

**IMPLEMENTASI GOOGLE SHEET API PADA WEBHOOK  
CHATBOT BERBASIS DIALOGFLOW UNTUK  
PENYIMPANAN DATA KOMPLAIN**



Disusun Oleh:

N a m a : Ibnu Ahmad Kamal  
NIM : 18523213

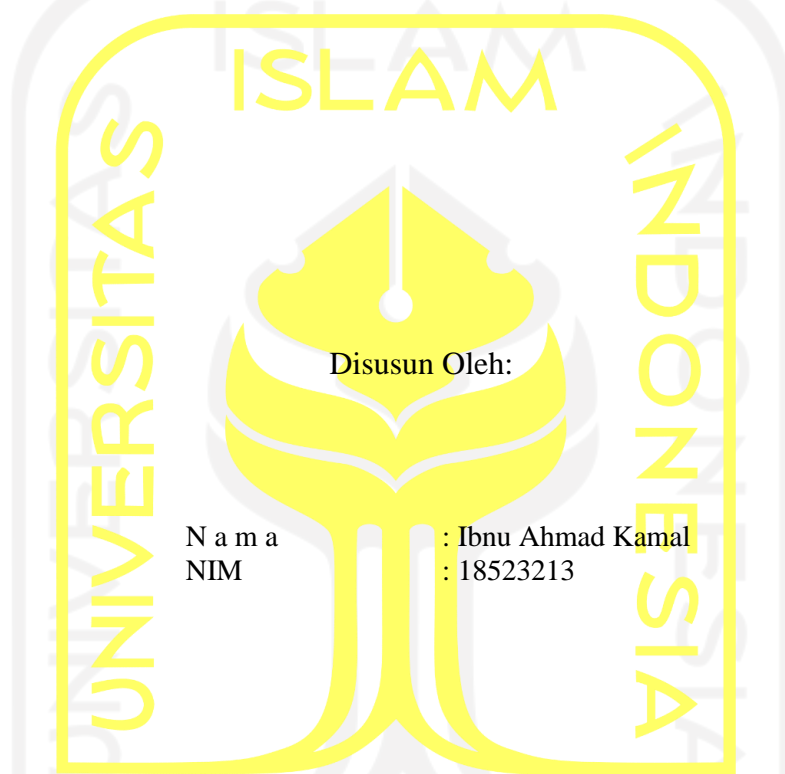
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2022**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**IMPLEMENTASI GOOGLE SHEET API PADA WEBHOOK  
CHATBOT BERBASIS DIALOGFLOW UNTUK  
PENYIMPANAN DATA KOMPLAIN**

**TUGAS AKHIR JALUR MAGANG**



الجامعة الإسلامية  
Yogyakarta, 16 Desember 2022

Pembimbing,




Andhik Budi Cahyono, S.T., M.T.

**HALAMAN PENGESAHAN DOSEN PENGUJI**

**IMPLEMENTASI GOOGLE SHEET API PADA WEBHOOK  
CHATBOT BERBASIS DIALOGFLOW UNTUK  
PENYIMPANAN DATA KOMPLAIN  
TUGAS AKHIR JALUR MAGANG**

Telah dipertahankan di depan sidang pengujian sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 05 Januari 2023

Tim Penguji	
Andhik Budi Cahyono, S.T., M.T.	
<b>Anggota 1</b>	
Andhika Giri Persada, S.Kom., M.Eng.	
<b>Anggota 2</b>	
Chandra Kusuma Dewa, S.Kom., M.Cs., Ph.D.	

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana  
Fakultas Teknologi Industri  
Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Ibnu Ahmad Kamal

NIM : 18523213

Tugas akhir dengan judul:

**IMPLEMENTASI GOOGLE SHEET API PADA WEBHOOK  
CHATBOT BERBASIS DIALOGFLOW UNTUK  
PENYIMPANAN DATA KOMPLAIN**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 16 Desember 2022



( Ibnu Ahmad Kamal )

## HALAMAN PERSEMBAHAN

*Assalamualaikum Warahmatullahi Wabarakatuh. Alhamdulillah* segala puji syukur kepada Allah SWT Yang maha Pengasih lagi Maha Penyayang dan tidak lupa salam shalawat serta salam penulis haturkan kepada Nabi Muhammad *Sallallahu Alaihi Wassalam*. Sehingga penulis dapat menyelesaikan laporan akhir ini. Laporan akhir ini penulis persembahkan kepada kedua orang tua, keluarga, dosen pembimbing serta semua teman yang telah memberikan dukungan, selamat, dan doa sehingga penulis dapat menyelesaikan laporan akhir ini. *Wassalamualaikum Warahmatullahi Wabarakatuh.*



**HALAMAN MOTO**

*“Apa yang melewatkanmu tidak akan pernah menjadi takdirmu, dan apa yang ditakdirkan untukmu tidak akan pernah melewatkanmu.”*

(Ummar Bin Khattab)

*“The programmers of tomorrow are the wizards of the future. You’re going to look like you have magic powers compared to everybody else.”*

(Gabe Newell)



## KATA PENGANTAR

*Assalamu'alaikum Wr. Wb.*

Alhamdulillah, segala puji dan syukur dihaturkan kepada Allah SWT yang telah melimpahkan berkah dan izin-Nya sehingga penulis dapat melaksanakan proses kuliah dan magang hingga pembuatan laporan ini dengan baik. Shalawat serta salam senantiasa penulis haturkan kepada Nabi Muhammad SAW yang telah membawa umat manusia dari zaman kebodohan hingga zaman yang terang benderang dengan Islam dan ilmu pengetahuan.

Laporan ini disusun untuk memenuhi persyaratan tugas penjurusan magang di Fakultas Teknologi Industri Jurusan Informatika Universitas Islam Indonesia. Penulis menyusun laporan ini tidak lepas dari bimbingan dan arahan berbagai pihak. Penulis mengucapkan rasa hormat dan terima kasih kepada semua pihak yang telah membantu. Adapun pihak-pihak yang terkait diantaranya sebagai berikut:

1. Kedua orang tua tercinta yang selalu mendukung dan memberikan doa selama kegiatan magang berlangsung.
2. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Teknik Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia
3. Bapak Andhik Budi Cahyono, S.T., M.T. selaku Dosen Pembimbing yang telah bersedia membimbing dan mengarahkan penyusunan laporan ini.
4. Bapak Ganjar Widiatmasyah selaku *Engineering Manager* yang telah menerima dan membantu penulis selama bekerja
5. Bapak Faris Saeful Ilmi selaku *Product Manager* yang telah membantu penulis selama bekerja
6. Saudara Faridhony M Zain selaku *Supervisor* dan mentor yang telah membimbing penulis dengan baik
7. Saudari Maria Gabriel selaku *Technical Leader* tim R&D dan mentor yang telah membimbing penulis dengan baik.

Yogyakarta, 16 Desember 2022



Ibnu Ahmad Kamal

## SARI

Qiscus Multichannel Chat merupakan produk dari Qiscus yang dapat membantu bisnis dalam mensentralisasikan percakapan dari beberapa *channel* media sosial. Qiscus Multichannel memiliki beberapa *add-on* yang dapat mendukung dalam menciptakan layanan *customer experience* yang baik. Salah satu *add-on* yang menjadi produk unggulan adalah Robolabs. Robolabs merupakan sebuah *add-on chatbot builder* yang dikembangkan di atas teknologi milik Google yang bernama Dialogflow. Chatbot yang dibuat menggunakan Robolabs akan memiliki fungsi utama untuk melakukan interaksi antara manusia dengan bot. fungsi utama tersebut terkadang masih dianggap kurang karena Dialogflow belum memiliki fungsionalitas dalam menyimpan data dari *chat* yang masuk. Untuk menambahkan fungsionalitas penyimpanan data dari *chat*, Dialogflow sudah menyediakan mekanisme yaitu dengan cara menambahkan *web intermediary* berjenis *webhook*. Untuk membuat sebuah *chatbot* pada Dialogflow yang memiliki fungsionalitas penyimpanan, maka dilakukan proyek implementasi Google Sheet API pada *custom webhook*. Chatbot yang dibuat akan menggunakan studi kasus pelayanan komplain pada sebuah *e-commerce*. Penyimpanan yang digunakan pada proyek ini menggunakan Google Sheet. Pemilihan Google Sheet sebagai tempat penyimpanan data komplain dikarenakan tim pengembang berpendapat Google Sheet lebih mudah digunakan untuk orang awam. Tujuan dari proyek tersebut adalah untuk menghasilkan kode program yang memanfaatkan Google Sheet pada *webhook chatbot* berbasis Dialogflow. Hasil implementasi membuktikan bahwa Google Sheet dapat digunakan sebagai tempat penyimpanan data komplain dari *chatbot*, akan tetapi memiliki beberapa hal yang perlu diperhatikan terutama pada waktu *respon time* *webhook*. *Webhook* yang dibuat menggunakan *library* Google Sheet API memiliki *respon time* antara 3-4 detik. Hal ini dapat mempengaruhi keberhasilan pemberian respon *chatbot*.

Kata kunci: Qiscus Multichannel Chat, Robolabs, Dialogflow, Data Komplain, Google Sheet API



## GLOSARIUM

Webhook	Metode komunikasi antar software berbasis pada sebuah event
Intent	Kategori topik pembicaraan pada Dialogflow
Intent Matching	Proses pemetaan input ke dalam sebuah topik pembicaraan pada Dialogflow
Qiscus <i>Multichannel Chat</i>	<i>Platform multi channel</i> yang memungkinkan bisnis untuk mengintegrasikan berbagai <i>channel</i> media sosial ke dalam satu <i>dashboard</i>
Dialogflow	<i>Chatbot Builder Framework</i> milik Google yang memiliki fungsionalitas untuk membuat sebuah <i>chatbot</i>



## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING .....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI .....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR.....	vii
SARI.....	viii
GLOSARIUM .....	ix
DAFTAR ISI .....	x
DAFTAR TABEL .....	xii
DAFTAR GAMBAR.....	xiii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Ruang Lingkup.....	3
1.3 Tujuan .....	3
1.4 Manfaat .....	3
1.5 Sistematika Penulisan .....	4
<b>BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1 <i>Chatbot</i> .....	5
2.2 <i>Dialogflow</i> .....	5
2.3 <i>Web Intermediary</i> .....	6
2.4 <i>Qiscus Multichannel Chat</i> .....	7
2.5 <i>Qiscus Robolabs</i> .....	7
2.6 <i>Tinjauan Pustaka</i> .....	8
<b>BAB III PELAKSANAAN MAGANG .....</b>	<b>10</b>
3.1 <i>Aktivitas Magang</i> .....	10
3.1.1 <i>Pengenalan Lingkungan Kerja Perusahaan</i> .....	10
3.1.2 <i>Mempelajari Alur Kerja Produk Robolabs</i> .....	11
3.1.3 <i>Mempelajari Wit.ai Service</i> .....	12
3.1.4 <i>Proyek Pengembangan Handover Agent 2.0</i> .....	12
3.1.5 <i>Pengembangan Fitur Auto Tag</i> .....	13

3.1.6	Melakukan <i>Weekly Meeting</i> .....	13
3.2	Pelaksanaan Proyek Implementasi Google Sheet API pada Custom Webhook ..	13
3.2.1	Perancangan Conversational Flow .....	14
3.2.2	Pembuatan <i>Chatbot</i> .....	16
3.2.3	Pembuatan Webhook.....	21
3.2.4	Pengujian <i>Chatbot</i> .....	35
3.3	Penutupan Proyek Implementasi Google Sheet API pada Custom Webhook .....	40
BAB IV REFLEKSI PELAKSANAAN MAGANG .....		41
4.1	Relevansi Akademik .....	41
4.2	Pembelajaran Magang.....	41
4.2.1	Manfaat Magang.....	41
4.2.2	Kendala, Hambatan dan Tantangan.....	42
BAB V PENUTUP.....		43
5.1	Kesimpulan .....	43
5.2	Saran.....	43
DAFTAR PUSTAKA.....		45
LAMPIRAN .....		46



**DAFTAR TABEL**

Tabel 2.1 Perbandingan Tinjauan Pustaka.....	9
Tabel 3.1 Aktivitas magang .....	10
Tabel 3.2 Daftar Intent Serta Parameter .....	14
Tabel 3.3 Daftar Method Google Sheet API yang Digunakan pada Webhook .....	23
Tabel 3.4 Daftar Method Google Sheet API yang Tidak Digunakan pada Webhook.....	23
Tabel 3.5 Daftar <i>Environment Variable</i> .....	33



## DAFTAR GAMBAR

Gambar 1.1 Alur Kerja Qiscus <i>Multichannel Chat</i> Menggunakan <i>Add-on</i> Robolabs .....	2
Gambar 2.1 Alur Proses Dialogflow .....	6
Gambar 3.1 Daftar Template pada Fitur Bot Template .....	11
Gambar 3.2 Alur Kerja <i>Chatbot</i> Menggunakan Webhook .....	16
Gambar 3.3 Tampilan Agent Pada Dialogflow .....	17
Gambar 3.4 Tampilan Intent dengan Parameter .....	18
Gambar 3.5 Tampilan Respon <i>Chatbot</i> Komplain Tidak Sesuai Format .....	19
Gambar 3.6 Tampilan Alur Percakapan Pengajuan Komplain .....	20
Gambar 3.7 Tampilan Respon Menampilkan Status Komplain .....	21
Gambar 3.8 Tampilan Menu Keys pada Google Cloud Platform .....	22
Gambar 3.9 Tampilan Fitur API&Services .....	22
Gambar 3.10 Tampilan Hak Akses pada Google Sheet .....	23
Gambar 3.11 Tampilan Struktur Folder Webhook .....	24
Gambar 3.12 Tampilan Kode Inisiasi <i>Service</i> .....	25
Gambar 3.13 Tampilan Kode Method Get .....	26
Gambar 3.14 Tampilan Kode Pembuatan Header pada Datframe .....	26
Gambar 3.15 Alur Kerja <i>Function</i> <i>Write_data</i> .....	27
Gambar 3.16 Tampilan Kode Pengecekan Format Pengajuan komplain .....	28
Gambar 3.17 Tampilan Kode Konversi Data Bertipe List Menjadi Dataframe .....	28
Gambar 3.18 Tampilan Kode Penulisan Data Menggunakan Method <i>Append</i> .....	29
Gambar 3.19 Tampilan Kode Mendapatkan Nilai Index .....	29
Gambar 3.20 Tampilan Kode Penulisan Data dengan <i>Session_Id</i> Bernilai Null .....	30
Gambar 3.21 Tampilan Kode Penulisan Data dengan <i>Session_Id</i> Bernilai Tidak Null .....	31
Gambar 3.22 Tampilan Kode Function <i>filter_data_by_ID</i> .....	31
Gambar 3.23 Tampilan Kode Route Webhook .....	32
Gambar 3.24 Tampilan Kode pada File Procfile .....	33
Gambar 3.25 Tampilan Fitur Fulfillment pada Dialogflow .....	34
Gambar 3.26 Tampilan Dokumen Research Notes .....	34
Gambar 3.27 Tampilan Repository Code .....	35
Gambar 3.28 Tampilan Menu Utama .....	36
Gambar 3.29 Tampilan Respon Intent <i>Purchased_item</i> .....	36
Gambar 3.30 Tampilan Respon Intent <i>Wrong_item</i> .....	37

Gambar 3.31 Tampilan Respon Data Berhasil Disimpan .....	37
Gambar 3.32 Tampilan Respon Input Tidak Sesuai Format.....	38
Gambar 3.33 Tampilan Respon Intent <i>complain_status</i> .....	38
Gambar 3.34 Tampilan Respon Id Komplain Ditemukan .....	39
Gambar 3.35 Tampilan Tabel Pada Google Sheet.....	40



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Qiscus merupakan perusahaan *B2B (Business to Business)* yang menyediakan layanan *In-App Real Time Communications (RTC)* untuk membantu perusahaan berskala besar atau kecil dalam mengelola layanan *communication services*. Qiscus didirikan pada tahun 2013 di Singapura oleh tiga orang pendiri. Pada mulanya Qiscus mengembangkan fitur *chat* pada proyek Phi Decks. Phi Decks merupakan *Learner Engagement System (LES)* yang membantu pengajar untuk melakukan aktivitas pembelajaran yang efektif. Proyek Phi decks dihentikan pada tahun 2013 tetapi fitur *chat* dilanjutkan pengembangannya sehingga menjadi produk utama Qiscus yang bernama Qiscus Multichannel Chat.

