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 Artificial Intelligence

Artificial Intelligence (AI) adalah bagian dari ilmu pengetahuan komputer yang khusus ditujukan dalam perancangan otomatisasi tingkah laku cerdas dalam sistem kecerdasan komputer (Kristanto, 2005). Definisi AI dibahas oleh beberapa pakar, diantaranya adalah:

a. Sebuah studi tentang bagaimana membuat komputer mengerjakan sesuatu yang dapat dikerjakan manusia (Rich, 1991).

b. Cabang ilmu komputer yang memusatkan perhatiannya pada otomatisasi tingkah laku cerdas (Luger dan Stubbfield, 1989).

c. Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang (dalam pandangan manusia) cerdas (Simon, 1987).

Keuntungan dari kecerdasan buatan adalah sebagai berikut (Suparman, 1991):

1. Menciptakan piranti lunak konvensional yang lebih cepat dan mudah dalam pengambilan data.
2. Komputer menjadi lebih berguna karena bisa digunakan untuk membantu dalam memecahkan berbagai masalah yang rumit.
3. Dengan menggunakan bahasa alami, para pemakai akan menjadi lebih akrab dengan komputer, dalam hal berkomunikasi dengan komputer, tanpa perlu mempelajari bahasa pemrograman tertentu.

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Sedangkan kerugian dari kecerdasan buatan, adalah sulitnya pengembangan piranti lunak kecerdasan buatan karena bagaimanapun juga program kecerdasan buatan tersebut masih tetap merupakan barang mewah yang rumit pembuatannya.

Kecerdasan buatan mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia. Aktivitas manusia yang ditirukan seperti penalaran, penglihatan, pembelajaran, pemecahan masalah, pemahaman bahasa alami dan sebagainya. Teknologi kecerdasan buatan dapat dipelajari dalam berbagai bidang bidang seperti robotika (*robotics*), penglihatan komputer (komputer vision), pengolahan bahasa alami (*natural language processing*), pengenalan pola (*pattern recognitiion*), sistem syaraf buatan (*artificial neural system*), pengenalan suara (*speech recognition*) dan sistem pakar (*expert system*). (Simarmata, 2006).

2.2 Artificial Intelligence Markup Language

Artificial Intelligence Markup Language (AIML) adalah salah satu teknologi yang umum digunakan dalam mengembangkan *chatbot*. AIML secara luas digunakan dalam pengembangan perangkat lunak yang berkomunikasi dengan penggunanya menggunakan bahasa alami. Bahasa AIML dikembangkan oleh Richard Wallace dan komunitas Open Source Alicebot antara tahun 1995 dan 2000. AIML menghasilkan file teks XML dengan struktur tertentu, yang merupakan basis pengetahuan (*knowledge base*) dari *chatbot*. "Kategori" adalah dasar dari basis pengetahuan (*knowledge base*), dan terdiri dari dua elemen: "pattern" dan "template" (Mikic dkk., 2012).

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Secara umum, kinerja AIML didasarkan pada *stimulus-response model*, dimana stimulus (input pengguna) sesuai dengan "pattern", dan respon yang ditampilkan *chatbot* kepada pengguna adalah "*template*". Semua itu untuk mencari pola yang memadai dan menunjukkan *template* yang terkait, yang dilakukan oleh mesin pengolah data, yang memiliki versi berbeda (Program D, Program E, dll) (Mikic dkk., 2012).

Bagian-bagian penting dari AIML adalah sebagai berikut (Azwari dkk., 2016).

a. 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>
```

Gambar 2.1 Contoh *category* (Azwari dkk., 2016)

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

b. 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. Suatu *pattern* seperti berikut :

```
Siapa nama *
```

Gambar 2.2 Contoh *pattern* (Azwari dkk., 2016)

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Cocok dengan masukan "siapa nama kamu", "siapa nama dosen kamu", dan sebagainya.

c. Template

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

Gambar 2.3 Contoh *template* (Azwari dkk., 2016)

Sebuah *template* juga dapat menggunakan variabel seperti.

Gambar 2.4 Contoh penggunaan variabel dalam *template* (Azwari dkk., 2016)

Variabel bernilai sama dengan nama *bot* dan disisipkan kedalam 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.

```
<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>
```

Gambar 2.5 Contoh elemen *srai* (Azwari dkk., 2016)

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

UNIVERSITAS MULTIMEDIA NUSANTARA

yang cocok dengan masukan “siapa nama kamu” dengan kata lain bahwa dua frase tersebut sebanding atau sama.

d. That

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

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. Misalnya seperti pada contoh berikut untuk menanyakan kode matakuliah Kecerdasan Buatan.

Tabel 2.1 Contoh daftar <pattern> pada *template* (Azwari dkk., 2016)

Pattern	**Template**
apa kode matakuliah kecerdasan buatan?	JFKB161
kode matakuliah kecerdasan buatan	kodenya JFKB161
kode matakuliah kecerdasan buatan apa?	kode matakuliah kecerdasan buatan adalah JFKB161
kode matkul kecerdasan buatan	kode matkul kecerdasan buatan JFKB161

Pada tabel 2.1 digunakan untuk menanyakan kode matakuliah Kecerdasan Buatan, ada beberapa kemungkinan masukan pengguna (*pattern*). Sebagai tanggapan atau respon (*template*) juga terdiri dari berbagai macam bentuk yang kemudian akan dimunculkan secara acak. Adapun proses perancangan dan pembuatan *template* AIML-nya adalah sebagai berikut.

a. Data dari hasil wawancara, ditentukan apa saja informasi yang sering ditanyakan.

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

b. Informasi tersebut kemudian di daftar berdasarkan variabel lain yang menjadi acuan pertanyaan. Pada tabel di atas, pertanyaan mengarah kepada kode mata kuliah, berdasarkan nama mata kuliah.

c. Menentukan kemungkinan bentuk pertanyaan atau input berdasarkan suatu informasi. Sebagai contoh dalam menanyakan kode mata kuliah pada tabel di atas. Bentuk pertanyaan tidak hanya terbatas pada "apa kode matakuliah kecerdasan buatan?", tapi juga bisa "kode matakuliah kecerdasan buatan apa?", dengan letak kata tanya di belakang. Selain itu bisa juga dalam bentuk tanpa mengunakan kata tanya.

d. Menyesuaikan pola pertanyaan dengan bentuk *pattern* AIML. Pada AIML terdapat simbol *wildcard* berupa "*" dan "_" dengan fungsional menerima inputan apapun dari pengguna, kecuali kosong atau tidak diisi input apapun sebagai *wildcard* tersebut.

Tag yang digunakan pada AIML dijelaskan sebagai berikut.

1. Tag <aiml>

Setiap *file* AIML dimulai dengan *tag* <aiml> dan diakhiri dengan *closing tag* </aiml>. *Tag* ini mengandung atribut *version* dan *encoding*. Atribut *version* mengindentifikasikan versi AIML yang digunakan pada *knowledge base chatbot*. Apabila atribut ini tidak didefinisikan, program tidak akan menghasilkan *error*, tetapi dapat menimbulkan masalah ketika proses *maintenance* atau *upgrade*. Atribut *encoding* digunakan untuk mendefinisikan tipe *encoding* karakter yang digunakan pada dokumen AIML ini. Pada contoh yang ditunjukkan pada Gambar 2.6, atribut *version* yang digunakan adalah 1.0.1 dan *encoding* UTF-8 (Marietto dkk., 2013).

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

