BAB II
LANDASAN TEORI

2.1 Sistem Informasi Surat *Online* (SISO) FTI

Sistem Informasi Surat *Online* (SISO) merupakan sistem informasi yang dikembangkan oleh Fakultas Teknologi Industri (FTI) Universitas Islam Indonesia (UII) dengan tujuan memudahkan administrasi surat. SISO merupakan sistem informasi yang dapat dimanfaatkan oleh mahasiswa maupun karyawan. Mahasiswa dapat melakukan pengajuan proses pemesanan surat secara *online*, dari manapun dan kapanpun menggunakan perangkat yang dimiliki. Sedangkan karyawan dapat memanfaatkan SISO untuk memudahkan penanganan pesanan surat yang masuk maupun administrasi surat tersebut. SISO dapat diakses secara *online* dengan mengakses alamat surat.fit.uii.ac.id. Secara spesifik, SISO memiliki fitur pesan surat *online* sebagai berikut:

a. Surat Keterangan Berkelakuan Baik
b. Surat Keterangan Aktif Kuliah
c. Surat Pernyataan Tidak Sedang Menerima Beasiswa Lain
d. Surat Pengantar Pembuatan Passport/Visa
e. Surat Keterangan Akreditasi
f. Surat Keterangan Alumni

2.2 Chatbot

Chatbot adalah sebuah program komputer yang bertujuan untuk mensimulasikan sebuah kecerdasan buatan untuk dapat melakukan sebuah percakapan dengan manusia (Shawar & Atwell, 2002). *Chatbot* merupakan implementasi dari bidang ilmu pengolahan bahasa alami, pembelajaran mesin, rekayasa perangkat lunak dan kecerdasan buatan. *Chatbot* dirancang untuk mensimulasikan percakapan dengan manusia menggunakan aturan atau kecerdasan buatan melalui melalui antarmuka percakapan melalui teks tertulis atau lisan. Sebuah *chatbot* dapat berjalan menggunakan pembelajaran mesin melalui kecerdasan buatan untuk menangkap pola percakapan yang memungkinkannya untuk meniru percakapan manusia dan bereaksi terhadap permintaan tertulis atau lisan sehingga *chatbot* mampu merespon dengan balasan yang sesuai untuk memberikan layanan atau informasi. *Chatbot* juga dapat diintegrasikan dengan sumber data yang ada sehingga dapat memberikan informasi atau layanan sesuai permintaan

pengguna, seperti memberikan informasi perkiraan cuaca, berita terbaru atau pemesanan kamar hotel.

Chatbot dapat dikembangkan sesuai dengan fungsi atau kebutuhan yang ada, namun setiap *chatbot* yang dikembangkan memiliki tipe tersendiri dalam memproses setiap permintaan pengguna, berikut adalah tipe–tipe *chatbot*:

a. Button-Based Chatbot

Button-based chatbot adalah tipe *chatbot* yang mengharuskan pengguna untuk membuat beberapa pilihan pengguna berdasarkan tombol perintah yang telah disediakan oleh *chatbot* tersebut. Tombol perintah yang disediakan oleh sebuah *button-based chatbot* adalah sebuah hirarki pohon keputusan. Pada tipe *chatbot* ini *knowledge base* yang disiapkan oleh pengembang tidak terlalu banyak, karena masukan pengguna hanya berdasarkan tombol yang disediakan oleh sistem. Kelemahan dari tipe *chatbot* ini adalah pengguna tidak dapat memberi masukan atau pertanyaan dengan leluasa, karena setiap tombol telah diatur dengan sebuah *query* untuk sebuah masukan.