Qiscus Multichannel Chat merupakan sebuah platform yang dapat membantu bisnis untuk mensentralisasikan percakapan dari beberapa *channel* sosial media ke dalam satu platform. Selain memiliki fungsi utama dalam mensentralisasikan *chat* dari beragam *channel*. Qiscus Multichannel Chat juga memiliki beberapa *add-on* yang mendukung untuk menciptakan *Customer Experience* yang baik.

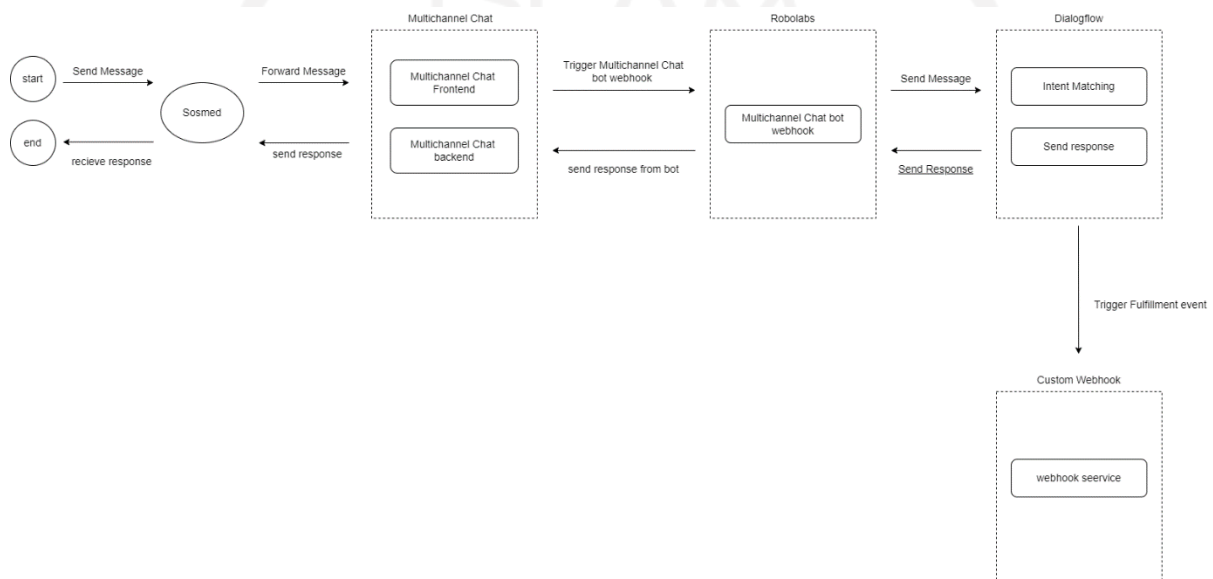
Salah satu *add-on* yang menjadi unggulan Qiscus Multichannel Chat yaitu Robolabs. Robolabs merupakan layanan *add-on chatbot builder* pada produk Qiscus Multichannel Chat. Layanan ini dikembangkan untuk membantu *client* dalam membuat *chatbot* sendiri yang terintegrasi dengan platform Qiscus Multichannel Chat. Tujuan dikembangkan Robolabs agar mempermudah *client* dalam membuat *chatbot* sendiri.

Robolabs memberikan kemudahan untuk client dalam meng-*handle* percakapan yang masuk melalui beberapa *platform chat*, selain sebagai *add-on chatbot builder* Robolabs juga memiliki fitur Handover Agent. Fitur Handover Agent berfungsi untuk mengalihkan percakapan dari *chatbot* menuju *agent* manusia jika *chat* yang masuk perlu penanganan dari manusia. Robolabs dikembangkan di atas teknologi *chatbot builder* yang bernama Dialogflow. Dialogflow merupakan platform *chatbot builder* yang dikembangkan oleh Google untuk membuat *chatbot*. Dialogflow dikembangkan menggunakan *Natural language processing (NLP)* dan *Natural language understanding (NLU)*.

Sebagai upaya untuk meningkatkan produk Robolabs, dibuatlah beberapa proyek pengembangan Robolabs. Selama melakukan magang di Qiscus terdapat dua proyek yang diikuti yaitu proyek implementasi Google sheet API pada custom webhook dan proyek

pengembangan Handover Agent 2.0. Google sheet API merupakan RESTful interface yang disediakan oleh Google untuk membuat, membaca, memutakhirkan dan mengelola data pada sheet. Google sheet API menyediakan beberapa bahasa pemrograman yang dapat digunakan oleh developer untuk pengembangan perangkat lunak.

Proyek pertama merupakan proyek pengembangan pada sisi custom webhook. Custom webhook akan dijalankan setelah Robolabs berhasil membuat dan mengkonfigurasi *chatbot* pada Dialogflow. Gambar 1.1 merupakan alur kerja *chat* yang masuk kemudian di-handle oleh Qiscus Multichannel Chat hingga custom webhook dijalankan.



Gambar 1.1 Alur Kerja Qiscus Multichannel Chat Menggunakan Add-on Robolabs

Alur kerja dimulai ketika terdapat *chat* yang masuk melalui media sosial yang terintegrasi dengan Qiscus Multichannel Chat. Pesan text akan di forward ke Qiscus Multichannel Chat yang akan men-trigger webhook Multichannel Chat Bot webhook pada Robolabs. Roblobas akan meneruskan pesan *text* ke Dialogflow yang kemudian akan diproses dengan melakukan *intent matching*. Dialogflow akan mentrigger custom webhook dan mengirimkan respon ke Robolabs. Proses selanjutnya akan diteruskan hingga *chat* dapat diterima oleh user yang berinteraksi dengan *chatbot* melalui media sosial yang digunakan.

Latar belakang dilakukan proyek pertama dikarenakan pihak Qiscus sering menjumpai pertanyaan dari client mengenai kemungkinan untuk menyimpan data dari respon konsumen. Google Sheet dipilih karena tim pengembang menilai *client* sudah familiar menggunakan aplikasi organisasi dan menyimpan data dalam bentuk tabel. *Library* yang digunakan pada



proyek ini adalah Google sheet API. Pengembangan webhook untuk *chatbot* menggunakan bahasa pemrograman Python dengan framework FAST API.

## 1.2 Ruang Lingkup

Kegiatan magang sebagai *Backend Engineer* di Qiscus dilakukan di tim *R&D*. tim *R&D* merupakan tim yang mengembangkan Robolabs. Kegiatan magang telah dilaksanakan selama enam bulan mulai dari September 2021 hingga Februari 2022. Kegiatan magang dilakukan secara *WFH (Work From Home)*. Selain sebagai tim yang menangani produk Robolabs, tim *R&D* juga memiliki peran untuk melakukan riset mengenai pengembangan produk baru atau penambahan fitur pada Robolabs.

Proyek pada tim *R&D* memiliki dua kategori, kategori yang pertama merupakan proyek berjenis penelitian yang menghasilkan sebuah purwarupa sistem atau dokumentasi *Research Note's*. Kategori yang kedua merupakan proyek berjenis pengembangan. Proyek jenis ini bertujuan untuk menambahkan sebuah fitur pada produk yang dipasarkan. Selama melakukan magang di Qiscus terdapat beberapa proyek dan pekerjaan yang sudah dilaksanakan, adapun proyek dan pekerjaan yang dikerjakan selama periode magang adalah sebagai berikut:

- a. Membuat Bot Template untuk studi kasus *food&Beverage* dan *Complaint*
- b. Mempelajari *wit.ai* service
- c. Melakukan Proyek Implementasi Google Sheet API pada Custom Webhook
- d. Melakukan Proyek Pengembangan Handover Agent 2.0 pada Robolabs
- e. Melakukan Pengembangan Fitur *Auto Tag* pada Robolabs.

## 1.3 Tujuan

Tujuan dari proyek yang dikerjakan pada magang ini adalah::

- a. Mengimplementasi Google Sheet API pada custom webhook untuk menambahkan fungsionalitas penyimpanan pada chatbot berbasis Dialogflow
- b. Mendapatkan batasan dan cara penggunaan *library* Google sheet API.

## 1.4 Manfaat

Tujuan dari proyek yang dikerjakan pada magang ini adalah:

- a. Hasil implementasi kode dapat digunakan sebagai acuan apabila tim pengembang memiliki rencana untuk menggunakan *Library* Sheet API pada sebuah fitur atau produk.
- b. Hasil laporan implementasi dapat digunakan sebagai pertimbangan ketika menambahkan fitur yang menggunakan teknologi Google Sheet sebagai basis data.

### 1.5 Sistematika Penulisan

Setiap materi yang tertera pada laporan ini dikelompokkan dalam beberapa bab. Adapun sistematika penyampaian bab adalah sebagai berikut:

a. BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, ruang lingkup magang, tujuan dan manfaat penelitian, dan sistematika penulisan.

b. BAB II DASAR TEORI

Bab ini berisi teori yang berupa pengertian dan definisi yang diambil dari media internet dan buku yang berkaitan dengan penyusunan laporan serta beberapa literature.

c. BAB III PELAKSANAAN MAGANG

Bab ini berisi dokumentasi dari kegiatan dan tugas yang dilaksanakan selama menjalani proses magang.

d. BAB IV REFLEKSI PELAKSANAAN MAGANG

Bab ini berisi penjelasan hasil dan refleksi yang didapatkan selama melakukan aktivitas magang.

e. BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran dari seluruh materi yang telah dijelaskan pada bab-bab sebelumnya

## BAB II

### LANDASAN TEORI DAN TINJAUAN PUSTAKA

#### 2.1 Chatbot

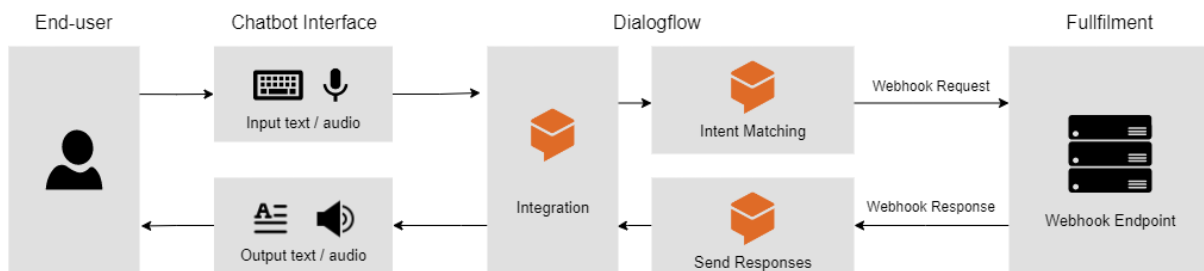
*Chatbot* adalah teknologi intelegensia buatan berbasis teks atau audio yang memiliki kemampuan untuk melakukan interaksi antara mesin dan manusia, respon *chatbot* akan berbeda sesuai dengan *input* yang diberikan. *Chatbot* memiliki dua bagian yaitu program utama (bot) dan *brain file*. Program utama memiliki fungsi untuk menerima *input* dari *user* dan melakukan parsing ke *brain file*. *Brain file* merupakan otak dari *chatbot* yang akan melakukan analisa dan menentukan respon yang diberikan (Oktavia, 2020).

Beberapa penelitian telah dilakukan mengenai pengembangan *chatbot* pada layanan customer service toko online atau *e-commerce*. Pengembangan tersebut bertujuan untuk menciptakan layanan customer service yang responsif. Seperti pada makalah yang ditulis oleh (Dwi et al., 2018) pada makalah tersebut dilakukan pengembangan *chatbot* customer service yang terintegrasi dengan web CMS. Hasil penelitian memberikan kesimpulan bahwa *chatbot* dapat melakukan tugas layanan customer service seperti interaksi tanya jawab dan melakukan pemesanan.

#### 2.2 Dialogflow

Dialogflow adalah platform *chatbot builder* yang dikembangkan oleh Google. Dialogflow dikembangkan menggunakan *Natural Language Processing (NLP)* dan *Natural Language Understanding (NLU)*. layanan Dialogflow berfokus untuk membuat *chatbot* yang dapat melakukan interaksi antara manusia dengan mesin menggunakan bahasa alami *NLP* dan analisis semantik menggunakan *NLU* (Komawar et al., 2013).

Alur proses Dialogflow dimulai ketika *end-user* mengirimkan *input* berupa teks atau audio melalui *chat interface* yang terintegrasi dengan Dialogflow. *Chat interface* merupakan platform yang digunakan oleh *chatbot* untuk berinteraksi dengan *end-user*. Media sosial merupakan contoh *chat interface* yang sering digunakan. *Input* yang masuk kemudian akan dianalisa dengan mencari *training phrase* yang mirip pada suatu *intent* (kategori topik), setelah menemukan *intent* yang sesuai dialogflow akan mengirimkan *trigger event* pada webhook endpoint apabila fitur Fulfillment diaktifkan. Webhook akan mengirimkan respon ke Dialogflow dan diteruskan ke *end-user* melalui *chat interface*. Jika fitur fulfillment tidak diaktifkan Dialogflow akan mengirimkan respon yang sudah ditentukan oleh developer. Visualisasi alur proses Dialogflow dapat dilihat pada Gambar 2.1.



Gambar 2.1 Alur Proses Dialogflow

### 2.3 Web Intermediary

*Web intermediary* merupakan sebuah *software* atau *web service* yang dijalankan pada sebuah server. *Web intermediary* berfungsi untuk membantu komunikasi antara dua aplikasi atau *client* dengan server (Anonim, 2011). *Web intermediary* memiliki beragam jenis salah satunya adalah *webhook*. *Web intermediary* yang dibuat pada proyek Implementasi Google Sheet API pada Custom Webhook menggunakan tiga teknologi yaitu: Google Sheet API, Google Sheet dan Webhook.

Google Sheet API adalah RESTful interface yang disediakan oleh Google untuk membuat, membaca, memutakhirkan dan mengelola data pada Google Sheet. Penggunaan Google Sheet API dapat melalui endpoint RESTful API atau *library*. *Library* Google Sheet API tersedia pada beberapa bahasa pemrograman yang dapat digunakan untuk mengembangkan perangkat lunak (Google, n.d.).

Google sheet adalah aplikasi web yang berfungsi untuk organisasi analisis dan penyimpanan data dalam bentuk tabel. Google Sheet merupakan aplikasi milik google yang memungkinkan pengguna untuk menggunakannya secara real-time. Aplikasi ini memerlukan akses internet untuk penggunaannya. Google Sheet dapat digunakan sebagai basis data yang mudah untuk melakukan editing data.

Webhook merupakan salah satu metode komunikasi yang digunakan pada dialogflow. *webhook* digunakan untuk komunikasi antara satu aplikasi dengan lainnya secara real time menggunakan *trigger event* (Turner, 2022). Webhook serupa dengan RESTful API yang menggunakan endpoint sebagai media komunikasinya, perbedaan antara keduanya terdapat pada saat proses berjalannya. RESTful API menggunakan proses *request-based* sedangkan *webhook* menggunakan *event-based*.

## 2.4 Qiscus Multichannel Chat

Qiscus Multichannel Chat merupakan *platform* yang mengintegrasikan beberapa *channel* media sosial ke dalam satu *dashboard*. Qiscus Multichannel Chat telah mendukung beberapa *channel* media sosial untuk diintegrasikan diantaranya: Whatsapp, LINE, Facebook Messenger, Telegram, Email, Twitter, Instagram, dan Qiscus Widget. Qiscus Multichannel Chat memiliki tiga fitur yang dapat membantu bisnis dalam meningkatkan layanan *customer service* (Qiscus, n.d.-a).

Fitur pertama adalah integrasi *channel* media sosial pada *dashboard* Qiscus Multichannel Chat, kemudian fitur kedua adalah fitur *filtering* yang bisa digunakan untuk memfilter pesan yang masuk dari *channel* media sosial tertentu. Fitur yang terakhir yaitu fitur manajemen *human agent*, fitur ini akan membantu dalam mendistribusikan *human agent* untuk menangani pesan masuk.

## 2.5 Qiscus Robolabs

Qiscus Robolabs merupakan *add-on* dari Qiscus Multichannel Chat, yang berfungsi sebagai *chatbot builder* dan integrator antara *bot engine* dengan *dashboard* Qiscus Multichannel Chat. Qiscus Robolabs memudahkan *client* untuk membuat chatbot, yang kemudian secara otomatis terintegrasi dengan akun Multichannel Chat. Hingga saat ini, Qiscus Robolabs telah terintegrasi dengan platform NLP Google, yaitu Dialogflow (Qiscus, n.d.-b).

Qiscus Robolabs menyediakan beragam fitur yang dapat membantu *client* dalam melakukan konfigurasi *chatbot* yang dibuat, adapun fitur yang dimiliki Qiscus Robolabs adalah sebagai berikut:

### a. Bot Template

Fitur ini dapat digunakan dengan mendownload beberapa *template* yang sudah disediakan dalam format excel atau csv. File template dapat diubah sesuai kebutuhan dan diunggah kembali.

