

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Keaslian Penelitian**

Berikut ini adalah merupakan penelitian-penelitian terkait yang pernah dilakukan sebelumnya :

Penelitian yang dilakukan oleh **Dhebys Suryani dan Eka LA** ,2017 dengan judul “Aplikasi Chatbot Objek Wisata Jawa Timur Berbasis AIML”, dari penelitian ini dihasilkan kesimpulan bahwa Sistem informasi rancang bangun aplikasi chatbot objek wisata Jawa Timur berbasis AIML dapat memberikan informasi kepada wisatawan yang ingin berwisata di kota Jawa Timur, Sistem informasi rancang bangun aplikasi chatbot objek wisata Jawa Timur berbasis AIML dapat membantu. Hasil penelitian ini berbeda dengan penelitian yang sedang dilakukan, pada penelitian sebelumnya aplikasi yang dihasilkan adalah sebuah aplikasi berbasis web, sedangkan pada penelitian yang sedang dilakukan akan menghasilkan aplikasi berbasis android yang dapat dipergunakan secara offline pada perangkat android.

Penelitian selanjutnya dilakukan oleh Wahyu Adie Saputro, dkk, 2020, mengangkat judul “Tinjauan Literatur Sistematis *Chatbot* Untuk Pelayanan Pelanggan” . Penelitian ini menemukan bahwa

penggunaan metode klasifikasi dalam perancangan chatbot terdapat beberapa opsi seperti cosine similarity, AIML, SVC, NER dan memiliki tingkat akurasi berbeda. Sepuluh metode klasifikasi di gunakan dalam penelitian ini dan

dari sepuluh penelitian terdapat tiga penelitian yang menggunakan metode klasifikasi cosine similarity. Performa tertinggi diperoleh pada metode cosine similarity, namun pada penelitian yang lain dengan fitur yang berbeda tidak diperoleh performa maksimal. Hal ini membuktikan bahwa metode yang digunakan tidak menjamin bahwa kinerja chatbot akan maksimal. Performa chatbot sangat dipengaruhi oleh fitur yang dipilih dan data yang digunakan. Pada penelitian ini tidak menghasilkan aplikasi dalam bentuk apapun, sedangkan pada penelitian yang akan dilakukan akan dihasilkan sebuah aplikasi android yang mengangkat tema mengenai informasi covid-19.

Penelitian selanjutnya dilakukan oleh Eka Larasati Amalia dan Dimas Wahyu Wibowo, 2019, dengan judul “Rancang Bangun Chatbot Untuk Meningkatkan Performa Bisnis” pada penelitian ini menggunakan layanan chatbot pihak ketiga yaitu Chatfuel. Chatfuel adalah platform bot untuk membuat chatbot AI yang terintegrasi pada Facebook. Mendesain fitur chatbot perlu dilakukan saat awal akan membangun chatbot. Pada awal pembuatan Chatfuel akan ditampilkan beberapa pilihan template bot. Kemudian Chatfuel diintegrasikan dengan Facebook Messenger yang ada pada Facebook Page. Jika telah terintegrasi pengaturan AI dapat dilakukan pada fitur Set Up AI, Group dan penambahan rule. Chatfuel mempunyai beberapa fitur block yang mendukung logika program IF-THEN. Frasa percakapan antara pengguna dan bot dilakukan dengan menambah kata kunci dan botreplies yaitu respon. Respon bot dapat berupa teks atau block yang sudah dibuat sebelumnya. Proses penentuan respon tidak hanya berupa teks tetapi dapat juga berupa suara, gambar, dan video. Pada penelitian ini

menghasilkan aplikasi berbasis web, sedangkan pada penelitian yang sedang dilakukan akan menghasilkan aplikasi berbasis android dengan implementasi AIML dalam memberikan respon kepada pengguna.