```
<aiml version="1.0.1" encoding="UTF-8"?>
<category>
   <pattern> HELLO BOT </pattern>
   <template>
      Hello my new friend!
   </template>
</category>
</aiml>
```

Gambar 2.6 Penggunaan Tag <aiml> (Marietto dkk., 2013) Tag <category>

Unit dasar dari dialog AIML adalah *category*. *Category* terdiri dari tiga bagian, yaitu *user input* dalam bentuk kalimat, respon dari *chatbot* terhadap *user input*, dan konteks opsional. *Knowledge base* yang dibuat dengan AIML merupakan kumpulan dari *category*. *Category* diorganisir dalam bentuk *file* ".aiml". *Category* dimulai dengan *tag* <category> dan diakhiri *closing tag* </category>. *Tag* <category></category> harus berada di dalam *tag* <aiml></aiml> dan memiliki *tag pattern* dan *template* di dalamnya. Gambar 2.7 menunjukkan contoh penggunaan *tag category*.

```
<category>
<pattern>MOTHER</pattern>
<template> Tell me more about your family. </template>
</category>
```

Gambar 2.7 Penggunaan AIML Tag (Wallace, 2003)

2. Tag <pattern>

Tag ini berisi *input user* yang mungkin atau pola kalimat dari *input user* yang telah melalui proses dengan NLP. *Tag* <pattern> hanya didefinisikan satu kali pada setiap *tag* <category> dan harus menjadi elemen pertama yang didefinisikan. Kata-kata di dalam *tag* <pattern> dipisahkan dengan satu spasi dan penggunaan *wildcards* dapat mensubtitusi bagian dari kalimat tersebut. Pada contoh yang ditunjukkan oleh Gambar 2.6, *tag* <pattern> berisi kalimat "HELLO BOT" yang

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

merupakan kalimat *input* dari *user*. Kalimat ditulis dalam huruf kapital untuk menyederhanakan proses *matching* (Marietto dkk., 2013).

3. Tag <template>

Tag <template> berisi respon yang diberikan oleh *chatbot* untuk menjawab kalimat pada *tag* <pattern>. *Tag* ini dapat menyimpan data, mengaktifkan program lain, dan memberikan jawaban kondisional atau jawaban dari *category* lain (Marrietto dkk., 2016). Pada Gambar 2.6, respon dari *input* "HELLO BOT" adalah "Hello my new friend!".

4. Tag <star index = "n">

Tag ini menyimpan dan memetakan komponen bagian dari kalimat pada *user input*. Indeks "n" mengindikasikan komponen yang dipetakan oleh *star index*. *Tag* ini merupakan konteks opsional pada *category*. Untuk memetakan *input user* pada *tag pattern*, digunakan simbol atau *wildcard* "*" (Marietto dkk., 2013). Gambar 2.8 menunjukkan contoh penggunaan *tag* <star index = "n">. Apabila *user input* adalah kalimat "A rose is a flower", maka respon yang diberikan oleh *chatbot* adalah "When a rose is not a flower?".

```
<category>
   <pattern> I LIKE * </pattern>
   <template>
      I like <star/> too.
   </template>
</category>

<category>
   <pattern> A * IS A * </pattern>
   <template>
      When a <star index="1"/> is not a <star index="2"/>?
   </template>
</category>
```

Gambar 2.8 Penggunaan Tag <star index = "n"> (Marietto dkk., 2013)

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

5. Tag <srai>

Salah satu properti yang sangat berguna dalam AIML adalah kemampuan untuk menargetkan *pattern input* yang berbeda untuk respon pada *tag template*. Dengan demikian, *interpreter* AIML dapat mencari jawaban dari *input user* yang berbeda. Hal ini dilakukan dengan menggunakan *tag* <srai> (Marietto dkk., 2013).

Pada Gambar 2.9 terdapat dua kategori yang membicarakan Alan Turing (baris 1-8) dan Albert Sabin (baris 10-16). *Pattern* pada kategori Alan Turing dan Albert Sabin menggunakan kalimat "WHO IS ALAN TURING?" dan "WHO IS ALBERT SABIN?". Namun, ada kemungkinan *user* bertanya dengan kalimat yang berbeda, sebagai contoh "DO YOU KNOW WHO ALAN TURING IS?" dan mengharapkan respon yang sama. Kategori baru dapat dibuat untuk menangani *input* tersebut dengan *pattern* "DO YOU KNOW WHO * IS?". *Wildcard* "*" menyatakan nama orang yang ditanyakan. Respon dibuat dengan "<srai> WHO IS <star/> </srai>" yang melakukan *redirect pattern* ke kategori yang lain.

```
<category>
   <pattern> WHO IS ALAN TURING? </pattern>
   <template>
      Alan Turing was a British mathematician, cryptographer,
      and computer scientist often credited as
      the founder of modern Computer Science.
   </template>
</category>

<category>
   <pattern> WHO IS ALBERT SABIN? </pattern>
   <template>
      Albert Sabin was the researcher who developed
      the vaccine that is the main defense against polio.
   </template>
</category>

<category>
   <pattern> DO YOU KNOW WHO * IS? </pattern>
   <template>
      <srai> WHO IS <star/> </srai>
   </template>
</category>
```

Gambar 2.9 Tag <srai> Untuk Symbolic Reduction (Marietto dkk., 2013)

NUSANTARA

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

```
<category>
   <pattern> BYE </patter>
      <template> Goodbye friend! </template>
</category>

<category>
   <pattern> BYE * </pattern>
   <template> <srai> BYE </srai> </template>
</category>
```

Gambar 2.10 Tag <srai> Divide and Conquer (Marietto dkk., 2013)

Pada Gambar 2.10 ditunjukkan penggunaan *tag* <srai> untuk konsep *divide and conquer*. Sebagai contoh kalimat "BYE *" mengharapkan respon yang sama dengan kalimat "BYE", maka *tag* <srai> dapat digunakan sehingga apabila kalimat *input* "BYE FRIEND karakter apapun setelah kata "BYE" akan mendapatkan respon "Goodbye friend!".

```
category>
   <pattern> INDUSTRY </pattern>
   <template>
      It is a development center.
   </template>
</category>

<category>
   <pattern> FACTORY </pattern>
   <template>
      <srai> INDUSTRY </srai>
   </template>
</category>
```

Gambar 2.11 Tag <srai> Untuk Sinonim (Marietto dkk., 2013)

Sebagai contoh, kategori pertama pada Gambar 2.11 menerima *input* "INDUSTRY" dengan respon "It is a development center.". Kata "FACTORY" merupakan sinonim dari "INDUSTRY" sehingga apabila *user input* berupa kata "FACTORY", maka respon yang diharapkan adalah "It is a development center.". *Tag* <srai> dapat digunakan untuk memanggil respon dari "INDUSTRY" sehingga *pattern* "FACTORY" akan direspon dengan respon pada *category* "INDUSTRY".

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Tag <srai> juga dapat digunakan untuk mendeteksi kata kunci tertentu seperti yang ditunjukkan pada Gambar 2.12. Penggunaan *wildcard* "_" dan "*" menandakan kalimat terdiri dari kata dan salah satu diantaranya adalah kata "FAMILY". *Tag* <srai> digunakan untuk menargetkan respon "Family is an important institution." untuk setiap kalimat yang mengandung kata kunci "FAMILY".