b. Keyword Recognition-Based Chatbot

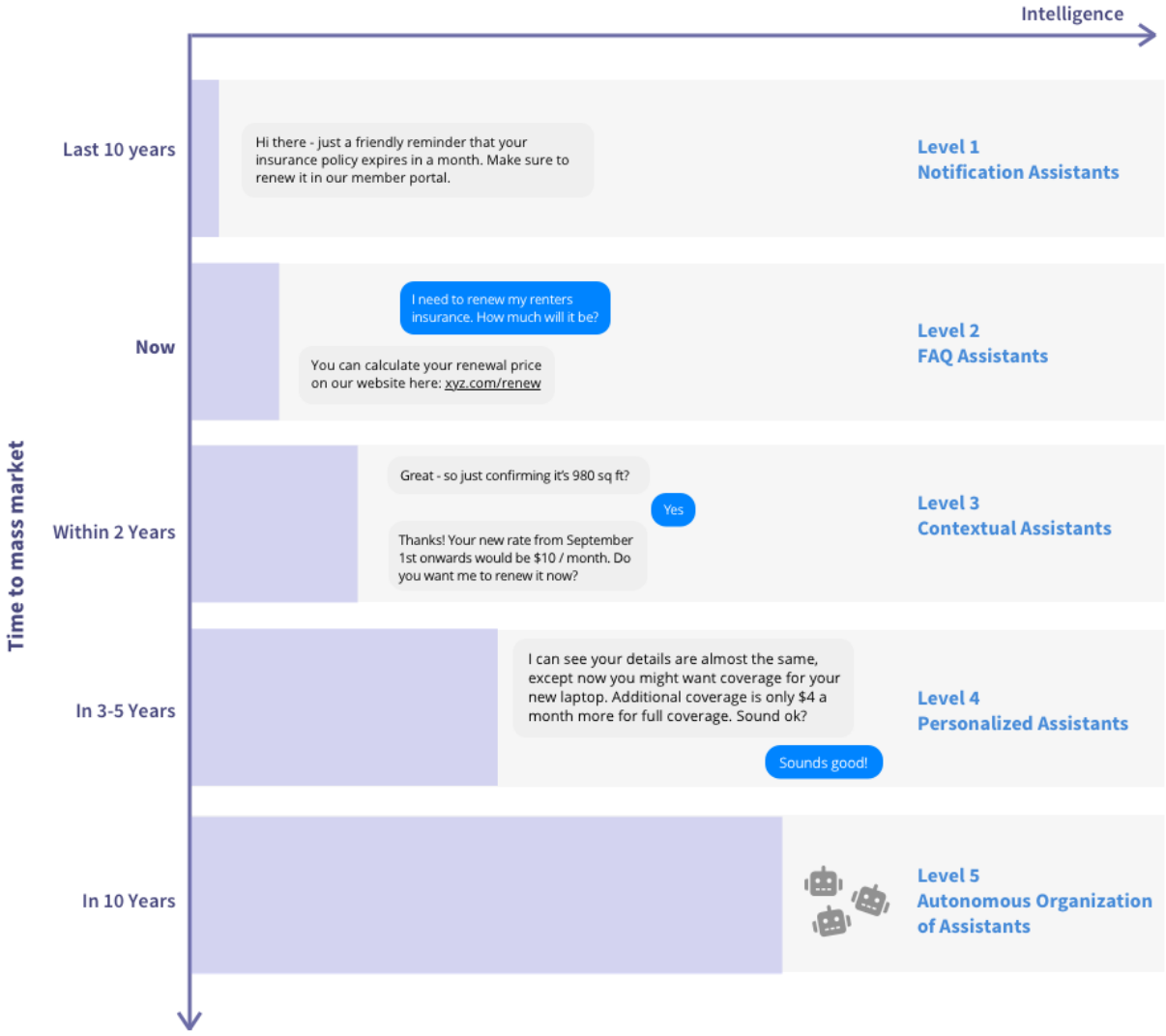
Keyword recognition-based chatbot adalah tipe sebuah *chatbot* yang dapat mengenali serta merespon permintaan pengguna berdasarkan beberapa kata kunci yang ada pada permintaan pengguna. *chatbot* dapat menerima permintaan pengguna berupa kalimat atau susunan kata. *chatbot* tipe ini dapat menerima permintaan pengguna berupa kalimat atau susunan kata. *Chatbot* ini menggunakan kata kunci dan kecerdasan buatan yang dapat disesuaikan untuk menentukan cara melayani respons yang sesuai bagi pengguna. Cara *chatbot* menentukan sebuah respon atau balasan yang tepat kepada sebuah masukan yang telah diberikan adalah mengecek setiap kata sampai menemukan kata kunci yang cocok dengan *knowledge base* pada sistem. Tipe *chatbot* ini mempunyai kelemahan yaitu ketika merespon sebuah masukan yang mempunyai kata kunci yang sama.

c. Contextual Chatbot

Contextual chatbot adalah sebuah tipe *chatbot* yang paling canggih dari tipe *chatbot* sebelumnya. *Chatbot* ini memanfaatkan *machine learning* dan *artificial intelligent* untuk mengolah dan mengingat percakapan dengan pengguna untuk dipelajari dan meningkatkan kemampuan percakapan. *Contextual chatbot* dalam menerima sebuah masukan sama dengan *keyword recognition-based chatbot* yaitu menerima sebuah masukan berupa susunan kata atau kalimat. Perbedaan *contextual chatbot* dengan tipe *chatbot* yang lain yaitu setiap masukan pengguna akan dapat diingat dan dipelajari lagi sehingga ketika ada sebuah masukan dengan

susunan kata berbeda namun memiliki arti yang sama, sistem mampu memberi respon dengan tepat. *Contextual chatbot* memerlukan pengembangan *knowledge base* yang harus dikembangkan secara berkala, agar dapat mempelajari setiap masukan pengguna.

