

Perbandingan Tahap Implementasi Dan Performa Chatbot Telegram Dengan Platform Google Dialogflow Dan Database Postgresql

Ibnu Qomaril Huda¹, Agus Prihanto²

^{1,2} Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

¹ibnu.19024@msh.unesa.ac.id

²agusprihanto@unesa.ac.id

Abstrak— Komunikasi di era saat ini banyak menggunakan media sosial sebagai perantaranya. Saat ini implementasi *chatbot* pada media sosial kerap dilakukan demi mempermudah pekerjaan. Dalam pembuatan *chatbot*, ada beberapa platform yang dapat mempermudah pembuatan *chatbot*, contohnya seperti *Google Dialogflow*. Selain memberikan dampak positif, implementasi *chatbot* menggunakan *Google Dialogflow* juga memiliki dampak negatif juga. Penelitian ini bertujuan untuk melakukan perbandingan tahap implementasi dan performa *chatbot Telegram* dengan menggunakan platform *Google Dialogflow* dan database *PostgreSQL*.

Hasil penelitian menunjukkan bahwa tahap implementasi *chatbot Telegram* menggunakan platform *Google Dialogflow* lebih sedikit daripada platform database *PostgreSQL*. Dalam pengujian ketepatan respon *chatbot*, platform *Google Dialogflow* dan database *PostgreSQL* dengan menggunakan metode komunikasi *webhook* dan *long polling* mampu mengeluarkan respon yang sesuai dengan input pengguna. Dalam pengujian kecepatan respon *chatbot*, pada metode komunikasi *webhook* platform *Google Dialogflow* memiliki waktu respon yang lebih lambat dengan total rata-rata waktu respon 13.51663451 detik daripada platform database *PostgreSQL* dengan total rata-rata waktu respon 2.775305791 detik. Sedangkan pada metode komunikasi *long polling*, platform *Google Dialogflow* memiliki waktu respon yang lebih lambat juga dengan total rata-rata waktu respon 6.613519083 detik daripada platform database *PostgreSQL* dengan total rata-rata waktu respon 4.709665404 detik. Pada pengujian kestabilan *chatbot* menunjukkan bahwa pada kedua platform, metode komunikasi *webhook* lebih stabil daripada metode komunikasi *long polling* karena perbedaan mekanisme alur kerja dari setiap metode.

Kata Kunci— Chatbot, Telegram, Ketepatan, Kecepatan, Kestabilan

I. PENDAHULUAN

Perkembangan teknologi komunikasi dan informasi di abad modern ini adalah sebuah kemajuan peradaban manusia. Perkembangan teknologi ini berjalan begitu pesat sehingga membawa pengaruh yang luas terhadap kehidupan sosial masyarakat. Oleh karena itu, perkembangan teknologi ini telah mencapai tingkat kebutuhan bagi manusia yang vital di kalangan masyarakat [1].

Komunikasi merupakan suatu interaksi yang mengaitkan tahapan ketika informasi dan instruksi dapat terhubung dari satu pihak ke pihak yang lain. Komunikasi dapat dilakukan secara langsung maupun virtual. Komunikasi virtual ini yang berkembang di kalangan masyarakat adalah menggunakan media sosial atau layanan pesan instan sebagai perantaranya.

Media sosial atau layanan pesan instan merupakan salah satu teknologi yang berhasil secara signifikan meningkatkan performa pengguna dalam bertukar informasi satu sama lain [2]. Salah satu contoh media sosial atau layanan pesan instan ini adalah *WhatsApp*, *Facebook*, *Telegram*, dan lain-lain.

Pada perkembangannya, layanan pesan instan ini tidak hanya digunakan untuk berkomunikasi saja, tetapi juga digunakan untuk kepentingan pekerjaan suatu organisasi atau instansi. Pemimpin organisasi atau instansi biasanya memanfaatkan fitur *group chat* yang tersedia pada layanan pesan instan supaya informasi yang diberikan dapat diterima oleh banyak orang tanpa harus mengirim pesan satu per satu. Selain penggunaannya sebagai alat komunikasi, perkembangan layanan pesan instan ini juga mampu memunculkan ide-ide baru, seperti pembuatan *chatbot* dalam mempermudah penyampaian informasi.

Chatbot merupakan salah satu contoh dari sistem *Artificial Intelligence* (AI) dan tersebar luas dari *Human-Computer Interaction* (HCI). *Chatbot* merupakan program komputer yang menanggapi layaknya sebuah entitas cerdas ketika berbicara melalui teks atau suara dan memahami bahasa manusia dengan *Natural Language Processing* (NLP) [3]. Saat ini banyak sekali teknologi-teknologi yang dapat mempercepat dan membantu dalam pembuatan dan pengembangan *chatbot*, seperti platform *Google Dialogflow*. *Google Dialogflow* adalah platform pemahaman bahasa alami dari *Google* yang membantu pengembang merancang antarmuka percakapan dan mengintegrasikan mereka ke dalam aplikasi mereka. *Google Dialogflow* dapat diintegrasikan dengan layanan kognitif seperti analisis sentimen, layanan basis pengetahuan, dan lain-lain [4].

Dalam teknologi *chatbot* pada layanan pesan instan *Telegram* terdapat metode *webhook* dan *long polling* yang mampu membuat layanan pesan instan dapat menjawab permintaan orang lain secara otomatis. Metode *webhook* merupakan salah satu metode pertukaran informasi dalam *bot engine* untuk merespon ataupun memproses informasi secara otomatis [5]. Sedangkan metode *long polling* merupakan metode default dari *Telegram* dimana setiap permintaan pengguna akan diantrikan karena proses transaksi informasi antara *Telegram Bot* dan server *Telegram* dilakukan secara periodik dalam survei server [6].

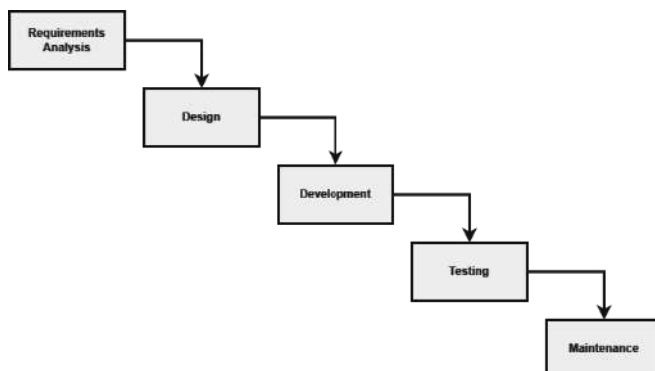
Dalam pembuatan *chatbot*, ada banyak sekali tahapan-tahapan yang perlu dilakukan supaya *chatbot* bekerja dengan baik dan mampu memenuhi kebutuhan pengguna. Dengan banyaknya tahapan-tahapan yang ada, tidak jarang para

pengembang aplikasi memerlukan waktu yang lama untuk mengembangkan dan mengimplementasikan *chatbot*. Disisi lain, ada beberapa faktor-faktor yang mempengaruhi kepuasan pengguna, contohnya adalah kecepatan respon dari *chatbot* dan kesesuaian data yang diminta oleh pengguna. Untuk mengetahui bagaimana hasil kinerja dari pembuatan *chatbot* adalah dengan melakukan pengujian. Pengujian merupakan tahapan di mana aplikasi diverifikasi apakah sudah memenuhi kebutuhan ataukah belum [7]. Dengan tahapan ini dapat dilakukan verifikasi *chatbot* apakah mampu mengeluarkan informasi yang sesuai dengan permintaan pengguna dan melakukan analisa performa yang dihasilkan *chatbot*.

