



### **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 program komputer yang dapat melakukan percakapan melalui media tulisan. Percakapan dapat terjadi dengan manusia atau *chatbot* yang lain. Secara harfiah *chatbot* berasal dari dua kata yaitu *chat* dan *bot*. Dalam dunia komputer *chat* dapat diartikan sebagai kegiatan komunikasi yang menggunakan sarana tulisan. Sedangkan *bot* merupakan program yang memiliki sejumlah data yang bila diberi input akan menghasilkan output sebagai jawaban (Suryani & Amalia, 2017).

#### 2.2 **Pencocokan Pola**

Pencocokan pola (*pattern-matching*) adalah kegiatan pemeriksaan serangkaian token yang diberikan untuk menemukan beberapa pola yang konsisten dalam *string*. Pencocokan Pola merupakan salah satu metode yang digunakan untuk proses pemilihan respon. Dalam metode ini pengetahuan berupa kumpulan pola yang saling berpasangan satu – satu dengan respon. *Chatbot* akan mencari pola dalam pengetahuan yang cocok dengan input dari *user* kemudian mengambil respon yang bersesuaian. Respon itu yang nantinya dikirimkan kembali kepada *user* (Dewi & Setiaji, 2014).

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



**Gambar 2. 1 Pencocokan Pola**

Sumber: (Dewi & Setiaji, 2014)

### 2.3 Artificial Intelligent Markup Language (AIML)

AIML adalah bahasa *scripting interpreter* yang merupakan turunan dari *Extensible Markup Language* (XML) dengan fungsi yang lebih spesifik. Salah satu fungsinya adalah membuat sistem *stimulus-response* berbasis pengetahuan. Dokumen AIML terdiri dari objek-objek yang dipisahkan oleh tag-tag tertentu seperti layaknya dokumen XML atau HTML. Bagian-bagian penting dari AIML adalah sebagai berikut (Pamungkas, 2018):

#### 1. *Category*

Pada AIML, *category* merupakan unit dasar dari pengetahuan. *Category* minimal terdiri dari dua *element* 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 sebuah rangkaian huruf yang diharapkan sesuai/cocok dengan satu atau bahkan lebih dengan masukan (*input*) pengguna. Suatu *pattern* dapat menggunakan *wildcard* yang akan cocok dengan satu atau lebih masukan pengguna.

SIAPA NAMA \*

cocok dengan masukan "siapa nama kamu", "siapa nama mama kamu", "siapa nama dosen kamu", dan sebagainya. Sintaks *pattern* AIML merupakan pola yang jauh lebih sederhana dari *regular expression*.

## 3. *Template*

Suatu *template* menentukan respon dari *pattern* yang sesuai. Sebuah *template* dapat berupa sebuah teks harafiah yang sederhana.

## 4. *That*

*That* merupakan element AIML yang mengacu pada keluaran sebelumnya. *That* sering digunakan pada pembuatan *category* agar respon yang dihasilkan masih berkaitan dengan pertanyaan atau jawaban sebelumnya.

## 2.4 MySQL

MySQL adalah sebuah "SQL *client/ server relational database management system*". MySQL menggunakan bahasa SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses *server*

*database*. Beberapa keunggulan MySQL dibandingkan dengan *database* lain adalah (Setiabudi, 2002):

1. Kecepatan: MySQL cepat. Para pengembang berpendapat bahwa MySQL adalah *database* yang tercepat yang didapat.
2. Kemudahan dalam penggunaan: MySQL adalah *simple database* sistem dengan performa tinggi dan tidak kompleks untuk *setup*, dan *administrator* dibanding dengan sistem yang lebih besar.
3. Biaya: MySQL gratis untuk semua pengguna.
4. Mendukung bahasa *Query*: MySQL memahami SQL, juga dapat mengakses MySQL menggunakan aplikasi yang mendukung *open database connectivity* (ODBC).
5. Kemampuan : banyak *client* dapat berhubungan dengan *server* pada saat yang bersamaan. *Clients* dapat menggunakan *multiple database* secara bersamaan.