### b. Channel Routing

Fitur ini membuat *chatbot* dapat diintegrasikan pada saluran yang berbeda sesuai dengan kebutuhan.

### c. Export and Import Project

Fitur ini digunakan untuk mengunggah atau mengunduh *chatbot* yang telah dibuat.

### d. Handover Agent.

Fitur ini berfungsi untuk membuat rute percakapan dari *chatbot* ke *human agent* sehingga pengguna dapat menentukan intent atau percakapan mana yang dapat diteruskan ke *human agent*.

e. Training Bot

Fitur ini berfungsi untuk melatih *chatbot* yang dibuat.

f. Auto Resolve Room

Fitur ini berfungsi untuk melakukan *resolve* ruang percakapan dengan customer secara otomatis. Ketika *customer* tidak membalas pesan dari *chatbot* selama 5 menit, *chatbot* akan mengirim *follow up message* untuk berinteraksi kembali dengan pengguna akhir. Apabila dalam 5 menit tidak ada respon apapun dari *customer*, maka *chatbot* akan mengirimkan *end follow up message* dan mengakhiri percakapan dengan *customer* tersebut.

g. Intent

Fitur ini terhubung ke Dialogflow, sehingga perubahan yang dibuat di Robolabs juga akan muncul di Dialogflow.

h. Send to External

Fitur ini memungkinkan pengguna untuk menghubungkan sistem mereka melalui webhook maupun *custom API* dengan Qiscus Robolabs System

i. Analitik

Salah satu fitur terpenting yang dibutuhkan oleh bisnis. Dengan fitur ini, *client* akan dapat meninjau metrik penggunaan *chatbot*, seperti jumlah percakapan yang dapat ditangani *chatbot* dan percakapan mana yang telah diteruskan ke *human agent*.

## 2.6 Tinjauan Pustaka

Subbab ini berisikan keterkaitan proyek dengan beberapa penelitian terdahulu yang menggunakan *chatbot* untuk menangani layanan *customer service*. Adapun beberapa penelitian terdahulu adalah sebagai berikut:

Penelitian yang dilakukan oleh Ananda Dwi R,dkk membahas permasalahan yang dihadapi oleh pelaku UKM Minsu dalam *customer service* terutama pada pelayanan pemesanan. Solusi yang diberikan pada penelitian tersebut adalah *chatbot* yang terintegrasi dengan CMS berbasis web yang berfungsi untuk mengelola pesanan yang masuk. Adapun hasil dari penelitian tersebut adalah *chatbot* yang terintegrasi dengan CMS berbasis web.

Penelitian lain dilakukan oleh Chaulina Alfianti Oktavia. penelitian tersebut bertujuan untuk membuat *Chatbot* berbasis Dialogflow dengan studi kasus layanan *customer service*

pada *e-commerce*. kesimpulan penelitian tersebut adalah *chatbot* yang dibuat menggunakan Dialogflow dapat berjalan dengan baik dalam menangani *chat* yang masuk. Untuk mengetahui perbandingan beberapa penelitian terdahulu dengan proyek saat ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan Tinjauan Pustaka

Judul	Bahasa Pemrograman	<i>Chatbot</i> Builder Platform	Hasil
Aplikasi <i>Chatbot</i> (Milki Bot) yang Terintegrasi Dengan Web Cms untuk Customer Service pada UKM Minsu	Python dan PHP		<i>Chatbot</i> yang terintegrasi dengan CMS berbasis web
Implementasi <i>Chatbot</i> Menggunakan Dialogflow dan Messenger Untuk Layanan Customer Service Pada E-Commerce		Dialogflow	<i>Chatbot</i> dengan studi kasus layanan customer service pada e-commerce
Proyek Implementasi Google Sheet API pada Custom Webhook	Python	Dialogflow	<i>Chatbot</i> dengan studi kasus pelayanan komplain pada e-commerce yang terintegrasi dengan Google Sheet

Berdasarkan pada Tabel 2.1 proyek yang dikerjakan memiliki kesamaan dengan penelitian 1 dan 2. Kesamaan tersebut ialah *chatbot* berfungsi untuk menyelesaikan permasalahan customer service, selain itu terdapat kesamaan penggunaan *chatbot builder* antara penelitian 2 dengan proyek yang dikerjakan. Persamaan tersebut dapat digunakan untuk pembelajaran pada proyek yang dikerjakan. Proyek yang dikerjakan merupakan gabungan ide dari dua penelitian terdahulu dengan menggunakan beberapa *tools* yang berbeda.

## BAB III PELAKSANAAN MAGANG

### 3.1 Aktivitas Magang

Kegiatan magang dilakukan selama kurang lebih enam bulan dimulai pada awal september 2021 hingga februari 2022. Terdapat beberapa aktivitas dan proyek yang telah dikerjakan selama kurun waktu tersebut. Aktivitas-aktivitas dan proyek yang dikerjakan dapat dilihat pada Tabel 3.1. Penjelasan terkait detail aktivitas yang dilakukan pada Tabel 3.1 akan dijelaskan pada subbab berikutnya.

Tabel 3.1 Aktivitas magang

No	Aktivitas	Waktu Mulai	Durasi
1	Pengenalan lingkungan kerja perusahaan	7 september 2021	3 hari
2	Mempelajari alur kerja produk Robolabs	8 september 2021	1.5 bulan
3	Mempelajari <i>wit.ai service</i>	18 Oktober 2021	1.5 bulan
4	Mengerjakan proyek implementasi Google sheet API pada <i>custom webhook</i>	6 Desember 2021	1 bulan
5	Mengerjakan proyek pengembangan <i>Handover Agent 2.0</i>	10 Januari 2022	1 bulan
6	Pengembangan fitur <i>Auto Tag</i>	7 Februari 2022	2 Minggu
7	Melakukan <i>weekly meeting</i>		Seminggu sekali

#### 3.1.1 Pengenalan Lingkungan Kerja Perusahaan

Aktivitas pengenalan lingkungan kerja dilakukan selama tiga hari yang terdiri dari dua sesi. Sesi pertama akan disampaikan oleh tim *HRD (Human Resource Development)* dan sesi selanjutnya akan disampaikan oleh *PM (Project Manager)* dari tim *R&D*. Materi pada sesi pertama membahas mengenai kode etik perusahaan. Pada sesi kedua materi yang disampaikan membahas produk yang dikembangkan oleh tim *R&D* dan *tools* apa saja yang digunakan selama melakukan magang. selain membahas *tools* yang digunakan *Project Manager* juga menyampaikan informasi mengenai *supervisor* yang akan mengawasi pekerjaan selama melakukan magang di Qiscus. Adapun *tools* yang digunakan selama melakukan magang yaitu:



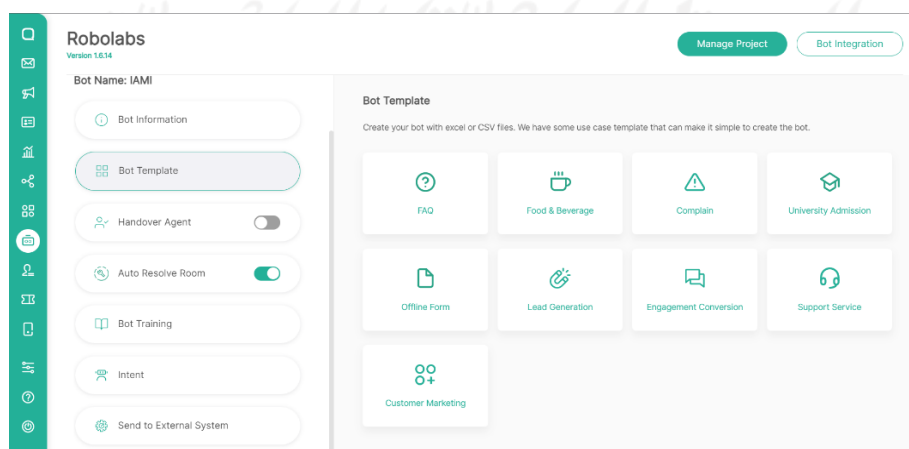
1. Visual Studio Code sebagai *code editor* yang digunakan selama melakukan magang
2. Python sebagai bahasa pemrograman untuk mengembangkan produk Robolabs
3. Dialogflow merupakan *chatbot builder framework* yang digunakan pada produk Robolabs
4. Slack sebagai alat komunikasi utama
5. Trello digunakan sebagai *tools* untuk manajemen proyek
6. Bitbucket sebagai repository penyimpanan kode program

### 3.1.2 Mempelajari Alur Kerja Produk Robolabs

Robolabs merupakan sebuah *add-on* pada Qiscus Multichannel Chat yang dikembangkan oleh tim R&D, *add-on* ini digunakan untuk membuat *chatbot* yang akan digunakan pada setiap *channel* sosial media yang terintegrasikan dengan Qiscus Multichannel Chat. Selain sebagai *chatbot builder* Robolabs memiliki fungsi untuk melakukan konfigurasi *chatbot* yang telah dibuat. Robolabs sendiri dibangun di atas *framework chatbot builder* milik Google yaitu Dialogflow.

*Supervisor* memberikan waktu selama satu bulan setengah untuk mempelajari fitur dan alur kerja Robolabs dari dokumen dan struktur program yang diberikan. selama kurun waktu tersebut *supervisor* juga akan memberikan sebuah *task* atau pekerjaan yang akan dikerjakan dan di implementasi ke dalam Robolabs. Adapun *task* yang diberikan oleh *supervisor* selama aktivitas ini meliputi pembuatan dua template alur percakapan *chatbot* dengan kasus *complain* dan *food&beverage*. Kedua template tersebut akan diimplementasi ke dalam Robolabs.

*Bot Template* merupakan sebuah fitur pada Robolabs yang berisikan beberapa template alur percakapan *chabot* dengan kasus tertentu. Template tersebut dapat langsung digunakan pada *chatbot* yang akan dibuat. Tampilan daftar template alur percakapan pada fitur *Bot Template* dapat dilihat pada Gambar 3.1.



Gambar 3.1 Daftar Template pada Fitur Bot Template

Alur kerja Bot Template dimulai dengan user memilih template alur percakapan yang akan dibuat, kemudian user mendownload file berekstensi csv atau excel. File yang terdownload akan diupload pada kolom yang tersedia. Penerapan alur percakapan *chatbot* akan dilakukan apabila user menyimpan perubahan.

### **3.1.3 Mempelajari Wit.ai Service**

Setelah memahami alur kerja dan fitur Robolabs, kegiatan magang dilanjutkan dengan mempelajari Wit.ai *service*. Kegiatan ini dilakukan karena hasil diskusi internal tim yang membahas untuk penambahan *framework chatbot builder* pada Robolabs. kegiatan mempelajari Wit.ai *service* bertujuan untuk mendapatkan informasi mengenai alur kerja pembuatan *chatbot* pada Wit.ai *service* dan batasan yang dimiliki.

Pelaksanaan kegiatan ini didampingi oleh supervisor sebagai pemberi arahan dan pengambilan keputusan dalam penambahan Wit.ai *service* pada Robolabs. Hasil dari kegiatan ini berupa dokumen penjelasan alur kerja sistem Wit.ai *service* beserta daftar batasan dan Endpoint API publik yang dapat digunakan.

### **3.1.4 Proyek Implementasi Google Sheet API pada Custom Webhook**

Proyek implementasi Google sheet API pada custom webhook merupakan sebuah proyek pengembangan webhook untuk *chatbot* berbasis Dialogflow. Webhook yang dibuat memiliki fungsionalitas untuk menyimpan data dari *chat* yang masuk ke Google Sheet API. Tujuan dari proyek ini adalah mendapatkan gambaran teknis mengenai penggunaan Google Sheet sebagai *database* pada *chatbot*.

Proyek ini, merupakan proyek jenis penelitian yang dilakukan oleh peserta magang. Hasil dari proyek ini berupa dokumentasi teknis baik berupa kode program atau research notes. Dengan adanya proyek ini diharapkan hasil yang didapat dapat digunakan sebagai dasar untuk pengembangan Qiscus Robolabs kedepannya. Tugas yang diberikan pada proyek ini adalah sebagai *backend engineer*.

### **3.1.4 Proyek Pengembangan Handover Agent 2.0**

Proyek selanjutnya yang diikuti oleh peserta magang adalah proyek pengembangan Handover Agent 2.0. Handover Agent merupakan sebuah fitur pada Roboloabs yang memiliki fungsi untuk melakukan konfigurasi pengalihan percakapan dari *chatbot* ke *Human Agent*. Fitur ini dibuat dengan tujuan untuk menangani kasus ketika diperlukannya *human action*. Adapun menu yang ditambahkan pada versi 2.0 yaitu: *Office Hours Message* dan *Find Online Message*.

*Office Hours Message* merupakan sebuah menu yang berfungsi untuk melakukan konfigurasi pesan peringatan. Pesan peringatan yang dikirimkan dapat berupa informasi jam kerja agent. Pesan peringatan tersebut akan dikirimkan oleh *chatbot* apabila terdapat *chat* yang masuk ke Qiscus Multichannel Chat diluar jam kerja agent.

Menu lainnya yang ditambahkan pada menu ini adalah *Find Online Message*. Menu ini berfungsi untuk melakukan konfigurasi pesan peringatan jika tidak ditemukannya agent yang online. Pesan tersebut akan diberikan oleh *chatbot* apabila tidak ditemukannya agent dengan status online ketika pengalihan percakapan dari *chatbot* ke *Human Agent*.

### **3.1.5 Pengembangan Fitur *Auto Tag***

Pengembangan terakhir yang dilakukan selama kegiatan magang adalah pembuatan fitur *Auto Tag* pada Robolabs. *Auto Tag* merupakan sebuah fitur tambahan yang digunakan untuk membuat tag secara otomatis pada *room chat* Qiscus Multichannel Chat. Tag akan dibuat ketika *chat* yang masuk terkategori sebagai sebuah intent yang memiliki konfigurasi tag yang akan dibuat pada fitur Manage Intent di Robolabs. Proses pengembangan dilakukan selama dua minggu.

### **3.1.6 Melakukan *Weekly Meeting***

Selama seminggu sekali seluruh anggota tim R&D melakukan meeting mingguan. Meeting ini dilakukan setiap hari jumat, pada meeting mingguan tiap orang akan menyampaikan progress dan kendala yang dihadapi dalam mengerjakan task atau proyek selama satu minggu. Pada saat penyampaian progress dan kendala anggota tim lain dapat membuka diskusi untuk menyelesaikan kendala yang didapat. Meeting mingguan diakhiri dengan pemberian task atau pekerjaan yang akan dilakukan untuk satu minggu ke depan.

## **3.2 Pelaksanaan Proyek Implementasi Google Sheet API pada Custom Webhook**

Proyek dimulai dengan melakukan perancangan *conversational flow* dengan studi kasus pelayanan komplain pada *e-commerce*. *Conversational flow* merupakan alur percakapan pada *chatbot*, *conversational flow* berguna untuk memberikan aturan agar percakapan antara *chatbot* dan pengguna dapat terjadi.

*Conversational flow* yang digunakan merupakan hasil dari analisis kebutuhan dari beberapa contoh *chatbot* yang menggunakan kasus serupa. *Chatbot* yang digunakan pada proyek ini akan menggunakan *conversational flow* dengan *style menu base*. *Style menu base* merupakan *style conversational flow* yang menggunakan daftar menu sebagai rujukan *input* pengguna. *Style menu base* dipilih karena dapat mengurangi kesalahan *chatbot* dalam memberikan respon. Selain untuk mengurangi kesalahan, style ini juga menjadi guideline umum

yang digunakan oleh tim pengembang Qiscus Robolabs untuk membuat sebuah chatbot. Untuk melakukan interaksi pertama kali dengan chatbot kata “hai” atau “hello” digunakan sebagai default input. Hal ini selaras dengan guideline yang digunakan oleh Qiscus.

### 3.2.1 Perancangan Conversational Flow

Perancangan *conversational flow* meliputi penentuan daftar intent yang digunakan dan menentukan nilai dari beberapa parameter yang terdapat pada dialogflow. Adapun daftar intent dan parameter yang digunakan dapat dilihat pada Tabel 3.2.

