

## **BAB III**

### **LANDASAN TEORI**

#### **3.1. *Artificial Intelligence***

*Artificial Intelligence* (AI) memiliki tujuan yang pasti yaitu memahami kecerdasan dan membangun sistem kecerdasan [7]. Bagi sarjana ilmu komputer, AI adalah mesin cerdas yang memiliki perilaku mirip manusia, dan menunjukkan perilaku cerdas. AI adalah studi tentang bagaimana membuat komputer melakukan hal-hal dimana, saat ini, orang lebih baik.

#### **3.2. *Machine Learning***

*Machine Learning* (ML) [8] adalah bagian dari AI, yang membuat model matematika berdasarkan data sampel, yang dikenal sebagai data latih, untuk menghasilkan prediksi atau keputusan tanpa diprogram secara eksplisit untuk melakukan tugas tersebut. ML sudah mengalami perkembangan yang pesat sehingga memunculkan banyak metode untuk menyelesaikan suatu permasalahan yang dapat dimodelkan dalam model matematika. Dengan ML, baik di bidang industri, pertanian, dan bidang lain juga semakin maju, mangkus dan sangkil.

#### **3.3 *Natural Language Processing***

*Natural Language Processing* (NLP) [9] adalah suatu bidang AI yang memberikan kemampuan kepada sistem komputer bagaimana menginterpretasikan dan mengontrol bahasa natural baik dalam bentuk lisan maupun tulisan. NLP digunakan untuk mengolah bahasa natural

sehingga sistem komputer dapat memahaminya dan melakukan suatu tugas yang berkaitan dengan bahasa natural. Contoh penerapan NLP, yaitu menerjemahkan kalimat, membuat rangkuman dari sekumpulan paragraf, analisis sentimen pada kalimat, mengubah suara menjadi teks dan sebaliknya.

### **3.4. *Natural Language Understanding***

*Natural Language Understanding* (NLU) menurut Gartner [10] adalah pemahaman komputer terhadap struktur dan makna dari bahasa manusia sehingga membuat pengguna dapat berinteraksi dengan komputer menggunakan kalimat natural. *Chatbot* menggunakan NLU [11] untuk mengambil konteks dari masukan pengguna dalam bahasa manusia yang tidak terstruktur dan merespon berdasarkan maksud dari pengguna. NLU dapat melakukan *intent classification* dan *entity extraction* berdasarkan konteks informasi. Entitas dapat didefinisikan oleh sistem atau pengguna. Konteks yang dimaksud pada NLU berupa string yang menyimpan objek yang pengguna acukan. *Intent classification* adalah proses klasifikasi *intent* dari *query* pengguna.

### **3.5. *Neural Networks***

*Neural Networks* (NNs) adalah jaringan-jaringan sel saraf dalam otak manusia dan hewan [12]. Pada konteks ilmu komputer, NNs dimodelkan dalam matematika sehingga dapat memberikan algoritma yang mampu menirukan cara kerja sel saraf manusia. Oleh karena itu, NNs juga bisa disebut sebagai *Artificial Neural Networks* (ANNs). NNs

terdiri dari satu atau lebih lapisan (*layer*) dari neuron atau unit kecil, yang juga disebut *perceptron* atau *node*. Setiap neuron menerima input dari neuron-neuron pada lapisan sebelumnya, mengalikannya dengan bobot (*weights*) tertentu, dan menjumlahkan hasilnya. Kemudian, input yang dihasilkan akan diproses melalui fungsi aktivasi sebelum diteruskan ke neuron-neuron pada lapisan selanjutnya. Beberapa konsep penting dalam NNs adalah:

1. *Input Layer*

Lapisan pertama dari jaringan, menerima input dari data yang diberikan.

2. *Hidden Layer*

Lapisan-lapisan antara input dan *output layer* yang berisi neuron-neuron tersembunyi yang melakukan proses pemrosesan dan representasi fitur dari input.

3. *Output Layer*

Lapisan terakhir dari jaringan yang menghasilkan *output* prediksi atau hasil dari proses komputasi.

4. Bobot (*Weights*)

Setiap sambungan antara neuron memiliki bobot tertentu yang menentukan seberapa kuat atau lemahnya pengaruh input terhadap neuron berikutnya.

5. Fungsi Aktivasi

Fungsi yang diterapkan pada *output* dari setiap neuron untuk

mengintroduksi non-linearitas dalam model dan membantu jaringan untuk mempelajari hubungan yang kompleks dalam data.

#### 6. *Forward Propagation*

Proses mengalirkan input data dari input layer hingga output layer untuk menghasilkan prediksi.

#### 7. *Backpropagation*

Algoritma untuk menghitung gradien kesalahan (*loss*) jaringan terhadap bobotnya, sehingga model dapat mengoreksi bobotnya selama proses pelatihan dan meningkatkan performa.