Dalam perkembangan *chatbot* juga terdapat beberapa level, setiap level perkembangan juga memiliki tingkat kecerdasan yang berbeda dan terus meningkat. level dari perkembangan *chatbot* sebagai kercerdasan buatan asisten dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Levels of Development Chatbot AI Assistant*

Sumber: https://blog.rasa.com/conversational-ai-your-guide-to-five-levels-of-ai-assistants-in-enterprise/ (Di akses 18 September 2019)

Berikut penjelasan dari setiap level perkembangan *chatbot*:

1. Level 1: *Notification Assistants*

Perkembangan *chatbot* pada tingkat ini terjadi pada sudah 10 tahun terakhir. *Chatbot* hanya sebatas pemberitahuan informasi yang dikirimkan kepada pengguna dan belum bisa menerima masukan dari pengguna.

2. Level 2: *FAQ Assistants*

Perkembangan *chatbot* pada tingkat ini merupakan jenis *chatbot* yang paling umum. Asisten memungkinkan pengguna untuk mengajukan pertanyaan sederhana dan mendapatkan respons. Beberapa asisten FAQ juga memungkinkan untuk dialog dasar yang berarti Anda dapat melakukan percakapan multi-langkah dan bolak-balik dasar.

3. Level 3: *Contextual Assistants*

Perkembangan *chatbot* pada tingkat ini sudah mulai banyak dimulai oleh pengembang perangkat lunak untuk membuat atau mengembangkan *Chatbot* Assistants. *Chatbot* dapat menerima permintaan pengguna berupa kalimat atau susunan kata. masukan atau permintaan dari pengguna akan dapat diingat dan dipelajari lagi sehingga ketika ada sebuah masukan dengan susunan kata berbeda namun memiliki arti yang sama, sistem mampu memahami dan menanggapi masukan yang berbeda dan tidak terduga dari pengguna.

4. Level 4: *Personalized Assistants*

Perkembangan *chatbot* pada tingkat ini adalah perkembangan yang mulai diharapkan oleh pengembang perangkat lunak pada umumnya dan diprekdisi mulai banyak digunakan pada 3-5 tahun mendatang, yaitu *Chatbot Assistants* yang dapat memahami kapan saat yang tepat untuk berkomunikasi dan secara proaktif menjangkau berdasarkan konteks percakapan serta mengingat preferensi pengguna dan memberi antarmuka utama yang dipersonalisasi. *Chatbot* akan mampu mengenal pengguna lebih detail, sehingga tidak perlu menanyakan setiap detail.

5. Level 5: *Autonomous Organization of Assistants*

Pada akhirnya, akan ada *Chatbot AI Assistants* yang dapat mengenal setiap pengguna secara pribadi dan pada akhirnya menjalankan sebagian besar pekerjaan yang dilakukan oleh manusia, mulai dari generasi pemimpin di bidang pemasaran, penjualan, SDM, atau keuangan. Ini adalah visi yang diharapkan oleh para pengembang perangkat lunak untuk satu dekade lagi, yaitu 10 tahun yang akan mendatang bukan tidak mungkin akan menjadi suatu kenyataan.

2.2.1 *Chatbot Development Platform*

Chatbot Development platform adalah sebuah sistem untuk membuat dan mengembangkan sebuah layanan *chatbot*. Adanya sebuah *platform* untuk mengembangkan sebuah *chatbot* membuat pengembang menjadi lebih mudah dalam membuat dan mengatur *knowledge base, intent, machine learning*, dan lain–lain. *Chatbot platform* dapat menghubungkan sebuah *chatbot* yang dikembangkan dengan aplikasi yang sedang dikembangkan maupun sebuah messenger *platform* yang mendukung sebuah *chatbot*, seperti Line, Facebook Messenger, Telegram, dan lain–lain. Berikut merupakan *chatbot develoment platform*:

