

## **BAB IV**

### **IMPLEMENTASI**

Pada bab ini menggambarkan mengenai implementasi perancangan yang sebelumnya telah dibahas pada Bab III. Bahasan dalam bab ini meliputi implementasi dan pengujian aplikasi. Implementasi dan pengujian sistem ini dilakukan untuk mengetahui apakah aplikasi yang telah dibangun sudah bekerja sesuai dengan rancangan aplikasi.

Pada struktural laporan di bab ini terdapat beberapa implementasi dan pengujian yang penulis lakukan yaitu implementasi *chatbot*, implementasi API, implementasi aplikasi dan pengujian *chatbot*.

#### **4.1 Implementasi Chatbot**

Pada tahap implementasi *chatbot*, penulis akan menjelaskan mengenai penerapan *chatbot* menggunakan *framework* rasa.ai. Ada beberapa tahapan penerapan *chatbot* yang penulis lakukan seperti membuat domain, nlu data, action dan *stories* atau alur percakapan *chatbot*.

##### **4.1.1 Domain**

Untuk membuat *chatbot* menggunakan *framework* rasa.ai, hal pertama yang harus dilakukan adalah membuat *domain*. *Domain* digunakan untuk mendefinisikan semesta percakapan yaitu *intent*, *entities*, *slot*, *action*. *Domain* disimpan dalam format (domain.yml).

###### **a. Intents**

Dalam implementasi *chatbot*, penulis telah membuat beberapa *intents* atau kategori maksud dalam percakapan. Implementasi untuk mendefinisikan *intents* dalam *framework* rasa.ai dapat dilihat pada Gambar 4.1.

```

intents:
- greet
- pesan
- surat
- maksud
- smst_jln
- alamat
- nama_ortu
- pekerjaan
- nip
- instansi
- pangkat
- lokasi
- kota
- proses
- history
- out_of_scope

```

Gambar 4.1 Mendefinisikan *Intents*

Gambar 4.1 menunjukkan mendefinisikan *intents* untuk membuat percakapan. Setiap *intents* yang telah didefinisikan mempunyai fungsi berbeda. Berikut ini merupakan tabel yang penulis buat untuk menjelaskan fungsi dari setiap *intents*, dapat dilihat pada Tabel 4.1

Tabel 4.1 Penjelasan Fungsi Setiap *Intent*

No	<i>Intents</i>	Fungsi
1	<i>Greet</i>	mengkategorikan percakapan <i>greet</i>
2	Pesan	mengkategorikan percakapan pemesanan surat
3	Surat	mengkategorikan percakapan jenis surat
4	Maksud	mengkategorikan percakapan maksud dalam pemesanan surat
5	smst_jln	mengkategorikan percakapan semester aktif mahasiswa
6	Alamat	mengkategorikan percakapan alamat mahasiswa
7	nama_ortu	mengkategorikan percakapan nama orangtua mahasiswa
8	Pekerjaan	mengkategorikan percakapan pekerjaan orangtua mahasiswa
9	Nip	mengkategorikan percakapan nip orangtua mahasiswa
10	Instansi	mengkategorikan percakapan instansi pekerjaan orangtua mahasiswa
11	Pangkat	mengkategorikan percakapan pangkat pekerjaan orangtua mahasiswa
12	Lokasi	mengkategorikan percakapan lokasi kantor imigrasi
13	Kota	mengkategorikan percakapan kota kantor imigrasi
14	Proses	mengkategorikan percakapan proses pemesanan surat
15	<i>history</i>	mengkategorikan percakapan history pemesanan surat
16	<i>out_of_scope</i>	mengkategorikan percakapan yang tidak terkategorikan

b. *Entities*

Penulis telah membuat beberapa *entities* atau kategori entitas dalam sebuah *intent* percakapan. Implementasi untuk mendefinisikan *entities* dalam *framework* rasa.ai dapat dilihat pada Gambar 4.2.

```
entities:  
- pesan  
- surat  
- maksud  
- smst_jln  
- alamat  
- nama_ortu  
- pekerjaan  
- nip  
- instansi  
- pangkat  
- lokasi  
- kota  
- proses  
- history
```

Gambar 4.2 Mendefinisikan *Entities*

Gambar 4.2 menunjukkan mendefinisikan *entities* untuk membuat percakapan. Setiap *entities* yang telah didefinisikan mempunyai fungsi berbeda. Berikut ini merupakan tabel yang penulis buat untuk menjelaskan fungsi dari setiap *entities*, dapat dilihat pada Tabel 4.2

