



ANALISIS DAN PERANCANGAN APLIKASI CHATBOT DALAM PELAYANAN PENERIMAAN MAHASISWA BARU PADA PERGURUAN TINGGI

Muhammad Rizki Herfian¹, Ahmad Rio Adriansyah²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
iyanvermilian@gmail.com, arasy@nurulfikri.ac.id

Abstract

This study aims to design and create a Chatbot that can answer general questions about New Student Admissions at Sekolah Tinggi Teknologi Terpadu Nurul Fikri (STT-NF). The benefit of this research is to help users or prospective new students to get information on New Student Admissions (PMB) at STT-NF. The research begins with Chatbot System Analysis and Literature Study on Chatbot Creation. Then the Chatbot is implemented using the Telegram Messaging API using the Python programming language. The program test using the Black Box Testing method while testing the user using the User Acceptance Testing (UAT) and Questionnaire methods. Based on the study results, it can conclude that the author has succeeded in designing and creating a Chatbot that can help users or prospective new students obtain information on New Student Admissions at the Sekolah Tinggi Teknologi Terpadu Nurul Fikri (STT-NF).

Keywords: Chatbot, Messaging API Telegram, Python, Admission of New Students, Web Service

Abstrak

Tujuan penelitian ini adalah untuk merancang dan membuat *Chatbot* yang dapat menjawab pertanyaan-pertanyaan umum seputar informasi Penerimaan Mahasiswa Baru (PMB) pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri (STT-NF). Manfaat dari penelitian ini yaitu membantu pengguna atau calon mahasiswa baru untuk mendapatkan informasi Penerimaan Mahasiswa Baru (PMB) pada STT-NF. Penelitian diawali dengan Analisis Sistem *Chatbot* dan Studi Literatur Pembuatan *Chatbot*. Kemudian *Chatbot* diimplementasikan menggunakan *Messaging API Telegram* dengan menggunakan bahasa pemrograman Python Program diuji dengan menggunakan metode Black Box Testing sedangkan pengujian kepada *user* menggunakan metode *User Acceptance Testing* (UAT) dan Kuesioner. Berdasarkan hasil penelitian dapat disimpulkan bahwa penulis telah berhasil merancang dan membuat sebuah *Chatbot* yang dapat membantu pengguna atau calon mahasiswa baru untuk mendapatkan informasi Penerimaan Mahasiswa Baru (PMB) pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri.

Kata kunci: *Chatbot, Messaging API Telegram, Python, Penerimaan Mahasiswa Baru, Web Service*

1. PENDAHULUAN

Setelah menyelesaikan jenjang pendidikan sekolah, ada pilihan untuk melanjutkan pendidikan di perguruan tinggi atau universitas. Sebelum memutuskan untuk mendaftar ke suatu perguruan tinggi atau universitas diperlukan informasi tentang seputar universitas tersebut seperti jurusan serta peminatan apa saja yang ada, lokasi kampus, biaya administrasi, cara pendaftaran, fasilitas yang tersedia, kegiatan dan agenda apa saja yang ada pada suatu universitas.. Untuk mendapatkan informasi yang dibutuhkan tentang universitas salah satunya dengan cara melihat informasi yang tersedia dari *website* resmi universitas tersebut. Tetapi untuk informasi yang lebih

detail harus menanyakan langsung ke kampus atau dengan menelpon. Kampus juga menyediakan *Live Chat* untuk menjawab pertanyaan-pertanyaan seputar kampus. *live chat* merupakan suatu layanan yang memungkinkan pelanggan/pengunjung untuk bertukar pesan dengan *Chat Agents* (orang yang menangani pertanyaan pelanggan/pengunjung) secara langsung [1]. Menurut *Econsultancy*, 79% konsumen mengatakan mereka lebih memilih fungsi *live chat* karena pertanyaan mereka dijawab dengan segera [2]. Namun layanan *live chat* tidak tersedia 24/7 jam karena dioperasikan oleh manusia yang membutuhkan istirahat.

