

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama Penulis melakukan magang di CODE.ID, penulis mengambil jabatan sebagai developer intern dan selama magang penulis dibimbing secara langsung oleh *Product Manager* sekaligus *Supervisor* di CODE.ID bernama Alliza beserta *IT Technical* lainnya bernama Yopi. Penulis melakukan proyek sendiri tanpa adanya tim.

Selama melakukan pengerjaan proyek ini, penulis melakukan pertemuan seminggu sekali dengan ibu Alliza selaku *Supervisor* untuk melakukan *review* dan untuk mengetahui *progress* pekerjaan yang sudah dilakukan oleh penulis.

Setiap minggu nya penulis selalu ada pertemuan secara daring pada *google meet* untuk membahas tiap-tiap pekerjaan yang telah dilakukan oleh penulis dan juga apa saja yang harus ditambahkan dan diperbaiki oleh penulis.

Akan tetapi tidak setiap pertemuan dihadiri oleh ibu Alliza dan Kak Yopi, ada juga di mana hanya Ibu Alliza yang hadir dan memberikan saran untuk perbaikan atau penambahan pada tugas yang telah dikerjakan dan juga memberikan beberapa mini proyek. Seperti pada saat itu pada mini proyek pertama penulis melakukan pertemuan secara daring bersama ibu Alliza dan Kak Yopi untuk membahas *bug* dan mereka memberikan saran untuk tugas tersebut. Kegiatan pertemuan ini dilakukan untuk mengetahui perkembangan saat proyek dikerjakan dan uji coba ketika proyek telah selesai.

3.2 Tugas yang Dilakukan

Penulis tidak langsung mengerjakan untuk membuat *chatbot*, akan tetapi penulis pertama kali diminta untuk melakukan eksplorasi perusahaan CODE.ID serta salah satu produk mereka yang bernama Activo untuk mencari tahu apa saja yang CODE.ID lakukan atau kerjakan terutama pada produk Activo. Setelah itu penulis diberikan waktu untuk mempelajari apa itu *Twilio* dan bagaimana cara menggunakan *Twilio*. Lalu selanjutnya penulis diberi waktu untuk mempelajari cara membuat *chatbot* dengan cara melihat *tutorial* pada *YouTube*.

Setelah melihat cara membuat *Chatbot* pada *YouTube*, penulis mencoba

bermain main terlebih dahulu dengan *Twilio* serta mencoba membuat *chatbot* sederhana pada Python untuk mencari tahu seperti apa cara kerja kedua tools tersebut. bermain dengan keduanya penulis selanjutnya membuat Flowchart dan prototype menggunakan figma. Selain itu penulis juga membuat *Powerpoint* untuk dipresentasikan kepada *supervisor* serta *technical IT* untuk memberikan gambaran seperti apa nanti akan dibuat dan juga untuk menerima saran apakah sudah sesuai yang dibutuhkan atau belum.

Setelah melakukan presentasi dan sudah disetujui oleh perusahaan. Selain pengerjaan *chatbot* penulis juga memiliki beberapa proyek kecil, sehingga setelah presentasi yang telah dilakukan disetujui oleh perusahaan penulis, penulis tidak langsung mengerjakan proyek tersebut, penulis diminta untuk melakukan *mini project* sebanyak 3 kali yaitu membuat sebuah *minigame* untuk situs web Activo, membuat *GeoMap* dan yang terakhir membuat aplikasi yang dapat menerima *software* serta *hardware* user dengan menggunakan Python.

Proyek yang pertama adalah pembuatan sebuah permainan yang sederhana untuk ditempatkan ke dalam situs web Activo. Permainan ini dinamakan *shooting range* di mana, pemain harus menembak sejumlah target yang telah diberikan dengan waktu yang telah ditentukan. Target yang telah ditembak oleh pemain akan berubah warna menjadi abu dan pemain akan memperoleh skor. Ketika waktu yang telah ditentukan habis maka akan dimunculkan layar permainan selesai dengan skor yang telah diperoleh para pemain. Akan tetapi permainan ini tidak dapat dimainkan kembali ketika selesai dimainkan, untuk memainkannya kembali pemain harus menutup permainan tersebut dan masuk kembali ke dalam permainan. Tujuan dibuatnya permainan ini adalah agar pengguna situs web Activo tidak merasa bosan ketika menunggu proses yang sedang dijalankan oleh situs web tersebut. Permainan ini dibuat dengan menggunakan *Visual Studio Code* dengan *HTML* dan *SCSS*.