Tabel 4.2 Penjelasan Fungsi Setiap *Entities*

No	<i>Entities</i>	Fungsi
1	Pesan	mendefinisikan entitas pesan dalam intent pesan
2	Surat	mendefinisikan entitas surat dalam intent surat
3	Maksud	mendefinisikan entitas maksud dalam intent maksud
4	smst_jln	mendefinisikan entitas smst_jln dalam intent smst_jln
5	Alamat	mendefinisikan entitas alamat dalam intent alamat
6	nama_ortu	mendefinisikan entitas nama_ortu dalam intent nama_ortu
7	Pekerjaan	mendefinisikan entitas pekerjaan dalam intent pekerjaan
8	Nip	mendefinisikan entitas nip dalam intent nip
9	Instansi	mendefinisikan entitas instansi dalam intent instansi
10	Pangkat	mendefinisikan entitas pangkat dalam intent pangkat
11	Lokasi	mendefinisikan entitas lokasi dalam intent lokasi
12	Kota	mendefinisikan entitas kota dalam intent kota
13	Proses	mendefinisikan entitas proses dalam intent proses
14	<i>History</i>	mendefinisikan entitas history dalam intent history

c. *Slots*

Penulis juga telah membuat beberapa *slots* untuk menyimpan entitas dalam sebuah percakapan. Implementasi untuk mendefinisikan *slots* dalam *framework* rasa.ai dapat dilihat pada Gambar 4.3.

```
slots:  
  alamat:  
    type: text  
  fakultas:  
    type: text  
  history:  
    type: text  
  id_surat:  
    type: text  
  instansi:  
    type: text  
  jenis_surat:  
    type: text  
  jurusan:  
    type: text  
  kota:  
    type: text  
  maksud:  
    type: text  
  nama:  
    type: text  
  nama_ortu:  
    type: text  
  lokasi:  
    type: text  
  nim:  
    type: text  
  nip:  
    type: text  
  pangkat:  
    type: text  
  pekerjaan:  
    type: text  
  pesan:  
    type: text  
  proses:  
    type: text  
  smst_jln:  
    type: text  
  surat:  
    type: text  
  tanggal_lahir:  
    type: text  
  tempat_lahir:  
    type: text  
  time:  
    type: text
```

Gambar 4.3 Mendefinisikan *Slots*

Gambar 4.3 menunjukkan mendefinisikan *slots* untuk membuat percakapan. Setiap *slots* yang telah didefinisikan mempunyai fungsi berbeda. Berikut ini merupakan tabel yang penulis buat untuk menjelaskan fungsi dari setiap *slots*, dapat dilihat pada Tabel 4.3

Tabel 4.3 Penjelasan Fungsi Setiap Slots

No	Slots	Type	Fungsi
1	Alamat	<i>text</i>	menyimpan entitas alamat
2	Fakultas	<i>text</i>	menyimpan data fakultas mahasiswa
3	<i>History</i>	<i>text</i>	menyimpan entitas history
4	id_surat	<i>text</i>	menyimpan data id_surat
5	Instansi	<i>text</i>	menyimpan entitas instansi
6	jenis_surat	<i>text</i>	menyimpan data jenis_surat
7	Jurusan	<i>text</i>	menyimpan data jurusan mahasiswa
8	Kota	<i>text</i>	menyimpan entitas kota
9	Lokasi	<i>text</i>	menyimpan entitas lokasi
10	Maksud	<i>text</i>	menyimpan entitas maksud
11	Nama	<i>text</i>	menyimpan data nama mahasiswa
12	nama_ortu	<i>text</i>	menyimpan entitas nama_ortu
13	Nim	<i>text</i>	menyimpan data nim mahasiswa
14	Nip	<i>text</i>	menyimpan entitas nip
15	Pangkat	<i>text</i>	menyimpan entitas pangkat
16	Pekerjaan	<i>text</i>	menyimpan entitas pekerjaan
17	Pesan	<i>text</i>	menyimpan entitas pesan
18	Proses	<i>text</i>	menyimpan entitas proses
19	smst_jalan	<i>text</i>	menyimpan entitas smst_jln
20	Surat	<i>text</i>	menyimpan entitas surat
21	tanggal_lahir	<i>text</i>	menyimpan data tanggal lahir mahasiswa
22	tempat_lahir	<i>text</i>	menyimpan data tempat lahir mahasiswa
23	<i>Time</i>	<i>text</i>	menyimpan data <i>time</i>

d. *Action*

Penulis juga telah membuat beberapa *action* untuk menentukan aksi dalam sebuah percakapan. Implementasi untuk mendefinisikan *action* dalam *framework* rasa.ai dapat dilihat pada Gambar 4.4.

```
actions:  
- utter_greet  
- utter_ask_pesan  
- utter_ask_data  
- utter_ask_maksud  
- utter_ask_semester  
- utter_ask_orangtua  
- utter_ask_alamat  
- utter_ask_pekerjaan  
- utter_ask_nip  
- utter_ask_instansi  
- utter_ask_pangkat  
- utter_ask_kota  
- utter_ask_lokasi  
- utter_thanks  
- utter_proses  
- utter_konfirmasi_surat01  
- utter_konfirmasi_surat02  
- utter_konfirmasi_surat03  
- utter_konfirmasi_surat04  
- utter_konfirmasi_surat05  
- action_jenis_surat  
- action_konfirmasi_surat  
- action_get_time  
- action_get_user  
- action_get_history  
- action_post  
- action_fallback
```