a. Dialogflow

Google Dialogflow adalah sebuah *platform* yang berbasis pengolahan bahasa alami untuk para pengembang perangkat lunak untuk membuat sebuah sistem *chatbot*. Google Dialogflow sebelum diakuisi oleh perusahaan Google pada September, 2016 bernama Api.ai. Api.ai awalnya dimiliki oleh perusahaan Speaktoit, sebuah perusahaan yang dikenal dengan aplikasi virtual buddy berbasis sistem operasi Android, iOS, dan windows phone yaitu Assistant. Perusahaan Google membeli Api.ai milik Speaktoit, karena *platform* tersebut menyediakan alat–alat untuk mengembangkan aplikasi – aplikasi pada Google Virtual Assistant. Pada 10 Oktober 2017 Api.ai resmi berganti nama menjadi Dialogflow. Dialogflow didukung oleh pembelajaran mesin Google, yang dapat digunakan untuk terhubung ke pengguna di Google Assistant, Amazon Alexa, aplikasi Mobile, Messenger, situs web, Slack, Twitter, dan lain-lain.

b. Luis.ai

Microsoft LUIS (*Language Understanding Intelligent Service*) atau Luis.ai merupakan sebuah produk kecerdasan buatan untuk layanan pengolahan bahasa alami milik perusahaan Microsoft. Luis.ai digunakan untuk aplikasi komunikasi antara pengguna dengan sebuah sistem menggunakan sebuah bahasa alami. Pengembang dapat mempublikasikan sebuah aplikasi yang dibuat pada Luis.ai untuk dapat mengambil sebuah *file* json sebagai sebuah respon sistem dari masukan pengguna. Luis.ai menawarkan satu set Rest Api yang dapat digunakan oleh pengembang perangkat lunak untuk mengotomatiskan proses pembuatan aplikasi. Bahasa yang didukung adalah Inggris, Prancis, Italia, Jerman, Spanyol, Portugis Brasil, Jepang, Korea, dan Cina. Pengembang dapat memilih bahasa yang akan digunakan oleh aplikasi dalam melakukan percakapan dengan pengguna

c. Wit.ai

Wit.ai adalah mesin pengolahan bahasa alami milik perusahaan Facebook yang dapat dikembangkan oleh pengembang perangkat lunak untuk membuat percakapan cerdas. Wit.ai menyediakan antarmuka yang mudah dan API pembelajaran cepat untuk memahami komunikasi manusia dari setiap interaksi dan membantu mengurai pesan kompleks (yang bisa berupa suara atau teks) menjadi data terstruktur. Bahasa pemrograman dan dukungan integrasi, yaitu Node.js, klien Python, klien Ruby, dan HTTP API.

d. Rasa.ai

Rasa.ai adalah *open source machine learning framework* (kerangka pembelajaran mesin) untuk percakapan cerdas berbasis teks atau lisan. Rasa.ai dapat memahami masukan pengguna, mengadakan percakapan dengan pengguna dan terhubung dengan *platform* komunikasi dan API. Rasa.ai bekerja pada dua komponen utama yaitu Rasa NLU dan Rasa Core. Rasa NLU adalah *natural language processing tools* atau alat pengolah bahasa alami *open source*, digunakan untuk klasifikasi maksud (*intent classification*) dan ekstraksi entitas (*entity extraction*) di dalam percakapan, kemudian menggunakan *machine learning* untuk mengambil pola dan menggeneralisasi untuk kalimat yang tidak terlihat. Rasa Core adalah sebuah *chatbot framework* atau kerangka obrolan *open source* untuk menangani percakapan kontekstual, digunakan untuk managemen percakapan berbasis *machine learning-based*.

Dari beberapa *platform chatbot development* yang telah penulis bahas sebelumnya, penulis telah membuat komparasi berdasarkan fitur unggulan, lisensi pengguna, *integrated channels* dan dukungan bahasa dari setiap *platform chatbot*. Tabel 2.1 Menunjukkan komparasi *platform chatbot* dari keempat *platform* yang telah penulis bahas sebelumnya.

