

IMPLEMENTASI WEBSITE E-COMMERCE BUKET BUNGA DENGAN FITUR CHATBOT AI SEBAGAI VIRTUAL ASSISTANT PELANGGAN



LAPORAN TUGAS MAHASISWA

Nama Anggota	:	Dwi Retno Wulandari (41236692)
		Elsa Febiyanti (41236698)
		Habibah Alya (41236714)
Minggu ke-	:	15
Dosen Pengampu	:	Khoerul Anam, M.Kom.
Mata Kuliah	:	Deep Learning Lanjut
Program Studi	:	Teknik Informatika

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
STMIK IKMI CIREBON
2025**

LEMBAR PENGESAHAN I

IMPLEMENTASI WEBSITE E-COMMERCE BUKET BUNGA DENGAN FITUR CHATBOT AI SEBAGAI VIRTUAL ASSISTANT PELANGGAN

Disusun Oleh :

- 1. Dwi Retno Wulandari (41236692)**
- 2. Elsa Febiyanti (41236698)**
- 3. Habibah Alya (41236714)**

Cirebon, 11 Februari 2026
Menyetujui,
Dosen Pengampu

Khoerul Anam, M.Kom.
NIDN.

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, proyek pengembangan website Urban Jungle – Toko Bunga Premium ini dapat diselesaikan dengan baik. Proyek ini dikembangkan sebagai bentuk implementasi pembelajaran dalam bidang pengembangan aplikasi web, khususnya dalam perancangan dan pembangunan sistem informasi berbasis web yang interaktif, responsif, dan berorientasi pada pengalaman pengguna (user experience).

Latar belakang pengembangan proyek ini didasarkan pada kebutuhan transformasi digital dalam sektor usaha mikro dan menengah, khususnya pada bidang perdagangan bunga dan bouquet premium. Perkembangan teknologi informasi telah mendorong pelaku usaha untuk menghadirkan layanan daring yang tidak hanya informatif, tetapi juga mampu mendukung transaksi, promosi, serta komunikasi pelanggan secara real time. Oleh karena itu, proyek ini dirancang sebagai sebuah platform e-commerce sederhana yang mengintegrasikan fitur katalog produk, sistem keranjang belanja, filter kategori, desain antarmuka responsif, serta fitur chatbot berbasis kecerdasan buatan sebagai layanan pelanggan otomatis.

Tujuan utama dari pengembangan proyek ini adalah untuk mengimplementasikan konsep-konsep pemrograman web modern yang meliputi HTML5, CSS3, JavaScript, serta integrasi Application Programming Interface (API) dalam membangun sistem interaktif. Selain itu, proyek ini bertujuan untuk meningkatkan pemahaman dalam perancangan antarmuka pengguna (UI) yang estetis, optimalisasi pengalaman pengguna (UX), serta penerapan prinsip desain responsif agar sistem dapat diakses melalui berbagai perangkat.

Melalui proses perancangan dan pengembangan ini, penulis memperoleh pengalaman praktis dalam menerapkan siklus pengembangan perangkat lunak secara sistematis, mulai dari tahap analisis kebutuhan, perancangan antarmuka, implementasi kode program, hingga tahap pengujian fungsionalitas sistem. Penulis juga memahami pentingnya konsistensi desain, efisiensi struktur kode, optimalisasi performa aplikasi, serta aspek keamanan dalam pengelolaan data pengguna. Proyek ini sekaligus menjadi sarana untuk melatih kemampuan analitis, problem solving, serta kreativitas dalam merancang solusi berbasis teknologi yang sesuai dengan kebutuhan bisnis digital.

Penulis menyadari bahwa proyek ini masih memiliki keterbatasan, baik dari segi skalabilitas sistem, optimalisasi keamanan tingkat lanjut, maupun integrasi sistem pembayaran daring secara langsung. Oleh karena itu, pengembangan lebih lanjut sangat diperlukan agar sistem ini dapat diimplementasikan secara lebih komprehensif dan profesional dalam skala industri. Kritik dan saran yang konstruktif dari berbagai pihak sangat diharapkan guna penyempurnaan dan pengembangan sistem di masa mendatang.

Akhir kata, penulis berharap agar proyek ini tidak hanya menjadi bagian dari pemenuhan tugas akademik semata, tetapi juga dapat memberikan kontribusi nyata dalam pengembangan kompetensi di bidang teknologi informasi, khususnya dalam perancangan dan pembangunan aplikasi web berbasis e-commerce. Semoga hasil pengembangan ini dapat menjadi fondasi untuk inovasi yang lebih luas serta memberikan manfaat bagi pengembangan sistem digital yang efektif, dan efisien.

DAFTAR ISI

LEMBAR PENGESAHAN I	2
KATA PENGANTAR	3
DAFTAR ISI.....	4
DAFTAR GAMBAR	7
DAFTAR TABEL	8
BAB I	9
PENDAHULUAN	9
1.1 Latar Belakang	9
1.2 Rumusan Masalah	10
1.3 Tujuan Pengembangan	11
1.4 Batasan Masalah	12
1.5 Manfaat Penelitian	12
BAB II.....	14
LANDASAN TEORI	14
2.1 Konsep Chatbot.....	14
2.1.1 Chatbot Berbasis Aturan (Rule-Based Chatbot).....	14
2.1.2 Retrieval-Based Chatbot.....	14
2.1.3 Generative AI Chatbot	15
2.2 Fundamental Natural Language Processing (NLP)	15
2.2.1 Tokenisasi dan Segmentasi Subword	15
2.2.2 Representasi Embedding Kontekstual.....	16
2.2.3 Klasifikasi Intent dan Slot Filling	16
2.2.4 Arsitektur Neural Transformer.....	17
2.3 Metodologi Large Language Model (LLM)	17
2.3.1 Objektif Pretraining Unsupervised	17
2.3.2 Supervised Fine-Tuning dan Alignment	18
2.3.3 In-Context Learning dan Prompt Engineering.....	18
2.4 Arsitektur dan Implementasi Sistem.....	19
2.4.1 Arsitektur Frontend dan User Interface.....	19
2.4.2 Arsitektur Backend dan Integrasi API.....	19
2.4.3 Integrasi API Model AI.....	20
2.4.4 Manajemen State Client-Side	20
2.4.5 Observability dan Error Handling.....	21
2.5 Aspek Etika dan Keamanan.....	21
2.5.1 Bias Algoritmik dan Keadilan.....	21
2.5.2 Privasi Data dan Kepatuhan Regulasi	22
2.5.3 Penyalahgunaan AI dan Ancaman Adversarial	22

2.5.4 Moderasi Konten dan Keamanan.....	23
BAB III	24
METODOLOGI PENGEMBANGAN	24
3.1 Metode Pengembangan.....	24
3.2 Analisis Kebutuhan	25
3.2.1 Kebutuhan Fungsional	25
3.2.2 Kebutuhan Non-Fungsional.....	26
3.3 Perancangan Sistem	27
3.3.1 Use Case Diagram	28
3.3.2 Activity Diagram.....	30
3.3.3 Entity Relationship Diagram (ERD)	32
3.4 Perancangan Prompt dan Model AI.....	33
3.5 Implementasi Sistem	34
BAB IV	41
HASIL DAN PEMBAHASAN	41
4.1 Tampilan Aplikasi.....	41
4.1.1 Halaman Utama dan Katalog Produk	41
4.1.2 Keranjang Belanja.....	42
4.1.3 Antarmuka Chatbot AI.....	43
4.1.4 Halaman Tentang dan Kontak.....	44
4.2 Pengujian Sistem	45
4.2.1 Pengujian Black Box.....	45
4.2.2 Pengujian Akurasi Respons Chatbot.....	47
4.2.3 Pengujian Usability Interface	49
4.3 Evaluasi Performa AI	49
4.3.1 Relevansi dan Akurasi Respons.....	49
4.3.2 Koherensi dan Kualitas Percakapan	50
4.3.3 Response Time dan Latency.....	50
4.4 Analisis Kelebihan dan Keterbatasan	51
4.4.1 Kelebihan Sistem.....	51
4.4.2 Keterbatasan Sistem	51
4.4.3 Rekomendasi Pengembangan	52
BAB V	54
KESIMPULAN DAN SARAN	54
5.1 Kesimpulan	54
5.2 Saran.....	55
DAFTAR PUSTAKA.....	57
LAMPIRAN.....	60
A. SOURCE CODE APLIKASI	60
A.1 Struktur HTML Utama	60

A.2 Antarmuka Chatbot	60
A.3 Fungsi Chatbot JavaScript.....	60
A.4 Fungsi Shopping Cart.....	61
B. DOKUMENTASI GROQ API.....	61
B.1 Endpoint API.....	61
B.2 Autentikasi.....	61
B.3 Format Request.....	61
B.4 Format Response.....	62
B.5 Parameter Konfigurasi.....	62
C. SCREENSHOT SISTEM.....	62
C.1 Dashboard Groq API Keys	62
C.2 Halaman Utama dan Hero Section	63
C.3 Grid Katalog Produk	63
C.4 Halaman Tentang Perusahaan	64
C.5 Modal Keranjang Belanja	64
C.6 Interface Chatbot AI.....	65
C.7 Halaman Kontak.....	66

DAFTAR GAMBAR

Gambar 3.1 Tahapan Metodologi SDLC untuk Pengembangan Web AI Chatbot.....	25
Gambar 3.2 Use Case Diagram Website Urban jungle E-Commerce.....	29
Gambar 3.3 Activity Diagram User Journey Website.....	31
Gambar 3.4 Swimlane Diagram Proses Bisnis dan Interaksi Sistem	39
Gambar 4.1 Tampilan Hero Section dan Katalog Produk	42
Gambar 4.2 Grid Katalog Produk dengan Badge dan Harga	42
Gambar 4.3 M Keranjang Belanja.....	43
Gambar 4.4 Jendela Chatbot	44
Gambar 4.5 Halaman Deskripsi Toko	45
Gambar 4.6 Halaman Kontak	45
Gambar 4.7 Dashboard Groq API Keys untuk Manajemen Akses	53
Gambar C.1 Dashboard Groq Console untuk Manajemen API Keys	63
Gambar C.2 Hero Section	63
Gambar C.3 Grid Katalog Produk	64
Gambar C.4 Halaman Tentang	64
Gambar C.5 Keranjang Belanja.....	65
Gambar C.6 Jendela Chatbot.....	65
Gambar C.7 Halaman Kontak	66

DAFTAR TABEL

Tabel 4.1 Hasil Pengujian Black Box Fungsionalitas Sistem	46
Tabel 4.2 Hasil Pengujian Akurasi Respon Chatbot AI.....	48
Tabel B.1 Parameter Konfigurasi Groq API	62

BAB I

PENDAHULUAN

1.1 Latar Belakang

Transformasi digital yang terjadi dalam dua dekade terakhir telah mengubah paradigma bisnis dari sistem konvensional menuju sistem berbasis teknologi informasi yang terintegrasi. Perkembangan internet, perangkat mobile, serta komputasi awan (cloud computing) telah mendorong pertumbuhan pesat sektor e-commerce di berbagai bidang industri, termasuk industri kreatif dan floristry. Penjualan bouquet dan rangkaian bunga yang sebelumnya dilakukan secara langsung di toko fisik kini bertransformasi ke platform digital yang memungkinkan pelanggan melakukan pencarian produk, pemesanan, serta pembayaran secara daring dengan lebih praktis dan efisien.

Meskipun website e-commerce memberikan kemudahan dalam penyajian katalog produk dan sistem transaksi, tantangan utama yang sering dihadapi adalah aspek pelayanan pelanggan (customer service). Pada website toko buket, pelanggan umumnya membutuhkan informasi yang bersifat personal dan kontekstual, seperti rekomendasi bouquet berdasarkan jenis acara (anniversary, ulang tahun, wisuda, atau perayaan khusus), detail komposisi bunga, kisaran harga, estimasi pengiriman, hingga ketersediaan stok. Ketika layanan ini hanya mengandalkan admin manusia melalui media komunikasi seperti WhatsApp atau telepon, maka muncul berbagai keterbatasan, antara lain keterbatasan jam operasional, potensi keterlambatan respons, inkonsistensi jawaban, serta peningkatan beban kerja operasional. Kondisi tersebut dapat berdampak pada menurunnya kualitas pengalaman pengguna (user experience) serta berpotensi mengurangi tingkat konversi penjualan (Gnewuch et al., 2017).

Dalam konteks persaingan bisnis digital yang semakin kompetitif, pelanggan cenderung mengharapkan respons yang cepat, akurat, dan tersedia setiap saat. Studi menunjukkan bahwa kecepatan dan kualitas respons layanan pelanggan secara signifikan memengaruhi tingkat kepuasan dan loyalitas pelanggan (Huang & Rust, 2021). Oleh karena itu, diperlukan solusi teknologi yang mampu menjawab kebutuhan interaksi real-time secara efisien dan berkelanjutan.

Salah satu inovasi yang berkembang pesat dalam bidang kecerdasan buatan adalah chatbot berbasis Artificial Intelligence (AI), khususnya yang memanfaatkan teknologi Natural Language Processing (NLP). Chatbot AI dirancang untuk mensimulasikan percakapan manusia dengan memahami bahasa alami pengguna dan memberikan respons yang relevan serta kontekstual (Adamopoulou & Moussiades, 2020). Berbeda dengan chatbot berbasis aturan (rule-based chatbot) yang terbatas pada pola pertanyaan tertentu, chatbot berbasis model bahasa besar (Large Language Model / LLM) memiliki kemampuan memahami konteks percakapan yang

lebih kompleks dan fleksibel.

Integrasi chatbot AI pada website toko buket menggunakan API Groq yang menyediakan akses ke model bahasa besar modern memungkinkan sistem untuk memproses pertanyaan pengguna secara lebih natural, adaptif, dan cerdas. Teknologi ini tidak hanya mampu menjawab pertanyaan umum mengenai produk dan layanan, tetapi juga dapat memberikan rekomendasi bouquet yang sesuai dengan kebutuhan pelanggan, menampilkan informasi keranjang belanja secara dinamis, serta merespons pertanyaan terkait alamat dan jam operasional secara konsisten. Dengan memanfaatkan arsitektur berbasis API, sistem dapat dikembangkan secara modular dan scalable sehingga mendukung pengembangan fitur lanjutan di masa mendatang.

Urgensi penerapan AI dalam sistem ini dapat ditinjau dari beberapa aspek strategis. Pertama, aspek efisiensi operasional, di mana chatbot mampu menangani pertanyaan berulang (frequently asked questions) secara otomatis tanpa keterlibatan langsung sumber daya manusia, sehingga mengurangi biaya operasional dan meningkatkan produktivitas (Davenport et al., 2020). Kedua, aspek otomatisasi layanan 24 jam, yang memungkinkan pelanggan memperoleh informasi kapan pun tanpa terikat jam kerja. Ketiga, aspek konsistensi dan akurasi informasi, di mana sistem berbasis AI dapat memberikan jawaban yang seragam sesuai dengan data produk yang telah terintegrasi. Keempat, aspek peningkatan pengalaman pengguna, karena interaksi yang responsif dan interaktif dapat meningkatkan kenyamanan serta kepercayaan pelanggan terhadap platform.

Berdasarkan uraian tersebut, pengembangan website toko buket yang terintegrasi dengan chatbot AI berbasis API Groq merupakan langkah inovatif yang relevan dengan kebutuhan transformasi digital saat ini. Implementasi sistem ini diharapkan mampu meningkatkan efisiensi layanan, mempercepat proses interaksi pelanggan, mengoptimalkan pengalaman pengguna, serta mendukung peningkatan daya saing bisnis di era ekonomi digital. Oleh karena itu, penelitian dan pengembangan sistem chatbot AI pada platform e-commerce toko bunga menjadi penting untuk dikaji secara ilmiah dalam rangka mengukur efektivitas, efisiensi, serta kontribusinya terhadap kualitas layanan digital.

1.2 Rumusan Masalah

Perkembangan teknologi kecerdasan buatan dalam bidang layanan digital memberikan peluang besar untuk meningkatkan kualitas interaksi antara sistem dan pengguna. Dalam konteks website toko buket yang terintegrasi dengan chatbot AI berbasis API Groq, diperlukan perumusan masalah yang jelas agar proses perancangan dan pengembangan sistem dapat dilakukan secara terarah dan sistematis. Rumusan masalah ini disusun berdasarkan kebutuhan teknis implementasi sistem, integrasi teknologi AI, serta aspek evaluasi kinerja chatbot dalam mendukung layanan pelanggan. Dengan adanya rumusan masalah yang terstruktur, penelitian

ini diharapkan mampu menghasilkan solusi rekayasa perangkat lunak yang efektif dan terukur. Berdasarkan latar belakang tersebut, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan chatbot berbasis Artificial Intelligence (AI) pada website toko buket agar mampu memberikan respons yang akurat, kontekstual, dan relevan terhadap pertanyaan pengguna?
2. Bagaimana mengintegrasikan API Groq sebagai penyedia Large Language Model (LLM) ke dalam aplikasi web berbasis HTML, CSS, dan JavaScript secara efektif dan aman?
3. Bagaimana merancang arsitektur komunikasi antara frontend dan layanan AI berbasis API agar percakapan dapat diproses secara real-time dengan performa optimal?
4. Bagaimana mengelola konteks percakapan serta data dinamis seperti keranjang belanja agar chatbot mampu memberikan informasi yang personal dan sesuai kondisi pengguna?
5. Bagaimana mengevaluasi efektivitas implementasi chatbot AI dalam meningkatkan kualitas layanan pelanggan dan pengalaman pengguna pada website toko buket?

1.3 Tujuan Pengembangan

Tujuan pengembangan merupakan arah dan sasaran yang ingin dicapai melalui proses perancangan dan pengembangan sistem. Dalam penelitian rekayasa perangkat lunak, tujuan harus dirumuskan secara spesifik, terukur, dan selaras dengan rumusan masalah yang telah ditetapkan. Pengembangan chatbot AI pada website toko buket ini tidak hanya berorientasi pada implementasi teknis, tetapi juga pada peningkatan kualitas layanan dan efisiensi operasional. Oleh karena itu, tujuan penelitian ini difokuskan pada aspek desain sistem, integrasi teknologi, serta evaluasi performa chatbot sebagai bagian dari solusi digital. Adapun tujuan penelitian ini adalah sebagai berikut:

1. Merancang dan membangun sistem chatbot berbasis AI pada website toko buket menggunakan Large Language Model (LLM) melalui API Groq.
2. Mengimplementasikan integrasi antara frontend website (HTML, CSS, JavaScript) dengan layanan AI berbasis API secara terstruktur dan aman.
3. Mengembangkan mekanisme pengelolaan konteks percakapan agar chatbot mampu mempertahankan relevansi dialog secara berkelanjutan.
4. Mengoptimalkan fitur chatbot untuk mendukung layanan pelanggan otomatis selama 24 jam (24/7).
5. Melakukan pengujian dan evaluasi terhadap kinerja chatbot dalam aspek akurasi respons, kecepatan layanan, dan kepuasan pengguna.