Tabel 3.2 Daftar Intent Serta Parameter

Nama Intent	Input	Respon	Fallback Type	Context
Greeting	Hai / hello	Halo saya asisten virtual Anda, silakan pilih keluhan yang ingin Anda laporkan, dengan mengetikkan nomer pada daftar menu  1. Status proses keluhan 2. Keluhan mengenai barang yang dibeli 3. Keluhan lainnya	False	
Default Fallback		mohon ketikkan (hai) untuk kembali ke menu	True	
Complain_status	1	Silakan masukkan ID keluhan Anda		Greeting
Purchased_Item	2	silakan pilih keluhan yang ingin Anda laporkan, dengan mengetikkan nomer pada daftar menu  1. Barang cacat 2. Salah Kirim barang 3. Barang tidak sesuai pesanan		Greeting
Other_complain	3	Isilah data mengikuti format nama#no-telpon#aduan		Greeting

Broken_item	1	Isilah data mengikuti format nama#no-telpon#aduan		Purchased_Item
Wrong_item	2	Isilah data mengikuti format nama#no-telpon#aduan		Purchased_Item
Items_don't_match	3	Isilah data mengikuti format nama#no-telpon#aduan		Purchased_Item
Complain_status-fallback				Komplain_status
Broken_item fallback				Broken_item
Wrong_item - fallback				Wrong_item
Other_complain-fallback				Other_komplain
Items_don't_match - fallback				Items_don't_match

*Chatbot* yang dibuat menggunakan Dialogflow sebagai *platform chatbot builder*. Dalam proses pembuatan *chatbot* pada Dialogflow *developer* perlu menentukan intent (kategori topik) yang akan digunakan. Intent digunakan sebagai modul yang digunakan *chatbot* dalam menganalisa *input*, pada proses pembuatan intent terdapat beberapa parameter yang harus ditentukan. Selain parameter dasar seperti nama intent, *input* dan respon parameter *fallback type* perlu memiliki nilai. Parameter *fallback type* merupakan parameter yang menentukan apakah intent termasuk kedalam jenis fallback. Fallback merupakan jenis intent yang digunakan oleh *chatbot* untuk mengkategorikan *input* yang tidak berhasil di analisa oleh *chatbot*. Parameter ini memiliki nilai TRUE atau FALSE.

Context merupakan parameter pada Dialogflow yang berfungsi sebagai flow *input* atau *output* pada intent. Penggunaan context pada Dialogflow menentukan relasi parent-child antar intent. Relasi tersebut berguna agar flow conversational tidak mempunyai banyak kemungkinan dan mengurangi kesalahan respon yang diberikan.

*Chatbot* yang dibuat memiliki dua fungsi utama yang pertama adalah menyimpan data komplain dan yang kedua menampilkan progress status komplain. Fungsi pertama akan dijalankan apabila *chatbot* berhasil menganalisis dan mengkategorikan pada empat intent diantaranya: *Broken\_item fallback*, *Wrong\_item - fallback*, *Other\_complain-fallback*, *Items\_don't\_match - fallback*. Pembuatan empat intent berbeda dengan fungsi yang sama

bertujuan agar data komplain tersimpan beserta jenis kategori komplain. Untuk fungsi kedua akan dijalankan apabila *chatbot* berhasil mengkategorikan pada intent *Complain\_status-fallback*.

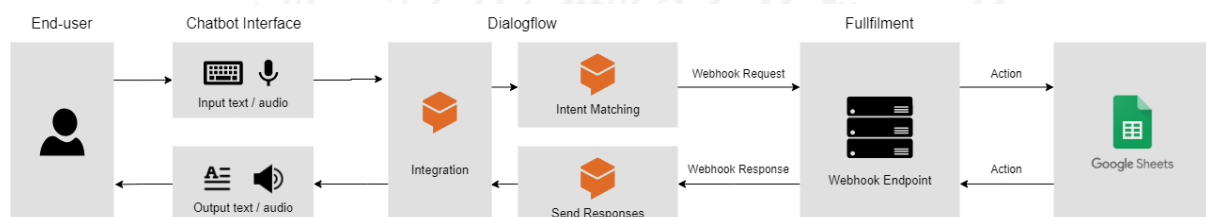
Intent yang digunakan untuk melakukan penyimpanan dan pengambilan data dari Google Sheet bertipe fallback. Alasan penggunaan tipe fallback dikarenakan agar Dialogflow dapat menganalisa dengan segala jenis *input* tanpa terjadinya *error*.

Proses penyimpanan dan pengambilan data dari *chatbot* memerlukan endpoint webhook dengan request method post pada *chatbot* berbasis Dialogflow. Untuk mengirimkan trigger event pada webhook, Pengembang perlu mendefinisikan nilai parameter fulfillment status. Fulfillment status merupakan parameter pada Dialogflow yang berfungsi menentukan apakah data *input* akan dieksekusi oleh webhook atau tidak. Parameter fulfillment status bertipe boolean dengan nilai *true* atau *false*. Daftar intent yang mengaktifkan parameter fulfillment status yaitu: *Complain\_status-fallback*, *Broken\_item fallback*, *Wrong\_item – fallback*, *Other\_komplain-fallback*, *Items\_don't\_match – fallback*.

Pemilihan Dialogflow sebagai *framework chatbot builder* pada proyek ini dikarenakan produk Qiscus Robolabs dikembangkan menggunakan *service* yang sama, sehingga diharapkan hasil yang di dapat dapat menjadi pertimbangan untuk penegembangan produk kedepannya.

### 3.2.2 Pembuatan Chatbot

Pembuatan *chatbot* memiliki dua tahap, tahap pertama merupakan pembuatan *chatbot* pada Dialogflow berdasarkan *conversational flow* yang sudah dibuat. Tahap selanjutnya merupakan pembuatan webhook atau *intermediary web* yang digunakan untuk mengintegrasikan *chatbot* dengan Google Sheet. Gambar 3.2 merupakan tampilan dari *flow chatbot* dan webhook yang akan dibuat



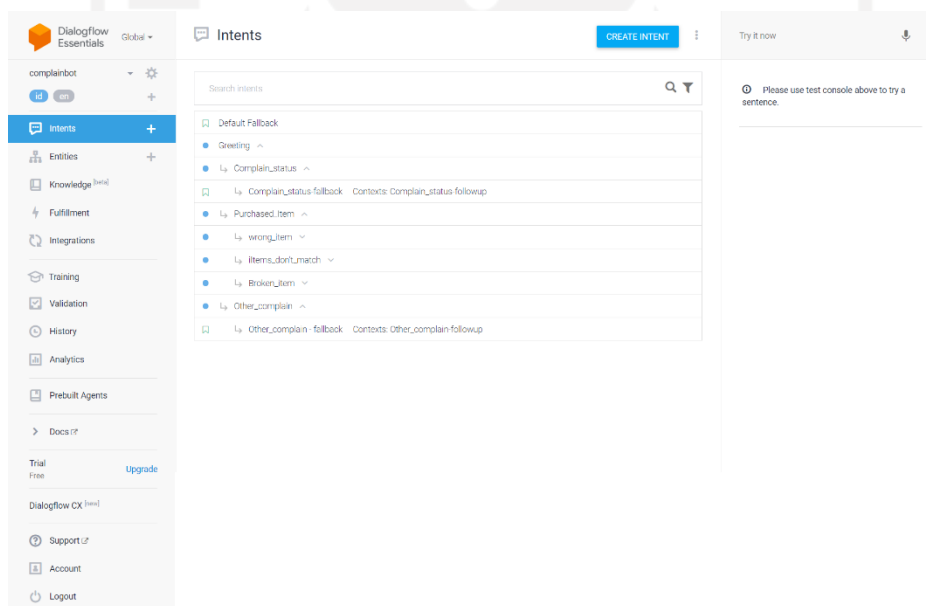
Gambar 3.2 Alur Kerja Chatbot Menggunakan Webhook

Webhook dibuat menggunakan bahasa pemrograman Python dengan framework FAST API. FAST API merupakan sebuah *framework* yang dikhususkan untuk pembuatan API. Flow dimulai dengan user mengetikkan *input* berupa audio atau teks. Kemudian dialogflow akan menganalisis dan mengkategorikan sesuai intent yang sudah dibuat. Setelah berhasil

mengkategorikan ke dalam intent, dialogflow akan mengecek nilai parameter fulfillment status. Apabila parameter fulfillment status bernilai *true* Dialogflow akan mengirimkan *trigger event* pada webhook endpoint. Webhook akan melakukan sesuai method yang dipilih untuk menyimpan atau membaca data dari Google Sheet.

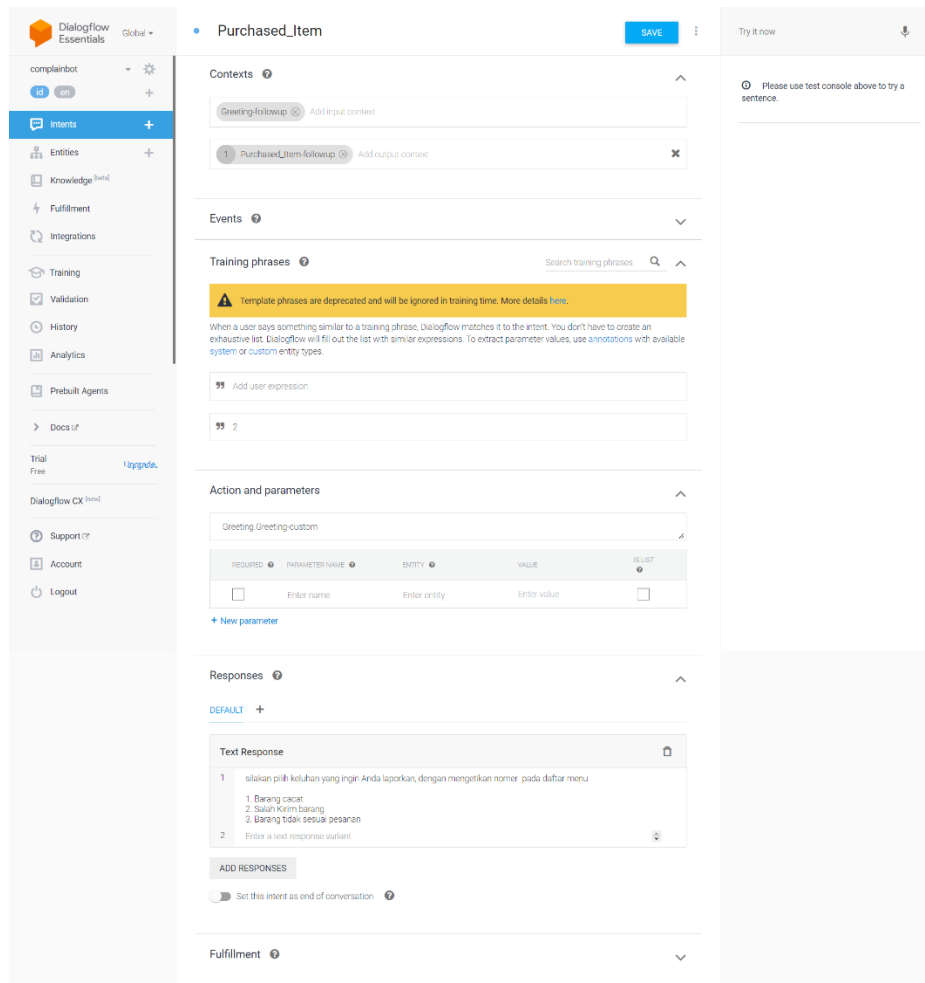
*Tools* yang digunakan untuk menjalankan service pada *cloud* adalah Heroku. Heroku dipilih dikarenakan memiliki *cost* yang sedikit diantara yang lainnya. Service pada heroku memerlukan sebuah file Procfile yang akan digunakan untuk menjalankan *code* yang telah dibuat.

Implementasi *conversational flow* pada *chatbot* dimulai dengan membuat sebuah Agent pada Dialogflow. Agent merupakan sebuah modul yang digunakan untuk mendefinisikan daftar Intent, dan respon yang akan ditampilkan oleh *chatbot*. *Chatbot* pada Dialogflow menggunakan intent sebagai basis analisisnya. Tampilan agent yang dibuat pada *platform* Dialogflow dapat dilihat pada Gambar 3.3



Gambar 3.3 Tampilan Agent Pada Dialogflow

Intent yang dibuat akan memiliki beberapa parameter yang perlu didefinisikan seperti pada Tabel 3.2. Dialogflow akan melakukan analisis pertama berdasarkan parameter *input* yang sudah ditentukan. hasil analisis Dialogflow akan dikategorikan berdasarkan intent. Semua alur suatu percakapan pada *chatbot* akan berlangsung berdasarkan parameter context yang digunakan untuk menentukan hubungan antar intent. Contoh tampilan intent beserta parameternya dapat dilihat pada Gambar 3.4.



Gambar 3.4 Tampilan Intent dengan Parameter

Setelah melakukan pembuatan *chatbot* sesuai *conversational flow* pada Dialogflow, *chatbot* sudah dapat digunakan akan tetapi fungsionalitas penyimpanan dan pengambilan data dari Google Sheet masih belum dapat berjalan. Hal ini dikarenakan *chatbot* membutuhkan sebuah webhook endpoint yang melakukan request pada *library* Google Sheet API. adapun beberapa fitur yang tersedia pada *chatbot* yaitu:

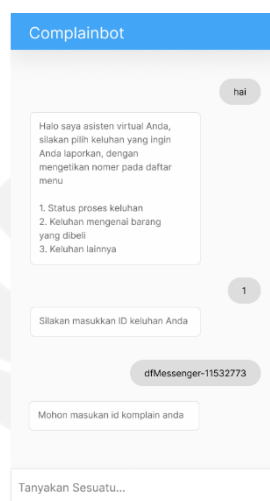
#### a. Menu Pengajuan Komplain

Menu pengajuan laporan merupakan sebuah fitur pada *chatbot* yang bertujuan untuk mengajukan komplain dari *customer*. menu ini memiliki dua kategori jenis komplain yaitu: komplain perihal barang yang dibeli dan komplain lainnya. Pada menu komplain perihal barang yang dibeli memiliki beberapa sub menu diantaranya: menu barang cacat, menu salah kirim barang dan menu barang tidak sesuai pesanan. Secara garis besar pengajuan komplain memiliki empat kategori jenis komplain. Empat kategori komplain tersebut akan disimpan pada Google



Sheet menggunakan nama masing-masing hasil analisa intent dari Dialogflow. Hal ini bertujuan untuk dapat mengkategorikan jenis komplain secara otomatis.

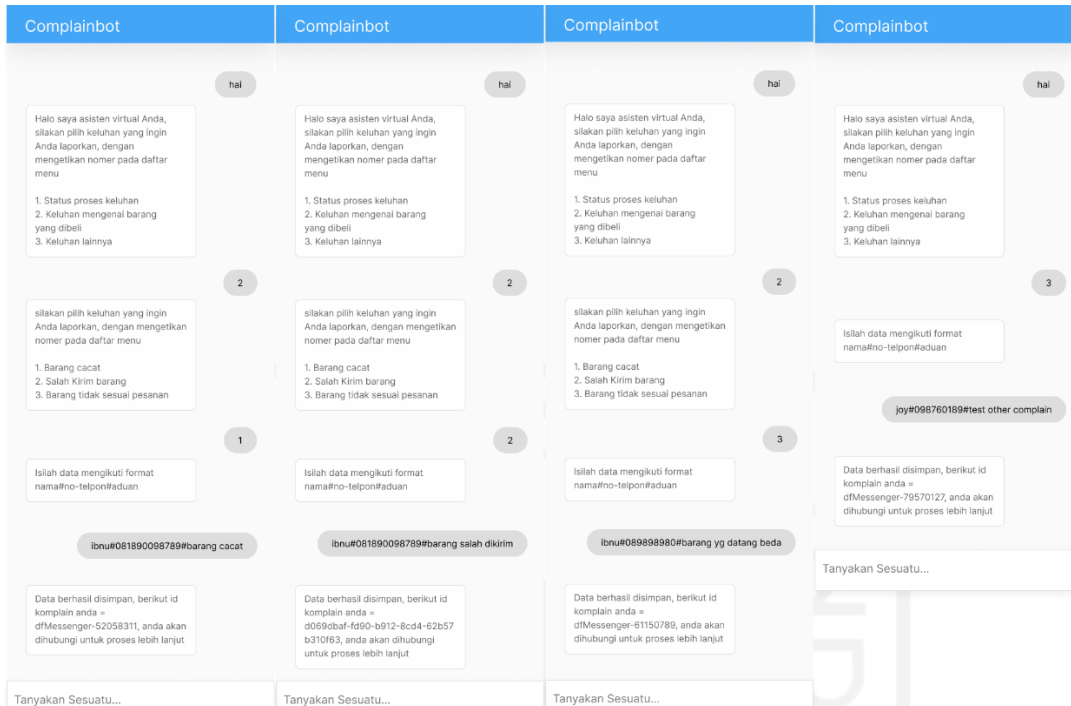
Pengajuan komplain dilakukan dengan mengikuti format yang sudah diberikan oleh *chatbot* agar data dapat tersimpan. Format pengajuan komplain pada *chatbot* adalah sebagai berikut: nama#no-telpon#aduan. Penggunaan format komplain agar mempermudah penyimpanan data pada Google Sheet. Apabila komplain tidak sesuai format yang diberikan *chatbot* akan memberikan respon berupa peringatan untuk mengajukan komplain sesuai format, Gambar 3.5 merupakan tampilan peringatan *chatbot* apabila komplain tidak sesuai dengan format.