Tabel 2.1 Komparasi *Platform Chatbot*

No	***Chatbot Platform***	**Fitur Unggulan**	**Lisensi Pengguna**	**Integrated Channels**	**Bahasa**
1	Google Dialogflow	- Mencocokkan *query* dengan intent yang paling cocok berdasarkan informasi yang ada pada suatu *intent* - *Prebuilt Agent* - Merubah sebuah masukan *query* menjadi sebuah JSON yang dapat diambil sebagai data tambahan pada saat memberikan *output*	*Free, Enterprise Edition*	Google Assistant, Facebook Messenger, Slack, Viber, Twitter, Twillio, Skype, Telegram, Line	Danish, German, English, Spanish, French, Hindi, Italian, Japanense, Korean, Dutch, Norwegian, Portuguese, Russian, Swedish, Ukrainian, Chinese.
2	Microsoft Luis.ai	- Prebuilt Domain Intent - Men-deploy sebuah model *chatbot* ke HTPP endpoint dengan mudah - *Review Endpoint Utterances*	*Free, Text Requests* dan *Speech Requests*	-	English
3	Facebook Wit.ai	- *Allow to use: Entities, Intents, Context, Actions* - *Natural Languange Process* (NLP)	*Free*	Voice text	English
4	Rasa.ai	- *Intents Classification* - *Entity Extractions* - *Custom Action*	*Open Source*	Runs Locally	German, English

2.3 RASA.AI

Rasa.ai adalah *open source machine learning framework* untuk membangun asisten AI dan obrolan kontekstual. Rasa memiliki lebih dari setengah juta unduhan sejak diluncurkan. Rasa.ai juga memiliki *platform* antarmuka pengguna yaitu Rasa X. Rasa X adalah alat yang

dirancang untuk penggunaan yang membantu pengembang perangkat lunak untuk membangun, meningkatkan, dan menggunakan asisten AI yang didukung oleh kerangka rasa. Rasa.ai bekerja pada dua komponen utama yaitu Rasa NLU dan Rasa Core.

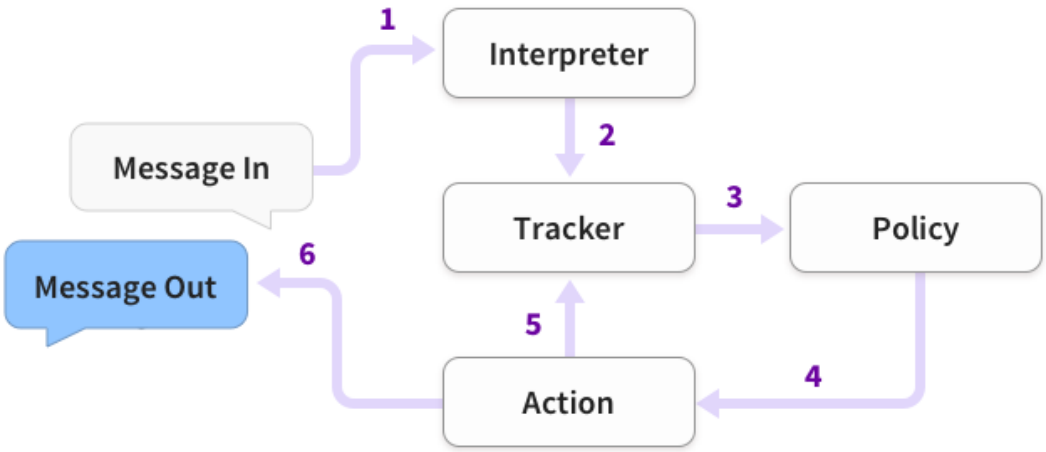
a. Rasa Core

Rasa Core adalah sebuah *chatbots framework* atau kerangka obrolan *open source* untuk menangani percakapan kontekstual, digunakan untuk managemen dialog untuk mengadakan percakapan dan memutuskan apa yang harus dilakukan selanjutnya.

b. Rasa NLU

Rasa NLU adalah *natural language processing tools* atau alat pengolah bahasa alami *open source*, digunakan untuk klasifikasi maksud (*intent classification*) dan ekstraksi entitas (entity extraction) di dalam percakapan, kemudian menggunakan machine learning untuk mengambil pola dan menggeneralisasi untuk kalimat yang tidak terlihat.

Gambar 2.2 menunjukkan langkah-langkah dasar tentang bagaimana *Assistant* AI yang dibangun dengan Rasa menanggapi pesan.



Gambar 2.2 *Architecture Message Handling of Rasa*

Sumber:

https://rasa.com/docs/rasa/*user*-guide/architecture/ (Di akses 18 September 2019)

Langkah-langkahnya adalah:

1. Pesan diterima dan diteruskan ke *interprete* atau penerjemah yang mengubahnya menjadi kamus termasuk teks asli, maksud, dan entitas apa pun yang ditemukan. Bagian ini ditangani oleh Rasa NLU.