## II.2. Uraian Teoritis

### II.2.1. Aplikasi

Menurut Baenil Huda dan Bayu Priyatna (2019) aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia. Selain pengertian di atas, ada banyak pengertian dari kata ‘Aplikasi’ yang dikemukakan oleh para ahli. Berikut ini beberapa definisi aplikasi menurut beberapa ahli yang cukup populer :

#### 1. Ali Zaki dan *Smitdev Community*

Menurut Ali Zaki dan *Smitdev Community*, Aplikasi merupakan komponen yang bermanfaat sebagai media untuk menjalankan pengolahan data ataupun berbagai kegiatan lainnya seperti pembuatan ataupun pengolahan dokumen dan file

#### 2. Sri Widiанти

Menurut Sri Widiанти, Aplikasi merupakan sebuah *software* (perangkat lunak) yang bertugas sebagai *front end* pada sebuah sistem yang dipakai untuk mengelolah berbagai macam data sehingga menjadi sebuah informasi yang bermanfaat untuk penggunanya dan juga sistem yang berkaitan

### 3. Harip Santoso

Menurut Harip Santoso, Aplikasi merupakan sebuah kelompok file (*class, form, report*) yang ditujukan sebagai pengeksekusi aktivitas tertentu yang saling berkaitan seperti contohnya aplikasi *payroll* dan aplikasi *fixed asset*.

### 4. Yuhelfizar

Menurut Yuhefizar, Aplikasi adalah program yang sengaja dibuat dan dikembangkan sebagai pemenuh kebutuhan penggunanya dalam menjalankan suatu pekerjaan tertentu

### 5. Hengky W. Pramana

Menurut Hengky W. Pramana, pengertian aplikasi adalah satu unit perangkat lunak yang sengaja dibuat untuk memenuhi kebutuhan akan berbagai aktivitas ataupun pekerjaan, seperti aktivitas perniagaan, periklanan, pelayanan masyarakat, game, dan berbagai aktivitas lainnya yang dilakukan oleh manusia.

## II.2.2. Chatbot

*Chatbot* merupakan sebuah sistem berbasiskan *Natural Language* (bahasa alami). *Chatbot* merupakan salah satu pengembangan sistem percakapan antara manusia dengan mesin/komputer. Percakapan yang terjadi terbatas atas knowledge base chatbot itu sendiri. Grammar (aturan berbahasa yang benar) dan struktur bahasa tersebut menjadi sebuah batasan untuk menjadi sebuah kata kunci dalam filtrasi untuk merespon inputan oleh pengguna. Berdasarkan studi dari Wallace (2010) didalam jurnal R, Bestalaga (2019) *Chatbot* adalah karakter

bahasa alami yang berkomunikasi dengan penggunanya, atau orang-orang yang sedang chatting di internet bahkan melalui komunikasi suara seperti telepon. Setiap chatbot diatur oleh seorang botmaster, yaitu orang dibelakang layar yang berandil besar dalam membentuk suatu kepribadian *bot* serta pengetahuan *chatbot*. (R.Bestralaga, 2019).

### **II.2.3. Informasi**

Menurut Hamdi Agustin (2018) didalam jurnalnya mengatakan bahwa Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang. Sedangkan Mc Leod (2001) mengatakan bahwa “Informasi adalah data yang telah diproses, atau data yang memiliki arti. Informasi juga merupakan salah satu sumber data yang tersedia bagi menejer dan dapat dikelola seperti halnya sumber daya yang lain”. Berdasarkan berbagai definisi tersebut disimpulkan bahwa informasi adalah data yang diolah dan berguna bagi pemakainya dalam pengambilan keputusan. Informasi yang baik adalah informasi yang memberikan nilai tambah (*value added*) bagi pemakainya. Pemakai akan menggunakan informasi untuk perencanaan, koordinasi, evaluasi dan pengambilan keputusan. Oleh karena itu informasi harus mempunyai ciri-ciri, yaitu dapat mengurangi ketidakpastian, dapat menggambarkan adanya berbagai peluang dan dapat mengevaluasi hasil.

#### II.2.4. Covid -19

*Coronavirus Disease 2019* (COVID-19) adalah penyakit jenis baru yang belum pernah diidentifikasi sebelumnya pada manusia. Virus penyebab COVID-19 ini dinamakan Sars-CoV-2. Virus corona adalah *zoonosis* (ditularkan antara hewan dan manusia).

Adapun, hewan yang menjadi sumber penularan COVID-19 ini masih belum diketahui. Berdasarkan bukti ilmiah, COVID-19 dapat menular dari manusia ke manusia melalui percikan batuk/bersin (*droplet*), Orang yang paling berisiko tertular penyakit ini adalah orang yang kontak erat dengan pasien COVID-19 termasuk yang merawat pasien COVID-19. Tanda dan gejala umum infeksi covid-19 termasuk gejala gangguan pernapasan akut seperti demam, batuk, dan sesak napas. Masa inkubasi rata-rata adalah 5 - 6 hari dengan masa inkubasi demam, batuk, dan sesak napas. Pada kasus yang parah, covid-19 dapat menyebabkan *pneumonia*, sindrom pernapasan akut, gagal ginjal, dan bahkan kematian. (Ririn Noviyanti Putri, 2020)

### II.3. AIML (Artificial Integence Markup Language)

AIML merupakan subset dari *Extensible Markup Language* (XML) namun dengan fungsi yang mendetail. Salah satu fungsinya adalah membuat sistem input pertanyaan-balasan berbasis pengetahuan, pada AIML tag dapat diartikan berbeda sesuai dengan fungsi masing-masing.

