

## **LAPORAN PROYEK III**

# **PENGEMBANGAN GOPOS (GO-LANG POS ASSISTANT) SEBAGAI CHATBOT CERDAS BERBASIS GEMINI AI PADA PT POS INDONESIA**

Diajukan untuk Memenuhi Kelulusan Matakuliah Proyek III  
pada Program Studi DIV Teknik Informatika

**Pembimbing:**

Rolly Maulana Awangga, S.T., MT., CAIP, SFPC.  
NIK. 117.86.219

**Disusun oleh:**

Muhammad Haitsam Izzuddin Azman (714230021)



**PROGRAM STUDI DIV TEKNIK INFORMATIKA  
UNIVERSITAS LOGISTIK & BISNIS INTERNASIONAL  
BANDUNG  
2025**

# **DAFTAR ISI**

<b>DAFTAR ISI</b>	<b>i</b>
<b>DAFTAR TABEL</b>	<b>iii</b>
<b>DAFTAR GAMBAR</b>	<b>iv</b>
<b>ABSTRAK</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vi</b>
<b>BAB I PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	2
1.3 Tujuan . . . . .	2
1.4 Ruang Lingkup . . . . .	2
1.5 Sistematika Penulisan . . . . .	3
<b>BAB II LANDASAN TEORI</b>	<b>4</b>
2.1 Tinjauan Pustaka . . . . .	4
2.1.1 Efisiensi Bahasa Pemrograman Golang (Go) . . . . .	4
2.1.2 Paradigma Serverless pada Google Cloud Platform . . . . .	4
2.1.3 Large Language Model (LLM) dan Google Gemini . . . . .	5
2.1.4 Fleksibilitas Basis Data MongoDB (NoSQL) . . . . .	5
2.1.5 Mekanisme Webhook pada WhatsApp Gateway . . . . .	5
2.2 Diagram Konseptual . . . . .	6
<b>BAB III METODOLOGI PENELITIAN</b>	<b>7</b>
3.1 Metodologi Pengembangan Sistem . . . . .	7
3.2 Analisis Kebutuhan Sistem . . . . .	8
3.2.1 Kebutuhan Perangkat Lunak (Software Requirements) . . . . .	8
3.2.2 Kebutuhan Fungsional . . . . .	8
3.3 Perancangan Sistem . . . . .	9
3.3.1 Arsitektur Sistem Global . . . . .	9
3.3.2 Perancangan Alur Logika (Activity Diagram) . . . . .	9
3.3.3 Perancangan Interaksi (Sequence Diagram) . . . . .	11
3.3.4 Use Case Diagram . . . . .	11
3.3.5 Class Diagram . . . . .	12
3.3.6 Statechart Diagram . . . . .	12

3.3.7	Component Diagram . . . . .	12
3.3.8	Perancangan Skema Database (NoSQL) . . . . .	12
3.3.9	Perancangan Antarmuka (User Interface) . . . . .	14
<b>BAB IV HASIL DAN PEMBAHASAN</b>		<b>16</b>
4.1	Pembahasan Hasil Implementasi . . . . .	16
4.2	Tampilan Antarmuka . . . . .	16
4.2.1	Halaman Beranda dan Autentikasi (index.html) . . . . .	16
4.2.2	Halaman Dashboard Admin (dashboard.html) . . . . .	17
4.2.3	Halaman Antarmuka Chat AI (chat.html) . . . . .	18
4.3	Pengujian (White Box) . . . . .	18
4.3.1	Analisis Hasil Pengujian . . . . .	19
<b>BAB V PENUTUP</b>		<b>21</b>
5.1	Kesimpulan . . . . .	21
5.2	Saran . . . . .	21
<b>DAFTAR PUSTAKA</b>		<b>23</b>
<b>LAMPIRAN</b>		<b>24</b>
Lampiran A: Code Coverage . . . . .	24	
<b>GLOSARIUM</b>		<b>25</b>
Istilah Teknis . . . . .	25	
Istilah Non-Teknis . . . . .	26	

## **DAFTAR TABEL**

1	Hasil Code Coverage . . . . .	19
---	-------------------------------	----

## **DAFTAR GAMBAR**

1	Diagram Konseptual Webhook . . . . .	6
2	Metode Agile . . . . .	7
3	Arsitektur Sistem Global . . . . .	9
4	Activity Diagram . . . . .	10
5	Sequence Diagram . . . . .	11
6	Use Case Diagram . . . . .	11
7	Class Diagram . . . . .	12
8	Statechart Diagram . . . . .	13
9	Component Diagram . . . . .	14
10	Tampilan index.html . . . . .	17
11	Tampilan dashboard.html . . . . .	17
12	Tampilan chat.html . . . . .	18

## ABSTRAK

Dalam era transformasi digital saat ini, pelanggan, terutama mereka yang bekerja di bidang logistik dan pengiriman, sangat membutuhkan akses data yang cepat dan mudah. Namun, keterbatasan layanan pelanggan konvensional, seperti *call center* atau loket fisik, seringkali menyebabkan masalah untuk mendapatkan informasi di luar jam kerja. Pelanggan mungkin kesulitan mendapatkan estimasi biaya kirim atau tahu di mana kantor pos terdekat. Selain itu, orang percaya bahwa mencari informasi secara manual melalui website dinilai kurang praktis untuk pengguna perangkat ponsel. Mengembangkan “GOPOS”, sebuah asisten virtual cerdas (*chatbot*) berbasis WhatsApp, adalah tujuan proyek untuk mengatasi masalah ini.

Arsitektur teknologi *serverless* Google Cloud Platform (GCP) digunakan untuk mengembangkan sistem ini. Karena kecepatan kompetitif dan kemampuan menangani webhook yang luar biasa, bahasa pemrograman Golang digunakan untuk membangun logika backend. Sistem menggunakan basis data NoSQL MongoDB untuk memberikan fleksibilitas dalam penyimpanan data sesi percakapan. Salah satu fitur unggulan aplikasi ini adalah model bahasa besar Gemini AI, yang memungkinkan *chatbot* memahami bahasa alami pengguna dengan mudah daripada bot biasa. *Chatbot* ini menjawab pertanyaan umum tentang layanan Pos Indonesia dan mengestimasikan tarif pengiriman secara dinamis.

Metodologi Agile digunakan dalam pengembangan perangkat lunak, yang memungkinkan penyesuaian berulang untuk logika bisnis dan respons AI. Untuk memastikan bahwa respons *chatbot* sesuai dengan pertanyaan, metode *White Box Testing* digunakan untuk menguji sistem. Hasil implementasi menunjukkan bahwa GOPOS dapat menawarkan layanan pelanggan yang responsif, akurat, dan tersedia 24/7 melalui integrasi kecerdasan buatan dengan layanan pesan instan WhatsApp.

**Kata Kunci:** *Chatbot*, WhatsApp Gateway, Gemini AI, Golang, *Serverless*, Google Cloud Platform, MongoDB.

## ABSTRACT

In today's era of digital transformation, customers, especially those working in logistics and shipping, desperately need fast and easy access to data. However, the limitations of conventional customer services, such as call centers or physical counters, often cause problems when trying to obtain information outside of business hours. Customers may find it difficult to get shipping cost estimates or find out where the nearest post office is. Additionally, people believe that manually searching for information through websites is impractical for mobile phones. Developing "GO-POS," an intelligent virtual assistant (chatbot) based on WhatsApp, is the project's goal to address these issues.

The Google Cloud Platform (GCP) serverless technology architecture was used to develop this system. Due to its competitive speed and exceptional webhook handling capabilities, the Golang programming language was used to build the backend logic. The system uses the NoSQL MongoDB database to provide flexibility in storing conversation session data. One of the standout features of this application is the Gemini AI large language model, which enables the chatbot to understand users' natural language more easily than regular bots. This chatbot answers general questions about Pos Indonesia's services and dynamically estimates shipping rates.

The Agile methodology is used in software development, which allows for iterative adjustments to business logic and AI responses. To ensure that chatbot responses are appropriate to questions, the White Box Testing method is used to test the system. The results of the implementation show that GOPOS can offer responsive, accurate, and 24/7 customer service through the integration of artificial intelligence with the WhatsApp instant messaging service.

**Keywords:** Chatbot, WhatsApp Gateway, Gemini AI, Golang, Serverless, Google Cloud Platform, MongoDB.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sektor logistik dan pengiriman merupakan bagian vital dalam mendukung pertumbuhan ekonomi dan koneksi masyarakat di era digital. Sebagai salah satu penyedia layanan logistik terbesar di Indonesia, PT Pos Indonesia terus berupaya melakukan transformasi digital untuk meningkatkan kualitas layanan pelanggan. Namun, saat ini masyarakat seringkali menghadapi tantangan dalam mengakses informasi layanan secara cepat, seperti pencarian estimasi ongkos kirim (ongkir) atau pencarian lokasi kantor pos yang mengharuskan navigasi manual melalui situs web yang terkadang dinilai kurang praktis bagi pengguna perangkat seluler.

Di sisi lain, perkembangan teknologi *Generative AI* membuka peluang untuk menciptakan interaksi yang lebih personal dan responsif melalui antarmuka percakapan (*chatbot*). Penggunaan aplikasi pesan instan seperti WhatsApp sebagai platform utama komunikasi masyarakat dapat dimanfaatkan sebagai gerbang akses layanan informasi yang tersedia 24/7 tanpa terkendala jam operasional kantor fisik.

Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah platform asisten virtual cerdas yang mampu memahami bahasa alami pengguna dan memberikan informasi layanan pos secara instan. Dalam proyek ini, dikembangkan sebuah sistem asisten virtual bernama “GOPOS” (*Go-Lang Pos Assistant*). Sistem ini dirancang dengan arsitektur teknologi modern yang menggabungkan performa tinggi dan kecerdasan buatan, antara lain:

- **Golang (Go)** sebagai backend utama yang berjalan di lingkungan *Serverless* Google Cloud Functions, dipilih karena efisiensinya dalam menangani webhook dan permintaan API secara cepat.
- **Gemini AI** sebagai mesin kecerdasan buatan yang memungkinkan sistem memahami maksud pertanyaan pengguna (NLP) secara luwes, sehingga interaksi tidak terasa kaku.
- **WhatsApp Gateway** sebagai antarmuka pengguna, memungkinkan akses layanan melalui aplikasi pesan yang sudah familiar bagi masyarakat.
- **MongoDB** sebagai basis data NoSQL untuk menyimpan sesi percakapan dan log aktivitas secara fleksibel.

Fitur unggulan dari GOPOS adalah kemampuannya dalam memberikan estimasi

tarif pengiriman secara otomatis melalui logika dinamis pada backend serta menjawab berbagai pertanyaan umum seputar layanan pos melalui integrasi Gemini AI. Proyek ini bertujuan untuk mengintegrasikan teknologi cloud, kecerdasan buatan, dan layanan pesan instan dalam satu ekosistem asisten virtual yang utuh.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam proyek ini adalah:

1. Bagaimana merancang dan membangun asisten virtual GOPOS berbasis *chatbot* WhatsApp yang responsif menggunakan bahasa pemrograman Golang?
2. Bagaimana mengimplementasikan kecerdasan buatan menggunakan Gemini AI untuk menangani pertanyaan pengguna terkait layanan logistik secara natural?
3. Bagaimana melakukan integrasi sistem antara layanan backend Golang pada Google Cloud Platform, basis data MongoDB, dan WhatsApp Gateway?

## 1.3 Tujuan

Tujuan dari pelaksanaan proyek ini adalah:

1. Mengembangkan asisten virtual GOPOS sebagai platform informasi layanan Pos Indonesia yang *user-friendly* melalui media WhatsApp.
2. Membangun fitur cek ongkir otomatis dan tanya jawab cerdas menggunakan Gemini AI untuk membantu pengguna menemukan informasi secara cepat.
3. Mengimplementasikan integrasi teknologi Golang, MongoDB, dan infrastruktur cloud untuk mendukung skalabilitas dan ketersediaan layanan.

## 1.4 Ruang Lingkup

Ruang lingkup dalam pengembangan sistem ini meliputi:

- **Fokus Pengembangan:** Asisten virtual berbasis teks (WhatsApp) dengan dukungan kecerdasan buatan untuk layanan logistik.
- **Layanan Backend:** Menggunakan bahasa pemrograman Golang yang di-deploy melalui Google Cloud Functions.
- **Kecerdasan Buatan:** Pemanfaatan Gemini AI untuk pemrosesan bahasa alami (NLP) pada percakapan umum.

- **Pengelolaan Data:** Penyimpanan data sesi dan konfigurasi sistem dilakukan menggunakan basis data NoSQL MongoDB.
- **Fitur Utama:** Mencakup pengecekan estimasi ongkos kirim domestik, pencarian informasi layanan, dan interaksi percakapan natural.
- **Pengguna:** Target pengguna adalah masyarakat umum yang memerlukan informasi layanan PT Pos Indonesia secara instan.

## 1.5 Sistematika Penulisan

Laporan penelitian ini terdiri dari beberapa bab dengan sistematika sebagai berikut:

- **BAB I Pendahuluan:** Menguraikan latar belakang, rumusan masalah, tujuan penelitian, ruang lingkup dan sistematika penulisan.
- **BAB II Landasan Teori:** Membahas teori penunjang seperti konsep *Serverless Architecture*, Golang, *Generative AI* (Gemini), MongoDB, dan WhatsApp Gateway.
- **BAB III Metodologi Penelitian:** Menjelaskan metodologi pengembangan (Agi-le), analisis kebutuhan sistem, perancangan arsitektur cloud, dan perancangan alur percakapan *chatbot*.
- **BAB IV Hasil dan Pembahasan:** Menampilkan hasil implementasi kode program, integrasi API, serta analisis terhadap performa *chatbot* dalam merespons pengguna.
- **BAB V Penutup:** Berisi kesimpulan dari pengembangan asisten virtual GOPOS serta saran untuk pengembangan fitur lebih lanjut.

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Pustaka

Bab ini menguraikan kerangka teori dan konsep teknologi yang menjadi fondasi dalam pengembangan asisten virtual “GOPOS”. Pemahaman mendalam mengenai teknologi-teknologi ini sangat esensial untuk menjelaskan bagaimana sistem dapat beroperasi secara responsif, mulai dari manajemen data yang tidak terstruktur, mekanisme komunikasi antar-server, hingga penerapan kecerdasan buatan generatif dalam memproses bahasa manusia.

##### 2.1.1 Efisiensi Bahasa Pemrograman Golang (Go)

Dalam pengembangan sistem backend modern yang menuntut kecepatan tinggi, pemilihan bahasa pemrograman menjadi faktor krusial. Go, atau sering disebut Golang, adalah bahasa pemrograman kompilasi (*compiled language*) yang dikembangkan oleh Google. Keunggulan utama Go terletak pada efisiensi memori dan kemampuannya menangani proses konkurensi (banyak tugas dalam satu waktu) melalui fitur *goroutines*. Dalam konteks aplikasi GOPOS, Go dipilih bukan hanya karena kecepatannya, tetapi karena ketegasan tipe datanya (*static typing*) yang meminimalisir kesalahan logika saat kompilasi. Hal ini sangat relevan ketika sistem harus memproses validasi struktur data JSON yang kompleks dari pesan WhatsApp secara *real-time* sebelum meneruskannya ke layanan lain (Dimara, 2024).

##### 2.1.2 Paradigma Serverless pada Google Cloud Platform

Pergeseran dari arsitektur server tradisional (monolitik) menuju arsitektur cloud modern telah melahirkan konsep *Serverless Computing*. Dalam proyek ini, teknologi yang diadopsi adalah Google Cloud Functions (GCF) sebagai bagian dari ekosistem Google Cloud Platform. Secara teoritis, *Serverless* atau *Function-as-a-Service* (FaaS) memungkinkan pengembang untuk mengunggah kode fungsi tunggal tanpa perlu memikirkan penyediaan atau pemeliharaan server fisik. Keuntungan signifikannya adalah skalabilitas otomatis; sistem akan secara cerdas menyesuaikan sumber daya komputasi berdasarkan volume pesan yang masuk. Jika tidak ada interaksi dari pengguna, sistem akan “tidur” sehingga tidak membebankan biaya operasional, menjadikannya solusi yang sangat efisien untuk aplikasi berbasis event seperti *chatbot* (P. R. Journal, 2015).