Gambar 4.4 Mendefinisikan *Actions*

Gambar 4.4 penulis telah mendefinisikan *actions* untuk menentukan aksi percakapan. Setiap *actions* yang telah didefinisikan mempunyai fungsi berbeda. Berikut ini merupakan tabel yang penulis buat untuk menjelaskan fungsi dari setiap *actions*, dapat dilihat pada Tabel 4.4

Tabel 4.4 Penjelasan Fungsi Setiap *Actions*

No	<i>Actions</i>	Fungsi
1	utter_greet	mendefinisikan template percakapan greet
2	utter_ask_pesan	mendefinisikan template percakapan pesan
3	utter_ask_data	mendefinisikan template percakapan data
4	utter_ask_maksud	mendefinisikan template percakapan maksud
5	utter_ask_semester	mendefinisikan template percakapan semester
6	utter_ask_orangtua	mendefinisikan template percakapan orangtua
7	utter_ask_alamat	mendefinisikan template percakapan alamat
8	utter_ask_pekerjaan	mendefinisikan template percakapan pekerjaan
9	utter_ask_nip	mendefinisikan template percakapan nip
10	utter_ask_instansi	mendefinisikan template percakapan instansi
11	utter_ask_pangkat	mendefinisikan template percakapan pangkat
12	utter_ask_kota	mendefinisikan template percakapan kota
13	utter_ask_lokasi	mendefinisikan template percakapan lokasi
14	utter_thanks	mendefinisikan template percakapan thanks
15	utter_proses	mendefinisikan template percakapan proses
16	utter_konfirmasi_surat01	mendefinisikan template percakapan konfirmasi surat01
17	utter_konfirmasi_surat02	mendefinisikan template percakapan konfirmasi surat02
18	utter_konfirmasi_surat03	mendefinisikan template percakapan konfirmasi surat03
19	utter_konfirmasi_surat04	mendefinisikan template percakapan konfirmasi surat04
20	utter_konfirmasi_surat05	mendefinisikan template percakapan konfirmasi surat05
21	action_jenis_surat	mendefinisikan custom actions jenis surat
22	action_konfirmasi_surat	mendefinisikan custom actions konfirmasi surat
23	action_get_time	mendefinisikan custom actions get time
24	action_get_user	mendefinisikan custom actions get user
25	action_get_history	mendefinisikan custom actions get history
26	action_post	mendefinisikan custom actions post
27	action_fallback	mendefinisikan default actions fallback

e. *Template*

Penulis juga telah membuat beberapa *template* untuk menuliskan *output* sebuah percakapan. Contoh implementasi untuk mendefinisikan *template* dalam *framework* rasa.ai dapat dilihat pada Gambar 4.5.

```

templates:
  utter_greet:
  - text: "Hai {nama}, Selamat {time}, Ada yang bisa dibantu? \n1. Pesan
    Surat \n2. Sejarah Pesan \n3. Bantuan"
    buttons:
    - title: '1'
      payload: Pesan
    - title: '2'
      payload: Sejarah
    - title: '3'
      payload: Bantuan
  utter_ask_pesan:
  - text: "Surat apa yang ingin anda pesan? \n1. Surat Keterangan Mahasiswa
    Aktif \n2. Surat Keterangan Akreditasi \n3. Surat Keterangan Berkelakuan
    Baik\n4. Surat Keterangan Beasiswa \n5. Surat Pengantar Pembuatan Paspor"
    buttons:
    - title: '1'
      payload: Surat Keterangan Mahasiswa Aktif
    - title: '2'
      payload: Surat Keterangan Akreditasi
    - title: '3'
      payload: Surat Keterangan Berkelakuan Baik
    - title: '4'
      payload: Surat Keterangan Beasiswa
    - title: '5'
      payload: Surat Pengantar Pembuatan Paspor
  utter_ask_data:
  - text: Silahkan jawab pertanyaan berikut ini untuk data pelengkap surat!
  utter_ask_maksud:
  - text: Apa maksud/keperluan surat?

```

Gambar 4.5 Mendefinisikan *Template*

Gambar 4.5 menunjukkan contoh implementasi untuk mendefinisikan *template* percakapan dalam *framework* rasa.ai. Kode implementasi *template* secara lengkap dirujuk pada lampiran laporan bagian “lampiran kode *template* percakapan dalam implementasi *chatbot*”.

#### 4.1.2 NLU Data

Setelah mendefinisikan semesta percakapan, langkah selanjutnya adalah membuat data training untuk automasi percakapan. *Data training* atau NLU (*Natural Language Understanding*) Data akan diterjemahkan oleh Rasa NLU. Rasa NLU merupakan alat pengolah bahasa alami untuk klasifikasi maksud, pengambilan *response* dan ekstraksi entitas di *chatbots* berdasarkan data training yang telah dibuat. *Data training* pada *framework* rasa.ai disimpan dengan menggunakan format *file* Markdown (*nlu\_data.md*). Contoh implementasi NLU Data dalam *framework* rasa.ai dapat dilihat pada Gambar 4.6.

```

## intent:greet
- hallo
- hai
- hey
- hay
- hy
- hi
- hay
- haii
- hallooo
- hii
- hyyy

## intent:surat
- [Surat Keterangan Mahasiswa Aktif](surat)
- [Surat Keterangan Akreditasi](surat)
- [Surat Keterangan Berkelakuan Baik](surat)
- [Surat Keterangan Beasiswa](surat)
- [Surat Pengantar Pembuatan Paspor](surat)

## intent:kota
- [Solo](kota)
- [Sleman](kota)
- [Klaten](kota)
- [Bogor](kota)
- [Bandung](kota)
- [Surabaya](kota)
- [Semarang](kota)
- [Balikpapan](kota)
- [Medan](kota)