Proyek yang selanjutnya adalah membuat sebuah *GeoMap* yang berguna untuk para pengguna produk Activo agar dapat mengetahui di mana saja aset mereka berada dengan cara melihat melalui map ini. *GeoMap* ini memiliki 2 halaman di mana halaman pengguna atau klien hanya dapat melihat di mana aset mereka, dan ada halaman admin di mana admin dapat menempatkan aset ke map dan data tersebut akan tersimpan dalam basis data yang telah disediakan. Pembuatan *GeoMap* ini menggunakan *PHP* dan basis data *MySQL* untuk penyimpanan data.

Proyek ketiga adalah membuat sebuah aplikasi komputer yang dapat memperoleh data *software* serta *hardware* pada komputer klien. Cara kerja aplikasi ini pengguna hanya perlu mengunduh aplikasi yang telah dibuat dan menjalankan apli-

kasi tersebut, maka aplikasi bisa langsung mendapatkan dan menyimpan data ke *notepad*. Aplikasi pengambil informasi ini dibuat dengan menggunakan Python.

Setelah proyek-proyek telah selesai dikerjakan penulis, selanjutnya penulis membuat *chatbot* dengan ketentuan yang sudah pernah disetujui sebelumnya. Penulis diminta untuk membuat *chatbot* yang dapat membuat sebuah aset dengan melemparkan pertanyaan-pertanyaan kepada pengguna dan juga dapat memeriksa aset pengguna pada aplikasi *WhatsApp*.

3.3 Uraian Pelaksanaan Magang

Berikut ini adalah uraian pelaksanaan magang yang dilakukan oleh penulis selama bekerja magang di CODE.ID produk Activo selama 6 bulan. Uraian pelaksanaan akan dibuat berbentuk tabel untuk mingguan dan penjelasan setelah tabel. Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Perkenalan mengenai perusahaan CODE.ID dan Activo, melakukan explore pada website CODE.ID dan Activo, memberitahukan tugas yang akan dilaksanakan. Pembelajaran cara pembuatan <i>chatbot</i> pada Python.
2	Membuat PPT mengenai apa saja yang dibuat dan apa saja yang dibutuhkan, mempelajari cara penggunaan Twilio serta pemasangan semua <i>software</i> yang dibutuhkan.
3 - 5	Proyek kecil 1 yaitu proyek untuk membuat <i>mini game</i> untuk sebuah situs web Activo.
6 - 8	Proyek kecil 2 yaitu proyek membuat sebuah <i>GeoMap</i> untuk peta untuk Activo.
9 - 12	Proyek kecil 3 yaitu membuat aplikasi untuk mengetahui <i>software</i> yang digunakan pengguna menggunakan Python.
13 - 17	Pembuatan <i>chatbot</i> dengan Python menggunakan API Twilio.

Tabel 3.1 merupakan uraian singkat dari penulis selama kerja magang di CODE.ID produk Activo. Uraian tersebut belum lengkap dikarenakan penulis masih bekerja hingga 17 Januari 2022.

Pada minggu pertama, penulis melakukan pengenalan CODE.ID dan Activo, kemudian penulis di beritahu oleh *supervisor* yaitu ibu Alliza apa saja tugas tugas yang akan dilakukan, selain itu juga penulis diperkenalkan juga dengan *website* perusahaan tersebut. Setelah melakukan perkenalan, penulis diminta untuk membuat akun *GitLab* dikarenakan akun tersebut akan menjadi tempat di mana tugas-tugas diberikan dan cara kita berkomunikasi selain melalui surat elektronik dan *Google Meet*. Pengenalan ini dilakukan secara daring menggunakan *Google Meet*, setelah pengenalan selesai penulis diminta untuk mempelajari cara membuat *WhatsApp Chatbot* pada Python.

Minggu kedua setelah mempelajari pembuatan *chatbot* penulis diminta untuk membuat sebuah powerpoint dan mempresentasikan powerpoint tersebut yang berisi seperti apa *chatbot* yang akan dibuat dan peralatan apa saja yang akan digunakan oleh penulis kepada ibu Alliza saat di pertemuan mingguan. Selain itu pada minggu kedua juga penulis melakukan *install* untuk peralatan-peralatan yang diperlukan di komputer penulis.

Pada minggu ketiga sampai dengan minggu kelima, penulis diberikan proyek kecil yaitu diminta untuk membuat sebuah *minigame* untuk situs web di Activo. Proyek kecil ini boleh menggunakan peralatan apa pun. Pada minggu ketiga seperti sebelumnya, penulis melakukan eksplorasi ulang *website* Activo, lalu penulis mencari ide untuk pembuatan *minigame*. Pada minggu keempat penulis memberikan 3 buah *minigame* dan membuat 3 buah *prototype minigame* menggunakan *figma* dan mempresentasikannya untuk ibu Alliza memilih *minigame* mana yang ingin dipilih. Pada minggu kelima penulis membuat *minigame* dari ide yang telah ditentukan oleh ibu Alliza yaitu *shooting range* menggunakan **HTML** beserta **SCSS**.

