

BAB 2

LANDASAN TEORI

1.1.1 Sistem

1.1.1.1 Definisi Sistem

Sistem adalah suatu kumpulan elemen atau komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu. Elemen-elemen ini bisa berupa orang, perangkat, prosedur, data, dan sumber daya lainnya yang terorganisir dalam suatu kerangka kerja yang teratur. Sistem bisa memiliki input (masukan), proses atau aktivitas yang dilakukan terhadap input tersebut, dan output (keluaran) yang merupakan hasil dari proses tersebut.

Sistem seringkali memiliki karakteristik seperti ketergantungan antar elemen, tujuan bersama, interaksi, dan adaptasi terhadap lingkungan atau perubahan kondisi. Sistem dapat ditemukan dalam berbagai konteks, seperti sistem komputer, sistem biologi, sistem ekonomi, sistem manajemen, dan banyak lagi. Definisi dan konsep sistem penting dalam memahami kompleksitas dan hubungan antar komponen dalam berbagai domain..[9]

1.1.1.2 Cara Kerja Sistem

Cara kerja suatu sistem dapat dijelaskan dalam beberapa langkah umum, meskipun detailnya bisa sangat bervariasi tergantung pada jenis sistem yang sedang dibicarakan. Berikut adalah langkah-langkah umum dalam cara kerja sistem:

1. **Input (Masukan):** Sistem menerima masukan atau input dari lingkungannya. Input ini bisa berupa data, informasi, energi, atau stimulus lainnya yang diperlukan agar sistem dapat beroperasi.
2. **Proses:** Setelah menerima input, sistem melakukan serangkaian aktivitas atau proses tertentu tergantung pada tujuannya. Proses ini melibatkan interaksi antara berbagai elemen di dalam sistem.

3. Pengolahan dan Transformasi: Proses ini melibatkan perubahan atau transformasi input menjadi bentuk atau hasil yang diinginkan. Pengolahan bisa meliputi komputasi, perubahan fisik, manipulasi data, dan sebagainya.
4. Interaksi Antar Elemen: Elemen-elemen dalam sistem berinteraksi satu sama lain untuk mencapai tujuan bersama. Interaksi ini bisa berupa pertukaran informasi, aliran energi, atau koordinasi aktivitas.
5. Output (Keluaran): Setelah melalui proses dan interaksi, sistem menghasilkan keluaran atau output yang merupakan hasil dari operasi sistem. Keluaran ini bisa berupa informasi, produk fisik, tindakan, atau efek lainnya.
6. Umpan Balik (Feedback): Beberapa sistem memiliki mekanisme umpan balik yang memungkinkan sistem untuk memantau hasilnya dan membandingkannya dengan tujuan yang diinginkan. Umpan balik ini dapat digunakan untuk mengatur atau mengoreksi proses berikutnya.[10]

1.1.1.3 Fungsi Sistem

Sistem memiliki berbagai fungsi yang bervariasi tergantung pada jenis sistem yang sedang dibicarakan dan tujuannya. Namun, secara umum, berikut adalah beberapa fungsi umum dari sistem:

1. Pencapaian Tujuan: Sistem dirancang untuk mencapai tujuan tertentu. Fungsi utama dari banyak sistem adalah mencapai hasil atau hasil akhir yang diinginkan. Sistem bekerja melalui berbagai proses dan interaksi untuk mencapai tujuan ini.
2. Organisasi dan Pengaturan: Sistem membantu mengatur dan mengelola elemen-elemen yang ada di dalamnya. Ini mencakup pengaturan tugas, peran, aliran informasi, dan sumber daya untuk mencapai tujuan dengan cara yang terstruktur dan terorganisir.
3. Pengolahan Informasi: Banyak sistem melibatkan pengolahan informasi, baik dalam bentuk data, pesan, atau input lainnya. Sistem komputer adalah contoh umum di mana data diambil, diproses, dan diubah menjadi informasi yang berarti.

4. **Distribusi Sumber Daya:** Sistem dapat digunakan untuk mendistribusikan sumber daya secara efisien. Ini bisa berupa distribusi energi, sumber daya manusia, atau sumber daya fisik lainnya dalam sistem yang lebih luas.
5. **Koordinasi dan Interaksi:** Sistem membantu mengkoordinasikan aktivitas dan interaksi antara elemen-elemen yang ada di dalamnya. Ini memungkinkan kerja sama dan sinergi di antara berbagai bagian sistem.
6. **Pemecahan Masalah:** Sistem dapat membantu dalam pemecahan masalah dengan menganalisis situasi, mengumpulkan informasi, dan menghasilkan solusi yang efektif atau keputusan yang bijaksana.[11]

1.1.2 Chatbot

1.1.2.1 Definisi Chatbot

Chatbot adalah sebuah program komputer yang bertujuan untuk mensimulasikan sebuah kecerdasan buatan untuk dapat melakukan sebuah percakapan dengan manusia.[6] Percakapan yang terjadi antara komputer dengan manusia merupakan bentuk respon dari program yang telah dideklarasikan pada database program pada komputer. Kemampuan komputer dalam menyimpan banyaknya data tanpa melupakan satu pun informasi yang disimpannya digabungkan dengan kepraktisan bertanya pada sumber informasi langsung dibandingkan dengan mencari informasi sendiri serta kemampuan learning yang dimilikinya menyebabkan chatbot adalah customer service yang handal.[12]

Teknologi chatbot merupakan salah satu bentuk aplikasi Natural Language Processing, NLP itu sendiri merupakan salah satu bidang ilmu Kecerdasan Buatan (Artificial Intelligence) yang mempelajari komunikasi antara manusia dengan komputer melalui bahasa alami. Model komputasi seperti ini berguna untuk memudahkan komunikasi antara manusia dengan komputer dalam hal pencarian informasi, sehingga dapat terjadi suatu interaksi antara keduanya dengan menggunakan bahasa alami.[13]

1.1.2.2 Cara Kerja Chatbot

Chatbot adalah program komputer yang dirancang untuk berinteraksi dengan manusia melalui percakapan teks atau suara. Cara kerja chatbot melibatkan berbagai komponen dan teknologi, tergantung pada jenis chatbot dan tingkat kompleksitasnya. Di bawah ini adalah gambaran umum tentang cara kerja chatbot:

1. Pemahaman Bahasa (Natural Language Understanding - NLU):

Chatbot pertama-tama harus memahami apa yang dikatakan oleh pengguna. Teknologi NLU digunakan untuk memproses masukan pengguna, menganalisis struktur kalimat, mengidentifikasi kata kunci, dan mengekstrak informasi penting. NLU membantu chatbot untuk menguraikan pesan pengguna ke dalam format yang dapat dipahami oleh sistem.

2. Pengolahan Informasi:

Setelah pesan pengguna diuraikan, chatbot memproses informasi tersebut. Ini bisa melibatkan pencocokan kata kunci dengan skrip atau aturan yang telah ditentukan sebelumnya. Pengolahan informasi ini membantu chatbot mengenali tujuan dari percakapan dan menentukan respons yang sesuai.

3. Logika dan Alur Percakapan:

Chatbot harus mengikuti alur percakapan yang masuk akal. Ini melibatkan logika untuk merespons berbagai skenario percakapan dengan cara yang koheren. Chatbot juga harus mampu menjaga konteks selama percakapan, mengingat informasi yang telah dibagikan oleh pengguna sebelumnya.

4. Pengambilan Keputusan:

Berdasarkan informasi yang diterima dari pengguna dan analisis internal, chatbot harus membuat keputusan tentang respons yang paling tepat. Ini bisa melibatkan memilih dari beberapa respons yang telah diprogram sebelumnya atau menghasilkan respons baru.

5. Generasi Teks (Natural Language Generation - NLG):

Setelah keputusan diambil, chatbot harus menghasilkan respons dalam bahasa manusia yang alami. Teknologi NLG digunakan untuk mengubah data atau keputusan dalam bentuk teks yang bisa dimengerti oleh pengguna.

6. Interaksi dengan Pengguna:

Respons yang dihasilkan oleh chatbot diteruskan ke pengguna. Ini adalah langkah akhir dalam siklus, dan pengguna dapat merespons kembali dengan pertanyaan atau pernyataan baru.

7. Pembelajaran Mesin (Machine Learning - ML):

Chatbot canggih menggunakan pembelajaran mesin untuk meningkatkan kinerjanya dari waktu ke waktu. Dengan menganalisis percakapan sebelumnya, chatbot dapat mengidentifikasi pola baru dan meningkatkan kemampuannya dalam memahami dan merespons percakapan.

Tergantung pada tujuan dan kompleksitas chatbot, teknologi yang digunakan dalam setiap langkah dapat berbeda. Ada chatbot yang lebih sederhana dengan aturan baku yang telah diprogram sebelumnya, sementara yang lain mungkin menggunakan pendekatan berbasis pembelajaran mesin yang lebih canggih untuk belajar dari interaksi dengan pengguna.[14]

1.1.2.3 Fungsi Chatbot

Chatbot memiliki berbagai fungsi dan aplikasi yang beragam tergantung pada konteksnya. Di bawah ini adalah beberapa fungsi umum dari chatbot:

1. Pelayanan Pelanggan:

Chatbot dapat digunakan untuk memberikan dukungan pelanggan secara cepat dan efisien. Mereka dapat menjawab pertanyaan umum, memberikan panduan langkah demi langkah, mengatasi masalah teknis sederhana, dan mengarahkan pengguna ke sumber daya yang tepat.

2. Pemasaran dan Penjualan:

Chatbot dapat membantu dalam mengarahkan prospek melalui proses pemasaran dan penjualan. Mereka dapat memberikan informasi produk, menjawab pertanyaan tentang harga dan fitur, serta membantu pengguna dalam mengambil keputusan pembelian.[15]

1.1.3 Line

1.1.3.1 Definisi Line

Line adalah sebuah aplikasi komunikasi lintas platform yang awalnya dikembangkan sebagai respons terhadap bencana gempa bumi dan tsunami di Jepang pada tahun 2011. Aplikasi ini memungkinkan pengguna untuk berkomunikasi melalui pesan teks, panggilan suara, panggilan video, dan berbagi berbagai jenis konten seperti foto, video, stiker, dan masih banyak lagi. Line juga menawarkan berbagai fitur tambahan, seperti saluran publik, permainan, layanan berita, dan layanan keuangan.

Line dikenal karena berbagai stiker (emoji besar) yang dapat digunakan dalam percakapan, dan ini telah menjadi salah satu fitur khasnya. Aplikasi ini sangat populer di berbagai negara, terutama di Asia, dan telah mendapatkan basis pengguna yang besar.