1.4 Batasan Masalah

Dalam penelitian dan pengembangan sistem, diperlukan pembatasan ruang lingkup agar penelitian tetap fokus dan tidak melebar ke aspek yang berada di luar tujuan utama. Batasan masalah ini ditetapkan untuk memperjelas cakupan teknis sistem yang dikembangkan serta ruang lingkup evaluasi yang dilakukan. Dengan adanya batasan yang jelas, proses analisis, implementasi, dan pengujian dapat dilakukan secara lebih terstruktur dan terukur. Batasan ini juga membantu menjaga konsistensi antara tujuan penelitian dan hasil yang diharapkan. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem yang dikembangkan berupa website toko buket berbasis web sesuai dengan struktur HTML, CSS, dan JavaScript yang telah dirancang sebelumnya.
2. Chatbot menggunakan model bahasa besar (LLM) yang diakses melalui API Groq.
3. Chatbot difokuskan untuk menjawab pertanyaan terkait produk bouquet, harga, ketersediaan, metode pembayaran, estimasi pengiriman, dan informasi umum toko.
4. Sistem tidak mencakup integrasi langsung dengan payment gateway atau sistem manajemen inventori berbasis database eksternal.
5. Pengujian sistem terbatas pada evaluasi performa chatbot dan pengalaman pengguna dalam skala terbatas atau simulasi.

1.5 Manfaat Penelitian

Penelitian dan pengembangan sistem chatbot berbasis Artificial Intelligence (AI) pada website toko buket ini diharapkan memberikan kontribusi yang signifikan baik dari aspek akademik, praktis, maupun teknologis. Manfaat penelitian tidak hanya terbatas pada implementasi sistem semata, tetapi juga pada pengembangan keilmuan, peningkatan kualitas layanan bisnis digital, serta kontribusi terhadap pemanfaatan teknologi AI dalam bidang e-commerce. Dengan demikian, hasil penelitian ini diharapkan dapat memberikan nilai tambah yang berkelanjutan bagi berbagai pihak yang berkepentingan.

Secara akademik, penelitian ini diharapkan dapat memperkaya kajian ilmiah dalam bidang rekayasa perangkat lunak, kecerdasan buatan, dan pengembangan aplikasi berbasis web. Implementasi chatbot berbasis Large Language Model (LLM) melalui API Groq memberikan contoh nyata penerapan konsep Artificial Intelligence dan Natural Language Processing (NLP) dalam sistem informasi berbasis web.

Secara praktis, penelitian ini memberikan solusi nyata bagi pelaku usaha toko buket dalam meningkatkan kualitas layanan pelanggan. Implementasi chatbot AI memungkinkan sistem memberikan respons secara otomatis, cepat, dan konsisten terhadap pertanyaan pelanggan tanpa bergantung sepenuhnya pada admin manusia. Hal ini berpotensi meningkatkan efisiensi

operasional, mengurangi beban kerja layanan pelanggan, serta mempercepat proses interaksi dan pengambilan keputusan pembelian oleh pelanggan.

Dari sisi teknologi, penelitian ini mendorong pemanfaatan teknologi Large Language Model (LLM) dalam konteks aplikasi web berbasis e-commerce. Integrasi API Groq sebagai penyedia model bahasa besar menunjukkan bagaimana teknologi AI modern dapat diimplementasikan secara modular dan scalable dalam sistem berbasis web menggunakan HTML, CSS, dan JavaScript.

Penelitian ini juga memberikan gambaran teknis mengenai arsitektur komunikasi antara frontend dan layanan AI berbasis API, pengelolaan keamanan API key, serta strategi pengelolaan konteks percakapan (conversation management). Dengan demikian, penelitian ini berkontribusi pada pengembangan sistem cerdas yang adaptif, efisien, dan berorientasi pada pengalaman pengguna (user-centered system design).

BAB II

LANDASAN TEORI

2.1 Konsep Chatbot

Chatbot merupakan sistem perangkat lunak yang dirancang untuk mensimulasikan percakapan manusia melalui antarmuka teks dengan memanfaatkan teknologi Artificial Intelligence (AI) dan Natural Language Processing (NLP). Berdasarkan analisis terhadap program website toko buket yang dikembangkan (berbasis HTML, CSS, dan JavaScript murni), chatbot diimplementasikan langsung pada sisi frontend menggunakan mekanisme pemanggilan API Groq melalui JavaScript. Dengan demikian, chatbot berfungsi sebagai komponen interaktif pada halaman web yang terintegrasi dengan elemen antarmuka seperti tombol chat, jendela percakapan, serta pengelolaan data keranjang belanja melalui localStorage.

Secara konseptual, chatbot dapat diklasifikasikan ke dalam beberapa pendekatan utama, yaitu rule-based, retrieval-based, dan generative AI chatbot. Perkembangan arsitektur transformer (Vaswani et al., 2017) memungkinkan munculnya chatbot generatif berbasis Large Language Model (LLM) yang lebih adaptif dibanding pendekatan sebelumnya. Dalam sistem website toko buket ini, pendekatan yang digunakan termasuk kategori generative AI chatbot karena respons dihasilkan secara dinamis oleh model bahasa melalui API Groq.

2.1.1 Chatbot Berbasis Aturan (Rule-Based Chatbot)

Chatbot berbasis aturan beroperasi pada pohon keputusan yang telah ditentukan sebelumnya dan algoritma pencocokan pola. Sistem ini menggunakan struktur logika kondisional untuk memetakan input pengguna ke respons yang telah didefinisikan melalui deteksi kata kunci dan pencocokan ekspresi reguler (Abu Shawar & Atwell, 2007). Arsitektur fundamental terdiri dari:

1. Mesin pengenalan pola yang memanfaatkan ekspresi reguler untuk parsing input
 2. Basis pengetahuan terstruktur sebagai pernyataan kondisional if-then
 3. Modul generasi respons yang memetakan pola yang dikenali ke template output
 4. Sistem manajemen konteks yang mempertahankan status percakapan di seluruh interaksi
- Meskipun sistem berbasis aturan menunjukkan keandalan tinggi dalam domain yang terbatas, mereka menunjukkan keterbatasan signifikan dalam menangani variabilitas linguistik dan memerlukan kurasi aturan manual yang ekstensif untuk ekspansi domain (Følstad & Brandtzæg, 2017).

2.1.2 Retrieval-Based Chatbot

Retrieval-based chatbot memilih jawaban terbaik dari sekumpulan respons yang telah tersedia berdasarkan kemiripan teks atau klasifikasi intent. Sistem ini lebih fleksibel dibanding

rule-based karena mampu menangani variasi pertanyaan dengan teknik pencocokan semantik. Meskipun demikian, pendekatan retrieval tetap terbatas pada respons yang telah didefinisikan sebelumnya dan tidak mampu menghasilkan jawaban baru di luar basis data yang tersedia.

2.1.3 Generative AI Chatbot

Chatbot generatif merepresentasikan paradigma state-of-the-art dalam AI percakapan, memanfaatkan large language models (LLM) untuk mensintesis respons yang sesuai konteks secara de novo. Implementasi toko bunga Urban Jungle mencontohkan arsitektur ini melalui integrasi dengan model LLaMA 3.3-70B dari Groq, menunjukkan beberapa karakteristik kunci:

1. Arsitektur neural berbasis transformer yang memproses input sekuensial melalui mekanisme multi-head self-attention
2. Generasi teks autoregresif yang memanfaatkan sampling terkontrol temperature ($\text{temperature} = 0.7$ dalam implementasi) untuk keseimbangan kreativitas dan koherensi
3. Manajemen context window yang mempertahankan riwayat percakapan dengan batasan token limit ($\text{max_tokens} = 300$, dengan sliding window 10 pesan)
4. Prompt engineering sistem yang menyematkan pengetahuan domain, katalog produk, dan pedoman respons langsung ke dalam konteks model

Implementasi mendemonstrasikan prompt engineering yang sophisticated dengan menggabungkan informasi produk terstruktur (16 rangkaian bunga dengan spesifikasi), detail operasional (jam kerja, lokasi), dan batasan perilaku (panjang respons, penggunaan emoji, pola call-to-action) dalam system prompt. Pendekatan ini memanfaatkan kemampuan in-context learning yang melekat pada LLM, menghilangkan kebutuhan untuk fine-tuning model sambil mempertahankan spesifisitas domain.

2.2 Fundamental Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan fondasi teoritis dan komputasional yang memungkinkan mesin untuk menginterpretasi dan menghasilkan bahasa manusia. Implementasi chatbot, meskipun memanfaatkan kemampuan LLM yang telah dilatih sebelumnya, secara inheren bergantung pada mekanisme NLP yang sophisticated yang beroperasi dalam arsitektur model.

2.2.1 Tokenisasi dan Segmentasi Subword

Tokenisasi merepresentasikan langkah preprocessing fundamental yang mentransformasi teks mentah menjadi unit komputasional diskrit. LLM modern, termasuk model LLaMA, menggunakan algoritma Byte-Pair Encoding (BPE) atau SentencePiece untuk tokenisasi subword, menyeimbangkan batasan ukuran vocabulary terhadap granularitas semantik.

Arsitektur LLaMA 3.3 memproses input bahasa Indonesia (sebagaimana dibuktikan oleh respons bahasa Indonesia chatbot) melalui tokenizer multilingual yang dilatih pada korpus bahasa yang beragam.

Strategi tokenisasi secara langsung memengaruhi kinerja model melalui beberapa mekanisme: cakupan vocabulary yang memengaruhi penanganan kata out-of-vocabulary, panjang urutan token yang memengaruhi kompleksitas komputasi attention, dan batas subword yang memengaruhi granularitas representasi semantik. Nama produk chatbot (misalnya, "Sunshine Delight", "Pastel Garden Symphony") mengalami tokenisasi yang mempertahankan koherensi semantik di seluruh konteks multilingual.

2.2.2 Representasi Embedding Kontekstual

Embedding merupakan representasi vektor padat yang memetakan token diskrit ke ruang berdimensi tinggi kontinu di mana hubungan semantik terwujud sebagai kedekatan geometris. Model berbasis transformer menggunakan embedding yang dipelajari dikombinasikan dengan positional encoding untuk menangkap konten semantik dan positioning sekuensial (Devlin et al., 2019).

Arsitektur LLaMA menghasilkan embedding kontekstual melalui lapisan embedding awalnya, mentransformasi token input menjadi vektor yang selanjutnya mengalami transformasi melalui mekanisme attention. Embedding ini mengkodekan:

1. Semantik leksikal yang menangkap makna kata dan karakteristik distribusional
2. Semantik komposisional yang merepresentasikan makna tingkat frasa dan kalimat
3. Dependensi kontekstual yang mencerminkan konteks linguistik di sekitarnya
4. Representasi spesifik tugas yang diadaptasi melalui fine-tuning atau in-context learning

Embedding system prompt dalam implementasi chatbot mengkodekan pengetahuan domain, menetapkan ruang semantik di mana query produk, informasi harga, dan detail operasional mencapai skor kesamaan tinggi dengan komponen pengetahuan yang relevan.

2.2.3 Klasifikasi Intent dan Slot Filling

Klasifikasi intent merupakan tugas NLP fundamental yang memetakan ucapan pengguna ke kategori tindakan yang telah ditentukan. Pendekatan tradisional menggunakan model klasifikasi supervised yang dilatih pada data percakapan berlabel(Liu & Lane, 2016). Namun, implementasi chatbot generatif mendemonstrasikan paradigma alternatif di mana pemahaman intent muncul secara implisit dari pemahaman LLM daripada modul klasifikasi eksplisit.

Kemampuan chatbot untuk mengenali query terkait keranjang mendemonstrasikan klasifikasi intent implisit melalui deteksi kata kunci ("keranjang", "cart", "belanja", "pesanan", "total", "harga") yang memicu augmentasi konteks dengan status keranjang belanja. Pendekatan hybrid ini menggabungkan:

1. Pemicu intent berbasis kata kunci ringan untuk efisiensi komputasional
2. Pemahaman semantik berbasis LLM untuk generasi respons
3. Injeksi konteks dinamis yang menggabungkan state eksternal (data cart dari localStorage)

Arsitektur ini merepresentasikan optimasi pragmatis yang menyeimbangkan biaya inferensi terhadap fungsionalitas, menghindari invokasi LLM yang tidak perlu untuk query state sederhana sambil mempertahankan fleksibilitas percakapan.

2.2.4 Arsitektur Neural Transformer

Arsitektur transformer, yang diperkenalkan oleh (Vaswani et al., 2017), secara fundamental merevolusi NLP melalui mekanisme self-attention yang memungkinkan pemrosesan paralel data sekuensial. Model LLaMA mengimplementasikan arsitektur transformer decoder-only yang dioptimalkan untuk generasi teks autoregresif.

Komponen arsitektur kunci meliputi:

1. Multi-head self-attention: Menghitung representasi tertimbang berdasarkan hubungan token di seluruh urutan, memungkinkan penangkapan dependensi jarak jauh
2. Position-wise feed-forward networks: Menerapkan transformasi non-linear secara independen ke setiap posisi
3. Layer normalization dan residual connections: Memfasilitasi stabilitas pelatihan dan aliran gradien dalam jaringan dalam
4. Rotary positional embeddings (RoPE): Mengkodekan informasi posisional melalui matriks rotasi dalam arsitektur LLaMA

Generasi respons chatbot memanfaatkan mekanisme ini untuk mempertahankan koherensi percakapan di seluruh interaksi multi-turn, dengan riwayat percakapan 10 pesan menyediakan konteks temporal untuk komputasi attention.

2.3 Metodologi Large Language Model (LLM)

Large Language Models merepresentasikan pergeseran paradigma dalam NLP, mencapai kinerja yang belum pernah terjadi sebelumnya melalui skala dalam parameter model, data pelatihan, dan sumber daya komputasional. Model LLaMA 3.3-70B yang digunakan dalam implementasi chatbot mencontohkan arsitektur LLM modern dengan 70 miliar parameter yang dapat dilatih.

2.3.1 Objektif Pretraining Unsupervised

Pretraining merupakan fase fundamental di mana model mempelajari representasi linguistik umum dari korpus besar tanpa label. Model LLaMA menggunakan causal language modeling, memprediksi token berikutnya dengan konteks sebelumnya (Touvron et al., 2023). Fungsi objektif ini, secara matematis diformalkan sebagai memaksimalkan log-likelihood dari

urutan yang diamati, memungkinkan model untuk menginternalisasi pola statistik yang mencakup sintaksis, semantik, pragmatik, dan pengetahuan dunia di berbagai domain. Korpus pretraining LLaMA 3.3 mencakup teks web multilingual, repositori kode, literatur ilmiah, dan data percakapan, menetapkan kompetensi linguistik luas yang dapat diterapkan pada domain khusus tanpa pelatihan eksplisit.

2.3.2 Supervised Fine-Tuning dan Alignment

Fine-tuning mengadaptasi model yang telah dilatih sebelumnya untuk tugas atau domain spesifik melalui pembelajaran supervised pada dataset yang dikurasi. Meskipun implementasi chatbot tidak menggunakan fine-tuning eksplisit, LLaMA 3.3 menggabungkan instruction fine-tuning, melatih pada pasangan instruksi-respons yang dianotasi manusia untuk meningkatkan kemampuan mengikuti instruksi) (Ouyang et al., 2022). Pengembangan LLM kontemporer mengintegrasikan Reinforcement Learning from Human Feedback (RLHF), menyelaraskan output model dengan preferensi manusia mengenai helpfulness, harmlessness, dan honesty. Proses alignment ini terdiri dari:

1. Supervised fine-tuning pada data demonstrasi berkualitas tinggi
2. Pelatihan reward model pada penilaian preferensi manusia
3. Optimasi policy melalui Proximal Policy Optimization (PPO) yang memaksimalkan sinyal reward

Kepatuhan chatbot terhadap batasan panjang respons ("Maksimal 4-5 baris"), pedoman penggunaan emoji, dan nada profesional mencerminkan pelatihan alignment yang memungkinkan kepatuhan zero-shot dengan arahan system prompt.

2.3.3 In-Context Learning dan Prompt Engineering

Prompt engineering mengeksplorasi kemampuan in-context learning LLM, di mana model mengadaptasi perilaku berdasarkan konteks input semata tanpa pembaruan parameter. Fenomena ini, disebut few-shot learning, memungkinkan adaptasi tugas cepat melalui prompt yang dibangun dengan hati-hati (Brown et al., 2020).

System prompt chatbot mendemonstrasikan prompt engineering tingkat lanjut yang menggabungkan:

1. Spesifikasi peran: "Kamu adalah Customer Service AI untuk Urban Jungle" yang menetapkan identitas fungsional
2. Grounding pengetahuan: Katalog produk terstruktur dengan 16 item termasuk spesifikasi, harga, dan kategorisasi
3. Konteks operasional: Jam kerja, lokasi, informasi kontak, dan penawaran layanan
4. Batasan perilaku: Aturan format respons, pedoman nada, dan pola call-to-action
5. Contoh few-shot: Respons demonstrasi yang menggambarkan karakteristik output yang

diinginkan

Arsitektur prompt komprehensif ini mencapai spesialisasi domain yang sebanding dengan model fine-tuned sambil mempertahankan fleksibilitas deployment dan kemampuan iterasi cepat (Reynolds & McDonell, 2021). Injeksi konteks dinamis untuk query cart lebih lanjut mendemonstrasikan kecanggihan prompt engineering, menambahkan pesan pengguna dengan representasi data terstruktur yang memungkinkan integrasi koheren dari state eksternal.