## 2.5 *Instant Messenger Line*

Line adalah aplikasi pesan instan (*instant messaging*) untuk *smartphone* dan komputer. Selain untuk melakukan kirim pesan teks, Line juga bisa mengirim gambar, video, pesan suara dan melakukan panggilan suara secara gratis. Aplikasi ini berkerja pada jaringan internet (Fitrianur, 2016). Aplikasi Line menggunakan sistem nomor telepon seluler penggunaanya sebagai basis untuk saling berhubungan. Aplikasi Line saat ini tersedia untuk telepon seluler yang memiliki sistem operasional iOS dan Android serta aplikasi dapat diunduh secara gratis di *App Store* dan *Google Play*. 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*, didukung dengan adanya fitur-fitur yang menarik dan membantu proses komunikasi interpersonal lebih efektif. Diantara fitur *instant messaging* Line yang sering digunakan adalah (Irianto, Yusanto, & Putri, 2015):

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 tentang apa saja.

2. *Share Foto atau Gambar*

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

3. *Free Call*

Dengan *Free Call* pengguna Line dapat menelpon pengguna Line lain dengan gratis karena menggunakan jaringan internet. Dengan cara memilih teman yang ingin ditelepon lalu pilih Panggil.

4. *Stiker*

Layaknya *emoticon*, stiker juga 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 layaknya *timeline* di facebook.

#### 6. Grup

Line menyediakan fitur grup agar pengguna dapat berbincang-bincang dengan pengguna Line lebih dari satu pengguna.

### 2.6 *Hypertext Preprocessor*

*Hypertext Preprocessor* atau disingkat dengan PHP adalah bahasa *scripting* yang digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server*. PHP dapat diintegrasikan dengan HTML, JavaScript, JQuery, Ajax. Namun pada umumnya PHP lebih banyak digunakan bersamaan dengan file bertipe HTML. Dengan menggunakan PHP bisa membuat *website powerfull* yang dinamis dengan disertai manajemen *database*-nya (Wahyuningsih & Wibawa, 2017).

### 2.7 *Rapid Application Development (RAD)*

Menurut James Martin "*Rapid Application Development (RAD)* yaitu pengembangan siklus yang dirancang yang dapat memberikan pengembangan yang jauh lebih cepat dan hasil yang lebih berkualitas tinggi dari pada yang dicapai dengan siklus hidup tradisional. Metode pengembangan aplikasi perangkat lunak ini terdiri dari 4 tahapan yaitu *Requirements Planning Phase*, *User Design Phase*, *Construction Phase* dan *Cotuver Phase* (Kusmiati & Ansor, 2015).



## 2.8 *User Acceptance Test (UAT)*

*User Acceptance Test (UAT)* dilakukan terutama oleh pengguna dengan dukungan dari proyek tim. Tujuannya adalah untuk memastikan bahwa sistem sudah lengkap memenuhi kebutuhan bisnis yang mendorong sistem untuk dikembangkan dan dapat diterima oleh pengguna. Persepsi pengguna terhadap sistem baru akan sangat dipengaruhi oleh pengalaman mereka selama pengujian penerimaan karena kesan pertama terkadang sulit untuk diubah maka harus memastikan bahwa *acceptance test* dilakukan setelah pengujian sistem yang tepat dan berhasil (Dennis, Wixom, & Roth, 2012)

## 2.9 Wawancara

Menurut KBBI wawancara adalah tanya jawab dengan seseorang (pejabat dan sebagainya) yang diperlukan untuk dimintai keterangan atau pendapatnya mengenai suatu hal, untuk dimuat dalam surat kabar, disiarkan melalui radio, atau ditayangkan pada layar *televise* (Kamus Besar Bahasa Indonesia, 2015).

## 2.10 Desain Sistem

### 2.10.1 *Activity Diagram*

*Activity Diagram* menggambarkan kegiatan utama dan hubungan di antara kegiatan dalam suatu proses. Element – element *activity diagram* sebagai berikut (Dennis, Wixom, & Tegarden, 2012):



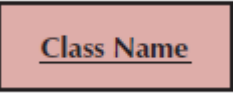





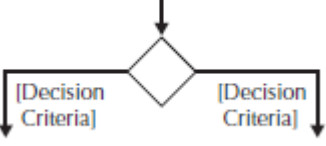
1. *Activity* dan *action* yang dilakukan untuk beberapa alasan bisnis yang spesifik digambarkan dalam diagram aktivitas sebagai *rounded rectangle* nama yang diawali dengan kata kerja dan diakhiri dengan kata benda (misalnya dapatkan informasi pasien atau lakukan pengaturan pembayaran).