```
<category>
   <pattern> FAMILY </pattern>
   <template>
      Family is an important institution.
   </template>
</category>

<category>
   <pattern> _ FAMILY </pattern>
   <template>
      <srai> FAMILY </srai>
   </template>
</category>

<category>
   <pattern> FAMILY * </pattern>
   <template>
      <srai> FAMILY </srai>
   </template>
</category>

<category>
    <pattern> _ FAMILY * </pattern>
    <template>
        <srai> FAMILY </srai>
    </template>
</category>
```

Gambar 2.12 Tag <srai> Untuk Deteksi Keyword (Marietto dkk., 2013)

6. Tag <random> dan

Tag <random> digunakan untuk memberikan respon dengan cara yang berbeda. Setiap kalimat respon yang mungkin ditulis dalam *tag* . *Interpreter* AIML membaca respon sebagai *list* dan memilih secara acak salah satu kalimat respon dalam *list* (Marietto dkk., 2013). Gambar 2.13 menunjukkan contoh penggunaan *tag* <random> dan <li>.

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

```
<category>
   <pattern> HI </pattern>
   <template>
      <random>
         <li> Hi! Nice to meet you </li>
         <li> Hello, How are you? </li>
         <li> Hello! </li>
      </random>
   </template>
</category>
```

Gambar 2.13 Tag <random> dan <li> (Marietto dkk., 2013)

7. Tag <set> dan <get>

Tag <set> dan <get> memungkinkan *chatbot* bekerja dengan menggunakan variabel. *Tag* <set> digunakan untuk menyimpan *value* ke dalam variabel. *Tag* ini harus berada di dalam *tag* <template>. Contoh penggunaan *tag* <set> ditunjukkan pada Gambar 2.14. *Value* yang direpresentasikan dengan *wildcard* "*" disimpan ke dalam variabel bernama "nameUser".

```
<category>
   <pattern> MY NAME IS * </pattern>
   <template>
      Hello <set name="nameUser"> <star/> </set>
   </template>
</category>
```

Gambar 2.14 Tag <set> (Marietto dkk., 2013)

```
<category>
   <pattern> GOOD NIGHT </pattern>
   <template>
      Good night <get name="nameUser"/>
   </template>
</category>
```

Gambar 2.15 Tag <get> (Marietto dkk., 2013)

Value yang tersimpan di dalam variabel dapat diambil dengan menggunakan *tag* <get>. Pada Gambar 2.15, variabel "nameUser" yang telah di-*set* pada Gambar 9 digunakan pada kalimat respon dengan *tag* "<get name="nameUser">". Apabila variabel "nameUser" berisi "FRIEND", maka kalimat respon dari "GOOD NIGHT" yaitu "GOOD NIGHT FRIEND".

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

8. Tag <that>

```
User: Make some question
Bot: Do you like movies?
User: No
Bot: OK. But I like movies.
```

Gambar 2.16 Skenario Percakapan (Marietto dkk., 2013)

```
<category>
   <pattern> MAKE SOME QUESTION </pattern>
   <template>
      Do you like movies?
   </template>
</category>

<category>
   <pattern> YES </pattern>
   <that> Do you like movies? </that>
   <template>
      Nice, I like movies too.
   </template>
</category>

<category>
   <pattern> NO </pattern>
   <that> Do you like movies? </that>
   <template>
      OK. But I like movies.
   </template>
</category>
```

Gambar 2.17 Penggunaan Tag <that> (Marietto dkk., 2013)

Tag <that> memungkinkan sistem untuk menganalisis kalimat respon *chatbot* sebelumnya. Percakapan dapat berjalan sesuai dengan konteks yang sedang dibicarakan. *Tag* ini harus berada di dalam *tag* <category> (Marietto dkk., 2013). Gambar 2.16 menunjukkan sebuah skenario percakapan dan Gambar 2.17 menunjukkan penggunaan *tag* <that> pada percakapan tersebut. *User* memberikan *input* "NO". Kemudian, Sistem akan melihat kalimat respon sebelumnya, yaitu "Do you like movies?" dan memberikan respon "OK. But I like movies".

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

9. Tag <topic>

Tag <topic> digunakan untuk menyatakan topik atau subjek pembicaraan *chatbot*. Topik ini mengelompokkan *tag category* sehingga mempercepat proses pencarian respon (Marietto dkk., 2013). Contoh penggunaan *tag* <topic> ditunjukkan pada Gambar 2.18. Pada pembicaraan "LET TALK ABOUT FLOWER", *chatbot* akan mengisi variabel *topic* dengan *flowers*. Pembicaraan selanjutnya dengan *input* "I LIKE IT SO MUCH" yang diterima, *chatbot* akan merespon dengan "I like flowers too" berdasarkan variabel *topic* yang telah didefinisikan.

```
<category>
   <pattern> LET TALK ABOUT FLOWERS. </pattern>
   <template>
      Yes <set name="topic">flowers</set>
   </template>
</category>

<topic name="flowers">
   <category>
      <pattern> * </pattern>
      <template>
         Flowers have a nice smell.
      </template>
   </category>

   <category>
      <pattern> I LIKE IT SO MUCH! </pattern>
      <template>
         I like flowers too.
      </template>
   </category>
</topic>
```

Gambar 2.18 Penggunaan Tag <topic> (Marietto dkk., 2013)

10. Tag <think>

```
<category>
   <pattern> MY NAME IS * </pattern>
   <template>
      <think> <set name="nameUser"> * </set> </think>
</template>
</category>
```

Gambar 2.19 Penggunaan Tag <think> (Marietto dkk., 2013)

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Tag <think> digunakan untuk memproses data, *conditional statements* dan *tests* yang tidak ditampilkan kepada *user*. Isi dari *tag* ini diproses oleh *chatbot* tanpa memperlihatkan proses tersebut kepada *user* (Marietto dkk., 2013). Pada Gambar 2.19, *chatbot* mengisi variabel "nameUser" dan *user* tidak mengetahui proses tersebut.

11. Tag <condition>

Tag <condition> digunakan ketika ada lebih dari satu respon yang dapat diberikan kepada *user* dan pemilihan respon tersebut dilakukan berdasarkan analisis *value* dari suatu variabel yang berubah seiring dengan berjalannya proses *chatting*. *Tag* <condition> dapat dikatakan ekuivalen dengan *command* "Case" pada bahasa pemrograman (Marietto dkk., 2013). Gambar 2.20 menunjukkan contoh penggunaan *tag* <condition>. Atribut "name=" berisi nama variabel yang dianalisis dan "value=" berisi *value* yang dibandingkan dengan *value* variabel tersebut.