Gambar 3.5 Tampilan Respon *Chatbot* Komplain Tidak Sesuai Format

Komplain yang sesuai dengan format akan tersimpan dan memberikan respons berupa id komplain, id komplain akan digunakan untuk *input* pada menu menampilkan status komplain. Alur pengajuan komplain pada masing-masing menu dapat dilihat dapat dilihat pada Gambar 3.6.

Komplain id mengikuti session id yang dibuat unik oleh dialogflow. Session id merupakan string dengan ukuran paling banyak 36 byte. Session id di generate otomatis oleh Dialogflow ketika terjadi percakapan antara *chatbot* dengan user. Waktu aktif suatu session id adalah 30 menit setelah interaksi terakhir dijalankan.

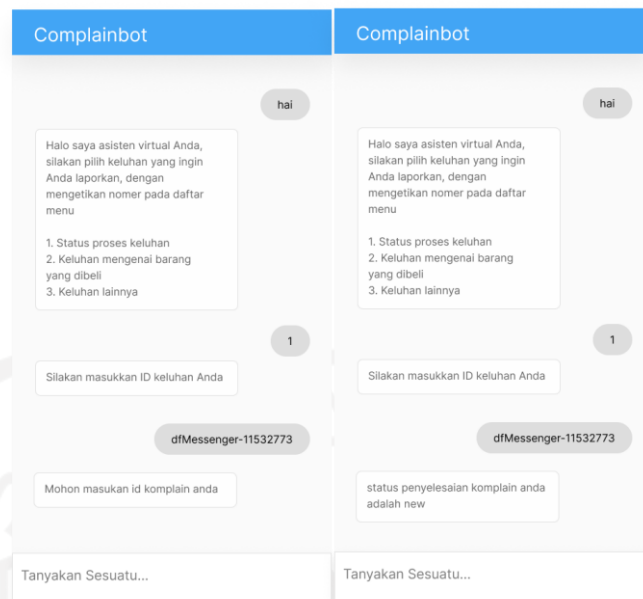


Gambar 3.6 Tampilan Alur Percakapan Pengajuan Komplain

#### b. Menu Menampilkan Status Komplain

Fitur menampilkan status penyelesai komplain digunakan untuk mendapatkan status perkembangan penyelesaian komplain. Data status perkembangan penyelesaian diambil dari Google Sheet, terdapat tiga status yang dibuat yaitu: *new*, *process*, *done*. Komplain akan memiliki status *new* pada saat pertama kali diajukan. Kemudian akan berubah *process* jika komplain yang masuk sedang melalui tahap penyelesaian. Komplain akan memiliki status *done* apabila sudah terselesaikan.

Pengubahan status dilakukan melalui Google Sheet yang terintegrasi dengan *chatbot* oleh penjual. *Chatbot* akan memberikan respon seperti sisi kiri Gambar 3.7 ketika user melakukan kesalahan dalam memasukan id komplain. *Chatbot* akan menampilkan status perkembangan penyelesaian komplain jika id komplain terdapat pada google sheet. Alur percakapan menu menampilkan status penyelesai komplain dapat dilihat pada sisi kanan Gambar 3.7.



Gambar 3.7 Tampilan Respon Menampilkan Status Komplain

### 3.2.3 Pembuatan Webhook

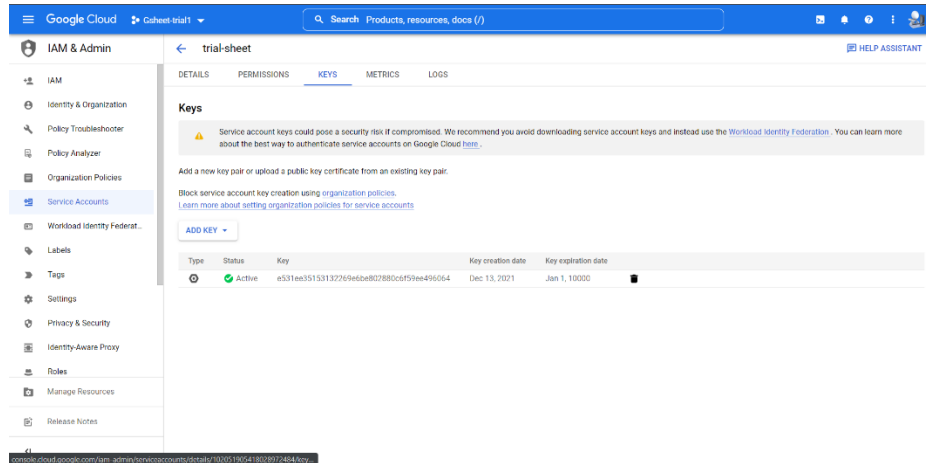
Bahasa pemrograman yang digunakan dalam membuat webhook adalah Python dengan *framework* Fast API. selama pembuatannya *library* Google Sheet API digunakan untuk melakukan fungsi untuk menyimpan dan membaca data dari Google Sheet. Pembuatan webhook meliputi beberapa kegiatan diantaranya: mempelajari penggunaan Google Sheet API, penerapan Google Sheet API dan pembuatan webhook, serta menjalankan webhook pada Heroku, kegiatan terakhir adalah pembuatan dokumentasi hasil penelitian implementasi webhook menggunakan *library* Google Sheet API.

#### a. Mempelajari Penggunaan Google Sheet API

*Library* Google Sheet API pada dasarnya merupakan sebuah API endpoint yang disediakan oleh Google. Tetapi untuk beberapa bahasa pemrograman seperti Go, Python, Javascript, Java, dan node js sudah tersedia *library* yang dapat digunakan. Penggunaan *library* dapat mempermudah pengembang dalam melakukan *request* dengan method yang disediakan. Dalam penggunaannya *library* Google sheet API memerlukan file kredensial yang digunakan sebagai *authorization* dalam melakukan *request*.

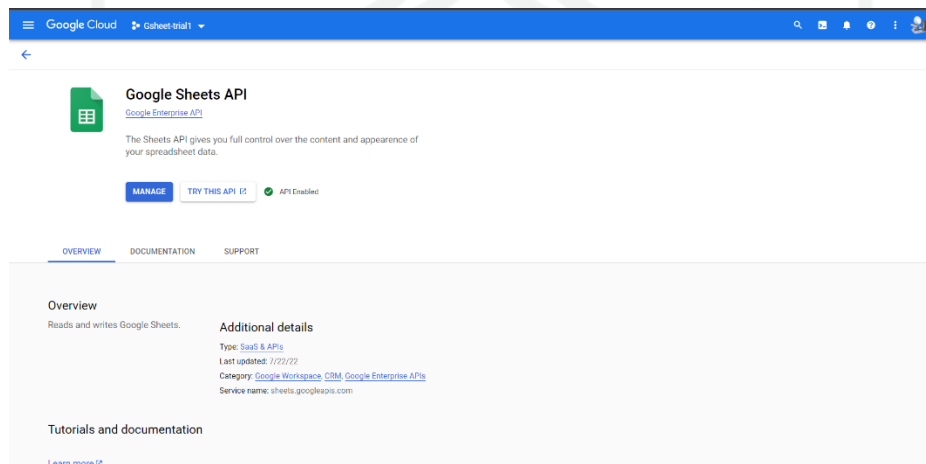
File kredensial dapat dibuat melalui Google Cloud Platform dengan membuat *service account* dengan *role editor*. *Service account* merupakan jenis akun khusus yang digunakan selain manusia untuk keperluan autentikasi dalam mengakses data melalui Google API. file kredensial dibuat dengan tipe json melalui fitur *keys* pada *service account*. File kredensial yang

dibuat akan terunduh otomatis dalam tipe json. Perlu diketahui file kredensial hanya bisa diunduh sekali pada saat pembuatan. Tampilan Google Cloud Platform yang sudah memiliki file kredensial dapat dilihat pada Gambar 3.8.

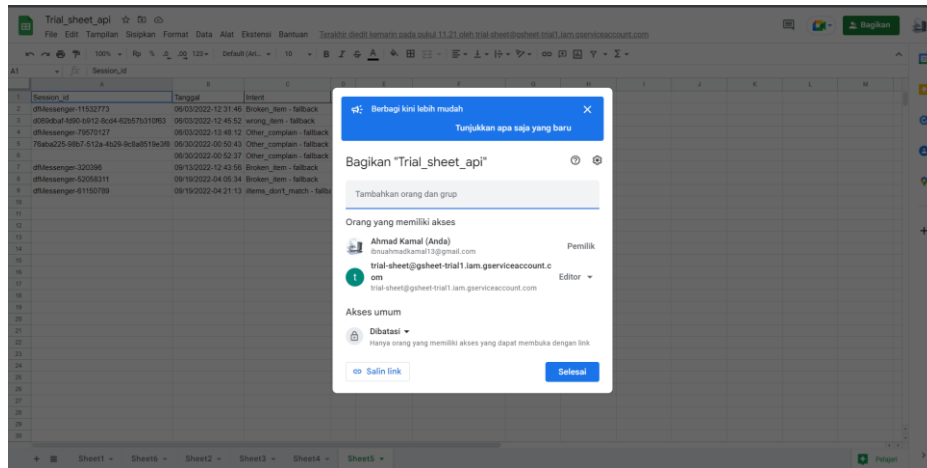


Gambar 3.8 Tampilan Menu Keys pada Google Cloud Platform

Layanan Google Sheet Api secara *default* akan berstatus *inactive* pada Google Cloud Platform. Untuk mengaktifkan layanan Google Sheet API dapat dilakukan pada fitur *API&Services*. Gambar 3.9 merupakan tampilan Layanan Google Sheet Api yang *active*. Layanan Google Sheet API akan dijalankan menggunakan *service account* dalam pengambilan, penulisan, dan perubahan data dari Google Sheet. Sehingga *service account* memerlukan akses dengan *role editor* pada Google sheet. Email yang digunakan *service account* dapat dilihat pada Google Cloud Platform. Gambar 3.10 merupakan tampilan Google Sheet yang sudah membagikan akses dengan *service account*.



Gambar 3.9 Tampilan Fitur API&Services



Gambar 3.10 Tampilan Hak Akses pada Google Sheet

*Library* Google Sheet API memiliki beberapa *method* yang dapat digunakan untuk memanipulasi data pada Google Sheet. *Method* yang disediakan pada dasarnya memiliki fungsionalitas untuk membaca, membuat, mengubah, dan menghapus data pada Google Sheet. Webhook yang dibuat tidak menggunakan semua *method* yang tersedia tetapi hanya menggunakan *method* dengan fungsionalitas membaca, membuat, dan mengupdate data. Adapun beberapa *method* yang digunakan pada webhook yang dibuat tertera pada Tabel 3.3.

Tabel 3.3 Daftar Method Google Sheet API yang Digunakan pada Webhook

Method	Deskripsi
Get	Membaca nilai dari rentang range yang ditentukan
Update	Menulis data pada range yang ditentukan
Append	Menambahkan data ke baris berikutnya pada range yang ditentukan

Selain *method* yang tertera pada tabel terdapat beberapa *method* yang dapat digunakan dalam mengelola data pada Google Sheet. *Method clear* merupakan salah satu *method* yang tidak digunakan pada webhook, *method* ini berfungsi untuk menghapus data sesuai range yang telah ditentukan. Tabel 3.4 merupakan daftar *method* pada *library* Google Sheet API yang tidak digunakan pada pembuatan webhook.

Tabel 3.4 Daftar Method Google Sheet API yang Tidak Digunakan pada Webhook

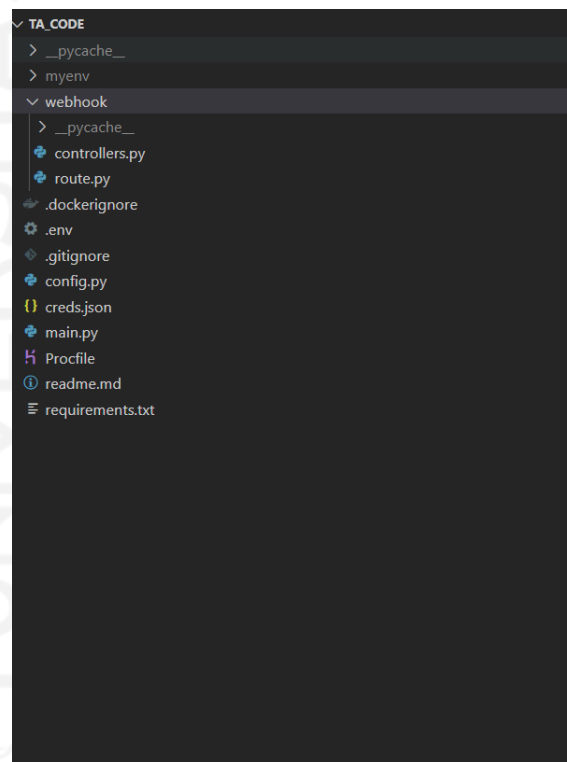
Method/function	Deskripsi
Clear	Menghapus nilai dari rentang range yang ditentukan
Batch Update	Menulis satu atau lebih nilai data pada range yang ditentukan
Batch Get	Membaca satu atau lebih nilai dari rentang range yang ditentukan

Kegiatan Selanjutnya setelah mempelajari penggunaan *Library* Google Sheet API adalah penerapan Google Sheet API dan pembuatan webhook. Webhook yang dibuat hanya akan

menggunakan method pada Tabel 3.3, hal ini dikarenakan fitur yang dirancang pada *chatbot* hanya sebatas menyimpan dan membaca data dari Google Sheet.

#### b. Penerapan Google Sheet API dan Pembuatan Webhook

*Library* Google Sheet API akan digunakan untuk di dalam controller webhook. Webhook hanya akan memiliki satu endpoint dengan request method post. Penggunaan jenis method ini dikarenakan Dialogflow hanya bisa menggunakan webhook bertipe post. Pengembangan dimulai dengan memilih versi Python yang akan digunakan. Versi Python yang digunakan pada penelitian ini adalah Python dengan versi 3.7, Python versi ini dipilih karena banyak proyek pada perusahaan magang yang dibuat menggunakan versi yang sama. Tampilan struktur folder dapat dilihat pada Gambar 3.11.



Gambar 3.11 Tampilan Struktur Folder Webhook

Selain menggunakan *library* Google Sheet API, pembuatan webhook menggunakan *library* pandas yang berfungsi untuk memanipulasi dan pembersihan data. *Library* pandas memiliki sebuah *object* yang dinamakan dataframe, dataframe adalah *object* yang memiliki struktur data tabular yang berorientasi pada kolom dan baris. Struktur data jenis ini cocok untuk digunakan untuk membaca dan memanipulasi data dengan bentuk tabel.

*Library* lainnya yang digunakan dalam pembuatan webhook adalah *Library Numpy*, *library* ini digunakan untuk mengatur data dalam bentuk dimensional array sehingga lebih mudah dalam mengelola data dalam bentuk tabel.

*Library* Google Sheet API digunakan dengan cara melakukan inisiasi *service*. Function inisiasi *service* memerlukan nilai parameter *credentials* yang didapat dengan cara memasukan nilai *scopes* dan nama file kredensial pada function *from\_service\_account\_file*. Konfigurasi inisiasi *service* dapat dilihat pada Gambar 3.12. Nilai parameter *scopes* merupakan url yang didalamnya terdapat jenis izin akses file. Read and Write merupakan jenis izin akses yang digunakan webhok dalam melakukan manipulasi dan membaca data dari Google Sheet.

```
creds = service_account.Credentials.from_service_account_file(
    SERVICE_ACCOUNT_FILE, scopes=SCOPES)

service = build('sheets', 'v4', credentials=creds)
sheet = service.spreadsheets()
```

Gambar 3.12 Tampilan Kode Inisiasi *Service*

Berdasarkan pada Gambar 3.12 baris kode 4 berfungsi untuk melakukan *build service* dengan cara menentukan versi *service* dan nilai parameter *credentials* yang akan digunakan. Setelah melakukan *build* proses selanjutnya adalah inisiasi *service* seperti yang tertera pada baris 5.

Webhook yang dibuat memiliki dua komponen utama yaitu controller dan route. Controller webhook akan berupa kumpulan *function* yang akan dipanggil pada Route, Route berfungsi untuk mendefinisikan jenis method request yang digunakan dan melakukan mapping berdasarkan nama intent. Mapping ini yang akan menentukan *function* mana yang dipanggil ketika Dialogflow berhasil menganalisa suatu intent. Controller webhook memiliki tiga function utama yaitu: *read\_all\_data*, *write\_data*, dan *filter\_data\_byID*.

Function *read\_all\_data* akan membaca keseluruhan data dan mengkonversikannya ke dalam object dataframe agar mempermudah melakukan manipulasi data. Selain melakukan konversi ke dalam object dataframe, function *read\_all\_data* juga melakukan pemanggilan method *get* pada *library* Google Sheet API. Contoh tampilan penggunaan method *get* dapat dilihat pada Gambar 3.13 baris 1-2. Parameter yang dibutuhkan untuk menjalankan *method* *get* adalah *spreadsheetId* dan *range*. Nilai parameter *spreadsheetId* diambil dari url Google Sheet yang digunakan. Sedangkan parameter *range* bernilai nama page pada Google Sheet atau range antara dua cell.

```

result = sheet.values().get(spreadsheetId=SAMPLE_SPREADSHEET_ID,
                           range=SAMPLE_RANGE_NAME).execute()

data = result.get('values')
df = pd.DataFrame(data)

```

Gambar 3.13 Tampilan Kode Method Get

Pengembalian nilai *method* get pada *library* Google Sheet Api akan bertipe dict dengan tiga key utama yaitu: range, majorDimension dan values. Key range, dan major dimension akan bertipe string sedangkan key values akan memiliki nilai bertipe list yang berisikan data dari Google Sheet dengan baris sebagai dimensi utama. Baris 4-5 pada Gambar 3.13 merupakan perintah untuk mendapatkan data dari Google sheet yang akan dikonversikan ke dalam object dataframe.

Pada function `read_all_data` sistem akan melakukan pengecekan nilai *shape* dari data frame yang dibuat. jika *shape* dataframe bernilai 0 maka sistem akan membuat header pada dataframe. Tampilan kode pembuatan header dataframe dapat dilihat pada Gambar 3.14 baris 1-5.

```

if df.shape[0] == 0:
    df = pd.DataFrame(data,columns=['Session_id','Tanggal','Intent',
                                   'Nama','Nomer Telp','Aduan','Status'])

    value = [df.columns.values.tolist()]
else:
    df = pd.DataFrame(data[1:], columns=data[0])

if not set(['Session_id','Tanggal','Intent',
           'Nama','Nomer Telp','Aduan','Status']).issubset(list(df.columns)):

    df = pd.DataFrame(data,columns=['Session_id','Tanggal','Intent',
                                   'Nama','Nomer Telp','Aduan','Status'])
    value = [df.columns.values.tolist()+df.values.tolist()]
else:
    value = data

```

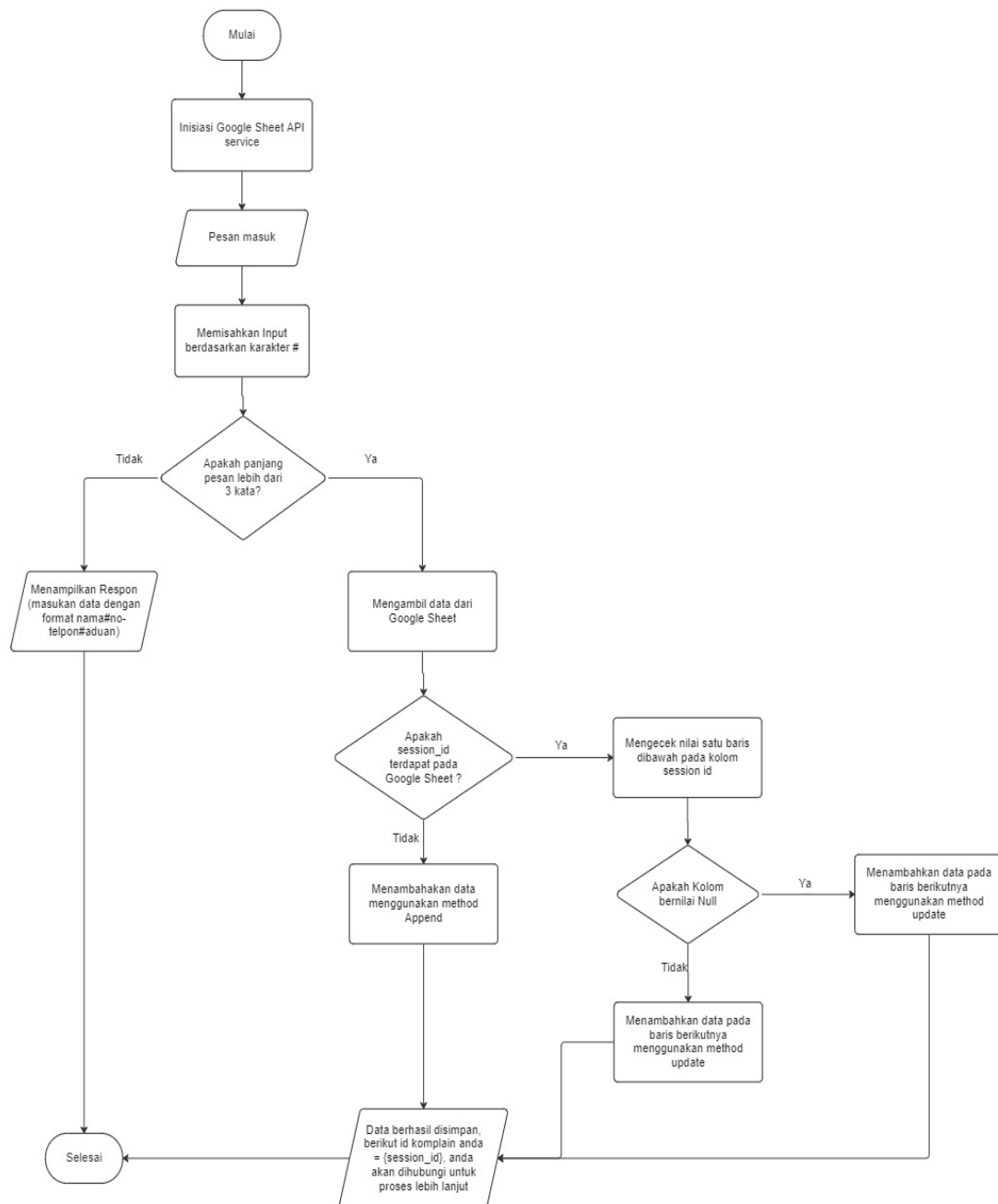
Gambar 3.14 Tampilan Kode Pembuatan Header pada Datframe

ketika sistem melakukan pengecekan *shape* dan tidak mendapatkan nilai 0 maka akan dilakukan pengecekan header dataframe. Baris 7-10 pada Gambar 3.14 akan dieksekusi apabila list header tidak terdapat pada index pertama dataframe. Dataframe akan dikonversikan ke dalam list dengan menambahkan header sebagai index pertama dan data yang didapat sebagai index setelahnya. Hal ini dapat dilihat pada Gambar 3.14 baris 12 – 14.

Function `read_all_data` akan memiliki pengembalian nilai bertipe list dengan index pertama berupa header yang sudah ditentukan, function ini akan digunakan sebagai logika



utama pada function `write_data`, dan `filter_data_byID`. Penulisan data ke dalam Google Sheet akan menggunakan *function* `write_data` dengan menggunakan tiga parameter yaitu: `session_id`, `intent_name` dan `message`. pada *function* ini akan memiliki logika yang lebih kompleks. Alur kerja *function* `write_data` dapat dilihat pada Gambar 3.15.



Gambar 3.15 Alur Kerja *Function* `Write_data`

Alur kerja *function* `write_data` dimulai dengan melakukan pengecekan pesan yang masuk apakah sudah sesuai dengan format yang ditentukan, apabila pesan teks yang masuk tidak sesuai dengan format maka *chatbot* akan memberikan respon seperti berikut: "masukan data dengan format nama#no-telpon#aduan".

Pesan yang sesuai dengan format akan dilanjutkan pada proses penentuan *method* penulisan data. *method* penulisan data akan menggunakan *method update* atau *append*, perbedaan kedua *method* dapat dilihat pada Tabel 3.3. Setelah melakukan penentuan *method*, proses dilanjutkan dengan mengirimkan respon seperti berikut: "Data berhasil disimpan, berikut id komplain anda = {Id komplain}, anda akan dihubungi untuk proses lebih lanjut". Id komplain akan dibuat secara otomatis menggunakan `session_id` pada *Dialogflow*. Tampilan kode untuk melakukan pengecekan format pesan dapat dilihat pada Gambar 3.16.

```
messages = message.strip(' ').split('#')

if len(messages) == 3:
    name = messages[0]
    number = messages[1]
    komplain = messages[2]
```

Gambar 3.16 Tampilan Kode Pengecekan Format Pengajuan komplain

Setelah melakukan pengecekan format pengajuan komplain, sistem akan membaca data dengan memanggil *function* `read_all_data` dan menyimpannya ke dalam sebuah variabel. Apabila pada proses pengecekan format panjang pesan tidak sama dengan 3 kata, maka sistem akan mengirimkan respon *chatbot* seperti berikut: "masukan data dengan format nama#no-telpon#aduan". Data yang didapat dari hasil pemanggilan *function* `read_all_data` akan dikonversikan ke dalam bentuk object *dataframe* agar mempermudah manipulasi data. Tampilan kode proses dapat dilihat pada Gambar 3.17.

```
data = read_all_data()
df = pd.DataFrame(data[1:], columns=data[0])
```

Gambar 3.17 Tampilan Kode Konversi Data Bertipe List Menjadi Dataframe

Proses akan dilanjutkan dengan pengecekan `session_id` dari data yang didapat. Jika `session_id` tidak terdapat pada data yang didapat maka penulisan akan menggunakan *method* `append`. Proses penulisan data pada *Google Sheet* dapat dilihat pada Gambar 3.18 baris 4-7.

```
if session_id not in set(df['Session_id']):
    values = [session_id,date,intent_name,name,number,komplain,'new']

    result = sheet.values().append(spreadsheetId=SAMPLE_SPREADSHEET_ID,
                                   range=SAMPLE_RANGE_NAME,valueInputOption="USER_ENTERED",
```

```

insertDataOption="INSERT_ROWS",
body={"values":[values]}).execute()

if result['spreadsheetId'] == SAMPLE_SPREADSHEET_ID:
    response = f"Data berhasil disimpan, berikut id komplain anda =
    {session_id}, anda akan dihubungi untuk proses lebih lanjut"
else:
    response = "Data tidak tersimpan"

```

Gambar 3.18 Tampilan Kode Penulisan Data Menggunakan Method *Append*

Hasil dari eksekusi method *append* data akan berupa dict yang memiliki tiga key yaitu: *spreadsheetId*, *tableRange* dan *ranges*. Apabila method mengalami *error* maka dict hanya memiliki satu key yaitu *error* yang memiliki nilai berupa string. Untuk menentukan status keberhasilan *method append* maka sistem akan mengecek nilai key *spreadsheetId* dengan id Google Sheet yang digunakan. Proses ini berfungsi untuk menentukan respon *chatbot* yang akan dikirimkan. Respon *chatbot* akan memiliki id komplain apabila nilai key *spreadsheetId* sama dengan id Google Sheet. Apabila kedua nilai tersebut berbeda maka *chatbot* akan memberikan respon “Data tidak tersimpan”. Proses ini dapat dilihat pada Gambar 3.18 baris 9-13.

*Method update* akan digunakan untuk menambahkan data jika pada waktu proses membaca data sistem mendapatkan *session\_id* yang sama. Alasan penggunaan *method* yang berbeda bertujuan untuk format penulisan data pada Google Sheet. Penulisan data pada Google Sheet hanya akan menyimpan satu *session\_id*, sehingga jika terdapat request penulisan data dengan *session\_id* yang sama maka data baru akan ditambahkan pada baris berikutnya. Data baru akan memiliki *session\_id* bernilai null karena *session\_id* data baru merujuk pada kolom *session\_id* baris sebelumnya.

Proses pengecekan *session\_id* akan menggunakan *dataframe* untuk mendapatkan index posisinya. Index tersebut akan digunakan untuk menentukan posisi penulisan data yang baru tampilan baris kode untuk mendapatkan index posisi dapat dilihat pada Gambar 3.19.

```
index_id = df.index[df['Session_id']==session_id].tolist()[0]
```

Gambar 3.19 Tampilan Kode Mendapatkan Nilai Index

Perulangan digunakan untuk melakukan pengecekan untuk menentukan posisi penulisan data baru. perulangan akan dimulai dari index posisi yang didapat dan akan menuliskan data baru pada baris setelahnya. Perulangan akan menggunakan baris sebagai dimensi utamanya. Gambar 3.20 baris 13-15 merupakan tampilan kode pada proses penulisan data menggunakan *method update*.

```

for i, row in df.iloc[index_id:].iterrows():

    if row['Session_id'] == "":
        index_id = i
        values = DataFrame({'Session_id': None, 'Tanggal': date,
                            'Intent': intent_name, 'Nama': name, "Nomer Telp": number,
                            'Aduan': komplain, 'Status': "new"}, index=[index_id+1])

        values_df = [df.columns.values.tolist()+
                    pd.concat([df.iloc[:index_id+1], values,
                               df.iloc[index_id+1:]]).reset_index(drop=True).values.tolist()]

        result = sheet.values().update(spreadsheetId=SAMPLE_SPREADSHEET_ID,
                                       range=SAMPLE_RANGE_NAME, valueInputOption="USER_ENTERED",
                                       body={"values": values_df}).execute()

    if result['spreadsheetId'] == SAMPLE_SPREADSHEET_ID
        response = f"Data berhasil disimpan,
                    berikut id komplain anda = {session_id}, anda akan dihubungi untuk proses lebih lanjut"
    else:
        response = "Data tidak tersimpan"

```

Gambar 3.20 Tampilan Kode Penulisan Data dengan Session\_Id Bernilai Null

Pemeriksaan lainnya yang perlu dilakukan adalah pemeriksaan nilai `session_id` pada data yang dibaca. Pemeriksaan ini bertujuan agar data baru ditambahkan setelah data lama. Tampilan pengecekan dan penambahan data baru dapat dilihat pada baris 3-11 Gambar 3.20. Selain melakukan pemeriksaan nilai `session_id`, pada kode tersebut juga dilakukan konversi dari data frame ke dalam list agar dapat digunakan pada *method update*. Penggunaan nilai null sebagai pengecekan agar data yang baru memiliki posisi pada baris berikutnya.

Respon akan diberikan sesuai kondisi keberhasilan menjalankan *method update*. Respon akan memiliki id komplain jika *method* berhasil dijalankan. Apabila *method* tidak berhasil dijalankan maka respon berupa “Data tidak tersimpan”. pemeriksaan ini dilakukan dengan cara mengecek nilai *key spreadsheetId* dengan id Google Sheet yang digunakan.

Kondisi lainnya yang dapat dijumpai dari kode pada Gambar 3.20 adalah `session_id` memiliki nilai tidak null, apabila `session id` memiliki nilai tidak null maka data baru akan ditambahkan pada baris berikutnya. Tampilan kode proses ini dapat dilihat pada Gambar 3.21.

```

else:

    values = DataFrame({'Session_id': None, 'Tanggal': date,
                        'Intent': intent_name, 'Nama': name, "Nomer Telp": number,
                        'Aduan': komplain, 'Status': "new"}, index=[index_id+1])

    values_df = [df.columns.values.tolist()+
                pd.concat([df.iloc[:index_id+1], values,
                           df.iloc[index_id+1:]]).reset_index(drop=True).values.tolist()]

    result = sheet.values().update(spreadsheetId=SAMPLE_SPREADSHEET_ID,

```

```

range=SAMPLE_RANGE_NAME, valueInputOption="USER_ENTERED",
body={"values": values_df}).execute()

if result['spreadsheetId'] == SAMPLE_SPREADSHEET_ID
    response = f"Data berhasil disimpan,
                berikut id komplain anda = {session_id}, anda akan dihubungi untuk proses lebih lanjut"
else:
    response = "Data tidak tersimpan"

```

Gambar 3.21 Tampilan Kode Penulisan Data dengan Session\_Id Bernilai Tidak Null

Perbedaan baris kode pada Gambar 3.20 dengan Gambar 3.21 adalah penggunaan index posisi yang digunakan. Pada Gambar 3.20 index yang digunakan adalah index dengan session\_id bernilai null, sedangkan index yang digunakan pada Gambar 3.21 merupakan index posisi session\_id dengan nilai tidak null.