2. *Object nodes* kegiatan atau tindakan biasanya memodifikasi atau mengubah objek. *Object nodes* memodelkan objek objek dalam diagram aktivitas, nama kelas objek ditulis di dalam empat persegi panjang. Pada dasarnya, *node* objek merepresentasikan aliran informasi dari satu aktivitas ke aktivitas lain.
3. *Control Flows* dan *Object Flows* memiliki dua jenis tipe aliran yang berbeda dalam suatu diagram aktivitas. *Control Flow* memodelkan jalur eksekusi melalui proses bisnis dan digambarkan dengan garis penuh anak panah yang menunjukkan arah sedangkan *object flows* memodelkan alur objek melalui proses bisnis karena kegiatan mengubah objek, *object flow* diperlukan untuk menunjukkan objek yang sebenarnya alur masuk dan keluar dari tindakan atau kegiatan.
4. *Control Note* dibagi menjadi tujuh yaitu *initial, final-activity, final-flow, decision, merge, fork*, dan *join* yang akan dijelaskan di tabel.
5. *Swimlane* diagram aktivitas dapat memodelkan proses bisnis yang independen dari implementasi objek. Namun, ada kalanya hal ini membantu untuk memecah diagram aktivitas sehingga dapat digunakan untuk menetapkan tanggung jawab untuk objek atau individu yang benar-benar akan melakukan aktivitas. Dengan pengguna *swimlane* sangat berguna ketika memodelkan alur kerja bisnis.

U M N  
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 Sintak Untuk aktivitas diagram**

<p><i>Action:</i></p> <ul style="list-style-type: none"> <li>• <i>Simple</i> dan tidak dapat diuraikan.</li> <li>• <i>Labeled by its name</i></li> </ul>	
<p><i>Activity</i></p> <ul style="list-style-type: none"> <li>• Digunakan untuk menggambarkan serangkaian tindakan</li> <li>• <i>Labeled by its name</i></li> </ul>	
<p><i>Object Node:</i></p> <ul style="list-style-type: none"> <li>• Untuk mepresentasikan suatu objek yang terhubung ke satu kumpulan <i>object flows</i></li> <li>• <i>Labeled by its name</i></li> </ul>	
<p><i>Control Flow:</i></p> <ul style="list-style-type: none"> <li>• Menunjukkan urutan pelaksanaan</li> </ul>	
<p><i>Objective Flow:</i></p> <ul style="list-style-type: none"> <li>• Menunjukkan alur suatu objek dari satu aktivitas (atau tindakan) ke aktivitas lain (atau tindakan).</li> </ul>	
<p><i>Initial Node:</i></p> <ul style="list-style-type: none"> <li>• Menggabarkan mulainya suatu <i>activity</i> atau <i>action</i>.</li> </ul>	
<p><i>Final Activity Node:</i></p> <ul style="list-style-type: none"> <li>• Digunakan untuk menghentikan semua <i>control flow</i> dan <i>object flow</i> dalam suatu aktivitas (atau tindakan).</li> </ul>	
<p><i>Final Flow Node:</i></p> <ul style="list-style-type: none"> <li>• Digunakan untuk menghentikan <i>control flow</i> atau <i>object flow</i></li> </ul>	
<p><i>Decision Node:</i></p> <ul style="list-style-type: none"> <li>• Menggabarkan kondisi pengujian untuk memastikan <i>control flow</i> atau <i>object flow</i> hanya turun ke satu jalur.</li> </ul>	

<ul style="list-style-type: none"> <li>• Diberi label dengan kriteria keputusan untuk melanjutkan menyusuri jalan spesifik.</li> </ul>	
<p><i>merge node:</i></p> <ul style="list-style-type: none"> <li>• Digunakan untuk mengebalikan jalur yang berbeda dengan membuat <i>decision node</i>.</li> </ul>	
<p><i>Fork Node:</i></p> <ul style="list-style-type: none"> <li>• Digunakan untuk membagi perilaku proses bisnis menjadi beberapa alur</li> </ul>	
<p><i>Join Node:</i></p> <ul style="list-style-type: none"> <li>• Digunakan untuk menggabungkan kembali alur aktivitas menjadi satu</li> </ul>	
<p><i>Swimlane:</i></p> <ul style="list-style-type: none"> <li>• Digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan aktivitas individu (atau tindakan) kepada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas (atau tindakan).</li> <li>• Diberi label dengan nama individu atau objek yang bertanggung jawab</li> </ul>	