```
<category>
   <pattern> HOW ARE YOU? </pattern>
   <template>
      <condition name="state" value="happy">
         It is nice being happy.
      </condition>
      <condition name="state" value="sad">
         Being sad is not nice.
      </condition>
   </template>
</category>
```

Gambar 2.20 Penggunaan Tag <condition> (Marietto dkk., 2013)

12. Tag <bot>

```
<category>
   <pattern> BOT'S PROPERTIES </pattern>
   <template>
      <bot name="age"/>
      <bot name="gender"/>
      <bot name="location"/>
      <bot name="nationality"/>
      <bot name="birthday"/>
      <bot name="sign"/>
      <bot name="botmaster"/>
   </template>
</category>
```

Gambar 2.21 Penggunaan Tag <bot> (Marietto dkk., 2013)

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Tag <bot> digunakan untuk mendefinisikan properti *chatbot* dan properti ini dapat dilihat oleh *user* saat percakapan berlangsung (Marietto dkk., 2013). Pada Gambar 2.21 ditunjukkan contoh penggunaan *tag* <bot>. Properti yang didefinisikan dalam contoh di atas adalah “age”, “gender”, “location”, “nationality”, “birthday”, “sign”, dan “botmaster”.

2.3 Chatbot

Chatbot adalah sebuah simulator percakapan yang berupa program komputer yang dapat berdialog dengan penggunanya dalam bahasa alami. Karena *chatbot* hanya sebuah program, dan bukan robot (*chatbot* tidak memiliki tubuh dan tidak memiliki mulut sehingga tidak dapat berbicara seperti manusia), maka yang dimaksud dengan dialog antar manusia sebagai pengguna dengan *chatbot* dilakukan dengan cara mengetik apa yang akan dibicarakan dan *chatbot* akan memberikan respon. Orang yang membuat dan mengembangkan program *chatbot* disebut *bot master* (Rudiyanto, 2005).

Chat dapat diartikan sebagai pembicaraan. *Bot* merupakan sebuah program yang mengandung sejumlah data, jika diberikan masukan maka akan memberikan jawaban. *Chatbot* dapat menjawab pertanyaan dengan membaca tulisan yang diketikkan oleh pengguna melalui keyboard. (Adriyani dan Mahdiyah, 2004). Meskipun banyak *bots* yang dapat menginterpretasikan dan menanggapi input manusia, sebenarnya *bots* tersebut hanya mengartikan kata kunci dalam input dan membalasnya dengan kata kunci yang paling cocok, atau pola kata-kata yang paling mirip dari data yang telah ada dalam *database* yang telah dibuat sebelumnya. (Richard, 2010).

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Chatbot telah dimanfaatkan untuk tujuan praktis seperti bantuan *online*, layanan personal, atau akuisisi informasi, dalam hal ini dapat dilihat fungsi program sebagai suatu jenis agen percakapan (*conversational agent*). Perbedaan *chatbot* dengan sistem pemrosesan bahasa alami (*Natural Language Processing System*) adalah kesederhanaan algoritma yang digunakan. Meskipun banyak *bots* yang tampaknya dapat menginterpretasikan dan menanggapi input manusia, sebenarnya *bots* tersebut hanya memindai kata kunci dalam input dan membalasnya dengan kata kunci yang paling cocok, atau pola kata-kata yang paling mirip dari basis data tekstual. Istilah "*Chatbot*" sendiri pertama kali dikemukakan oleh Michael Mauldin (pencipta Verbot pertama, Julia) pada tahun 1994 (Mauldin, 1994).

2.3.1 Perkembangan Chatbot

Program *chatbot* pertama ditulis oleh Joseph Weizenbaum, profesor MIT pada tahun 1966. Pada zamannya program *chatbot* dibuat dengan sederhana. Meskipun perkembangan kecerdasan buatan saat ini sangat pesat dan canggih, namun *chatbot* tetap mempertahankan kedudukannya dalam dunia *Artificial Intellegence* (Weizenbaum, 1966).

Sejarah klasik dari *chatbot* awal adalah ELIZA (1966) dan PARRY (1972). (Güzeldere, 1995) (Computer History Museum, 2006) (Sondheim, 1997) (Network Working Group, 1973). Program yang baru-baru saja dikembangkan yaitu A.L.I.C.E, Jabberwacky dan D.U.D.E. Pada masanya, ELIZA dan PARRY digunakan untuk menstimulasi percakapan tertulis, namun banyak *chatbot* kini mendukung fitur fungsional seperti permainan dan kemampuan pencarian *website*. Tahun 1984, sebuah buku berjudul *The Policeman's Beard is Half*

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Constructed dipublikasikan. Buku ini diduga ditulis oleh sebuah *chatbot* Racter – walaupun program ini dirilis untuk tidak mampu melakukannya.

Salah satu penelitian penting di bidang kecerdasan buatan (AI) adalah pemrosesan bahasa alami (atau *Natural Language Processing*). Biasanya, bidang AI lemah memberdayakan perangkat lunak (atau *software*) khusus atau bahasa pemrograman yang dibuat secara spesifik dengan fungsi yang lebih sempit. Contohnya, A.L.I.C.E menggunakan bahasa pemrograman yang disebut sebagai AIML yang fungsinya spesifik yaitu sebagai agen percakapan, yang selanjutnya banyak diadopsi oleh pengembang *Alicebots* lain. Meskipun demikian, A.L.I.C.E masih murni berdasarkan teknik pencocokan pola tanpa kemampuan penalaran – teknik yang sama yang digunakan ELIZA pada tahun 1966. Berbeda dengan AI kuat, yang membutuhkan cita rasa dan kemampuan penalaran logis.

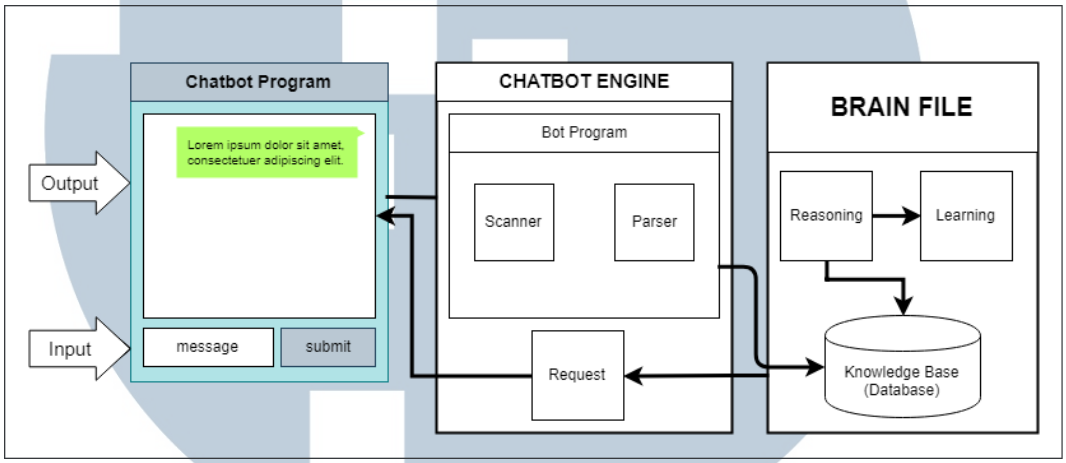
Jabberwacky mempelajari respons baru dan berbasis pada konteks interaksi pengguna waktu nyata (atau *real-time*), bukan dengan digerakan basis data statis. Beberapa *chatbot* terbaru juga mengkombinasikan pembelajaran waktu nyata dengan algoritma evolusioner yang mengoptimalkan kemampuan komunikasi berbasis percakapannya, dengan contoh populernya yaitu Kyle, pemenang Penghargaan Leodis AI 2009. Meskipun, saat ini belum ada tujuan umum percakapan kecerdasan buatan, dan beberapa pengembang perangkat lunak berfokus pada aspek praktis, pengambilan informasi (atau *information retrieval*).