### 3.6. *Chatbot*

*Chatbot* adalah sebuah program komputer yang melakukan konversasi menggunakan dialog atau teks [13]. Dalam referensi [14] ada 2 pendekatan dalam mengembangkan *chatbot* berdasarkan algoritma dan teknik yang digunakan, yaitu pendekatan *pattern matching* dan pendekatan ML. *Chatbot* dapat mengakses sebuah cakupan pengetahuan, yang menunjukkan ranah pengetahuannya. *Chatbot* yang dapat menjawab semua pertanyaan yang mencakup apapun ranah pengetahuannya disebut *generic chatbot*. *Chatbot* yang dapat menjawab pertanyaan dalam jumlah ranah pengetahuan lebih dari satu disebut *open domain chatbot*. *Chatbot* yang hanya dapat menjawab pertanyaan dalam satu ranah pengetahuan saja disebut *closed domain chatbot*.

Berdasarkan pelayanan yang diberikan, *chatbot* dikategorikan menjadi 3. *Interpersonal chatbot* adalah *chatbot* yang memberikan

pelayanan seperti pemesanan tiket pesawat, pemesanan makanan di restoran, dan mencari informasi dari FAQ tanpa menjadi semakin ramah terhadap pengguna. *Intrapersonal chatbot* adalah *chatbot* yang menguasai ranah penggunanya dan mengerti kebutuhan penggunanya. *Inter-agent chatbot* adalah *chatbot* yang dapat berkomunikasi dengan *chatbot* lain seperti yang dilakukan oleh Alexa dan Cortana.

Berdasarkan tujuan utama *chatbot* dibuat, *chatbot* dibagi menjadi *informative*, *chat-based/conversational*, dan *task-based chatbot*. *Informative chatbot* digunakan untuk memberikan informasi spesifik kepada pengguna. *Chat-based/Conversational chatbot* dapat berkomunikasi secara natural dengan pengguna. *Task-based chatbot* dapat melakukan berbagai macam tugas, seperti memesan kamar hotel, dan bagus dalam meminta informasi dan merespon pengguna dengan tepat.

Dalam arsitektur *chatbot* secara umum, terdapat beberapa komponen, yaitu *User Interface Component* (UIC) yang merupakan antarmuka *chatbot* untuk berkomunikasi dengan pengguna, *User Message Analysis Component* (UMAC) yang berfungsi untuk melakukan *intent classification* dan *entity extraction*, *Dialog Management Component* (DMC) yang mengontrol dan memperbarui konteks konversasi, *backend* yang terdiri dari API eksternal dan/atau basis data, dan *Response Generation Component* (RGC) yang menghasilkan respon kepada pengguna.

### 3.7. Python

Python adalah bahasa pemrograman berorientasikan objek yang *powerful* dan *high-level* [15]. Python dibuat pada akhir tahun 1980 dan mulai diimplementasikan oleh Guido van Rossum pada tahun 1989 di Centrum Wiskunde & Informatica (CWI) di Belanda. Python memiliki *syntax* yang terbilang sederhana dan mudah dipahami. Terdapat banyak modul Python yang dapat diimplementasikan dalam berbagai platform, seperti *web*, *desktop*, *microcontroller*, *artificial intelligence*, dan *operating system*.

### 3.8. Flask

Flask adalah salah satu *package* yang disediakan oleh bahasa pemrograman Python untuk pembuatan web. Karena sifat fleksibelnya, Flask juga dapat digunakan sebagai *webhook* atau untuk pembuatan API yang menerima *Hypertext Transfer Protocol* (HTTP) *request*. Flask tidak membatasi arsitektur atau pola desain yang harus diterapkan untuk membangun *website*.

### 3.9. Dialogflow

Dialogflow adalah platform dari Google yang digunakan untuk membuat *chatbot agent*. Dialogflow adalah platform NLU yang mempermudah untuk mendesain dan mengintegrasikan sebuah antarmuka konversasi ke dalam aplikasi *mobile*, aplikasi web, perangkat, bot, sistem respon suara interaktif dan sebagainya [16]. Dialogflow dalam arsitektur *chatbot* mencakup komponen UMAC, DMC, dan RGC. Dialogflow

tersedia gratis untuk penggunaan personal dan memiliki metode pembayaran *pay as you go* sehingga biaya yang dikeluarkan sesuai penggunaan.

### 3.10. WhatsApp-Web.js

WhatsApp-Web.js adalah sebuah proyek *open-source* yang menyediakan antarmuka untuk mengontrol dan mengotomatisasi WhatsApp Web menggunakan bahasa pemrograman JavaScript. Ini memungkinkan pengembang untuk berinteraksi dengan layanan WhatsApp Web melalui kode JavaScript dan melakukan berbagai tindakan, seperti mengirim pesan, membaca pesan, dan mengelola sesi WhatsApp Web. WhatsApp-Web.js adalah paket Node.js yang berfungsi sebagai pustaka untuk mengontrol WhatsApp Web melalui browser yang diotomatisasi dengan menggunakan DevTools Protocol (CDP) [17].

### 3.11. Webhook

*Webhook* adalah HTTP *callback*. Konsep dasar dari *webhook* yaitu bekerja bila mendapatkan pemicu berupa *post request* atau *get request* [17]. *Webhook* bertugas untuk memberikan respons secara spesifik dari HTTP *request* yang diterima dan/atau meneruskan HTTP *request* dengan atau tanpa modifikasi ke tujuan yang ditentukan. Tujuan tersebut dapat berupa alamat *Uniform Resource Locator* (URL) yang menyediakan API *gateway* dan/atau basis data. *Webhook* dalam arsitektur *chatbot* sebagai *backend*.