Sumber: (Dennis, Wixom, & Tegarden, 2012)

### 2.10.2 Use case Diagram

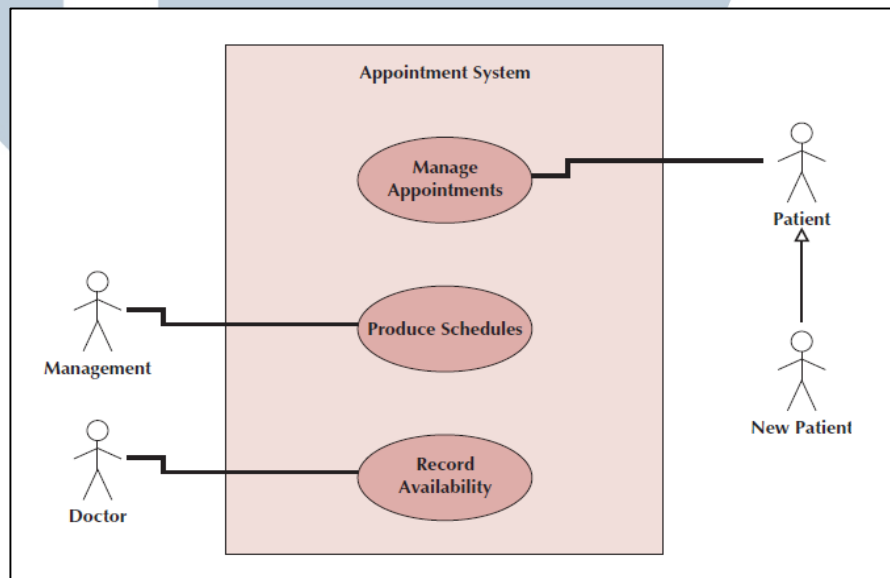
Unsur-unsur dari diagram *use case* yaitu aktor, *use case*, batas-batas subjek dan suatu hubungan antar aktor, aktor dan *use case*. Hubungan terdiri dari asosiasi, *include*, *extend*, generalisasi. Berikut ini merupakan elemen – elemen yang digunakan untuk *use case diagram* (Dennis, Wixom, & Tegarden, 2012).

### 1. Aktor

Aktor bukan pengguna tertentu, melainkan sebagai peran yang dapat dimainkan pengguna saat berinteraksi dengan sistem. aktor juga dapat mewakili sistem lain di mana sistem saat ini berinteraksi.

### 2. Asosiasi

*Use case* terhubung dengan *actor* melalui hubungan asosiasi, hubungan menunjukkan dengan mana *use case* dan aktor berinteraksi. Asosiasi biasanya mewakili komunikasi antara *use case* dengan aktor.



**Gambar 2. 2 Use Case Diagram dengan Spesial Aktor**

Sumber : (Dennis, Wixom, & Tegarden, 2012)

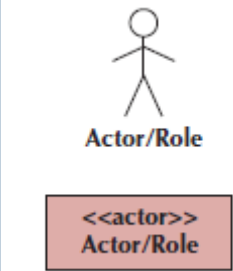
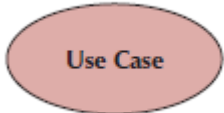
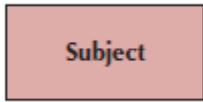

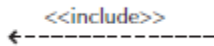
### 3. Use Case

*Use case* gambarkan dalam bentuk oval di UML adalah proses utama yang dilakukan sistem dan diberi label menggunakan frase kata kerja-kata benda deskriptif.

#### 4. Subject Boundary

*Use case* tertutup dalam batas subjek, yang merupakan kotak yang mendefinisikan ruang lingkup sistem dan secara jelas menggambarkan bagian mana dari diagram eksternal atau internal.