*Function* terakhir yang terdapat pada controller webhook adalah *filter\_data\_byID*. *Function* ini tidak menggunakan *method* dari *library* Google Sheet Api, akan tetapi menggunakan *function* *read\_all\_data* untuk membaca data dari Google Sheet. Data yang didapat akan dikonversikan ke dalam *dataframe* untuk mempermudah pencarian data id komplain. *Function* *filter\_data\_byID* hanya memiliki satu parameter yaitu *message* yang akan diisi dengan id komplain. Id komplain memiliki nilai yang sama dengan session\_id, penggunaan nilai yang sama agar mempermudah pembuatan webhook.

Proses yang terjadi ketika *function* *filter\_data\_byID* dipanggil diantaranya: melakukan pengambilan data melalui *function* *read\_all\_data* dan melakukan pencarian berdasarkan id komplain. Gambar 3.22 merupakan tampilan baris kode pada *function* *filter\_data\_by\_ID*

```

data = read_all_data()
df = pd.DataFrame(data[1:], columns=data[0])

index_id = df.index[df['Session_id']==message].tolist()[0]

# checking input
if message == df['Session_id'][index_id]:
    response = df['Status'][index_id]
    response = f'status penyelesaian komplain anda adalah {status}'
else:
    response = "Mohon masukan id komplain anda"

```

Gambar 3.22 Tampilan Kode *Function* *filter\_data\_by\_ID*

Pengembalian nilai pada *function* *write\_data* dan *filter\_data\_byID* akan berupa string. Pengembalian nilai kedua *function* tersebut akan digunakan sebagai respon yang akan dikirimkan oleh *chatbot*.

Mapping function dilakukan di dalam route dengan nama intent sebagai acuan. Webhook tidak memerlukan request model body karena akan mengikuti request model body yang dikirimkan dari Dialogflow. Return dari route akan digunakan dalam mengisi parameter yang akan digunakan oleh *chatbot*, sehingga pengembalian yang dibuat harus mengikuti aturan dari Dialogflow. Pengembalian route akan berbentuk json agar dapat digunakan oleh Dialogflow. Pada pengembalian webhook terdapat dua key utama yang akan digunakan yaitu: “fulfillmentText” dan “source”. Key “fulfillmentText” akan diisi dengan respon dari setiap *function* sedangkan key “source” akan bernilai “webhook data”. Tampilan baris kode dapat dilihat pada Gambar 3.23.

```
@webhook.post("/webhook")
async def webhook(resource:Request):

    # get dialogflow request JSON
    req = await resource.json()

    # get data from dialogflow request
    session_id = req['session'].split('/')[-1]
    intent_name = req['queryResult']['intent']['displayName']
    message = req['queryResult']['queryText']

    # mapping response based dialogflow intent
    if intent_name == "Komplain_status-fallback":
        result = controllers.filter_data_byID(message)
    elif intent_name == "Broken_item - fallback":
        result = controllers.write_data(session_id,
            intent_name,message)
    elif intent_name == "wrong_item - fallback":
        result = controllers.write_data(session_id,
            intent_name,message)
    elif intent_name == "iItems_don't_match - fallback":
        result = controllers.write_data(session_id,
            intent_name,message)
    elif intent_name == "Other_komplain - fallback":
        result = controllers.write_data(session_id,
            intent_name,message)

    return {
        "fulfillmentText": result,
        "source": "webhook data"
    }
```

Gambar 3.23 Tampilan Kode Route Webhook

### c. Menjalankan Webhook pada Heroku

Webhook akan dijalankan pada cloud server. Heroku dipilih sebagai layanan penyedia cloud server karena memiliki biaya yang lebih murah. Penempatan webhook pada cloud server bertujuan agar sistem dapat menjalankan fungsinya secara real-time. Untuk melakukan deploy

*service* pada heroku server dapat dilakukan menggunakan dua cara, yang pertama menggunakan git kemudian yang kedua menggunakan container.

Deploy menggunakan git dilakukan dengan cara membuat sebuah Procfile yang berisikan perintah untuk menjalankan *service*. Deploy menggunakan cara ini sama dengan melakukan upload code file ke suatu repository. Untuk cara yang kedua deploy menggunakan container dibutuhkan Dockerfile dan Container docker yang sudah dibuat menggunakan *image* Dockerfile yang dibuat.

Kegiatan *deploy* ke heroku server pada penelitian ini akan menggunakan cara yang pertama. File procfile dibuat dan ditempatkan pada folder utama, tampilan struktur folder tertera pada Gambar 3.11. Kode baris pada procfile file berisikan perintah menjalankan webhook dan jumlah worker yang digunakan. Tampilan kode pada Procfile file dapat dilihat pada Gambar 3.24.

```
web: gunicorn -w 3 -k uvicorn.workers.UvicornWorker main:app
```

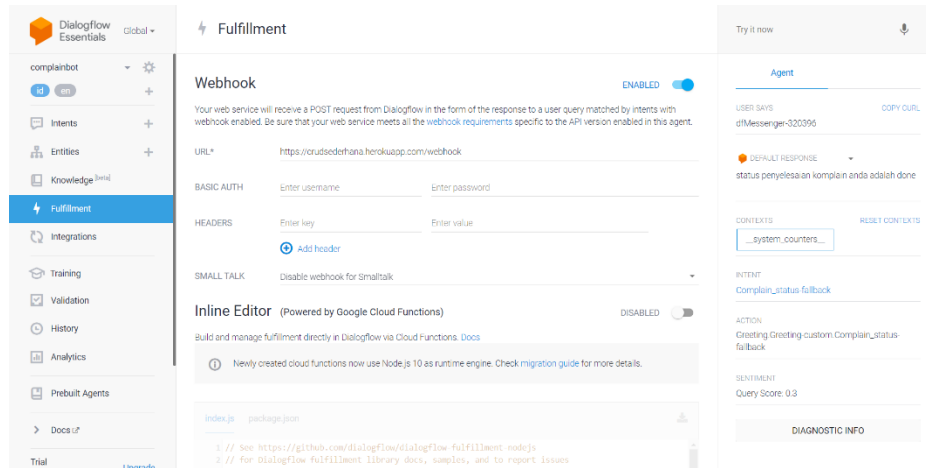
Gambar 3.24 Tampilan Kode pada File Procfile

*Service* yang di *deploy* belum dapat digunakan karena environment variable pada heroku tidak menggunakan file env. Untuk melakukan pendefinisian nilai environment variable perlu melakukan konfigurasi pada fitur *setting* menu *config vars*. Variabel yang digunakan dapat dilihat Tabel 3.5.

Tabel 3.5 Daftar *Environment Variable*

Nama Variabel	Deskripsi
SCOPES	Bernilai sebuah url yang didalamnya terdapat jenis izin akses file
SERVICE_ACCOUNT_FILE	Bernilai sebuah nama file credentials yang digunakan
SAMPLE_SPREADSHEET_ID	Bernilai id Google Sheet yang digunakan
SAMPLE_RANGE_NAME	Bernilai range yang akan dieksekusi dengan method

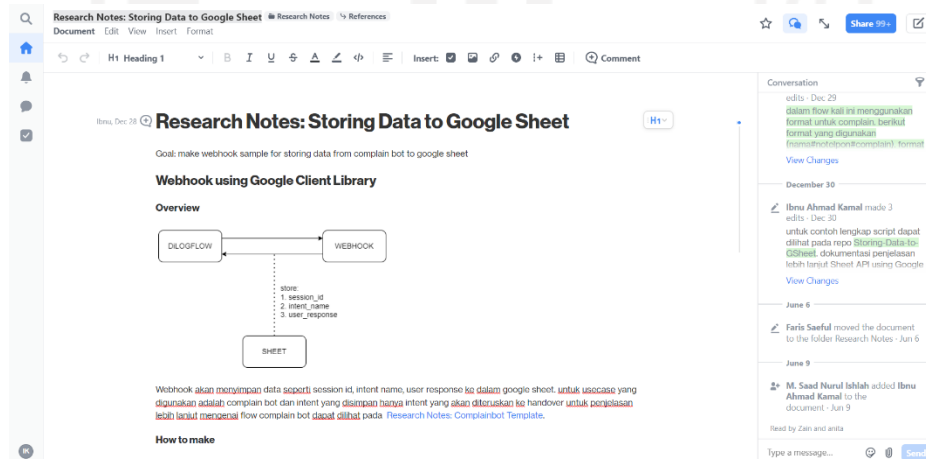
Status keberhasilan penempatan webhook pada heroku server dapat dilihat pada log dan *deploy activity*. Setelah webhook berhasil berjalan pada heroku server, endpoint webhook perlu ditambahkan pada Dialogflow. Endpoint akan ditambahkan pada fitur fulfillment. Tampilan fitur fulfillment dengan endpoint dapat dilihat pada Gambar 3.25.



Gambar 3.25 Tampilan Fitur Fulfillment pada Dialogflow

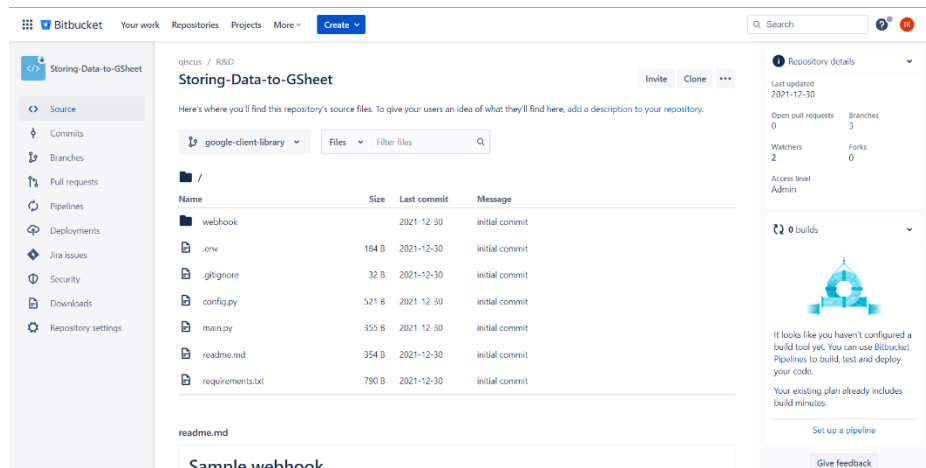
#### d. Pembuatan Dokumentasi Penelitian

Kegiatan terakhir yang dilakukan dalam pelaksanaan proyek adalah membuat dokumentasi penerapan. Dokumentasi ini berisikan cara implementasi *library* Google Sheet API pada webhook berbasis Dialogflow. Dokumentasi yang dihasilkan pada proyek penelitian ini berupa dokumen *research notes* dan *repository code*. Dokumen *research notes* disimpan pada folder quip perusahaan sedangkan *repository code* di simpan menggunakan ke dalam *project* bitbucket milik perusahaan. Tampilan dokumen *research notes* dapat dilihat pada Gambar 3.26. Gambar 3.27 merupakan tampilan repository code untuk webhook yang dibuat.



Gambar 3.26 Tampilan Dokumen Research Notes





Gambar 3.27 Tampilan Repository Code

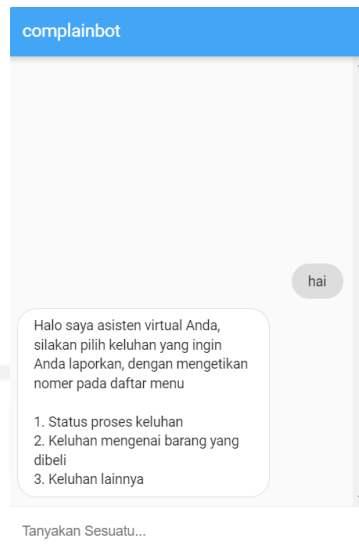
### 3.2.4 Pengujian *Chatbot*

*Chatbot* akan diuji menggunakan metode blackbox testing, pengujian jenis ini merupakan pengujian fungsionalitas. Pengujian dilakukan dengan cara menjalankan *chatbot* pada dua menu utama yaitu pengajuan komplain dan menampilkan status penyelesain komplain. Menu pengajuan komplain akan diuji dengan menjalankan fungsionalitas empat jenis pengajuan komplain yang berbeda. Pengujian bertujuan untuk mengetahui *chatbot* yang dibuat berjalan dengan baik.

Pengujian pertama akan dilakukan dengan menguji menu pengajuan laporan komplain. Menu laporan komplain memiliki empat sub menu yang akan diuji yaitu: menu barang cacat, menu salah kirim barang, menu barang tidak sesuai pesanan, dan menu komplain masalah lainnya.

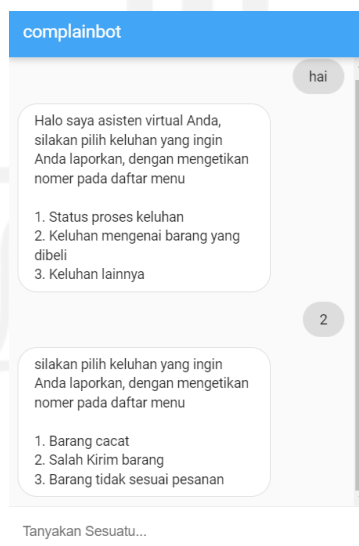
#### a. Simulasi Pengujian Menu Pengajuan Laporan Komplain

Pengujian dimulai dengan melakukan simulasi percakapan pengajuan komplain untuk salah satu menu yaitu menu salah kirim barang. Percakapan dimulai dengan user menginputkan kata “hai”, kemudian *chatbot* akan memberikan respon dengan menampilkan daftar menu yang dapat digunakan sesuai kebutuhan pengguna. Tampilan gambar daftar menu utama dapat dilihat pada Gambar 3.28.



Gambar 3.28 Tampilan Menu Utama

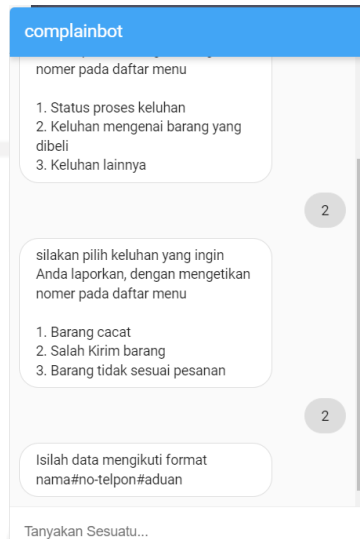
Percakapan dilanjutkan dengan pengguna mengetikkan angka 2 sebagai *input* untuk membuat laporan komplain dengan kategori keluhan barang yang dibeli. Simulasi dilanjutkan dengan user mengetikkan angka 2 pada *chatbot*. Selanjutnya Dialogflow akan mengkategorikan ke dalam intent “*purchased\_item*” dan memberikan respon seperti pada Gambar 3.29. pada intent ini dialogflow belum melakukan request pada webhook endpoint.



Gambar 3.29 Tampilan Respon Intent *Purchased\_item*

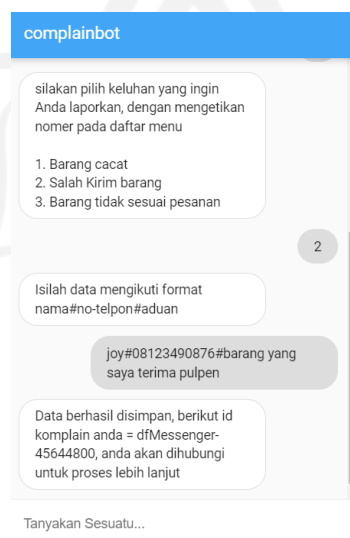
Pengguna akan memilih menu yang sesuai dengan kendala yang dihadapi. Pemberian tiga kategori pada pelaporan komplain bertujuan untuk mempermudah pemantauan data pada

Google Sheet. Pada simulasi percakapan kali ini pengguna memilih melaporkan salah kirim barang dengan menginputkan angka 2. *Input* yang masuk akan dikategorikan oleh Dialogflow sebagai intent “*Wrong\_item*” sehingga Dialogflow akan memberikan respon seperti pada Gambar 3.30.

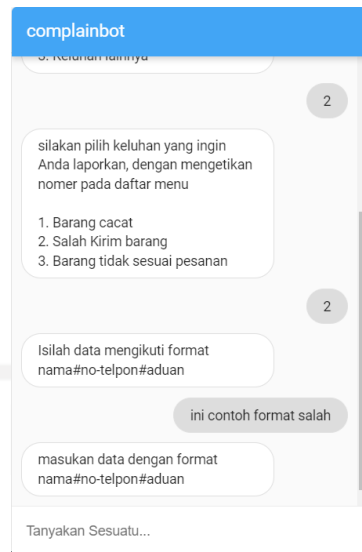


Gambar 3.30 Tampilan Respon Intent *Wrong\_item*

Respon yang diberikan berupa format yang digunakan untuk mengajukan laporan, user akan memasukan *input* sesuai format yang ditentukan. *Chatbot* akan memberikan respon beserta id komplain seperti pada Gambar 3.31 apabila data berhasil di simpan ke Google Sheet. Apabila format pengajuan tidak sesuai *chatbot* akan memberikan respon seperti pada Gambar 3.32.