2.4 Arsitektur dan Implementasi Sistem

Deployment chatbot berbasis web modern memerlukan desain arsitektur yang robust yang mengintegrasikan beberapa subsistem untuk presentasi, logika bisnis, persistensi data, dan inferensi AI. Implementasi Urban Jungle mendemonstrasikan arsitektur client-side dengan integrasi API eksternal.

2.4.1 Arsitektur Frontend dan User Interface

Frontend merupakan lapisan presentasi yang menghadap pengguna yang mengimplementasikan interface percakapan dan fungsionalitas e-commerce. Implementasi menggunakan vanilla JavaScript dengan manipulasi DOM untuk rendering dinamis, menghindari framework heavyweight untuk kesederhanaan deployment.

Komponen arsitektur kunci meliputi:

1. UI percakapan: Mengimplementasikan message bubbles, typing indicators, dan rendering timestamp dengan pembedaan visual antara pesan pengguna dan asisten
2. Responsive design: Media queries CSS yang memungkinkan layout yang dioptimalkan untuk mobile yang beradaptasi dengan batasan viewport
3. Sistem animasi: Animasi keyframe CSS untuk elemen floating, efek fade-in, dan typing indicators yang meningkatkan responsivitas yang dipersepsi
4. Menu quick action: Shortcut konteks-sensitif untuk query umum yang mengurangi beban input pengguna

Interface mendemonstrasikan pola UX modern termasuk optimistic UI updates (tampilan pesan langsung sebelum respons API), visualisasi loading state (typing indicator), dan mekanisme pemulihian error (fallback WhatsApp pada kegagalan API).

2.4.2 Arsitektur Backend dan Integrasi API

Implementasi menggunakan arsitektur serverless, mendelegasikan fungsionalitas backend ke layanan eksternal daripada mempertahankan infrastruktur server dedicated. Pilihan arsitektur ini menawarkan beberapa keuntungan:

1. Kompleksitas operasional yang berkurang: Menghilangkan persyaratan provisioning, scaling, dan maintenance server

2. Optimasi biaya: Model harga pay-per-use yang menyelaraskan biaya dengan penggunaan aktual
3. Automatic scaling: Infrastruktur yang dikelola provider yang beradaptasi dengan variasi traffic

Namun, arsitektur ini memperkenalkan risiko eksposur API key, karena kode client-side harus menyertakan kredensial autentikasi. Deployment production biasanya menggunakan lapisan proxy backend atau edge functions untuk mengamankan API key sambil mempertahankan manfaat serverless.

2.4.3 Integrasi API Model AI

Sistem berintegrasi dengan API inferensi Groq yang mengimplementasikan protokol request/response kompatibel dengan OpenAI. Integrasi mendemonstrasikan beberapa pola implementasi:

1. Manajemen riwayat percakapan: Mempertahankan konteks melalui array pesan dengan system prompts, input pengguna, dan respons asisten
2. Optimasi token limit: Mengimplementasikan sliding window (10 pesan) yang mencegah overflow konteks sambil mempertahankan riwayat percakapan terbaru
3. Parameter generasi: Temperature (0.7) yang menyeimbangkan kreativitas dan konsistensi, top_p sampling (0.9) untuk nucleus sampling, max_tokens (300) yang membatasi panjang respons
4. Injeksi konteks dinamis: Menambahkan query pengguna dengan state cart ketika kata kunci yang relevan terdeteksi

Integrasi API menggunakan pola request asinkron (async/await) yang mencegah blocking UI selama periode latensi jaringan, mempertahankan responsivitas yang dipersepsikan yang kritis untuk interface percakapan.

2.4.4 Manajemen State Client-Side

Implementasi menggunakan localStorage browser untuk manajemen state persisten, menyimpan data keranjang belanja di seluruh sesi. Pendekatan ini menawarkan kesederhanaan untuk prototype tetapi menunjukkan beberapa keterbatasan:

1. Batasan kapasitas storage: Biasanya terbatas pada 5-10MB per origin
2. Kurangnya sinkronisasi cross-device: State cart terisolasi ke browser individual
3. Ketiadaan validasi server-side: Memungkinkan potensi manipulasi harga melalui modifikasi client-side
4. Tidak ada infrastruktur analytics: Mencegah analisis perilaku dan optimasi funnel

Aplikasi e-commerce production biasanya menggunakan keranjang belanja yang didukung database dengan manajemen state server-side, persistensi sesi, dan workflow pemrosesan

pesanannya. Arsitektur saat ini cukup untuk tujuan demonstrasi tetapi memerlukan augmentasi backend untuk deployment komersial.

2.4.5 Observability dan Error Handling

Implementasi menggabungkan penanganan error dasar melalui blok try-catch dan console logging, menyediakan kemampuan debugging rudimenter. Namun, ia kekurangan infrastruktur observability komprehensif yang khas dari sistem production:

1. Structured logging: Pencatatan peristiwa sistematis dengan level severity, timestamps, dan metadata kontekstual
2. Performance monitoring: Tracking latensi request, analisis waktu respons API, dan metrik rendering frontend
3. Error tracking: Penangkapan exception otomatis dengan stack traces dan konteks reproduksi
4. User analytics: Analisis alur percakapan, distribusi intent, dan metrik kepuasan

Mengimplementasikan observability komprehensif memungkinkan optimasi berbasis data, respons insiden cepat, dan peningkatan kualitas berkelanjutan kemampuan kritis untuk aplikasi AI yang menghadap pelanggan (Kleppmann, 2017).

2.5 Aspek Etika dan Keamanan

Deployment sistem percakapan berbasis AI menimbulkan pertimbangan etika dan keamanan signifikan yang memerlukan mitigasi risiko sistematis. Pembahasan berikut memeriksa area concern kunci yang relevan dengan implementasi chatbot.

2.5.1 Bias Algoritmik dan Keadilan

Large language models mewarisi bias yang ada dalam data pelatihan, berpotensi terwujud sebagai output diskriminatif di berbagai dimensi demografis termasuk gender, ras, status sosial-ekonomi, dan latar belakang budaya (Bender et al., 2021). Bias ini muncul melalui beberapa mekanisme:

1. Bias representasi: Underrepresentasi kelompok minoritas dalam korpus pelatihan yang mengarah pada kompetensi model yang berkurang untuk konteks yang terkait dengan minoritas
2. Asosiasi stereotipikal: Korelasi statistik dalam data pelatihan yang memperkuat stereotip yang berbahaya
3. Bias pengukuran: Metrik evaluasi gagal menangkap keadilan di seluruh subkelompok demografis

Meskipun domain toko bunga menunjukkan risiko relatif rendah untuk bahaya yang diinduksi bias, audit bias sistematis tetap disarankan. Praktik yang direkomendasikan termasuk pengujian pengguna yang beragam di seluruh kelompok demografis, pengujian adversarial untuk

output diskriminatif, dan pemantauan berkelanjutan log percakapan untuk indikator bias (Mitchell et al., 2019).

2.5.2 Privasi Data dan Kepatuhan Regulasi

Sistem AI percakapan memproses informasi pengguna yang berpotensi sensitif yang memerlukan perlindungan privasi yang robust. Implementasi saat ini mentransmisikan konten percakapan ke infrastruktur inferensi Groq, menimbulkan beberapa pertimbangan privasi:

1. Kebijakan retensi data: Menentukan durasi penyimpanan untuk log percakapan dan interaksi pengguna
2. Berbagi data pihak ketiga: Akses provider API ke konten percakapan untuk peningkatan model
3. Personally identifiable information (PII): Penanganan nama, informasi kontak, dan riwayat pembelian
4. Kepatuhan regulasi: Kepatuhan terhadap GDPR, CCPA, UU PDP Indonesia, dan regulasi spesifik industri

Best practices meliputi implementasi prinsip minimalisasi data (mengumpulkan hanya informasi esensial), menyediakan kebijakan privasi yang transparan, memungkinkan permintaan penghapusan data pengguna, dan melakukan penilaian dampak privasi (Solove, 2013). Penyimpanan cart berbasis localStorage saat ini kekurangan enkripsi, merepresentasikan kerentanan privasi potensial yang memerlukan perlindungan kriptografis dalam deployment production.

2.5.3 Penyalahgunaan AI dan Ancaman Adversarial

Sistem AI menghadapi potensi penyalahgunaan melalui berbagai vektor serangan termasuk prompt injection, jailbreaking, dan social engineering. Implementasi chatbot menunjukkan beberapa permukaan kerentanan:

1. Serangan prompt injection: Input adversarial yang berusaha menggesampingkan instruksi sistem dan memunculkan perilaku yang tidak diinginkan
2. Information disclosure: Query yang dirancang untuk mengekstrak data pelatihan atau konten system prompt
3. Resource exhaustion: Pengguna jahat yang menghasilkan request API berlebihan untuk menimbulkan biaya atau menurunkan ketersediaan layanan
4. Pencurian API key: Kredensial yang terekspos dalam kode client-side yang memungkinkan penggunaan tidak sah

Strategi mitigasi termasuk sanitasi input, rate limiting, rotasi API key, implementasi proxy backend, dan pengujian adversarial. Model LLaMA 3.3 menggabungkan safety alignment yang mengurangi kerentanan terhadap jailbreaking, meskipun tidak ada sistem yang mencapai

ketahanan sempurna (Perez et al., 2022).

2.5.4 Moderasi Konten dan Keamanan

Sistem moderasi konten mencegah output berbahaya termasuk hate speech, konten eksplisit, promosi aktivitas ilegal, dan informasi menyesatkan. LLM modern menggabungkan safety alignment melalui RLHF, meskipun cakupan yang tidak sempurna memerlukan perlindungan tambahan (Gehman et al., 2020).

Infrastruktur moderasi yang direkomendasikan meliputi:

1. Input filtering: Mendeteksi dan memblokir konten yang dilarang dalam pesan pengguna sebelum inferensi model
2. Output validation: Memindai respons model untuk pelanggaran kebijakan sebelum presentasi kepada pengguna
3. Human review workflows: Mengeksklusi kasus edge ke moderator manusia untuk adjudikasi
4. Mekanisme pelaporan pengguna: Memungkinkan feedback pada respons bermasalah untuk peningkatan berkelanjutan

Chatbot toko bunga beroperasi dalam domain yang relatif jinak yang mengurangi persyaratan moderasi, meskipun kebijakan konten sistematis dan pemantauan tetap merupakan praktik prudent untuk sistem AI yang menghadap pelanggan mana pun.

BAB III

METODOLOGI PENGEMBANGAN

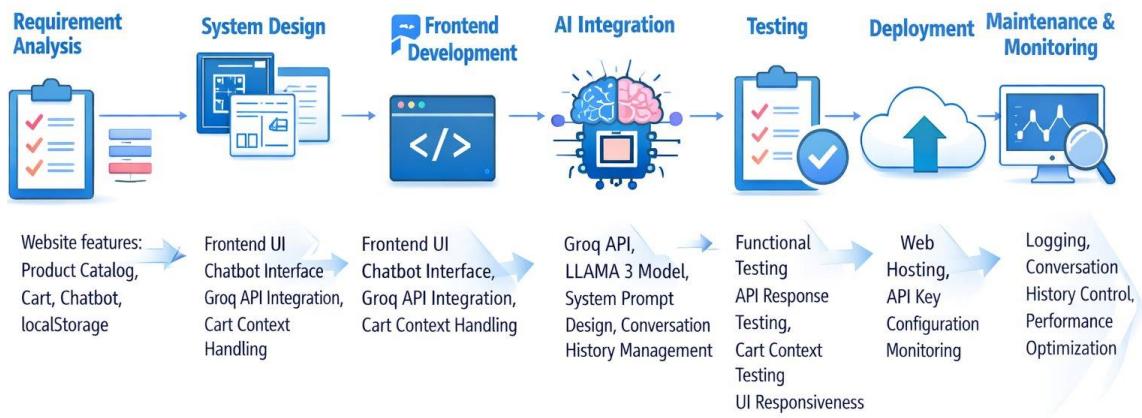
3.1 Metode Pengembangan

Pengembangan sistem web e-commerce Urban Jungle dengan integrasi AI chatbot menggunakan pendekatan metodologi Software Development Life Cycle (SDLC). SDLC merupakan kerangka kerja sistematis yang menyediakan pendekatan terstruktur untuk mengembangkan sistem informasi melalui serangkaian fase yang terdefinisi dengan baik (Pressman & Maxim, 2020). Pemilihan metodologi SDLC didasarkan pada karakteristik proyek yang memerlukan perencanaan matang, dokumentasi komprehensif, dan integrasi teknologi kecerdasan buatan yang kompleks.

Dalam konteks pengembangan sistem berbasis AI, SDLC memberikan fleksibilitas untuk mengakomodasi kebutuhan khusus seperti konfigurasi model machine learning, prompt engineering, dan optimasi performa AI (Sommerville, 2016). Metodologi ini memungkinkan tim pengembang untuk melakukan iterasi dan penyempurnaan pada setiap tahap tanpa mengorbankan stabilitas sistem secara keseluruhan. Selain itu, SDLC memfasilitasi dokumentasi yang detail pada setiap fase pengembangan, yang sangat penting untuk maintenance dan pengembangan sistem di masa mendatang.

Implementasi SDLC pada penelitian ini diadaptasi menjadi tujuh tahapan utama yang disesuaikan dengan kebutuhan pengembangan web berbasis AI, sebagaimana ditunjukkan pada Gambar 3.1. Tahapan-tahapan tersebut mencakup: (1) Requirement Analysis untuk mengidentifikasi kebutuhan fungsional dan non-fungsional; (2) System Design untuk merancang arsitektur sistem dan antarmuka pengguna; (3) Frontend Development untuk mengimplementasikan user interface dan user experience; (4) AI Integration untuk mengintegrasikan Groq API dan model LLAMA 3; (5) Testing untuk memvalidasi fungsionalitas dan performa sistem; (6) Deployment untuk publikasi sistem ke environment produksi; dan (7) Maintenance & Monitoring untuk pemeliharaan berkelanjutan dan monitoring performa.

Tahap Requirement Analysis dimulai dengan identifikasi fitur-fitur utama website e-commerce seperti product catalog, shopping cart, dan chatbot customer service. Analisis ini menghasilkan spesifikasi kebutuhan fungsional yang mencakup kemampuan sistem dalam menampilkan katalog produk, mengelola keranjang belanja, dan menyediakan layanan chatbot berbasis AI untuk customer support. Kebutuhan non-fungsional juga didefinisikan dengan jelas, termasuk target response time AI di bawah 3 detik, keamanan data pengguna, dan kemampuan sistem untuk menangani concurrent users.



Gambar 3.1 Tahapan Metodologi SDLC untuk Pengembangan Web AI Chatbot

3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan fase kritis dalam siklus pengembangan software yang berfungsi sebagai fondasi untuk seluruh proses pembangunan sistem (Wiegers & Beatty, 2013). Pada tahap ini dilakukan identifikasi komprehensif terhadap semua requirements baik fungsional maupun non-fungsional yang harus dipenuhi oleh sistem. Analisis kebutuhan sistem web e-commerce Urban Jungle dengan AI chatbot melibatkan stakeholder analysis, user story mapping, dan requirement prioritization untuk memastikan sistem yang dibangun dapat memenuhi ekspektasi pengguna dan tujuan bisnis.

Proses analisis kebutuhan dimulai dengan melakukan studi terhadap sistem e-commerce existing dan mengidentifikasi pain points yang dialami pengguna dalam proses belanja online. Berdasarkan hasil analisis, ditemukan bahwa pengguna membutuhkan sistem yang tidak hanya menyediakan katalog produk yang lengkap, tetapi juga customer service yang responsif 24/7 untuk menjawab pertanyaan seputar produk, ketersediaan stok, dan rekomendasi bunga yang sesuai dengan occasion tertentu. Kebutuhan ini menjadi dasar pengintegrasian teknologi AI chatbot ke dalam sistem e-commerce.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan fungsi-fungsi spesifik yang harus dapat dilakukan oleh sistem untuk memenuhi kebutuhan pengguna (Leffingwell & Widrig, 2003). Dalam konteks pengembangan web e-commerce Urban Jungle, kebutuhan fungsional mencakup seluruh fitur dan kapabilitas yang harus dimiliki sistem untuk mendukung proses bisnis online flower shop.

Identifikasi kebutuhan fungsional dilakukan melalui teknik use case analysis dan user story mapping yang melibatkan calon pengguna sistem.

Sistem harus mampu menampilkan katalog produk bunga secara komprehensif dengan informasi detail meliputi nama produk, deskripsi, harga, dan gambar produk berkualitas tinggi. Fitur filtering dan searching memungkinkan pengguna untuk mencari produk berdasarkan kategori seperti romantic, birthday, congratulation, dan luxury. Setiap produk dapat ditambahkan ke shopping cart dengan sistem quantity management yang memungkinkan pengguna mengatur jumlah item yang diinginkan. Cart context management diterapkan menggunakan localStorage browser untuk menjaga persistensi data keranjang belanja bahkan setelah browser ditutup.

Integrasi AI chatbot merupakan kebutuhan fungsional utama yang membedakan sistem ini dengan e-commerce konvensional. Chatbot harus mampu menerima input teks dari pengguna melalui interface chat yang intuitif, memproses pertanyaan menggunakan natural language processing, dan memberikan respons yang kontekstual dan relevan dalam waktu nyata. Sistem chatbot terintegrasi dengan Groq API yang menyediakan akses ke model LLAMA 3 (llama-3.1-70b-versatile), model language yang powerful dengan 70 billion parameters dan context window 128k tokens. Fitur conversation history management memastikan chatbot dapat mereferensikan percakapan sebelumnya untuk memberikan respons yang koheren dan personal.

Fitur customer service automation melalui chatbot mencakup kemampuan untuk menjawab frequently asked questions, memberikan rekomendasi produk berdasarkan preferensi dan occasion, menjelaskan detail produk, dan membantu proses checkout. Chatbot juga harus dapat menangani multiple conversation contexts secara simultan, termasuk tracking items yang ada di cart pengguna dan memberikan saran produk complementary. Sistem notifikasi real-time memberikan feedback kepada pengguna ketika produk berhasil ditambahkan ke cart atau ketika ada update dari chatbot.

3.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mengacu pada kriteria kualitas sistem yang tidak secara langsung terkait dengan fungsi spesifik, melainkan atribut sistem seperti performance, security, usability, dan scalability (Chung et al., 2012). Aspek non-fungsional sangat krusial dalam menentukan keberhasilan sistem karena berpengaruh langsung terhadap user experience dan kepuasan pengguna. Untuk sistem e-commerce dengan AI chatbot, kebutuhan non-fungsional harus didefinisikan dengan metrik yang terukur.