**Tabel 2. 2 Sintak Untuk Use Case**

<p><b>Aktor</b></p> <ul style="list-style-type: none"> <li>• Orang atau sistem yang memperoleh manfaat dan berada di luar subjek.</li> <li>• Digambarkan sebagai figur <i>stick</i> (default) atau, jika aktor bukan dilibatkan sebagai manusia, sebagai persegi panjang dengan &lt;&lt; aktor &gt;&gt; di dalamnya (alternatif).</li> <li>• Dapat dikaitkan dengan aktor lain menggunakan asosiasi spesialisasi/ superclass dilambangkan dengan panah dengan panah kosong.</li> <li>• Diberi label dengan perannya.</li> <li>• Diletakkan di luar batas subjek.</li> </ul>	
<p><b>Use Case</b></p> <ul style="list-style-type: none"> <li>• Bagian utama dari fungsi sistem</li> <li>• Dapat <i>extend</i> antara <i>use case</i></li> <li>• Dapat <i>include</i> antara <i>use case</i></li> <li>• Ditempatkan di dalam <i>system boundary</i></li> <li>• Diberi label dengan frase verba-kata benda deskriptif.</li> </ul>	
<p><b>Subject Boundary</b></p> <ul style="list-style-type: none"> <li>• Termasuk label subjek di dalam atau di atas.</li> <li>• Mewakili lingkup subjek, misalnya sistem atau proses bisnis individual.</li> </ul>	
<p><b>Hubungan Asosiasi</b></p> <ul style="list-style-type: none"> <li>• Menghubungkan seorang aktor dengan <i>use case</i> (s) yang berinteraksi dengannya.</li> </ul>	
<p><b>Hubungan Include</b></p> <ul style="list-style-type: none"> <li>• Mewakili penyertaan fungsi dari satu <i>use case</i> dalam <i>use case</i> lainnya.</li> <li>• Memiliki panah yang ditarik dari <i>use case</i> dasar ke <i>use case</i>.</li> </ul>	

<p><i>Hubungan Extend</i></p> <ul style="list-style-type: none"> <li>• Merupakan perpanjangan dari use case untuk menyertakan perilaku opsional.</li> <li>• Memiliki panah yang ditarik dari <i>use case</i> ekstensi ke kasus penggunaan dasar.</li> </ul>	
<p><i>Hubungan Generalization</i></p> <ul style="list-style-type: none"> <li>• Menggambarkan spesialis <i>use case</i> terhadap yang lebih umum.</li> <li>• Memiliki panah yang ditarik dari <i>use case</i> khusus ke <i>use case</i> dasar.</li> </ul>	

Sumber : (Dennis, Wixom, & Tegarden, 2012)

### 2.10.3 Class Diagram

*Class Diagram* adalah model statis yang menunjukkan kelas dan hubungan di antara kelas yang tetap konstan dalam sistem dari waktu ke waktu. *Class diagram* menggambarkan kelas yang meliputi perilaku (*behaviors*) dan *state* dengan hubungan antar kelas (Dennis, Wixom, & Tegarden, System analysis Design UML Versi 2.0 An - Object Oriented Approach, 2012).

**Tabel 2. 3 Sintak Untuk Class Diagram**

<p><i>Class</i></p> <ul style="list-style-type: none"> <li>• Mewakili jenis orang, tempat, atau hal tentang sistem mana yang perlu diambil dan disimpan.</li> <li>• Memiliki nama yang diketik dengan huruf tebal dan di bagian paling atas kompartemen</li> <li>• Memiliki daftar atribut di tengah kompartemen</li> <li>• Memiliki daftar operasi di bawah kompartemen</li> <li>• Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas</li> </ul>	
<p><i>Attribute</i></p> <ul style="list-style-type: none"> <li>• Merupakan properti yang menggambarkan keadaan suatu objek</li> </ul>	<p>attribute name /derived attribute name</p>