Sistem percakapan otomatis kini telah berkembang, dan perusahaan-perusahaan sudah menggunakan sitem-sistem tersebut untuk membantu *call center* memberikan panduan kontak. *Chatbot* pun sudah dimplementasikan melalui jejaring sosial, seperti twitter dan Windows Live Messenger. Portal *online* populer

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

seperti eBay dan PayPal juga menggunakan agen *virtual* multi bahasa untuk memudahkan penggunanya. Selain itu *Chatbot* juga di implementasikan untuk bidang komersial, pendidikan, entertainment dan sektor pelayanan publik. (Kerly dkk., 2006)

2.3.2 Arsitektur Chatbot



Gambar 2.22 Arsitektur *Chatbot* (Rudiyanto, 2005)

Chatbot terdiri dari dua komponen utama yakni *bot* program dan *brain file*.

A. Bot program

Bot Program merupakan program utama pada *chatbot* yang akan mengakses input dari pengguna, melakukan *parsing* dan kemudian membawanya ke *brain file* (*knowledge base*) untuk kemudian diberikan respon. Adapun *bot* program sendiri terdiri dari komponen *Scanner* dan *parser*. (Rudiyanto, 2005)

1. *Scanner*

Scanner merupakan salah satu bagian dari kompilator bahasa pada komputer yang bertugas melakukan analisis leksikal. *Scanner* menerima input berupa *stream* karakter kemudian memilah program sumber yang akan menjadi *input* bagi *parser*. Di dalam aplikasi *chatbot*, yang dimaksud dengan program sumber yang diolah oleh *Scanner* adalah berupa kalimat *input* dari pengguna. Contohnya pada saat ada

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

inputan dari *user*, maka otomatis dari *Scanner* akan memproses inputan, dan akan meneruskan ke tahap selanjutnya, namun pada tahap ini lebih khusus mendeteksi keberadaan inputan yang dimasukkan dan mengubah inputan menjadi bentuk normal kondisi yang diterima sistem. (Rudiyanto, 2005)

2. *Parser*

Parser atau *syntactic analyzer* pada kompilator bahasa pemrograman berfungsi untuk memeriksa kebenaran kemunculan inputan yang kemudian akan dijadikan token. Pada *Chatbot system*, fungsi dari *parser* ini agak berbeda karena token yang akan diolah, semuanya memiliki tipe yang sama yaitu berupa kata (*word*). Urutan token akan diolah dengan mengacu pada *brain file* agar didapatkan makna kalimat yang sesungguhnya. Dengan kata lain, tahap analisa semantik terjadi di bagian *brain file*. Kemampuan dari *parser* untuk mengolah token dan bekerja sama dengan *brain file* inilah yang paling menentukan tingkat kecerdasan dari sebuah *chatbot*. Contoh penerapan inputan yang masuk pada chatbot akan dipecah menjadi token-token sehingga pada tahapan seanjutnya akan dilakukan pencocokan dengan kosakata kata kunci yang sudah ada pada *brain file*. (Rudiyanto, 2005)

B. Brain file

Brain File merupakan otak dari *chatbot* itu sendiri yang menentukan bagaimana cara *chatbot* berpikir dan akan memberikan respon. *Brain file* berfungsi sebagaimana tabel informasi (*knowledge base*). Di dalam *brain file* inilah disimpan semua kosakata, kepribadian, dan pengetahuan (*knowledge*) dari *chatbot*. Semakin banyak pengetahuan yang dimiliki *chatbot* maka akan semakin besar ukuran *file* dari *brain file* tersebut. Secara lebih rinci komponen *chatbot* itu berupa:

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

1. *Reasoning*

Reasoning adalah teknik penyelesaian masalah dengan cara mempresentasikan masalah ke dalam basis pengetahuan (*knowledge base*) menggunakan *logic* atau bahasa formal. Pada penerapan *chatbot* proses ini akan dikerahkan bila kata kunci terdapat dalam *knowledge base chatbot* dan dilakukan untuk mengembalikan respon ke pengguna. Proses ini menunjukkan bahwa masukan yang diberikan oleh pengguna tidak diproses sebagai satuan kata, tetapi sebagai kalimat utuh.

2. *Learning*

Learning merupakan pendefinisian aturan tertentu secara otomatis dalam menemukan aturan yang diharapkan bisa berlaku umum untuk data-data yang belum pernah kita ketahui. Pada penerapan *chatbot* proses *learning* dijalankan bila kata kunci pada masukan pengguna tidak terdapat dalam *knowledge base*. Kata kunci yang tidak ditemukan tersebut akan disimpan sebagai dialog *repository* untuk kemudian akan ditanyakan pada *bot* program. Contoh penerapannya pada *chatbot* adalah sewaktu *chatbot* tidak mendapati kesamaan antara kata kunci dalam *knowledge base* dengan inputan *user*, maka *chatbot* akan menyimpan kosakata pertanyaan tersebut untuk dipelajari dikemudian hari. (Hadinata dan Novita, 2007).

2.3.3 Prinsip Kerja Chatbot

Bot program atau bagian aplikasi menentukan kemampuan dan keterampilan *chatbot* untuk berbicara pada anda atau pada pengguna lainnya, atau dengan kata lain *bot* program berperan sebagai mulut. *Chatbot* tidak tahu apa-apa tidak punya nama dan tidak punya kepribadian. Untuk itu pengguna harus mengajarinya berbagai hal sehingga ia dapat berbicara dengan baik dan semua pelajaran tersebut dimasukkan ke *brain file* (Hadinata dan Novita, 2007).

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Sebagai contoh, pengguna mengeluh bahwa *chatbot* yang akan dirancang berbicara omong kosong dengan mengetik kalimat: "*You are talking nonsense, pinhead*!". Agar *chatbot* dapat memberikan respon terhadap kalimat tersebut, maka dapat ditambahkan baris berikut pada *brain file* dari *chatbot*:

You are talking nonsense ← trigger line

I am smarter than you ← respon bot

Dengan demikian jika pengguna sekarang *chatting* menggunakan kalimat yang disebutkan di atas maka *bot* program akan masuk ke dalam *brain file* dan kemudian memberikan respon dengan *output* yang sesuai dengan *trigger line* yaitu *you are talking nonsense* (Rudiyanto, 2005).