Pada minggu keenam hingga minggu kedelapan, penulis diberikan proyek kecil kedua yaitu dimana penulis diminta oleh ibu Alliza untuk membuat sebuah *GeoMap* secara dinamis. Penulis membuat *geomap* menggunakan **PHP** dan **MySQL**. Pada minggu keenam penulis mempelajari cara membuat *GeoMap* melalui *YouTube* dengan menggunakan API **Mapbox**, akan tetapi belum dinamis dimana ketika memasukkan lokasi dan memasukkan marker harus melakukan *edit* satu per satu. Pada minggu kedelapan penulis mengubah mengubah *GeoMap* menjadi dinamis menggunakan **PHP** dan **MySQL**.

Pada minggu kesembilan hingga minggu keduabelas, penulis diberikan proyek kecil ketiga yaitu membuat sebuah program aplikasi untuk mengetahui *software* dan *hardware* user. User membuat program tersebut menggunakan Python,

serta perubahan Python menjadi file *.exe* pun dilakukan dengan *library* yang ada pada Python.

Pada minggu ketigabelas hingga minggu ketujubelas penulis mulai mengerjakan pembuatan *chatbot* pada Whatsapp dimulai dari perancangan flowchart, pembuatan database, pemrograman hingga melakukan ***Black-Box testing***.

3.3.1 Sistem yang digunakan

Dalam proses perancangan dan pembuatan *chatbot* penulis menggunakan hardware dan software di bawah ini untuk merancang dan membuat Chatbot:

A. Hardware

Berikut Hardware yang digunakan oleh penulis ketika membuat *chatbot*:

1. Sistem Operasi(OS) : Windows 10 Enterprise 64-bit
2. Processor : AMD Ryzen 5 2600 Six-Core Processor
3. RAM : 16GB
4. Graphic Card: Radeon RX 580 Series

B. Software

Berikut adalah *software* yang digunakan oleh penulis ketika membuat *ChatBot*:

- Visual Studio Code 1.62
- Twilio
- WhatsApp
- Python v3.8.10
- Gitlab

3.3.2 System Requirement

Bedasarkan dari apa yang diminta oleh ibu Alliza, berikut adalah *requirement* yang dibutuhkan untuk membuat *chatbot*:

A. Check Data Asset

Chatbot akan di-trigger oleh *user* dengan mengirim pesan 'check asset' atau mengetik kan angka 1 jika *chatbot* ada menggunakan angka.

Sistem akan mencari data berdasarkan yang diketik di *Whatsapp* lalu akan menampilkan data tersebut.

B. Create Data Asset

Chatbot ini akan dijalankan oleh pengguna dengan cara mengirim pesan 'create asset' atau mengetik kan angka 2 jika *chatbot* ada menggunakan angka. Sistem akan memberikan instruksi kepada pengguna untuk mengirimkan / memberi data kepada bot untuk informasi yang diminta (seperti: ambil gambar aset, ketik nama aset, pilih tipe aset, pilih lokasi, selesai).