<ul style="list-style-type: none"> <li>Dapat diturunkan dari atribut lain, ditampilkan dengan menempatkan garis miring sebelum nama atribut.</li> </ul>	
<p><i>Operation</i></p> <ul style="list-style-type: none"> <li>Merupakan tindakan atau fungsi bahwa sebuah class dapat melakukan.</li> <li>Dapat diklasifikasikan sebagai konstruktor, query, atau memperbaharui operasi.</li> <li>Termasuk tanda kurung yang mungkin mengandung parameter khusus atau informasi yang dibutuhkan untuk melakukan operasi.</li> </ul>	
<p><i>Association</i></p> <ul style="list-style-type: none"> <li>Merupakan hubungan antara beberapa class atau class dirinya sendiri.</li> <li>Dilabeli menggunakan frasa kata kerja atau nama peran, mana yang lebih baik mewakili hubungan.</li> <li>Bisa Diantara satu kelas atau lebih.</li> <li>Berisi simbol multiplisitas, yang mewakili waktu minimum dan maksimum kelas Misalnya dapat dikaitkan dengan contoh kelas lain.</li> </ul>	
<p><i>Generalization</i></p> <ul style="list-style-type: none"> <li>Menunjukkan hubungan antara beberapa kelas.</li> </ul>	
<p><i>Aggregation</i></p> <ul style="list-style-type: none"> <li>Menunjukkan hubungan yang logis antara beberapa kelas dan kelas itu sendiri</li> <li>Bentuk khusus dari asosiasi</li> </ul>	
<p><i>Composition</i></p> <ul style="list-style-type: none"> <li>Menunjukkan hubungan fisik antara beberapa kelas dan kelas itu sendiri</li> <li>Bentuk khusus dari asosiasi</li> </ul>	

Sumber : (Dennis, Wixom, & Tegarden, 2012)



#### 2.10.4 Site Map

*Sitemap* adalah informasi tentang tautan-tautan yang terdapat dalam sebuah *website* yang dapat mempermudah pengunjung untuk menemukan halaman yang diinginkan. *Sitemap* dapat dibuat dengan format *flowchart* untuk memvisualisasikan halaman-halaman yang ada didalam sebuah *website* (Krishnasari, 2018).

#### 2.11 Penelitian Sebelumnya

**Tabel 2. 4 Penelitian Sebelumnya**

No	Penulis	Judul	Nama Jurnal	Kesimpulan
1	Fajrin Azwary, Fatma Indriani, Dodon Turianto Nugrahadi	<i>Question Answering System Berbasis Artificial Intelligent Markup Language Sebagai Media Informasi</i> (2016)	Kumpulan jurnal, ilmu computer (KLIK) Volume 04, No. 01 Februari 2016 ISSN:2406-7857	Penelitian ini menghasilkan sebuah <i>chatbot Qusetion-Answering System</i> dengan menggunakan AIML dengan pencocokan pola dapat berkomunikasi dengan baik.
2	Herlinda Kusmiati, Medhy Ansori	<i>Penerapan Rapid Application Development Pada Aplikasi Pencabutan Layanan Reguler Smart</i> (2015)	Vol. 5, No. 2, Juli 2015	Dengan menggunakan metode RAD, pengerjaan dalam pembuatan sistem pencabutan layanan regular smart. Metode RAD membuat batasan-batasan dalam pembuatan sistem supaya tidak keluar dari kebutuhan.

Jurnal yang berjudul “*Question Answering System Berbasis Artificial Intelligent Markup Language Sebagai Media Informasi*” dijelaskan bahwa Fajrin Azwary, Fatma Indriani dan Dodon Turianto Nugrahadi menghasilkan sebuah

*chatbot Qusetion-Answering System* dengan menggunakan AIML dengan pencocokan pola dapat berkomunikasi dengan baik maka dari jurnal yang sudah dijelaskan akan digunakan metode *Artificial Intellegent Markup Language (AIML)* dan pencocokan pola dalam menyelesaikan penelitian ini. Sedangkan pada jurnal yang berjudul “Penerapan *Rapid Application Development* Pada Aplikasi Pencabutan Layanan *Reguler Smart*” dijelaskan bahwa dengan menggunakan metode RAD, pengerjaan dalam pembuatan sistem pencabutan layanan *regular smart* metode RAD dapat membuat batasan-batasan dalam pembuatan sistem supaya tidak keluar dari kebutuhan maka dari jurnal yang sudah dijelaskan akan digunakan yaitu metode *Rapid Application Development (RAD)* yang akan digunakan untuk membantu dalam pengembangan sistem.

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA