.6 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah cabang dari *Artificial Intelligence (AI)* yang fokus pada interaksi antara komputer dan manusia melalui bahasa alami. *NLP* memungkinkan komputer untuk memahami, menafsirkan, dan menghasilkan teks atau percakapan dalam bahasa manusia (Rumaisa et al., 2021).

Komponen-komponen yang ada pada *NLP*:

1. *Tokenization*: Proses memecah teks menjadi unit-unit kecil yang disebut token, seperti kata atau frasa. Contoh: Kalimat "Saya membutuhkan laptop gaming " akan dipecah menjadi: ["Saya", "membutuhkan", "laptop", "gaming"] (Kurniawan et al., 2022).
2. *Lemmatization* dan *Stemming*: Mengubah kata menjadi bentuk dasarnya dengan menghapus awalan dan akhiran kata. Contoh: Kata "membutuhkan" akan diubah menjadi "butuh" (Iffan Alfanar & Sudanawati Rozas, 2020).

3. *Part-of-Speech (POS) Tagging*: Mengidentifikasi jenis kata dari setiap token, seperti kata benda, kata kerja atau kata sifat. Contoh: Kalimat “saya butuh laptop” *POS tagging* akan mengidentifikasi “saya” sebagai kata ganti, “butuh” sebagai kata kerja dan “laptop” sebagai kata benda (Rizki Amalia & Cahyana Aminuallah, 2022).
4. *Named Entity Recognition (NER)*: Komponen yang mengenali dan mengkategorikan entitas penting dalam teks, seperti nama orang, tempat, organisasi atau tanggal. Contoh: Kalimat “Filbert Young lahir di Hawaii” *NER* akan mengenali “Filbert Young” sebagai entitas orang dan “Hawaii” sebagai entitas tempat (Nuli Giarsyani et al., 2020).
5. *Sentiment Analysis*: Menentukan emosi yang terkandung dalam teks, apakah itu positif, negatif atau netral. Contoh: Kalimat “Saya sangat senang hari ini” akan diidentifikasi sebagai memiliki emosi yang positif (Ardiani et al., 2020).
6. *Bag of Words (BoW)*: metode representasi teks dalam bentuk angka yang digunakan untuk memproses data teks dalam *Natural Language Processing (NLP)*. Metode ini bekerja dengan menghitung frekuensi kemunculan setiap kata unik dalam sebuah kumpulan data dan menyusunnya menjadi sebuah vektor. (HaCohen-Kerner et al., 2020).

Setiap komponen ini bekerja secara berurutan dalam sistem *NLP* untuk memahami teks secara lebih dalam. Misalnya, dalam *chatbot*, setelah kalimat

pengguna *ditokenization*, setiap token dianalisis dengan *POS tagging* dan *lemmatization* untuk memahami kata-kata dalam konteks yang lebih luas.

Dalam pengembangan chatbot untuk rekomendasi laptop, *NLP* digunakan untuk memahami dan menafsirkan input dari pengguna, yang memungkinkan *chatbot* untuk memberikan respon yang lebih relevan. *NLP* memungkinkan *chatbot* untuk menganalisis konteks percakapan, dan memberikan saran yang sesuai dengan situasi yang dihadapi pengguna.

2.1.7 Artificial Intelligence (AI)

Artificial Intelligence (AI) adalah cabang ilmu komputer yang bertujuan untuk mengembangkan sebuah sistem yang dapat melakukan tugas yang umumnya memerlukan kecerdasan manusia. *AI* melibatkan penggunaan algoritma dan model dari *Machine Learning (ML)* dan *Deep Learning (DL)* untuk membangun sistem yang dapat belajar, beradaptasi, dan membuat keputusan secara otomatis. *AI* berperan dalam menciptakan sistem yang tidak hanya memahami dan merespons input pengguna tetapi juga dapat melakukan tugas-tugas kompleks seperti analisis sentimen dan pengelolaan percakapan yang lebih natural. Teknologi *AI* membantu dalam mengembangkan *chatbot* yang lebih pintar dan efektif (Luh Putu Ary Sri Tjahyanti et al., 2022).