#### II.3.1. Bagian-Bagian AIML

##### 1. *Category*.

*Category* adalah inti dari pengetahuan, disetiap *category* wajib terdapat *pattern* dan *template*. Dibawah ini adalah gambaran penggunaan *category*.

```
<category>
  <pattern>siapa orang tuamu </pattern>
  <template>kamu tidak perlu tahu orang
  tuaku</template>
</category>
```

**Gambar II. 1 Penggunaan *Category***

Pada saat *category* tersebut dipanggil maka bot akan membalas inputan user "siapa orang tuamu" dengan "kamu tidak perlu tahu orang tuaku" (R.Bestralaga, et al, 2019)

## 2. *Pattern*.

Pattern merupakan sebuah pertanyaan/inputan, *pattern* dapat direspon dengan satu atau lebih Balasan. Suatu pattern yang dianggap sama disebut *wildcard*, dibawah ini adalah cara penggunaan *wildcard*:

```
Siapa nama *
```

**Gambar II. 2 Penggunaan *Pattern***

Sepadan dengan inputan "siapa nama kamu", "siapa nama dosen kamu", dan sebagainya

## 3. *Template*

Template merupakan jawaban dari satu pattern atau lebih. Variabel atau sebuah inputan yang berarti sama seperti nama *bot* dan disisipkan ke dalam kalimat. Template juga memungkinkan untuk meneruskan ke *pattern* lain dengan menggunakan elemen AIML bernama *srai*.

Elemen *srai* dapat digunakan untuk mengimplementasikan persamaan arti seperti pada contoh berikut. Dibawah adalah contoh penggunaan Template:

```
<category>
  <pattern>dadah</pattern>
  <template>Selamat tinggal/>.</template>
</category>
<category>
  <pattern>sampai jumpa</pattern>
  <template><srai>dadah</srai></template>
</category>
```

**Gambar II. 3 Penggunaan *Template***

Pada *Category* yang berisikan *Pattern* “dadah” yang berarti inputan selamat tinggal yang akan dibalas “selamat tinggal”. *Category* selanjutnya menjawab inputan “sampai jumpa” sama seperti inputan “dadah”, dengan kata lain sebuah inputan akan diteruskan ke *category* yang dianggap sama dengan di sintakskan *<srai>* yang menunjukkan bahwa dia satu *category* inputan, maka dapat diartikan bahwa kedua inputan kata diatas bernilai sama. (R.Bestralaga, et al, 2019)

#### 4. *That*

*That* adalah element AIML yang merujuk pada balasan, tanggapan sebelumnya. *That* digunakan pada *category* ketika balasan/tanggapan masih bersangkutan dengan inputan atau balasan

sebelumnya. (R.Bestralaga, et al, 2019)

Dalam perancangan template Wallace (2003) didalam jurnal R.Bestralaga, et al, (2019) berpendapat bahwa pattern diartikan sebagai inputan bisa berarti sebuah pertanyaan, template diartikan sebagai sebuah balasan, template sendiri bisa berarti sebuah jawaban tunggal maupun jawaban random.

#### **II.4. Android**

Android adalah platform open source yang komprehensif dan dirancang untuk mobile devices. Dikatakan komprehensif karena Android menyediakan semua tools dan frameworks yang lengkap untuk pengembangan aplikasi pada suatu mobile device. Sistem Android menggunakan database untuk menyimpan informasi penting yang diperlukan agar tetap tersimpan meskipun device dimatikan. (Alamsa dan Muhammad Bunyamin, 2019).

#### **II.5. Unified Modeling Language (UML)**

*Unified Modeling Language* merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah software yang berorientasikan pada objek. UML merupakan sebuah standar penulisan atau semacam blue print dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. (Julianto Simatupang, 2019).