Berdasarkan latar belakang yang telah diuraikan, penelitian ini bertujuan untuk membandingkan tahap implementasi dan performa *chatbot Telegram* dengan platform *Google Dialogflow* dan database *PostgreSQL*.

II. METODE PENELITIAN

Metode penelitian adalah tata cara atau sebuah mekanisme untuk menyelesaikan masalah penelitian secara sistematis. Metode penelitian dapat juga diartikan sebagai ilmu yang mempelajari bagaimana penelitian dilakukan secara ilmiah [8]. Pada penelitian ini akan menggunakan metode *waterfall* dalam mengembangkan *chatbot*. Metode *waterfall* merupakan salah satu model SDLC yang banyak digunakan dalam perancangan maupun pengembangan sistem perangkat lunak. Tahapan metode *waterfall* antara lain analisis kebutuhan (*requirements analysis*), desain (*design*), implemmtasi (*development*), pengujian (*testing*), dan pemeliharaan (*maintenance*) [9].



Gambar 1. Tahapan metode *waterfall*

A. Analisis Kebutuhan

1. Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan dari fasilitas yang diperlukan serta aktivitas apa saja yang akan dilakukan oleh sistem. Kebutuhan fungsional adalah jenis kebutuhan yang berisi kegiatan-kegiatan yang nantinya dilakukan oleh sistem, selain itu kebutuhan fungsional berisi tentang informasi-informasi yang harus ada dan dikeluarkan oleh sistem. Analisis kebutuhan fungsional bertujuan untuk mengetahui apa saja yang diperlukan oleh aplikasi nantinya. Analisis kebutuhan fungsional mempunyai beberapa syarat yaitu kegiatan-kegiatan yang harus dilakukan dalam sistem berdasarkan prosedur dan fungsi-fungsi bisnis, serta didokumentasikan dalam model. Kebutuhan fungsional meliputi laporan baik *hardcopy* maupun *softcopy*, *updating* data, penyimpanan data, dan pencarian data [10].

TABEL I
CONTOH HASIL ANALISIS KEBUTUHAN FUNGSIONAL APLIKASI

No	Kebutuhan Utama	Tujuan
1	Informasi pengurusan SKCK	<i>Chatbot</i> harus mampu mengeluarkan informasi mengenai bagaimana dan apa saja yang diperlukan dalam pengurusan dokumen SKCK.
2	Informasi pengambilan KTP	<i>Chatbot</i> harus mampu mengeluarkan informasi mengenai bagaimana dan apa saja yang diperlukan dalam pengambilan KTP di kantor desa.
3	Informasi pengurusan pindah KK	<i>Chatbot</i> harus mampu mengeluarkan informasi mengenai bagaimana dan apa saja yang diperlukan dalam pengurusan dokumen untuk pindah KK.
4	Informasi pengurusan surat hibah	<i>Chatbot</i> harus mampu mengeluarkan informasi mengenai bagaimana dan apa saja yang diperlukan dalam pengurusan surat hibah.
5	Informasi pengurusan surat keterangan lahir	<i>Chatbot</i> harus mampu mengeluarkan informasi mengenai bagaimana dan apa saja yang diperlukan dalam pengurusan surat keterangan lahir.
6	Informasi data diri	<i>Chatbot</i> harus mampu mengeluarkan informasi mengenai data pengguna saat berinteraksi.

2. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dapat dianggap sebagai informasi dari spesifikasi yang harus dipenuhi oleh sistem perangkat lunak dan ini merupakan hal utama yang harus ditangani selama proses pembangunan perangkat lunak. Kebutuhan non-fungsional sedikit rumit dalam menganalisisnya karena harus menganalisa karakteristik dan batasan dari sistem aplikasi. Identifikasi dan verifikasi kebutuhan non-fungsional menjadi hal utama dalam *software requirement* karena dengan detilnya kebutuhan non-fungsional yang diperoleh, akan mempercepat penentuan desain software dan mudah untuk direalisasikan oleh pengembang aplikasi [11].

TABEL II
CONTOH SPESIFIKASI *HARDWARE* PADA RANCANG BANGUN APLIKASI

No	Jenis Barang	Spesifikasi
1	Prosesor	Intel Core i7-1000H
2	RAM	8.00 GB
3	Penyimpanan	SSD 128 GB
4	Sistem Operasi	Ubuntu 16.02 LTE

TABEL III
CONTOH SPESIFIKASI *SOFTWARE* PADA RANCANG BANGUN APLIKASI

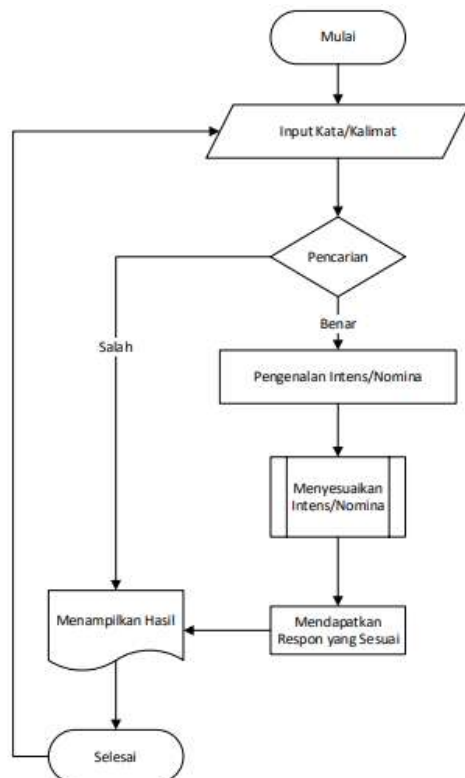
No	Jenis <i>Software</i>	Spesifikasi
1	Google Chrome	Versi 115.0.5790.101 (<i>Official Build</i>)
2	PostgreSQL	Versi 14.0
3	Go	Versi 1.19.0

B. Desain

1. Desain Diagram Flowchart Chatbot

Desain diagram *flowchart chatbot* merupakan sebuah gambaran bagaimana *chatbot* memproses dan

mengembalikan data berdasarkan input dari pengguna. Desain diagram akan menggunakan *flowchart* supaya mudah memahami jenis proses data dan juga bagaimana alur kerjanya.



Gambar 2. Contoh diagram *flowchart* alur kerja *chatbot*

2. Desain Percakapan Chatbot

Desain alur chatbot merupakan sebuah gambaran bagaimana chatbot nantinya bekerja memenuhi permintaan penggunaannya.

TABEL IV
CONTOH ENTITIES PADA CHATBOT UNTUK MENDETEKSI INPUT PENGGUNA

No	Nama Entities	Kata Kunci	Sinonim
1	IntroductionChat	hi	Wassap, Yo, Hai, Gass, Mulai, Start, Coba, Woi, Testing, Tes, Hi, Halo, Selamat
		not-found	Not Found
2	MenusChat	informasi-pengurusan-skck	1, Informasi Pengurusan SKCK,
		informasi-pembuatan-ktp	2, Informasi Pembuatan KTP, KTP
		informasi-pengurusan-pindah-kk-nikah	3, Informasi Pengurusan Pindah KK karena Nikah, KK, Nikah
		informasi-pengurusan-surat-hibah-tanah	4, Informasi Pengurusan Surat Hibah Tanah, Hibah, Tanah
		informasi-pengurusan-surat-keterangan-lahir	5, Informasi Pengurusan Surat Keterangan Lahir, Lahir

No	Nama Entities	Kata Kunci	Sinonim
		informasi-data-diri	6, Informasi Data Diri, Data Diri