Respon yang sama juga dapat muncul jika anda juga menuliskan baris berikut pada *brain file*:

Nonsense ← trigger line

I am smarter than you ← respon bot

Respon yang sama juga akan muncul jika anda menuliskan:

a. *Talking nonsense ← trigger line*

   I am smarter than you ← respon bot

b. *Pinhead ← trigger line*

   I am smarter than you ← respon bot

Namun akan lebih bagus jika *chatbot* diberikan kemampuan untuk merespon kalimat yang beragam misalnya:

c. *You are talking nonsense ← trigger line*

   I am smarter than you ← respon bot

d. *Talking Nonsense ← trigger line*

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Talking is funny, isnt it? ← *respon bot*

e. *Nonsense* ← *trigger line*

Don't be stupid! ← *respon bot*

f. *Pinhead* ← *trigger line*

Please try not to be insulting,buddy ← *respon bot*

Dengan *brain file* seperti itu maka *chatbot* akan terasa lebih hidup karena dapat memberikan respon dengan berbagai kalimat yang berbeda.

2.4 Natural Language Processing

Menurut Eugene Charniak (1984) N*atural Language Processing* (NLP) adalah cabang ilmu komputer dan linguistik yang mengkaji interaksi antar komputer dengan bahasa (alami) manusia (Charniak, 1984).

NLP memodelkan pengetahuan terhadap bahasa, baik dari segi kata, bagaimana kata-kata bergabung menjadi satu kalimat dan konteks kata dalam kalimat (Barakbah, 2010).

NLP berupaya memecahkan masalah untuk memahami bahasa alami manusia, dengan segala aturan gramatika dan semantiknya, dan mengubah bahasa tersebut menjadi representasi formal yang dapat diproses oleh komputer (Putstejovsky, 2012).

Berdasarkan (Putstejovsky, 2012) dalam penerapannya, tujuan NLP untuk memahami bahasa manusia ini memiliki banyak tantangan, antara lain sebagai berikut:

UNIVERSITAS MULTIMEDIA NUSANTARA

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

1. Penandaan kelas kata (*part-of-speech tagging*).

Sulit untuk menandai kelas kata (kata benda, kata kerja, kata sifat, dsb.) suatu kata dalam teks karena pengelasan kata sangat bergantung kepada konteks penggunaannya.

2. Segmentasi teks (*text segmentation*).

Penentuan segmentasi sulit dilakukan pada bahasa tulis yang tidak memiliki pembatasan kata spesifik (misal: bahasa Mandarin, Jepang, dan Thailand) serta pada bahasa lisan yang kadang membaurkan bunyi antarkata.

3. Disambiguasi makna kata (*word sense disambiguation*).

Banyak kata memiliki lebih dari satu makna, baik dalam bentuk homonim (makna berbeda dan tidak terkait) maupun polisemi (makna berbeda, namun terkait). Pembedaan makna hanya dapat dilakukan dengan melihat konteks penggunaan.

4. Ambiguitas sintaksis (*syntactic ambiguity*).

Suatu bahasa memiliki berbagai kemungkinan struktur kalimat. Pemilihan struktur yang paling tepat biasanya membutuhkan gabungan informasi semantik dan kontekstual.

5. Masukan yang tak sempurna atau tak teratur (*imperfect or irregular input*).

Aksen dalam bahasa lisan serta kesalahan ejaan dan gramatikal dalam bahasa tulis menyulitkan pemrosesan bahasa alami.

6. Penuturan (*speech act*).

Struktur kalimat saja kadang tidak dapat dengan tepat menggambarkan maksud penutur atau penulis. Kadang gaya bahasa dan konteks menentukan maksud yang diinginkan.

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Disamping kesulitan-kesulitan tersebut, NLP juga berhasil diterapkan untuk berbagai tugas yang semua hanya dapat dilakukan oleh manusia. Beberapa bidang populer dalam penerapan NLP adalah sebagai berikut:

1. Pemerolehan informasi (*information retrieval*). Pencarian dokumen yang relevan, pencarian informasi spesifik di dalam dokumen, serta pembuatan metadata.
2. Penjawaban pertanyaan (*question answering*). Secara otomatis menjawab pertanyaan yang diajukan dengan bahasa alami dengan jawaban dalam bahasa alami pula.
3. Perangkuman otomatis (*automatic summarization*). Pembuatan versi singkat berisi butir-butir penting dari suatu dokumen dengan menggunakan program komputer.
4. Penerjemahan mesin (*machine translation*). Penerjemahan otomatis dari suatu bahasa alami ke bahasa lain.
5. Pengenalan wicara (*speech recognition*). Pengubahan bahasa lisan menjadi masukan yang dikenali oleh mesin.
6. Sintesis wicara (*speech synthesis*). Pengubahan bahasa tulis menjadi bahasa lisan, kebalikan dari pengenalan wicara.
7. Pengenalan karakter opsis (*optical character recognition*). Pengubahan tulisan tangan atau teks tercetak (melalui pemindai) menjadi dokumen yang dapat dikenali oleh mesin.
8. Analisis sentimen (*Sentiment Analysis*). Ekstraksi informasi dari sumber data teks untuk mendeteksi pangangan positif atau negatif terhadap suatu objek.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Biasanya diterapkan untuk mengidentifikasi tren opini publik terhadap suatu produk atau perusahaan.

Bidang-bidang pengetahuan yang berhubungan dengan pengolahan bahasa alami adalah sebagai berikut:

1. Fonetik dan fonologi : berhubungan dengan suara yang menghasilkan kata yang dapat dikenali. Bidang ini penting dalam aplikasi yang memakai metode *speech-based system*.
2. Morfologi : yaitu pengetahuan tentang kata dan bentuknya yang dimanfaatkan untuk membedakan satu kata dengan kata lainnya. Pada tingkat ini juga dapat dipisahkan antara kata dan elemen lain seperti tanda baca.
3. Sintaksis : yaitu pemahaman tentang urutan kata dan pembentukan kalimat dan hubungan antar kata tersebut dalam proses perubahan bentuk dari kalimat menjadi sesuatu yang sistematis.
4. Semantik : yaitu pemetaan bentuk struktur sintaksis dengan memanfaatkan tiap kata ke dalam bentuk yang lebih mendasar dan tidak tergantung dengan struktur kalimat.