Selain fitur komunikasi, Line juga telah berkembang menjadi platform yang menyediakan layanan lainnya, termasuk toko daring untuk pembelian stiker, konten digital, dan barang-barang fisik, serta layanan dompet digital dan pembayaran di beberapa wilayah..[16]

1.1.3.2 Cara Kerja Line

Cara kerja aplikasi Line melibatkan berbagai fitur dan komponen yang bekerja bersama untuk menyediakan platform komunikasi dan berbagai layanan tambahan. Berikut adalah gambaran umum tentang cara kerja aplikasi Line:

1. Pendaftaran dan Instalasi:

Pengguna harus mengunduh dan menginstal aplikasi Line dari toko aplikasi yang sesuai dengan perangkat mereka, seperti App Store (iOS) atau Google Play Store (Android). Setelah instalasi selesai, pengguna perlu mendaftar dengan menggunakan alamat email, nomor telepon, atau akun media sosial yang valid.

2. Pemilihan Kontak dan Teman:

Setelah mendaftar, pengguna dapat mengimpor kontak dari buku telepon mereka atau mencari teman dengan menggunakan nama pengguna atau nomor telepon. Ini memungkinkan pengguna untuk menemukan dan terhubung dengan kontak yang sudah ada di Line.

3. Percakapan Pesan:

Pengguna dapat memulai percakapan dengan kontak mereka melalui pesan teks, stiker, gambar, dan video. Pesan-pesan ini dapat dikirim dalam mode teks, suara, atau video.

4. Panggilan Suara dan Video:

Line memungkinkan pengguna untuk melakukan panggilan suara dan video dengan kontak mereka. Pengguna dapat memilih untuk berbicara langsung atau melalui panggilan video.

5. Berbagi Konten:

Pengguna dapat berbagi berbagai jenis konten dalam percakapan mereka, seperti foto, video, catatan suara, lokasi, tautan, dan stiker. Line juga memiliki toko stiker yang memungkinkan pengguna untuk membeli dan menggunakan stiker unik dalam percakapan mereka.

6. Fitur Lainnya:

Line menawarkan berbagai fitur tambahan seperti Line Timeline, yang memungkinkan pengguna membagikan pembaruan status dan konten dengan teman-

teman mereka, serta fitur Grup yang memungkinkan beberapa orang untuk berkomunikasi dalam percakapan kelompok.[17]

1.1.4 Media Promosi

1.1.4.1 Definisi Media Promosi

Media promosi merujuk pada berbagai saluran atau platform yang digunakan untuk mengkomunikasikan pesan promosi atau iklan kepada khalayak target. Tujuan dari media promosi adalah untuk meningkatkan kesadaran, mempengaruhi preferensi, dan mendorong tindakan dari audiens yang dituju. Berbagai jenis media promosi digunakan untuk mencapai berbagai tujuan dan mencakup berbagai format pesan, seperti teks, gambar, audio, dan video. Beberapa contoh media promosi meliputi:

1. Media Cetak: Ini termasuk iklan di surat kabar, majalah, brosur, pamflet, poster, dan katalog. Media cetak memberikan keleluasaan dalam merancang pesan dengan detail dan mencapai audiens yang lebih tertarget.
2. Media Elektronik: Ini mencakup iklan di televisi, radio, dan bioskop. Iklan di media elektronik dapat mencapai audiens yang luas dalam waktu yang relatif singkat.
3. Media Digital: Ini adalah media promosi yang berkaitan dengan internet dan teknologi digital. Ini termasuk iklan di situs web, media sosial, platform streaming, email, dan mesin pencari. [18]

1.1.4.2 Cara Kerja Media Promosi

Ada berbagai jenis media promosi yang dapat digunakan, dan cara kerjanya dapat bervariasi tergantung pada jenis media yang dipilih dan tujuan promosi. Berikut adalah gambaran umum tentang cara kerja media promosi:

1. Iklan Televisi dan Radio:

Iklan televisi dan radio adalah cara yang efektif untuk mencapai audiens yang luas. Iklan ini biasanya berupa audio dan visual yang menarik perhatian penonton atau pendengar. Mereka disiarkan pada waktu tertentu di saluran televisi atau stasiun radio yang dituju, berdasarkan profil target audiens.

2. Iklan Cetak:

Iklan cetak muncul di surat kabar, majalah, brosur, pamflet, dan media cetak lainnya. Pesan dan gambar ditempatkan secara visual, dan iklan ini dapat dibaca berulang kali. Membantu dalam menyampaikan informasi rinci tentang produk atau layanan.

3. Iklan Online:

Iklan online mencakup berbagai bentuk, seperti iklan banner, iklan video, iklan berbayar per klik (PPC), dan iklan media sosial. Iklan online dapat disesuaikan dengan demografi, minat, dan perilaku target audiens. Pengukuran dan analisis yang mendalam memungkinkan pengiklan untuk melacak kinerja iklan dan mengoptimalkan strategi promosi.

4. Media Sosial:

Media sosial memungkinkan pengiklan untuk berinteraksi secara langsung dengan audiens. Konten promosi dapat berupa posting organik, iklan berbayar, atau kampanye kreatif yang melibatkan audiens. Keterlibatan dan interaksi dengan audiens dapat mempengaruhi persepsi merek dan menciptakan koneksi emosional.[19]

1.1.4.3 Fungsi Media Promosi

Media promosi memiliki berbagai fungsi penting dalam memasarkan produk, layanan, atau merek kepada khalayak target. Berikut adalah beberapa fungsi utama dari media promosi:

1. **Meningkatkan Kesadaran:** Salah satu fungsi utama media promosi adalah meningkatkan kesadaran di kalangan audiens potensial tentang produk, layanan, atau merek yang Anda tawarkan. Melalui media promosi, Anda dapat memperkenalkan produk Anda kepada orang-orang yang belum pernah mendengar tentangnya sebelumnya.

2. **Memotivasi Pembelian:** Media promosi dapat membantu memotivasi calon konsumen untuk melakukan pembelian. Dengan menyajikan manfaat, fitur, dan nilai unik dari produk atau layanan Anda, Anda dapat mendorong audiens untuk mengambil tindakan dan membeli apa yang Anda tawarkan.
3. **Pendekatan Target Tertentu:** Berbeda dari media tradisional yang lebih bersifat umum, media promosi sering kali memungkinkan Anda untuk mendekati audiens yang lebih tersegmentasi dan relevan. Anda dapat menyesuaikan pesan dan kampanye Anda agar lebih sesuai dengan preferensi, demografi, dan kebutuhan spesifik dari target Anda.
4. **Meningkatkan Penjualan:** Salah satu tujuan utama dari media promosi adalah untuk meningkatkan penjualan. Dengan cara menarik perhatian konsumen, menunjukkan nilai produk atau layanan Anda, dan memberikan insentif pembelian, Anda dapat merangsang pertumbuhan penjualan.
5. **Membangun Citra Merek:** Media promosi membantu dalam membangun citra merek yang positif di mata konsumen. Pesan dan gaya yang Anda sampaikan melalui media promosi dapat membentuk persepsi konsumen tentang nilai, kualitas, dan karakter merek Anda.
6. **Komunikasi Informasi:** Media promosi memungkinkan Anda untuk menyampaikan informasi penting tentang produk atau layanan Anda, seperti fitur, manfaat, cara penggunaan, harga, dan lainnya. Ini membantu konsumen potensial memahami produk Anda secara lebih baik sebelum mereka membuat keputusan pembelian.
7. **Menghadapi Persaingan:** Dalam lingkungan bisnis yang kompetitif, media promosi memungkinkan Anda untuk bersaing dengan pesaing Anda dengan cara menonjolkan keunggulan dan manfaat unik yang dimiliki produk atau layanan Anda.
8. **Mendukung Lini Waktu Penjualan:** Media promosi dapat digunakan untuk mendukung kampanye penjualan tertentu, seperti penjualan musiman, penawaran khusus, atau peluncuran produk baru. Ini membantu dalam menghasilkan lonjakan minat dan pembelian dalam periode waktu tertentu.

9. Mendorong Interaksi dan Keterlibatan: Media promosi, terutama dalam bentuk media sosial dan kampanye digital, dapat mendorong interaksi langsung antara merek dan konsumen. Ini menciptakan peluang untuk mengembangkan hubungan yang lebih dalam dan membangun komunitas di sekitar merek Anda.
10. Mengukur Kinerja: Media promosi sering dapat diukur dalam hal efektivitas dan pengembalian investasi. Dengan menganalisis data dan metrik yang terkait dengan kampanye promosi, Anda dapat mengevaluasi apakah upaya Anda berhasil atau perlu disesuaikan.

Kombinasi dari berbagai jenis media promosi, seperti iklan televisi, radio, media sosial, pemasaran konten, dan lainnya, dapat membantu Anda mencapai tujuan-tujuan ini sesuai dengan strategi pemasaran yang Anda pilih.[20]

1.1.5 Penjualan

1.1.5.1 Definisi Penjualan

Penjualan adalah proses atau tindakan menjual produk, layanan, atau barang kepada konsumen atau pelanggan dengan tujuan untuk mendapatkan pendapatan atau keuntungan. Ini melibatkan serangkaian langkah yang mencakup pendekatan, komunikasi, negosiasi, dan penutupan transaksi yang mengarah pada pertukaran barang atau layanan dengan nilai yang diinginkan oleh kedua belah pihak: penjual dan pembeli. Proses penjualan umumnya melibatkan langkah-langkah berikut:

Penjualan dapat dilakukan dalam berbagai cara, baik langsung (tatap muka) maupun tidak langsung (melalui telepon, online, atau saluran lainnya). Tujuannya adalah untuk memenuhi kebutuhan dan keinginan konsumen sambil mencapai tujuan keuntungan dan pendapatan perusahaan atau individu yang melakukan penjualan.[21]

Dalam kegiatan ekonomi, usaha/penjualan sering dimaknai sebagai bisnis. Bisnis menurut Kamus bahasa Indonesia adalah suatu kegiatan yang mengerahkan tenaga, pikiran, atau tubuh untuk mencapai suatu tujuan kerja (tindakan, prakarsa,

usaha, usaha) untuk mencapai sesuatu dan kegiatan di bidang perdagangan dengan maksud untuk mendapatkan keuntungan [22].

1.1.5.2 Cara Kerja Penjualan

Cara kerja penjualan melibatkan serangkaian langkah yang bertujuan untuk meyakinkan calon pembeli agar melakukan pembelian produk atau layanan. Berikut adalah langkah-langkah umum yang terlibat dalam proses kerja penjualan:

1. Penelitian dan Persiapan:

Identifikasi Target Pasar: Menentukan siapa target pasar atau calon pelanggan yang paling cocok untuk produk atau layanan yang ditawarkan. **Memahami Kebutuhan Pelanggan:** Mengumpulkan informasi tentang kebutuhan, masalah, dan preferensi calon pelanggan sehingga penjualan dapat dipersonalisasi.

2. Pendekatan Awal:

Memulai Kontak: Mendekati calon pelanggan melalui salam atau pengenalan yang ramah. **Membangun Koneksi:** Membuat hubungan awal yang positif dengan mendengarkan dan menunjukkan minat terhadap kebutuhan mereka.