Chatbot dapat menggantikan peran manusia dalam menjawab pertanyaan *user*, *Chatbot* merupakan layanan obrolan yang dijalankan oleh *robot* atau tokoh virtual untuk bisa melakukan percakapan otomatis dengan *user*. Dalam *Chatbot* diberikan kecerdasan buatan atau *AI (Artificial Intelligence)* yang memiliki kemampuan untuk menirukan percakapan seperti sedang *chatting* atau berbicara dengan manusia. Kemampuan *AI* yang memberikan kecerdasan virtual kepada *Chatbot* membuatnya bisa menilai jawaban yang paling tepat untuk pertanyaan yang diberikan kepadanya. Cara yang dipakai *Chatbot* agar bisa mendapatkan jawaban yang tepat adalah melalui *keyword*. Dengan sistem pengoperasian yang sudah ditanamkan kepadanya, maka *Chatbot* bisa melihat *keyword* yang diberikan dan melihat mana jawaban yang paling dekat dengan *keyword* tersebut sehingga setiap pertanyaan yang diberikan bisa dijawab dengan cepat dan tepat, yang dapat berinteraksi dengan pengguna melalui antarmuka *chat*. *Chatbot* dapat berfungsi tanpa pengawasan manusia, dan *online* selama mungkin 24/7 jam [3].

Jurnal ini akan membahas tentang perancangan *Chatbot* berbasis *Telegram*. Pengembangan *Chatbot* berbasis *Telegram* ini akan diimplementasikan untuk memberikan informasi dan menyediakan layanan di Sekolah Tinggi Teknologi Terpadu Nurul Fikri (STT-NF). *Chatbot* ini akan membantu pelayanan kampus menjadi lebih efektif khususnya layanan penerimaan mahasiswa baru (PMB) dalam melayani calon pendaftar mahasiswa baru dengan menjawab dan memberikan informasi yang diminta atau ditanyakan oleh calon pendaftar yang memerlukan informasi kampus secara lengkap. Aplikasi *Telegram* dipilih karena aplikasi ini gratis, ringan dan multiplatform. *Telegram* juga memiliki *Bot API* yang cukup lengkap dan makin berkembang, sehingga memungkinkan untuk membuat *Bot* pintar yang dapat merespon pesan dari pengguna [4].

2. PENELITIAN TERKAIT

Pada penelitian yang dilakukan oleh [5], telah diidentifikasi bahwa terdapat ketertarikan yang besar terhadap penggunaan *Chatbot* untuk mengotomatisasi layanan. Hal ini didukung dengan perkembangan layanan *messenger* yang terus berkembang ditambah lagi kemajuan di bidang kecerdasan buatan yang membawa kepada penggabungan layanan *Chatbot* dan implementasi teknik kecerdasan buatan.

Hal ini pun sudah diimplementasikan pada [6] yang membuat *Bot Telegram* yang mampu mengklasifikasikan jenis daun berdasarkan gambar daun yang diunggah ke *Bot Telegram*. Aplikasi yang dinamakan *Botanicum* dapat mengidentifikasi 20 jenis spesies daun yang hidup di Rusia. Akurasi klasifikasi yang didapatkan mencapai 97.8%. Pemilihan *messenger Telegram* didasarkan pada kemudahan yang disediakan oleh *Telegram* bagi *developer* untuk membuat *bot* dengan fungsionalitas yang beragam.

Pada penelitian lainnya, Nufusula [7] mengembangkan *Chatbot Telegram* untuk melayani pengisian pulsa secara otomatis. Pengguna dapat mengisi pulsa dari berbagai operator dengan hanya memasukkan perintah-perintah yang sudah disediakan. Metode yang digunakan pada penelitian ini adalah menggunakan metode *Long Polling*.

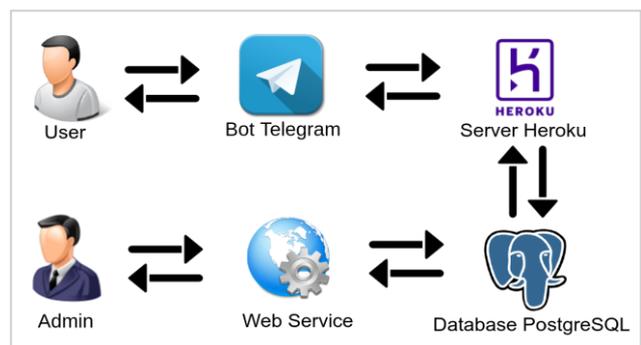
Dari penelitian-penelitian yang sudah dijelaskan di atas, bahwa penggunaan *Telegram* sebagai media untuk mengembangkan *bot* dengan berbagai fungsi sangat mungkin dilakukan karena telah disediakan *BOT API (Application Programming Interface)* yang dapat digunakan untuk pengembangan berbagai *bot* dengan fungsi yang berbeda-beda. Selain itu kemudahan dalam implementasi dan dapat berjalan pada berbagai macam platform juga menjadi pertimbangan pemilihan *Telegram* [8].