### 2.1.3 Large Language Model (LLM) dan Google Gemini

Berbeda dengan pendekatan sistem pakar terdahulu yang mengandalkan aturan kaku (*rule-based*), perkembangan kecerdasan buatan kini telah mencapai tahap Generatif. Google Gemini merupakan salah satu model bahasa besar (*Large Language Model*) mutakhir yang memiliki kemampuan multimodal. Teori dasar di balik teknologi ini adalah *Natural Language Processing* (NLP), di mana mesin dilatih dengan dataset masif untuk memahami, memprediksi, dan menghasilkan teks yang menyerupai gaya bicara manusia. Dalam implementasi GOPOS, Gemini tidak sekadar mencocokkan kata kunci, melainkan memahami konteks percakapan. Hal ini memungkinkan sistem untuk memberikan jawaban yang luwes dan relevan terhadap pertanyaan pengguna yang bervariasi, meskipun pertanyaan tersebut belum pernah diprogram sebelumnya secara spesifik (Rachmat & Kesuma, 2024).

### 2.1.4 Fleksibilitas Basis Data MongoDB (NoSQL)

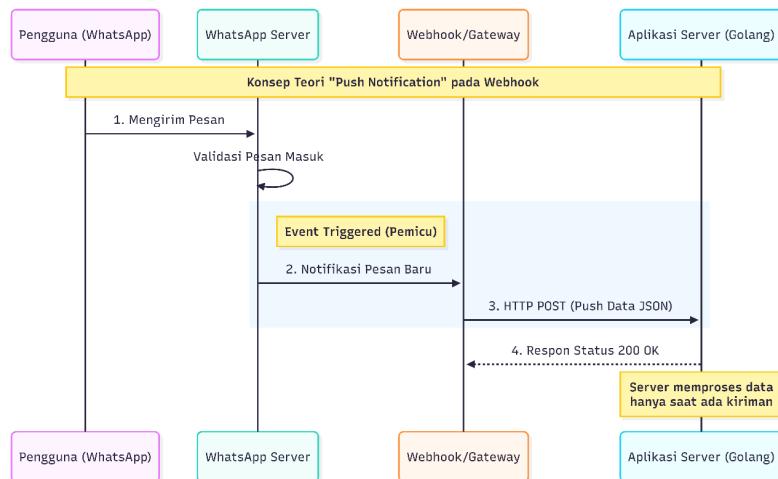
Sistem aplikasi percakapan menghasilkan data yang sangat dinamis dan sering kali tidak memiliki struktur yang pasti. Oleh karena itu, penggunaan basis data relasional (RDBMS) seperti MySQL terkadang kurang efisien karena skemanya yang kaku. Sebagai alternatif, proyek ini menerapkan MongoDB, sebuah basis data NoSQL berbasis dokumen. MongoDB menyimpan data dalam format BSON (*Binary JSON*), yang selaras dengan format pertukaran data pada aplikasi web modern. Fleksibilitas ini memungkinkan GOPOS untuk menyimpan riwayat percakapan (*chat logs*) yang panjangnya bervariasi, serta menyimpan metadata sesi pengguna tanpa perlu melakukan perubahan struktur tabel yang rumit. Hal ini mempercepat proses pengembangan dan adaptasi fitur baru di masa mendatang (Barman, 2025).

### 2.1.5 Mekanisme Webhook pada WhatsApp Gateway

Komunikasi antara pengguna WhatsApp dan server aplikasi terjadi melalui perantara yang disebut Gateway. Teknologi kunci yang digunakan di sini adalah Webhook. Secara konsep, Webhook adalah metode komunikasi “reaktif”. Berbeda dengan metode *polling* di mana server harus terus-menerus bertanya “apakah ada pesan baru?”, Webhook bekerja dengan cara mengirimkan notifikasi data (HTTP POST) secara otomatis ke URL server tujuan segera setelah peristiwa terjadi (pesan diterima). Dalam sistem GOPOS, mekanisme ini memastikan bahwa setiap pesan yang dikirim pengguna dapat diterima dan diproses oleh sistem backend dengan latensi yang sangat rendah, menciptakan pengalaman chat yang responsif (Aidyn, 2024).

## 2.2 Diagram Konseptual

Konsep Dasar Komunikasi Webhook Diagram ini menggambarkan bagaimana teori Webhook bekerja dalam menghantarkan pesan secara *real-time*.



**Gambar 1:** Diagram Konseptual Webhook

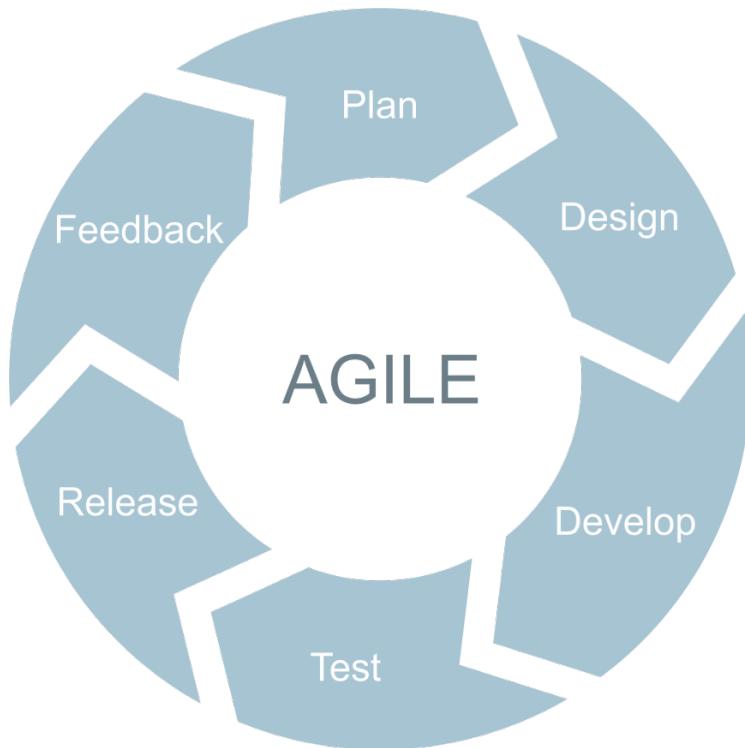
Penjelasan Singkat Diagram: Diagram di atas mengilustrasikan landasan teori komunikasi *event driven*. Server aplikasi tidak perlu aktif memantau (*polling*), melainkan hanya menunggu kiriman data (*push*) dari Webhook saat terjadi pemicu berupa pesan masuk dari pengguna.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Metodologi Pengembangan Sistem

Dalam membangun sistem asisten virtual GOPOS, metode pengembangan perangkat lunak yang diterapkan adalah *Agile Development* dengan pendekatan Scrum. Pemilihan metode ini didasari oleh karakteristik proyek berbasis kecerdasan buatan (*Artificial Intelligence*) yang dinamis, di mana akurasi respons AI dan logika bisnis seringkali membutuhkan penyesuaian berulang (iterasi) berdasarkan hasil uji coba.



Gambar 2: Metode Agile

Tahapan pengembangan yang dilakukan meliputi:

1. **Perencanaan (Planning):** Mengidentifikasi masalah pada layanan pelanggan Pos Indonesia dan menentukan fitur prioritas (Cek Ongkir & Chatbot AI).
2. **Analisis & Desain (Analysis & Design):** Merancang arsitektur *serverless* pada Google Cloud dan skema data NoSQL.
3. **Implementasi (Development):** Pengkodean sistem backend menggunakan Go-lang dan integrasi API Gemini.
4. **Pengujian (Testing):** Melakukan *Unit Testing* pada fungsi logika dan *User*

*Acceptance Testing* (UAT) melalui WhatsApp.

5. **Penyebaran (Deployment):** Mengunggah kode ke Google Cloud Functions melalui GitHub Actions.

### 3.2 Analisis Kebutuhan Sistem

Tahap analisis kebutuhan bertujuan untuk memetakan spesifikasi teknis dan fungsional agar sistem dapat berjalan optimal sesuai tujuan penelitian.