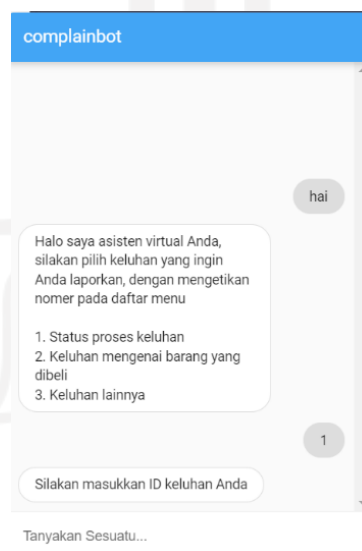
Gambar 3.31 Tampilan Respon Data Berhasil Disimpan



Gambar 3.32 Tampilan Respon Input Tidak Sesuai Format

#### b. Simulasi Pengujian Menu Menampilkan Status Penyelesaiain Komplain

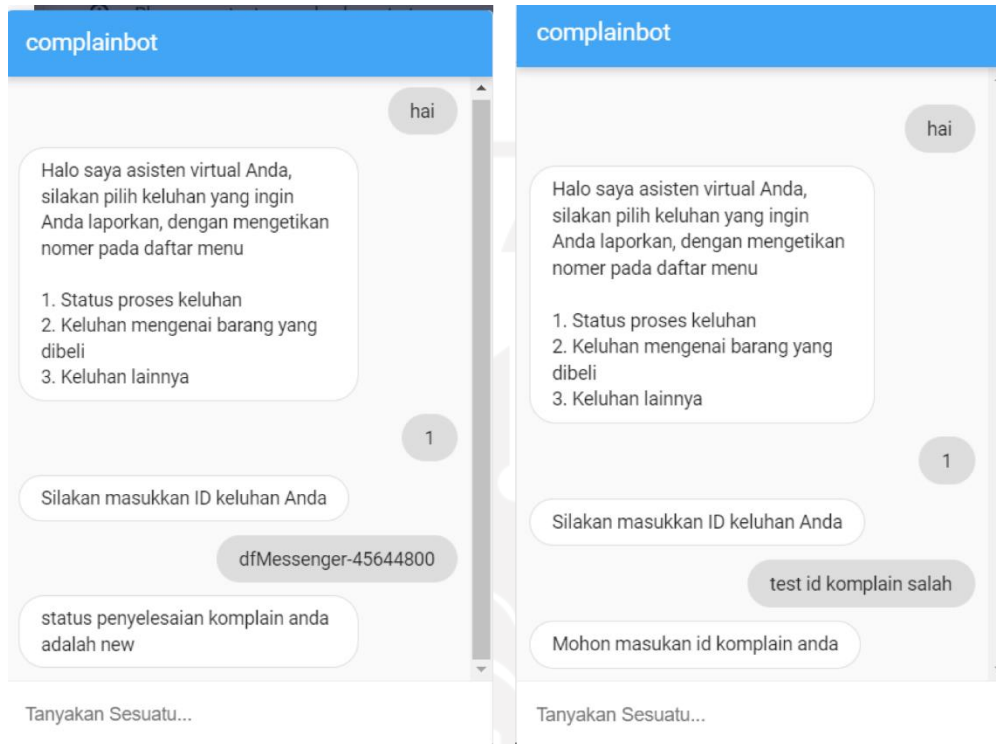
Percakapan menu menampilkan status penyelesaian komplain dimulai dengan pengguna menginputkan “hai”, *chatbot* akan menampilkan daftar menu yang dapat dipilih sesuai kebutuhan. Tampilan daftar menu dapat dilihat pada Gambar 3.28. Percakapan dilanjutkan dengan user mengetikkan angka 1 sebagai *input*. *Chatbot* akan mengkategorikan ke dalam *intent* “*complain\_status*” dan memberikan respon seperti pada Gambar 3.33.



Gambar 3.33 Tampilan Respon Intent *complain\_status*

Id komplain yang diberikan saat berhasil mengajukan laporan akan digunakan sebagai *input* saat mengecek status komplain. Status komplain akan dikategorikan ke dalam 3 jenis

yang pertama “new”, “proce”, “done”. Tampilan respon *chatbot* ketika berhasil mendapatkan data berdasarkan id komplain dapat dilihat pada sisi kiri Gambar 3.34. Sisi kanan pada Gambar 3.34. merupakan tampilan respon *chatbot* jika id komplain tidak berhasil ditemukan pada Google Sheet.



Gambar 3.34 Tampilan Respon Id Komplain Ditemukan

Data yang tersimpan akan berbentuk tabel dengan tujuh kolom, kolom yang dibuat terdiri dari *session\_id*, tanggal, nama intent, nomer telp, aduan dan status. Kolom *session\_id* merupakan id komplain yang dibuat dengan unik pada setiap user. Kolom *session\_id* akan memiliki nilai kosong apabila pengajuan komplain dilakukan dua kali menggunakan device yang sama. Kolom *session\_id* yang kosong akan merujuk pada baris sebelumnya.

Penggunaan kolom intent berfungsi untuk mengkategorikan komplain yang masuk. Kolom status berfungsi untuk mengetahui progress status penyelesaian dari sebuah komplain. Status komplain akan secara otomatis memiliki nilai new ketika pertama kali komplain diajukan, status tersebut dapat diganti secara manual melalui Google Sheet yang terhubung dengan webhook chatbot. Tampilan tabel pada Google Sheet dapat dilihat pada Gambar 3.35.

	A	B	C	D	E	F	G
1	Session_id	Tanggal	Intent	Nama	Nomer Telp	Aduan	Status
2	dfMessenger-11532773	06/03/2022-12:31:46	Broken_item - fallback	wijoyo	0810990123123	barang yang diterima rusak	new
3	d069dbaf-fd90-b912-8cd4-62b57b310f63	06/03/2022-12:45:52	wrong_item - fallback	ibnu	09090900909	barang salah dikirim	proses
4	dfMessenger-79570127	06/03/2022-13:48:12	Other_complain - fallback	joy	098760189	test other complain	new
5	76aba225-98b7-512a-4b29-9c8a8519e3f8	06/30/2022-00:50:43	Other_complain - fallback	ibnu	090909090	oke	new
6		06/30/2022-00:52:37	Other_complain - fallback	wijoyo	0909090	test	new
7	dfMessenger-320396	09/13/2022-12:43:56	Broken_item - fallback	budi	086666698765	barang diterima penyok	done
8	dfMessenger-52058311	09/19/2022-04:05:34	Broken_item - fallback	ibnu	081890098789	barang cacat	new
9	dfMessenger-61150789	09/19/2022-04:21:13	items_don't_match - fallback	ibnu	089898980	barang yg datang beda	new
10	dfMessenger-45644800	09/28/2022-07:20:05	wrong_item - fallback	joy	08123490876	barang yang saya terima pu	new
11							

Gambar 3.35 Tampilan Tabel Pada Google Sheet

### 3.3 Penutupan Proyek Implementasi Google Sheet API pada Custom Webhook

Proyek implementasi Google Sheet API pada custom webhook, ditutup dengan pengujian dan penulisan dokumen *research notes*. Proyek penelitian tersebut ditutup karena tim peneliti sudah mendapatkan tujuan awal. Tujuan awal proyek penelitian ini untuk mendapatkan dokumentasi implementasi Google Sheet API pada webhook sebuah *chatbot*. Selama melakukan penelitian proyek terdapat beberapa poin-poin yang ditemukan setelah melakukan pengujian pada webhook. Adapun poin-poin tersebut adalah sebagai berikut:

- Webhook yang dibuat menggunakan *library* Google Sheet API memiliki *response time* antara 3 – 4 detik, sehingga terkadang dapat mempengaruhi keberhasilan pemberian respon Dialogflow
- Library* Google Sheet API belum memiliki *method* untuk pencarian data, sehingga pada penerapannya masih dilakukan dengan membuat sintaks saat pengambilan data.
- Library* Google Sheet API memerlukan kredensial berupa file bertipe JSON yang didapat melalui layanan Google Cloud Platform.

## **BAB IV**

### **REFLEKSI PELAKSANAAN MAGANG**

#### **4.1 Relevansi Akademik**

Proyek yang dikerjakan selama melakukan kegiatan magang, tidak lepas dari penerapan teori pembelajaran di perkuliahan. Pelaksanaan proyek Implementasi Google Sheet API pada Custom Webhook memiliki keterkaitan dengan beberapa teori pada mata kuliah pengembangan aplikasi berbasis web. Keterkaitan tersebut ditemukan pada saat pembuatan sebuah intermediary service. Intermediary service yang dibuat berupa sebuah Endpoint API yang menggunakan arsitektur RESTful dengan method POST. Penerapan lainnya dari mata kuliah pengembangan aplikasi berbasis web adalah penggunaan konsep MVC untuk membuat sebuah API endpoint baru.

Manajemen Proyek merupakan teori lainnya yang diterapkan pada saat magang. penerapan teori ini dijumpai ketika melakukan Meeting OKR yang dilakukan setiap sekali dalam satu kuartal. Objective Key Result (OKR) menjadi tolak ukur untuk pengadaan proyek pada satu kuartal kedepan. Tahap Inisiasi dan perancangan proyek pada tempat magang memiliki kesamaan dengan teori yang diajarkan pada perkuliahan.

Selain penggunaan teori yang didapat pada perkuliahan, terdapat beberapa hal pada tempat magang yang dapat membantu untuk meningkatkan keefisienan sebuah kode pada sistem. Hal tersebut adalah membuat script unittest yang akan digunakan untuk melakukan test dan menampilkan coverage dari keseluruhan sistem atau modul. Coverage merupakan sebuah presentasi efisiensi kode dengan cara menjalankan fungsionalitas pada setiap baris.

#### **4.2 Pembelajaran Magang**

Pembelajaran yang didapat selama melakukan kegiatan magang akan dijelaskan pada subbab berikut:

##### **4.2.1 Manfaat Magang**

Selama melakukan kegiatan magang terdapat manfaat yang telah dirasakan oleh penulis. Manfaat utama yang sangat dirasakan adalah mendapatkan pengalaman langsung mengerjakan proyek dalam dunia kerja. Pengalaman ini nantinya dapat dijadikan sebagai modal awal untuk berkarir dalam industri IT. Manfaat lainnya yang dirasakan adalah peningkatan dalam hal teknis maupun non teknis.

Peningkatan dalam hal teknis dirasakan selama melakukan magang. hal ini dapat terjadi karena proyek yang dikerjakan menggunakan teknologi yang tidak di diajarkan pada

perkuliahan. Selain itu peningkatan teknis yang paling dirasakan adalah penulisan kode program yang lebih baik. Peningkatan non teknis yang didapatkan dari kegiatan magang adalah cara berkomunikasi dengan baik dan melakukan manajemen waktu dengan baik.

#### **4.2.2 Kendala, Hambatan dan Tantangan**

Selama pelaksanaan kegiatan magang terdapat beberapa hambatan dan tantangan yang dihadapi. Hambatan utama yang dirasakan adalah kesulitan beradaptasi pada saat awal mulai magang. Hambatan terjadi dikarenakan proyek yang dikerjakan menggunakan teknologi yang belum pernah dipelajari sebelumnya. Teknologi tersebut yaitu *NLP service* untuk pembuatan *chatbot*.

Hambatan lainnya dirasakan ketika melakukan research mengenai Wit.ai service dikarenakan dokumentasi mengenai pricing dan limitation yang kurang lengkap. Hambatan juga dirasakan saat melakukan pengembangan pada produk Robolabs. Hambatan tersebut dikarenakan pembuatan kode program harus mengikuti standar dari perusahaan. Kode program yang dibuat juga harus memiliki coverage di atas 75% sehingga tidak terdapat banyak kode program yang redundan.



## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Selama pelaksanaan magang di Qiscus, penulis berkontribusi pada dua proyek yaitu proyek implementasi Google Sheet API pada custom webhook dan proyek pengembangan Handover Agent 2.0. Selain pada dua proyek tersebut terdapat kontribusi lain pada pengembangan fitur di Robolabs diantaranya: pengembangan fitur Auto Tag dan penambahan dua template dengan studi kasus baru pada fitur Bot Template.

Pengerjaan proyek implementasi Google Sheet API pada custom webhook meliputi proses pembuatan conversational flow, hingga pembuatan webhook dan implementasinya pada sisi Dialogflow. Hasil yang didapat dari proyek ini menunjukkan bahwa Google Sheet dapat digunakan untuk penyimpanan data dari chatbot berbasis Dialogflow menggunakan library Google Sheet API, akan tetapi memiliki poin-poin yang harus diperhatikan. Poin utama yang harus diperhatikan adalah waktu respon time webhook yang menggunakan Google Sheet API memiliki durasi antara 3- 4 detik, hal ini dapat mempengaruhi keberhasilan Dialogflow dalam memberikan respon.

Hasil dari proyek implementasi Google Sheet API pada custom webhook dapat menjadi rujukan pertimbangan untuk developer dalam membuat sebuah service atau fitur baru. Selain menjadi rujukan pertimbangan, kode program yang dibuat juga dapat digunakan untuk webhook chatbot dengan case serupa.

#### **5.2 Saran**

Adapun saran yang diberikan setelah melakukan kegiatan magang ditujukan kepada:

- Prodi Informatika UII

Administrasi dan pemberitahuan informasi untuk pengambilan penjaluran magang sudah dilakukan dengan baik, tetapi perlu ada beberapa hal yang ditingkatkan. Pertama dalam hal administrasi diharapkan prodi dapat memberikan respon lebih cepat dalam menangani surat pengajuan magang. Saran ini bertujuan apabila mahasiswa/i membutuhkan lebih dari satu surat pengajuan magang. Hal ini dapat terjadi dikarenakan ditolak oleh perusahaan opsi pertama. Saran selanjutnya terkait dengan pemberitahuan informasi, diharapkan pemberitahuan informasi terkait hal-hal penting tidak disampaikan secara mendadak.

- Diri Sendiri

Saran untuk diri sendiri adalah perlu ditingkatkannya pemahaman terkait hal-hal teknis. Pengalaman kerja dan pembelajaran hal teknis selama magang dapat digunakan sebagai modal awal, tetapi untuk hal teknis perlu ditingkatkan terutama dalam merancang dan membuat sebuah sistem. Untuk non teknis perlu ditingkatkan terkait manajemen waktu.



## DAFTAR PUSTAKA

- Anonim. (2011). *What is Web Intermediary (WBI)? - Definition from Techopedia*.  
<https://www.techopedia.com/definition/24378/web-intermediary-wbi>
- Dwi, A. R., Imamah, F., Mei Andre, Y. S., Ilmu Komputer, F., Mercu Buana Jalan Menteng Raya No, U., & Sirih, K. (2018). APLIKASI CHATBOT (MILKI BOT) YANG TERINTEGRASI DENGAN WEB CMS UNTUK CUSTOMER SERVICE PADA UKM MINSU. *Jurnal Cendikia*, 16(2 Oktober), 100–106.  
<http://jurnal.dcc.ac.id/index.php/JC/article/view/108>
- Google. (n.d.). *Sheets API | Google Developers*. Retrieved December 4, 2022, from  
<https://developers.google.com/sheets/api>
- Komawar, O., Thakar, P., Shetty, R., Bartakke, A., & Desai, M. (2013). An Internet Relay Chat Bot using AIML. *International Journal of Science and Research (IJSR) ISSN (Online Index Copernicus Value Impact Factor, 14611(10)*.
- Oktavia, C. A. (2020). Implementasi Chatbot Menggunakan Dialogflow dan Messenger Untuk Layanan Customer Service Pada E-Commerce. *J I M P - Jurnal Informatika Merdeka Pasuruan*, 4(3). <https://doi.org/10.37438/jimp.v4i3.230>
- Qiscus. (n.d.-a). *Qiscus Omnichannel Chat*. Retrieved December 15, 2022, from  
<https://documentation.qiscus.com/omnichannel-chat>
- Qiscus. (n.d.-b). *Qiscus Robolabs*. Retrieved December 15, 2022, from  
<https://documentation.qiscus.com/robo labs>
- Turner, D. S. (2022). *APIs vs Webhooks: what are the differences? - Imburse*.  
<https://imbursepayments.com/apis-webhooks-differences/>

LAMPIRAN