### 3.12. PostgreSQL

PostgreSQL adalah sistem basis data yang bersifat sumber terbuka dan disebarluaskan secara bebas. PostgreSQL menyediakan sistem basis data *object-relational* yang menggunakan dan memperluas bahasa *Structured Query Language* (SQL) yang dikombinasikan dengan sejumlah fitur yang secara aman menyimpan dan mengatur skala beban kerja data paling rumit [19]. PostgreSQL dibuat pada tahun 1986 yang pada saat itu menjadi bagian dari proyek POSTGRES di Universitas California, Berkeley dan telah melalui lebih dari 30 tahun pengembangan secara aktif pada inti platform. Untuk mengakses basis data PostgreSQL, dapat melalui *terminal* (Linux) atau *command prompt* (Windows) atau pgAdmin (*web-based*).

### 3.13. Visual Studio Code

Visual Studio Code (VSCode) adalah editor sumber kode dari Microsoft yang ringan dan manjur [20]. VSCode dapat berjalan di desktop dan tersedia untuk sistem operasi Windows, macOS, dan Linux. VSCode sudah dilengkapi dengan dukungan bawaan untuk bahasa pemrograman JavaScript, TypeScript, dan Node.js dan memiliki ekstensi ekosistem yang melimpah untuk bahasa pemrograman yang lain, seperti C++, C#, Java, Python, PHP, dan Go. VSCode juga menyediakan kemudahan untuk menggunakan layanan Microsoft Azure dengan instal ekstensi terkait. VSCode dapat mengidentifikasi jenis bahasa pemrograman yang digunakan secara otomatis dan memberikan variasi

warna sesuai setiap fungsi dalam rangkaian kode yang diketik.

### 3.14. Mozilla Firefox

Mozilla Firefox adalah peramban yang bersifat *open source* dan didukung oleh Mozilla Foundation [21]. Mozilla Firefox dapat diakses lintas platform baik di web maupun *mobile*. Fitur yang ada pada peramban ini meliputi penggunaan tab, pencarian, *bookmark*, mengembalikan *session*, pengunduhan, *private browsing*, pemblokiran *tracking* dari pihak ketiga, *screenshot* halaman, dan beberapa fitur lainnya. Firefox menerapkan beberapa standar web yang meliputi HTML 4 dan 5, XML, XHTML, CSS, JavaScript, dan DOM. Meskipun tidak selengkap peramban Chrome, Firefox juga menyediakan instalasi *add-ons* mulai dari tema antarmuka hingga *plugin*.

### 3.15. Django

Berdasarkan Mozilla [22], Django adalah kerangka kerja (*framework*) web Python tingkat tinggi yang memungkinkan pengembangan cepat situs web yang aman dan terpelihara. Django awalnya dibuat antara tahun 2003 dan 2005 oleh tim surat kabar pengembang situs web. Tim ini memfaktorkan dan menggunakan kembali banyak pola desain dan kode umum yang mereka gunakan diimplementasikan ke situs web surat kabar, dan dikembangkan menjadi versi pertama Django, pengembangan web *framework*. Beberapa kelebihan Django adalah *complete*, *versatile*, *secure*, *scalable*, *maintainable*, dan *portable*.

*Complete* berarti Django memiliki semua yang mungkin diperlukan oleh pengembang *built-in* dan ini berarti dibundel fungsionalitas bekerja dengan mudah bersama, mengikuti prinsip desain yang sama, sementara juga memiliki detail dan pembaruan dokumentasi. *Versatile* berarti Django dapat digunakan untuk membuat berbagai jenis situs web seperti wiki, jejaring sosial, dan masih banyak lagi.

Django juga kompatibel dengan *client-side framework* apapun dan juga memiliki kemampuan untuk mengirimkan konten beberapa jenis format. *Secure* berarti Django dibuat untuk memastikan pengembang terhindar dari kesalahan keamanan dengan membuat Django berfungsi dengan menggunakan praktik keamanan yang baik sekaligus melindungi aplikasi dari ancaman eksternal seperti *SQL injection*, *cross-site scripting*, dan sejenisnya. *Scalable* berarti Django menerapkan arsitektur berbasis komponen yang berarti bahwa setiap modul aplikasi dapat ditingkatkan atau diganti mandiri.

*Maintainable* berarti bahwa Django ditulis dengan cara yang memotivasi pembuat kode untuk membuat kode yang dapat dipelihara dan digunakan kembali. Ini dilakukan dengan mengimplementasikan (*Don't Repeat Yourself*) DRY yang mengurangi jumlah kode yang tidak perlu. Django juga dibangun dengan cara di mana fungsi-fungsi terkait dikelompokkan bersama. *Portable* berarti Django tidak terbatas pada satu jenis server platform atau penyedia *hosting* web mengingat Django dibuat menggunakan bahasa pemrograman Python.