#### 3.2.1 Kebutuhan Perangkat Lunak (Software Requirements)

Lingkungan pengembangan sistem ini membutuhkan dukungan perangkat lunak sebagai berikut:

- **Bahasa Pemrograman:** Go (Golang) versi 1.2x ke atas.
- **Cloud Provider:** Google Cloud Platform (Cloud Functions, Cloud Build).
- **Database:** MongoDB Atlas (Cloud NoSQL).
- **AI Service:** Google Gemini Pro API.
- **Messaging Gateway:** WhatsAuth / WhatsApp API.
- **Editor Kode:** Visual Studio Code.

#### 3.2.2 Kebutuhan Fungsional

Berdasarkan analisis alur kerja, sistem GOPOS memiliki spesifikasi fungsional utama:

1. **Penerimaan Pesan Otomatis:** Sistem harus mampu menerima data pesan (*payload*) dari WhatsApp secara *real-time* melalui metode Webhook.
2. **Identifikasi Maksud (Intent Detection):** Sistem mampu membedakan jenis pesan pengguna, apakah berupa permintaan cek tarif (transaksional) atau pertanyaan umum (konvensional).
3. **Kalkulasi Tarif Dinamis:** Sistem dapat mengekstrak entitas kota asal, kota tujuan, dan berat barang untuk menghitung estimasi biaya pengiriman.
4. **Respon Cerdas Berbasis AI:** Sistem terintegrasi dengan *Large Language Model* (Gemini) untuk menjawab pertanyaan di luar konteks tarif dengan gaya bahasa natural.

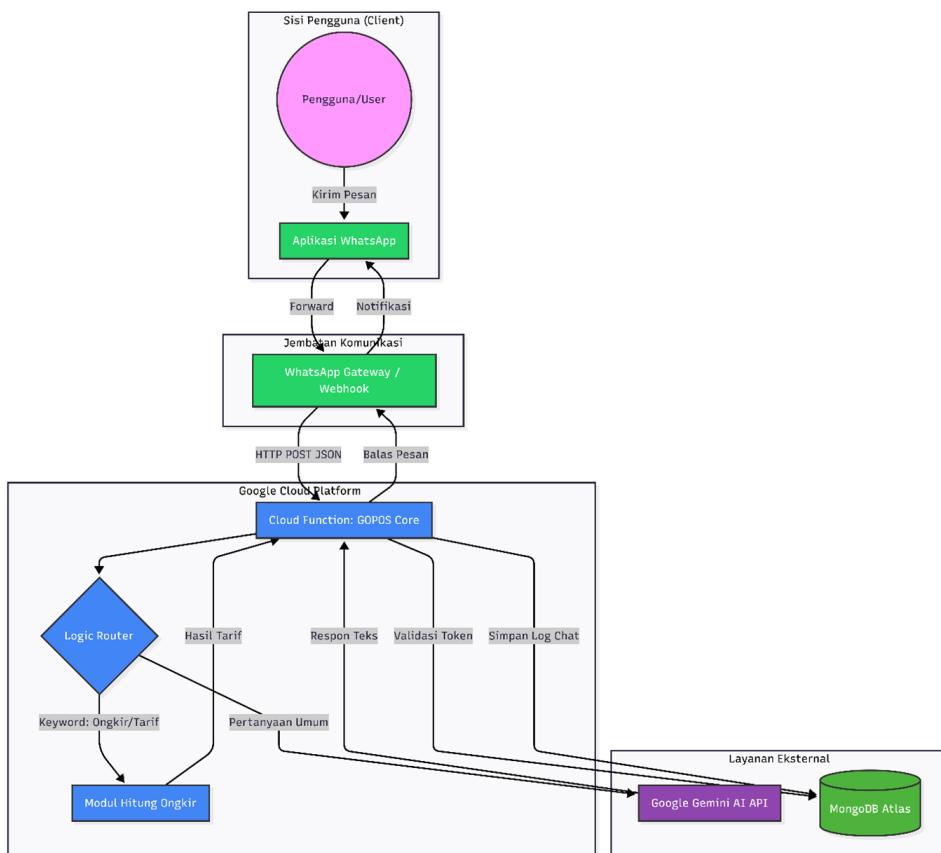
5. **Manajemen Riwayat Percakapan:** Sistem menyimpan log percakapan ke dalam basis data untuk menjaga konteks interaksi.

### 3.3 Perancangan Sistem

Perancangan sistem merupakan cetak biru (*blueprint*) teknis yang menggambarkan bagaimana komponen-komponen aplikasi saling berinteraksi.

#### 3.3.1 Arsitektur Sistem Global

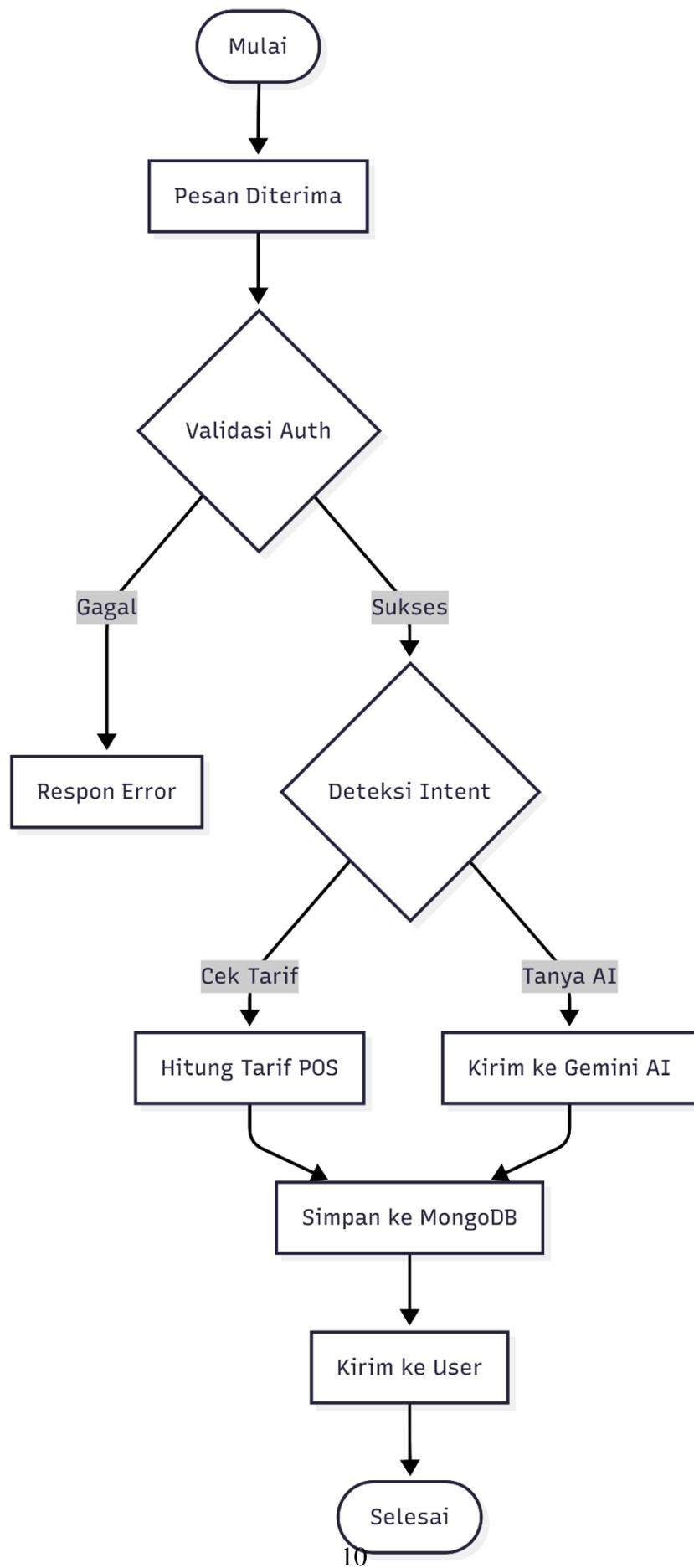
GOPOS menerapkan arsitektur *Serverless Microservices*. Diagram berikut mengilustrasikan ekosistem menyeluruh mulai dari sisi pengguna (WhatsApp) hingga pemrosesan di cloud backend.