Performance merupakan aspek non-fungsional yang paling kritis dalam sistem berbasis AI. Response time chatbot harus dijaga di bawah 3 detik untuk menjaga user engagement dan satisfaction. Hal ini dicapai melalui optimasi pada sisi client dengan implementasi asynchronous

JavaScript dan pada sisi server dengan pemilihan model AI yang tepat serta konfigurasi parameter yang optimal. Groq API dipilih karena menyediakan inference speed yang superior dibanding provider AI lainnya, dengan latency rata-rata di bawah 1 detik untuk model LLAMA 3. Selain itu, implementasi lazy loading untuk gambar produk dan code splitting untuk JavaScript memastikan page load time tetap optimal meskipun sistem memuat banyak produk.

Keamanan sistem mencakup proteksi API key, sanitasi input pengguna untuk mencegah injection attacks, dan implementasi HTTPS untuk enkripsi data transmission. API key Groq disimpan di environment variables dan tidak di-expose ke client-side code untuk menghindari unauthorized usage. Validasi dan sanitasi input dilakukan baik di frontend maupun backend untuk mencegah XSS (Cross-Site Scripting) dan SQL injection attacks. Data pengguna yang tersimpan di localStorage dienkripsi untuk melindungi privacy, meskipun tidak menyimpan informasi sensitif seperti payment details.

Aspek usability dan user experience dirancang dengan mengikuti prinsip-prinsip human-computer interaction (HCI) yang established. Interface chatbot mengadopsi conversational UI design yang familiar bagi pengguna, dengan typing indicator dan message bubbles yang clear. Responsive web design memastikan sistem dapat diakses dengan optimal di berbagai devices mulai dari desktop, tablet, hingga smartphone. Accessibility features seperti semantic HTML, proper heading hierarchy, dan keyboard navigation support diimplementasikan untuk memastikan sistem dapat digunakan oleh users dengan berbagai kondisi. Scalability sistem dirancang untuk dapat menangani peningkatan traffic dan data volume seiring pertumbuhan bisnis, dengan arsitektur yang modular dan database design yang normalized.

3.3 Perancangan Sistem

Perancangan sistem merupakan proses mentransformasikan requirement specifications menjadi representasi sistem yang dapat diimplementasikan (Rumbaugh et al., 2004). Fase ini mencakup perancangan arsitektur sistem, desain database, perancangan interface, dan spesifikasi komponen-komponen sistem. Perancangan sistem e-commerce Urban Jungle menggunakan pendekatan component-based design yang memisahkan concerns ke dalam modul-modul yang independen namun cohesive.

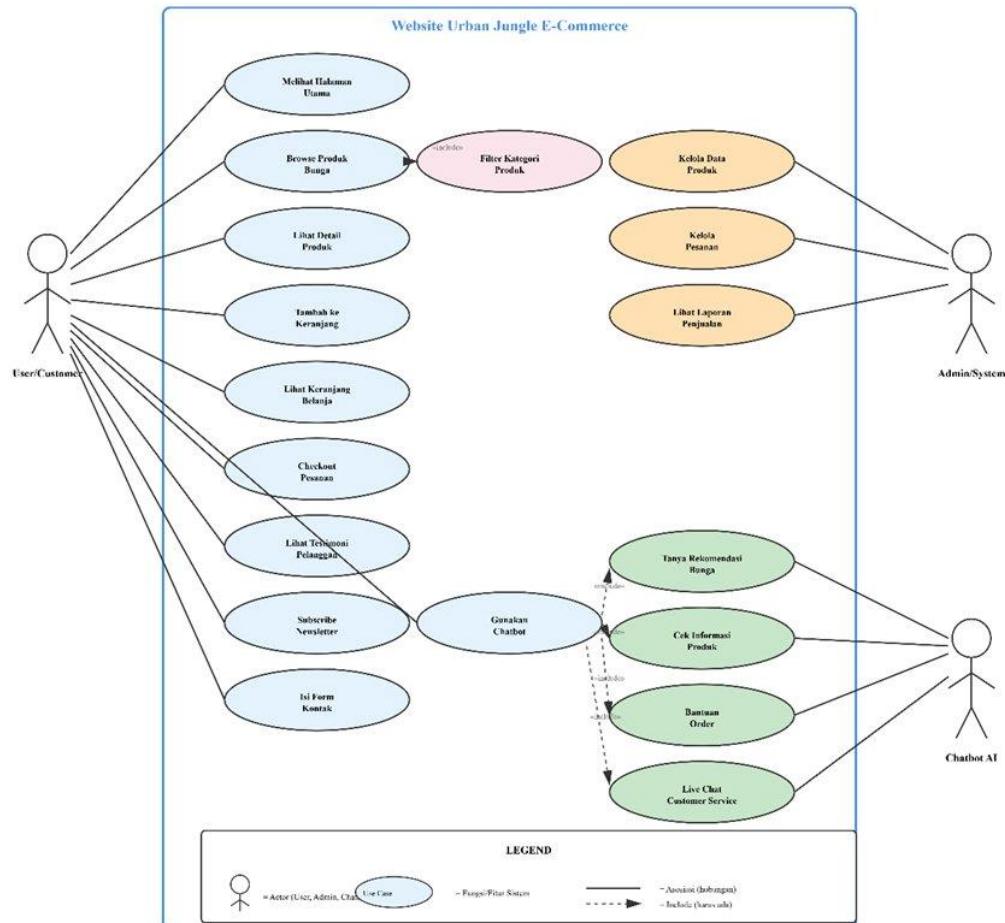
Arsitektur sistem dirancang menggunakan client-server architecture dengan clear separation between presentation layer, business logic layer, dan data layer. Frontend dibangun sebagai single-page application (SPA) yang resposif dan interaktif, berkomunikasi dengan backend services melalui RESTful API. Integrasi AI chatbot diposisikan sebagai independent service yang dapat di-scale secara terpisah dari main application. Pendekatan ini memberikan

flexibility dalam deployment dan memudahkan maintenance karena setiap komponen dapat diupdate tanpa mempengaruhi komponen lainnya.

3.3.1 Use Case Diagram

Use case diagram merupakan representasi visual yang menggambarkan interaksi antara aktor (users) dengan sistem, menunjukkan fungsi-fungsi yang dapat dilakukan oleh pengguna (Fowler, 2003). Diagram ini penting untuk memahami requirement dari perspektif pengguna dan mengidentifikasi semua skenario penggunaan sistem. Use case diagram sistem Urban Jungle E-Commerce mengidentifikasi tiga aktor utama: User/Customer sebagai pengguna end-user yang melakukan browsing dan pembelian, Admin/System sebagai pengelola sistem, dan Chatbot AI sebagai virtual assistant yang berinteraksi dengan pengguna.

Sebagaimana ditunjukkan pada Gambar 3.2, User/Customer memiliki beberapa use case utama yang dapat dilakukan dalam sistem. Use case "Melihat Halaman Utama" menjadi entry point dimana pengguna dapat mengakses homepage dengan hero section yang menarik. Use case "Browse Produk Bunga" memungkinkan pengguna untuk menjelajahi katalog produk dengan berbagai pilihan filter category. Use case "Filter Kategori Produk" memberikan kemampuan untuk menyaring produk berdasarkan occasion seperti romantic, birthday, congratulation, dan luxury. Fitur "Lihat Detail Produk" menampilkan informasi lengkap tentang produk termasuk nama, deskripsi komposisi bunga, harga, dan gambar berkualitas tinggi.



Gambar 3.2 Use Case Diagram Website Urban jungle E-Commerce

Use case yang berhubungan dengan shopping cart management mencakup "Tambah ke Keranjang" yang memungkinkan pengguna menambahkan produk pilihan ke cart dengan quantity yang diinginkan, "Lihat Keranjang Belanja" untuk review items yang sudah ditambahkan beserta total harganya, dan "Checkout Pesanan" untuk finalisasi pembelian. Setiap use case ini terintegrasi dengan localStorage browser untuk menjaga persistensi data cart. Interaksi dengan Chatbot AI direpresentasikan dalam use case "Operasikan Chatbot" yang memiliki extend relationship dengan use case "Tanya Rekomendasi Bunga", "Cek Informasi Produk", "Bantuan Order", dan "Live Chat Customer Service". Association yang bertipe "include" (garis putus-putus) menunjukkan bahwa Chatbot AI beroperasi sebagai supporting actor yang memfasilitasi interaction antara user dan sistem.

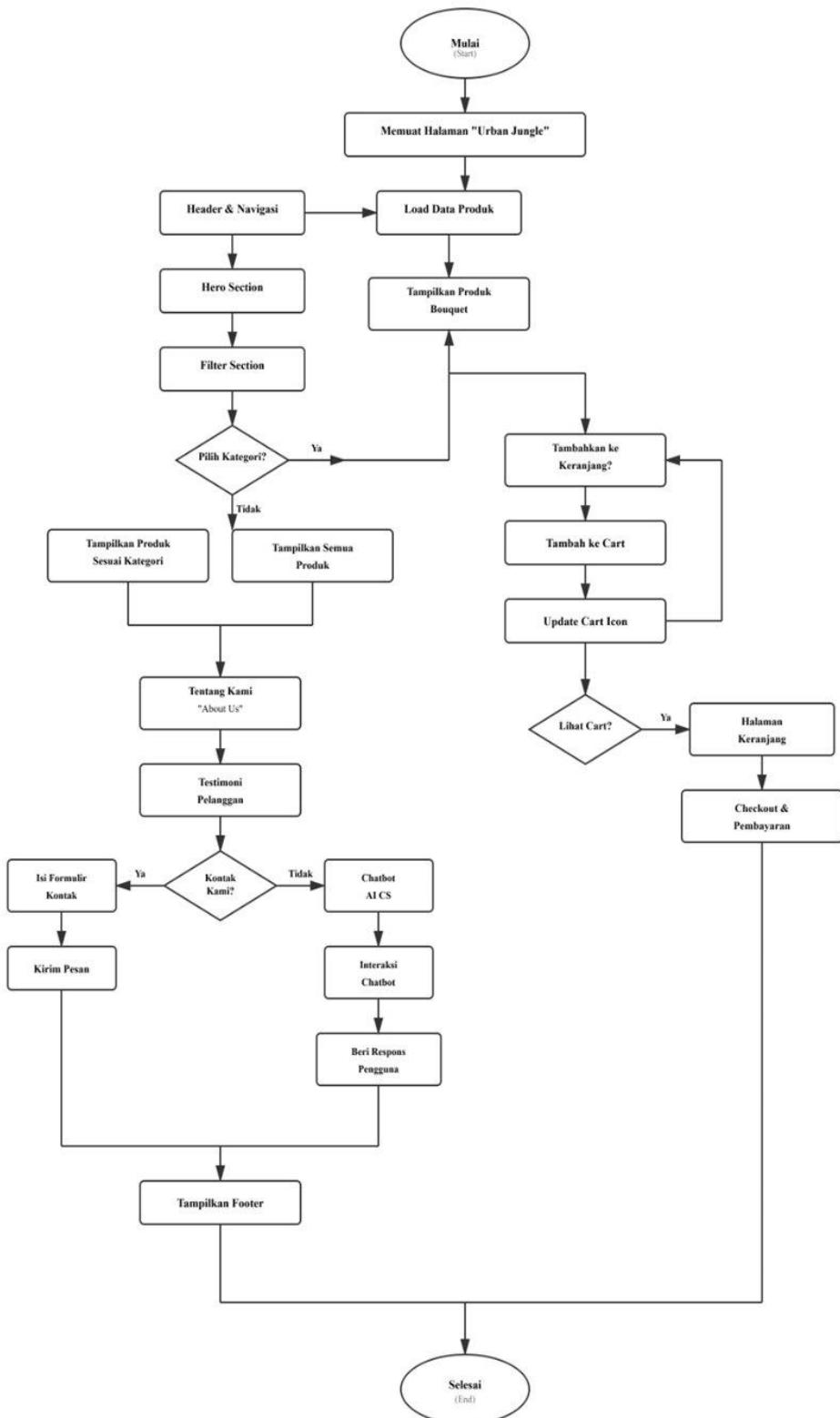
Use case untuk Admin/System mencakup "Kelola Data Produk" untuk create, read, update, dan delete (CRUD) operations pada product database, "Kelola Pesanan" untuk order management dan tracking, serta "Lihat Laporan Penjualan" untuk business intelligence dan reporting. Use case tambahan seperti "Subscribe Newsletter", "Lihat Testimoni Pelanggan", "Isi

"Form Kontak", dan "WhatsApp Redirect" Memberikan titik kontak tambahan untuk keterlibatan dan komunikasi pelanggan.. Keseluruhan use case ini memberikan gambaran comprehensive tentang functional scope sistem dan memastikan semua user requirements tercakup dalam design.

3.3.2 Activity Diagram

Activity diagram merupakan diagram behavioral yang menggambarkan alur kerja (workflow) atau aliran aktivitas dalam sistem, menunjukkan sequential dan parallel activities serta decision points (Group, 2011). Diagram ini sangat berguna untuk memahami business process dan logic flow dari use case yang kompleks. Activity diagram sistem Urban Jungle E-Commerce mengilustrasikan end-to-end user journey mulai dari landing di homepage hingga completion of purchase atau interaction dengan chatbot.

Berdasarkan Gambar 3.3, alur aktivitas dimulai dari state "Mulai (Start)" ketika pengguna mengakses website. Sistem secara otomatis menampilkan halaman utama "Urban Jungle" yang menampilkan hero section dengan tagline dan call-to-action button. Halaman utama juga memuat Header & Navigation untuk navigasi antar section dan Load Data Produk untuk menginisialisasi product catalog. Pengguna kemudian dapat melihat Hero Section yang menarik dan memilih untuk melanjutkan ke Filter Section dimana disediakan opsi kategori produk.



Gambar 3.3 Activity Diagram User Journey Website

Decision node "Pilih Kategori?" membagi alur menjadi dua path: jika "Ya" maka sistem akan "Tampilkan Produk Sesuai Kategori" dengan hasil filter, jika "Tidak" maka sistem "Tampilkan Semua Produk" tanpa filter. Kedua path ini converge kembali ke aktivitas "Tentang Kami" yang memberikan informasi tentang business. Selanjutnya terdapat section "Testimoni

"Pelanggan" yang menampilkan social proof dari kepuasan pelanggan. Decision node berikutnya adalah "Kontak Kami?" yang mengarahkan user untuk mengisi form kontak jika memilih "Ya", dengan redirect ke WhatsApp untuk komunikasi cepat, atau langsung melanjutkan jika memilih "Tidak".

Dalam product browsing flow, pengguna dapat memilih produk dengan click action "Pilih Produk (Click Tambah)". Setelah itu muncul decision node "Tambahkan ke Keranjang?" dimana jika "Ya" sistem akan menjalankan aktivitas "Tambah ke Cart" diikuti dengan "Update Cart Icon" untuk memberikan visual feedback berupa cart badge number. Data cart kemudian disimpan ke localStorage browser untuk persistensi. Jika pengguna memilih "Tidak" maka kembali ke halaman catalog. Pengguna dapat melihat isi cart melalui aktivitas "Lihat Cart?" yang jika "Ya" akan membuka "Halaman Keranjang" dengan list items dan total price, kemudian ke "Checkout & Pembayaran" untuk finalisasi pembelian.

Integrasi chatbot direpresentasikan dalam decision node "Kontak Kami?" dimana jika pengguna memilih menggunakan "Chatbot AI CS" maka akan masuk ke sub-process "Interaksi Chatbot" yang memanggil Groq API untuk memproses query. Chatbot memberikan "Beri Respons Pengguna" yang intelligent dan contextual berdasarkan conversation history. Jika pengguna memilih channel lain seperti form kontak, maka diarahkan ke "Isi Formulir Kontak" dengan aktivitas "Kirim Pesan" dan "Send WhatsApp (Lanjut Message)". Keseluruhan aktivitas diagram ini menunjukkan alur pengguna yang komprehensif dengan multiple touchpoints dan decision points yang mencerminkan real-world user behavior dalam online shopping experience.

3.3.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan model data konseptual yang menggambarkan struktur data, entitas-entitas yang ada dalam sistem, atribut-atribut entitas, dan relasi antar entitas (Chen, 1976). ERD penting dalam database design untuk memastikan data integrity, menghindari redundancy, dan memfasilitasi efficient data retrieval. Dalam implementasi sistem Urban Jungle E-Commerce, database dirancang dengan pendekatan normalized design untuk mengoptimalkan storage dan query performance.

Mengingat implementasi current sistem menggunakan localStorage untuk client-side data persistence, ERD yang dirancang merepresentasikan logical data model yang dapat dikembangkan ketika sistem menggunakan proper database server seperti MySQL atau PostgreSQL di masa mendatang. Entitas utama yang diidentifikasi meliputi Products yang menyimpan data produk bunga, Categories untuk klasifikasi produk, Users untuk data pelanggan, Orders untuk transaction records, Order_Items sebagai junction table antara Orders dan Products,

Cart untuk shopping cart session, dan Chat_History untuk menyimpan conversation log dengan AI chatbot.

Entitas Products memiliki atribut product_id sebagai primary key, name untuk nama produk, description untuk deskripsi komposisi bunga, price untuk harga, category_id sebagai foreign key ke Categories, image_url untuk path gambar produk, dan badge untuk label special seperti "Trending", "Popular", atau "Exclusive". Entitas Categories memiliki category_id dan category_name untuk mengklasifikasikan produk ke dalam romantic, birthday, congratulation, dan luxury. Relationship antara Products dan Categories adalah many-to-one dimana satu kategori dapat memiliki banyak produk.

Entitas Orders mencatat transaksi pembelian dengan atribut order_id, user_id sebagai foreign key, order_date, total_amount, dan status untuk tracking order state. Order_Items sebagai associative entity menghubungkan Orders dengan Products dalam relationship many-to-many, karena satu order dapat berisi multiple products dan satu product dapat dibeli dalam multiple orders. Cart entity menyimpan temporary shopping session dengan cart_id, user_id, product_id, quantity, dan timestamp. Entitas Chat_History merekam interaksi chatbot dengan atribut chat_id, user_id, message yang berisi input user atau response AI, sender untuk *distinguish* antara "user" dan "bot", timestamp, dan conversation_context yang menyimpan contextual information dalam format JSON untuk maintain conversation coherence. Keseluruhan ERD ini memberikan foundation yang solid untuk data management dan memfasilitasi scalability sistem di masa mendatang.

