



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1 *Chatbot*

*Chatbot* adalah sebuah program komputer yang dirancang untuk memberikan respon dari sebuah percakapan dengan satu atau lebih manusia. Secara harfiah *chatbot* berasal dari dua kata yaitu *chat* dan *bot*. *Chat* dapat diartikan sebagai sebuah kegiatan komunikasi yang didukung dengan media dan sarana berupa tulisan sedangkan *bot* merupakan program yang dapat memberikan respon dari *input* menjadi *output* berupa jawaban (Suryani & Amalia, 2017).

Tujuan dari pembuatan *chatbot* ini adalah *chatbot* tersebut dapat membuat *pengguna* memperoleh informasi.

#### 2.2 *Artificial Intelligence Markup Language (AIML)*

*Artificial Intelligence Markup Language (AIML)* adalah sebuah bahasa yang merupakan turunan dari *Extensible Markup Language (XML)* yang digunakan untuk mendeskripsikan mengenai objek data dan perilaku program komputer yang memprosesnya.

AIML digunakan oleh *chatbot* untuk penelusuran jawaban setiap kalimat yang berisi kumpulan pola dan respon. *Interpreter AIML* diperlukan untuk menerima *input* dan melakukan penelusuran jawaban pada dokumen AIML. Saat ini tersedia banyak interpreter AIML dalam berbagai bahasa pemrograman sehingga proses pembuatan *chatbot* dapat terfokus pada penyusunan dokumen AIML. Bagian-bagian penting dari AIML adalah sebagai berikut :

### 1. *Category*

Pada AIML, *category* merupakan unit dasar dari pengetahuan. *Category* terdiri dari dua elemen AIML yaitu *pattern* dan *template*. Berikut adalah contoh *category* yang sederhana:

```
<category>
  <pattern>siapa nama kamu</pattern>
  <template>Nama saya BOT</template>
</category>
```

Ketika *category* di atas dimuat di *memory* maka *bot* AIML akan menjawab pertanyaan "Siapa nama kamu" dengan "Nama saya *BOT*".

### 2. *Pattern*

*Pattern* adalah serangkaian huruf yang diharapkan sesuai dengan satu atau bahkan lebih masukan (*input*) pengguna.

### 3. *Template*

Suatu *template* menentukan respon dari *pattern* yang sesuai.

```
<category>
  <pattern>siapa nama kamu</pattern>
  <template>Nama saya <bot name="name"/> </template >
</category>
<category>
  <pattern>kamu dipanggil apa</pattern>
  <template>
    <srai>siapa nama kamu</srai>
  </template>
</category>
```

*Category* pertama akan menjawab sebuah masukan "siapa nama kamu" dengan sebuah pernyataan mengenai nama bot. *Category* kedua akan

menjawab masukan “kamu dipanggil apa” dengan meneruskan pertanyaan tersebut ke *category* pertama yang cocok dengan masukan “siapa nama kamu” dengan kata lain bahwa dua *frase* tersebut sebanding atau sama.

Dalam perancangan *template*, dilakukan pembuatan daftar <pattern> sebagai perkiraan masukan dari pengguna. *Pattern* tersebut terdiri dari berbagai bentuk untuk suatu respon <template>. *Template* juga terdiri dari berbagai bentuk sebagai jawaban masukan dari pengguna agar tidak terlihat kaku dengan merespon jawaban yang sama (Suryani & Amalia, 2017).

### 2.3 *Line*

*Messenger LINE* merupakan salah satu aplikasi pesan instan (*instant messaging*) untuk *smartphone* dan *PC*. *Line* diluncurkan pada tanggal 23 Juni 2011 oleh NHN Jepang setelah gempa Tohoku. Mereka memutuskan untuk merancang sebuah aplikasi yang dapat diakses oleh *smartphone*, *tablet* dan *desktop* yang bekerja pada jaringan internet dan akan layanan ini secara gratis dapat digunakan, selain untuk melakukan kirim pesan teks, *line* juga bisa mengirim gambar, video, pesan suara dan melakukan panggilan suara secara gratis (Fitrianur, 2016).