3. METODE PENELITIAN

Pada bagian ini akan menjelaskan mengenai metode penelitian, tahapan penelitian, arsitektur *Chatbot*, jenis sumber data dan metode pengumpulan data.

Penelitian ini menggunakan metode *R&D (Research and Development)* karena memiliki karakteristik adanya produk yang dihasilkan dari penelitian ini yaitu berupa *Chatbot*. Tahapan penelitian meliputi analisis kebutuhan, studi literatur, pembuatan *Chatbot*, uji coba lapangan dan revisi hasil uji coba, uji kelayakan, revisi produk akhir dan implemntasi produk.

Terdapat beberapa teknologi seperti *Telegram* yang merupakan aplikasi dimana *Bot* berjalan yang dapat dilihat dan diakses oleh *User*, *Python (Python Telegram Bot)*, *Server Heroku* dan *PostgreSQL* untuk pengaturan respon dari *request* yang dilakukan oleh *User* melalui *Bot Telegram*, serta *website* yang hanya dapat diakses oleh *Admin* untuk melakukan pengiriman pesan ke *User* melalui *API* yang disediakan oleh *Telegram*. Arsitektur *Chatbot* dapat dilihat pada Gambar 1.



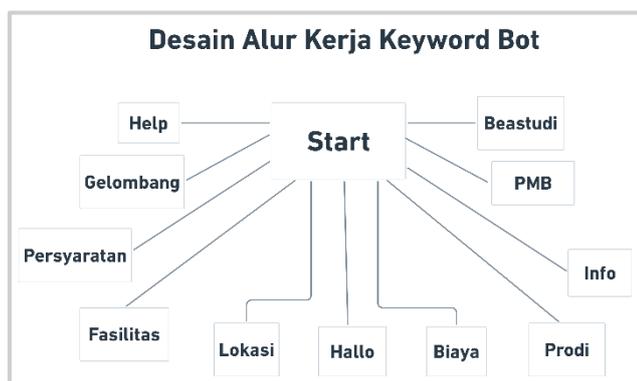
Gambar 1. Arsitektur *Chatbot Telegram*

Pengumpulan data dilakukan dengan melibatkan bagian akademik sebagai penyedia layanan informasi. Kemudian diidentifikasi apa saja layanan yang perlu disediakan di *Chatbot* yang akan dikembangkan. Data yang akan digunakan pada penelitian ini adalah semua informasi yang dimiliki oleh kampus khususnya informasi yang dibutuhkan bagi para calon pendaftar mahasiswa baru. Kemudian data foto yang diambil langsung dari lokasi penelitian dengan metode dokumentasi untuk memberikan informasi yang lebih spesifik terkait fasilitas apa saja yang tersedia di kampus. Tempat yang akan dokumentasi adalah tempat-tempat yang memiliki fungsi sebagai sarana fasilitas.

4. HASIL DAN PENGUJIAN

Pada bab ini akan menjelaskan mengenai implementasi perancangan *Chatbot* yang memiliki fungsi untuk memberikan informasi yang berkaitan dengan kampus dengan adanya *Chatbot* ini diharapkan dapat membantu pihak kampus dalam pelayanan pemberian informasi kepada calon mahasiswa baru khususnya informasi seputar Penerimaan Mahasiswa Baru (PMB).

Terdapat *keyword* yang akan menjadi *command* utama untuk menjalankan *Chatbot* antara lain informasi penerimaan mahasiswa baru (PMB), Persyaratan, Gelombang, Info kampus, Lokasi, Biaya, Fasilitas Kampus, Prodi yang tersedia dan Info Beasiswa., alasan mengapa *keyword* tersebut yang dijadikan *command*, karena dari fungsi utama *Chatbot* itu sendiri adalah memberikan informasi umum kampus kepada calon mahasiswa baru. Dari riset yang penulis lakukan dapat disimpulkan bahwa *keyword-keyword* tersebut sering ditanyakan oleh calon mahasiswa baru ketika ingin mendaftar ke suatu kampus seperti info kampus, biaya, fasilitas, prodi dan lain-lain, *keyword* yang akan di *convert* merupakan pertanyaan umum yang sering ditanyakan, Berikut desain alur cara kerja *keyword Chatbot*.