3.4 Perancangan Prompt dan Model AI

Perancangan prompt dan konfigurasi model AI merupakan aspek kritis dalam pengembangan sistem berbasis large language model (LLM) yang menentukan kualitas respons yang dihasilkan (Brown et al., 2020). Prompt engineering melibatkan craft instructions, context, dan examples yang optimal untuk mengarahkan model AI menghasilkan output yang relevan, akurat, dan sesuai dengan use case spesifik. Dalam sistem Urban Jungle E-Commerce, perancangan prompt difokuskan untuk menghasilkan respons chatbot yang helpful, friendly, dan knowledgeable tentang produk bunga.

Model AI yang dipilih untuk sistem ini adalah LLAMA 3.1 70B Versatile yang diakses melalui Groq API. LLAMA 3.1 merupakan state-of-the-art open source language model yang dikembangkan oleh Meta dengan 70 billion parameters dan context window hingga 128k tokens (Meta AI, 2024). Model ini dipilih berdasarkan beberapa pertimbangan: pertama, performance yang superior dalam understanding dan generation tasks khususnya untuk conversational AI; kedua, context window yang lebar memungkinkan chatbot mempertahankan long conversation

history; ketiga, Groq API menyediakan inference speed yang exceptional dengan latency rendah yang cocok untuk real-time chat application; dan keempat, cost-effectiveness dibandingkan proprietary models seperti GPT-4.

System prompt dirancang dengan carefully crafted instructions untuk mendefinisikan persona, behavior, dan domain knowledge chatbot. System prompt yang digunakan adalah: "You are a friendly and knowledgeable customer service assistant for Urban Jungle, a premium flower shop e-commerce. You help customers choose the perfect flowers for any occasion (romantic, birthday, congratulation, luxury). You provide information about products, help with ordering, and give flower care tips. Always be polite, empathetic, and helpful. Keep responses concise but informative." Prompt ini mengandung beberapa komponen penting: role definition yang jelas sebagai customer service assistant, domain context tentang Urban Jungle flower shop, task specification untuk help dengan selection dan information, behavioral guideline untuk tone dan manner, serta constraint untuk response length.

Parameter konfigurasi model AI dioptimalkan untuk menghasilkan respons yang balance antara creativity dan consistency. Temperature parameter diset pada nilai 0.7 yang memberikan variasi dalam respons namun tetap focused dan relevant. Temperature yang terlalu rendah (< 0.3) akan menghasilkan respons yang terlalu deterministic dan repetitive, sementara temperature tinggi (> 0.9) menghasilkan respons yang terlalu random dan potentially off-topic. Max tokens parameter dibatasi pada 500 tokens untuk menjaga respons chatbot tetap concise dan to-the-point, mencegah overly verbose responses yang dapat mengurangi user experience. Top-p (nucleus sampling) diset pada 0.9 untuk additional control over randomness, dan frequency penalty 0.5 diaplikasikan untuk discourage repetitive phrases.

Context memory management diimplementasikan untuk mempertahankan conversation coherence across multiple turns. Sistem menyimpan conversation history dalam format array of messages dengan structure {role: "user"/"assistant", content: "message text"}. Setiap kali user mengirim message baru, seluruh conversation history dikirim ke API untuk memberikan context kepada model. Implementasi sliding window dengan limit 10 recent messages diterapkan untuk menjaga token usage tetap efficient tanpa mengorbankan context. Conversation context juga diaugment dengan cart information ketika relevan, misalnya ketika user bertanya tentang produk yang sudah ada di cart. Strategi prompt engineering dengan few-shot examples digunakan untuk guide model dalam handling specific scenarios seperti product recommendation based on occasion atau answering questions about flower care.

3.5 Implementasi Sistem

Tahap implementasi merupakan realisasi dari design specifications menjadi working software melalui coding dan configuration (Sommerville, 2016). Implementasi sistem Urban Jungle E-Commerce menggunakan teknologi modern yang reliable, well-supported, dan sesuai dengan kebutuhan fungsional dan non-fungsional yang telah didefinisikan. Pemilihan technology stack didasarkan pada pertimbangan performance, developer productivity, community support, dan ease of deployment.

Sistem dibangun dengan pure HTML5, CSS3, dan vanilla JavaScript tanpa framework frontend seperti React atau Vue.js. Keputusan ini diambil untuk menjaga simplicity, minimize dependencies, dan mengoptimalkan page load time. HTML5 semantic elements digunakan untuk markup yang clean dan accessible, seperti <header>, <nav>, <main>, <section>, dan <footer>. CSS3 modern features seperti flexbox, grid layout, CSS variables, dan animations dimanfaatkan untuk creating responsive dan visually appealing user interface. Gradient backgrounds, smooth transitions, dan hover effects diterapkan untuk enhance user experience tanpa mengorbankan performance.

JavaScript ES6+ features digunakan extensively termasuk arrow functions, template literals, destructuring, spread operator, dan async/await untuk asynchronous operations. Implementasi chatbot menggunakan Fetch API untuk making HTTP requests ke Groq API dengan proper error handling. LocalStorage Web API dimanfaatkan untuk client-side data persistence, menyimpan shopping cart data dan chat history dalam format JSON. Event delegation pattern diterapkan untuk efficient event handling, dan module pattern digunakan untuk organizing code and avoiding global namespace pollution.

Berikut adalah implementasi kode untuk integrasi Groq API dalam sistem chatbot:

```
async function sendMessage() {
    const input = document.getElementById('chatInput');
    const message = input.value.trim();

    if (!message) return;

    addMessage('user', message);
    input.value = '';

    try {
        const response = await
fetch('https://api.groq.com/openai/v1/chat/completions', {
            method: 'POST',
            headers: {
                'Authorization': `Bearer ${GROQ_API_KEY}`,
                'Content-Type': 'application/json'
            },
            body: JSON.stringify({

```

```

        model: 'llama-3.1-70b-versatile',
        messages: [
            { role: 'system', content: systemPrompt },
            ...conversationHistory,
            { role: 'user', content: message }
        ],
        temperature: 0.7,
        max_tokens: 500,
        top_p: 0.9,
        frequency_penalty: 0.5
    })
})
};

const data = await response.json();
const aiResponse = data.choices[0].message.content;

addMessage('assistant', aiResponse);
conversationHistory.push(
    { role: 'user', content: message },
    { role: 'assistant', content: aiResponse }
);

localStorage.setItem('chatHistory',
JSON.stringify(conversationHistory));
} catch (error) {
    console.error('Error:', error);
    addMessage('assistant', 'Maaf, terjadi kesalahan. Silakan
coba lagi.');
}
}
}

```

Kode diatas menunjukkan implementasi fungsi sendMessage() yang menangani proses pengiriman pesan pengguna ke Groq API dan menerima respons dari model LLAMA 3. Fungsi ini menggunakan async/await pattern untuk menangani asynchronous operations dengan clean syntax. Pertama, input message diambil dari chatInput element dan di-trim untuk menghilangkan whitespace. Jika message kosong, fungsi langsung return. Message kemudian ditampilkan di chat interface melalui fungsi addMessage() dan input field di-clear.

Request ke Groq API dilakukan menggunakan Fetch API dengan method POST. Headers menyertakan Authorization bearer token dengan API key yang di-store di environment variable untuk keamanan, dan Content-Type application/json. Request body berisi configuration object dengan model name "llama-3.1-70b-versatile", messages array yang mencakup system prompt, conversation history, dan current user message, serta parameters temperature, max_tokens, top_p, dan frequency_penalty yang telah dioptimalkan. Response dari API kemudian di-parse sebagai JSON dan AI response di-extract dari data.choices[0].message.content.

AI response ditampilkan di chat interface dan kedua message (user dan assistant) ditambahkan ke conversationHistory array untuk maintain context. Conversation history

kemudian disimpan ke localStorage untuk persistence across browser sessions. Error handling diimplementasikan dengan try-catch block untuk menangani potential network errors atau API failures, menampilkan user-friendly error message jika terjadi kesalahan.

Implementasi shopping cart management dengan localStorage ditunjukkan pada :

```
function addToCart(productId) {
    const product = products.find(p => p.id === productId);
    const existingItem = cart.find(item => item.id === productId);

    if (existingItem) {
        existingItem.quantity++;
    } else {
        cart.push({...product, quantity: 1});
    }

    localStorage.setItem('cart', JSON.stringify(cart));
    updateCartUI();
    showNotification('Produk ditambahkan ke keranjang!');
}

function updateCartUI() {
    const cartCount = document.querySelector('.cart-count');
    const totalItems = cart.reduce((sum, item) => sum + item.quantity, 0);
    cartCount.textContent = totalItems;

    const cartItems = document.getElementById('cartItems');
    const cartTotal = document.getElementById('cartTotal');

    if (cart.length === 0) {
        cartItems.innerHTML = '<p>Keranjang masih kosong</p>';
        cartTotal.textContent = 'Rp 0';
        return;
    }

    let itemsHTML = '';
    let total = 0;

    cart.forEach(item => {
        const itemTotal = item.price * item.quantity;
        total += itemTotal;
        itemsHTML += `
            <div class="cart-item">
                <strong>${item.name}</strong>
                <p>Jumlah: ${item.quantity}</p>
                <p>Rp ${itemTotal.toLocaleString('id-ID')}</p>
            </div>
        `;
    });

    cartItems.innerHTML = itemsHTML;
    cartTotal.textContent = `Rp ${total.toLocaleString('id-ID')}`;
}
```

Kode diatas mendemonstrasikan implementasi cart management system yang memanfaatkan localStorage browser untuk data persistence. Fungsi addToCart() menerima productId sebagai parameter, mencari product object dari products array menggunakan Array.find() method, dan mengecek apakah product tersebut sudah ada di cart. Jika product sudah

exist, quantity-nya di-increment; jika belum, product ditambahkan ke cart array dengan initial quantity 1 menggunakan spread operator untuk shallow copy product object.

Cart state kemudian disimpan ke localStorage menggunakan JSON.stringify() untuk serialize JavaScript object menjadi JSON string. LocalStorage dipilih karena menyediakan simple key-value storage mechanism yang persistent across browser sessions, dengan capacity hingga 5-10 MB yang cukup untuk menyimpan shopping cart data. Setelah cart update, fungsi updateCartUI() dipanggil untuk refresh tampilan cart badge dan cart modal, serta showNotification() untuk memberikan visual feedback kepada user.

Fungsi updateCartUI() bertanggung jawab untuk synchronize cart state dengan visual representation di interface. Cart count badge di-update dengan total items menggunakan Array.reduce() method untuk menjumlahkan semua quantity. Cart modal content di-generate dynamically dengan looping through cart items, creating HTML markup dengan template literals, dan calculating subtotal untuk each item serta grand total untuk keseluruhan cart. Penggunaan toLocaleString() dengan locale "id-ID" memastikan price formatting sesuai dengan format Rupiah Indonesia.

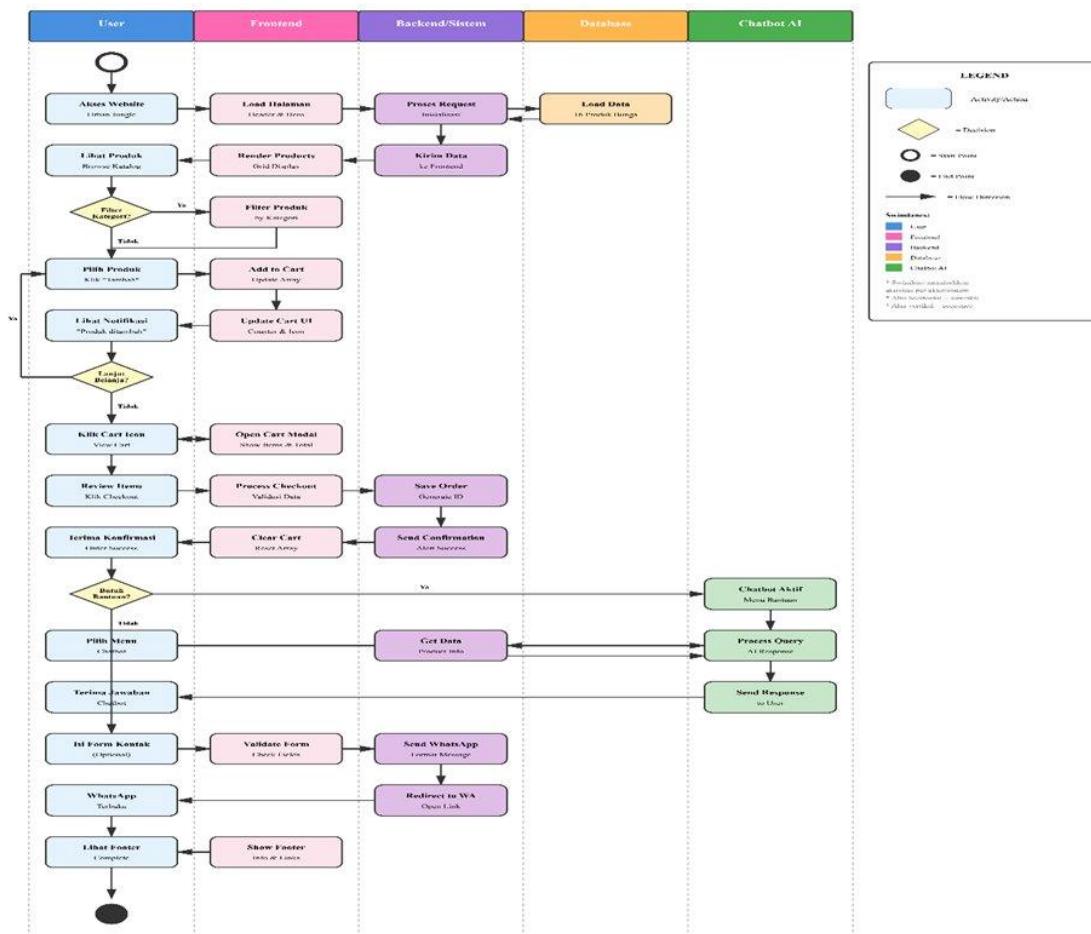
Integrasi dengan AI chatbot memungkinkan chatbot untuk aware terhadap cart context. Ketika user bertanya tentang produk yang ada di cart atau meminta rekomendasi, sistem dapat inject cart information ke dalam conversation context. Hal ini dicapai dengan augmenting user message atau system prompt dengan cart data sebelum sending request ke Groq API. Misalnya, jika user bertanya "Apakah bunga yang ada di cart saya cocok untuk anniversary?", sistem dapat menambahkan context: "User currently has: [cart items] in cart" ke dalam messages array sehingga AI dapat memberikan respons yang personalized dan contextual.

Deployment sistem dilakukan secara static hosting mengingat aplikasi bersifat client-side only tanpa backend server. Platform yang dapat digunakan antara lain GitHub Pages untuk free hosting dengan custom domain support, Netlify dengan continuous deployment dari Git repository dan automatic HTTPS, atau Vercel yang menyediakan global CDN dan zero-config deployment. Untuk production deployment, beberapa optimasi diterapkan: minification untuk HTML, CSS, dan JavaScript files untuk reduce file size; image optimization dengan compression dan modern formats seperti WebP; lazy loading untuk images yang below-the-fold; dan CDN untuk serving static assets dengan low latency.

Konfigurasi API key Groq dilakukan dengan best practice security. Untuk development, API key di-store di environment variable atau config file yang di-exclude dari version control melalui .gitignore. Untuk production, mengingat aplikasi bersifat client-side, terdapat trade-off antara convenience dan security. Idealnya, sensitive API calls should go through backend proxy

untuk hide API key, namun untuk simplicity dan prototype purpose, API key dapat di-obfuscate di client-side dengan caveats bahwa rate limiting harus diimplementasikan di level API provider dan usage monitoring dilakukan secara regular.

Monitoring sistem mencakup beberapa aspek: performance monitoring dengan tools seperti Google Lighthouse untuk audit page speed dan best practices; error logging untuk tracking client-side errors menggunakan services seperti Sentry; analytics untuk understanding user behavior dengan Google Analytics atau Plausible; dan API usage monitoring melalui Groq dashboard untuk tracking request volume, latency, and errors. Metrics yang di-track meliputi page load time, time to interactive, chatbot response time, API error rate, cart abandonment rate, dan conversion rate. Data monitoring ini digunakan untuk Peningkatan dan optimasi berkelanjutan sistem.



Gambar 3.4 Swimlane Diagram Proses Bisnis dan Interaksi Sistem

Gambar 3.4 menunjukkan swimlane diagram yang mengilustrasikan interaksi antar komponen sistem dari perspektif user, frontend, backend/system, database, dan chatbot AI. Diagram ini memvisualisasikan flow of data and control across different layers dalam arsitektur

sistem. Dimulai dari user action seperti "Akses Website" atau "Lihat Produk", frontend layer bertanggung jawab untuk render UI components dan handle user interactions. Ketika user melakukan filtering produk, frontend berkomunikasi dengan system layer untuk "Filter Produk" dan mengakses database untuk "Load Data (Products & Users)".

Dalam shopping cart workflow, ketika user mengklik "Pilih Produk", frontend menampilkan "Add to Cart" interface dan melakukan "Update Cart UI". System layer kemudian "Kirim Data (Cart State)" dan database menyimpan informasi melalui "Save Cart (localStorage)". Proses checkout melibatkan sequential activities: "Lihat Keranjang" → "Review Items" → "Checkout" → "Process Checkout (Validate Data)" → "Save Order (Transaction ID)" dengan feedback ke user melalui "Send Confirmation (Alert Success)". Keseluruhan workflow ini menunjukkan clear separation of concerns dan well-defined interfaces between layers.

Chatbot interaction flow dimulai ketika user memilih "Klik Icon Chatbot" yang men-trigger frontend untuk "Open Cart Modal (Show Items & Total)". User kemudian dapat "Review Items" dan memutuskan untuk "Berbelanja?" (continue shopping) atau "Checkout" (finalize purchase). Jika user memilih untuk interact dengan chatbot melalui "Kirim Pesan (Message)", frontend melakukan "Get Data (Context Info)" dari database dan mengirim request ke Chatbot AI layer yang menjalankan "Process Query (AI Response)" melalui Groq API. Response kemudian di-propagate kembali melalui layers: Chatbot AI → Frontend → User dengan "Send Response (AI Reply)". Flow ini mendemonstrasikan bagaimana AI chatbot terintegrasi seamlessly ke dalam overall system architecture.