TABEL V
CONTOH INTENTS PADA CHATBOT UNTUK MENGELOUARKAN RESPON BERDASARKAN KATA KUNCI

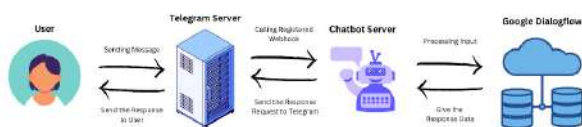
No	Nama Intents	Kata Kunci	Respon
1	Starting Chat	hi	Halo saudara/saudari nama selamat pagi, selamat datang di Sistem Informasi Terpadu. Silahkan pilih layanan yang anda inginkan: 1. Informasi Pengurusan SKCK 2. Informasi Pembuatan KTP 3. Informasi Pengurusan Pindah KK 4. Informasi Pengurusan Surat Hibah 5. Informasi Pengurusan Surat Keterangan Lahir 6. Informasi Data Diri
2	Unknown Command	not-found	Pesan yang anda masukkan tidak sesuai. Silahkan coba kembali
3	Starting Chat - Informasi Pengurusan SKCK	informasi-pengurusan-skck	Berikut adalah dokumen yang perlu dipersiapkan untuk mengurus SKCK: 1. Fotokopi KTP (Kartu Tanda Penduduk) atau Surat Izin Mengemudi (SIM). Bawa KTP/SIM asli sebagai bukti. 2. Fotokopi KK (Kartu Keluarga). 3. Fotokopi Akta Kelahiran / Surat Kenal Lahir. 4. Fotokopi Paspor (Bagi yang punya paspor). 5. Pas foto 4x6 berlatar/background merah sebanyak 6 lembar. Berpakaian sopan dan berkerah, serta tidak menggunakan aksesoris wajah, tampak muka dan bagi yang mengenakan jilbab, pas foto harus tampak muka secara utuh sebagai syarat membuat SKCK.
4	Starting Chat - Informasi Pembuatan KTP	informasi-pembuatan-ktp	Berikut adalah dokumen yang perlu dipersiapkan untuk membuat KTP: 1. Fotokopi Kartu Keluarga 2. Fotokopi Akta Kelahiran 3. Fotokopi Ijasah Terakhir. Apabila sudah melakukan perekaman KTP dan hendak mengambil KTP, berikut dokumen yang perlu dipersiapkan: 1. Fotokopi Kartu Keluarga 2. Surat Keterangan Perekaman Asli
5	Starting Chat - Informasi Pengurusan Pindah KK	informasi-pengurusan-pindah-kk-nikah	Untuk tahapan mengurus pindah KK karena menikah adalah sebagai berikut:

No	Nama Intents	Kata Kunci	Respon
	Pindah KK Karena Nikah		1. Datang ke kantor dukcapil terdekat 2. Mengambil nomor antrian 3. Mengisi berkas yang telah disediakan oleh dukcapil 4. Melakukan verifikasi berkas kepada petugas. Silahkan persiapkan dokumen yang perlu dibawa, yaitu: 1. Fotokopi buku nikah atau akta perkawinan 2. Kartu keluarga masing-masing orang tua
6	Starting Chat - Informasi Pengurusan Surat Hibah Tanah	informasi-pengurusan-surat-hibah-tanah	Silahkan persiapkan dokumen-dokumen untuk mengurus surat hibah: 1. Surat keterangan waris 2. Membawa dokumen hasil dari persyaratan peralihan hak atas tanah (silahkan datang ke kantor Wilayah Badan Pertanahan Nasional terdekat) 3. Menyiapkan biaya pembuatan akta hibah
7	Starting Chat - Informasi Pengurusan Surat Keterangan Lahir	informasi-pengurusan-surat-keterangan-lahir	Siapkan dokumen yang diperlukan, yaitu: 1. Surat keterangan kelahiran dari dokter 2. Buku nikah/akta pernikahan 3. KK orang tua 4. KTP orang tua. Setelah dokumen selesai silahkan datang ke kantor pencatatan sipil untuk mengisi formulir pencatatan sipil.
8	Starting Chat - Informasi Data Diri	informasi-data-diri	Berikut adalah data akun anda: - User ID: user_id - Nama: name

3. Desain Arsitektur Chatbot

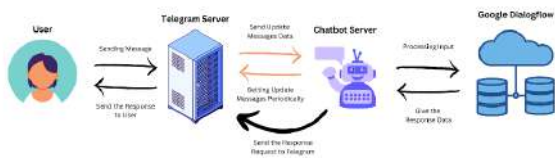
Desain arsitektur *chatbot* merupakan tahapan dimana bagaimana arsitektur bekerja dari user mengirim pesan hingga user mendapatkan pesan balasan dari *chatbot*.

Webhook + Google Dialogflow Architecture



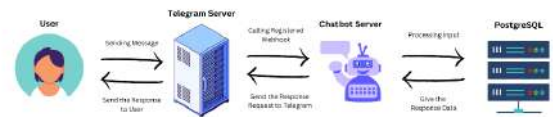
Gambar 3. Arsitektur chatbot menggunakan platform Google Dialogflow pada metode komunikasi webhook

Long Polling + Google Dialogflow Architecture



Gambar 4 Arsitektur chatbot menggunakan platform Google Dialogflow pada metode komunikasi long polling

Webhook + PostgreSQL Architecture



Gambar 5. Arsitektur chatbot menggunakan platform database PostgreSQL pada metode komunikasi webhook

Long Polling + PostgreSQL Architecture

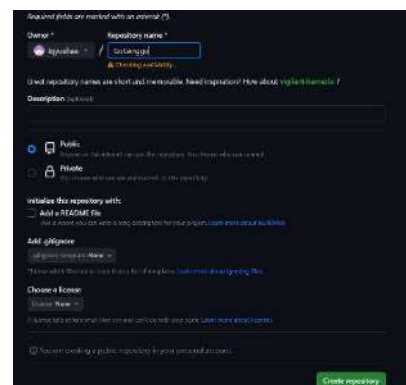


Gambar 6 Arsitektur chatbot menggunakan platform database PostgreSQL pada metode komunikasi long polling

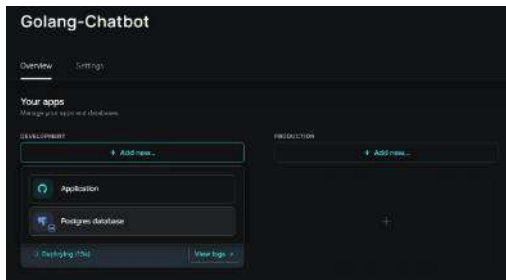
C. Implementasi

1. Persiapan Repository dan Hosting

Tahapan pertama adalah mempersiapkan *repository* di *Github* dan *hosting* yang akan digunakan untuk menyimpan program dari penelitian ini. Penelitian ini akan menggunakan layanan *hosting* dari f10.com. Untuk membuat *repository* baru di *Github*, pengembang aplikasi harus masuk dengan akun yang telah dimiliki pengembang aplikasi, menekan tombol *New* pada *list repository* dan mengisi nama *repository* yang akan digunakan nantinya. Setelah itu masuk ke layanan *hosting* f10.com dan membuat akun baru. Pada layanan ini pengembang aplikasi bisa memanfaatkan akun *gmail* sehingga tidak perlu repot membuat akun baru secara manual. Setelah itu pengembang aplikasi perlu membuat *workspace* baru. Kemudian ketika sudah selesai, agar kode yang ada di *Github* dapat terhubung dengan layanan dari *f10*, maka pilih tombol *deploy* menggunakan *Github*. Pilih *repository* yang telah dibuat sebelumnya dan hubungkan *branch* yang akan digunakan untuk menjalankan aplikasi ke layanan *f10*. Proses persiapan *repository* dan *hosting* sudah selesai, selanjutnya adalah mempersiapkan *database* PostgreSQL. Untuk membuat *database* baru, pengembang aplikasi hanya perlu menambahkan layanan *database* dan ketika sudah selesai layanan *f10* akan memberikan akses kredensial dari *database* PostgreSQL.