```

Gambar 4.6 Contoh Implementasi NLU Data

Gambar 4.6 menunjukkan contoh implementasi data NLU dalam *framework* rasa.ai. Penulis juga telah membuat data NLU sesuai dengan *intent* yang telah didefinisikan sebelumnya dirujuk pada lampiran laporan bagian “lampiran kode NLU Data dalam implementasi *chatbot*”.

### 4.1.3 Actions

Setelah mendefinisikan semesta percakapan dan membuat *data training*, berikutnya adalah menuliskan *custom action*. *Custom action* merupakan suatu fungsi yang dapat dijalankan untuk menentukan aksi dalam sebuah percakapan. *Custom action* dalam *framework* rasa.ai dapat dituliskan dengan menggunakan bahasa pemrograman python dan disimpan dalam format python (*actions.py*). Penulis menggunakan *custom action* untuk dapat berkomunikasi dengan aplikasi eksternal melalui REST API yang juga akan dibuat nantinya. Contoh implementasi *custom action* dalam *framework* rasa.ai yang berfungsi untuk mengirimkan request pada REST API dapat dilihat pada Gambar 4.7.

```

class ActionPost(Action):
    def name(self):
        return "action_post"

    def run(self, dispatcher, tracker, domain):
        id_jenis_surat = tracker.get_slot('id_surat')
        nim = tracker.get_slot('nim')
        maksud = tracker.get_slot('maksud')
        alamat = tracker.get_slot('alamat')
        smst_jln = tracker.get_slot('smst_jln')
        nama_ortu = tracker.get_slot('nama_ortu')
        nip = tracker.get_slot('nip')
        pangkat = tracker.get_slot('pangkat')
        pekerjaan = tracker.get_slot('pekerjaan')
        instansi = tracker.get_slot('instansi')

        body = {'id_jenis_surat':id_jenis_surat,
                'nim':nim,
                'maksud':maksud,
                'alamat':alamat,
                'smst_jln':smst_jln,
                'nama_ortu':nama_ortu,
                'nip':nip,
                'pangkat':pangkat,
                'pekerjaan':pekerjaan,
                'instansi':instansi}

        url = 'http://api.masholeh.web.id/apply'
        r = requests.post(url, data=body)

        return []

```

Gambar 4.7 Contoh Implementasi *Actions*

Gambar 4.7 merupakan salah satu contoh implementasi *custom action* dalam *framework* rasa.ai yang berfungsi untuk mengirimkan *request* pada REST API. Selain itu, penulis juga telah membuat beberapa *custom action* lainnya dalam implementasi *chatbot* dirujuk pada lampiran laporan bagian “lampiran kode *custom action* dalam implementasi *chatbot*”.

#### 4.1.4 *Stories*

Langkah terakhir dalam implementasi *chatbot* yaitu membuat *stories*. *Stories* merupakan alur percakapan dalam *chatbot*. Pada tahap ini, penulis telah membuat beberapa alur percakapan yaitu *stories* untuk melakukan pemesanan surat berdasarkan jenis surat dan *stories* untuk melihat daftar riwayat pemesanan surat. Contoh penulisan *stories* untuk melakukan pemesanan surat keterangan akreditasi dapat dilihat pada Gambar 4.8.

```

## Surat Keterangan Akreditasi
* greet
  - action_get_time
  - slot{"time": "siang"}
  - action_get_user
  - slot{"nim": "15523208"}
  - slot{"nama": "Almafazi"}
  - slot{"tempat_lahir": "KEBUMEN"}
  - slot{"tanggal_lahir": "1998-03-23"}
  - slot{"jurusan": "Teknik Informatika"}
  - slot{"fakultas": "Fakultas Teknologi Industri"}
  - utter_greet
* pesan{"pesan": "pesan"}
  - slot{"pesan": "pesan"}
  - utter_ask_pesan
* surat{"surat": "Surat Keterangan Akreditasi"}
  - slot{"surat": "Surat Keterangan Akreditasi"}
  - action_jenis_surat
  - slot{"id_surat": "2"}
* proses
  - action_post
  - utter_thanks

```

Gambar 4.8 Contoh Implementasi *Stories*

Gambar 4.8 merupakan salah satu contoh penulisan *stories* pada *rasa.ai* yaitu *stories* untuk melakukan pemesanan surat keterangan akreditasi pada *chatbot*. Kode untuk menentukan alur percakapan dalam implementasi *chatbot* dirujuk pada lampiran laporan bagian “lampiran kode untuk menentukan alur percakapan dalam implementasi *chatbot*”.