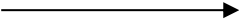
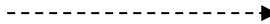
##### **A. Use Case Diagram**

*Use Case Diagram* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. *Use case* mendeskripsikan sebuah interaksi antara satu atau

lebih aktor dengan sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

**Tabel II. 2 Simbol *Use Case Diagram***

| Simbol                                                                                                          | Diskripsi                                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Use Case</i><br>            | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor: biasanya dinyatakan dengan menggunakan kata kerja diawal di awal frase nama use case.                                                                                                                                |
| Aktor/actor<br>              | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.                                                                                                                                                                              |
| Asosiasi<br>/association<br> | Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor.                                                                                                                                                                                                          |
| Ekstensi / extend<br>        | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan misal Arah panah mengarah |

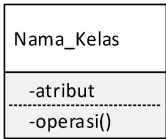
|                                                                                                                    |                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                    | pada use case yang ditambahkan.                                                                                                                                |
| Generalisasi / generalization<br> | Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum lainnya.                 |
| Menggunakan / include<br>         | Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan |

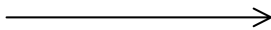
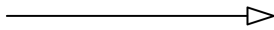
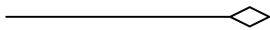
(Sumber : Julianto Simatupang, 2019)

## B. Class Diagram

*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

**Tabel II. 3 Simbol *Class Diagram***

| Simbol                                                                                                                        | Diskripsi                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| Kelas<br>                                  | Kelas pada struktur sistem.                                                  |
| Antar muka/interface<br><br>nama_interface | Sama dengan konsep interface dalam pemrograman berorientasi objek.           |
| Asosiasi / association<br>                 | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan |

|                                                                                                                              |                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                              | multiplicity.                                                                                                                        |
| Asosiasi berarah / directed association<br> | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity. |
| Generalisasi<br>                            | Relasi antar kelas dengan makna generalisasi- generalisasi- spesialisasi (umum khusus).                                              |
| Kebergantungan / dependency<br>           | Relasi antar kelas dengan makna kebergantungan antar kelas                                                                           |
| Agregasi /aggregation<br>                 | Relasi antar kelas dengan makna semua-bagian (whole- part).                                                                          |

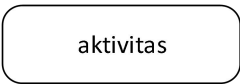
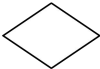
(Sumber : Julianto Simatupang, 2019)

### C. Activity Diagram

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan actor.

**Tabel II. 4 Simbol Activity Diagram**

| Simbol | Diskripsi |
|--------|-----------|
|--------|-----------|

|                                                                                                             |                                                                                            |
|-------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| Status awal<br>            | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.        |
| Aktivitas<br>              | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.             |
| Percabangan / decision<br> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu                     |
| Penggabungan / Join<br>    | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.           |
| Status akhir<br>         | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

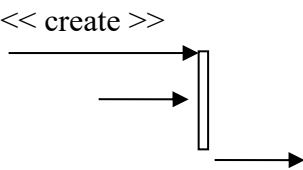
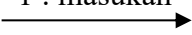
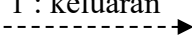
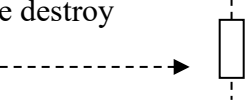
(Sumber : Julianto Simatupang, 2019)

#### **D. Sequence Diagram**

*Sequence Diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

**Tabel II. 5 Simbol Sequence Diagram**

| Simbol | Diskripsi |
|--------|-----------|
|--------|-----------|

|                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Aktor</p>  <p>atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">nama aktor</div> | <p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.</p>                                                                                                                                                         |
| <p>Garis hidup / lifeline</p>                                                                                         | <p>Menyatakan kehidupan suatu objek.</p>                                                                                                                                                                                                                                                                         |
| <p>Objek</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">nama objek</div>                                                                                               | <p>Menyatakan objek yang berinteraksi pesan.</p>                                                                                                                                                                                                                                                                 |
| <p>Waktu aktif</p>                                                                                                    | <p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya</p>                                                                                                                                                       |
| <p>Pesan tipe create</p>                                                                                            | <p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p> |
| <p>Pesan tipe send</p> <p>1 : masukan</p>                                                                           | <p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>                                                                                                                                                                            |
| <p>Pesan tipe return</p> <p>1 : keluaran</p>                                                                        | <p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>                                                                                                                     |
| <p>Pesan tipe destroy</p>                                                                                           | <p>Menyatakan suatu objek mengakhiri hidup objek lain, arah panah yang mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.</p>                                                                                                                                                        |

(Sumber : Julianto Simatupang, 2019)