Sejarah *Artificial Intelligence (AI)* dimulai pada tahun 1940 ketika para ilmuwan mulai mencoba potensi komputer dalam menyelesaikan berbagai tugas kompleks. Pada tahun 1943, Warren McCulloch dan Walter Pitts menerbitkan karya berjudul "*A Logical Calculus of Ideas Immanent in Nervous Activity*," yang

memperkenalkan model jaringan. Karya ini menjadi landasan penting bagi pengembangan jaringan saraf tiruan di masa depan. Selain itu, Alan Turing memberikan kontribusi signifikan melalui berbagai tulisannya, termasuk pengenalan istilah "*Computer Intelligence*" pada tahun 1947 dan "*Turing Test*" pada tahun 1950, yang berfungsi untuk mengukur kecerdasan mesin (Sheikh et al., n.d.).

Seiring dengan kemajuan teknologi, penerapan *AI* semakin luas di berbagai sektor, mulai dari layanan pelanggan hingga analisis data besar. Contohnya, dalam industri layanan kesehatan, *AI* digunakan untuk mendeteksi penyakit dari hasil pemindaian dengan akurasi yang tinggi, membantu dokter dalam pengambilan keputusan (Trenggono & Bachtiar, 2023).

Tantangan-tantangan yang dihadapi dalam penerapan kecerdasan buatan:

1. Etika dan Moralitas: Penggunaan kecerdasan buatan dalam pengambilan keputusan yang mempengaruhi kehidupan manusia menimbulkan isu etis.
2. Privasi dan Keamanan Data: Implementasi *AI* melibatkan pengumpulan dan pemrosesan data dalam jumlah besar, yang dapat menimbulkan risiko terhadap privasi, terutama jika data pribadi disalahgunakan atau diakses tanpa izin. Keamanan data menjadi perhatian utama, mengingat adanya potensi ancaman dari serangan siber.
3. Risiko Diskriminasi: Sistem *AI* berpotensi membuat keputusan yang tidak adil atau bias jika dilatih menggunakan data yang mengandung

bias. Menemukan dan mengatasi bias dalam *dataset* dan algoritma menjadi tantangan yang perlu diatasi.

4. Perubahan dalam Dunia Kerja: Otomatisasi yang dipicu oleh *AI* dapat mengubah struktur dunia kerja. Hal ini memunculkan pertanyaan tentang bagaimana kita dapat beradaptasi dengan perubahan tersebut dan menjaga kesejahteraan tenaga kerja.
5. Keamanan Sistem *AI*: Melindungi sistem *AI* dari potensi ancaman dan manipulasi menjadi semakin penting seiring dengan berkembangnya kompleksitas teknologi ini (Rozali et al., 2024).

Penerapan *AI* juga membuka berbagai peluang yang sangat besar. Di sektor kesehatan, misalnya, *AI* dapat dimanfaatkan untuk mendiagnosis penyakit, memprediksi hasil pengobatan, serta memberikan perawatan yang lebih personal dan terarah. Di bidang manufaktur, *AI* berpotensi untuk meningkatkan efisiensi dalam produksi serta kualitas produk yang dihasilkan. Sementara itu, dalam dunia transportasi, *AI* dapat digunakan untuk mengembangkan kendaraan otonom yang mampu mengurangi angka kecelakaan dan kemacetan. Peluang-peluang ini menggambarkan betapa besar potensi *AI* dalam mendorong kemajuan yang signifikan di berbagai bidang serta meningkatkan kualitas hidup manusia (Masrichah, 2023).

2.1.8 Machine Learning (ML)

Machine Learning (ML) adalah cabang dari *Artificial Intelligence (AI)* yang telah berkembang dengan cepat dan telah membantu untuk mengatasi masalah

dalam berbagai bidang. *ML* memanfaatkan algoritma untuk menganalisis data, mengenali pola, dan membuat keputusan berdasarkan data yang ada. *ML* digunakan untuk mengajarkan *chatbot* cara merespons berbagai input dari pengguna dengan relevan dan akurat (Heryadi et al., 2020).

Machine Learning (ML) terbagi menjadi 3 kategori: *Supervised Learning*, *Unsupervised Learning* dan *Reinforcement Learning*. Setiap kategori memiliki karakteristik dan metode pendekatan yang berbeda, yang dapat disesuaikan dengan kebutuhan spesifik suatu bidang.