2. Kemudian diteruskan ke *tracker*. *Tracker* adalah objek yang melacak status percakapan. Pada proses ini menerima info bahwa pesan baru telah masuk.
3. Setelah itu *Policy* menerima status pelacak saat ini.
4. Kemudian *policy* akan memilih tindakan mana yang akan diambil selanjutnya.
5. *Action* atau tindakan yang dipilih dicatat oleh *tracker*.
6. Tanggapan dikirimkan kepada pengguna.

2.3.1 *Stories*

Stories merupakan gambaran umum percakapan nantinya, kita harus menulis beberapa *story* untuk memulai membuat data training. Rasa Core bekerja dengan belajar dari contoh percakapan,berikut ini merupakan contoh dari *stories* data, *stories* data disimpan dalam format *markdown* (*stories*.md). Dalam percakapan yang sangat sederhana ini, pengguna menyapa *bot*, dan *bot* menyapa kembali. Contoh penulisan *stories* dapat dilihat pada Gambar 2.3.

```
## story1
 * salam
  - utter_greet
```

Gambar 2.3 Contoh dokumen *stories*

Sebuah *story* dimulai dengan ## diikuti oleh nama dari *story* tersebut (opsional). baris yang dimulai dengan * adalah pesan yang dikirim oleh pengguna. Meskipun pengguna tidak menulis pesan yang sebenarnya, melainkan niat (dan entitas) yang mewakili apa yang dimaksud pengguna. Garis yang dimulai dengan - adalah tindakan yang diambil oleh *bot*. Dalam hal ini semua tindakan *bot* hanyalah pesan yang dikirim kembali ke pengguna, seperti utter_greet, tetapi secara umum suatu tindakan dapat melakukan apa saja, termasuk memanggil API dan berinteraksi dengan aplikasi lain.

2.3.2 NLU Data

Sebuah *bot* harus dapat memahami bahasa asli (makna), bukan hanya *Input* terstruktur, Interpeter bertanggung jawab atas penguraian pesan. Interpreter melakukan *Natural Language Understanding* (NLU) dan mengubah pesan menjadi *output* terstruktur. Maka dari itu, dalam Rasa NLU perlu untuk mendefinisikan pesan pengguna yang harus bisa ditangani oleh *bot*,

yaitu dalam *markdown* format (nlu.md). Contoh penulisan NLU Data dapat dilihat pada Gambar 2.4.

```
## intent:introduction
- my name is [john](name)
- I am [smith](name)
- I'm [peter cech](name)
- you call me [jamesh](name)
```

Gambar 2.4 Contoh dokumen NLU Data

Sebuah *entity* terdapat di dalam sebuah varian *intent*, "`[john]`" menunjukkan frasa *entity* di dalam *entity* "`(name)`".

2.3.3 *Action*

Action adalah hal-hal yang akan dijalankan bot sebagai respon terhadap *Input* pengguna. *Action* disimpan dalam format Python (actions.py). Ada tiga jenis *Action* dalam Rasa Core:

a. Default Action

Terdapat 3 *action default* di dalam Rasa AI yaitu *action_listen, action_restart, action_default_fallback*. *action_listen* digunakan agar Rasa Core berhenti memprediksi *Input*, dan menunggu pengguna meng*Input*kan sesuatu. action_restart digunakan untuk menyetel ulang seluruh percakapan, biasanya dipicu dengan menggunakan */restart*. *action_default_fallback* digunakan untuk membatalkan pesan pengguna terakhir (seolah-olah pengguna tidak mengirimnya) atau ketika pengguna mengucapkan pesan yang tidak dimengerti oleh *chatbot*.

b. Utter Action

Action ini hanya digunakan untuk melakukan pengiriman pesan sebagai *response* kepada pengguna (Respon dari *chatbot*).

c. Custom Action

Suatu *action* yang dapat menjalankan kode apa pun yang diinginkan biasanya digunakan untuk mengakses aplikasi eksternal melalui REST API. Rasa Core akan memanggil endpoint yang telah ditentukan ketika prediksi *Input user* mengarah ke *Custom Action*, *endpoint* harus merupakan server web yang sudah di atur agar dapat bereaksi ketika mendapatkan panggilan (REST API) dari *webhook* dari Rasa Core, lalu server web menjalankan kode dan secara

opsional mengembalikan informasi untuk mengubah percakapan. *Action* memerlukan *rasa_core_sdk* agar dapat berjalan, untuk instalasi *rasa_core_sdk* akan dijelaskan pada bab penginstalan.