## 4.2 Implementasi API

Salah satu tujuan dari penelitian ini di antaranya adalah membuat sebuah aplikasi *chatbot* yang dapat terintegrasi dengan sistem informasi surat *online* (SISO) FTI. Untuk dapat mengintegrasikan aplikasi *chatbot* dan SISO FTI maka diperlukan sebuah web service berupa API, maka dari itu penulis membuat sebuah web service untuk dapat mengintegrasikan aplikasi *chatbot* dengan SISO FTI. Penulis mengimplementasikan web service tersebut dengan menggunakan *framework* *lumen*. Dalam web service tersebut penulis membuat beberapa *function* controller yaitu *function* *login*, *function* *get\_time*, *function* *get\_user*, *function* *apply*, *function* *get\_history* dan *function* *logout*. Berikut ini merupakan tabel yang penulis buat untuk menjelaskan kegunaan dari setiap *function*, dapat dilihat pada Tabel 4.5

Tabel 4.5 Kegunaan Dari Setiap *Function* Pada REST API

No	nama <i>function</i>	kegunaan
1	<i>function</i> login	berfungsi untuk Authentication <i>User</i> pada saat awal menggunakan aplikasi
2	<i>function</i> get_time	berfungsi untuk mengambil waktu real pada saat awal menggunakan aplikasi
3	<i>function</i> get_user	berfungsi untuk mengambil data <i>user</i> saat menggunakan aplikasi berdasarkan authentication
4	<i>function</i> get_history	berfungsi untuk mengambil data riwayat pemesanan surat berdasarkan <i>user</i>
5	<i>function</i> apply	berfungsi untuk menyimpan permintaan pemesanan surat yang dilakukan oleh <i>user</i>
6	<i>function</i> logout	berfungsi untuk keluar dari penggunaan aplikasi

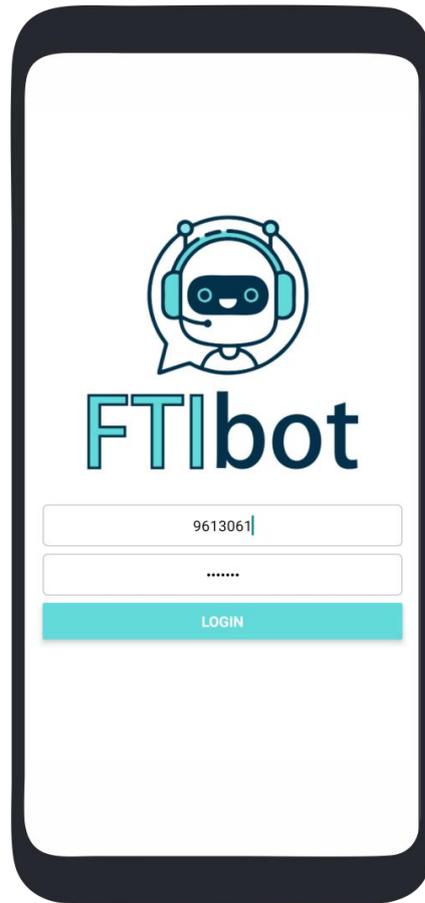
Penulis telah membuat sebuah *web service* yang dapat diakses melalui url <http://api.masholeh.web>. Kode yang berupa *function* controller untuk digunakan untuk mengintegrasikan aplikasi *chatbot* dengan sistem informasi surat *online* (SISO) FTI dapat dirujuk pada lampiran laporan bagian “lampiran kode *function* yang terdapat pada REST API”.

### 4.3 Implementasi Aplikasi

Untuk dapat memudahkan mahasiswa dalam mengakses *chatbot* untuk melakukan pemesanan surat, maka dari itu diperlukan suatu *platform* chat. Pada tahap ini, penulis mengimplementasikan suatu aplikasi berbasis android yang dapat digunakan sebagai *platform chat* agar dapat digunakan mahasiswa dalam mengakses *chatbot* untuk melakukan pemesanan surat di FTI. Penulis mengimplementasikan aplikasi *chatbot* berbasis android menggunakan *framework* react-native dengan bahasa pemrograman Java. Aplikasi *chatbot* ini juga akan terintegrasi dengan *web service* dan *chatbot* rasa.ai yang telah di implementasikan penulis sebelumnya. Pada tahap ini, penulis membuat dua halaman yaitu halaman login dan halaman *chat*.