Gambar 2. Desain Alur *Keyword* pada *Chatbot*

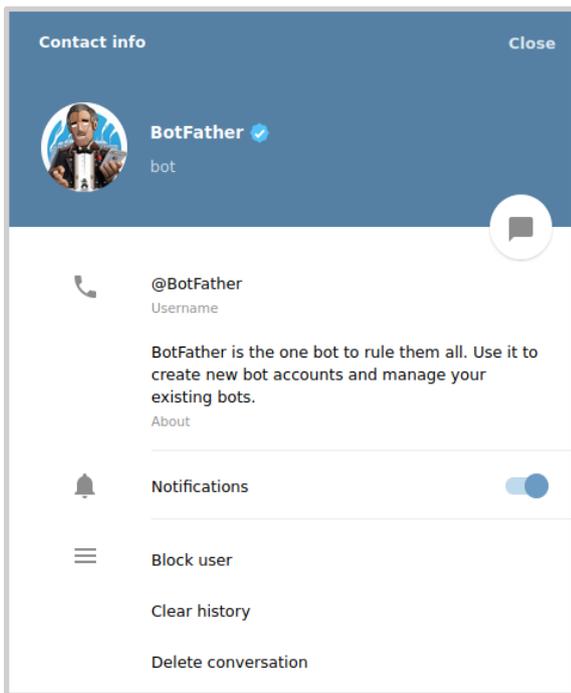
Chatbot yang dirancang merupakan *Chatbot* yang memberikan menu pilihan dengan menggunakan *keyword* tertentu, cara kerja *Chatbot*, *user* hanya perlu mengirimkan suatu pesan kemudian *bot* akan langsung memberikan keterangan atau penjelasan mengenai *keyword* yang telah

diinput *user*. Berikut penjelasan masing-masing *keyword*-nya:

Tabel 1. Penjelasan *Keyword* yang Tersedia dalam *Chatbot*

<i>User (Input)</i>	<i>Chatbot (Output)</i>
Start	<i>Chatbot</i> memulai percakapan dengan <i>user</i> dengan cara <i>bot</i> memperkenalkan diri terlebih dahulu.
Halo	Menyapa nama <i>user</i> dan menanyakan kabar <i>user</i> kemudian memberikan <i>keyword help</i> untuk membantu <i>user</i> dalam penggunaan <i>Chatbot</i> .
Help	Memberikan info apa saja <i>keyword</i> yang tersedia dalam <i>bot</i> untuk diinputkan oleh <i>user</i> ketika ingin mengetahui informasi tertentu.
Info PMB	Memberikan informasi yang berkaitan dengan Penerimaan Mahasiswa Baru.
Persyaratan	Menjelaskan alur atau prosedur untuk mendaftar kampus, apa saja yang harus dipenuhi.
Gelombang	Menampilkan pesan yang memberikan informasi jadwal gelombang PMB.
Info kampus	Memberikan informasi terkait kampus secara keseluruhan terkait akreditasi, visi misi dan lain lain.
Lokasi	Menampilkan pesan yang berisi Alamat kampus.
Biaya	Menampilkan pesan berupa rincian biaya administrasi kampus.
Fasilitas	Memberikan informasi fasilitas apa saja yang tersedia di kampus.
Prodi	Memberikan informasi prodi apa saja yang tersedia di kampus.
Beasiswa	Memberikan Informasi seputar beasiswa yang tersedia dan persyaratannya.

Sebelum membuat program *bot Telegram* dengan bahasa pemrograman *Python*, langkah yang harus dilakukan terlebih dahulu adalah membuat *Bot Telegram*, cara membuat *bot Telegram* itu sendiri menggunakan *Botfather* yang tersedia di *Telegram* dengan menggunakan *username @Botfather* pada kolom pencarian untuk mendapatkan *API* yang akan digunakan dalam pembuatan *Chatbot*.