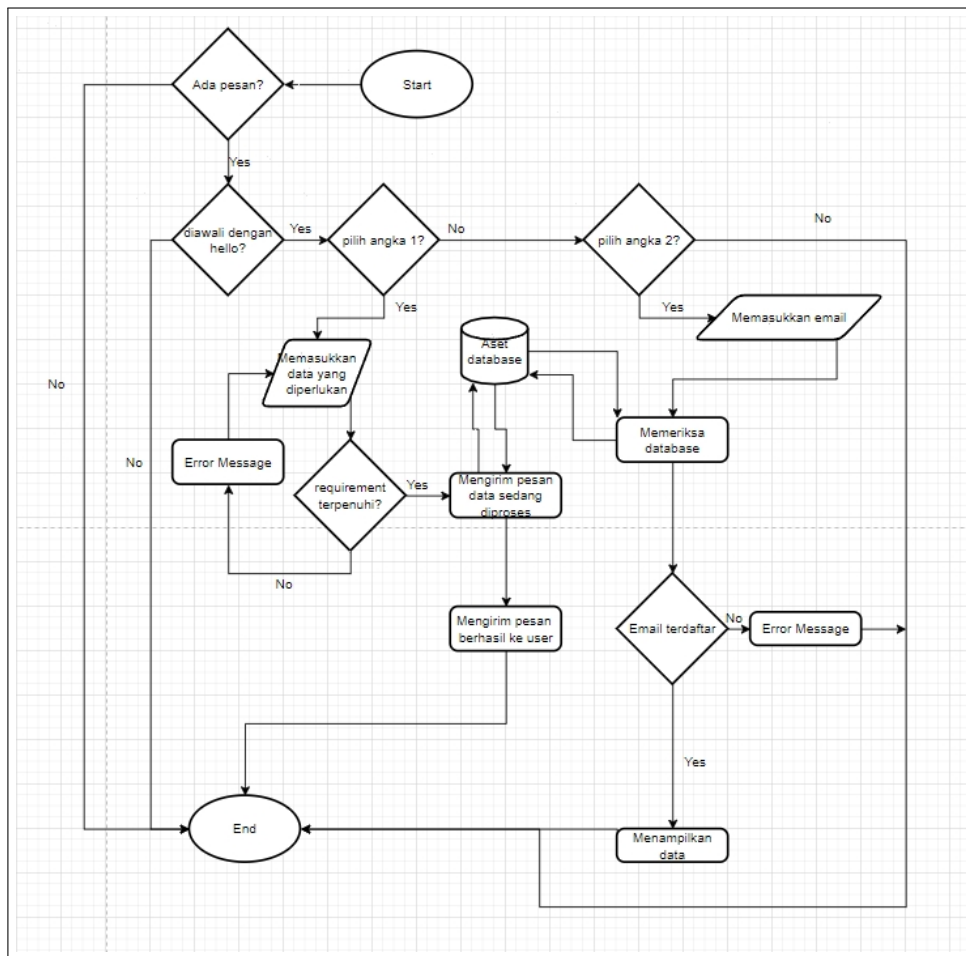
3.3.3 Perancangan

Bagian perancangan ini merupakan bagian atau alur di mana penulis mengerjakan pembuatan *Chatbot*. Pada bagian perancangan terdapat bagian bagan alur, basis data, implementasi serta pengujian bot yang dibuat.

A. Flowchart

Alur kerja dari *chatbot* akan digambarkan ke dalam sebuah flowchart. Flowchart yang dibuat merupakan flowchart secara keseluruhan.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.1. Flowchart dari chatbot

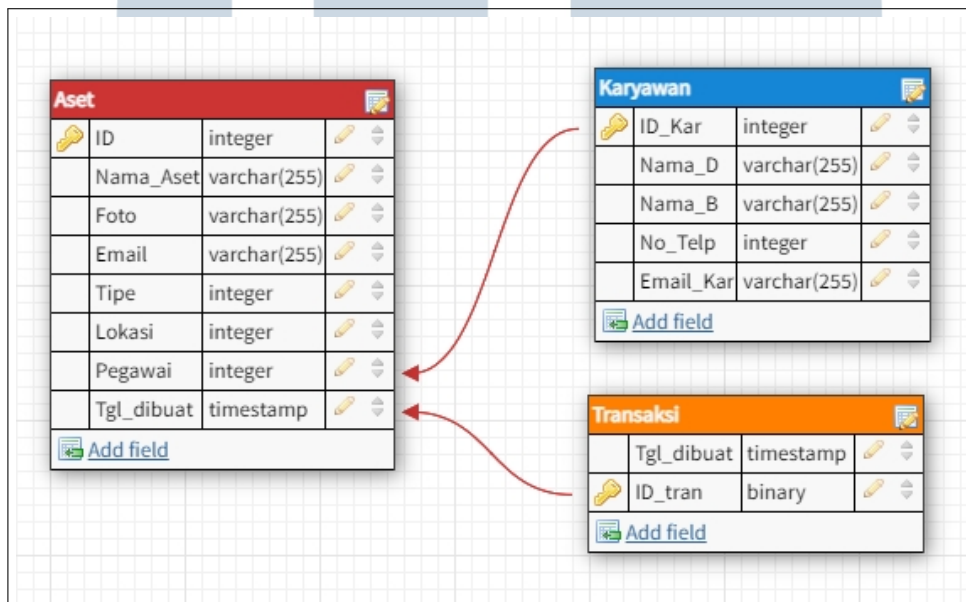
Pada Gambar 3.1. dapat dilihat, pertama kali bot akan memeriksa apakah ada pesan atau tidak, jika ada akan dilakukan pengecekan apakah pesan tersebut dibuka dengan kata *hello* atau tidak, jika tidak maka langsung berakhir atau bot tidak membalas sekali. Jika diawali dengan kata *hello* maka bot akan membalas pesan tersebut dengan mengatakan *hi* kembali dan memberikan menu. Jika pengguna mengetik angka 1 maka akan masuk dan diberikan pertanyaan pertanyaan yang harus diisi untuk melengkapi data, jika semua pertanyaan sudah terpenuhi maka bot akan menyimpan data tersebut pada basis data dan bot akan memberikan pesan berhasil, akan tetapi jika tidak terpenuhi maka bot akan memunculkan pesan *error* dan pengguna harus memenuhi ketentuan tersebut.