#### 4.3.1 Halaman Login

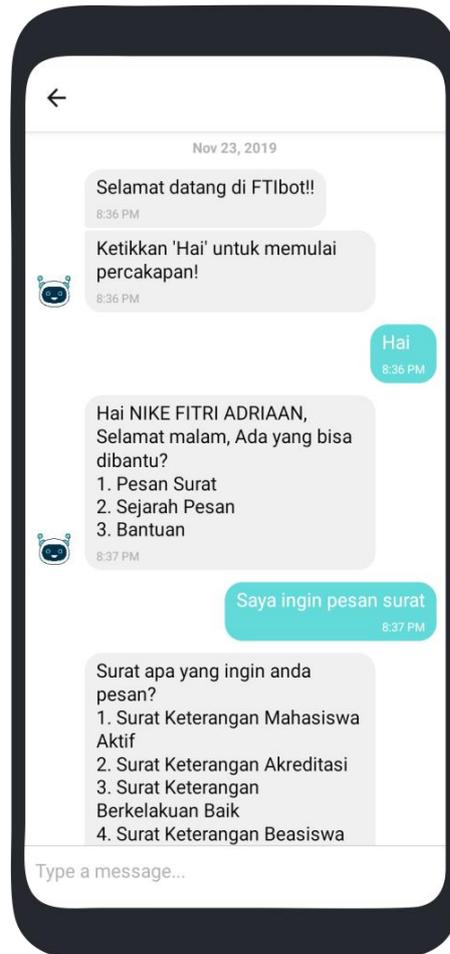
Halaman ini adalah halaman yang pertama kali akan muncul pada saat aplikasi *chatbot* ini diakses. Untuk dapat menggunakan fungsionalitas yang terdapat di dalam aplikasi ini, pengguna harus melakukan login dengan memasukkan nim dan *password* pada *form* yang telah tersedia di tampilan antarmuka aplikasi. Tampilan halaman login dapat dilihat pada Gambar 4.9.



Gambar 4.9 Halaman Login Aplikasi

#### 4.3.2 Halaman *Chat*

Halaman *chat* adalah halaman utama untuk dapat menggunakan aplikasi *chatbot*. Pada halaman ini pengguna dapat melakukan komunikasi dengan *chatbot* untuk melakukan pemesanan surat di FTI. Tampilan halaman *chat* dapat dilihat pada Gambar 4.10.



Gambar 4.10 Halaman *Chat* Aplikasi

#### 4.4 Skenario Penggunaan Aplikasi

Padang Istigfar Almafazi atau biasa dipanggil Almafazi adalah mahasiswa jurusan teknik informatika FTI UII. Almafazi memiliki orang tua yang bekerja sebagai pegawai negeri sipil. Suatu saat Almafazi dihubungi oleh orang tuanya untuk meminta surat keterangan mahasiswa aktif dari kampus untuk keperluan pengurusan tunjangan sertifikasi. Oleh sebab itu ia membutuhkan sebuah surat keterangan mahasiswa aktif yang dikeluarkan oleh pihak kampus.

Dengan adanya aplikasi *chatbot* untuk pengajuan proses pemesanan surat, dimungkinkan dapat membantu Almafazi untuk bisa melakukan pengajuan pemesanan surat tanpa harus datang ke kampus. Hal tersebut di dukung dengan adanya aplikasi *chatbot* yang terintegrasi dengan sistem informasi surat di FTI UII, sehingga dinilai cukup memudahkan mahasiswa untuk dapat melakukan pengajuan permintaan surat di lingkungan FTI UII.

Untuk dapat menggunakan aplikasi *chatbot* tersebut cukup mudah, Almafazi hanya perlu untuk instalasi aplikasi pada perangkat seluler. Setelah itu login menggunakan nim dan

password yang dimiliki, kemudian Almafazi dapat berkomunikasi interaktif dengan maksud untuk mengajukan pemesanan surat menggunakan aplikasi *chatbot*. Hal tersebut dirasa sangat memudahkan mahasiswa karena aplikasi *chatbot* tersebut dapat memberikan automasi layanan dan juga dapat diakses 24 jam.

#### **4.5 Evaluasi**

Aplikasi *chatbot* untuk pengajuan proses pemesanan surat di FTI merupakan aplikasi yang diharapkan dapat memudahkan mahasiswa dalam melakukan pemesanan surat. Aplikasi ini diberi nama FTIbot yang berarti aplikasi *chatbot* FTI. Aplikasi *chatbot* ini merupakan aplikasi berbasis android, agar lebih fleksibel saat digunakan oleh pengguna dimanapun dan kapanpun. Selain itu, aplikasi *chatbot* juga dibuat dengan bahasa sehari-hari agar tidak membuat bingung penggunanya. Hal ini bertujuan membuat komunikasi lebih interaktif digunakan oleh pengguna. Aplikasi ini juga diharapkan mampu melakukan automasi layanan untuk melayani mahasiswa yang ingin mengajukan pemesanan surat.

#### **4.6 Pengujian Fungsionalitas**

Pada tahap ini, dilakukan pengujian yang berupa pengujian fungsionalitas terhadap tugas percakapan untuk menangani pemesanan surat dan melihat riwayat pemesanan surat. Pengujian dilakukan dengan membuat skenario pengujian dan *output* yang diharapkan atas skenario pengujian yang telah dibuat. Setelah itu dilakukan pengujian aplikasi sesuai skenario yang telah dibuat. Jika hasil yang diharapkan sesuai maka pengujian bernilai valid. Di bawah ini merupakan tabel hasil pengujian fungsionalitas *chatbot* untuk pengajuan proses pemesanan surat yang ditunjukkan pada Tabel 4.6.