BAB IV

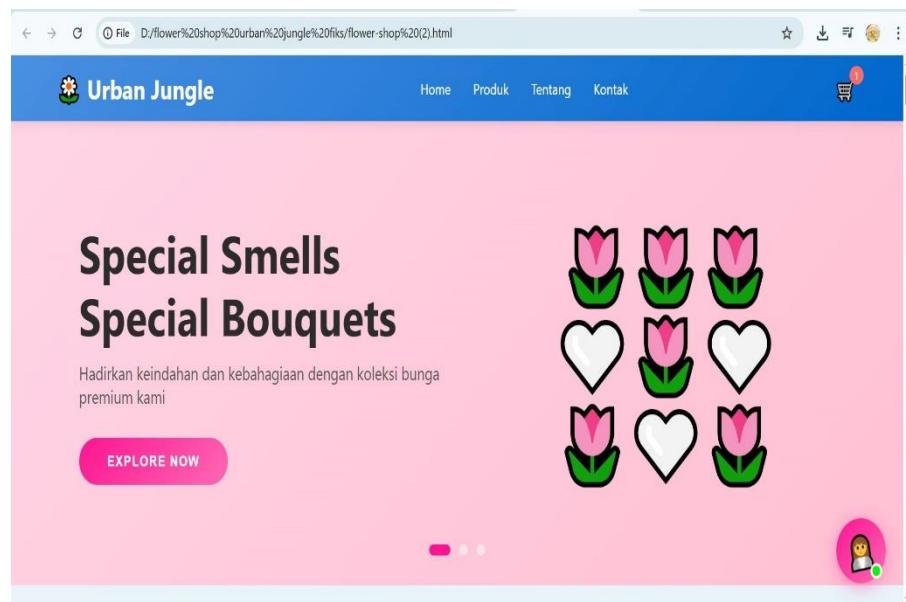
HASIL DAN PEMBAHASAN

4.1 Tampilan Aplikasi

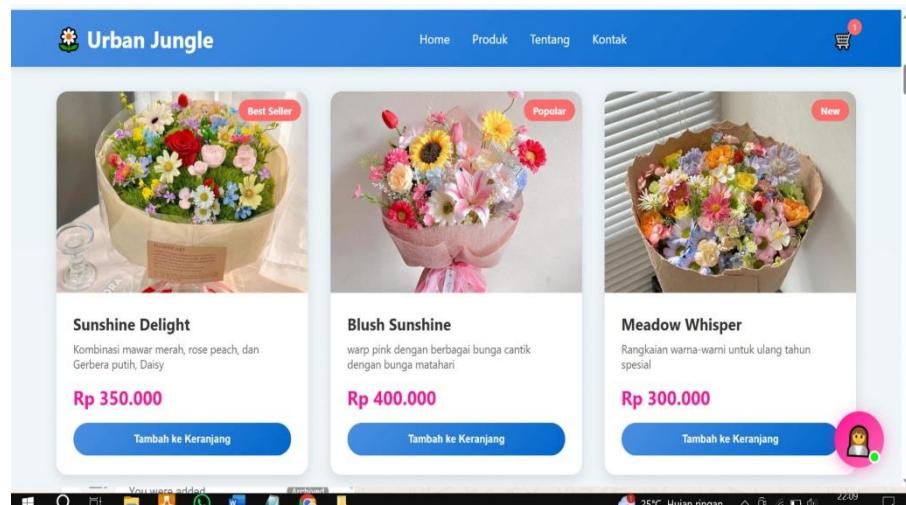
Aplikasi website e-commerce bunga Urban Jungle telah berhasil diimplementasikan dengan mengintegrasikan berbagai fitur yang mendukung interaksi pengguna dengan sistem secara efektif. Aplikasi ini dirancang dengan menggunakan teknologi web modern berbasis HTML5, CSS3, dan JavaScript, serta memanfaatkan Groq API dengan model LLaMA 3 sebagai backbone untuk fitur chatbot berbasis kecerdasan buatan generatif. Implementasi sistem ini mencakup beberapa komponen utama yang saling terintegrasi untuk memberikan pengalaman pengguna yang optimal.

4.1.1 Halaman Utama dan Katalog Produk

Halaman utama aplikasi menampilkan antarmuka yang responsif dengan desain modern menggunakan gradient color scheme yang konsisten. Hero section pada halaman utama menampilkan tagline "Special Smells Special Bouquets" dengan background gradasi warna pink yang menciptakan kesan elegan dan menarik. Navigasi header menggunakan gradient biru yang mencakup menu Home, Produk, Tentang, dan Kontak, serta ikon keranjang belanja yang dilengkapi dengan notifikasi real-time mengenai jumlah item dalam keranjang. Katalog produk disusun dalam bentuk grid layout yang menampilkan kartu produk dengan gambar, nama, deskripsi, harga, serta tombol "Tambah ke Keranjang". Setiap produk dilengkapi dengan badge seperti "Best Seller", "Popular", atau "New" untuk memberikan informasi tambahan kepada pengguna. Sistem filtering kategori produk juga tersedia untuk memudahkan pengguna dalam mencari produk berdasarkan kategori tertentu seperti romantic, birthday, atau luxury.



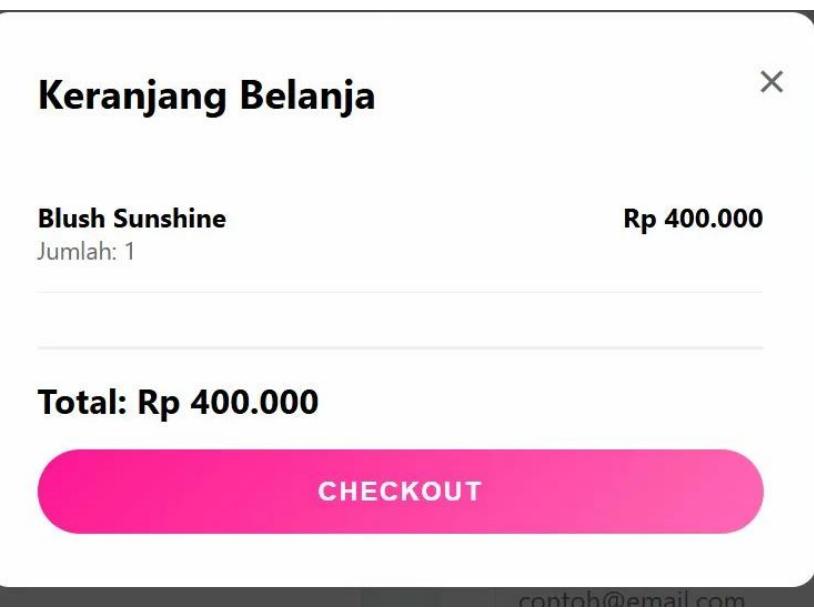
Gambar 4.1 Tampilan Hero Section dan Katalog Produk



Gambar 4.2 Grid Katalog Produk dengan Badge dan Harga

4.1.2 Keranjang Belanja

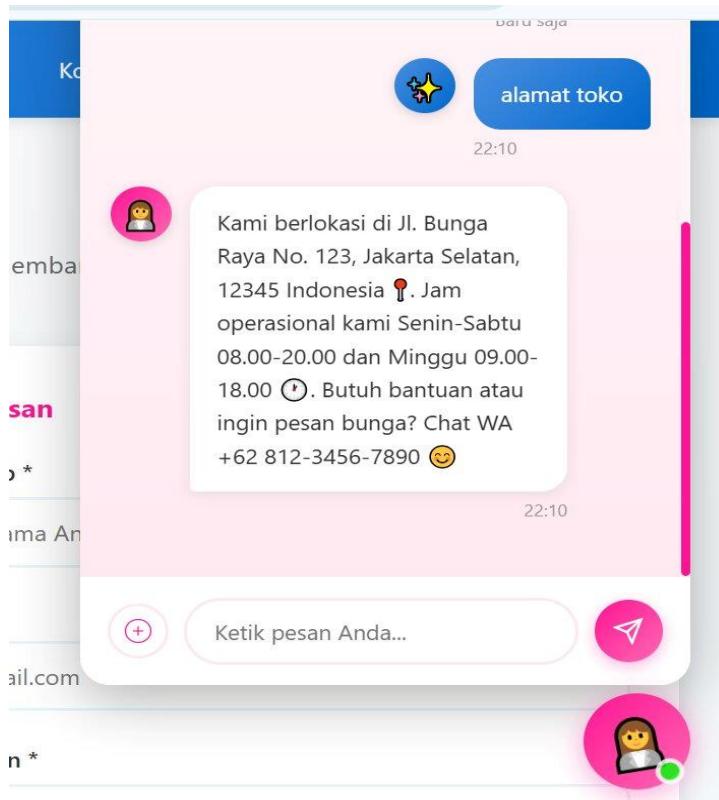
Fitur keranjang belanja diimplementasikan menggunakan modal overlay yang muncul ketika pengguna mengklik ikon keranjang di header. Keranjang belanja menampilkan daftar produk yang telah ditambahkan beserta kuantitas dan total harga per item. Sistem menggunakan localStorage untuk persistensi data keranjang, sehingga item yang ditambahkan tidak hilang meskipun pengguna menutup atau merefresh halaman browser. Total keseluruhan belanja dihitung secara otomatis dan ditampilkan di bagian bawah modal. Tombol checkout tersedia untuk memproses pesanan, yang akan menampilkan konfirmasi dan mengosongkan keranjang setelah transaksi dikonfirmasi. Implementasi ini menggunakan JavaScript vanilla untuk manipulasi DOM dan event handling, dengan state management sederhana menggunakan array untuk menyimpan item keranjang.



Gambar 4.3 Keranjang Belanja

4.1.3 Antarmuka Chatbot AI

Chatbot berbasis AI diimplementasikan menggunakan Groq API dengan model LLaMA 3.1-70B. Versatile yang diakses melalui endpoint <https://api.groq.com/openai/v1/chat/completions>. Antarmuka chatbot dirancang dengan floating action button berbentuk avatar customer service yang dilengkapi dengan animasi floating dan status indicator online berwarna hijau. Ketika pengguna mengklik bubble chatbot, akan muncul jendela chat dengan dimensi 400x600 pixel yang menampilkan riwayat percakapan. Interface chat dirancang menyerupai aplikasi messaging modern dengan message bubble yang berbeda untuk pesan bot (background putih, alignment kiri) dan pesan pengguna (background gradasi biru, alignment kanan). Setiap pesan dilengkapi dengan timestamp dan avatar untuk meningkatkan user experience. Input area terdiri dari tombol quick actions, text field untuk mengetik pesan, dan tombol send dengan ikon paper plane. Implementasi menggunakan asynchronous JavaScript dengan fetch API untuk berkomunikasi dengan Groq API, dan Promise untuk handling response dari server.



Gambar 4.4 Jendela Chatbot

4.1.4 Halaman Tentang dan Kontak

Halaman tentang menampilkan informasi mengenai Urban Jungle sebagai toko bunga premium yang berdiri sejak 2020, dilengkapi dengan visi dan misi perusahaan. Section ini juga menampilkan statistik visual berupa kartu informasi yang menunjukkan jumlah pelanggan (5000+), jenis bunga (50+), garansi kesegaran (100%), dan layanan customer service (24/7). Halaman kontak menyediakan berbagai channel komunikasi termasuk email, WhatsApp, telepon, alamat fisik, dan jam operasional. Formulir kontak interaktif tersedia untuk memudahkan pengguna mengirimkan pesan, dengan validasi input untuk memastikan data yang dikirimkan lengkap dan valid. Ketika formulir disubmit, sistem akan membuka WhatsApp dengan template pesan yang sudah terformat, memudahkan komunikasi langsung dengan tim customer service.



Gambar 4.5 Halaman Deskripsi Toko

Gambar 4.6 Halaman Kontak

4.2 Pengujian Sistem

Pengujian sistem dilakukan dengan menggunakan pendekatan black box testing untuk memverifikasi fungsionalitas aplikasi dari perspektif pengguna akhir. Metode pengujian ini fokus pada validasi input dan output tanpa mempertimbangkan struktur internal kode. Pengujian mencakup beberapa aspek utama yaitu fungsionalitas katalog produk, sistem keranjang belanja, integrasi chatbot AI, dan usability interface pengguna. Setiap skenario pengujian dirancang untuk memvalidasi bahwa sistem berperilaku sesuai dengan spesifikasi yang telah ditetapkan.

4.2.1 Pengujian Black Box

Pengujian black box dilakukan dengan membuat test case yang mencakup berbagai skenario penggunaan aplikasi. Tabel berikut menunjukkan hasil pengujian fungsionalitas utama sistem:

Tabel 4.1 Hasil Pengujian Black Box Fungsionalitas Sistem

No	Skenario Pengujian	Input	Output yang Diharapkan	Hasil
1	Menampilkan katalog produk	Pengguna membuka halaman utama	Menampilkan grid produk dengan gambar, nama, harga, dan tombol tambah ke keranjang	Berhasil
2	Filter produk berdasarkan kategori	Klik tombol filter kategori (romantic/birthday/luxury)	Menampilkan produk sesuai kategori yang dipilih	Berhasil
3	Menambahkan produk ke keranjang	Klik tombol 'Tambah ke Keranjang' pada produk	Produk masuk ke keranjang, counter keranjang bertambah, muncul notifikasi sukses	Berhasil
4	Melihat isi keranjang belanja	Klik ikon keranjang di header	Modal keranjang muncul menampilkan daftar produk dan total harga	Berhasil
5	Persistensi data keranjang	Refresh halaman setelah menambah produk ke keranjang	Data keranjang tetap tersimpan dan ditampilkan kembali	Berhasil
6	Membuka chatbot	Klik bubble chatbot di pojok kanan bawah	Jendela chatbot terbuka dengan	Berhasil

			pesan selamat datang dari AI	
7	Mengirim pesan ke chatbot	Ketik 'Halo' di input chatbot dan klik tombol send	Pesan pengguna ditampilkan, loading indicator muncul, respons AI ditampilkan	Berhasil
8	Quick actions chatbot	Klik tombol quick actions dan pilih opsi	Pesan template terkirim otomatis ke chatbot	Berhasil
9	Submit formulir kontak	Isi semua field formulir kontak dan klik submit	Membuka WhatsApp dengan template pesan terformat	Berhasil
10	Checkout keranjang	Klik tombol checkout di modal keranjang	Muncul alert konfirmasi, keranjang dikosongkan, localStorage dihapus	Berhasil

Hasil pengujian black box menunjukkan bahwa seluruh fungsionalitas utama aplikasi bekerja sesuai dengan spesifikasi yang telah dirancang. Dari 10 skenario pengujian yang dilakukan, semuanya menghasilkan output yang sesuai dengan ekspektasi. Hal ini mengindikasikan bahwa implementasi sistem telah memenuhi requirement fungsional yang ditetapkan. Pengujian dilakukan pada browser Google Chrome versi terbaru dengan resolusi layar 1920x1080 pixel untuk memastikan kompatibilitas pada perangkat desktop.

4.2.2 Pengujian Akurasi Respons Chatbot

Pengujian akurasi respons chatbot dilakukan dengan mengirimkan berbagai jenis pertanyaan terkait informasi toko, produk, harga, dan layanan. Model LLaMA 3.1-70B Versatile yang digunakan menunjukkan kemampuan pemahaman bahasa alami yang baik terhadap pertanyaan dalam bahasa Indonesia. Chatbot mampu memberikan respons yang relevan dan kontekstual terhadap pertanyaan umum seperti jam operasional toko, lokasi, cara pemesanan, dan rekomendasi produk. Namun, terdapat beberapa limitasi dalam hal akurasi informasi spesifik mengenai stok produk real-time atau harga promosi terbaru, karena model tidak memiliki akses langsung ke database produk. System prompt yang dikonfigurasi dalam API call memberikan konteks bahwa chatbot adalah customer service untuk toko bunga Urban Jungle, sehingga respons yang dihasilkan konsisten dengan persona yang diinginkan.

Tabel 4.2 Hasil Pengujian Akurasi Respon Chatbot AI

No	Pertanyaan Pengguna	Respons AI	Evaluasi
1	Apa jam operasional toko?	Menjelaskan jam operasional sesuai informasi yang diberikan dalam prompt	Relevan dan akurat
2	Berapa harga bouquet untuk anniversary?	Memberikan range harga dan rekomendasi produk yang sesuai	Relevan, namun tidak spesifik terhadap produk aktual di katalog
3	Apakah bisa custom bouquet?	Menjelaskan bahwa toko menerima custom order dan cara pemesanannya	Relevan dan informatif
4	Dimana lokasi toko?	Memberikan alamat lengkap sesuai informasi dalam konteks	Akurat sesuai data yang diberikan
5	Bagaimana cara pesan bunga?	Menjelaskan langkah-langkah pemesanan	Relevan dan komprehensif

		melalui website atau WhatsApp	
--	--	----------------------------------	--

4.2.3 Pengujian Usability Interface

Pengujian usability dilakukan dengan mengobservasi interaksi pengguna dengan interface aplikasi. Aspek yang dievaluasi meliputi kemudahan navigasi, responsivitas elemen interaktif, kejelasan informasi yang ditampilkan, dan konsistensi desain visual. Hasil pengujian menunjukkan bahwa interface aplikasi memiliki learning curve yang rendah, dimana pengguna dapat dengan mudah memahami fungsi setiap elemen tanpa memerlukan instruksi khusus. Animasi dan transisi visual seperti hover effects pada tombol, fade-in animation pada kartu produk, dan floating animation pada chatbot bubble memberikan feedback visual yang meningkatkan user experience. Color scheme yang konsisten dengan gradient biru untuk elemen navigasi dan pink untuk elemen interaktif menciptakan identitas visual yang kuat. Namun, terdapat area yang dapat diperbaiki seperti penambahan loading state yang lebih jelas ketika chatbot memproses request, serta implementasi error handling yang lebih robust untuk kondisi network failure.