Jika pilih angka 2 maka akan masuk ke tahap *check asset* di mana pengguna akan ditanya apa email pengguna, dan jika email

tersebut ada dalam basis data maka bot akan memunculkan data dari basis data, data yang dimunculkan berupa: Nama aset, foto aset, tipe, lokasi, dan nama karyawan yang menangani aset tersebut. Akan tetapi jika email yang dimasukkan tidak ada dalam basis data maka bot akan mengirimkan pesan *error*.

B. Database

Berikut merupakan Skema dari basis data yang dirancang untuk membuat *chatbot* pada Gambar 3.2.



Gambar 3.2. Skema basis data dari *chatbot*

3.3.4 Implementasi

Pada tahapan ini, sebelum membuat aplikasi *chatbot* pertama kita harus menyiapkan terlebih dahulu *virtual environment* pada Python. Penulis membuat *virtual environment* pada *directory* yang berbeda dengan proyek *chatbot* yang akan dibuat, setelah itu penulis melakukan peng-*install*-an *Python package* yang dibutuhkan untuk pembuatan *chatbot* nanti[2].


```
1 | $ mkdir whatsapp-bot
2 | $ cd whatsapp-bot
3 | $ python3 -m venv whatsapp-bot-venv
4 | $ whatsapp-bot-venvScripts\activate
5 | (whatsapp-bot-venv) $ pip install twilio flask requests
```

Gambar 3.3. Perintah untuk melakukan *install Python package*
sumber: Visual Studio Code penulis

Perintah terakhir yang memiliki tulisan ***pip*** merupakan perintah yang akan melakukan *install Python package* yang akan kita gunakan pada proyek ini, yang isinya adalah :

1. Flask framework
2. Twilio Python Helper Library
3. Requests package

Ketika selesai maka akan muncul sebuah file yang memiliki format *.txt* yang bernama *requirement.txt* di dalam *requirement* tersebut berisi seperti Gambar 3.4.



```
☰ requirements.txt
1  certifi==2021.10.8
2  charset-normalizer==2.0.8
3  click==8.0.3
4  colorama==0.4.4
5  Flask==2.0.2
6  idna==3.3
7  itsdangerous==2.0.1
8  Jinja2==3.0.3
9  MarkupSafe==2.0.1
10 PyJWT==2.3.0
11 pytz==2021.3
12 requests==2.26.0
13 twilio==7.3.2
14 urllib3==1.26.7
15 Werkzeug==2.0.2
```

Gambar 3.4. Hasil ketika Python *package* telah berhasil dipasang
sumber: Visual Studio Code Penulis

Setelah melakukan penginstalan *Python package*, selanjutnya membuat *Flask chatbot service* dengan cara melakukan *import flask*. Lalu selanjutnya melakukan *webhook*, *webhook* pada Twilio API untuk Whatsapp ini berguna untuk mengaktifkan notifikasi ketika ada pesan yang masuk. Untuk Twilio agar bisa melakukan komunikasi dengan aplikasi *chatbot* kita harus menentukan titik akhir yang akan di konfigurasi sebagai *webhook*. Dengan kerangka *Flask*, sangat mudah untuk mendefinisikan *webhook*. Gambar 3.5 merupakan *command* untuk melakukan *webhook* dengan menggunakan *Flask[2]*.

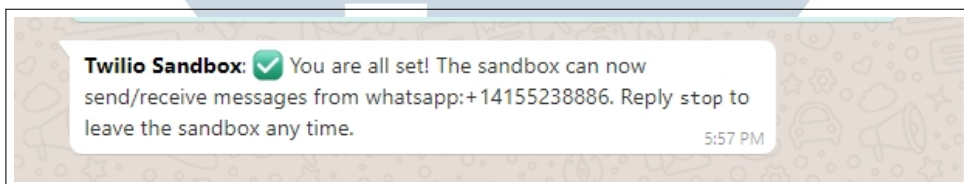
```

1  from flask import Flask
2
3  app = Flask(__name__)
4
5
6  @app.route('/bot', methods=['POST'])
7  def bot():
8      # add webhook logic here and return a response
9
10
11  if __name__ == '__main__':
12      app.run()

```

Gambar 3.5. Perintah untuk melakukan *webhook* dengan *Flask*
sumber: Visual Studio Code Penulis

Ketika telah selesai melakukan *webhook* maka selanjutnya kita mengirimkan pesan kepada Twilio melalui whatsapp, dan Twilio akan mengirimkan kembali pesan bahwa kita sudah selesai melakukan *setup* dan sudah dapat menggunakan Twilio sebagai aplikasi *chatbot* kita[2].



Gambar 3.6. tampilan awal ketika *setup* telah berhasil

Setelah *setup* selesai dilakukan, selanjutnya membuat *codingan* untuk menu *chatbot*. Dalam rancangan ini, whatsapp bot memiliki 2 menu yang menjadi *requirement*. Detail menu dapat dilihat pada Tabel 3.2.

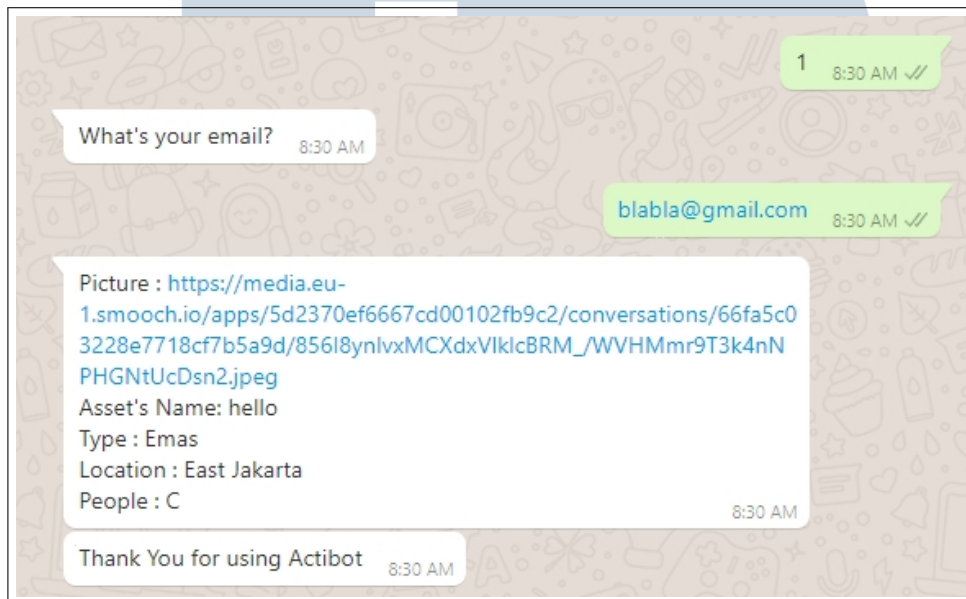
Tabel 3.2. Menu yang disediakan

Menu	Keterangan
Check Asset	Memberikan informasi mengenai foto aset, nama aset, lokasi, dan status.
Create Asset	Membuat serta menyimpan informasi seperti: foto aset, nama aset, lokasi, dan tipe aset.

A. Implementasi Check Asset

Pada *check asset*, yang digunakan adalah sistem *fetch*, di mana terjadinya proses pengiriman *request* dari *whatsapp* pengguna,

lalu *request* tersebut akan dibawa menuju ke basis data melalui API Twilio yang merupakan API legal dan sudah memiliki kontrak dengan **Meta** (sebelumnya **facebook**). Selanjutnya basis data akan memberikan data yang di-*request* dan data yang didapat dari basis data dirangkai menjadi susunan kalimat. Susunan kalimat tersebut yang akan dikirim kembali kepada pengguna seperti pada Gambar 3.7[3].



Gambar 3.7. Hasil *check* aset

B. Implementasi Create Asset

Pada *create asset endpoint* yang digunakan adalah *endpoint POST create asset*. *Endpoint* ini merupakan fitur yang digunakan untuk mengirim data ke Tabel *create asset*. Cara kerja *endpoint* ini dilakukan dengan *query* yang digunakan untuk menyimpan data ke dalam Tabel aset.