Tabel 4.6 Hasil Pengujian Fungsionalitas *Chatbot*

Skenario pengujian	Input pengguna	Output yang diharapkan	Output hasil	Keterangan
Memulai percakapan dengan <i>chatbot</i>	Hai	<i>Chatbot</i> merespon dengan menawarkan bantuan dan menampilkan pilihan menu	Hai Masholeh, Selamat malam, Ada yang bisa dibantu? 1. Pesan Surat 2. Sejarah Pesan 3. Bantuan	Valid ( <i>chatbot</i> merespon)
Memasukkan percakapan yang bermaksud untuk melakukan pemesanan surat	Saya ingin pesan surat	<i>Chatbot</i> merespon dengan menampilkan pilihan jenis surat yang dapat di pesan	Hai Masholeh, Selamat malam, Ada ya Surat apa yang ingin anda pesan? 1. Surat Keterangan Mahasiswa Aktif 2. Surat Keterangan Akreditasi 3. Surat Keterangan Berkelakuan Baik 4. Surat Keterangan Beasiswa 5. Surat Pengantar Pembuatan Paspor	Valid ( <i>chatbot</i> merespon)
Melakukan pemesanan surat berdasarkan jenis surat yang di pilih	Surat keterangan Mahasiswa Aktif	<i>Chatbot</i> merespon dengan menanyakan untu kelengkapan data sesuai jenis surat yang di pesan	Silahkan jawab pertanyaan berikut ini untuk data pelengkap surat! - Apa maksud/keperluan surat? - Semester berapa anda? - Siapa nama orangtua/wali anda? - Dimana alamat rumah anda? - Apa pekerjaan orangtua/wali? kalau tidak ada isikan (-) - NIP orangtua/wali? kalau tidak ada isikan (-) - Dimana instansi orangtua/wali bekerja? kalau tidak ada isikan (-) - Apa pangkat/Golongan orangtua/wali? kalau tidak ada isikan (-)	Valid ( <i>chatbot</i> merespon)
Pengecekan riwayat pemesanan surat	Saya ingin cek riwayat pemesanan surat	<i>Chatbot</i> merespon dengan memberikan informasi daftar riwayat pemesanan surat sesuai dengan authentication pengguna	No : 1 - Jenis Surat : Surat Keterangan Mahasiswa Aktif - Tanggal Pesan : 24-11-2019 - Status : Menunggu Persetujuan  No : 2 - Jenis Surat : Surat Keterangan Berkelakuan Baik - Tanggal Pesan : 03-10-2019 - Status : Sudah diambil	Valid ( <i>chatbot</i> merespon)

#### 4.7 Pengujian Usabilitas

Tahap pengujian *chatbot* ini adalah dengan cara pengujian usabilitas, dimana menggunakan kuesioner sebagai medianya. Setelah aplikasi *chatbot* selesai dibangun, maka dijalankan sebuah pengujian menggunakan kuesioner untuk mendapat respon dari pengguna terhadap aplikasi yang telah dibuat. Pengujian dilakukan kepada 10 pengguna yang merupakan mahasiswa FTI. Pengujian dilaksanakan dengan cara meminta pengguna mencoba menggunakan aplikasi *chatbot* dan setelah itu, pengguna mengisi kuesioner yang sudah dibuat. Setelah kuesioner diisi oleh pengguna, akan didapat respon kelayakan atas *chatbot* yang telah dibuat, sudah sesuai atau masih ada yang kurang. Tabel 4.7 menunjukkan hasil dari kuesioner yang telah diisi oleh sampel pengguna.

Tabel 4.7 Hasil Pengujian Usabilitas *Chatbot*

No	Indikator	Pertanyaan	TS	KS	CS	S	SS
1	Lea	Apakah aplikasi chatbot memiliki tampilan aplikasi yang mudah dipahami dan mudah digunakan oleh pengguna?	-	-	-	7	3
2		Apakah aplikasi chatbot menggunakan bahasa percakapan yang mudah dipahami oleh pengguna?	-	-	-	3	7
3	Efe	Apakah aplikasi chatbot mampu memahami masukan pengguna dalam melakukan pemesanan surat?	-	-	1	5	4
4		Apakah aplikasi chatbot mampu memberikan respon yang sesuai dengan yang diharapkan?	-	1	2	4	3
5	MeR	Apakah aplikasi chatbot tidak membingungkan saat digunakan?	-	-	1	3	6
6		Apakah aplikasi chatbot minim memberikan kesalahan saat merespon permintaan pengguna?	-	1	4	4	1
7	Efi	Apakah aplikasi chatbot dapat memberikan manfaat kepada pengguna dalam melakukan pemesanan surat?	-	-	1	4	5
8		Apakah aplikasi chatbot dapat memproses permintaan pemesanan surat yang dilakukan oleh pengguna?	-	-	1	3	6
9	Sat	Apakah dengan adanya aplikasi chatbot untuk melakukan pemesanan surat dapat memberikan kemudahan kepada mahasiswa dalam melakukan pemesanan surat?	-	-	2	4	4
10		Apakah pengguna puas dengan adanya aplikasi chatbot untuk melakukan pemesanan surat?	-	1	2	4	3
Total			0	3	14	41	42