Gambar 3: Arsitektur Sistem Global

#### 3.3.2 Perancangan Alur Logika (Activity Diagram)

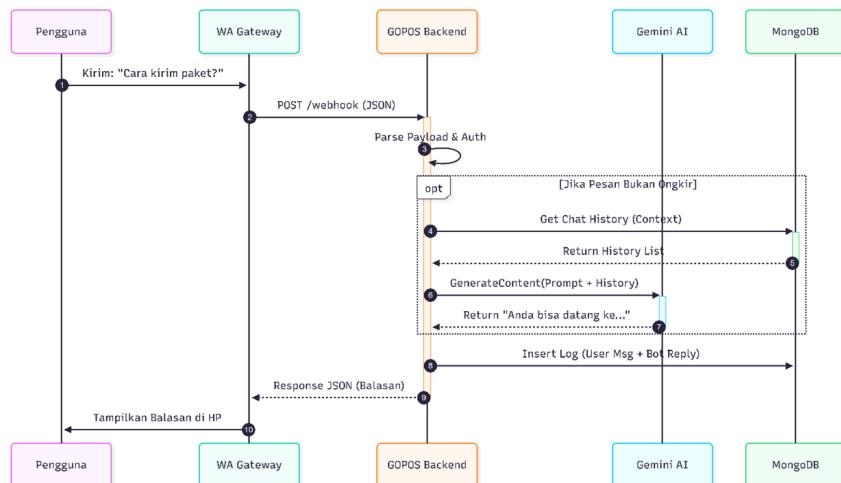
Diagram alir berikut mendetailkan algoritma pengambilan keputusan (*decision making*) saat sebuah pesan masuk. Sistem menggunakan pendekatan *hybrid*, menggabungkan logika deterministik (untuk hitungan pasti) dan probabilistik (AI).



Gambar 4: Activity Diagram

### 3.3.3 Perancangan Interaksi (Sequence Diagram)

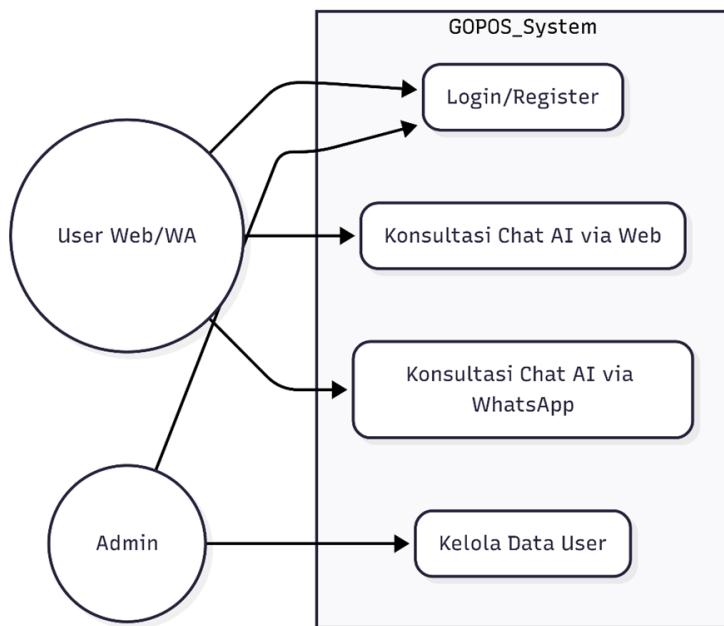
Diagram sekuensial menggambarkan urutan waktu pertukaran pesan antar objek dalam sistem. Diagram ini menunjukkan bagaimana latensi diminimalisir dengan pemrosesan paralel antara penyimpanan database dan pembuatan respon AI.



**Gambar 5:** Sequence Diagram

### 3.3.4 Use Case Diagram

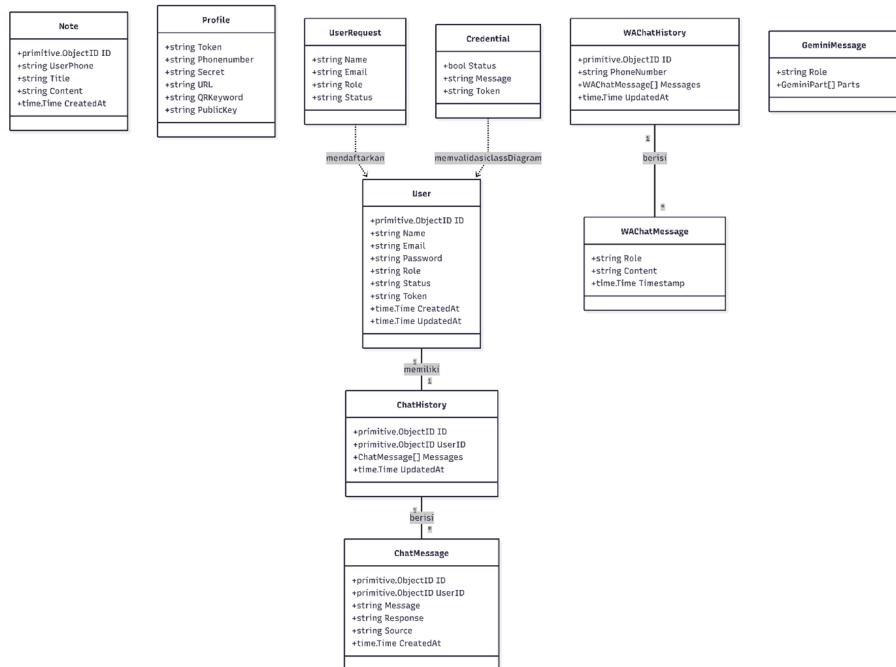
Menggambarkan interaksi aktor dengan fungsionalitas sistem GOPOS.



**Gambar 6:** Use Case Diagram

### 3.3.5 Class Diagram

Menggambarkan seluruh struktur data yang diimplementasikan pada kode Go.



**Gambar 7:** Class Diagram

### 3.3.6 Statechart Diagram

Siklus hidup pesan di dalam server GOPOS.

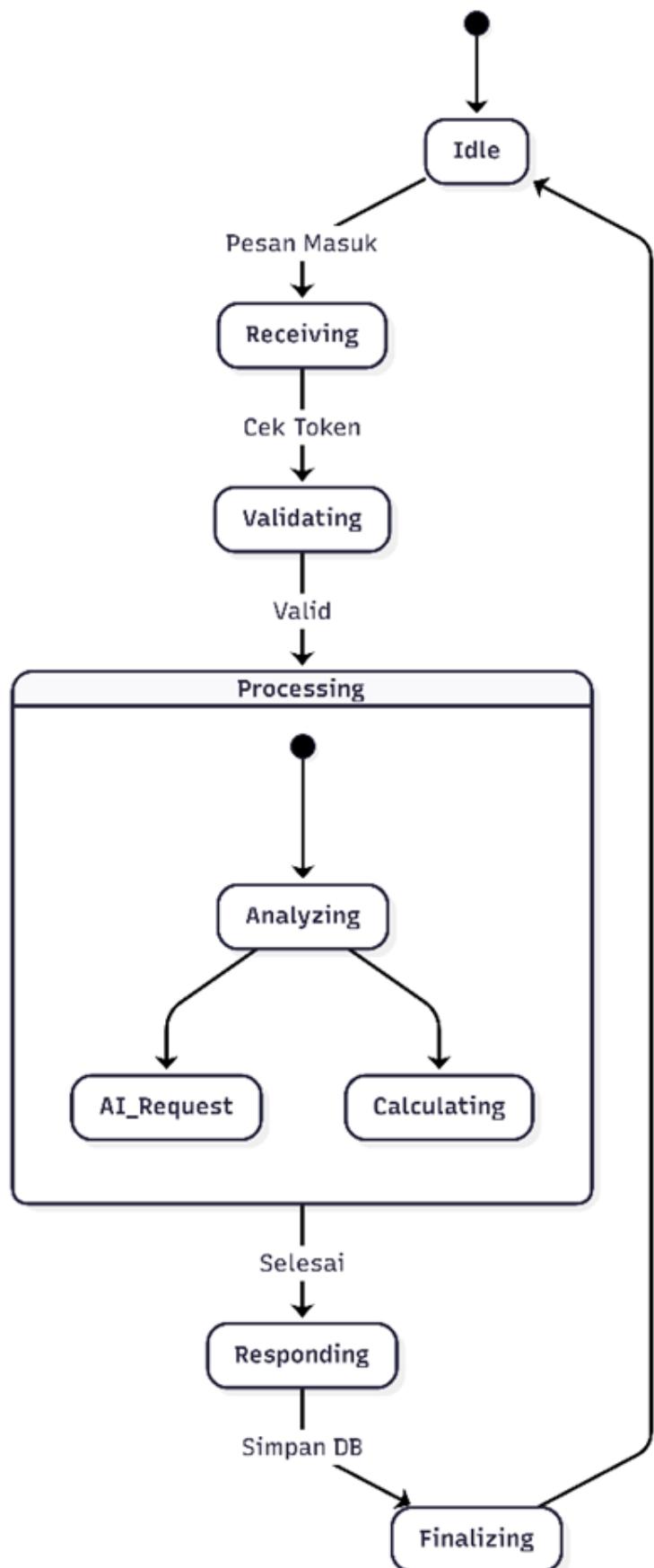
### 3.3.7 Component Diagram

Hubungan antar komponen arsitektur sistem.