5. Pragmatik : berkaitan dengan tingkatan pengetahuan masing-masing konteks yang berbeda tergantung pada situasi dan tujuan pembuatan sistem.
6. *Discourse knowledge* : melakukan pengenalan apakah suatu kalimat yang sudah dibaca dan dikenali sebelumnya dalam mempengaruhi arti dari kalimat selanjutnya. Informasi ini penting diketahui untuk melakukan pengolahan arti terhadap kata ganti orang dan untuk mengartikan aspek sementara dari informasi.

UNIVERSITAS MULTIMEDIA NUSANTARA

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

7. *World knowledge* : mencakup arti sebuah kata secara umum dan apakah ada arti khusus bagi suatu kata dalam suatu percakapan dengan konteks tertentu.

Jenis aplikasi yang terdapat pada bidang pengolahan bahasa alami antara lain adalah :

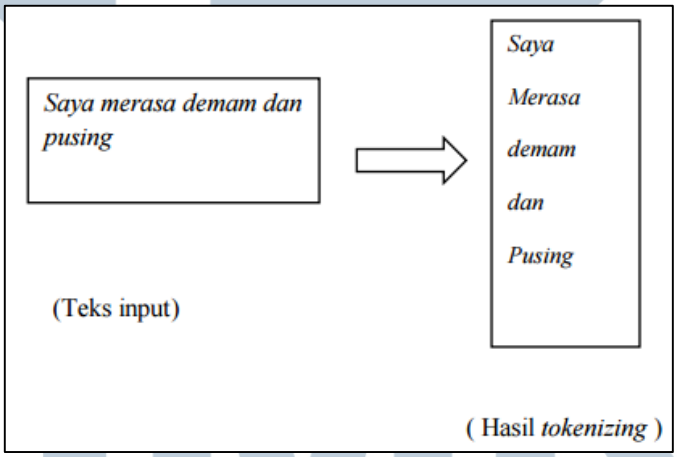
1. *Text-based application* : mencakup segala macam aplikasi yang melakukan terhadap teks tertulis seperti pada buku, berita di surat kabar, *e-mail* dan lain sebagainya. Contoh penggunaan aplikasi ini antara lain adalah mencari topik tertentu dari buku yang ada di perpustakaan, mencari isi dari surat atau *e-mail*, menterjemahkan dokumen dari satu bahasa ke bahasa yang lain. Akan tetapi, tidak semua sistem dapat melakukan hal yang demikian, misalnya pada pencarian topik dari suatu buku di perpustakaan dapat dilakukan dengan pendekatan sistem *database* yang cukup lengkap. Salah satu bentuk yang cukup menarik adalah jika sistem diminta mencari isi dari suatu buku atau artikel, dimana pendekatan yang dilakukan sama seperti pendekatan yang dilakukan oleh manusia jika menghadapi suatu *tes reading* and *comprehension*.
2. *Dialogue-based application* : merupakan pendekatan yang melibatkan bahasa lisan atau pengenalan suara. Akan tetapi, bidang ini juga memasukkan interaksi dengan cara memasukkan teks pertanyaan melalui keyboard. Aplikasi yang sering ditemui dalam bidang ini antara lain seperti sistem tanya jawab dimana *natural language* digunakan dalam mendapatkan informasi dari *database*, sistem otomasi pelayanan melalui telepon, kontrol suara pada peralatan elektronik, sistem *problem-solving* yang membantu untuk melakukan penyelesaian masalah yang umum dihadapi dalam suatu pekerjaan.

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Perlu diketahui bahwa untuk sistem yang dapat melakukan interaksi melalui bahasa lisan ada pada bagian *speech recognition* yang merupakan bagian terpisah dari *natural language*.

2.4.1 Tokenizing (Parsing)

Tokenisasi adalah suatu proses pemecahan sebuah teks menjadi bagian yang lebih kecil. Bagian yang lebih kecil ini dikenal dengan istilah "token". Tujuannya adalah untuk memecah suatu kalimat ke dalam sebuah *array* kata (NlpTools, 2016). Tokenisasi sangat dibutuhkan karena sistem tidak hanya menganalisa keseluruhan kalimat, tetapi menganalisa setiap kata yang membentuk sebuah kalimat. Tahap tokenisasi akan menghasilkan sebuah *array* kata yang siap diproses oleh fungsi NLP berikutnya seperti *part of speech tagging*. (Domarco, 2016).



Gambar 2.23 Tahap *Tokenizing* (Suhendro, 2014)

2.4.2 Scanner (Analisis Leksikal)

Analisis Leksikal (*Scanner*) merupakan antarmuka antara kode program sumber dan analisa sintaktik (parser). Atau dalam pengertiannya adalah sebuah proses yang mendahului *parsing* sebuah rangkaian karakter. *Scanner* melakukan pemeriksaan karakter per karakter pada teks masukan, memecah sumber program

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

menjadi bagian-bagian disebut Token. Proses *parsing* akan lebih mudah dilakukan bila inputnya sudah berupa token.

Analisis leksikal membuat pekerjaan membuat sebuah *parser* jadi lebih mudah daripada membangun nama setiap fungsi dan variabel dari karakter-karakter yang menyusunnya, dengan analisis leksikal *parser* cukup hanya berurusan dengan sekumpulan token dan nilai sintaksis masing-masing.

Terlepas dari efisiensi pemrograman yang dapat dicapai dengan penggunaannya, proses kerja analisis leksikal yang membaca lebih dari sekali setiap karakter dari input yang diberikan menjadikan penganalisa leksikal sebagai sub-sistem yang paling intensif melakukan komputasi, terutama bila digunakan dalam sebuah kompilator.

Kompilator adalah sebuah program yang membaca suatu program yang ditulis dalam suatu bahasa sumber (*source language*) dan menterjemahkannya ke dalam suatu bahasa sasaran (*target language*). Dalam penguraian struktur kalimat, penganalisa leksikal menganalisa setiap kata dalam kalimat, kemudian menentukan jenis kelas katanya.

Hasil dari penganalisa leksikal ini digunakan oleh penganalisa sintaks yang akan memeriksa urutan simbol-simbol kelas kata tersebut dalam kalimat. Analisa kata dalam kalimat ini dilakukan oleh penganalisa leksikal berdasarkan kecocokan kata dengan aturan-aturan leksikal berupa ekspresi regular yang sudah didefinisikan.

Tugas dari *scanner* adalah sebagai berikut :

1. Melakukan pembacaan kode sumber dengan mengurut karakter demi karakter
2. Mengenali besaran leksik

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

3. Mentransformasi menjadi sebuah token dan menentukan jenis tokennya.
4. Mengirimkan token
5. Membuang mengabaikan *blank* dan komentar dalam program
6. Menangani kesalahan
7. Menangani tabel simbol

Ketika *scanner* menerima *input* berupa *stream* karakter kemudian memilah menjadi satuan leksik, satuan leksik tersebut terdiri atas simbol-simbol satuan yang jika dikombinasikan akan mempunyai arti yang berbeda-beda. Simbol-simbol yang bisa dipergunakan dalam sebuah bahasa tentunya terbatas jumlahnya, yang membentuk sebuah himpunan dan disebut sebagai abjad (alphabet).