Sejak peluncuran pertamanya, pengguna *Line* di seluruh dunia mencapai 400 juta pengguna. Majalah *SWA online* menyebutkan bahwa pengguna *Line* kini digunakan lebih dari 170 juta pengguna aktif setiap bulannya. Jumlah pengguna aktif di Indonesia sebanyak 30 juta pengguna menempatkan negara Indonesia di posisi kedua pengguna *Line* terbanyak di dunia. Berkaitan dengan populernya aplikasi *instant messaging* tentu didukung dengan adanya fitur-fitur yang menarik

dan membantu proses komunikasi interpersonal lebih efektif. Fitur-fitur *instant messaging Line* yang sering digunakan adalah :

1. *Personal Chat*

Fitur ini merupakan fitur utama yang diberikan oleh *Line* sebagai sarana komunikasi dengan pengguna *Line* lainnya secara *private*. Dalam *personal chat* ini pengguna *Line* dapat melakukan percakapan secara bebas mengenai apapun.

2. Berbagi Foto, Video, Dokumen

*Line* memberikan fitur berbagai foto, video, dokumen baik secara *personal* melalui *personal chat*, ataupun melalui diskusi grup. Pada fitur ini pengguna diberikan pilihan untuk mengambil foto atau video secara langsung dengan kamera ataupun mengambil dari galeri.

3. *Freecall* dan *Videocall*

Dengan *freecall* dan *videocall* pengguna *Line* dapat menghubungi pengguna lain dengan gratis karena menggunakan jaringan internet. Dengan cara memilih teman yang ingin dihubungi.

4. *Sticker*, *Theme* serta *emoticon*

*Sticker*, *Theme* serta *emoticon* dapat digunakan untuk mengekspresikan sesuatu dengan bentuk dan gambar yang lebih besar, lebih lucu, dan lebih menarik.

5. *Timeline Line*

Menyediakan fitur *timeline* yang bisa digunakan untuk bersosial media seperti *timeline* di *facebook*.

#### 6. *Group Line*

Menyediakan fitur grup agar pengguna dapat berkomunikasi dengan pengguna *Line* lebih dari satu pengguna (Irianto, Yusanto, & Satria Putri, 2015).

### 2.4 *Hypertext Preprocessor (PHP)*

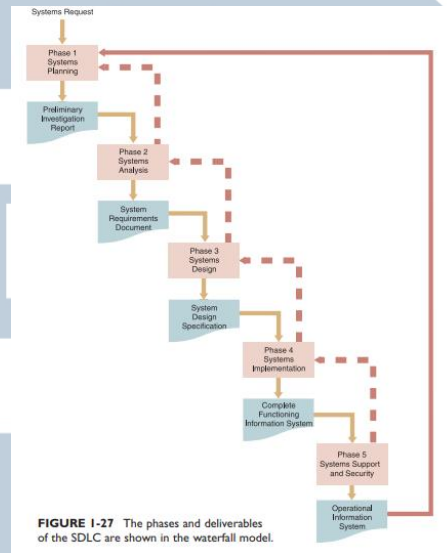
Singkatan dari PHP adalah *Hypertext Preprocessor*. PHP adalah salah satu bahasa pemrograman yang sangat cocok atau dikhususkan untuk pengembangan *web*. Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang isinya berupa sekumpulan *script* yang digunakan untuk mengolah data formulir dari *web* lalu Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilis kode ini maka banyak programmer yang tertarik untuk ikut mengembangkan PHP (Anisya, 2013).

Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti *C*, *Java*, dan *Perl* serta mudah untuk dipelajari, pemrosesan datanya dilakukan pada sisi *server* kemudian hasilnya akan dikirim kepada *client* (Firman, Wowor, & Najoan, 2016).

Contoh program Hello world yang ditulis menggunakan PHP adalah sebagai berikut:

```
<?php  
echo "Hello world";  
?>
```

## 2.5 Waterfall



**Gambar 2.1 Waterfall Model**

**Sumber:** (Shelly & Rosenblatt, 2012)

Tahapan–tahapan pada metode *Waterfall* adalah sebagai berikut :

### 1. *Systems Planning*

Tahap perencanaan biasanya dimulai dengan permintaan sistem yang menggambarkan masalah atau perubahan yang diinginkan dalam sistem informasi atau proses bisnis. Bagian terpenting adalah studi kelayakan yang meninjau biaya dan manfaat yang diantisipasi dan merekomendasikan serangkaian tindakan berdasarkan faktor operasional, teknis, ekonomi, dan waktu.