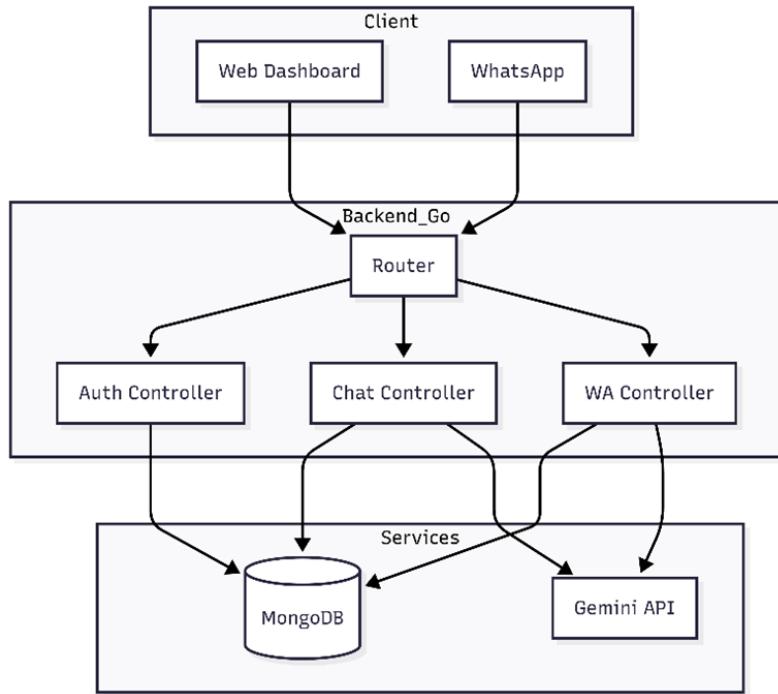
### 3.3.8 Perancangan Skema Database (NoSQL)

Sistem ini menggunakan MongoDB sebagai basis data NoSQL untuk menyimpan data secara fleksibel dalam format dokumen (BSON). Pemilihan NoSQL bertujuan untuk mempermudah penyimpanan riwayat percakapan yang memiliki panjang data tidak tetap. Berdasarkan analisis model data pada backend, koleksi-koleksi utama (*struct*) yang dirancang adalah sebagai berikut:

**Koleksi Users:** Berfungsi menyimpan profil dan kredensial autentikasi pengguna *dashboard*. Field Utama: `name`, `email`, `password` (disimpan dalam bentuk hash), `role` (admin/user), dan `status` (active/inactive).



Gambar 8: Statechart Diagram



**Gambar 9:** Component Diagram

**Koleksi ChatHistories:** Berfungsi menyimpan riwayat interaksi pengguna aplikasi web dengan AI. Struktur: Menggunakan konsep *embedded documents* di mana setiap dokumen menyematkan array *messages* yang berisi detail pesan (*message*), balasan AI (*response*), dan sumber model (*source*).

**Koleksi WaChatHistories:** Menyimpan konteks percakapan khusus untuk pengguna yang berinteraksi via WhatsApp. Field Utama: *phone\_number* sebagai pengenal unik dan array *messages* yang berisi *role* (user/model) serta *content* pesan.

**Koleksi Notes:** Berfungsi menyimpan catatan singkat (Notes) yang dikirimkan pengguna melalui bot. Field Utama: *user\_phone*, *title*, dan *content*.

**Koleksi Profiles:** Menyimpan konfigurasi teknis integrasi seperti API Token, Secret, dan URL webhook untuk menghubungkan sistem dengan penyedia layanan WhatsApp.

### 3.3.9 Perancangan Antarmuka (User Interface)

Perancangan antarmuka (UI) GOPOS menitikberatkan pada aspek fungsionalitas dan kemudahan akses melalui berbagai perangkat (responsif). Berdasarkan struktur frontend yang dikembangkan, antarmuka dibagi menjadi tiga halaman utama:

**Halaman Autentikasi (index.html):** Dirancang minimalis dengan fokus pada formulir login. Pengguna memasukkan email dan kata sandi untuk diverifikasi oleh backend sebelum mendapatkan token sesi.

**Halaman Dashboard Utama (dashboard.html):** Berfungsi sebagai pusat kendali pengguna setelah berhasil masuk. Menampilkan informasi profil pengguna dan ringkasan aktivitas. Sesuai ralat sistem, manajemen profil dan hak akses kini diintegrasikan langsung pada tampilan *dashboard* ini untuk menyederhanakan navigasi.

**Halaman Chat AI (chat.html):** Merupakan antarmuka interaksi utama asisten virtual. Komponen:

- **Panel Samping (Sidebar):** Menampilkan daftar riwayat percakapan sebelumnya yang diambil dari koleksi `chathistories`.
- **Jendela Chat:** Menampilkan gelembung pesan (*chat bubbles*) secara berurutan sesuai alur waktu percakapan.
- **Input Area:** Kolom teks di bagian bawah untuk mengirimkan pertanyaan baru kepada model Gemini AI.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Pembahasan Hasil Implementasi**

Implementasi sistem “GOPOS (*Go-Lang POS Assistant*)” sebagai *chatbot* cerdas telah berhasil memenuhi kebutuhan pengguna dalam mengakses informasi layanan Pos Indonesia secara efektif dan efisien. Dengan fitur utama seperti asisten virtual berbasis AI serta fitur cek tarif pengiriman (Cek Ongkir), sistem ini mampu berjalan sesuai dengan perancangan yang telah ditetapkan. Pengguna dapat berinteraksi dengan sistem secara optimal melalui antarmuka web yang responsif serta mudah dipahami.

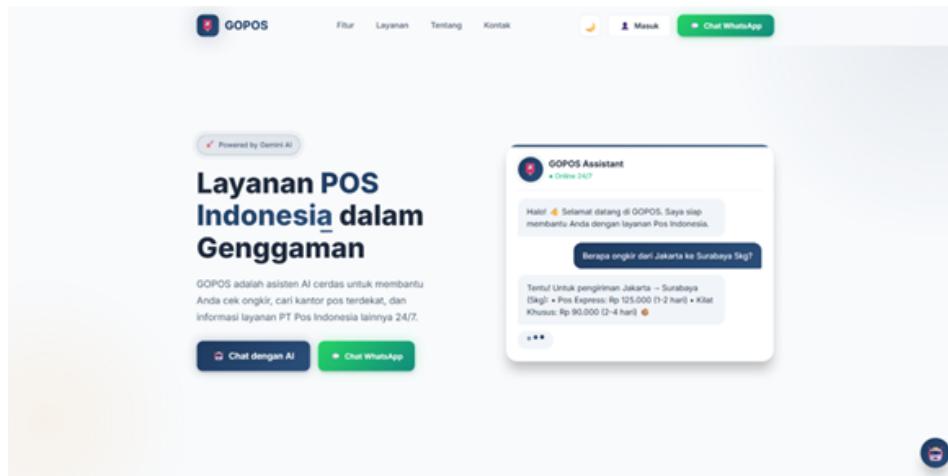
Melengkapi fungsionalitas utama tersebut, pengembangan sistem juga mencakup implementasi integrasi WhatsApp Gateway dan manajemen catatan (Notes). Berdasarkan hasil pengujian, integrasi menggunakan metode Webhook terbukti stabil dalam memproses data pesan secara *real-time*, sementara penggunaan model Gemini Pro API memberikan nilai tambah signifikan dengan menyajikan jawaban yang natural dan kontekstual terhadap berbagai pertanyaan pengguna di luar layanan transaksional.