Gambar 3. Botfather Telegram

Setelah berhasil membuat dan mendapatkan *API bot Telegram*, kemudian implementasi rancangan *Chatbot* yang ingin dibuat ke dalam pemrograman. Seperti gambar berikut:

```

1 #Import pyTelegramBotAPI
2 import telebot
3 import datetime
4
5 api = '1506887897:AAGj6mTkwsxw1RqKXvVJVvuw4-eD7nqokk4'
6 bot = telebot.TeleBot(api)
7
8 def log(message,perintah):
9     tanggal = datetime.datetime.now()
10    tanggal = tanggal.strftime('%d-%B-%Y')
11    nama_awal = message.chat.first_name
12    nama_akhir = message.chat.last_name
13    id_user = message.chat.id
14    text_log = '{} {}, {} {}, {} \n'.format(tanggal,id_user,
15    print(text_log)
16    log_bot = open('log_bot.txt','a')
17    log_bot.write(text_log)
18    log_bot.close()

```

Gambar 4. Program Awal Membuat *Chatbot* Telegram

Langkah pertama yang dilakukan adalah mengimport *library* yang diperlukan, kemudian *insert API* yang sudah didapatkan tadi ke dalam program, Lalu fungsi *library datetime* berhubungan dengan kode di *line 8*, Algoritma ini memiliki fungsi untuk melihat siapa saja nanti yang akan mengakses atau berkomunikasi dengan *Chatbot*, keterangan yang digunakan adalah keterangan tanggal, nama *user Telegram* dan perintah apa saja yang diinput. kemudian semua informasi ini nantinya akan disimpan di *file log_bot.txt*.

Langkah berikutnya adalah membuat program inti dari *Chatbot* yang akan dirancang, program memiliki algoritma yang berfungsi untuk memproses *keyword* yang di *input*

kemudian *output* yang dihasilkan berupa informasi sesuai *keyword* yang telah di *input*, Seperti gambar berikut:

```

@bot.message_handler(commands=['start'])
def action_start(message):
    log(message,'start')
    first_name = message.chat.first_name
    last_name = message.chat.last_name
    bot.reply_to(message,'Hi {} {}, Selamat datang di
    .format(first_name,last_name))

@bot.message_handler(commands=['halo'])
def action_hallo(message):
    log(message,'halo')
    first_name = message.chat.first_name
    last_name = message.chat.last_name
    bot.reply_to(message,'Halo {} {} gimana kabarnya

```

Gambar 5. Program Dasar Pembuatan *Command* pada *Chatbot*

Penjelasan pada gambar 5 program *message handler* memiliki fungsi untuk menyimpan *keyword* atau *command* yang akan digunakan oleh *user* untuk menjalankan *Chatbot*, Pada contoh pertama *keyword* yang digunakan adalah 'halo' kemudian penulis menambahkan fungsi untuk menyapa nama *user* sesuai dengan nama *Telegram user* dengan mengambil data *first & last name user* yang mengakses *bot*. Terakhir membuat kode *bot reply* yang merupakan eksekusi dari *keyword* yang telah diinput oleh *user*, disini penulis memberikan contoh kode program untuk *keyword start* dan *halo*.

Chatbot yang dirancang untuk bisa beroperasi 24 jam perlu dilakukan *web hosting*, pada perancangan ini penulis menggunakan layanan *hosting python anywhere*, Tahap testing merupakan tahap penyelesaian implementasi membuat *Chatbot*, hal yang akan dilakukan adalah membuat *command* untuk mengeksekusi program, berikut adalah *command*-nya:

```

print('Bot Start Running..')
bot.polling()

```

Gambar 6. Program Eksekusi *Chatbot*

Langkah terakhir adalah menjalankan perintah untuk *hosting*, kembali lagi ke halaman *python anywhere* kemudian pada bagian *console* ada fungsi *new console* lalu klik button *\$Bash*, setelah diklik akan diarahkan ke halaman *bash console* seperti pada gambar berikut:

```

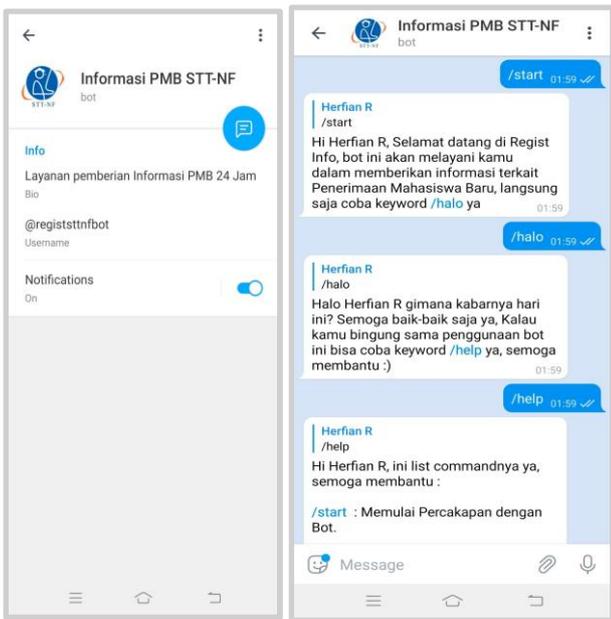
Bash console 19194510
06:17 ~ $ cd registbot/
06:17 ~/registbot $ ls
log_bot.txt regist.py
06:17 ~/registbot $ python3 regist.py
Bot Start Running..