### 2. *Systems Analysis*

Tujuan dari tahap analisis sistem adalah membangun logika model sistem. Langkah pertama adalah pemodelan persyaratan dengan menyelidiki proses bisnis dan mendokumentasikan apa yang dibutuhkan sistem untuk



memuaskan pengguna. Melakukan pencarian fakta menggunakan teknik wawancara, survei, tinjauan dokumen, observasi, dan pengambilan sampel.

Hasil dari tahap analisis sistem adalah dokumen persyaratan sistem. Dokumen persyaratan sistem menggambarkan manajemen dan persyaratan pengguna, biaya dan manfaat, serta menguraikan strategi pengembangan alternatif.

### 3. *Systems Design*

Tujuan dari fase desain sistem adalah untuk membuat model fisik yang akan memenuhi semua persyaratan yang terdokumentasi untuk sistem. Pada tahap ini dilakukan perancangan antarmuka pengguna dan mengidentifikasi output, input, dan proses yang diperlukan. Selama fase ini juga menentukan arsitektur aplikasi yang akan digunakan programmer untuk mengubah desain ke dalam modul dan kode program. Hasil dari fase ini adalah spesifikasi desain sistem, yang disajikan kepada manajemen dan pengguna untuk diperiksa dan disetujui.

### 4. *Systems Implementation*

Selama fase implementasi sistem akan dibangun. Tujuan dari tahap implementasi adalah untuk memberikan informasi mengenai fungsi dan dokumentasi sistem. Pada akhir fase ini, sistem siap digunakan. Persiapan akhir termasuk mengonversi data ke file sistem baru, melatih pengguna, dan melakukan transisi yang sebenarnya ke sistem baru. Fase implementasi sistem juga mencakup penilaian, yang disebut sistem evaluasi, untuk



menentukan apakah sistem beroperasi dengan benar dan biaya serta manfaat sesuai kesepakatan awal.

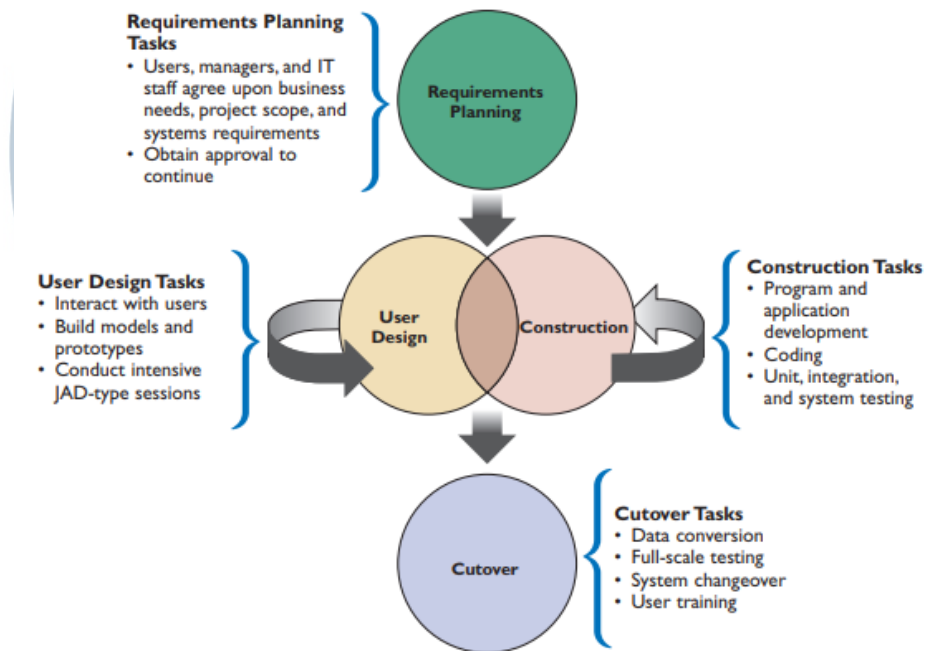
#### 5. *Systems Support and Security*

Selama fase dukungan sistem dan keamanan, staf TI memelihara, meningkatkan, dan melindungi sistem. Tujuan selama fase ini adalah kontrol keamanan dengan menjaga sistem dari ancaman eksternal dan juga internal. Sistem yang dirancang dengan baik harus aman, andal, dapat dipelihara, dan terukur.