#### **4.2 Tampilan Antarmuka**

Perancangan antarmuka pengguna pada sistem GOPOS mengedepankan prinsip desain yang bersih, intuitif, dan responsif guna memastikan kenyamanan interaksi baik melalui perangkat desktop maupun seluler. Implementasi antarmuka ini dibagi menjadi beberapa bagian utama yang mewakili alur kerja pengguna dalam sistem.

##### **4.2.1 Halaman Beranda dan Autentikasi (index.html)**

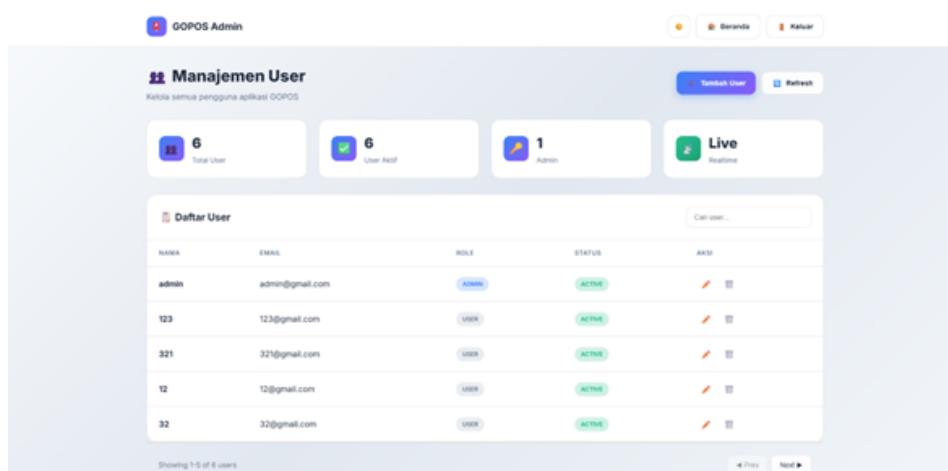
Halaman beranda atau `index.html` berfungsi sebagai pusat informasi publik dan pintu masuk utama ke dalam ekosistem GOPOS. Halaman ini memiliki desain visual yang dinamis dengan animasi latar belakang, dan memiliki bagian *Hero Section* yang menunjukkan nilai utama aplikasi sebagai asisten bertenaga AI. Bagian informasi di halaman ini mencakup statistik penggunaan *real-time*, daftar fitur unggulan, seperti cek ongkir cepat, dan penjelajah. Seluruh skema warna dan elemen tombol telah dioptimalkan untuk memudahkan navigasi pengguna sebelum masuk ke fitur inti, memungkinkan pengguna berinteraksi dengan modal autentikasi yang terintegrasi secara elegan untuk melakukan proses login dan registrasi akun baru.



Gambar 10: Tampilan index.html

#### 4.2.2 Halaman Dashboard Admin (dashboard.html)

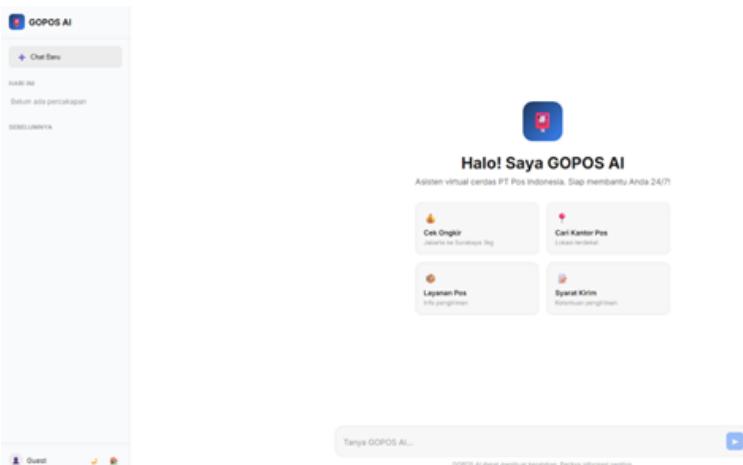
Halaman *dashboard* admin (dashboard.html) dibuat sebagai pusat manajemen data pengguna yang efektif dengan tata letak yang terorganisir untuk administrator sistem. Tampilan utama halaman terdiri dari panel kontrol yang menampilkan metrik statistik penting seperti jumlah pengguna total, pengguna aktif, dan status koneksi sistem, yang ditampilkan dalam bentuk kartu statistik kontemporer. Bagian utama halaman terdiri dari tabel data pengguna yang berfungsi, dilengkapi dengan fitur tambahan seperti daftar pengguna aktif. Dengan antarmuka yang bersih, administrator dapat secara langsung memantau status akun dan melakukan tugas administratif. Ini memastikan bahwa pengelolaan hak akses dan manajemen pengguna tetap tepat dan jelas.



Gambar 11: Tampilan dashboard.html

#### 4.2.3 Halaman Antarmuka Chat AI (chat.html)

Halaman antarmuka chat AI (`chat.html`) adalah tempat utama di mana pengguna web dapat berinteraksi secara langsung dengan asisten virtual GOPOS. Halaman ini memiliki struktur yang sangat fungsional, dengan panel riwayat percakapan di sisi kiri yang memungkinkan pengguna untuk beralih secara instan di antara sesi percakapan terdahulu. Di bagian tengah, jendela percakapan menyajikan pesan dalam bentuk gelembung chat yang dinamis. Layar selamat datang, juga disebut sebagai layar selamat datang, didukung oleh kartu saran pertanyaan otomatis yang membantu pengguna baru memulai percakapan. Integrasi API Gemini memastikan bahwa setiap teks yang dimasukkan ke dalam kolom pesan di bagian bawah akan menerima balasan cerdas dalam hitungan detik. Ini juga dilengkapi dengan indikator pengetikan, yang memberikan kesan interaksi yang lebih hidup dan responsif bagi pengguna.



**Gambar 12:** Tampilan chat.html

### 4.3 Pengujian (White Box)

Pengujian *White Box* dilakukan untuk memastikan struktur dan logika kode program di dalamnya. Pengujian ini mengukur cakupan kode pada sisi backend, yang dibangun menggunakan bahasa pemrograman GoLang. Tujuannya adalah untuk memastikan bahwa setiap fungsi kontroler telah dioperasikan dan diuji menggunakan skenario uji unit.

Berikut adalah hasil pengujian *Code Coverage* untuk modul Controllers:

**Tabel 1:** Hasil Code Coverage

No	Modul Controller	Coverage (%)	Test Functions
1	Chatbot Controller	93.2	20
2	Auth Controller	69.1	24
3	Chat History Controller	58.5	13
4	User Management Controller	29.1	22
5	WhatsAuth Controller	27.3	18
<b>Rata-rata</b>		<b>55.4</b>	<b>97 (Total)</b>

#### 4.3.1 Analisis Hasil Pengujian

Berdasarkan tabel di atas, persentase cakupan kode pada modul-modul inti seperti *Chatbot* dan *Authentication* menunjukkan hasil yang sangat memuaskan, dengan nilai mencapai lebih dari 60%. Angka ini mengindikasikan bahwa alur bisnis utama pada kedua modul tersebut telah melalui pengujian yang komprehensif.

Beberapa catatan penting dari hasil pengujian ini adalah:

- **Chatbot Controller (93.2%):** Persentase cakupan sangat optimal disebabkan modul *chatbot* memiliki sejumlah besar fungsi utilitas yang dapat diuji secara independen tanpa ketergantungan eksternal, meliputi ekstraksi data ongkir (asal, tujuan, berat), formatting numerik untuk display harga, klasifikasi zona pengiriman internasional, serta berbagai validasi input pengguna. Terdapat 20 *test functions* yang mencakup *helper functions*, *edge cases*, dan skenario integrasi.
- **Auth Controller (69.1%):** Tingkat cakupan tergolong baik untuk aspek validasi masukan dan penanganan kesalahan. Limitasi terdapat pada alur sukses (registrasi dan login berhasil) yang membutuhkan koneksi aktif ke database MongoDB guna melakukan operasi create/read user serta pembangkitan token autentikasi JWT. Modul ini diuji dengan 24 *test functions* yang mencakup validasi input, *error handling*, dan berbagai *edge cases*.
- **Chat History Controller (58.5%):** Cakupan meliputi validasi konten pesan, handling untuk JSON yang tidak valid, serta pemeriksaan authorization header. Bagian yang belum tercakup adalah proses penyimpanan dan pengambilan riwayat percakapan dari database MongoDB yang memerlukan koneksi database aktif. Terdapat 13 *test functions* yang fokus pada validasi message dan auth scenarios.
- **User Management Controller (29.1%):** Persentase berada di kategori cukup mengingat fungsionalitas CRUD manajemen pengguna memiliki dependensi tinggi terhadap operasi database, seperti validasi email duplicate, pencarian user

berdasarkan ID MongoDB ObjectId, dan verifikasi role admin yang mengharuskan query ke database. Diuji dengan 22 *test functions* mencakup auth validation dan input handling.

- **WhatsAuth Controller (27.3%):** Cakupan relatif moderat karena fungsi webhook WhatsApp (`PostInboxNomor`) memiliki ketergantungan kuat pada berbagai layanan eksternal seperti Gemini AI API untuk generate respons, Fonnte WhatsApp API untuk pengiriman pesan, dan Kimseok webhook sebagai fallback, sehingga sebagian besar pengujian unit harus menggunakan *mock object* atau melewatkannya integrasi tersebut. Terdapat 18 *test functions* untuk message format validation dan *edge cases*.

Secara keseluruhan, arsitektur backend telah melalui validasi yang memadai pada mayoritas modul fungsional, ditunjukkan dengan rata-rata coverage 59.2% yang diperoleh dari total 97 *test functions* dengan tingkat keberhasilan 100%.

## BAB V

# PENUTUP

### 5.1 Kesimpulan

Berdasarkan seluruh rangkaian tahapan pengembangan sistem yang dimulai dari analisis kebutuhan hingga tahap implementasi, dapat disimpulkan bahwa aplikasi “GOPOS (*Go-Lang POS Assistant*)” telah berhasil diwujudkan sebagai asisten virtual cerdas yang handal bagi layanan PT Pos Indonesia. Penggunaan bahasa pemrograman Go yang dikombinasikan dengan arsitektur *serverless* dan basis data MongoDB Atlas terbukti mampu menghasilkan sistem dengan performa tinggi yang dapat menangani data riwayat percakapan secara dinamis. Keberhasilan sistem ini juga didukung oleh integrasi model Gemini Pro AI yang mampu memberikan respon bahasa natural secara kontekstual, baik untuk kebutuhan informasi tarif pengiriman maupun layanan umum lainnya melalui platform web dan WhatsApp.

Implementasi antarmuka pada sisi frontend yang mencakup halaman beranda, *dashboard* admin, dan fitur chat AI telah berfungsi secara optimal dengan tingkat responsivitas yang baik pada berbagai perangkat. Dari sisi validasi teknis, pengujian logika melalui metode *White Box Testing* menggunakan *unit testing* pada bahasa Go menunjukkan bahwa alur data internal, struktur koleksi pada database, serta fungsi normalisasi teks telah berjalan dengan akurat sesuai dengan spesifikasi yang dirancang. Secara teknis, sistem ini telah memenuhi standar kelayakan operasional tanpa adanya kesalahan logika krusial pada modul-modul utamanya, sehingga siap digunakan untuk meningkatkan efisiensi layanan pelanggan.

### 5.2 Saran

Meskipun sistem GOPOS saat ini telah berjalan dengan stabil, terdapat beberapa peluang pengembangan strategis yang dapat diimplementasikan untuk meningkatkan kapabilitas asisten virtual ini di masa depan. Pengembangan selanjutnya disaranankan untuk mengintegrasikan sistem secara langsung dengan API pelacakan kiriman (*tracking*) resmi dari PT Pos Indonesia guna memberikan kemudahan bagi pengguna dalam memantau status paket mereka secara instan melalui percakapan. Selain itu, proses *fine-tuning* pada model AI menggunakan data internal perusahaan yang lebih spesifik serta penambahan fitur *payment gateway* untuk transaksi ongkos kirim akan memberikan nilai tambah yang signifikan bagi ekosistem aplikasi. Peningkatan dari sisi keamanan melalui autentikasi dua faktor dan ekspansi ke berbagai kanal pesan instan lainnya juga menjadi langkah penting untuk menjamin perlindungan

an privasi pengguna sekaligus menjangkau basis pelanggan yang lebih luas secara *multi-channel*.

## **DAFTAR PUSTAKA**

- Aidyn, A. G. (2024). “Integrasi Layanan Cloud Storage dengan WhatsApp Bot Menggunakan Webhook dan Firebase”.
- Barman, P. K. (2025). “A Secure and Personalized AI Chatbot for School Performance Monitoring using Local LLM and MongoDB”.
- Dimara, Denis Lizard Sambawo (2024). “Integrasi Model Artificial Intelligence Llama3-70B dalam Pembuatan API Chatbot Menggunakan Golang untuk Aplikasi Backend”. Thesis.
- P. R. Journal (2015). “Boosting Application Functionality: Integrating Cloud Functions with Google Cloud Services”, 369–375.
- Rachmat, N. & D. P. Kesuma (2024). “Implementasi Large Language Models Gemini Pada Pengembangan Aplikasi Chatbot Berbasis Android”. 1, 40–52.

## LAMPIRAN

### Lampiran A: Code Coverage

Berikut adalah detail *code coverage* untuk setiap modul controller pada sistem GO-POS:

- **auth.go** - 69.1%
- **chat\_history.go** - 58.5%
- **chatbot.go** - 93.2%
- **user\_management.go** - 29.1%
- **whatsauth.go** - 27.3%

## GLOSARIUM

### Istilah Teknis

<b>Agile (Scrum)</b>	Metodologi pengembangan perangkat lunak yang bersifat iteratif (berulang) dan kolaboratif untuk merespons perubahan dengan cepat.
<b>API</b>	<i>Application Programming Interface</i> , antarmuka yang memungkinkan dua aplikasi atau lebih untuk saling berkomunikasi dan bertukar data.
<b>BSON</b>	<i>Binary JSON</i> , format penyimpanan data berbasis dokumen yang digunakan oleh MongoDB, mendukung efisiensi ruang dan kecepatan.
<b>Chatbot</b>	Program komputer yang dirancang untuk mensimulasikan percakapan dengan pengguna manusia, terutama melalui internet.
<b>Code Coverage</b>	Metrik untuk mengukur sejauh mana kode sumber program telah dijalankan selama pengujian unit.
<b>Gemini AI</b>	Model bahasa besar ( <i>Large Language Model</i> ) mutakhir dari Google yang mampu memahami dan menghasilkan teks secara natural.
<b>Golang (Go)</b>	Bahasa pemrograman kompilasi yang dikembangkan Google, dikenal karena efisiensi memori dan dukungan konkurensi.
<b>MongoDB</b>	Basis data NoSQL berbasis dokumen yang menyimpan data dalam format fleksibel seperti JSON/BSON.
<b>NLP</b>	<i>Natural Language Processing</i> , cabang kecerdasan buatan yang fokus pada interaksi antara komputer dan bahasa manusia agar mesin dapat memahami konteks bicara.
<b>Serverless</b>	Model komputasi di mana pengembang dapat menjalankan kode tanpa harus mengelola infrastruktur server fisik secara manual.
<b>Webhook</b>	Metode komunikasi otomatis di mana satu sistem mengirimkan data ke sistem lain segera setelah suatu peristiwa

terjadi.

**White Box Testing** Metode pengujian perangkat lunak di mana penguji mengetahui dan menguji struktur internal atau alur kode program.

### Istilah Non-Teknis

<b>Real-time</b>	Kondisi di mana sistem merespons perintah pengguna secara langsung dan seketika tanpa penundaan waktu yang lama.
<b>User Interface</b>	Tata letak visual dari aplikasi yang memungkinkan pengguna untuk berinteraksi dengan sistem.
<b>Scalability</b>	Kemampuan sistem untuk menangani beban kerja yang meningkat dengan menambah sumber daya secara otomatis.