3. Presentasi dan Penjelasan:

Menyajikan Nilai: Menyampaikan fitur dan manfaat produk atau layanan secara rinci, menghubungkannya dengan solusi untuk masalah atau kebutuhan calon pelanggan. **Membangun Argumen:** Menggunakan fakta, data, testimoni, dan informasi yang relevan untuk membuktikan nilai produk atau layanan.

4. Penanganan Keberatan dan Pertanyaan:

Mengatasi Keberatan: Menanggapi kekhawatiran atau keberatan calon pelanggan dengan penjelasan yang memadai dan solusi yang memungkinkan.

Menjawab Pertanyaan: Memberikan informasi tambahan untuk membantu calon pelanggan memahami produk atau layanan lebih baik.

5. Negosiasi:

Diskusi Harga dan Syarat: Jika perlu, melakukan negosiasi mengenai harga, syarat pembayaran, atau fitur tambahan. Menyampaikan Nilai Tambah: Memperjelas nilai tambahan yang ditawarkan untuk membantu membenarkan harga.

6. Penutupan:

Mengajukan Pertanyaan Penutup: Mengajukan pertanyaan yang mengarah pada kesepakatan atau tindakan pembelian, seperti "Apakah Anda ingin membelinya?" Meningkatkan Kepastian: Menghapus keraguan atau kekhawatiran yang mungkin muncul pada tahap penutupan.

7. Tindak Lanjut:

Konfirmasi Pembelian: Setelah pembeli setuju, mengonfirmasi detail pembelian, seperti jumlah, harga, dan metode pembayaran. Mengatur Pengiriman atau Implementasi: Jika relevan, mengatur pengiriman produk atau pelaksanaan layanan.

8. Pelacakan dan Pemeliharaan Hubungan:

Memantau Kepuasan Pelanggan: Memastikan bahwa pelanggan puas dengan produk atau layanan yang diberikan. Menjaga Hubungan: Melakukan tindak lanjut periodik, mengirimkan penawaran atau informasi terbaru, dan membangun hubungan jangka panjang.

9. Analisis dan Peningkatan:

Mengevaluasi Kinerja: Menganalisis hasil penjualan, mencari peluang perbaikan, dan memahami apa yang berhasil dan tidak berhasil dalam

penjualan. Adaptasi dan Perbaikan: Berdasarkan analisis, mengubah strategi, pendekatan, atau taktik untuk meningkatkan hasil penjualan di masa depan.

Cara kerja penjualan adalah proses yang dinamis dan fleksibel, di mana penjual harus dapat beradaptasi dengan kebutuhan dan respon calon pelanggan. Itu melibatkan keterampilan komunikasi yang baik, empati, pemahaman tentang produk atau layanan, dan kemampuan untuk membangun hubungan yang kuat dengan pelanggan.[23]

1.1.5.3 Tujuan Penjualan

Kemampuan perusahaan dalam menjual produknya menentukan keberhasilan dalam mencari keuntungan, apabila perusahaan tidak mampu menjual maka perusahaan akan mengalami kerugian. Adapun tujuan umum penjualan dalam perusahaan yaitu :

1. Tujuan yang dirancang untuk meningkatkan volume penjualan total atau meningkatkan penjualan produk- produk yang lebih menguntungkan.
2. Tujuan yang dirancang untuk mempertahankan posisi penjualan yang efektif melalui kunjungan penjualan reguler dalam rangka menyediakan informasi mengenai produk baru.
3. Menunjang pertumbuhan perusahaan[24]

2.1.6 Rumah Rasfur Bandung

Rumah Rasfur Bandung adalah sebuah perusahaan yang bergerak di bidang penjualan produk seperti Sajadah, karpet lipat, baju muslim dan masih banyak lagi. Rumah Rasfur Bandung berlokasi di Kota Bandung, tepatnya di komp, Jl. Kanayakan Blk. D No.50, Dago, Kecamatan Coblong, Kota Bandung, Jawa Barat 40135.

Perusahaan Rumah Rasfur Bandung memiliki akun E-Commers untuk menjual produk yang dijualnya, selain itu Perusahaan Rumh Rasfur Bandung juga menjual produknya secara offline.

2.1.7 Kecerdasan Buatan

2.1.7.1 Definisi Kecerdasan Buatan

Kecerdasan Buatan (Artificial Intelligence atau AI) mengacu pada kemampuan komputer atau mesin untuk meniru beberapa aspek kemampuan manusia yang terkait dengan pemahaman, pembelajaran, penalaran, interaksi sosial, dan adaptasi terhadap situasi yang berubah. Tujuan utama dari kecerdasan buatan adalah membuat komputer mampu melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia, seperti pemecahan masalah kompleks, pengambilan keputusan, pengenalan pola, dan bahkan pembelajaran mandiri. Kecerdasan Buatan mencakup berbagai subbidang dan pendekatan, termasuk:

1. **Machine Learning (Pembelajaran Mesin):** Pendekatan ini melibatkan penggunaan algoritma dan model statistik untuk mengajarkan komputer bagaimana memahami data dan belajar dari pengalaman sebelumnya. Ini termasuk pengenalan pola, klasifikasi, regresi, dan tugas-tugas lain yang melibatkan pembelajaran dari data.
2. **Deep Learning:** Sebuah cabang dari pembelajaran mesin yang fokus pada jaringan saraf tiruan (neural networks) dengan banyak lapisan (layer). Deep learning telah sangat sukses dalam pengenalan wajah, pengenalan suara, pengenalan bahasa alami, dan tugas-tugas lain yang memerlukan pemrosesan data yang sangat kompleks.
3. **Pengolahan Bahasa Alami (Natural Language Processing or NLP):** Teknik ini memungkinkan komputer untuk memahami, menginterpretasi, dan merespons bahasa manusia dalam bentuk yang lebih alami. Aplikasi NLP termasuk chatbot, analisis sentimen, terjemahan otomatis, dan lainnya.
4. **Visi Komputer (Computer Vision):** Memungkinkan komputer untuk memproses, menganalisis, dan memahami informasi visual dari dunia nyata, seperti gambar dan video. Ini digunakan dalam pengenalan objek, deteksi wajah, kendaraan otonom, dan banyak aplikasi lainnya.

5. Robotika: Integrasi kecerdasan buatan dalam robot fisik, memungkinkan mereka untuk berinteraksi dengan lingkungan dan melakukan tugas-tugas fisik yang kompleks.
6. Reinforcement Learning: Pendekatan di mana komputer belajar melalui percobaan dan kesalahan. Dalam hal ini, komputer mencoba tindakan berulang kali dalam suatu lingkungan untuk mencapai tujuan tertentu.

Kecerdasan Buatan telah memiliki dampak besar di berbagai bidang, seperti kesehatan, otomotif, finansial, manufaktur, dan banyak lagi. Meskipun AI telah membuat kemajuan signifikan, perkembangannya masih terus berlanjut, dan berbagai tantangan etika, hukum, dan sosial juga harus diatasi dalam mengimplementasikan teknologi ini.[25]

2.1.7.2 Cara Kerja Kecerdasan Buatan

Kecerdasan Buatan (Artificial Intelligence atau AI) bekerja dengan menggabungkan berbagai teknik, algoritma, dan model komputasional untuk memungkinkan komputer atau sistem untuk menyelesaikan tugas-tugas yang biasanya memerlukan kecerdasan manusia. Cara kerja kecerdasan buatan bervariasi tergantung pada jenis tugas yang ingin diselesaikan. Berikut adalah langkah-langkah umum yang terlibat dalam cara kerja kecerdasan buatan:

1. Pemrosesan Data:

Data yang relevan dan terstruktur harus dikumpulkan untuk pelatihan dan pengujian algoritma AI. Ini bisa berupa teks, gambar, suara, atau jenis data lainnya tergantung pada tugas yang akan diselesaikan.

2. Pemilihan Model atau Algoritma:

Terdapat berbagai model dan algoritma AI yang dapat dipilih berdasarkan jenis tugas. Misalnya, jika tugas melibatkan klasifikasi gambar, Anda mungkin akan menggunakan jaringan saraf tiruan (neural networks) dalam pendekatan deep learning.

3. Pelatihan Model:

Model atau algoritma AI harus dilatih menggunakan data pelatihan. Proses ini melibatkan memberikan contoh data yang sudah diberi label atau jawaban yang benar kepada model, sehingga model dapat belajar pola dan fitur dari data tersebut.

4. Pengujian dan Penyetelan:

Setelah model dilatih, ia diuji dengan data pengujian yang belum pernah dilihat sebelumnya. Hasil pengujian ini membantu mengevaluasi kinerja model dan mengidentifikasi apakah penyetelan tambahan diperlukan.

5. Peningkatan dan Optimisasi:

Berdasarkan hasil pengujian, model dapat ditingkatkan dengan penyetelan parameter atau optimisasi algoritma. Tujuannya adalah untuk meningkatkan akurasi dan kinerja model.

6. Implementasi dan Interaksi:

Setelah model dianggap cukup baik, ia dapat diimplementasikan dalam sistem nyata. Dalam beberapa kasus, seperti chatbot atau asisten virtual, interaksi langsung dengan pengguna menjadi bagian penting dari implementasi.

7. Pemantauan dan Pemeliharaan:

Setelah diimplementasikan, model AI perlu dipantau secara berkala untuk memastikan kinerjanya tetap baik. Data yang terus dikumpulkan juga dapat digunakan untuk meningkatkan model di masa depan.

8. Adaptasi dan Peningkatan Lanjutan:

Lingkungan dan kebutuhan dapat berubah seiring waktu. Oleh karena itu, model AI perlu diadaptasi dan ditingkatkan secara berkala agar tetap relevan dan efektif.

9. Belajar Terus Menerus (Jika Berlaku):

Dalam beberapa jenis AI, seperti sistem berbasis reinforcement learning, model terus belajar melalui interaksi dengan lingkungan. Model ini melakukan tindakan, melihat hasilnya, dan mengoptimalkan tindakan berdasarkan umpan balik yang diterima.

10. Evaluasi dan Analisis Kinerja:

Kinerja model AI dievaluasi berdasarkan metrik yang relevan dengan tugas yang harus dilakukan. Analisis kinerja membantu menilai sejauh mana AI berhasil mencapai tujuan yang ditetapkan.