Dalam kategori *Supervised Learning*, data yang digunakan sudah memiliki label sehingga memungkinkan algoritma seperti *Support Vector Machine (SVM)*, *Random Forest* dan *Neural Networks* untuk mempelajari pola yang kemudian dapat diaplikasikan pada data baru. Contoh penerapannya meliputi pemantauan lalu lintas, klasifikasi data produksi industri serta diagnosis dan klasifikasi penyakit dalam bidang medis.

Dalam kategori *Unsupervised Learning*, bekerja dengan data yang tidak berlabel dan digunakan untuk menemukan pola tersembunyi atau mengelompokkan data. Algoritma seperti *k-means* dan *DBSCAN* digunakan dalam analisis data besar di sektor keuangan dan untuk segmentasi gambar medis. Kategori ini sering digunakan untuk memahami struktur data yang kompleks tanpa perlu diawasi secara langsung.

Kategori *reinforcement learning* adalah pendekatan di mana algoritma belajar melalui interaksi dengan lingkungan dan memperbaiki kinerjanya

berdasarkan umpan balik. Contoh penerapan di bidang teknologi adalah optimisasi proses dalam *Internet of Things* (IoT). Algoritma seperti *Q-Learning* dan *Markov Decision Process* digunakan untuk memecahkan masalah yang memerlukan keputusan dalam lingkungan yang dinamis (Roihan et al., 2019).

2.1.9 Deep Learning (DL)

Deep Learning (DL) adalah cabang dari *Machine Learning* yang algoritmanya terinspirasi oleh struktur otak manusia. Struktur ini dikenal sebagai *Artificial Neural Networks (ANN)* atau Jaringan Saraf Tiruan (JTT). *Deep Learning* memiliki berbagai algoritma khusus, di antaranya adalah *Convolutional Neural Network (CNN)*, *Long Short-Term Memory (LSTM)*, *Recurrent Neural Network (RNN)*, dan *Self Organizing Maps (SOM)* (Riziq sirfatullah Alfarizi et al., 2023).

Convolutional Neural Network adalah salah satu algoritma dalam Deep Learning yang dirancang untuk mengolah data dalam bentuk dua dimensi. CNN sering digunakan untuk mengklasifikasi data berlabel melalui metode supervised learning. Algoritma ini sangat efektif dalam pengenalan pola dan analisis citra, sehingga banyak diterapkan dalam berbagai bidang, termasuk visi komputer dan pengolahan gambar (Sriyati et al., 2020).

Long Short-Term Memory Network adalah algoritma yang dapat digunakan untuk mengklasifikasi, memproses dan membuat prediksi berdasarkan data *time series*. *LSTM* memanfaatkan memori ini untuk memberdayakan jaringan berulang dengan kemampuan menghafal yang lebih kuat serta kemampuan untuk

menemukan hubungan di antara data. Hal ini menjadikannya sangat berguna dalam pengolahan data sekuensial seperti teks dan audio (Khumaidi et al., 2020).

Recurrent Neural Network merupakan algoritma yang dirancang khusus untuk mempelajari pola dari data yang kompleks. *RNN* biasanya digunakan untuk pengenalan gambar, pengenalan suara dan bidang lainnya. Keunggulan *RNN* terletak pada kemampuannya untuk mempertahankan informasi dari input sebelumnya, sehingga dapat memahami konteks dalam urutan data (Lubis et al., 2024).

Self Organizing Maps adalah algoritma yang bertujuan untuk visualisasi data dengan mengurangi dimensi data sehingga lebih mudah dipahami oleh manusia. Dengan memetakan data ke dalam dimensi yang lebih rendah, *SOM* memungkinkan pengguna untuk melihat pola dan struktur dalam data yang kompleks secara lebih intuitif (Yahya et al., 2024).