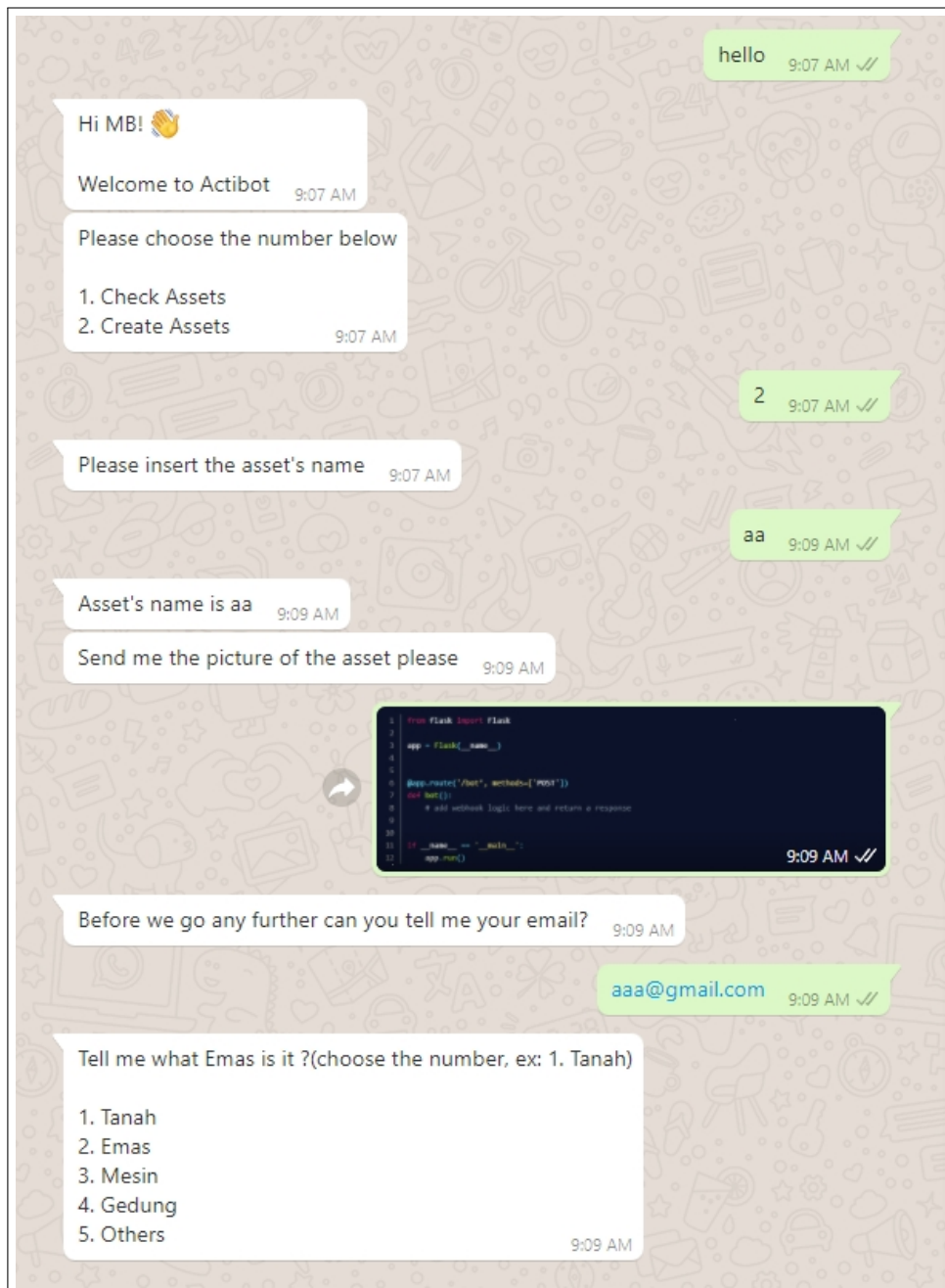
Tabel 3.3 merupakan rincian untuk *endpoint* dan *request body* untuk menjalankan *create asset*.

Tabel 3.3. Rincian API *Create Aset*

Endpoint	POST /aset
Request body	<ul style="list-style-type: none"> • Asset Name: string (required) • Asset Picture: string (required) • Email: String (required) • type : int (required) • location: int (required) • employee: int (required)