## 2.6 Rapid Application Development (RAD)

Perangkat lunak Rapid Application Development yang tahapan-tahapan nya digambar dengan bagan di bawah ini:



**Gambar 2.2 Rapid Application Model**

**Sumber:** (Shelly & Rosenblatt, 2012)

Tahapan-tahapan yang dilakukan dapat dijelaskan sebagai berikut :

### 1. *Requirement Planning Phase*

Fase perencanaan kebutuhan menggabungkan perencanaan dan fase analisis sistem SDLC. Pengguna, pengelola, dan TI anggota staf mendiskusikan dan menyetujui kebutuhan bisnis, ruang lingkup proyek, batasan, dan sistem Persyaratan. Fase perencanaan kebutuhan berakhir ketika tim setuju pada masalah utama dan mendapatkan otorisasi manajemen untuk melanjutkan ke fase berikutnya.

## 2. *User Design Phase*

Selama fase desain, pengguna berinteraksi dengan analis sistem dan mengembangkan model yang mewakili semua proses sistem, output, dan input.

Desain pengguna adalah proses interaktif dan berkelanjutan yang memungkinkan pengguna untuk memahami, memodifikasi, dan akhirnya menyetujui model kerja dari sistem yang memenuhi kebutuhan mereka.

## 3. *Construction Phase*

Fase konstruksi berfokus pada pengembangan program. Namun dalam RAD, pengguna terus berpartisipasi dan masih dapat menyarankan perubahan atau peningkatan sistem.

## 4. *Cutover Phase*

Fase *cutover* adalah fase akhir dalam implementasi SDLC yaitu konversi data, pengujian, peralihan ke sistem baru, dan pelatihan pengguna. Hasil dari, sistem baru dibangun, dikirimkan kepada pengguna.

## 2.7 *System Design*

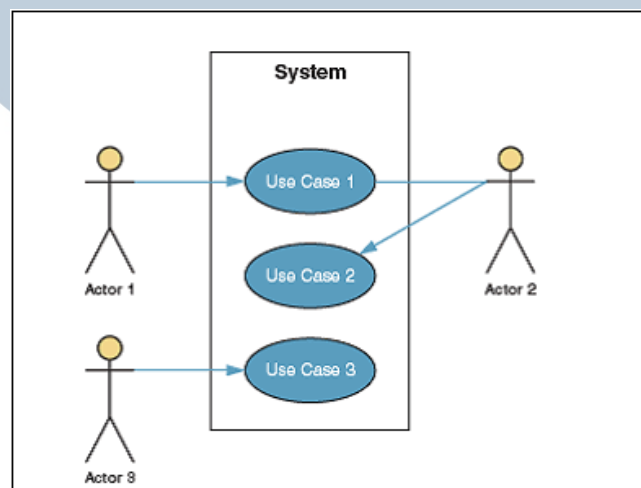
### 2.7.1 *Use Case Diagram*

Sebuah use case diagram menggambarkan secara sederhana fungsi-fungsi utama dari sistem dan berbagai user yang akan berinteraksi dengan sistem tersebut.

Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem ini berinteraksi dengan dunia luar, misalnya menyusun sebuah daftar layanan kesehatan. Use case diagram dapat digunakan untuk

memperoleh kebutuhan sistem dan memahami bagaimana sistem seharusnya bekerja. Komponen yang terdapat pada sebuah use case diagram terdiri dari:

- a. Actor : pengguna perangkat lunak aplikasi, bisa berupa manusia, perangkat keras atau sistem informasi yang lain. Actor dapat memasukan informasi ke dalam sistem, menerima informasi dari sistem, atau keduanya.
- b. Use case : perilaku atau apa yang dikerjakan pengguna sistem aplikasi, termasuk interaksi antar actor dengan perangkat lunak aplikasi tersebut (Syafitri, 2016).



**Gambar 2.3 Use-Case Diagram**

**Sumber:** (Whitten & Bentley, 2007)

### 2.7.2 Class Diagram

(Sukanto & Shalahuddin, 2013), “Diagram kelas atau 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. Atribut merupakan variabel-variabel yang

dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:






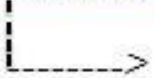
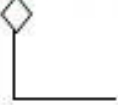
1. Kelas main  
Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem (view)  
Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
3. Kelas yang diambil dari pendefinisian use case (controller)  
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian use case, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.
4. Kelas yang diambil dari pendefinisian data (model)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas menurut (Sukanto & Shalahuddin, 2013):

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

Tabel 2.1 Class Diagram

No.	Simbol	Deskripsi
1.	<p><b>Kelas</b></p> 	Kelas pada struktur sistem
2.	<p><b>Antarmuka/interface</b></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	<p><b>Asosiasi/association</b></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	<p><b>Asosiasi berarah/ directed association</b></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	<p><b>Generalisasi</b></p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
6.	<p><b>Kebergantungan/dependensi</b></p> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7.	<p><b>Agrgasi/aggregation</b></p> 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )

Sumber: (Sukamto & Shalahuddin, 2013)


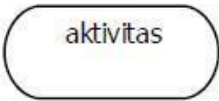



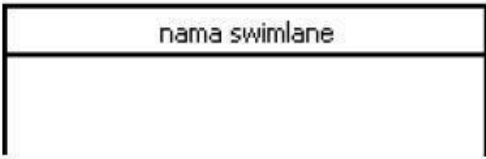
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



### 2.7.3 Activity Diagram

(Sukamto & Shalahuddin, 2013), “Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Berikut adalah simbol-simbol yang ada pada diagram aktivitas menurut (Sukamto & Shalahuddin, 2013):

**Tabel 2.2 Activity Diagram**

Simbol	Deskripsi
status awal 	status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
aktivitas 	aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
percabangan / decision 	asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
penggabungan / join 	asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
status akhir 	status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
swimlane 	memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

**Sumber:** (Sukamto & Shalahuddin, 2013)



## 2.8 MySQL

*MySQL* adalah RDBMS *open source* dan *multithreaded* yang dibuat oleh Michael “Monty” Widenius pada 1995. Pada tahun 2000, *MySQL* dirilis dengan lisensi ganda yang mengizinkan publik untuk menggunakannya secara gratis di bawah lisensi GNU GPL (*General Public License*) yang menyebabkan popularitasnya melambung. Perusahaan yang memiliki dan mengembangkan *MySQL* adalah *MySQL AB* (AB = *aktiebolag*, istilah Swedia untuk perusahaan saham), yang sekarang menjadi anak perusahaan dari *Sun Microsystems*. Saat ini *MySQL AB* memperkirakan ada lebih dari 6 juta instalasi *MySQL* di seluruh dunia, dan melaporkan rata-rata jumlah unduhan instalasi *MySQL* dari situsnya dan situs mirror sebanyak 50.000 per hari. Keberhasilan *MySQL* sebagai basis data terkemuka adalah tidak hanya karena harganya, tetapi juga karena kehandalan, kinerja dan fitur-fiturnya.

Banyaknya fitur *MySQL* membuat database ini tetap menjadi sistem basis data yang hebat. Kecepatan adalah salah satu fiturnya yang menonjol. Dalam perbandingan oleh *eWeek* pada beberapa basis data (*MySQL*, *Oracle*, *MS SQL*, *IBM DB2*, dan *Sybase ASE*), *MySQL* dan *Oracle* menunjukkan performa dan skalabilitas terbaik. *MySQL* mampu menangani puluhan ribu tabel dan miliaran baris data dengan cepat dan lancar.

Mesin penyimpanan, yang menangani *query* dan menghubungkan pernyataan SQL pengguna dengan penyimpanan, adalah perangkat lunak yang sangat penting dalam semua DBMS. *MySQL* menawarkan beberapa mesin penyimpanan dengan keunggulan yang berbeda. Beberapa diantaranya adalah

mesin penyimpanan *transaction-safe* yang memperbolehkan pengembalian data ke keadaan sebelumnya (*rollback*). Selain itu, *MySQL* mempunyai banyak sekali fungsi-fungsi di dalamnya. *MySQL* juga sangat terkenal karena kecepatan dan peningkatan kestabilan (Silalahi & Wahyudi, 2018).