Deep learning memiliki sejumlah keunggulan yang menjadikannya teknologi yang sangat kuat. Salah satunya adalah kemampuannya untuk belajar dari data dalam jumlah besar dan kompleks tanpa perlu program eksplisit, sehingga mampu mengidentifikasi pola dan hubungan yang rumit. Dalam beberapa kasus, seperti pengenalan gambar dan penerjemahan bahasa, *deep learning* bahkan telah mencapai tingkat akurasi yang melampaui kemampuan manusia. Selain itu, model *deep learning* dapat diaplikasikan pada data baru yang belum pernah digunakan selama pelatihan, memungkinkan penerapan yang luas. Dengan dukungan perangkat keras yang canggih, *deep learning* juga dapat memproses data dalam

jumlah besar secara efisien. Namun, di balik keunggulannya, *deep learning* memiliki beberapa keterbatasan. Salah satunya adalah kebutuhan akan data pelatihan dalam jumlah besar, yang tidak selalu tersedia. Kinerja *deep learning* juga sangat tergantung pada kualitas data, data yang bias atau bising dapat menghasilkan model yang kurang akurat. Selain itu, model *deep learning* sering dianggap sebagai "kotak hitam," sehingga sulit untuk memahami bagaimana keputusan dibuat, yang dapat memengaruhi interpretasi dan kepercayaan. Proses pelatihan model *deep learning* juga membutuhkan sumber daya komputasi yang besar, sehingga tidak selalu mudah diakses oleh semua kalangan (Bintang & Imaduddin, 2024).

Salah satu penerapan *Deep Learning* yang sering ditemukan adalah dalam sistem algoritma permainan catur. Kecerdasan buatan dalam permainan ini, terutama pada tingkat kesulitan tinggi, sangat sulit untuk dikalahkan, bahkan oleh pemain yang sangat terampil. Hal ini disebabkan oleh algoritma yang digunakan dalam game tersebut yang mampu menganalisis jutaan kemungkinan langkah yang tercatat dari pertandingan sebelumnya. Langkah-langkah tersebut disimpan dengan rapi, dan komputer dapat menemukan solusi terbaik untuk menghadapi langkah-langkah tersebut, semua itu dilakukan dalam waktu singkat. Dengan demikian, dapat dikatakan bahwa *deep learning* adalah model yang dapat mempelajari metode komputasinya sendiri menggunakan "otaknya" sendiri (Raup et al., 2022).

2.1.10 SQLite

SQLite adalah sistem manajemen basis data relasional yang ringan dan bersifat embedded, artinya tertanam langsung pada aplikasi tanpa memerlukan

server terpisah. Database ini sangat cocok untuk mendukung pengelolaan data yang sederhana, seperti menyimpan informasi log percakapan. *SQLite* mendukung akses paralel ke berbagai file database, menjadikannya ideal untuk aplikasi kecil hingga menengah. Selain itu, *SQLite* bersifat open-source, tidak memerlukan instalasi, dan dapat digunakan pada berbagai platform, sehingga cocok untuk pengelolaan log percakapan dalam aplikasi chatbot (Nugraha & A. Susetyo, 2023).

2.1.11 Draw.io

Draw.io adalah sebuah platform berbasis web yang dirancang untuk mempermudah pembuatan diagram *UML* secara online. Draw.io menawarkan kemudahan dalam penggunaan dan dilengkapi dengan antarmuka yang sederhana namun responsif, sehingga cocok digunakan oleh berbagai kalangan, mulai dari pelajar hingga profesional. Salah satu keunggulan utamanya adalah integrasi dengan layanan penyimpanan berbasis cloud seperti Google Drive, memungkinkan pengguna untuk menyimpan, mengakses, dan berbagi file diagram mereka kapan saja dan di mana saja.

Keberadaan Draw.io memberikan solusi praktis untuk menyusun diagram secara efisien tanpa memerlukan perangkat lunak tambahan. Dengan fitur-fitur yang mendukung produktivitas, platform ini membantu pengguna mempercepat proses penyusunan diagram *UML* sekaligus memastikan bahwa hasil yang diperoleh tetap profesional dan sesuai kebutuhan. Penggunaan Draw.io tidak hanya meningkatkan efisiensi waktu tetapi juga memberikan alternatif hemat biaya dibandingkan dengan aplikasi berbayar lainnya.

Dengan alat ini, pengguna dapat mengatasi berbagai tantangan, termasuk keterbatasan waktu dan pengetahuan teknis dalam membuat diagram. Selain itu, Draw.io juga mendukung berbagai format file, sehingga fleksibel untuk diintegrasikan dengan alat atau perangkat lunak lainnya. Dengan semua keunggulan ini, Draw.io menjadi pilihan yang ideal untuk mendukung berbagai kebutuhan pembuatan diagram, baik untuk keperluan pendidikan, bisnis, maupun proyek lainnya (Noneng Marthiawati et al., 2024).