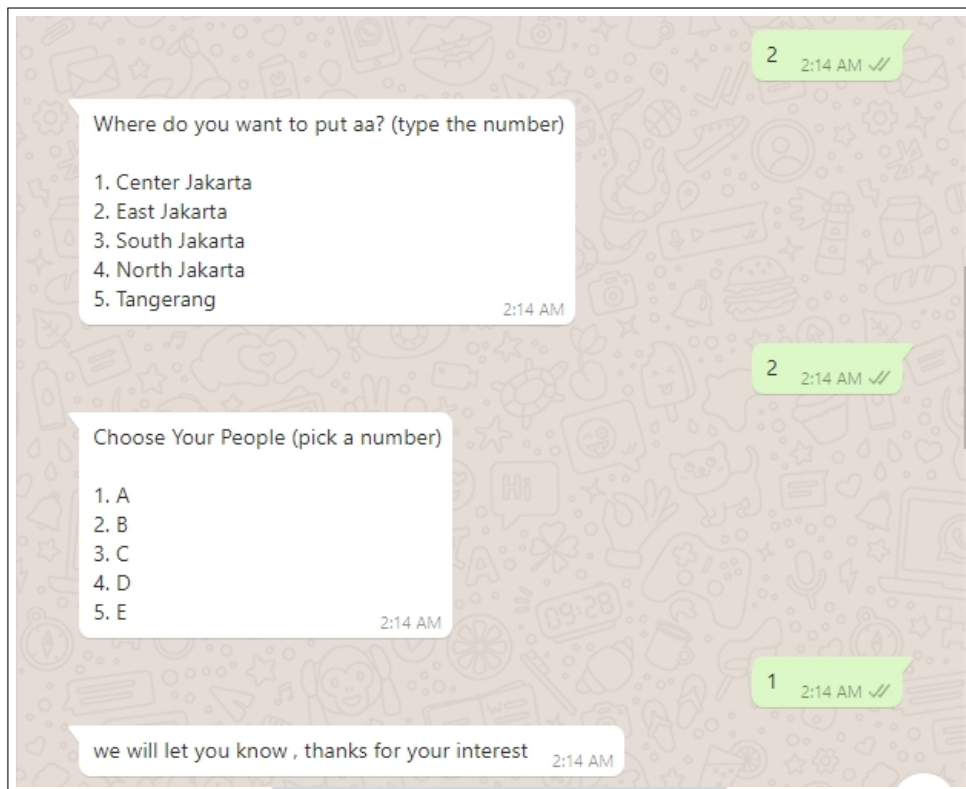
Setelah sistem sudah dirancang dan di implementasi kan, akan diperoleh sebuah hasil seperti pada tangkapan layar yang berada pada Gambar 3.8. dan 3.9.





Gambar 3.8. Hasil *create* aset

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.9. Lanjutan hasil *create* aset sumber: Whatsapp penulis

Pada Gambar 3.8. dapat diketahui bahwa untuk memulai percakapan diawali dengan 'hello' ke *bot* dan *bot* akan membalas dengan sapaan dan menu yang dapat di akses oleh pengguna. Selanjutnya untuk membuat sebuah aset baru pengguna dapat mengetik angka 1 dan *bot* akan membalas dengan pertanyaan pertanyaan untuk memenuhi Hasil yang telah terbuat akan tersimpan dalam *database* aset seperti yang ditunjukkan pada Gambar 3.10.

ID	Nama_Aset	Foto	Email	Tipe	Lokasi	Pegawai	Tgl_dibuat
1	aa	https://media.eu-1.smooch.io/apps/5d2370ef6667cd00...	aaa@gmail.com	2	2	1	2021-12-13 14:24:06

Gambar 3.10. Data yang telah dibuat akan tersimpan ke dalam database

C. Pengujian

Pengujian *black-box* tentang fitur-fitur yang ada pada bot akan ditulis dalam bentuk tabel seperti pada tabel 3.4 [4].

Tabel 3.4. Hasil pengujian dari *Chatbot*

Keterangan	Fitur
Berhasil	Menerima dan memberi sapaan kepada pengguna
Berhasil	Memberi menu kepada pengguna
Berhasil	Memberikan pertanyaan untuk pengisian data
Berhasil	Memberikan pesan error ketika data tidak sesuai dengan kebutuhan
Berhasil	Menyimpan data pada basis data
Berhasil	menampilkan data yang telah tersimpan
Berhasil	Memberikan pesan error kepada pengguna

3.4 Kendala dan Solusi yang Ditemukan

Setiap pekerjaan pasti ada suatu kendala yang kita temukan, dan setiap kendala pasti ada jalan keluar atau solusinya. Berikut Kendala dan Solusi yang penulis hadapi selama bekerja di CODE.ID produk Activo.

3.4.1 Kendala yang Ditemukan

Berikut adalah kendala yang penulis temukan selama pengerjaan *chatbot* untuk Activo:

1. Kesulitan ketika menghadapi *bug* dikarenakan penulis bekerja tanpa tim, *bug* yang dialami oleh penulis yaitu bot tidak dapat membalas serta bot tidak dapat menyimpan data.
2. Kesulitan dalam memahami keinginan pengguna saat membuat proyek seperti pengguna bot harus melakukan *scan barcode* untuk memeriksa aset.

3.4.2 Solusi yang Diberikan

Meski pun ada kendala yang ditemukan saat bekerja, penulis juga menemukan solusi dari kendala-kendala yang dihadapi. Solusi yang dilakukan adalah sebagai berikut:

1. Melakukan pertemuan untuk berdiskusi langsung dengan *supervisor* dan *technical IT* yang bekerja pada perusahaan tersebut.
2. Bertanya secara langsung melalui *Google Meet* kepada pembimbing lapangan mengenai apa saja dan seperti apa yang dibutuhkan, serta berdiskusi apakah itu pilihan terbaik atau tidak.