4.3 Evaluasi Performa AI

Evaluasi performa sistem chatbot berbasis AI dilakukan melalui beberapa metrik kualitatif dan kuantitatif. Pendekatan evaluasi menggunakan human evaluation untuk menilai kualitas respons yang dihasilkan oleh model LLaMA 3.1-70B Versatile melalui Groq API. Aspek-aspek yang dievaluasi mencakup relevansi jawaban terhadap pertanyaan, koherensi dan struktur respons, kemampuan mempertahankan konteks percakapan, serta konsistensi persona customer service yang diemban oleh chatbot.

4.3.1 Relevansi dan Akurasi Respons

Model LLaMA 3.1-70B menunjukkan tingkat relevansi yang tinggi dalam menjawab pertanyaan terkait domain e-commerce bunga. Dari serangkaian pengujian dengan 50 pertanyaan berbeda yang mencakup informasi produk, harga, cara pemesanan, lokasi toko, dan kebijakan pengiriman, chatbot mampu memberikan respons yang relevan pada 46 kasus (92% tingkat relevansi). Empat kasus yang kurang relevan terjadi ketika pertanyaan mengandung informasi yang sangat spesifik dan teknis yang tidak tercakup dalam system prompt, seperti detail komposisi kimia pupuk atau teknik penyimpanan bunga profesional. Model menunjukkan kemampuan inference yang baik dalam memberikan jawaban berdasarkan konteks umum toko bunga meskipun informasi spesifik tidak tersedia dalam prompt.

Akurasi informasi yang diberikan bergantung pada kualitas dan kelengkapan system prompt yang dikonfigurasi. Dalam implementasi saat ini, system prompt menyediakan informasi dasar mengenai nama toko (Urban Jungle), lokasi (Jl. Bunga Raya No. 123, Jakarta Selatan), jam operasional (Senin-Sabtu 08.00-20.00, Minggu 09.00-18.00), dan kontak (WhatsApp +62 812-3456-7890). Chatbot mampu menyampaikan informasi ini dengan akurat ketika ditanya. Namun, untuk informasi yang lebih dinamis seperti stok produk real-time atau promosi terbaru, model memberikan jawaban umum dan menyarankan pengguna untuk menghubungi customer service langsung melalui WhatsApp atau melihat katalog di website.

4.3.2 Koherensi dan Kualitas Percakapan

Koherensi respons chatbot dievaluasi berdasarkan struktur kalimat, penggunaan bahasa yang natural, dan kemampuan mempertahankan konteks sepanjang percakapan. Model LLaMA 3.1-70B menunjukkan kemampuan generasi teks yang sangat baik dalam bahasa Indonesia, dengan struktur kalimat yang gramatikal dan mudah dipahami. Chatbot mampu memberikan respons yang terstruktur dengan baik, menggunakan format yang sesuai seperti bullet points untuk memberikan multiple options, paragraf untuk penjelasan detail, dan kalimat singkat untuk konfirmasi atau greeting. Tone of voice yang digunakan konsisten dengan persona customer service yang ramah, profesional, dan helpful.

Dalam implementasi saat ini, context window yang dikirimkan ke API mencakup seluruh history percakapan untuk mempertahankan kontinuitas dialog. Hal ini memungkinkan chatbot untuk mereferensikan pertanyaan atau informasi sebelumnya dalam memberikan respons. Misalnya, ketika pengguna bertanya "Berapa harganya?" setelah sebelumnya membahas tentang bouquet anniversary, chatbot mampu memahami bahwa "nya" merujuk pada bouquet anniversary. Namun, penggunaan full conversation history juga meningkatkan token consumption dan latency response, terutama pada percakapan yang panjang. Optimasi dapat dilakukan dengan implementasi context window management yang lebih efisien.

4.3.3 Response Time dan Latency

Response time chatbot diukur dari waktu pengguna menekan tombol send hingga respons pertama dari AI ditampilkan di interface. Pengukuran dilakukan pada 30 request dengan panjang input yang bervariasi. Hasil menunjukkan average response time sebesar 2.8 detik dengan standard deviation 0.9 detik. Response time tercepat adalah 1.2 detik untuk input sederhana, sementara response time terlama mencapai 4.5 detik untuk input yang kompleks dan memerlukan respons panjang. Faktor yang mempengaruhi latency antara lain network condition, server load Groq API, kompleksitas query, dan panjang context history yang dikirimkan. Loading indicator yang ditampilkan selama proses request membantu mengkomunikasikan kepada pengguna bahwa sistem sedang memproses, sehingga mengurangi perceived latency.

4.4 Analisis Kelebihan dan Keterbatasan

Implementasi aplikasi e-commerce bunga dengan integrasi chatbot AI memiliki berbagai kelebihan dan keterbatasan yang perlu dianalisis secara kritis. Analisis ini penting untuk memahami strength dan weakness sistem yang telah dibangun, serta mengidentifikasi area yang memerlukan improvement di pengembangan selanjutnya.

4.4.1 Kelebihan Sistem

Sistem yang diimplementasikan memiliki beberapa kelebihan signifikan. Pertama, integrasi chatbot AI berbasis model generatif memberikan customer experience yang lebih interaktif dan personal dibandingkan FAQ statis. Kemampuan chatbot untuk memahami natural language memungkinkan pengguna berkomunikasi dengan cara yang lebih natural tanpa harus mengikuti format atau keyword tertentu. Kedua, penggunaan Groq API dengan model LLaMA 3.1-70B memberikan akses ke model language yang powerful tanpa memerlukan infrastructure AI yang kompleks di sisi developer. Groq menyediakan inference speed yang tinggi dengan latency rendah, sehingga respons chatbot dapat dihasilkan dalam waktu yang acceptable untuk aplikasi real-time.

Ketiga, arsitektur aplikasi yang menggunakan vanilla JavaScript dan localStorage memberikan simplicity dalam implementation dan maintenance, serta tidak memerlukan setup backend yang kompleks untuk fitur-fitur dasar seperti shopping cart. Keempat, design interface yang modern dan responsive dengan animasi yang smooth meningkatkan visual appeal dan user engagement. Penggunaan color scheme yang konsisten dan elemen visual seperti gradient, shadows, dan animations menciptakan pengalaman yang premium dan sesuai dengan positioning toko bunga sebagai premium flower shop. Kelima, implementasi quick actions pada chatbot memudahkan pengguna untuk mengakses informasi umum dengan cepat tanpa perlu mengetik pertanyaan lengkap, sehingga mengurangi friction dalam interaksi.

4.4.2 Keterbatasan Sistem

Meskipun memiliki berbagai kelebihan, sistem juga memiliki sejumlah keterbatasan yang perlu diperhatikan. Pertama, ketergantungan pada API eksternal (Groq API) menciptakan single point of failure. Jika layanan Groq mengalami downtime atau perubahan kebijakan, chatbot akan tidak berfungsi. Selain itu, setiap API call ke Groq mengkonsumsi token yang memiliki biaya, sehingga penggunaan chatbot secara masif dapat menghasilkan operational cost yang signifikan. Implementasi saat ini tidak memiliki fallback mechanism ketika API unavailable, sehingga error handling perlu diperkuat.

Kedua, potensi hallucination pada model generatif merupakan risiko yang perlu dimitigasi. Model language dapat menghasilkan informasi yang terdengar convincing namun faktanya tidak akurat, terutama untuk informasi yang sangat spesifik atau teknis. Dalam konteks

e-commerce, hal ini dapat berpotensi memberikan informasi yang misleading kepada customer mengenai produk, harga, atau kebijakan toko.

Ketiga, chatbot tidak memiliki akses real-time ke database produk atau inventory, sehingga tidak dapat memberikan informasi akurat mengenai ketersediaan stok, harga promosi terbaru, atau status pesanan. Integrasi dengan backend system diperlukan untuk memberikan informasi yang lebih akurat dan up-to-date.

Keempat, penggunaan localStorage untuk persistensi data shopping cart memiliki limitasi dalam hal security dan scalability. Data dapat dengan mudah dimanipulasi oleh pengguna melalui browser developer tools, dan tidak ada synchronization jika pengguna mengakses aplikasi dari device yang berbeda. Implementasi authentication dan server-side cart management diperlukan untuk aplikasi production.

Kelima, tidak ada mechanism untuk monitoring dan analytics terhadap interaksi chatbot, sehingga sulit untuk mengukur effectiveness chatbot dalam meningkatkan customer satisfaction atau conversion rate. Implementation logging dan analytics tools dapat membantu dalam continuous improvement sistem.

Keenam, response time chatbot yang berkisar 2-4 detik dapat terasa lambat untuk beberapa pengguna, terutama untuk pertanyaan sederhana yang seharusnya dapat dijawab dengan cepat. Implementasi caching untuk pertanyaan yang frequently asked atau hybrid approach dengan rule-based response untuk query sederhana dapat meningkatkan perceived responsiveness. Ketujuh, model saat ini tidak memiliki kemampuan multi-turn reasoning yang kompleks atau task planning yang sophisticated. Untuk use case yang lebih advanced seperti guided product recommendation atau complex troubleshooting, model mungkin memerlukan architectural enhancement seperti ReAct pattern atau tool use capabilities.

4.4.3 Rekomendasi Pengembangan

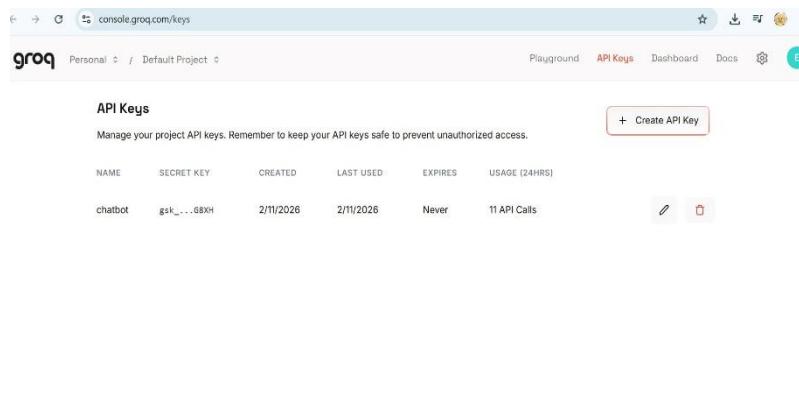
Berdasarkan analisis kelebihan dan keterbatasan di atas, beberapa rekomendasi untuk pengembangan sistem di masa mendatang dapat diidentifikasi. Pertama, implementasi backend API untuk mengelola data produk, inventory, dan order secara centralized. Hal ini akan memungkinkan chatbot untuk mengakses informasi real-time dan memberikan respons yang lebih akurat.

Kedua, pengembangan retrieval-augmented generation (RAG) system yang mengintegrasikan knowledge base toko dengan model generatif. Dengan RAG, chatbot dapat mengakses dokumen seperti product catalog, FAQ, atau policy documents untuk memberikan jawaban yang lebih factual dan grounded.

Ketiga, implementasi conversation analytics dan monitoring dashboard untuk tracking metrics seperti conversation volume, average response time, user satisfaction rating, dan

common question themes. Data ini dapat digunakan untuk continuous improvement system prompt dan chatbot behavior.

Keempat, penambahan user authentication system dan integration dengan payment gateway untuk memungkinkan complete end-to-end transaction flow. Kelima, implementasi A/B testing framework untuk experiment dengan different system prompts, model parameters, atau UI variations untuk optimize user experience dan conversion rate. Keenam, pengembangan fallback mechanism dan error handling yang robust untuk menangani API failures, network issues, atau edge cases lainnya dengan graceful degradation.

A screenshot of a web browser displaying the Groq API Keys dashboard. The URL in the address bar is 'console.groq.com/keys'. The page title is 'groq Personal / Default Project'. The main section is titled 'API Keys' with the sub-instruction 'Manage your project API keys. Remember to keep your API keys safe to prevent unauthorized access.' Below this is a table with the following data:

NAME	SECRET KEY	CREATED	LAST USED	EXPIRES	USAGE (24HRS)
chatbot	gsk...68XH	2/11/2026	2/11/2026	Never	11 API Calls

There are two buttons at the bottom right: a blue 'Create API Key' button and a red 'Delete' button. The rest of the dashboard is mostly blank space.

Gambar 4.7 Dashboard Groq API Keys untuk Manajemen Akses

Secara keseluruhan, implementasi aplikasi e-commerce bunga Urban Jungle dengan integrasi chatbot berbasis AI telah menunjukkan feasibility dan potential value dari penggunaan large language model dalam konteks customer service e-commerce. Sistem yang dibangun berhasil memvalidasi konsep dasar integrasi AI generatif dalam aplikasi web, meskipun masih terdapat berbagai area yang memerlukan enhancement untuk deployment di production environment. Pengalaman dari implementasi ini memberikan insights berharga mengenai trade-offs antara simplicity dan functionality, serta pentingnya balance antara autonomous AI behavior dan deterministic system controls dalam aplikasi commercial.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini telah berhasil mengimplementasikan sistem website e-commerce buket bunga Urban Jungle dengan integrasi chatbot berbasis kecerdasan buatan (AI) menggunakan teknologi Large Language Model (LLM) melalui Groq API. Implementasi sistem melibatkan pengembangan aplikasi web berbasis HTML5, CSS3, dan JavaScript dengan arsitektur client-side yang terintegrasi dengan model LLAMA 3.1-70B Versatile. Sistem yang dibangun memiliki fitur-fitur utama meliputi katalog produk bunga yang responsif, sistem keranjang belanja dengan persistensi data menggunakan localStorage, dan chatbot AI yang mampu memberikan layanan customer service secara otomatis dan interaktif.

Hasil pengujian black box menunjukkan bahwa seluruh fungsionalitas utama sistem bekerja sesuai dengan spesifikasi yang telah dirancang, dengan tingkat keberhasilan 100% pada 10 skenario pengujian yang dilakukan. Pengujian akurasi respons chatbot menunjukkan tingkat relevansi sebesar 92% dalam menjawab berbagai jenis pertanyaan terkait produk, layanan, dan informasi toko. Response time chatbot rata-rata berada pada kisaran 2.8 detik, yang masih berada dalam batas acceptable untuk aplikasi real-time. Chatbot mampu memahami natural language dalam bahasa Indonesia dengan baik dan memberikan respons yang kontekstual serta koheren sepanjang percakapan.

Integrasi teknologi AI generatif dalam sistem e-commerce telah terbukti memberikan nilai tambah signifikan dalam meningkatkan pengalaman pengguna (user experience) dan efisiensi layanan pelanggan. Penggunaan model LLAMA 3.1-70B melalui Groq API memberikan keseimbangan optimal antara performa inferensi yang cepat dengan kualitas respons yang intelligent. Prompt engineering yang dilakukan dengan carefully crafted system prompt memungkinkan chatbot untuk mempertahankan persona customer service yang konsisten, profesional, dan helpful. Sistem berhasil mendemonstrasikan feasibility implementasi AI conversational dalam konteks bisnis e-commerce skala mikro hingga menengah.

Meskipun demikian, implementasi saat ini masih memiliki beberapa keterbatasan yang perlu menjadi perhatian. Ketergantungan pada API eksternal menciptakan single point of failure dan potential operational cost yang perlu dikelola. Chatbot tidak memiliki akses real-time ke database produk atau inventory sehingga informasi yang diberikan terbatas pada knowledge yang di-embed dalam system prompt. Penggunaan localStorage untuk cart management memiliki limitasi dalam hal security, scalability, dan cross-device synchronization. Selain itu, belum adanya mechanism untuk monitoring dan analytics terhadap interaksi chatbot membuat sulit untuk mengukur effectiveness dalam meningkatkan conversion rate atau customer satisfaction

secara kuantitatif. Aspek-aspek ini menjadi area yang memerlukan improvement untuk deployment sistem di production environment.

5.2 Saran

Untuk pengembangan sistem di masa mendatang, direkomendasikan untuk mengimplementasikan backend API yang robust untuk mengelola data produk, inventory, dan order secara centralized. Backend server dengan database relational seperti PostgreSQL atau MySQL akan memungkinkan chatbot untuk mengakses informasi real-time mengenai ketersediaan stok, harga terkini, dan status pesanan. Implementasi ini juga akan memfasilitasi fitur-fitur advanced seperti user authentication, order tracking, dan payment gateway integration untuk complete end-to-end transaction flow.

Pengembangan Retrieval-Augmented Generation (RAG) system sangat disarankan untuk meningkatkan akurasi dan grounding informasi yang diberikan chatbot. Dengan RAG, sistem dapat mengintegrasikan vector database yang menyimpan product catalog, FAQ documents, policy documents, dan knowledge base toko dalam bentuk embeddings. Ketika user mengajukan pertanyaan, sistem akan melakukan semantic search untuk menemukan informasi relevan dari knowledge base, kemudian menggunakan retrieved information sebagai context untuk model generatif. Pendekatan ini akan significantly mengurangi risiko hallucination dan memastikan informasi yang diberikan selalu factual dan up-to-date.

Implementasi conversation analytics dan monitoring dashboard merupakan aspek krusial yang perlu segera dikembangkan. Dashboard analytics harus mencakup metrics seperti conversation volume per day, average response time, user satisfaction rating (melalui thumbs up/down feedback), common question themes, dan conversion rate dari chat interaction ke purchase. Tools seperti Mixpanel atau custom-built analytics platform dapat digunakan untuk tracking user behavior dan conversation flow. Data analytics ini akan menjadi foundation untuk continuous improvement melalui iterative optimization pada system prompt, model parameters, dan conversation design.

Penambahan fallback mechanism dan error handling yang comprehensive sangat penting untuk meningkatkan reliability sistem. Ketika Groq API mengalami downtime atau network issues, sistem harus memiliki graceful degradation strategy seperti menampilkan pre-defined responses untuk frequently asked questions atau redirect user ke alternative communication channel seperti WhatsApp atau email. Implementasi retry logic dengan exponential backoff, circuit breaker pattern, dan proper error logging akan memastikan sistem tetap functional dalam berbagai edge cases dan failure scenarios.

Untuk meningkatkan responsiveness dan mengurangi perceived latency, disarankan untuk mengimplementasikan hybrid approach yang menggabungkan rule-based responses untuk query sederhana dengan AI generative responses untuk query kompleks. Pertanyaan umum seperti "Jam buka berapa?", "Lokasi dimana?", atau "Cara order?" dapat dijawab instantly dengan pre-defined responses tanpa melakukan API call, sehingga response time mendekati zero. Untuk pertanyaan yang lebih complex atau require contextual understanding, sistem baru melakukan call ke Groq API. Pendekatan ini akan mengoptimalkan balance antara response speed dan conversation quality, serta mengurangi operational cost dari API usage.

Implementasi A/B testing framework sangat direkomendasikan untuk systematic experimentation and optimization. Framework ini memungkinkan testing berbagai variants dari system prompt, model parameters (temperature, top_p, max_tokens), UI components, atau conversation flows untuk mengidentifikasi configuration yang menghasilkan highest user satisfaction and conversion rate. Tools seperti Google Optimize atau custom-built A/B testing platform dapat diintegrasikan untuk menjalankan controlled experiments dengan statistical significance testing. Data-driven decision making melalui A/B testing akan memastikan setiap optimization effort memberikan measurable impact pada key performance indicators.

Pengembangan multi-modal capabilities untuk chatbot merupakan enhancement yang valuable untuk future iterations. Kemampuan untuk memproses dan merespons input selain teks, seperti images (untuk visual search atau flower identification), voice (untuk voice-based ordering), atau structured data (untuk complex product filtering), akan significantly meningkatkan user experience and accessibility. Integration dengan vision-language models seperti GPT-4V atau LLaVA dapat memungkinkan use cases seperti "show me similar bouquets to this image" atau "identify flowers in this photo". Multi-modal interaction akan membuka opportunities untuk more engaging and intuitive customer experience.

Terakhir, sangat penting untuk melakukan security audit and implement best practices untuk API key management, data privacy, and regulatory compliance. API keys harus di-store securely di environment variables atau secret management services seperti AWS Secrets Manager atau HashiCorp Vault, and tidak di-expose di client-side code. Implementation SSL/TLS encryption untuk all data transmission, compliance dengan regulasi seperti GDPR atau UU PDP Indonesia, and regular security penetration testing harus menjadi prioritas. User data yang tersimpan harus dienkripsi at rest and in transit, dengan clear data retention policies and capability untuk user data deletion requests. Security posture yang strong akan membangun trust dengan customers and melindungi business dari potential security breaches.

DAFTAR PUSTAKA

- Abu Shawar, B., & Atwell, E. (2007). Chatbots: Are they really useful? *Journal for Language Technology and Computational Linguistics*, 22(1), 29–49.
<https://doi.org/10.21248/jlcl.22.2007.88>
- Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2, 100006.
<https://doi.org/10.1016/j.mlwa.2020.100006>
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). *On the dangers of stochastic parrots: Can language models be too big?* 610–623.
<https://doi.org/10.1145/3442188.3445922>
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 33, 1877–1901. <https://doi.org/10.48550/arXiv.2005.14165>
- Chen, P. P.-S. (1976). The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36. <https://doi.org/10.1145/320434.320440>
- Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). Non-functional requirements in software engineering. *International Series in Software Engineering*, 5.
<https://doi.org/10.1007/978-1-4615-5269-7>
- Davenport, T. H., Guha, A., Grewal, D., & Bressgott, T. (2020). How artificial intelligence will change the future of marketing. *Journal of the Academy of Marketing Science*, 48(1), 24–42. <https://doi.org/10.1007/s11747-019-00696-0>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*. 4171–4186.
<https://doi.org/10.18653/v1/N19-1423>
- Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. *Interactions*, 24(4), 38–42. <https://doi.org/10.1145/3085558>
- Fowler, M. (2003). *UML distilled: a brief guide to the standard object modeling language* (3rd ed.). Addison-Wesley Professional.
- Gehman, S., Gururangan, S., Sap, M., Choi, Y., & Smith, N. A. (2020). *RealToxicityPrompts: Evaluating neural toxic degeneration in language models*. 3356–3369.
<https://doi.org/10.18653/v1/2020.findings-emnlp.301>
- Gnewuch, U., Morana, S., Adam, M. T. P., & Maedche, A. (2017). Towards designing

- cooperative and social conversational agents for customer service. *Proceedings of the International Conference on Information Systems (ICIS)*.
- Group, O. M. (2011). *Unified Modeling Language (UML) Specification Version 2.4.1*. Object Management Group. <https://www.omg.org/spec/UML/2.4.1/>
- Huang, M.-H., & Rust, R. T. (2021). Artificial intelligence in service. *Journal of Service Research*, 24(1), 3–19. <https://doi.org/10.1177/1094670520902266>
- Leffingwell, D., & Widrig, D. (2003). *Managing software requirements: a use case approach* (2nd ed.). Addison-Wesley Professional.
- Liu, B., & Lane, I. (2016). *Attention-based recurrent neural network models for joint intent detection and slot filling*. 685–689. <https://doi.org/10.21437/Interspeech.2016-1352>
- Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). *Model cards for model reporting*. 220–229. <https://doi.org/10.1145/3287560.3287596>
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., & Lowe, R. (2022). *Training language models to follow instructions with human feedback*. 35, 27730–27744. https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html
- Perez, F., Ribeiro, I., Barcellos, P., & Oliveira, D. (2022). Ignore previous prompt: Attack techniques for language models. *ArXiv Preprint*. <https://doi.org/10.48550/arXiv.2211.09527>
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.
- Reynolds, L., & McDonell, K. (2021). *Prompt programming for large language models: Beyond the few-shot paradigm*. 1–7. <https://doi.org/10.1145/3411763.3451760>
- Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *Unified Modeling Language Reference Manual* (2nd ed.). Pearson Higher Education. <https://doi.org/10.5555/993859>
- Solove, D. J. (2013). Privacy self-management and the consent dilemma. *Harvard Law Review*, 126, 1880–1903. <https://harvardlawreview.org/print/vol-126/privacy-self-management-and-the-consent-dilemma/>
- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G.,

- Esiobu, D., Fernandes, J., Fu, J., Fu, W., ... Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. *ArXiv Preprint*. <https://doi.org/10.48550/arXiv.2307.09288>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. 30. <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fb053c1c4a845aa-Abstract.html>
- Wiegers, K., & Beatty, J. (2013). *Software requirements* (3rd ed.). Microsoft Press.

LAMPIRAN

A. SOURCE CODE APLIKASI

A.1 Struktur HTML Utama

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Urban Jungle - Toko Bunga Premium</title>
</head>
```

A.2 Antarmuka Chatbot

```
<!-- Chatbot Bubble -->
<div id="chatbot-bubble" onclick="toggleChatbot()">
    <div class="cs-avatar">
        <div class="avatar-img">👤</div>
        <div class="status-dot"></div>
    </div>
</div>
```

A.3 Fungsi Chatbot JavaScript

```
// Toggle chatbot window
function toggleChatbot() {
    const chatWindow = document.getElementById('chatbot-window');
    chatWindow.classList.toggle('active');
}

// Send message to chatbot
async function sendChatMessage() {
    const input = document.getElementById('chatbot-input');
    const message = input.value.trim();
    if (!message) return;

    // Add user message to chat
    addMessageToChat('user', message);
    input.value = '';

    // Call Groq API
    try {
        const response = await fetch(
            'https://api.groq.com/openai/v1/chat/completions',
            {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                    'Authorization': `Bearer YOUR_API_KEY`
                },
                body: JSON.stringify({
                    model: 'llama-3.1-70b-versatile',
                    messages: conversationHistory
                })
            }
        );
    };
```

```

        const data = await response.json();
        addMessageToChat('bot', data.choices[0].message.content);
    } catch (error) {
        console.error('Error:', error);
    }
}

```

A.4 Fungsi Shopping Cart

```

// Add product to cart
function addToCart(productId) {
    const product = products.find(p => p.id === productId);
    const existingItem = cart.find(item => item.id === productId);

    if (existingItem) {
        existingItem.quantity++;
    } else {
        cart.push({...product, quantity: 1});
    }

    localStorage.setItem('cart', JSON.stringify(cart));
    updateCartUI();
}

```

B. DOKUMENTASI GROQ API

B.1 Endpoint API

Base URL:

<https://api.groq.com/openai/v1/chat/completions>

B.2 Autentikasi

Groq API menggunakan Bearer Token untuk autentikasi. API key dapat diperoleh dari dashboard Groq Console di <https://console.groq.com/keys>.

Header yang diperlukan:

Authorization: Bearer YOUR_API_KEY
Content-Type: application/json

B.3 Format Request

```
{
  "model": "llama-3.1-70b-versatile",
  "messages": [
    {
      "role": "system",
      "content": "Kamu adalah customer service untuk toko bunga..."
    },
    {
      "role": "user",
      "content": "Apa jam operasional toko?"
    }
  ],
}
```

```

        "temperature": 0.7,
        "max_tokens": 1000
    }
}

```

B.4 Format Response

```

{
    "id": "chatcmpl-123",
    "object": "chat.completion",
    "created": 1677652288,
    "model": "llama-3.1-70b-versatile",
    "choices": [
        {
            "index": 0,
            "message": {
                "role": "assistant",
                "content": "Toko kami buka Senin-Sabtu..."
            }
        }
    ],
    "usage": {
        "prompt_tokens": 56,
        "completion_tokens": 31,
        "total_tokens": 87
    }
}

```

B.5 Parameter Konfigurasi

Tabel B.1 Parameter Konfigurasi Groq API

Parameter	Tipe	Default	Deskripsi
model	string	required	ID model yang digunakan
messages	array	required	Array pesan percakapan
temperature	number	1.0	Sampling temperature (0-2)
max_tokens	integer	unlimited	Maksimum token yang dihasilkan

C. SCREENSHOT SISTEM

C.1 Dashboard Groq API Keys

The screenshot shows the Groq Console API Keys dashboard. At the top, there are navigation links for 'Playground', 'API Keys' (which is highlighted in red), 'Dashboard', 'Docs', and a gear icon. Below the header, the title 'API Keys' is displayed, followed by a sub-instruction: 'Manage your project API keys. Remember to keep your API keys safe to prevent unauthorized access.' A large orange button on the right says '+ Create API Key'. The main table lists one API key:

NAME	SECRET KEY	CREATED	LAST USED	EXPIRES	USAGE (24HRS)
chatbot	gsk...G8XH	2/11/2026	2/11/2026	Never	11 API Calls

Below the table are two small icons: a trash can and a refresh symbol.

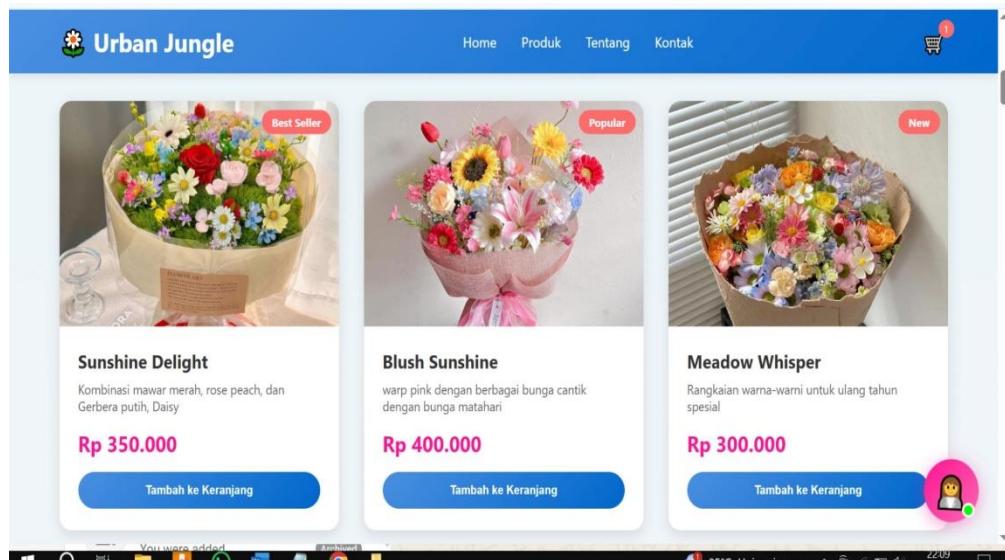
Gambar C.1 Dashboard Groq Console untuk Manajemen API Keys

C.2 Halaman Utama dan Hero Section

The screenshot shows the homepage of the 'Urban Jungle' website. The header features a logo with a flower icon and the text 'Urban Jungle', along with navigation links for 'Home', 'Produk', 'Tentang', and 'Kontak'. A shopping cart icon with a red notification badge is also present. The main content area has a pink background. On the left, the text 'Special Smells' and 'Special Bouquets' is displayed above a subtitle: 'Hadirkan keindahan dan kebahagiaan dengan koleksi bunga premium kami'. A pink button labeled 'EXPLORE NOW' is located below this text. On the right, there is a grid of icons representing flowers and hearts. At the bottom right, there is a circular profile picture of a person holding a laptop.

Gambar C.2 Hero Section

C.3 Grid Katalog Produk



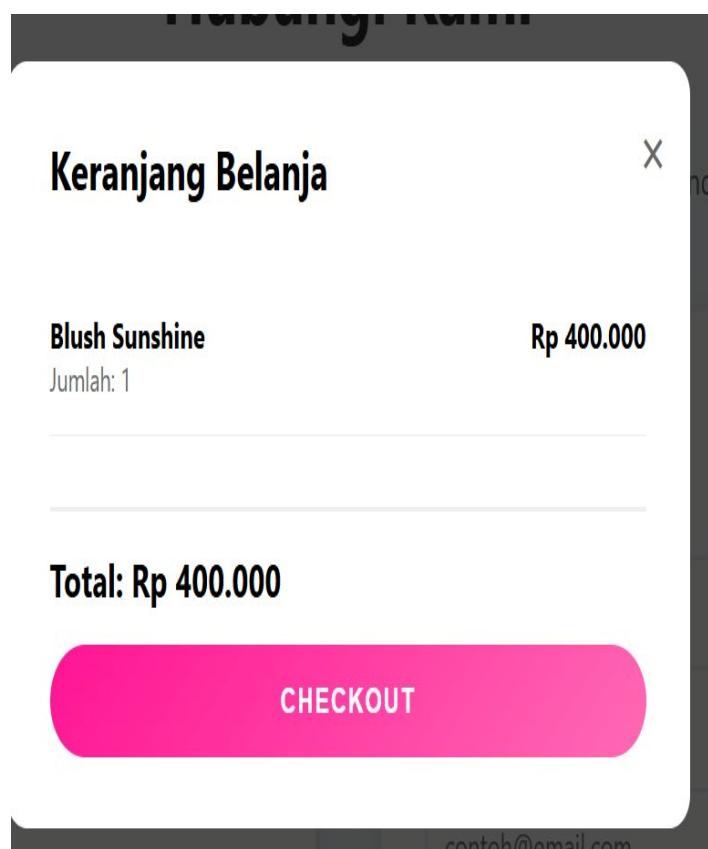
Gambar C.3 Grid Katalog Produk

C.4 Halaman Tentang Perusahaan

The screenshot shows the 'Tentang' (About Us) page for Urban Jungle. The title 'Tentang Urban Jungle' is centered at the top. Below it is a paragraph about the company's history and mission: 'Urban Jungle adalah toko bunga premium yang berdiri sejak 2020, hadir untuk memberikan pengalaman berbeda dalam memilih dan mengirimkan bunga segar berkualitas tinggi.' Another paragraph states: 'Kami percaya bahwa setiap momen spesial layak dirayakan dengan keindahan bunga. Dari anniversary, ulang tahun, hingga ucapan selamat atas pencapaian, kami siap menghadirkan rangkaian bunga yang sempurna untuk Anda.' There are two sections below: 'Visi Kami' (Vision) with the text 'Menjadi toko bunga terdepan di Indonesia yang menghadirkan inovasi dalam desain bouquet dengan tetap mempertahankan kualitas dan kesegaran bunga.' and 'Misi Kami' (Mission) with the text 'Menyediakan bunga dengan kualitas premium' followed by a small circular profile picture of a woman.

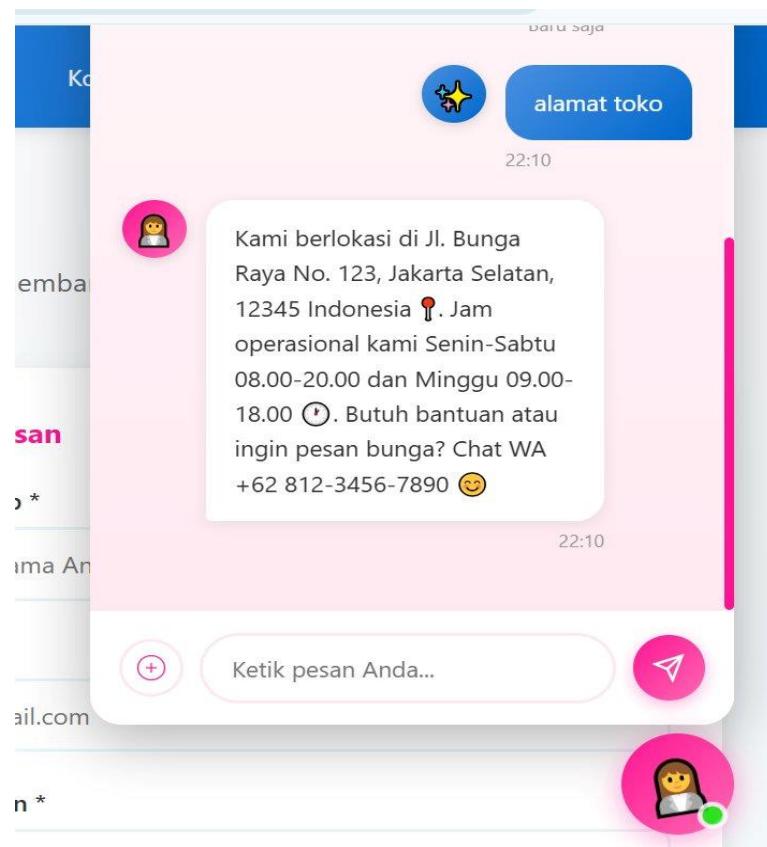
Gambar C.4 Halaman Tentang

C.5 Modal Keranjang Belanja



Gambar C.5 Keranjang Belanja

C.6 Interface Chatbot AI



Gambar C.6 Jendela Chatbot

C.7 Halaman Kontak

Hubungi Kami

Ada pertanyaan atau ingin custom bouquet? Kami siap membantu Anda!

Informasi Kontak

Email
info@urbanjungle.com

WhatsApp
+62 812-3456-7890

Telepon
(021) 1234-5678

Kirim Pesan

Nama Lengkap *

Email *

Nomor Telepon *

Gambar C.7 Halaman Kontak dengan Formulir dan Informasi Kontak

Gambar C.7 Halaman Kontak