```

Gambar 7. Eksekusi Program di *Console Pythonanywhere*

Setelah berhasil melakukan *web hosting* di *python anywhere* untuk program *Chatbot* agar bisa beroperasi 24 jam, langkah

berikutnya adalah *testing Chatbot* di *Telegram*, untuk langkahnya sendiri yang pertama adalah mencari *username Chatbot* yang sudah dibuat di dalam layanan *Botfather*, untuk mencari *username*nya hanya menginput *username bot Telegram* di fitur pencarian, setelah sudah menemukan langkah selanjutnya adalah demo percakapan dengan *Chatbot*, jika menginput *keyword* yang sudah dijadikan *command* skemudian *bot* memberikan *output* sesuai *input* maka program *Chatbot* dan *hosting* berhasil, pada gambar 8. penulis memberikan contoh demo percakapan dengan *Chatbot* menggunakan *command /start*, */halo* dan */help*, kemudian *Chatbot* memberikan *output* yang sesuai.



Gambar 8. Menjalankan Chatbot Telegram

Pengujian menggunakan metode *Black Box Testing* untuk menguji aplikasi *Chatbot*, metode *User Acceptance User* dan Kuesioner untuk pengujian kepada *user*.

Pengujian pertama, penulis melakukan pengujian *Black Box Testing* dengan membuat tabel yang berisi deskripsi pengujian, hasil yang diharapkan dan juga hasil pengujian yang menunjukkan bahwa fitur yang dibuat sudah berjalan. Berikut ini adalah tabel hasil pengujiannya:

Tabel 2. Pengujian Black Box Testing

No	Pengujian	Hasil yang diharapkan	Status
1	Mencari <i>bot</i> melalui fitur pencarian <i>Telegram</i> dengan menginput <i>username bot</i> .	<i>Username</i> yang digunakan mudah untuk menemukan <i>bot</i> karena <i>output</i> dari <i>input username</i> hanya menampilkan 1 hasil pencarian.	Berhasil
		<i>Username</i> dan <i>input search</i> terdapat <i>typo</i> sehingga <i>bot</i> tidak dapat ditemukan.	Berhasil

No	Pengujian	Hasil yang diharapkan	Status
2	Menjalank-an perintah <i>/start</i> .	<i>Chatbot</i> merespon <i>keyword /start</i> dengan mulai memperkenalkan fungsi utama <i>Chatbot</i> kemudian mentrigger <i>keyword /halo</i> .	Berhasil
3	Menjalank-kan perintah <i>/halo</i>	<i>Chatbot</i> memulai percakapan dengan menanyakan kabar dan menampilkan pesan berisi <i>keyword /help</i> guna untuk mempermudah mengetahui fungsi utama <i>bot</i> .	Berhasil
4	Dapat menjalank-an perintah <i>/help</i>	Hasil eksekusi perintah <i>/help</i> akan menampilkan seluruh command yang tersedia pada <i>Chatbot</i> untuk mempermudah <i>user</i> menemukan informasi yang ingin diketahui dahulu dengan menginput <i>keyword</i> sesuai.	Berhasil
5	Dapat menginput semua <i>keyword</i> untuk menjalanka-n fungsi <i>Chatbot</i> .	<i>Chatbot</i> membalas pesan berupa informasi sesuai dengan <i>command</i> yang diinput.	Berhasil
		<i>Chatbot</i> tidak merespon inputan <i>keyword</i> yang mengandung kesalahan seperti tidak memakai tanda “/”, mengandung huruf besar seperti <i>/Start</i> , <i>/hElp</i> dan lain-lain	Berhasil
		<i>Chatbot</i> tidak merespon pesan sama sekali jika inputan <i>keyword</i> tidak tersedia pada <i>Chatbot</i> .	Berhasil
6	Menjalank-an semua <i>keyword</i> pada <i>Chatbot</i>	Berhasil menjalankan semua <i>keyword</i> tanpa adanya error atau suatu <i>keyword</i> yang tidak bekerja.	Berhasil

Hasil yang didapat dari pengujian *black box testing* yang dibuat penulis semua pengujian menunjukkan *status* Berhasil dan dapat disimpulkan bahwa aplikasi *Chatbot* sudah layak untuk diuji kepada *user* dengan menggunakan pengujian *User Acceptance Testing* dan Kuesioner.

Pengujian kedua menggunakan UAT, Pada bagian ini penulis membuat tabel yang berisi kasus pengujian dan hasil pengujian yang menunjukkan bahwa fitur yang dibuat berhasil dapat berjalan atau tidak berhasil. Data pengujian yang diisi oleh 22 responden. Berikut ini adalah tabel hasil pengujiannya:

Tabel 3. Pengujian *User Acceptance Testing*

No	Pengujian	Hasil
1	User dapat mencari <i>bot</i> melalui fitur pencarian di <i>Telegram</i>	- Berhasil : 22 - Tidak Berhasil : 0
2	User dapat menemukan <i>bot</i> dengan menginput <i>username bot '@registsttnfbot'</i> atau ' <i>registsttnfbot</i> ' di fitur pencarian <i>Telegram</i>	- Berhasil : 22 - Tidak Berhasil : 0
3	User dapat memulai menjalankan <i>bot</i> dengan menginput command <i>'/start'</i>	- Berhasil : 21 - Tidak Berhasil : 1
4	User dapat menginput command <i>'/halo'</i> untuk mulai berkomunikasi dengan <i>bot</i>	- Berhasil : 22 - Tidak Berhasil : 0
5	User dapat menginput command <i>'/help'</i> untuk menampilkan pesan yang berisi semua <i>keyword</i> yang tersedia pada <i>bot</i>	- Berhasil : 22 - Tidak Berhasil : 0
6	User dapat menjalankan semua <i>input command</i> yang tersedia pada <i>bot</i>	- Berhasil : 22 - Tidak Berhasil : 0
7	Respon atau balasan chat dari <i>bot</i> sesuai dengan <i>command</i> yang diinput <i>user</i>	- Berhasil : 22 - Tidak Berhasil : 0
8	<i>Bot</i> aktif 24/7 dan respon <i>bot</i> kepada <i>user</i> sangat cepat	- Berhasil : 20 - Tidak Berhasil : 2

Dari 8 pengujian UAT yang telah diuji oleh 22 responden, hasil yang didapat memperoleh total 173 status berhasil dan 3 status tidak berhasil.

Kemudian rancangan pengujian menggunakan kuesioner untuk menilai *Chatbot* yang telah dibuat. Kuesioner ini dinilai oleh responden yang sebelumnya telah melakukan pengujian *user acceptance testing* dan metode penilaiannya menggunakan skala likert dengan 4 pilihan yaitu Sangat Tidak Setuju (STS) dengan skor 1, Tidak Setuju (TS) dengan skor 2, Setuju (S) dengan skor 3, dan Sangat Setuju (SS) dengan skor 4, berikut kuesioner yang akan diberikan melalui bentuk tabel:

Tabel 4. Data Pengujian Kuesioner

No	Pernyataan	STS	TS	S	SS	Persentase
1	<i>Chatbot</i> mudah ditemukan melalui fitur <i>search</i>	0	0	8	14	90,90%

No	Pernyataan	STS	TS	S	SS	Persentase
2	Command <i>'/start'</i> <i>Chatbot</i> berjalan dengan baik	0	1	4	17	92,04%
3	Fitur <i>input keyword</i> mempermudah untuk pengaplikasian komunikasi dengan <i>bot</i>	0	0	7	15	92,04%
4	Semua inputan <i>command</i> yang tersedia di <i>Chatbot</i> berjalan dengan baik	0	0	9	13	89,77%
5	<i>Chatbot</i> memberikan respon sesuai dengan <i>command</i> yang diinput dan berjalan dengan baik	0	0	9	13	89,77%
6	Informasi yang ditampilkan <i>Chatbot</i> sangat jelas dan spesifik	0	0	12	10	86,36%
7	<i>Chatbot</i> membantu dalam mencari informasi yang berkaitan dengan PMB	0	0	8	14	90,90%
8	Dengan adanya penerapan <i>Chatbot</i> , mendapatkan informasi terkait PMB menjadi lebih efektif	0	1	10	11	85,22%

Berdasarkan hasil perhitungan yang didapatkan, penulis mendapatkan kesimpulan dengan menggunakan skala likert. Pada pengujian kuesioner nomor 8 tentang efektivitas penerapan *Chatbot* dapat disimpulkan bahwa penelitian ini berhasil menjawab rumusan masalah penulis, dengan skor 85,22% dengan interpretasi Sangat Baik, Kemudian total persentase keseluruhan kuesioner mendapatkan skor 71,93% dengan interpretasi Baik maka dapat disimpulkan penerapan *Chatbot* sangat layak dan sesuai dengan kebutuhan *user* dalam mencari informasi terkait PMB.