Setelah didapatkan data dari kuesioner, data dimasukkan ke dalam rumus perhitungan yang dapat dilihat pada bab sebelumnya yakni persamaan ( 3.1 ) dan persamaan ( 3.2 ). Perhitungan hasil pengujian usabilitas *chatbot* adalah sebagai berikut:

a. Nilai Pengujian

$$P1 = \frac{(1.0)+(2.0)+(3.0)+(4.7)+(5.3)}{10} = \frac{43}{10} = 4,3$$

$$P2 = \frac{(1.0)+(2.0)+(3.0)+(4.3)+(5.7)}{10} = \frac{47}{10} = 4,7$$

$$P3 = \frac{(1.0)+(2.0)+(3.1)+(4.5)+(5.4)}{10} = \frac{43}{10} = 4,3$$

$$P4 = \frac{(1.0)+(2.1)+(3.2)+(4.4)+(5.3)}{10} = \frac{39}{10} = 3,9$$

$$P5 = \frac{(1.0)+(2.0)+(3.1)+(4.3)+(5.6)}{10} = \frac{45}{10} = 4,5$$

$$P6 = \frac{(1.0)+(2.1)+(3.4)+(4.4)+(5.1)}{10} = \frac{35}{10} = 3,5$$

$$P7 = \frac{(1.0)+(2.0)+(3.1)+(4.4)+(5.5)}{10} = \frac{44}{10} = 4,4$$

$$P8 = \frac{(1.0)+(2.0)+(3.1)+(4.3)+(5.6)}{10} = \frac{45}{10} = 4,5$$

$$P9 = \frac{(1.0)+(2.0)+(3.2)+(4.4)+(5.4)}{10} = \frac{42}{10} = 4,2$$

$$P10 = \frac{(1.0)+(2.1)+(3.2)+(4.4)+(5.3)}{10} = \frac{39}{10} = 3,9$$

b. Skor Usabilitas  $= \frac{42,3}{10} = 4,23$

$$= 4,23$$

Dari hasil perhitungan, maka didapat nilai pengujian dan skor usabilitas. Setelah mengetahui nilai pengujian dan skor usabilitas, maka akan diketahui seberapa layak aplikasi yang telah dibuat oleh penulis. Kesimpulan hasil dari pengujian usabilitas dapat dilihat pada Tabel 4.8.

Tabel 4.8 Kesimpulan Hasil Pengujian Usabilitas *Chatbot*

No	Indikator	Pertanyaan	Nilai Pengujian	Keterangan	Rata-rata Pengujian	keterangan
1	Lea	Apakah aplikasi chatbot memiliki tampilan aplikasi yang mudah dipahami dan mudah digunakan oleh pengguna?	4,3	Layak	4,5	Layak
2		Apakah aplikasi chatbot menggunakan bahasa percakapan yang mudah dipahami oleh pengguna?	4,7	Sangat Layak		
3	Efe	Apakah aplikasi chatbot mampu memahami masukan pengguna dalam melakukan pemesanan surat?	4,3	Layak	4,1	Layak
4		Apakah aplikasi chatbot mampu memberikan respon yang sesuai dengan yang diharapkan?	3,9	Layak		
5	MeR	Apakah aplikasi chatbot tidak membingungkan saat digunakan?	4,5	Layak	4,0	Layak
6		Apakah aplikasi chatbot minim memberikan kesalahan saat merespon permintaan pengguna?	3,5	Cukup Layak		
7	Efi	Apakah aplikasi chatbot dapat memberikan manfaat kepada pengguna dalam melakukan pemesanan surat?	4,4	Layak	4,45	Layak
8		Apakah aplikasi chatbot dapat memproses permintaan pemesanan surat yang dilakukan oleh pengguna?	4,5	Layak		
9	Sat	Apakah dengan adanya aplikasi chatbot untuk melakukan pemesanan surat dapat memberikan kemudahan kepada mahasiswa dalam melakukan pemesanan surat?	4,3	Layak	4,1	Layak
10		Apakah pengguna puas dengan adanya aplikasi chatbot untuk melakukan pemesanan surat?	3,9	Layak		
Skor Usabilitas					4,23	Layak

Kesimpulan dari pengujian ini yaitu aplikasi chatbot sudah memenuhi kelayakan untuk dapat digunakan oleh mahasiswa untuk melakukan pengajuan proses pemesanan surat di FTI. Aplikasi chatbot yang dibuat sudah baik dan sudah memenuhi harapan dari tujuan aplikasi chatbot ini dibuat, yaitu aplikasi chatbot dapat memberikan kemudahan dan manfaat kepada mahasiswa FTI. Aplikasi chatbot yang telah dibangun juga masih terdapat beberapa kekurangan, yaitu aplikasi chatbot ini belum dapat digunakan oleh mahasiswa FTI karena masih bersifat lokal.