Berikut merupakan contoh penulisan *action*, contohnya adalah ketika pengguna mengatakan "*show me a Mexican restaurant*", *bot* dapat melakukan tindakan *ActionCheckRestaurants*. Contoh penulisan *action* yang ditulis menggunakan bahasa Python dapat dilihat pada Gambar 2.5.

```
from rasa_core_sdk import Actionfrom rasa_core_sdk.events import SlotSet
class ActionCheckRestaurants(Action):
def name(self):
# type: () -> Text
return "action_check_restaurants"

def run(self, dispatcher, tracker, domain):
# type: (CollectingDispatcher, Tracker,Dict[Text, Any]) ->
     List[Dict[Text, Any]]

cuisine = tracker.get_slot('cuisine')
q = "select * from restaurants where cuisine='{0}' limit
    1".format(cuisine)
result = db.query(q)
return [SlotSet("matches", result if result   is not None else [])]
```

Gambar 2.5 Contoh dokumen *Action*

Nama *action* *action_check_restaurants* harus ditambahkan ke *file domain*. Pada kode python di atas, metode '*run*' menerima tiga argumen. untuk mengakses nilai *slot* dan pesan terbaru yang dikirim oleh pengguna dapat menggunakan *object tracker*, untuk mengirim pesan kembali ke pengguna dapat menggunakan *object dispatcher*.

2.3.4 *Endpoint*

Endpoints berfungsi untuk memberi tahu di *port* mana aplikasi *chatbot* akan berjalan. *Endpoints* disimpan dalam format (endpoints.yml). Contoh penulisan *endpoints* dapat dilihat pada Gambar 2.6.

```
action_endpoint:
  url: "http://localhost:5055/webhook"
```

Gambar 2.6 Contoh dokumen *Endpoint*

2.3.5 NLU *Config*

NLU *Config* merupakan konfigurasi untuk Rasa NLU yang berisi mengenai *pipeline* dan bahasa yang digunakan. NLU *Config* disimpan dalam format (nlu_config.yml). Contoh penulisan NLU *Config* dapat dilihat pada Gambar 2.7.

```
language: "en"
pipeline: tensorflow_embedding
```

Gambar 2.7 Contoh dokumen NLU Config

2.3.6 *Policy*

Policy memutuskan tindakan apa yang harus diambil pada setiap langkah dalam dialog. *Policy* disimpan dalam format (policy.yml). Contoh penulisan *Policy* dapat dilihat pada Gambar 2.8.

```
policies:
   - name: KerasPolicy
     epochs: 200
  max_history: 3
   - name: MemoizationPolicy
 max_history: 3
```

Gambar 2.8 Contoh dokumen Policy

2.3.7 *Domain*

Domain mendefinisikan semesta percakapan. *domain* menentukan *intents*, *entities*, *slot*, dan *action* yang harus diketahui *bot*. *Domain* disimpan dalam format (domain.yml). Contoh penulisan *domain* dapat dilihat pada Gambar 2.9.

```
intents:
 - greet
 - default
 - goodbye
 - affirm
 - thank_you
 - change_bank_details
 - simple
 - hello
 - why
 - next_intent

entities:
 - name

slots:
  name:
    type: text

templates:
  utter_greet:
    - "hey there {name}!"
  utter_goodbye:
    - "goodbye 😢"
    - "bye bye 😢"
  utter_default:
    - "default message"

actions:
  - utter_default
  - utter_greet
  - utter_goodbye
```

Gambar 2.9 Contoh dokumen Domain

2.4 Lirerature Review

Literature review yang dilakukan pada penelitian ini adalah dengan meninjau ulang penelitian – penelitian terdahulu yang mempunyai topik yang sama, yaitu tentang pemngembangan *chatbot* untuk kasus serupa sebagai referensi penulis untuk mengembankan aplikasi yang sedang penulis buat yaitu aplikasi *chatbot* untuk pengajuan proses pemesanan

surat di FTI UII. Berikut ini adalah beberapa penelitian serupa yang penulis rangkum ke dalam beberapa paragraf.

Sorna Shanthi.D, Keerthana.S, Nandha Kumar.PK, Nithya.D (2019) melakukan penelitian dengan judul "*Hexabot: A Text-Based Assistive Chatbot To Explore Library Resources*" untuk mempermudah mahasiswa untuk mengakses perpustakaan universitas untuk memastikan ketersediaan buku tanpa mengunjungi perpustakaan. Tujuan utama dari penelitian tersebut adalah mengeksplorasi sumber daya perpustakaan untuk pengguna dengan memanfaatkan teknologi kecerdasan buatan untuk komunikasi pengguna dan *chatbot*.

Angga Bachtiar (2018) melakukan penelitian dengan judul "Pengembangan Aplikasi Web Q&A untuk Prosedur KP, TA dan KKN Jurusan Teknik Informatika UII dengan Menggunakan Google Dialogflow" untuk mempermudah mahasiswa dalam mencari informasi seputar KP, TA, dan KKN melalui layanan tanya-jawab yang interaktif. Dalam penelitian tersebut aplikasi dikembangkan dengan memanfaatkan teknologi kecerdasan buatan yaitu *chatbot*, dengan menggunakan *chatbot* mahasiswa dapat menggunakan aplikasi tersebut kapan pun dan dapat diakses diamana pun, agar dapat secara cepat mengetahui informasi seputar prosedur KP, TA, dan KKN.

Ruspandi R. Benedictus, Hans Wowor, Alwin Sambul (2017) melakukan penelitian dengan judul "Rancang Bangun *Chatbot Helpdesk* untuk Sistem Informasi Terpadu Universitas Sam Ratulangi", pada penelitian tersebut menghasilkan aplikasi *Chatbot Helpdesk* yang bisa membantu *user* dengan menjawab pertanyaan seputar penggunaan aplikasi-aplikasi dalam Sistem Informasi Terpadu Universitas Sam Ratulangi. *Chatbot* yang dibuat merupakan aplikasi *helpdesk* bertipe *Question Answering*, sehingga bantuan yang ditawarkan adalah menjawab pertanyaan *user*.

Setelah melakukan ulasan terhadap beberapa peneletian sebelumnya, penulis mendapati kesimpulan bahwa semua penelitian memanfaatkan teknologi kecerdasan buatan untuk memudahkan pengguna yaitu mahasiswa untuk mencari informasi dengan menggunakan *chatbot*. *Chatbot* yang dibuat pada dasarnya untuk layanan interaktif berupa tanya-jawab yang dilakukan oleh pengguna dan *bot.* Adapun dalam penelitian yang penulis lakukan sekarang ini memiliki sebuah pembeda dari penelitian yang dilakukan sebelumnya di mana penulis akan mengimplementasikan *chatbot* tidak hanya sebagai layanan interaktif berupa tanya-jawab namun juga dibuat untuk automatisasi layanan untuk pengajuan proses pemesanan surat di FTI-UII. Berikut ini merupakan tabel tambahan yang penulis buat untuk menunjukkan perbedaan antara setiap penelitian pada Tabel 2.2

Tabel 2.2 *Literature Review*

No	Penelitian	Judul	*Platform*	Analisis	Keterangan
1	Sorna Shanthi.D, Keerthana.S, Nandha Kumar.PK, Nithya.D (2019)	*Hexabot: A Text-Based Assistive Chatbot To Explore Library Resources*	Dialogflow	*Chatbot* digunakan untuk mempermudah mahasiswa untuk mengakses perpustakaan universitas untuk memastikan ketersediaan buku tanpa mengunjungi perpustakaan	Tipe *chatbot* yang digunakan adalah *FAQ Assistants*
2	Angga Bachtiar (2018)	Pengembangan Aplikasi Web Q&A untuk Prosedur KP, TA dan KKN Jurusan Teknik Informatika UII dengan Menggunakan Google Dialogflow	Dialogflow	*Chatbot* digunakan untuk mempermudah mahasiswa dalam mencari informasi seputar KP, TA, dan KKN melalui layanan tanya-jawab yang interaktif	Tipe *chatbot* yang digunakan adalah *FAQ Assistants*
3	Ruspandi R. Benedictus, Hans Wowor, Alwin Sambul (2017)	Rancang Bangun *Chatbot Helpdesk* untuk Sistem Informasi Terpadu Universitas Sam Ratulangi	-	*Chatbot* digunakan untuk membantu *user* dengan menjawab pertanyaan seputar penggunaan aplikasi-aplikasi dalam Sistem Informasi Terpadu Universitas Sam Ratulangi	Tipe *chatbot* yang digunakan adalah *FAQ Assistants*