Cara kerja kecerdasan buatan sangat bergantung pada jenis tugas, data yang tersedia, dan algoritma yang digunakan. Kecerdasan buatan memiliki fleksibilitas untuk menyelesaikan berbagai macam tugas, dari analisis data kompleks hingga pengenalan wajah atau bahkan mengemudi kendaraan otonom.[26]

2.1.7.3 Fungsi Kecerdasan Buatan

Kecerdasan Buatan (Artificial Intelligence atau AI) memiliki berbagai fungsi yang dapat diaplikasikan di berbagai bidang. Fungsi-fungsi ini mencakup pengembangan sistem dan teknologi yang mampu melakukan tugas-tugas tertentu dengan kemampuan yang menyerupai atau bahkan melebihi kecerdasan manusia. Berikut adalah beberapa fungsi utama kecerdasan buatan:

1. Analisis Data dan Prediksi:

AI digunakan untuk menganalisis dan mengolah data dalam skala besar untuk mengidentifikasi pola, tren, dan wawasan yang tidak terlihat dengan mudah oleh manusia. Hal ini membantu dalam membuat prediksi yang lebih akurat terkait pasar, perilaku konsumen, atau bahkan risiko medis.

2. Pengenalan Pola:

AI dapat digunakan untuk mengenali pola kompleks dalam data, seperti pengenalan wajah, tulisan tangan, atau suara. Aplikasi ini digunakan dalam teknologi pengenalan suara, kendaraan otonom, keamanan, dan lebih banyak lagi.

3. Pengolahan Bahasa Alami (NLP):

AI dapat memahami, menganalisis, dan menghasilkan bahasa manusia. Ini memungkinkan pengembangan asisten virtual, chatbot, terjemahan otomatis, analisis sentimen, dan aplikasi lain yang melibatkan bahasa manusia.

4. Rekomendasi dan Personalisasi:

AI dapat mengumpulkan data tentang preferensi dan perilaku pengguna untuk memberikan rekomendasi yang lebih relevan, seperti rekomendasi produk di platform e-commerce atau rekomendasi konten di platform streaming.

5. Otomatisasi Tugas Rutin:

AI dapat digunakan untuk mengotomatisasi tugas-tugas yang monoton dan berulang, seperti pengolahan data, pemrosesan formulir, atau tugas administratif lainnya.

6. Pengolahan Gambar dan Video:

AI memungkinkan komputer untuk memahami dan mengolah informasi visual, seperti pengenalan objek, deteksi pola, dan analisis visual lainnya.[27]

2.1.8 OOP (*Object Oriented Programming*)

2.1.8.1 Definisi OOP (*Object Oriented Programming*)

OOP (Object-Oriented Programming atau Pemrograman Berorientasi Objek) adalah paradigma pemrograman yang berfokus pada pengorganisasian dan pengelompokan kode ke dalam objek-objek yang merepresentasikan entitas-entitas nyata atau konsep-konsep dalam dunia nyata. Dalam OOP, objek adalah unit dasar yang menggabungkan data (atribut) dan fungsi (metode) yang berkaitan dengan objek tersebut ke dalam satu kesatuan yang koheren. Konsep utama dalam OOP adalah sebagai berikut:

1. Objek: Objek adalah instansi dari kelas yang memiliki atribut (data) dan metode (fungsi) yang dapat memanipulasi atribut tersebut. Objek merepresentasikan entitas atau konsep tertentu dalam dunia nyata.
2. Kelas: Kelas adalah blueprint atau template yang mendefinisikan struktur dan perilaku dari suatu tipe objek. Kelas mendefinisikan atribut apa yang dimiliki oleh objek tipe tersebut dan metode apa yang dapat dijalankan oleh objek tersebut.
3. Enkapsulasi: Konsep ini melibatkan penggabungan data (atribut) dan metode yang beroperasi pada data tersebut ke dalam satu entitas yang disebut objek. Enkapsulasi memungkinkan informasi dalam objek tetap terlindungi dan hanya dapat diakses melalui antarmuka yang telah ditentukan.
4. Pewarisan (Inheritance): Pewarisan memungkinkan pembuatan kelas baru (kelas turunan atau subclass) yang mewarisi atribut dan metode dari kelas yang sudah ada (kelas induk atau superclass). Ini memungkinkan untuk membagi kode yang dapat digunakan kembali dan mengatur hierarki kelas.
5. Polimorfisme: Polimorfisme memungkinkan suatu metode memiliki banyak implementasi yang berbeda dalam kelas-kelas yang berbeda. Ini memungkinkan pemanggilan metode dengan nama yang sama untuk menghasilkan perilaku yang berbeda tergantung pada kelas objek yang sesungguhnya.

Paradigma OOP digunakan untuk mengorganisasi dan mengelola kompleksitas dalam pengembangan perangkat lunak dengan memisahkan kode menjadi modul-modul yang berinteraksi satu sama lain melalui objek-objek. Ini membantu dalam pembuatan perangkat lunak yang lebih mudah dipahami, dikelola, dan dikembangkan, serta mendorong pendekatan yang lebih modular dan reusable dalam pengembangan perangkat lunak.[28]

2.1.8.2 Cara Kerja OOP (*Object Oriented Programming*)

Cara kerja OOP (Object-Oriented Programming) melibatkan penggunaan konsep-konsep fundamental dalam paradigma ini untuk mengorganisasi,

memanipulasi, dan berinteraksi dengan objek-objek yang merepresentasikan entitas-entitas dalam program. Berikut adalah langkah-langkah umum yang terlibat dalam cara kerja OOP:

1. **Pemahaman Konsep:**

Pahami konsep-konsep dasar dalam OOP, seperti objek, kelas, enkapsulasi, pewarisan, dan polimorfisme. Ini penting agar Anda dapat merancang dan mengimplementasikan program secara efektif.

2. **Pendefinisian Kelas:**

Pertama, Anda harus mendefinisikan kelas. Kelas adalah blueprint yang menentukan atribut dan metode yang akan dimiliki oleh objek yang dibuat dari kelas tersebut.

3. **Pembuatan Objek:**

Setelah kelas didefinisikan, Anda dapat membuat objek atau instansi dari kelas tersebut. Objek adalah entitas nyata yang memiliki atribut dan metode yang ditentukan oleh kelas.

4. **Atribut dan Metode:**

Objek memiliki atribut (variabel) yang merepresentasikan karakteristik atau data yang dimiliki objek tersebut. Objek juga memiliki metode (fungsi) yang merepresentasikan tindakan atau operasi yang dapat dilakukan oleh objek.

5. **Enkapsulasi:**

Prinsip enkapsulasi melibatkan pengaturan akses ke atribut dan metode dalam objek. Anda dapat menentukan apakah atribut atau metode tertentu dapat diakses dari luar objek atau hanya dapat diakses secara internal oleh objek itu sendiri.

6. **Pewarisan:**

Pewarisan memungkinkan Anda membuat kelas turunan yang mewarisi atribut dan metode dari kelas induk. Ini membantu dalam berbagi dan mengorganisasi kode yang dapat digunakan kembali.

7. Polimorfisme:

Polimorfisme memungkinkan Anda menggunakan metode dengan nama yang sama dalam kelas-kelas yang berbeda, tetapi dengan implementasi yang berbeda. Hal ini membantu dalam menghadirkan fleksibilitas dalam pemanggilan metode tergantung pada jenis objek yang digunakan.

8. Interaksi Antara Objek:

Objek-objek dapat berinteraksi satu sama lain melalui metode yang mereka eksekusi. Objek dapat mengirim pesan ke objek lain, mengubah status atributnya, atau meminta operasi tertentu.

9. Pemecahan Masalah:

Dalam pengembangan program, Anda dapat memecahkan masalah dengan memecahnya menjadi objek-objek yang merepresentasikan elemen-elemen masalah. Anda kemudian merancang metode yang tepat untuk objek-objek tersebut sehingga berinteraksi sesuai dengan kebutuhan.

10. Modularitas dan Reusabilitas:

Pendekatan OOP memungkinkan Anda untuk membuat kode yang lebih modular dan dapat digunakan kembali. Anda dapat menggunakan kelas yang sudah ada dalam berbagai proyek dan mempercepat proses pengembangan.

Cara kerja OOP memungkinkan pengembang untuk merancang program yang lebih terstruktur, mudah dimengerti, dan dapat dikelola dengan lebih baik, serta meningkatkan efisiensi dan fleksibilitas dalam pengembangan perangkat lunak.[29]

2.1.8.3 Fungsi OOP (*Object Oriented Programming*)

OOP (Object-Oriented Programming) adalah paradigma pemrograman yang berfokus pada pengorganisasian dan pengelompokan kode ke dalam objek-objek yang memiliki atribut (data) dan metode (fungsi) yang berkaitan. Tujuan utama dari OOP adalah untuk meningkatkan modularitas, reusabilitas, dan pemeliharaan kode. Berikut adalah beberapa fungsi atau manfaat dari OOP:

1. **Modularitas:** OOP memungkinkan kode program dibagi menjadi modul-modul kecil yang disebut objek. Setiap objek memiliki tanggung jawab tertentu dan dapat berinteraksi dengan objek lain melalui antarmuka yang didefinisikan dengan jelas. Ini membantu mengurangi kompleksitas dan mempermudah pengembangan dan pemeliharaan.
2. **Reusabilitas:** Dalam OOP, Anda dapat membuat objek yang dapat digunakan ulang dalam berbagai proyek. Ini menghemat waktu dan usaha karena Anda tidak perlu menulis ulang kode yang serupa. Objek-objek yang sudah ada dapat diambil dan dimodifikasi sesuai kebutuhan.
3. **Pemahaman yang Lebih Baik:** Dengan memodelkan dunia nyata ke dalam objek, kode menjadi lebih intuitif dan mudah dimengerti. Objek merepresentasikan entitas nyata atau konsep dalam domain yang sedang Anda kerjakan.

Secara keseluruhan, OOP membantu dalam mengelola kompleksitas proyek perangkat lunak dengan menyediakan struktur yang terorganisir dan memisahkan konsep-konsep yang berbeda.[30]

2.1.9 Use Case Diagram

2.1.9.1 Definisi Use Case Diagram

Diagram Use Case (Use Case diagram) adalah jenis diagram dalam rekayasa perangkat lunak yang digunakan untuk menggambarkan interaksi antara aktor-aktor (entitas di luar sistem) dan kasus penggunaan (use case) dalam sebuah sistem perangkat lunak. Diagram ini memberikan gambaran tingkat tinggi tentang cara pengguna atau

entitas lain berinteraksi dengan sistem dan bagaimana sistem merespons interaksi tersebut.

Diagram Use Case membantu dalam memvisualisasikan interaksi antara pengguna dan sistem, mengidentifikasi kasus penggunaan utama yang harus diperhatikan dalam pengembangan perangkat lunak, dan memberikan dasar untuk perancangan fungsionalitas sistem. Diagram ini biasanya merupakan bagian awal dalam analisis kebutuhan perangkat lunak dan membantu dalam komunikasi antara tim pengembang, pengguna, dan stakeholder lainnya.[31]

2.1.9.2 Cara Kerja Use Case Diagram

Cara kerja diagram Use Case (Use Case diagram) melibatkan langkah-langkah untuk merencanakan, mendefinisikan, dan memvisualisasikan interaksi antara aktor dan kasus penggunaan dalam sebuah sistem perangkat lunak.

Penting untuk diingat bahwa diagram Use Case adalah alat untuk komunikasi dan pemahaman yang kuat antara anggota tim dan stakeholder. Tujuannya adalah memberikan pandangan visual tentang interaksi pengguna dengan sistem dan bukan untuk menggambarkan detail teknis implementasi.[32]

2.1.9.3 Fungsi Use Case Diagram

Diagram Use Case memiliki beberapa fungsi penting dalam pengembangan perangkat lunak dan analisis kebutuhan. Beberapa fungsi utamanya adalah:

1. **Memahami Interaksi Pengguna dengan Sistem:** Diagram Use Case membantu tim pengembang dan stakeholder untuk memahami bagaimana pengguna atau aktor lain akan berinteraksi dengan sistem. Ini membantu dalam mengidentifikasi skenario utama yang harus diperhatikan dalam perancangan.
2. **Menentukan Kebutuhan Fungsional:** Diagram ini membantu dalam mengidentifikasi dan mendefinisikan kebutuhan fungsional sistem dengan cara yang lebih intuitif. Setiap kasus penggunaan merepresentasikan fungsi atau tindakan yang diinginkan oleh pengguna.

3. Mengidentifikasi Aktor dan Pengguna: Diagram Use Case membantu mengidentifikasi semua aktor yang berinteraksi dengan sistem, termasuk pengguna manusia dan entitas eksternal lainnya. Ini membantu dalam memahami siapa yang berpengaruh pada sistem.
4. Merencanakan Pengembangan dan Testing: Kasus penggunaan dalam diagram dapat membantu dalam perencanaan pengembangan dan testing. Setiap kasus penggunaan bisa dijadikan titik awal untuk merancang dan menguji fungsionalitas tertentu.
5. Komunikasi Antar Tim: Diagram ini berfungsi sebagai alat komunikasi yang kuat antara anggota tim pengembang, desainer, analis, dan stakeholder lainnya. Semua orang dapat dengan mudah memahami interaksi utama dalam sistem dan kebutuhan pengguna.

Secara keseluruhan, fungsi utama dari diagram Use Case adalah menyediakan pandangan yang jelas dan terstruktur tentang interaksi pengguna dengan sistem, serta mengidentifikasi kebutuhan fungsional utama yang harus diimplementasikan dalam perangkat lunak yang sedang dikembangkan.[33]

2.1.10 Class Diagram

2.1.10.1 Definisi Class Diagram

Diagram kelas (Class diagram) adalah jenis diagram dalam rekayasa perangkat lunak yang digunakan untuk menggambarkan struktur statis dari suatu sistem perangkat lunak. Diagram kelas memberikan gambaran tentang kelas-kelas yang ada dalam sistem, hubungan antara kelas-kelas tersebut, atribut-atribut yang dimiliki oleh kelas, serta metode-metode atau fungsi-fungsi yang dapat diakses oleh kelas.

Diagram kelas membantu dalam merencanakan struktur kode dan perancangan perangkat lunak. Mereka memungkinkan pengembang untuk memahami bagaimana kelas-kelas dalam sistem berinteraksi dan bergantung satu sama lain. Diagram ini juga membantu dalam komunikasi antara tim pengembang dan

stakeholder lainnya serta memberikan panduan tentang bagaimana sistem akan diorganisasi dan diimplementasikan.[34]

2.1.10.2 Cara Kerja *Class Diagram*

Cara kerja diagram kelas (Class diagram) melibatkan serangkaian langkah untuk merencanakan, merancang, dan memvisualisasikan struktur statis dari suatu sistem perangkat lunak. Berikut adalah langkah-langkah umum dalam cara kerja diagram kelas:

1. **Identifikasi Kelas:** Identifikasi semua kelas yang ada dalam sistem. Kelas bisa mewakili objek, entitas, atau konsep yang ada dalam domain yang sedang Anda kerjakan.
2. **Menentukan Atribut dan Metode:** Untuk setiap kelas, tentukan atribut-atribut yang dimiliki oleh kelas tersebut. Atribut adalah data yang dimiliki oleh kelas. Selanjutnya, tentukan metode-metode atau fungsi-fungsi yang bisa diakses oleh kelas tersebut.
3. **Menentukan Hubungan:** Tentukan hubungan antara kelas-kelas. Gunakan hubungan-hubungan seperti asosiasi, agregasi, komposisi, pewarisan, dan ketergantungan untuk menggambarkan interaksi antara kelas-kelas.
4. **Menggambar Diagram:** Gunakan alat pemodelan yang sesuai, seperti alat pemodelan UML, untuk menggambar diagram kelas. Pilih simbol yang tepat untuk merepresentasikan kelas, atribut, metode, dan hubungan.
5. **Menggambar Kelas:** Gambar bentuk-bentuk kelas di diagram. Setiap kelas biasanya direpresentasikan oleh persegi panjang yang dibagi menjadi tiga bagian: bagian atas untuk nama kelas, bagian tengah untuk atribut, dan bagian bawah untuk metode.
6. **Menambahkan Atribut dan Metode:** Di dalam masing-masing kelas, tambahkan atribut dan metode yang sudah Anda tentukan sebelumnya.
7. **Menghubungkan Kelas:** Hubungkan kelas-kelas yang berhubungan menggunakan hubungan-hubungan yang sesuai. Gunakan tautan asosiasi, garis panah, atau simbol khusus tergantung pada jenis hubungan.

8. Menggunakan Hubungan Lainnya: Jika diperlukan, gunakan hubungan-hubungan lain seperti agregasi, komposisi, pewarisan, dan ketergantungan untuk menggambarkan keterkaitan yang lebih kompleks antara kelas-kelas.
9. Menambahkan Deskripsi: Sertakan deskripsi atau catatan untuk setiap kelas dan hubungan untuk memberikan konteks lebih lanjut tentang hubungan dan peran masing-masing kelas.
10. Review dan Revisi: Setelah diagram selesai, lakukan review bersama tim pengembang dan stakeholder untuk memastikan diagram menggambarkan struktur sistem dengan akurat.
11. Penggunaan dan Pengembangan Lanjutan: Diagram kelas ini bisa digunakan sebagai panduan selama pengembangan perangkat lunak. Ini membantu dalam perancangan struktur kode dan organisasi sistem.
12. Pemeliharaan dan Perubahan: Selama pengembangan berlanjut, diagram ini dapat diperbarui untuk mencerminkan perubahan dalam struktur kelas atau hubungan antar kelas.

Diagram kelas membantu dalam merencanakan struktur perangkat lunak secara lebih formal dan mengilustrasikan bagaimana komponen-komponen utama dalam sistem berinteraksi. Diagram ini juga memberikan dasar bagi pengembang untuk merancang kode dan memahami arsitektur perangkat lunak secara menyeluruh.[35]

2.1.10.3 Fungsi *Class Diagram*

Diagram kelas (Class diagram) memiliki beberapa fungsi penting dalam rekayasa perangkat lunak dan pengembangan sistem. Berikut adalah beberapa fungsi utama dari diagram kelas:

1. Pemahaman Struktur Sistem: Diagram kelas membantu dalam memahami struktur dan komposisi keseluruhan dari sistem perangkat lunak. Ini memberikan gambaran tentang kelas-kelas yang ada, atribut, dan hubungan antara kelas-kelas tersebut.
2. Merencanakan Struktur Kode: Diagram ini memberikan panduan visual tentang bagaimana komponen-komponen perangkat lunak akan diorganisasi

dalam kode. Ini membantu pengembang dalam perencanaan kode yang lebih efisien dan terstruktur.

3. **Mengidentifikasi Atribut dan Metode:** Diagram kelas memungkinkan identifikasi dan dokumentasi atribut-atribut yang dimiliki oleh setiap kelas, serta metode-metode yang dapat diakses oleh kelas. Ini membantu dalam merinci fungsionalitas masing-masing kelas.
4. **Hubungan Antar Kelas:** Diagram ini menggambarkan hubungan antara kelas-kelas. Ini membantu pengembang untuk memahami bagaimana kelas-kelas berinteraksi satu sama lain, termasuk asosiasi, pewarisan, komposisi, agregasi, dan lainnya.[36]

2.1.11 Activity Diagram

2.1.11.1 Definisi Activity Diagram

Diagram Aktivitas (Activity diagram) adalah jenis diagram dalam rekayasa perangkat lunak yang digunakan untuk menggambarkan aliran kerja atau aliran aktivitas dalam suatu proses. Diagram ini membantu dalam memodelkan urutan aktivitas, keputusan, dan paralelisme yang terjadi dalam suatu sistem atau proses bisnis.

Diagram aktivitas membantu dalam memahami dan merancang alur kerja, proses bisnis, atau aliran aktivitas dalam suatu sistem. Ini berguna dalam mendokumentasikan, menganalisis, dan memodelkan interaksi dan tindakan yang terjadi dalam proses yang kompleks. Diagram ini juga dapat digunakan untuk merencanakan, mengoptimalkan, dan mengomunikasikan proses kepada tim pengembang dan stakeholder lainnya.[37]

Activity Diagram digunakan untuk menggambarkan aktivitas-aktivitas yang ada pada suatu sistem, yang merupakan gambaran dari aktivitas kasus yang ada pada Use Case Diagram. Diagram aktivitas untuk membangun pembelajaran bahasa Inggris Aplikasi terdiri dari diagram aktivitas login, diagram aktivitas kursus, diagram aktivitas pengujian dan diagram kegiatan evaluasi.[38]

2.1.11.2 Cara Kerja *Activity Diagram*

Cara kerja dari *activity diagram* yaitu menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses, Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis, Menggambarkan aliran paralel, bercabang dan bersamaan dari sistem. Dengan adanya *activity diagram* ini maka user akan lebih mudah dalam mengetahui mengenai alur dari sistem yang akan dibuat.[39]

2.1.11.3 Fungsi *Activity Diagram*

Cara kerja diagram aktivitas melibatkan serangkaian langkah untuk merencanakan, memodelkan, dan memvisualisasikan aliran kerja atau aktivitas dalam suatu proses.

1. Identifikasi Aktivitas Utama: Identifikasi aktivitas-aktivitas utama yang terlibat dalam proses yang akan dimodelkan. Aktivitas ini mewakili langkah-langkah utama dalam aliran kerja.
2. Menentukan Aliran Kerja: Tentukan urutan aliran kerja atau aktivitas. Mulailah dengan aktivitas awal dan tentukan langkah-langkah apa yang harus dilakukan secara berurutan.
3. Tambahkan Keputusan dan Cabang: Jika ada keputusan yang harus dibuat dalam aliran kerja, tambahkan node keputusan (decision) untuk menggambarkan cabang aliran yang mungkin terjadi berdasarkan kondisi tertentu.
4. Tambahkan Paralelisme: Jika ada aktivitas yang bisa dilakukan secara paralel, tambahkan node garpu (fork) untuk membagi aliran aktivitas ke jalur-jalur yang berbeda, dan tambahkan node penggabungan (join) untuk menggabungkan kembali jalur-jalur tersebut.
5. Tambahkan Aksi dan Tindakan: Untuk setiap langkah dalam aliran kerja, tambahkan aksi atau tindakan yang harus dilakukan. Ini bisa berupa tugas atau tindakan lain yang perlu diselesaikan.

6. Menghubungkan Node: Hubungkan node-node aktivitas, keputusan, garpu, dan lainnya dengan menggunakan panah atau tautan. Ini menunjukkan urutan jalannya aliran kerja.
7. Definisikan Kondisi: Jika ada keputusan, tentukan kondisi atau kriteria yang digunakan untuk memilih jalur yang sesuai. Ini dapat berupa kondisi yang harus dipenuhi atau nilai yang harus dipenuhi.
8. Menambahkan Node Akhir: Tentukan node akhir untuk menunjukkan di mana aliran kerja selesai. Ini bisa berupa node akhir normal (end) atau node akhir penghentian (terminate).
9. Menambahkan Node Pengiriman Pesan atau Tindakan Pemanggilan: Jika ada komunikasi antara objek-objek dalam sistem, tambahkan node pengiriman pesan atau tindakan pemanggilan untuk menggambarkan interaksi tersebut.
10. Menggambar Diagram: Gunakan alat pemodelan yang sesuai, seperti perangkat lunak UML, untuk menggambar diagram aktivitas. Anda dapat menggunakan simbol-simbol yang sesuai untuk merepresentasikan node dan aliran.
11. Menambahkan Deskripsi: Sertakan deskripsi atau catatan untuk setiap aktivitas atau tindakan untuk memberikan konteks lebih lanjut tentang apa yang dilakukan dalam langkah tersebut.
12. Review dan Revisi: Setelah diagram selesai, lakukan review bersama tim pengembang dan stakeholder untuk memastikan diagram menggambarkan alur kerja dengan akurat.
13. Penggunaan dan Analisis: Diagram aktivitas ini dapat digunakan untuk menganalisis alur kerja, mengidentifikasi potensi masalah, dan merencanakan perbaikan atau optimalisasi alur kerja.

Diagram aktivitas membantu dalam memahami dan merancang aliran kerja yang kompleks, serta memberikan panduan tentang bagaimana aktivitas-aktivitas berinteraksi dalam suatu proses. Diagram ini juga digunakan untuk komunikasi, dokumentasi, dan analisis dalam proses pengembangan perangkat lunak atau perancangan proses bisnis.[40]

2.1.12 Sequence Diagram

2.1.12.1 Definisi Sequence Diagram

Diagram urutan (Sequence diagram) adalah jenis diagram dalam rekayasa perangkat lunak yang digunakan untuk menggambarkan interaksi antara objek-objek atau entitas dalam suatu skenario tertentu. Diagram urutan menggambarkan urutan pesan yang dikirim antara objek-objek dan urutan eksekusi dari metode-metode yang dipanggil.

Diagram urutan membantu dalam memodelkan urutan interaksi antara objek-objek dalam sebuah skenario dan memberikan pandangan yang jelas tentang bagaimana pesan dan metode dipertukarkan dalam waktu. Diagram ini berguna dalam menggambarkan proses yang dinamis dalam perangkat lunak, berfokus pada hubungan antar objek dan aliran eksekusi pesan.[41]

2.1.12.2 Cara Kerja Sequence Diagram

Cara kerja diagram urutan (Sequence diagram) melibatkan langkah-langkah untuk merencanakan, memodelkan, dan memvisualisasikan interaksi antara objek-objek atau entitas dalam suatu skenario. Berikut adalah langkah-langkah umum dalam cara kerja diagram urutan:

1. Identifikasi Objek: Identifikasi objek-objek atau aktor yang berpartisipasi dalam skenario yang ingin Anda modelkan. Objek bisa berupa instance dari kelas dalam pemrograman berorientasi objek atau aktor-aktor luar seperti pengguna manusia.
2. Membuat Hidup (Lifeline): Untuk setiap objek atau aktor, gambar hidup (lifeline), yaitu garis vertikal yang merepresentasikan waktu hidup atau eksistensi objek selama skenario.
3. Menentukan Pesan dan Metode: Tentukan pesan-pesan yang akan dikirim antara objek-objek. Ini bisa berupa pemanggilan metode, permintaan informasi, atau respons. Identifikasi juga metode-metode yang terlibat dalam pesan-pesan tersebut.

4. Menggambar Pesan: Gambar panah horizontal untuk setiap pesan yang dikirim antara objek-objek. Panah tersebut menggambarkan urutan pesan yang dikirim dan diterima.
5. Menyusun Urutan: Atur pesan-pesan dalam urutan kronologis sesuai dengan bagaimana interaksi berlangsung dari awal hingga akhir skenario. Pesan yang dikirim lebih awal akan berada di bagian atas diagram.
6. Menambahkan Pesan Return: Jika ada pemanggilan metode sinkron, tambahkan pesan return (respons) untuk menunjukkan hasil dari pemanggilan metode yang dikirim kembali ke objek pemanggil.
7. Menambahkan Pesan Pengembalian Nilai: Jika ada pesan return yang mengembalikan nilai atau data, tambahkan pesan pengembalian nilai untuk menggambarkan respons dengan nilai tersebut.
8. Menambahkan Pesan Asinkron: Jika ada pesan asinkron, tambahkan pesan-pesan tersebut ke dalam diagram dengan tanda khusus yang menunjukkan sifat asinkron pesan.
9. Menambahkan Pesan Penghentian: Jika ada objek yang berhenti berinteraksi dalam skenario, tambahkan pesan penghentian untuk menunjukkan akhir hidup objek.
10. Menggunakan Atribut dan Data: Jika diperlukan, tambahkan atribut atau data yang dikirimkan dalam pesan-pesan. Ini membantu dalam memahami konten dan informasi yang dikirim.
11. Menambahkan Deskripsi: Sertakan deskripsi atau catatan untuk setiap pesan atau langkah dalam diagram untuk memberikan konteks lebih lanjut tentang apa yang terjadi.
12. Review dan Revisi: Setelah diagram selesai, lakukan review bersama tim pengembang dan stakeholder untuk memastikan diagram menggambarkan interaksi dengan akurat.
13. Analisis dan Perbaikan: Diagram urutan ini dapat digunakan untuk menganalisis alur interaksi dan mengidentifikasi potensi masalah atau peluang perbaikan dalam skenario yang dimodelkan.

Diagram urutan membantu dalam memvisualisasikan dan merancang interaksi dinamis antara objek-objek dalam sebuah skenario. Diagram ini digunakan untuk komunikasi, dokumentasi, dan analisis dalam proses pengembangan perangkat lunak atau perancangan interaksi sistem.[42]

2.1.12.3 Fungsi *Sequence Diagram*

Diagram urutan (Sequence diagram) memiliki beberapa fungsi penting dalam pengembangan perangkat lunak dan analisis sistem. Berikut adalah beberapa fungsi utama dari diagram urutan:

1. **Visualisasi Interaksi:** Diagram urutan membantu dalam visualisasi interaksi dan komunikasi antara objek-objek atau entitas dalam suatu skenario atau proses. Ini memberikan pandangan yang jelas tentang bagaimana pesan dan metode dipertukarkan dalam waktu.
2. **Pemodelan Dinamika:** Diagram ini membantu dalam memodelkan aspek dinamika sistem, dengan fokus pada urutan eksekusi pesan dan metode di antara objek-objek.
3. **Analisis Use Case:** Diagram urutan digunakan untuk menganalisis bagaimana objek-objek berinteraksi dalam skenario yang berkaitan dengan kasus penggunaan tertentu. Ini membantu dalam memahami alur kerja dan proses yang terlibat.
4. **Verifikasi Desain:** Diagram ini digunakan untuk memverifikasi desain perangkat lunak dan memastikan bahwa interaksi antara objek-objek sesuai dengan kebutuhan dan spesifikasi.
5. **Pengujian:** Diagram urutan dapat digunakan sebagai dasar untuk perencanaan pengujian fungsional, khususnya pada pengujian integrasi yang melibatkan interaksi antara komponen-komponen yang berbeda.[43]

2.1.13 Rest API

2.1.13.1 Definisi Rest API

REST API (Representational State Transfer Application Programming Interface) adalah suatu jenis antarmuka pemrograman yang digunakan untuk menghubungkan dan berinteraksi antara berbagai aplikasi atau sistem melalui jaringan. REST API memungkinkan aplikasi untuk berkomunikasi dan bertukar data dengan cara yang standar dan terstruktur.

Konsep utama dari REST API adalah representasi sumber daya (resource) sebagai entitas yang dapat diakses melalui URL (Uniform Resource Locator) dan dioperasikan menggunakan metode HTTP standar seperti GET, POST, PUT, dan DELETE. REST API didasarkan pada prinsip-prinsip REST, yang menyediakan pendekatan yang sederhana dan konsisten untuk membangun layanan berbasis web.

REST API banyak digunakan dalam pengembangan perangkat lunak, aplikasi web, dan sistem yang memerlukan interaksi dengan sumber daya melalui jaringan. Mereka memungkinkan aplikasi untuk berkomunikasi dengan sistem lain secara terstandarisasi dan fleksibel.[44]

2.1.13.2 Cara Kerja Rest API

Cara kerja REST API (Representational State Transfer Application Programming Interface) melibatkan serangkaian langkah yang memungkinkan aplikasi untuk berkomunikasi dan bertukar data dengan sistem melalui jaringan menggunakan protokol HTTP. Berikut adalah langkah-langkah umum dalam cara kerja REST API:

1. Identifikasi Sumber Daya (Resources): Pertama-tama, Anda harus mengidentifikasi sumber daya atau objek yang akan diakses atau dimanipulasi melalui API. Ini bisa berupa entitas seperti pengguna, postingan, produk, dan sebagainya.
2. Membuat Endpoint: Setiap sumber daya direpresentasikan oleh URL unik yang disebut endpoint. Endpoint ini akan menjadi titik masuk untuk

berinteraksi dengan sumber daya tertentu. Misalnya, jika Anda memiliki sumber daya "pengguna", endpoint-nya bisa menjadi /users.

3. Menggunakan Metode HTTP: REST API menggunakan metode HTTP seperti GET, POST, PUT, dan DELETE untuk berinteraksi dengan sumber daya.
4. Representasi Data: Data yang dikirimkan atau diterima melalui REST API biasanya dalam format JSON (JavaScript Object Notation) atau XML (eXtensible Markup Language). Data ini menggambarkan sumber daya atau tindakan yang ingin dilakukan.
5. Permintaan dan Respons: Klien (aplikasi yang menggunakan API) membuat permintaan HTTP ke endpoint yang sesuai dengan metode yang relevan. Permintaan ini dapat berisi parameter, data yang dikirim, atau header khusus.
6. Server Menanggapi: Server menerima permintaan, memprosesnya, dan menghasilkan respons HTTP. Respons ini biasanya berisi status kode (seperti 200 untuk sukses atau 404 untuk tidak ditemukan), data yang diminta atau hasil dari tindakan yang dilakukan.
7. Data Pertukaran: Data pertukaran antara klien dan server dapat berupa informasi pengguna, postingan, konfigurasi, dan lainnya, tergantung pada fungsi API.
8. Autentikasi dan Otorisasi: Terkadang, API memerlukan autentikasi (memastikan pengguna yang mengakses memiliki izin) dan otorisasi (memastikan pengguna memiliki hak akses yang diperlukan).
9. Hypermedia (HATEOAS): Prinsip HATEOAS memungkinkan API untuk memberikan tautan (link) kepada klien untuk menavigasi dan berinteraksi dengan sumber daya lain yang relevan.
10. Penggunaan Format Data: Klien dapat membaca respons dan menafsirkan data dalam format JSON atau XML sesuai dengan kebutuhan aplikasi.

11. Komersialisasi dan Analisis: Data yang diperoleh dari REST API dapat digunakan untuk analisis, pengembangan aplikasi, integrasi dengan sistem lain, atau tujuan bisnis lainnya.
12. Secara keseluruhan, REST API memungkinkan komunikasi antara aplikasi dan sistem lain melalui protokol HTTP, dengan mengikuti prinsip-prinsip REST seperti penggunaan endpoint, metode HTTP, representasi data, dan lainnya. API ini sangat penting dalam menghubungkan dan mengintegrasikan berbagai sistem dalam lingkungan yang lebih luas.[45]

2.1.13.3 Fungsi *Rest API*

REST API (Representational State Transfer Application Programming Interface) memiliki berbagai fungsi penting dalam pengembangan perangkat lunak, integrasi sistem, dan berbagai aspek lainnya dalam dunia teknologi. Berikut adalah beberapa fungsi utama dari REST API:

1. Integrasi Aplikasi: REST API memungkinkan berbagai aplikasi berbeda untuk berkomunikasi dan berinteraksi satu sama lain. Ini memungkinkan pengembang untuk mengintegrasikan berbagai sistem yang ada untuk bekerja bersama dan berbagi data.
2. Berbagi Data: REST API memungkinkan sistem untuk berbagi data dengan aman dan terstruktur melalui jaringan. Ini sangat penting dalam situasi di mana data harus diambil atau dimanipulasi oleh berbagai aplikasi atau perangkat.
3. Mengakses Layanan Eksternal: Dengan REST API, aplikasi dapat mengakses layanan eksternal atau sumber daya yang disediakan oleh penyedia pihak ketiga. Contohnya, aplikasi bisa menggunakan layanan pembayaran, pemetaan, otentikasi, dan lainnya.[46]

2.1.13.4 Definisi *Restful API*

RESTful API adalah jenis API (Application Programming Interface) yang mengikuti prinsip-prinsip dan arsitektur REST (Representational State Transfer). RESTful API dirancang untuk memungkinkan aplikasi untuk berkomunikasi dan bertukar data dengan cara yang terstandarisasi, terstruktur, dan mudah diakses melalui jaringan.

RESTful API banyak digunakan dalam pengembangan perangkat lunak, aplikasi web, layanan web, dan sistem lainnya yang memerlukan antarmuka untuk berkomunikasi dan bertukar data. API ini memberikan pendekatan yang terstandarisasi dan fleksibel untuk interaksi sistem dalam lingkungan yang terdistribusi.[44]

2.1.14 Perangkat Lunak

2.1.14.1 Definisi Perangkat Lunak

Perangkat lunak, atau disebut juga perangkat keras lunak (software), adalah kumpulan program komputer, skrip, dan instruksi yang ditulis untuk mengendalikan dan mengkoordinasi perilaku perangkat keras komputer serta memberikan berbagai fungsionalitas kepada pengguna. Perangkat lunak adalah komponen inti dalam operasi dan penggunaan komputer dan perangkat elektronik modern.

Dalam konteks yang lebih luas, perangkat lunak mencakup semua instruksi, kode, dan data yang menjalankan fungsi tertentu dalam sistem komputer, mulai dari sistem operasi hingga aplikasi khusus. Ini termasuk berbagai jenis perangkat lunak seperti sistem operasi, perangkat lunak aplikasi, perangkat lunak utilitas, dan lainnya.[47]

2.1.14.2 Cara Kerja Perangkat Lunak

Perangkat lunak (software) bekerja dengan mengikuti serangkaian langkah dan proses yang telah ditentukan dalam kode program. Cara kerja perangkat lunak bervariasi tergantung pada jenisnya, apakah itu sistem operasi, perangkat lunak

aplikasi, atau jenis lainnya. Berikut adalah gambaran umum tentang cara kerja perangkat lunak:

1. **Kompilasi atau Interpretasi Kode:** Perangkat lunak umumnya ditulis dalam bahasa pemrograman. Pertama, kode yang ditulis harus dikompilasi (dalam beberapa kasus) atau diinterpretasikan menjadi instruksi yang dapat dimengerti oleh komputer. Proses ini mengubah kode manusia menjadi kode mesin yang dapat dijalankan oleh komputer.
2. **Eksekusi Program:** Setelah kode telah dikompilasi atau diinterpretasikan, perangkat lunak siap untuk dieksekusi. Ini berarti komputer akan membaca dan menjalankan instruksi-instruksi dalam program sesuai dengan urutan yang telah ditentukan.
3. **Memuat dalam Memori:** Program yang dieksekusi dimuat dalam memori komputer. Instruksi-instruksi dan data yang diperlukan disimpan dalam memori untuk diakses oleh CPU (Central Processing Unit) dan komponen lainnya.[48]

2.1.14.3 Fungsi Perangkat Lunak

Perangkat lunak memiliki berbagai fungsi penting dalam dunia komputasi dan teknologi. Fungsi-fungsi ini mencakup berbagai aspek, mulai dari mengelola perangkat keras hingga menyediakan fungsionalitas kepada pengguna akhir. Berikut adalah beberapa fungsi utama dari perangkat lunak:

1. **Mengelola Perangkat Keras:** Perangkat lunak, khususnya sistem operasi, mengelola dan mengontrol perangkat keras komputer. Ini termasuk alokasi sumber daya seperti CPU, RAM, penyimpanan, dan perangkat input/output.
2. **Memberikan Antarmuka Pengguna:** Perangkat lunak menyediakan antarmuka yang memungkinkan pengguna berinteraksi dengan komputer atau perangkat elektronik. Antarmuka ini bisa berupa GUI (Graphical User Interface) atau CLI (Command Line Interface).
3. **Menjalankan Aplikasi dan Tugas:** Perangkat lunak aplikasi memungkinkan pengguna untuk menjalankan berbagai tugas atau aplikasi seperti pengolahan

kata, lembar kerja, perangkat lunak desain grafis, permainan, dan banyak lagi.[49]

2.1.15 *Blackbox Testing*

2.1.15.1 Definisi *Blackbox Testing*

Black-box testing adalah metode pengujian perangkat lunak yang fokus pada pengujian fungsionalitas dari luar, tanpa memperhatikan rincian implementasi internal atau struktur kode program. Dalam black-box testing, pengujian dilakukan dengan melihat perangkat lunak sebagai kotak hitam di mana input diberikan dan output yang dihasilkan dievaluasi tanpa memperhatikan bagaimana perangkat lunak mencapai hasil tersebut.

Pendekatan ini bertujuan untuk mengidentifikasi kesalahan, cacat, atau ketidakcocokan dalam perangkat lunak berdasarkan spesifikasi fungsionalitas, dan bukan berdasarkan cara perangkat lunak diimplementasikan. Black-box testing umumnya dilakukan oleh tim pengujian yang tidak memiliki pengetahuan mendalam tentang kode sumber atau struktur internal perangkat lunak.[50]

2.1.15.2 Cara Kerja *Blackbox Testing*

Cara kerja black-box testing melibatkan serangkaian langkah untuk menguji fungsionalitas perangkat lunak dari luar, tanpa pengetahuan tentang implementasi internalnya. Berikut adalah langkah-langkah umum dalam cara kerja black-box testing:

1. Analisis Spesifikasi: Tim pengujian memeriksa dan memahami spesifikasi fungsional perangkat lunak yang akan diuji. Spesifikasi ini mencakup detail tentang fitur, fungsi, dan perilaku yang diharapkan dari perangkat lunak.
2. Identifikasi Kasus Uji: Tim pengujian mengidentifikasi kasus uji yang mencakup berbagai skenario penggunaan dan kondisi yang berbeda. Setiap kasus uji didasarkan pada spesifikasi fungsional dan tujuan pengujian.
3. Pembuatan Data Uji: Data uji yang diperlukan untuk menjalankan kasus uji dibuat. Ini mencakup input yang akan diberikan kepada perangkat lunak dan kondisi awal yang diperlukan.

4. Eksekusi Kasus Uji: Kasus uji dieksekusi dengan memberikan input yang telah ditentukan. Perangkat lunak menjalankan operasi yang sesuai dan menghasilkan output.
5. Pemantauan dan Evaluasi: Tim pengujian memantau perilaku perangkat lunak dan mengevaluasi output yang dihasilkan. Hasil diuji dibandingkan dengan hasil yang diharapkan berdasarkan spesifikasi.[51]

2.1.15.3 Fungsi *Blackbox Testing*

Black-box testing memiliki beberapa fungsi penting dalam siklus pengembangan perangkat lunak dan memastikan bahwa perangkat lunak berperilaku sesuai dengan harapan dan spesifikasi fungsional. Berikut adalah beberapa fungsi utama dari black-box testing:

1. Memastikan Fungsionalitas yang Benar: Black-box testing membantu memastikan bahwa perangkat lunak berperilaku sesuai dengan spesifikasi fungsional yang telah ditetapkan. Ini membantu memastikan bahwa perangkat lunak memberikan hasil yang diharapkan kepada pengguna.
2. Verifikasi vs. Spesifikasi: Fungsi utama black-box testing adalah memverifikasi apakah perangkat lunak telah mematuhi spesifikasi yang telah ditetapkan. Pengujian ini membantu dalam mengidentifikasi jika ada kesalahan atau penyimpangan dari spesifikasi yang diberikan.
3. Identifikasi Kesalahan dan Cacat: Melalui pengujian berbagai kasus uji, black-box testing membantu dalam mengidentifikasi kesalahan, cacat, atau kesalahan logika dalam perangkat lunak yang mungkin mempengaruhi fungsionalitasnya.[52]

2.1.16 *MySQL*

2.1.16.1 Definisi *MySQL*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau DBMS yang multialur, multipengguna, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL* tersedia untuk beberapa platform, di antara nya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap

MySQL, anda dapat menggunakan software tertentu, di antara nya adalah *phpmyadmin* dan *mySQL yog*. [53]

MySQL atau program *multithread*, *MySQL* mendukung beberapa pemrograman bahasa seperti *c++*, *java*, dan *PHP*, untuk sistem operasi *windows* *MySQL* memiliki database terbuka konektivitas yang merupakan antarmuka pemrograman aplikasi (API) standar untuk mengakses database sistem manajemen (DBMS) [54].

2.1.16.2 Cara Kerja *MySQL*

MySQL adalah sistem manajemen basis data (DBMS) yang bersifat open source dan digunakan secara luas untuk menyimpan, mengelola, dan mengakses data dalam basis data. Berikut adalah cara kerja *MySQL* secara umum:

1. Pengaturan Server *MySQL*: *MySQL* bekerja sebagai server basis data yang berjalan di komputer atau server fisik. Server ini memiliki konfigurasi dan pengaturan tertentu yang harus ditentukan sebelum dapat digunakan. Pengguna biasanya menggunakan perintah dan konfigurasi untuk mengelola server *MySQL*.
2. Koneksi ke Server: Aplikasi atau klien menghubungkan diri ke server *MySQL* menggunakan protokol komunikasi tertentu, seperti *TCP/IP*. Koneksi ini dibuat melalui port yang telah ditentukan.
3. Autentikasi: Setelah koneksi terhubung, server akan memeriksa identitas klien dengan memvalidasi username dan password yang diberikan. Jika autentikasi berhasil, klien diizinkan untuk mengakses basis data yang sesuai.
4. Eksekusi Perintah *SQL*: Klien dapat mengirimkan perintah *SQL* ke server *MySQL*. Perintah ini bisa berupa perintah untuk mengambil data (*SELECT*), menyisipkan data (*INSERT*), memperbarui data (*UPDATE*), menghapus data (*DELETE*), atau perintah lain yang berhubungan dengan manajemen basis data.
5. Optimasi dan Eksekusi: Setelah menerima perintah *SQL*, server *MySQL* akan mengoptimalkan perintah tersebut untuk mendapatkan hasil yang seefisien

mungkin. Ini melibatkan pemilihan indeks yang sesuai, penggunaan cache, dan berbagai teknik lainnya. Setelah perintah dioptimalkan, server akan menjalankannya pada basis data yang sesuai.[55]

2.1.16.3 Fungsi *MySQL*

MySQL memiliki berbagai fungsi penting sebagai sistem manajemen basis data relasional (RDBMS). Fungsi-fungsi ini memungkinkan pengguna untuk mengelola, menyimpan, dan mengakses data dengan efisien. Berikut adalah beberapa fungsi utama MySQL:

1. Penyimpanan Data: Fungsi paling dasar dari MySQL adalah menyimpan data dalam basis data. Data disimpan dalam bentuk tabel yang memiliki kolom-kolom dengan tipe data yang sesuai.
2. Pengambilan Data: MySQL memungkinkan pengguna untuk mengambil data dari basis data dengan menggunakan perintah SQL seperti SELECT. Ini memungkinkan untuk melakukan pencarian data yang spesifik atau menampilkan seluruh isi tabel.
3. Pembaruan dan Penambahan Data: Pengguna dapat memperbarui data yang ada atau menambahkan data baru ke dalam tabel menggunakan perintah UPDATE, INSERT, dan DELETE.
4. Manipulasi Data: MySQL mendukung berbagai operasi manipulasi data seperti penggabungan (JOIN) tabel, pengelompokan (GROUP BY), pengurutan (ORDER BY), dan fungsi matematika.[56]

2.1.17 *Webhook*

2.1.17.1 Definisi *Webhook*

Webhook adalah sebuah mekanisme di dalam perangkat lunak yang memungkinkan satu aplikasi atau layanan untuk secara otomatis mengirim data atau pemberitahuan ke aplikasi atau layanan lain ketika suatu peristiwa atau kejadian tertentu terjadi. Webhook adalah cara bagi sistem untuk berkomunikasi secara real-time dan memberikan informasi yang relevan kepada sistem lain tanpa perlu polling (pengecekan berulang) secara terus-menerus.

Dalam konsepnya, webhook berfungsi seperti panggilan telepon yang diterima oleh sistem tujuan ketika peristiwa yang ditunggu terjadi. Misalnya, dalam konteks pengiriman email, sebuah sistem layanan email dapat mengirim webhook kepada aplikasi lain ketika sebuah email baru diterima atau dikirim.[57]

2.1.17.2 Cara Kerja *Webhook*

Cara kerja webhook melibatkan serangkaian langkah yang memungkinkan suatu aplikasi atau layanan untuk memberi tahu aplikasi lain tentang peristiwa tertentu yang terjadi.

Contoh konkret penggunaan webhook adalah ketika sebuah layanan pembayaran online ingin memberi tahu toko daring tentang konfirmasi pembayaran. Toko daring akan mengonfigurasi webhook dengan URL endpoint yang siap menerima panggilan dari layanan pembayaran. Ketika pembayaran sukses, layanan pembayaran akan mengirimkan pemberitahuan melalui panggilan HTTP POST ke URL endpoint toko daring. Toko daring kemudian akan mengambil data dari pemberitahuan tersebut dan memperbarui status pemesanan atau mengirim email konfirmasi kepada pelanggan. Penting untuk memahami format data yang diharapkan dan cara memprosesnya sesuai dengan dokumentasi webhook yang disediakan oleh sumber pemberitahuan.

2.1.18 *Messaging API*

2.1.18.1 Definisi *Messaging API*

Messaging API (Application Programming Interface) adalah serangkaian protokol, aturan, dan instruksi yang memungkinkan aplikasi untuk berkomunikasi dan bertukar pesan dengan aplikasi atau sistem lain. Messaging API memungkinkan pengembang untuk mengintegrasikan kemampuan komunikasi langsung dalam aplikasi mereka, baik untuk mengirim pesan teks, gambar, video, atau bentuk komunikasi lainnya.

Messaging API biasanya digunakan dalam berbagai konteks, seperti aplikasi pesan instan, platform media sosial, aplikasi bisnis, dan banyak lagi. Ini

memungkinkan pengembang untuk memanfaatkan fitur-fitur komunikasi yang ada dalam platform lain atau mengembangkan solusi komunikasi kustom sesuai kebutuhan.[58]

2.1.18.2 Cara Kerja *Messaging API*

Cara kerja Messaging API melibatkan sejumlah langkah yang memungkinkan aplikasi atau sistem untuk berkomunikasi dan bertukar pesan dengan aplikasi atau sistem lain. Berikut adalah langkah-langkah umum tentang bagaimana Messaging API bekerja:

1. **Registrasi dan Konfigurasi:** Pengembang harus mendaftar dan mengonfigurasi aplikasi mereka dengan platform atau layanan yang menyediakan Messaging API. Ini biasanya melibatkan pembuatan akun pengembang, pemberian izin akses, dan pengaturan parameter yang diperlukan.
2. **Membuat API Key atau Token:** Setelah aplikasi terdaftar dan dikonfigurasi, pengembang akan mendapatkan API key atau token yang diperlukan untuk mengakses Messaging API. API key atau token ini adalah tanda pengenal yang digunakan untuk mengotentikasi permintaan API.
3. **Memilih Jenis Pesan:** Pengembang memilih jenis pesan yang ingin dikirim melalui Messaging API. Ini bisa berupa teks, gambar, video, audio, pemberitahuan, dan lainnya.
4. **Membuat Permintaan API:** Pengembang membuat permintaan API menggunakan API key atau token yang telah diberikan. Permintaan ini biasanya berupa permintaan HTTP dengan menggunakan metode POST atau metode lain yang sesuai.
5. **Menyusun Pesan:** Pengembang menyusun pesan yang ingin dikirim melalui API. Pesan ini bisa berupa teks, gambar, atau format lain yang sesuai dengan jenis pesan yang dipilih.
6. **Mengirim Permintaan API:** Permintaan API yang berisi pesan disampaikan ke URL endpoint yang ditentukan oleh platform atau layanan yang menyediakan Messaging API.

7. Proses Oleh Penerima: Platform atau layanan yang menerima permintaan API akan memproses pesan yang diterima. Ini mungkin melibatkan penyaringan pesan, verifikasi pengirim, dan validasi konten.
8. Kirim Balasan atau Tanggapan: Setelah pesan diproses, penerima Messaging API dapat mengirimkan balasan atau tanggapan kembali kepada aplikasi pengirim melalui permintaan HTTP yang dikirimkan ke URL endpoint yang telah dikonfigurasi oleh pengembang.
9. Terima dan Proses Balasan: Aplikasi pengirim menerima balasan atau tanggapan dari penerima Messaging API. Aplikasi ini akan memproses balasan dan melakukan tindakan sesuai dengan konteks.
10. Interaksi Lanjutan: Interaksi melalui Messaging API bisa berlanjut dengan mengirim pesan-pesan tambahan, menerima notifikasi, atau melakukan operasi lain yang sesuai dengan fitur-fitur yang diberikan oleh Messaging API.[59]

2.1.18.3 Fungsi *Messaging API*

Messaging API memiliki beberapa fungsi penting dalam dunia komunikasi modern, memungkinkan aplikasi dan sistem berinteraksi secara real-time dan mengirimkan informasi kepada pengguna dengan efisien. Berikut adalah beberapa fungsi utama dari Messaging API:

1. Komunikasi Real-Time: Messaging API memungkinkan komunikasi yang instan dan real-time antara aplikasi atau sistem. Pengguna dapat menerima dan mengirim pesan tanpa perlu menunggu atau memeriksa secara manual.
2. Interaksi Pelanggan: Messaging API memungkinkan bisnis berinteraksi dengan pelanggan mereka dalam berbagai konteks, seperti dukungan pelanggan, pemberitahuan pembayaran, pembaruan pesanan, dan lain-lain.
3. Pengiriman Pemberitahuan: Messaging API digunakan untuk mengirimkan pemberitahuan kepada pengguna tentang peristiwa yang penting atau informasi terbaru. Ini bisa berupa pemberitahuan melalui pesan teks, email, atau notifikasi push.[60]

2.1.20 Interaktif

2.1.20.1 Definisi Interaktif

Interaktif adalah sebuah konsep yang mengacu pada kemampuan suatu sistem atau lingkungan untuk memungkinkan komunikasi, pertukaran, atau respons antara individu atau pengguna dengan sistem tersebut. Dalam konteks ini, interaktif menggambarkan kemampuan suatu entitas untuk merespons aksi atau input dari pengguna dan memberikan umpan balik yang sesuai.

Interaktif sering kali terkait dengan teknologi digital, seperti perangkat lunak, situs web, aplikasi, dan perangkat elektronik lainnya yang memungkinkan pengguna berinteraksi langsung dengan komputer atau sistem. Namun, konsep interaktif juga bisa diterapkan pada komunikasi antarindividu dalam berbagai konteks, termasuk diskusi, presentasi, dan komunikasi dalam kehidupan sehari-hari.[61]