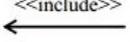
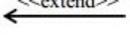
2.1.12 UML

Unified Modeling Language (UML) merupakan salah satu alat bantu yang sangat andal dalam dunia pengembangan sistem berbasis pemrograman berorientasi objek. Keunggulan *UML* terletak pada kemampuannya menyediakan berbagai jenis diagram pemodelan yang mendukung proses analisis, perancangan, dan dokumentasi sistem secara efektif. Dengan menggunakan *UML*, pengembang dapat menciptakan representasi visual yang terstruktur, sehingga mempermudah komunikasi ide dan rancangan antar anggota tim pengembang. Selain itu, *UML* membantu dalam menggambarkan hubungan antar komponen sistem, alur kerja, serta interaksi pengguna, menjadikannya alat yang penting untuk memastikan sistem yang dirancang sesuai kebutuhan dan lebih mudah dipahami oleh berbagai pemangku kepentingan (Voutama, 2022). Berikut di bawah ini merupakan diagram *UML* yang sering dipakai dalam pengembangan sistem.

Use Case

Use case merupakan kumpulan skenario yang terorganisasi untuk membantu pengguna mencapai tujuan tertentu dalam sebuah sistem. *Use case* berfungsi sebagai alat yang membantu pengembang perangkat lunak memahami pola interaksi antara pengguna (aktor) dan sistem yang dirancang. Dalam sebuah sistem, seorang aktor dapat berpartisipasi dalam berbagai *use case*, sementara sebuah *use case* juga dapat melibatkan beberapa aktor yang bekerja secara bersamaan atau bergantian untuk mencapai tujuan sistem.

Hubungan antara *use case* dan aktor, atau antar *use case* itu sendiri, dapat direpresentasikan melalui beberapa jenis koneksi, seperti *include*, yang menggambarkan hubungan di mana sebuah *use case* mengikutsertakan perilaku dari *use case* lain; *extend*, yang menambahkan perilaku tambahan ke *use case* utama berdasarkan kondisi tertentu; serta *generalization*, yang menunjukkan hubungan hierarkis antara *use case* atau aktor yang lebih umum dan lebih spesifik. Semua hubungan ini memberikan fleksibilitas dan kejelasan dalam memodelkan interaksi kompleks, sehingga pengembang dapat merancang sistem yang lebih sesuai dengan kebutuhan pengguna (Setiyani, 2021).

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Gambar 2. 1 Elemen Use Case Diagram

Sumber: jagoanhosting.com

Activity Diagram

Activity diagram adalah salah satu jenis diagram yang digunakan untuk memvisualisasikan alur atau urutan proses dalam sebuah sistem. Misalnya, dalam konteks sistem manajemen salon, *activity* diagram dapat digunakan untuk menggambarkan secara rinci tahapan-tahapan proses seperti perawatan salon, konsultasi dengan dokter, dan pembelian produk. Diagram ini membantu mengidentifikasi langkah-langkah utama dalam setiap proses, termasuk pengambilan keputusan dan interaksi antar aktor atau komponen dalam sistem.

Dengan adanya *activity* diagram, pengembang maupun pemangku kepentingan dapat dengan mudah memahami bagaimana setiap proses bekerja secara terstruktur. Diagram ini juga bermanfaat untuk mengidentifikasi potensi hambatan atau inefisiensi dalam alur kerja, sehingga memudahkan perbaikan atau optimasi sistem. Selain itu, *activity* diagram memberikan pandangan yang jelas tentang keterkaitan antar proses, yang sangat membantu dalam pengembangan, implementasi, dan pemeliharaan sistem yang lebih efektif (Nabila et al., 2021).

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2. 2 Elemen Activity Diagram

Sumber: mavink.com

Sequence Diagram

Sequence diagram adalah alat pemodelan visual yang digunakan untuk membantu memahami kebutuhan atau persyaratan dalam pengembangan sistem baru. Diagram ini memberikan gambaran yang terperinci tentang urutan interaksi antara objek atau aktor dalam sistem, termasuk pesan yang dikirim dan diterima selama eksekusi (*runtime*). Dengan menggambarkan aliran komunikasi secara kronologis, *sequence* diagram membantu mendokumentasikan proses sistem secara sistematis.

Manfaat utama dari *sequence* diagram adalah kemampuannya untuk memvisualisasikan skenario teknis yang kompleks, memungkinkan pengembang dan pemangku kepentingan untuk melihat bagaimana sistem akan berperilaku dalam situasi tertentu. Diagram ini tidak hanya membantu dalam memahami proses yang ada, tetapi juga memungkinkan pengguna untuk memprediksi respons sistem terhadap berbagai kondisi atau permintaan. Selain itu, *sequence* diagram sangat efektif dalam mengidentifikasi potensi masalah atau konflik dalam desain sistem, sehingga mempermudah pengembang untuk mengoptimalkan performa dan efisiensi sistem sebelum implementasi (Rohmanto & Setiawan, 2022).

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

Gambar 2. 3 Elemen *Sequence* Diagram

Sumber: detik.com

Class Diagram

Class diagram adalah representasi visual yang menggambarkan hubungan antar kelas serta rincian detail masing-masing kelas dalam model desain suatu sistem. Diagram ini memuat informasi tentang atribut, metode, aturan, dan tanggung jawab setiap entitas yang berperan dalam menentukan perilaku sistem secara keseluruhan. Dengan kata lain, *class* diagram merupakan cerminan dari struktur sistem program yang diorganisasi berdasarkan jenis-jenis kelas yang terbentuk. *Class* diagram sering disebut sebagai kumpulan berbagai kelas beserta relasinya, di mana setiap kelas direpresentasikan dalam bentuk persegi panjang yang terdiri dari nama kelas, atribut, dan metode. Ketika kelas ini dibuat, ia akan

menghasilkan objek, menjadikannya bagian inti dalam pengembangan dan desain berbasis objek.

Dalam tahap analisis sistem, *class* diagram digunakan untuk menggambarkan aturan, tanggung jawab, serta perilaku entitas yang memengaruhi sistem. Pada tahap desain, diagram ini berperan penting dalam menangkap struktur lengkap dari semua kelas yang membangun arsitektur sistem. *Class* diagram tidak hanya menunjukkan relasi antar kelas seperti asosiasi, pewarisan, agregasi, dan komposisi, tetapi juga menggambarkan alur informasi yang akan diterapkan pada *database* sistem. Hal ini membuat *class* diagram menjadi alat penting dalam pengembangan sistem berbasis objek, memastikan elemen-elemen dalam sistem terdefinisi dengan baik dan sesuai kebutuhan (Ramdany et al., 2024).

SIMBOL CLASS DIAGRAM

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

Gambar 2. 4 Elemen Class Diagram

Sumber: mavink.com

2.2 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

No	Judul Penelitian	Nama Peneliti	Hasil Penelitian
1	A Literature Survey of Recent Advances in Chatbots	(Caldarini et al., 2022)	<i>Chatbot</i> terbukti meningkatkan efisiensi dan produktivitas karena kemampuannya untuk melayani banyak pengguna secara bersamaan.
2	A Review of AI-Driven Conversational Chatbots Implementation Methodologies and Challenges (1999–2022)	(Lin et al., 2023)	Dengan menggunakan <i>NLP</i> jawaban yang diberikan oleh <i>chatbot</i> menjadi lebih alami.
3	Penggunaan Chatbots dalam Meningkatkan Pengalaman Pelanggan pada Situs E-commerce	(Yanti, 2024)	<i>Chatbot</i> mampu memahami kebutuhan pengguna secara cepat dan memberikan solusi yang tepat.

4	Chatbot using Natural Language Processing (NLP) Techniques	(Sunny Kaushik & Rahul, 2023)	<i>NLP</i> menjadi alat untuk membuat <i>chatbot</i> menjadi lebih mudah untuk memahami dan memberikan respon terhadap pertanyaan pengguna.
5	View of Natural Language Processing in Chatbots_ A Review	(Bhupesh Patra & Mahendra Kumar, 2020)	<i>NLP</i> menjadikan <i>chatbot</i> menjadi lebih pintar, serbaguna dan efektif dalam berbagai aplikasi
6	CHATBOT USING NATURAL LANGUAGE PROCESS (NLP)	(Adarsh et al., 2023)	<i>Chatbot</i> ini terbukti dapat meningkatkan kepuasan pengguna karena layanan yang ditawarkan dapat diakses kapan saja dan memberikan solusi yang cepat serta akurat.
7	Layanan pelanggan berbasis Natural Language Processing melalui chatbot pada aplikasi pesan	(Ardiansyah & Sulaksono, 2023)	<i>Chatbot</i> dapat mengurangi waktu tunggu pelanggan dan memungkinkan untuk menjawab pertanyaan

			pengguna secara otomatis dan responsif
8	Chatbots and Customer Experience: Enhancing Engagement and Satisfaction	(Sahil Shetty, 2024)	<i>Chatbot</i> meningkatkan efisiensi dalam melayani pelanggan dikarenakan selalu tersedia selama 24 jam.
9	Penggunaan Chatbot Artificial Intelligence dan Pembangunan Karakter Mahasiswa: Sebuah Studi Empiris	(Prastyono et al., 2023)	<i>Chatbot</i> membantu dalam pembangunan karakter seseorang.
10	Penerapan Natural Language Processing (NLP) Di Bidang Pendidikan	(Rumaisa et al., 2021)	<i>NLP</i> dapat meningkatkan kualitas pengajaran bagi siswa dan membantu pendidik meningkatkan kualitas pembelajaran.
11	Chatbot Artificial Intelligence as Educational Tools in Science and Engineering Education: A Literature Review and Bibliometric	(Al Husaeni et al., 2024)	Penelitian ini menunjukkan bahwa penggunaan <i>chatbot</i> berbasis <i>AI</i> di pendidikan semakin diminati, dengan peningkatan signifikan

	Mapping Analysis with Its Advantages and Disadvantages		pada tahun 2023 (205 publikasi di Scopus). <i>Chatbot</i> mendukung pembelajaran inovatif, dan meningkatkan hasil belajar
12	Analisis dan perancangan aplikasi <i>chatbot</i> menggunakan <i>framework</i> RASA dan sistem informasi pemeliharaan aplikasi (studi kasus: <i>chatbot</i> penerimaan mahasiswa baru POLITEKNIK ASTRA)	(Laksmi Anindyati, 2022)	Hasil penelitian dalam jurnal ini menunjukkan bahwa pengembangan <i>chatbot</i> dapat mendukung interaksi real-time selama 24 jam.
13	Implementasi <i>Natural Language Processing</i> dalam perancangan aplikasi <i>chatbot</i> pada FIKTI UMSU	(Mubarok & Abdi, 2024)	Hasil penelitian dalam jurnal ini menunjukkan bahwa <i>chatbot</i> berbasis <i>Natural Language Processing (NLP)</i> mampu menjawab pertanyaan secara akurat berdasarkan data intents

			yang telah didefinisikan. <i>Chatbot</i> ini berhasil mengenali kata kunci dari input pengguna dan memberikan respons yang sesuai.
14	Implementasi <i>Chatbot</i> Berbasis <i>AI</i> dalam Meningkatkan Interaksi Pengguna pada Platform Web	(Nelfara Nasya & Wibowo, 2024)	Hasil penelitian menunjukkan bahwa implementasi chatbot berbasis <i>AI</i> pada platform web memberikan dampak positif dengan meningkatkan kepuasan pengguna hingga 75% dan menurunkan rata-rata waktu respons dari 3 menit menjadi 30 detik.
15	Chatbot <i>AI</i> dalam identifikasi awal gangguan kesehatan mental di Indonesia: tantangan dan prospek	(Khairan & Habib, 2024)	Penelitian ini menunjukkan bahwa <i>chatbot AI</i> berpotensi meningkatkan akses dan kualitas layanan

			<p>kesehatan mental di Indonesia, khususnya di daerah yang kekurangan tenaga profesional.</p> <p><i>Chatbot</i> dapat mendukung deteksi dini, memberikan dukungan awal, dan merekomendasikan tindakan lanjutan.</p>
--	--	--	---