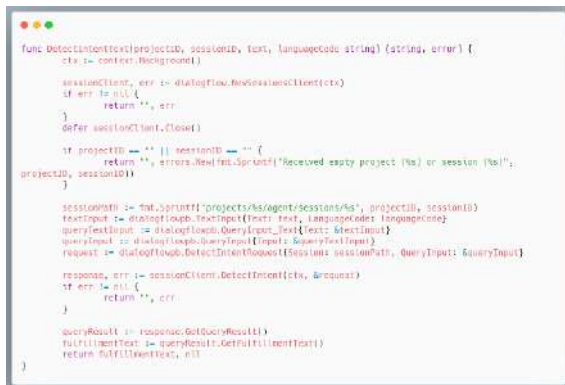
Gambar 7. Contoh membuat *repository* baru di *Github*



Gambar 8. Contoh hasil service di f10

2. Koneksi Google Dialogflow

Koneksi *Google Dialogflow* adalah sebuah mekanisme untuk menghubungkan aplikasi yang ada dengan platform *Google Dialogflow*. Dalam menghubungkan *chatbot* dengan platform *Google Dialogflow*, pengguna harus mengunduh *Google Application Credentials* dari platform *Google Cloud Platform*. Setelah mendapatkan kredensial, alamat file tersebut harus disimpan ke dalam *variable environment* dengan nama "GOOGLE_APPLICATION_CREDENTIALS". Lalu dengan memasang *package* dari *Google Dialogflow*, maka *chatbot* dapat terhubung.



Gambar 9. Contoh penggunaan *Google Dialogflow* menggunakan bahasa Go

3. Koneksi Database

Koneksi database merupakan sebuah mekanisme bagaimana menghubungkan aplikasi dengan *database*. Menurut dokumentasi resmi dari *PostgreSQL*, format data yang perlu dipersiapkan untuk menghubungkan aplikasi ke *database PostgreSQL* adalah sebagai berikut:

- 1) *host*: alamat dari *database PostgreSQL*, dapat berupa URL atau *IP Address*.
- 2) *port*: nomor layanan yang digunakan pada *database*. Dalam *database PostgreSQL* secara umum akan menggunakan *port 5432*.
- 3) *user*: nama pengguna untuk mengidentifikasi pengguna apakah terdaftar atau tidak.
- 4) *password*: kredensial dari pengguna atau *user* untuk memverifikasi apakah pengguna adalah pemilik akunnya.
- 5) *dbname*: sebuah nama *database* yang telah didefinisikan oleh pengguna dimana data akan disimpan.
- 6) *sslmode*: sebuah parameter yang digunakan *database PostgreSQL* untuk mengenkripsi seluruh komunikasi klien/server.



Gambar 10. Contoh format koneksi ke *database PostgreSQL*

4. Koneksi Telegram

Koneksi *Telegram* merupakan sebuah mekanisme bagaimana *Telegram Bot* dapat terhubung dengan server *Telegram*. Menurut dokumentasi resmi dari *Telegram*, hal yang pertama dilakukan ialah membuat sebuah akses kode yang akan digunakan untuk mengirim permintaan ke server *Telegram* dengan mengirimkan pesan "/newbot" ke akun *Telegram @BotFather*. Setelah itu akun *@BotFather* akan memandu kita untuk memasukkan nama dan *username* yang akan dibuat untuk membuat akses kode untuk *Telegram Bot*. Ketika seluruh proses selesai maka *@BotFather* akan mengirimkan akses kode yang nantinya akan digunakan untuk mengirim permintaan ke server *Telegram*.



Gambar 11. Contoh percakapan dengan *@BotFather* dalam membuat akses kode *Telegram Bot*

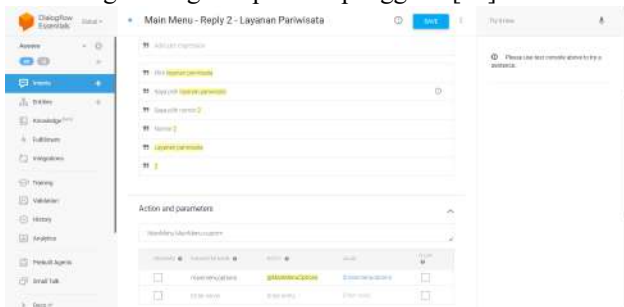


Gambar 12. Kode QR contoh cara berkomunikasi dengan server *Telegram* menggunakan bahasa Go

5. Pembangunan Chatbot

Dalam membangun *chatbot* dengan menggunakan *Google Dialogflow*, terdapat *intents* dan *entities* dimana

proses interaksi *chatbot* disimpan. *Intents* pada *Google Dialogflow* merupakan suatu modul yang mendefinisikan input yang masuk ke dalam percakapan dan melakukan tugasnya berdasarkan frasa tertentu [12]. Sedangkan *entities* merupakan suatu modul yang memiliki beberapa sampel dan kata dimana sampel-sampel tersebut berhubungan dengan input dari pengguna [12].



Gambar 13. Contoh *intents* pada platform *Google Dialogflow*



Gambar 14. Contoh *entities* pada platform *Google Dialogflow*

Sedangkan untuk *chatbot* yang menggunakan *database* berikut adalah contoh tabel sederhana untuk menyimpan input dan respon dari *chatbot* serta penggunaannya pada aplikasi.

id	chat_bot	chat_respon
1	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.
2	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.
3	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.
4	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.
5	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.
6	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.
7	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.
8	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.
9	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.
10	Siapa kamu?	Peran yang akan melakukan tugas utama. Silahkan coba kembali.

Gambar 15. Contoh struktur tabel untuk *chatbot* menggunakan *database* PostgreSQL



Gambar 16. Kode QR contoh penggunaan *database* untuk *chatbot* menggunakan bahasa Go

6. Penerapan Metode Webhook

Dalam menerapkan metode *webhook*, agar *chatbot* kita dapat saling berkomunikasi dengan platform *Telegram* maka perlu membuat satu *Application Programming Interface (API)* yang dimana nantinya platform *Telegram* akan mengirimkan data dari suatu *event* yang terjadi kepada

aplikasi milik sendiri. Dalam mengisi parameter data, ada beberapa parameter yang telah disediakan oleh *Telegram*, yaitu:

- 1) url: *HTTPS Uniform Resource Locator (URL)* yang akan menerima data dari server *Telegram* ketika ada pesan yang masuk.
- 2) certificate: Kunci publik dari domain ketika menggunakan *HTTPS*.
- 3) ip_address: sebuah alamat nomor sebagai pengganti dari domain atau url.
- 4) max_connections: Maksimal waktu yang diperlukan untuk mengirim data ketika ada pesan masuk dari 1 – 100 detik.
- 5) drop_pending_updates: Untuk menghilangkan *event update* dengan status *pending*.
- 6) secret_token: kode akses yang didapat dari @BotFather ketika membuat kode akses.



Gambar 17. Contoh request untuk mendaftarkan *endpoint webhook*



Gambar 18. Contoh respon berhasil mendaftarkan *endpoint webhook*

7. Penerapan Metode Long Polling

Dalam menerapkan metode *long polling* menurut dokumentasi resmi *Telegram*, *chatbot* dapat menggunakan satu *Application Programming Interface* yang telah diberikan oleh *Telegram*, yaitu *"/getUpdates"*. *Endpoint* ini hanya bisa digunakan selama metode *webhook* tidak diaktifkan, dan apabila metode *webhook* diaktifkan maka *endpoint* ini tidak akan bekerja. Dalam melakukan *request* ke server *Telegram* dengan metode *long polling*, ada beberapa parameter-parameter yang disediakan oleh *Telegram* untuk mengambil data secara spesifik, yaitu:

- 1) offset: Mengidentifikasi data pertama yang di-update yang akan dihasilkan. Harus berupa nomor dan lebih dari nol.
- 2) limit: Jumlah data yang akan dihasilkan dengan jarak 1 – 100.
- 3) timeout: Waktu yang digunakan untuk mengambil data dalam kurun waktu sekian detik bergantung pada isi dari parameter.
- 4) allowed_updates: Mengatur *bot* untuk menerima dan melakukan *update* pada suatu *event* tertentu.



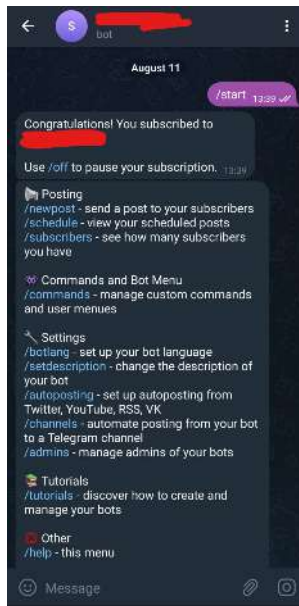
Gambar 19. Contoh request untuk mengambil pesan masuk menggunakan metode *long polling*



Gambar 20. Contoh respon berhasil mengambil data pesan menggunakan metode *long polling*

D. Pengujian

Pengujian merupakan tahapan di mana aplikasi diverifikasi apakah sudah memenuhi kebutuhan ataukah belum. Dalam lingkup rancang bangun aplikasi, ada dua metode pengujian yang sering digunakan, yaitu metode pengujian *whitebox* dan metode pengujian *blackbox*. Metode pengujian yang akan digunakan pada penelitian ini adalah metode pengujian *blackbox* untuk menguji hasil dari pembuatan *chatbot*-nya. Pengujian dengan metode *blackbox* merupakan pengujian yang sangat disesuaikan dan memverifikasi apakah spesifikasi fungsional telah terpenuhi dari hasilnya. Pengujian *blackbox* mampu mendeteksi kesalahan manufaktur yang tidak bisa dideteksi oleh pengujian *whitebox* [7]. Dibandingkan dengan pengujian *whitebox*, pengujian *blackbox* memeriksa input hingga *output* langsung daripada elemen-elemen individu dari fungsi aplikasi.



Gambar 21. Contoh pengujian *blackbox* pada *chatbot*

E. Pemeliharaan

Pemeliharaan dalam pengembangan aplikasi bertujuan untuk menjaga operasional sistem aplikasi dapat memenuhi kebutuhan pengguna. Pemeliharaan aplikasi merupakan fase terpanjang dari siklus pengembangan aplikasi karena tidak hanya memakan waktu saja, tetapi membutuhkan tenaga kerja manusia yang banyak juga [13]. Dalam tahapan pemeliharaan aplikasi, ada beberapa hal yang dapat dilakukan selain menjaga aplikasi dapat memenuhi kebutuhan pengguna, yaitu dengan melakukan *refactoring*. *Refactor* merupakan proses mengoptimalkan struktur internal kode tanpa mengubah perilaku eksternalnya [14]. Dengan melakukan *refactoring* kode aplikasi akan membuat program lebih berkualitas dan juga mudah untuk diperbaiki ketika terjadi kesalahan program.

III. HASIL PENELITIAN DAN PEMBAHASAN

A. Integrasi Dialog Chatbot

Pada sub bab ini akan membahas integrasi dialog *chatbot* menggunakan platform *Google Dialogflow* dan *PostgreSQL*. Tahapan-tahapan yang akan dijadikan acuan adalah koneksi menuju platform, pembangunan interaksi *chatbot*, dan proses implementasi *chatbot* ke program.

TABEL VI
TAHAPAN INTEGRASI DIALOG *CHATBOT* ANTAR PLATFORM

No	Tahapan	Tahapan Integrasi	
		Google Dialogflow	PostgreSQL
1	Koneksi menuju platform	1. Masuk ke <i>Google Cloud Platform</i> . 2. Pilih <i>project</i> yang digunakan. 3. Buat <i>role</i> baru di menu <i>Roles</i> dan masukkan seluruh <i>permission</i> <i>dialogflow</i> . 4. Buat <i>Service Account</i> di <i>IAM & Admin</i> . 5. Setelah selesai, masuk ke <i>Service Account</i> dan buat <i>private key</i> dengan tipe <i>JSON</i> di menu <i>Keys</i> . 6. Setelah <i>keys</i> <i>JSON</i> terunduh, pasang alamat file ke <i>variable environment</i> dengan nama " <i>GOOGLE_APPLICATION_CREDENTIALS</i> ".	1. Masuk ke <i>hosting f10.com</i> . 2. Pilih <i>project</i> , lalu buka <i>PostgreSQL database</i> . 3. Setelah mendapatkan data kredensial <i>database</i> , buat program untuk menghubungkan program ke <i>database</i> .
2	Pembangunan interaksi <i>chatbot</i>	1. Masuk ke platform <i>Google Dialogflow</i> menggunakan <i>browser</i> . 2. Buat pendeteksi input di menu <i>Entities</i> dengan memasukkan kata kunci dan sinonimnya. 3. Buat respon <i>chatbot</i> berdasarkan input dari <i>Entities</i> di menu <i>Intents</i> .	1. Membuat desain tabel untuk menyimpan data input dan respon <i>chatbot</i> . 2. Masuk ke <i>database PostgreSQL</i> menggunakan aplikasi <i>remote database</i> . 3. Membuat kueri untuk membangun tabel ke <i>database</i> . 4. Membuat kueri untuk menyimpan input dan respon <i>chatbot</i> .
3	Proses implementasi <i>chatbot</i> ke program	1. Memasang <i>library Google Dialogflow</i> . 2. Membuat program untuk menyimpan <i>session</i> pengguna. 3. Membuat program untuk mendapatkan respon berdasarkan input pengguna.	1. Membuat program untuk menyimpan <i>session</i> pengguna. 2. Membuat program untuk mendeteksi input pengguna. 3. Mengecek kedalaman alur pengguna berdasarkan <i>session</i> . 4. Membuat program untuk mendapatkan respon berdasarkan input pengguna.

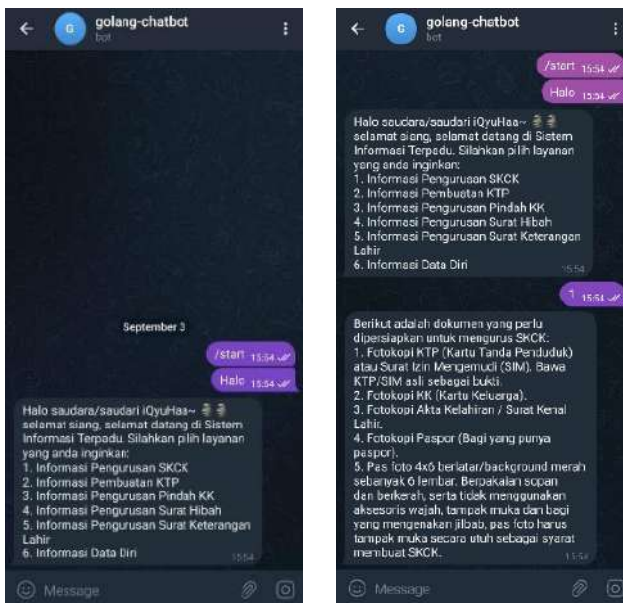
Berdasarkan tabel di atas menunjukkan bahwa pada tahapan implementasi dialog pada *chatbot*, platform *Google Dialogflow* memiliki kelebihan dalam membuat interaksi *chatbot* dan

mengintegrasikannya ke program aplikasi daripada PostgreSQL. Walaupun secara tahapan koneksi menuju platform PostgreSQL lebih unggul, tahapan pembangunan interaksi *chatbot* dan juga implementasi ke program adalah bagian utama dari implementasi dialog pada *chatbot* sehingga hasilnya adalah platform *Google Dialogflow* mampu mempercepat proses implementasi dialog pada *chatbot*.

B. Analisa Hasil Chatbot

1. Ketepatan Data Respon Chatbot

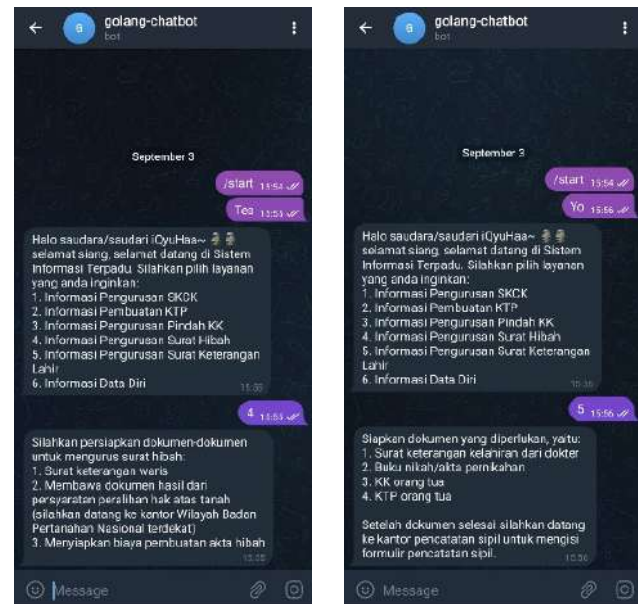
Pengujian ketepatan data yang dikeluarkan oleh *chatbot* berdasarkan input pengguna akan diuji berdasarkan analisis kebutuhan fungsional dan desain percakapan *chatbot*. Hasil uji coba menggunakan metode *webhook* dan *long polling* menghasilkan luaran yang sama berdasarkan input pengguna. Penggunaan platform *Google Dialogflow* dan PostgreSQL dalam proses pengolahan input pengguna juga menghasilkan respon yang sama pula. Seluruh uji coba dilakukan dengan mengirim pesan ke akun *telegram* @golang_chatbot yang sesuai dengan yang ada di Gambar 22, Gambar 23, Gambar 24, dan Gambar 25.



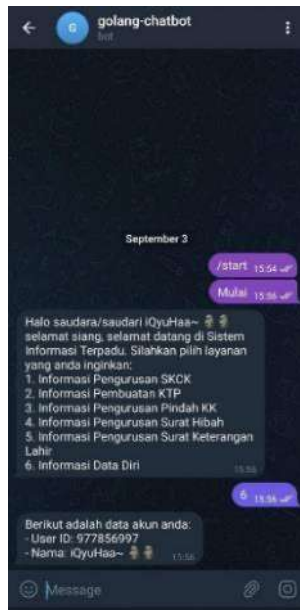
Gambar 22. Pengujian percakapan awal *chatbot* dan percakapan informasi pengurusan SKCK



Gambar 23. Pengujian percakapan informasi pengambilan KTP dan pengurusan pindah Kartu Keluarga



Gambar 24. Pengujian percakapan informasi pengurusan surat hibah dan keterangan lahir



Gambar 25. Pengujian percakapan informasi data diri

Berdasarkan seluruh uji coba yang telah dilakukan mengenai ketepatan respon data dari *chatbot*, dengan memperhatikan seluruh desain percakapan *chatbot* dapat disimpulkan bahwa penggunaan kedua metode dan kedua platform memiliki hasil respon yang sesuai dengan input yang diberikan.

TABEL VII
HASIL PENGUJIAN KETEPATAN DATA RESPON *CHATBOT*

No	Kebutuhan Utama	Google Dialogflow		PostgreSQL	
		Webhook	Long Polling	Webhook	Long Polling
1	Informasi pengurusan SKCK	✓	✓	✓	✓
2	Informasi pengambilan KTP	✓	✓	✓	✓
3	Informasi pengurusan pindah KK	✓	✓	✓	✓
4	Informasi pengurusan surat hibah	✓	✓	✓	✓
5	Informasi pengurusan surat keterangan lahir	✓	✓	✓	✓
6	Informasi data diri	✓	✓	✓	✓

2. Kecepatan Respon Chatbot

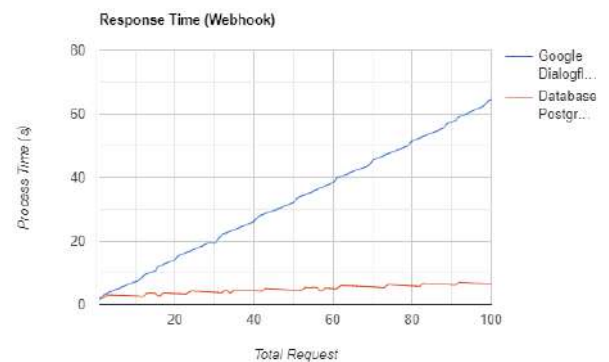
Pada pengujian ini akan melakukan skenario pengiriman pesan secara masal dengan jumlah 1 kali, 10 kali, 25 kali, 50 kali, dan 100 kali untuk menguji seberapa cepat *chatbot* merespon pesan. Berikut adalah tabel dan grafik perbandingan hasil pengujian kecepatan respon antar platform pada metode komunikasi *webhook*.

TABEL VIII
HASIL PENGUJIAN KECEPATAN RESPON *CHATBOT* PADA METODE KOMUNIKASI *WEBHOOK* DENGAN *GOOGLE DIALOGFLOW*

No	Jumlah (kali)	Minimal (detik)	Maksimal (detik)	Rata-rata (detik)
1	1	1.467681011	1.467681011	1.467681011
2	10	1.857764091	9.052640225	5.497899522
3	25	1.757844314	18.47142917	10.26259288
4	50	1.408469660	32.95071118	17.38775812
5	100	1.605186225	64.48504135	32.96724100
Rata-rata		1.619389060	25.28550059	13.51663451

TABEL IX
HASIL PENGUJIAN KECEPATAN RESPON *CHATBOT* PADA METODE KOMUNIKASI *WEBHOOK* DENGAN *POSTGRSQL*

No	Jumlah (kali)	Minimal (detik)	Maksimal (detik)	Rata-rata (detik)
1	1	1.594886467	1.594886467	1.594886467
2	10	2.472906778	5.532089101	3.188127759
3	25	0.980287559	2.701948919	1.957010981
4	50	0.917804635	3.510921464	2.257733685
5	100	2.125928170	7.076828242	4.878770061
Rata-rata		0.980287559	2.701948919	1.957010981



Gambar 26. Grafik perbandingan waktu respon *chatbot* antar platform berdasarkan jumlah pesan pada metode komunikasi *webhook*

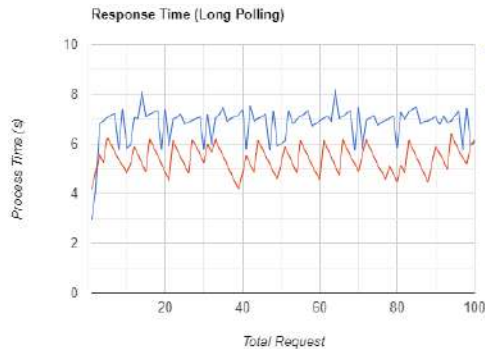
Selanjutnya adalah tabel dan grafik hasil pengujian kecepatan respon antar platform pada metode komunikasi *long polling* dengan pengecekan pesan setiap 5 detik sebanyak 10 pesan.

TABEL X
HASIL PENGUJIAN KECEPATAN RESPON *CHATBOT* PADA METODE KOMUNIKASI *LONG POLLING* DENGAN *POSTGRSQL*

No	Jumlah (kali)	Minimal (detik)	Maksimal (detik)	Rata-rata (detik)
1	1	5.239612343	5.239612343	5.239612343
2	10	2.041349456	8.377227186	6.643929137
3	25	5.895075950	8.938607606	7.592389958
4	50	1.861392879	8.306497026	6.768258177
5	100	2.960252071	8.175752282	6.823405801
Rata-rata		3.599536549	7.807539289	6.613519083

TABEL XI
HASIL PENGUJIAN KECEPATAN RESPON *CHATBOT* PADA METODE
KOMUNIKASI *LONG POLLING* DENGAN POSTGRESQL

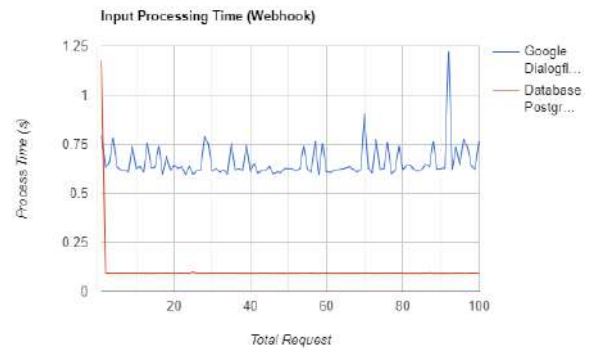
No	Jumlah (kali)	Minimal (detik)	Maksimal (detik)	Rata-rata (detik)
1	1	3.198433450	3.198433450	3.198433450
2	10	1.814428954	6.167530391	4.587111563
3	25	2.058322284	6.134316699	5.141352747
4	50	3.558991535	6.158904272	5.243115336
5	100	4.160891718	6.393106312	5.378313925
Rata-rata		2.958213588	5.610458225	4.709665404



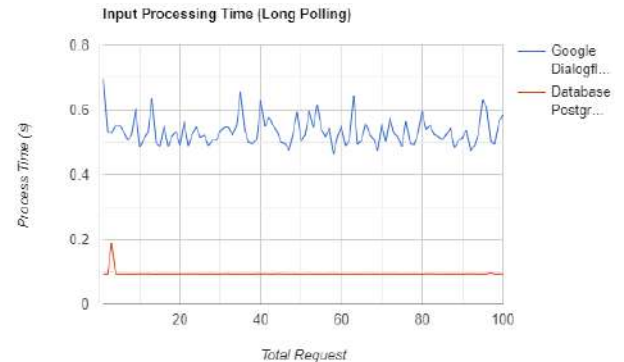
Gambar 27. Grafik perbandingan waktu respon *chatbot* antar platform berdasarkan jumlah pesan pada metode komunikasi *long polling*

Berdasarkan pengujian yang telah dilakukan dapat disimpulkan bahwa,

- 1) Pada metode komunikasi *webhook*, platform *Google Dialogflow* memiliki waktu respon lebih lambat dengan total rata-rata waktu respon 13.51663451 detik daripada platform PostgreSQL dengan total rata-rata waktu respon 2.775305791 detik. Sedangkan pada metode komunikasi *long polling*, platform *Google Dialogflow* memiliki waktu respon lebih lambat juga dengan total rata-rata waktu respon 6.613519083 detik daripada platform PostgreSQL dengan total rata-rata waktu respon 4.709665404 detik.
- 2) Semua metode komunikasi yang menggunakan platform *Google Dialogflow* cenderung memiliki waktu respon lebih lama daripada yang menggunakan platform PostgreSQL. Hal ini bisa terjadi karena platform *Google Dialogflow* menggunakan gRPC sebagai media komunikasinya yang ada pada *library*-nya yang dimana ketika memanggil fungsi gRPC-nya masih ada proses yang dijalankan di platform *Google Dialogflow*, berbeda dengan platform PostgreSQL yang langsung mengambil data dari *datastore*. Hal ini juga didukung dengan lama waktu proses pengolahan input antar platform yang dapat dilihat pada Gambar 28 dan Gambar 29.



Gambar 28. Grafik perbandingan waktu proses pengolahan input pengguna antar platform pada metode komunikasi *webhook*



Gambar 29. Grafik perbandingan waktu proses pengolahan input pengguna antar platform pada metode komunikasi *long polling*

3. Kestabilan *Chatbot* Merespon Pesan

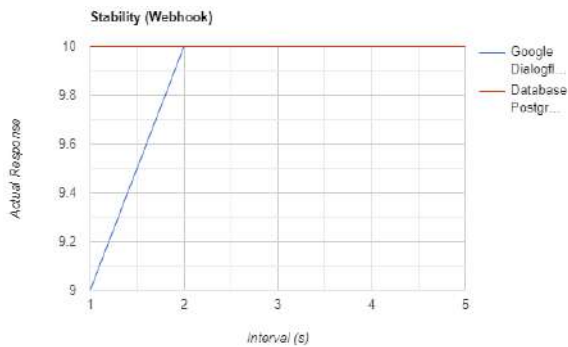
Pada pengujian ini akan menggunakan skenario pengiriman 10 pesan dengan interval waktu 1 detik untuk menguji metode dan platform yang digunakan untuk mengirim pesan secara stabil. Untuk hasil pengujian kestabilan *chatbot* antar platform pada metode komunikasi *webhook*.

TABEL XII
HASIL PENGUJIAN KESTABILAN *CHATBOT* PADA METODE KOMUNIKASI
WEBHOOK DENGAN *GOOGLE DIALOGFLOW*

No	Interval (detik)	Jumlah Respon
1	1	9
2	2	10
3	3	10
4	4	10
5	5	10

TABEL XIII
HASIL PENGUJIAN KESTABILAN *CHATBOT* PADA METODE KOMUNIKASI
WEBHOOK DENGAN *POSTGRESQL*

No	Interval (detik)	Jumlah Respon
1	1	1
2	2	10
3	3	10
4	4	10
5	5	10



Gambar 30. Grafik perbandingan pengujian kestabilan chatbot antar platform pada metode komunikasi webhook

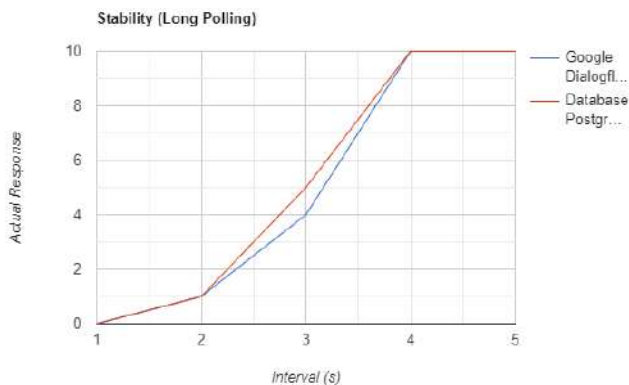
Selanjutnya untuk pengujian kestabilan chatbot antar platform pada metode komunikasi long polling.