## **2.9 Black-box testing**

*Black-Box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program (Pressman, 2010). Adapun teknik pengujian *Black-box* yang di gunakan dalam penelitian sebagai berikut:

1. *Acceptance testing* adalah pengujian terakhir sebelum sistem dipakai oleh user yang melibatkan pengujian dengan data dari pengguna sistem.
2. *User acceptance test* adalah uji terima perangkat lunak yang dilakukan ditempat pengguna oleh user perangkat lunak tersebut. Tujuan pengujian ini adalah untuk menguji apakah sistem sudah sesuai dengan apa yang tertuang dalam spesifikasi fungsional sistem (*validation*).

Skala *likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena social. Dengan skala likert ,maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator-indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrument yang dapat berupa pernyataan atau pernyataan (Sugiyono, 2013).

## 2.10 Penelitian Sebelumnya

Berikut merupakan beberapa penelitian yang berkaitan dengan *chatbot* yang sudah pernah dilakukan sebelumnya:

**Tabel 2.3 Penelitian Sebelumnya**

No	Penulis	Judul	Metodologi	Kesimpulan
1	Dhebys Suryani, Eka Larasati Amalia (2017)	Aplikasi <i>chatbot</i> objek wisata jawa timur berbasis AIML	<i>Artificial Intellegence Markup Language</i> (AIML)	<p>a. Sistem informasi rancang bangun aplikasi <i>chatbot</i> objek wisata Jawa Timur berbasis AIML dapat memberikan informasi kepada wisatawan yang ingin berwisata di kota Jawa Timur.</p> <p>b. Sistem informasi rancang bangun aplikasi <i>chatbot</i> objek wisata Jawa Timur berbasis AIML dapat membantu wisatawan untuk mengetahui informasi wisata, lokasi wisata, harga tiket wisata, rute wisata, fasilitas wisata.</p>
2	Haris Suryamen, Habibi Hasibuan (2017)	Pembangunan Sistem Informasi Komoditi Berbasis <i>Web GIS</i> untuk Pertanian Perkebunan dan Kehutanan Daerah Tanjung Raya Maninjau.	<i>Rapid Application Development</i> (RAD)	<p>Aplikasi Sistem Informasi Komoditi Pertanian, Perkebunan Dan Kehutanan Kecamatan Tanjung Raya Kabupaten Agam Berbasis <i>Web GIS</i> yang telah dibangun dengan menggunakan metode <i>Rapid Application Development</i> (RAD), diperoleh kesimpulan bahwa analisis kebutuhan fungsional dilakukan dengan cara menganalisis penelitian</p>

No	Penulis	Judul	Metodologi	Kesimpulan
				dan aplikasi terkait. Kemudian ditambahkan dari analisis data yang diperoleh dan permintaan BP4K2P Tanjung Raya.

Dalam membuat aplikasi *chatbot* untuk objek wisata Jawa Timur dengan menggunakan metode AIML yang dapat memberikan informasi kepada wisatawan yang ingin berwisata di kota Jawa Timur untuk mengetahui informasi wisata, lokasi wisata, harga tiket wisata, rute wisata, fasilitas wisata (Suryani & Amalia, 2017). Dari jurnal tersebut akan diadopsi *chatbot* dan metode penggunaan AIML untuk membantu dan menyelesaikan penelitian ini.

Sedangkan dalam membuat aplikasi *chatbot* untuk Pembangunan Sistem Informasi Komoditi Berbasis Web GIS untuk Pertanian Perkebunan dan Kehutanan Daerah Tanjung Raya Maninjau dijelaskan bahwa aplikasi Sistem Informasi Komoditi Pertanian, Perkebunan dan Kehutanan Kecamatan Tanjung Raya Kabupaten Agam Berbasis Web GIS dibangun dengan metode *Rapid Application Development* (RAD), maka diperoleh kesimpulan bahwa analisis kebutuhan fungsional dilakukan dengan cara menganalisis penelitian dan aplikasi terkait. Dari jurnal penelitian tersebut akan diadopsi metode RAD sebagai pedoman untuk metode dalam penelitian ini.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A