5. KESIMPULAN

Dari hasil penelitian penulis, dapat disimpulkan bahwa penelitian ini berhasil menjawab permasalahan pada penelitian yang saat ini sedang dilakukan. Penelitian dimulai dari analisis sistem, observasi studi literasi, perancangan sistem, dilanjutkan dengan hosting aplikasi dan diakhiri dengan pengujian aplikasi. Kemudian penelitian yang dilakukan juga memiliki kesimpulan yang menjawab perumusan masalah penulis, yaitu:

1. Pembuatan aplikasi *Chatbot* menggunakan bahasa pemrograman *python* kemudian diimplementasikan di platform media sosial *Telegram*. *Chatbot* yang dibuat menerapkan kode program berdasarkan hasil studi literatur yang dipelajari penulis. Lalu menjalankan program *Chatbot* di *web service python anywhere* untuk pengujian terhadap *user*, sehingga *user* dapat mengakses *bot* selama 24 jam.
 2. Efektivitas *Chatbot* diukur melalui pengujian Kuesioner nomor 8 dan pengujian tersebut menghasilkan skor 85.22% dengan interpretasi Sangat Baik dan berhasil menjawab rumusan masalah penelitian penulis tentang seberapa efektif penerapan *Chatbot* dalam mendapatkan informasi seputar PMB.
 3. Pengujian aplikasi *Chatbot* menggunakan *black box testing* berjalan dengan baik, tingkat keberhasilan 100% dan hasil semua pengujian sesuai dengan hasil yang diharapkan.
 4. Pengujian *User Acceptance Testing* menghasilkan tingkat keberhasilan 173 dari 176 total pengujian yang dilakukan oleh 22 responden, Sehingga dapat disimpulkan penerapan aplikasi *Chatbot* dalam mendapatkan informasi terkait PMB sesuai dengan kebutuhan *user*.
 5. Pengujian dengan menggunakan kuesioner secara keseluruhan menghasilkan skor 71,93% dengan intepretasi Baik.
- chat-for-customer-service-stats/ (accessed Jul. 26, 2020).
- [3] “*Chatbot vs. Live Chat: Which is Winning in Customer Service?*,” Acquire, Sep. 04, 2017. [Online]. Available: <https://acquire.io/blog/Chatbot-vs-live-chat/> (accessed Jul. 26, 2020)
 - [4] M. B. Chaniago and A. Junaidi, “*Student Presence Using RFID and Telegram Messenger Application*,” p. 5.
 - [5] Accenture, “*Chatbots In Customer Service*,” 2016.
 - [6] D. Korotaeva, M. Khlopotov, A. Makarenko, E. Chikshova, N. Startseva, and A. Chemysheva, “*Botanicum: a Telegram Bot for Tree Classification*,” in 2018 22nd Conference of Open Innovations Association (FRUCT), Jyvaskyla, May 2018, pp. 88–93, doi: 10.23919/FRUCT.2018.8468278
 - [7] R. Nufusula and A. Susanto, “*Rancang Bangun ChatBot pada Server Pulsa menggunakan Telegram Bot API*,” p. 10, 2018.
 - [8] A. Zubaidi and R. Ramdani, “*Layanan Dan Informasi Akademik Berbasis Bot Telegram di Program Studi Teknik Informatika Universitas Mataram*,” *J. Teknol. Inf. Komput. Dan Apl. JTIKA*, vol. 1, no. 1, pp. 103–110, May 2019, doi: 10.29303/jtika.v1i1.27

DAFTAR PUSTAKA

- [1] J. T. Campbell, “*Web Design: Introductory*,” 6 ed. Boston: Chengage Learning, 2017.
- [2] G. Charlton, “*Consumers prefer live chat for customer service: stats*,” Econsultancy, Nov. 25, 2013, [Online]. Available: <https://econsultancy.com/consumers-prefer-live->