TABEL XIV
HASIL PENGUJIAN KESTABILAN CHATBOT PADA METODE KOMUNIKASI
LONG POLLING DENGAN GOOGLE DIALOGFLOW

No	Interval (detik)	Jumlah Respon
1	1	0
2	2	1
3	3	4
4	4	10
5	5	10

TABEL XV
HASIL PENGUJIAN KESTABILAN CHATBOT PADA METODE KOMUNIKASI
LONG POLLING DENGAN POSTGRESQL

No	Interval (detik)	Jumlah Respon
1	1	0
2	2	1
3	3	5
4	4	10
5	5	10



Gambar 31. Grafik perbandingan pengujian kestabilan chatbot antar platform pada metode komunikasi long polling

Berdasarkan hasil penelitian menunjukkan bahwa,

- 1) Pada kedua metode komunikasi, platform PostgreSQL lebih stabil daripada platform Google Dialogflow. Hal ini terjadi karena terdapat proses yang ada pada platform Google Dialogflow pada saat pemanggilan fungsi gRPC pada library Google Dialogflow.
- 2) Pada metode komunikasi webhook semua platform mampu menghasilkan respon yang lebih tinggi daripada metode komunikasi long polling. Hal ini disebabkan karena perbedaan mekanisme alur kerja dari setiap

metode komunikasi dimana metode webhook akan langsung mengirim data dari server Telegram ke server chatbot sehingga pesan bisa langsung diproses. Sedangkan metode long polling perlu melakukan pengecekan pesan terbaru selama 5 detik secara berkala, hal ini yang membuat metode long polling tidak bisa menjawab pesan secara langsung.

IV. KESIMPULAN

Penelitian ini menganalisa kecepatan tahap implemementasi dan performa chatbot Telegram dengan platform Google Dialogflow dan database PostgreSQL. Berdasarkan hasil analisa yang telah dilakukan, kesimpulan yang dapat ditarik adalah sebagai berikut:

- 1) Penggunaan platform Google Dialogflow memiliki tahapan implementasi chatbot Telegram lebih singkat daripada platform database PostgreSQL. Pemanfaatan intents dan entities pada platform Google Dialogflow memudahkan pengembang aplikasi untuk dalam merancang percakapan chatbot. Dengan adanya library yang telah disediakan oleh Google Dialogflow menjadikan platform ini lebih cepat implementasinya daripada platform database PostgreSQL yang harus membuat program secara manual.
- 2) Dalam pengujian performa chatbot, hasil pengujian ketepatan data respon menunjukkan bahwa kedua metode komunikasi dengan kedua platform mampu mengeluarkan respon yang sesuai dengan permintaan pengguna. Dalam pengujian kecepatan respon, pada metode komunikasi webhook platform Google Dialogflow memiliki waktu respon yang lebih lambat dengan total rata-rata waktu respon 13.51663451 detik daripada platform PostgreSQL dengan total rata-rata waktu respon 2.775305791 detik. Sedangkan pada metode komunikasi long polling, platform Google Dialogflow memiliki waktu respon yang lebih lambat juga dengan total rata-rata waktu respon 6.613519083 detik daripada platform PostgreSQL dengan total rata-rata waktu respon 4.709665404 detik. Dalam pengujian kestabilan chatbot dalam merespon pesan, pada kedua platform metode webhook lebih stabil daripada metode long polling karena secara mekanisme alur kerja metode webhook langsung mengirimkan data ketika pesan diterima, sedangkan pada metode long polling perlu melakukan pengecekan pesan secara berkala (dalam penelitian ini dilakukan setiap 5 detik) yang mengakibatkan pesan harus menunggu sekitar 5 detik supaya pesan bisa diproses dan direspon.

UCAPAN TERIMA KASIH

Puji syukur atas kehadiran Allah SWT, atas segala rahmat, nikmat, dan hidayah-Nya penulis mampu menyelesaikan artikel ini. Terima kasih kepada orang tua dan seluruh anggota keluarga saya yang telah memberikan doa dan dukungan yang tiada henti supaya penulis tetap semangat mengerjakan artikel ini. Terima kasih juga kepada dosen pembimbing dan penguji saya yang telah memberikan masukan dan saran yang sangat bermanfaat agar artikel ini bisa lebih baik lagi. Dan terakhir, terima kasih kepada seluruh teman-teman dan pihak kampus yang telah membantu saya dalam mencari ilmu serta mendukung proses pengembangan artikel ini.

REFERENSI

- [1] S. S. Yoga, "PERUBAHAN SOSIAL BUDAYA MASYARAKAT INDONESIA DAN PERKEMBANGAN TEKNOLOGI KOMUNIKASI," 2018.
- [2] H. Soeroso, A. Zuhri Arfianto, N. Eka Mayangsari, and M. Taali, "Penggunaan Bot Telegram Sebagai Announcement System pada Intansi Pendidikan," *In Seminar Master PPN*, vol. 2, no. 1, pp. 45–48, 2017.
- [3] E. Adamopoulou and L. Moussiades, "An Overview of Chatbot Technology," in *IFIP Advances in Information and Communication Technology*, Springer, 2020, pp. 373–383. doi: 10.1007/978-3-030-49186-4_31.
- [4] N. Sabharwal and A. Agrawal, *Cognitive Virtual Assistants Using Google Dialogflow*. Apress, 2020. doi: 10.1007/978-1-4842-5741-8.
- [5] Moh. A. A. Widya and P. Airlangga, "PENGEMBANGAN TELEGRAM BOT ENGINE MENGGUNAKAN METODE WEBHOOK DALAM RANGKA PENINGKATAN WAKTU LAYANAN E-GOVERNMENT," *SAINTEKBU: Jurnal Sains dan Teknologi*, vol. 12, no. 2, pp. 13–22, 2020.
- [6] E. A. Kavats and A. A. Kostenko, "ANALYSIS OF CONNECTION METHODS OF TELEGRAM ROBOTS WITH SERVER PART," *System technologies*, vol. 3, no. 122, pp. 19–24, Oct. 2019, doi: 10.34185/1562-9945-3-122-2019-03.
- [7] R. Pan, Z. Zhang, X. Li, K. Chakrabarty, and X. Gu, "Black-Box Test-Cost Reduction Based on Bayesian Network Models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 2, pp. 386–399, Feb. 2021, doi: 10.1109/TCAD.2020.2994257.
- [8] M. Patel and M. Pater, "Exploring Research Methodology Review Article," *International Journal of Research and Review*, vol. 6, no. 3, pp. 48–55, 2019.
- [9] Ritzkal, P. P. Amalia, A. H. Hendrawan, F. Riana, and Kodarsyah, "Application Of The Waterfall Method In The Final Project Guidance Realization Information System," *Jurnal Mantik*, vol. 6, no. 2, pp. 1449–1458, 2022.
- [10] L. Setiyani and E. Tjandra, "ANALISIS KEBUTUHAN FUNGSIONAL APLIKASI PENANGANAN KELUHAN MAHASISWA STUDI KASUS:STMIK ROSMA KARAWANG," *Jurnal Inovasi Pendidikan dan Teknologi Informasi (JIPTI)*, vol. 2, no. 1, pp. 8–17, 2021, doi: 10.52060/pti.v2i01.465.
- [11] A. Aulia Aziiza and A. N. Fadhillah, "Analisis Metode Identifikasi dan Verifikasi Kebutuhan Non Fungsional," *Applied Technology and Computing Science Journal*, vol. 3, no. 1, 2020.
- [12] D. W. Harahap and L. Fitria, "APLIKASI CHATBOT BERBASIS WEB MENGGUNAKAN METODE DIALOGFLOW," *Jurnal Informatika dan Teknologi Komputer*, vol. 1, no. 1, pp. 6–13, 2020.
- [13] H. U. Rahman, M. Raza, P. Afsar, and H. U. Khan, "Empirical Investigation of Influencing Factors Regarding Offshore Outsourcing Decision of Application Maintenance," *IEEE Access*, vol. 9, pp. 58589–58608, 2021, doi: 10.1109/ACCESS.2021.3073315.
- [14] E. A. AlOmar, M. W. Mkaouer, C. Newman, and A. Ouni, "On preserving the behavior in software refactoring: A systematic mapping study," *Information and Software Technology*, vol. 140. Elsevier B.V., Dec. 01, 2021. doi: 10.1016/j.infsof.2021.106675.