Tata bahasa (grammatika) adalah sekumpulan dari himpunan variabel-variabel, simbol-simbol terminal, simbol non-terminal, simbol awal yang dibatasi oleh aturan-aturan produksi. Aturan produksi adalah pusat dari tata bahasa yang menspesifikasikan bagaimana suatu tata bahasa melakukan transformasi suatu *string* ke bentuk lainnya.

Dalam pembicaraan *grammar*, anggota alfabet dinamakan simbol terminal atau token. Kalimat adalah *string* yang tersusun atas simbol-simbol terminal. Bahasa adalah himpunan kalimat-kalimat. Anggota bahasa bisa berupa tak berhingga hingga kalimat.

Simbol-simbol berikut adalah simbol terminal (Suhendro, 2014):

1. Huruf kecil alphabet, misalnya :a, b, c
2. Simbol operator, misalnya : +, -, dan ‘
3. Simbol tanda baca, misalnya : (,), dan ;

Sedangkan simbol-simbol berikut adalah simbol non terminal:

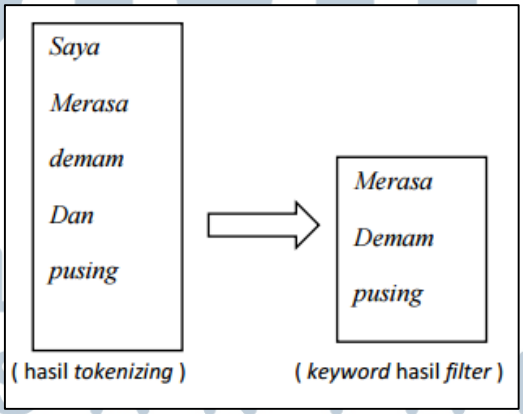
Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

1. Huruf besar awal alphabet, misalnya : A, B, C
2. Huruf S sebagai simbol awal

A. Stopword Removal / Filtering

Kebanyakan bahasa resmi di berbagai negara memiliki kata fungsi dan kata sambung seperti artikel dan preposisi yang hampir selalu muncul pada dokumen-dokumen teks. Biasanya kata-kata ini memiliki arti yang lebih di dalam memenuhi kebutuhan seorang *searcher* di dalam mencari informasi. Kata-kata tersebut (misalnya a, an, the on pada bahasa inggris) disebut sebagai *stopwords*. Di dalam bahasa Indonesia *stopwords* dapat disebut sebagai kata tidak penting misalnya "di", "oleh", "pada", "sebuah", "karena". Sebelum proses *stopwords removal* dilakukan, terlebih dulu dibuat daftar *stopwords* (*stoplist*). Preposisi, kata hubung dan partikel biasanya merupakan kandidat *stoplist*.

Stopwords removal merupakan proses penghilangan kata tidak penting pada deskripsi melalui pengecekan kata-kata hasil *parsing* deskripsi apakah termasuk di dalam daftar kata tidak penting (*stoplist*) atau tidak. Jika termasuk di dalam *stoplist* maka kata-kata tersebut akan di-*remove* dari deskripsi sehingga kata-kata yang tersisa di dalam deskripsi di anggap sebagai kata-kata penting atau *keywords.* (Suhendro, 2014)



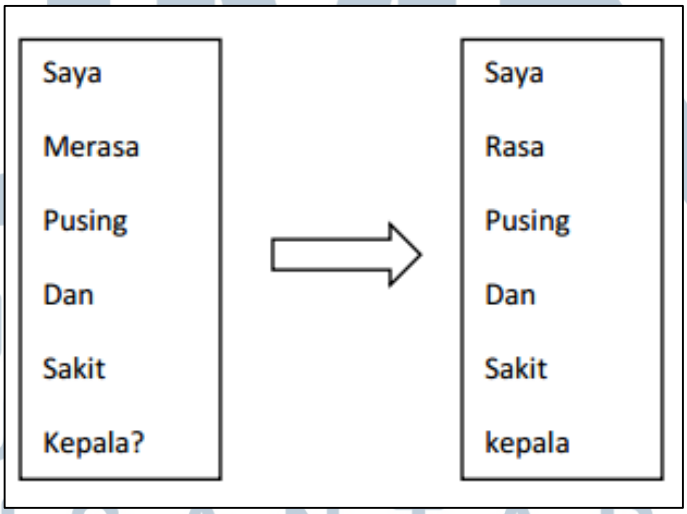
Gambar 2.24 Tahap *Filtering* (Suhendro, 2014)

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

B. Stemming

Stemming adalah proses pemetaan dan penguraian berbagai bentuk (variants) dari suatu kata menjadi bentuk kata dasarnya (*stem*). Proses ini juga disebut sebagai *conflation*. Proses *stemming* secara luas sudah digunakan di dalam *Information retrieval* (pencarian informasi) untuk meningkatkan kualitas informasi yang didapatkan. Kualitas informasi yang dimaksud misalnya untuk mendapatkan hubungan antara varian kata yang satu dengan yang lainnya. Sebagai contoh kata "diculik", "menculik" (melakukan tindakan menculik) dan "penculik" (orang yang menculik) yang semula mengandung arti yang berbeda dapat di-*stem* menjadi sebuah kata "culik" yang memiliki arti yang sama sehingga kata-kata diatas saling berhubungan.

Selain itu *stemming* juga dapat digunakan untuk mengurangi ukuran dari suatu ukuran index file. Misalnya dalam suatu deskripsi terdapat varian kata "memberikan", "diberikan", "memberi" dan "diberi" hanya memiliki akar kata (*stem*) yaitu "beri". Ukuran file daftar index yang semula berjumlah lima *record* akan di-*reduce* sehingga menjadi satu *record* saja.



Gambar 2.25 Tahap *Stemming* (Suhendro, 2014)

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

2.5 Algoritma Enhanced Confix Stripping Stemmer

Stemming dapat dimanfaatkan untuk berbagai sistem yang membutuhkan *term preprocessing* sebagai awal prosesnya. Perkembangan *stemming* bermula pada algortima Porter *Stemmer* tahun 1980, lalu Algoritma Nazief-Adriani tahun 1996, *Confix Stripping Stemmer* tahun 2007 dan *terakhir Enhanced Confix Stripping Stemmer* tahun 2008. Algoritma *Enhanced Confix Stripping Stemmer* dikembangkan oleh Putu Adhi Kerta Mahendra pada tahun 2008 sebagai hasil evaluasi penelitian sebelumnya, yaitu memperbaiki kekurangan pada algoritma *Confix Stripping Stemmer*. Perbaikan dilakukan dengan menambah dan memodifikasi tabel aturan pemenggalan, lalu menambah fungsionalitas pengembalian akhiran untuk mengantisipasi kesalahan pemenggalan yang seharusnya tidak dilakukan (Anggara dkk., 2013).

Algoritma *Enhanced Confix Stripping* (ECS) *Stemmer* adalah algoritma *steaming* kata pada Bahasa Indonesia dengan performa yang paling baik (memiliki jenis kesalahan *stemming* yang paling sedikit) (Arifin dkk., 2009). Algoritma ECS *Stemmer* ini merupakan algoritma perbaikan dari algoritma *Confix Stripping* (CS) *Stemmer*. Perbaikan yang dilakukan oleh ECS *Stemmer* adalah perbaikan beberapa aturan pada tabel acuan pemenggalan imbuhan. Selain itu, algoritma ECS *Stemmer* juga menambahkan langkah pengembalian akhiran jika terjadi penghilangan akhiran yang seharusnya tidak dilakukan (Tahitoe dan Purwitasari, 2010).

Enhanced Confix Stripping (ECS) *Stemmer* diperkenalkan sebagai metode perbaikan terhadap beberapa kata yang gagal di-*stemming* oleh metode *confix stripping stemmer*. Selain itu algoritma ECS *stemmer* juga menambahkan langkah pengembalian akhiran jika terjadi penghilangan akhiran yang seharusnya tidak

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

dilakukan. Pada sistem ini hasil dari *stemming* pada saat proses pemenggalan imbuhan selesai dilakukan, hasil *stemming* dengan metode ini harus ada pada kamus kata dasar, jika tidak ada *term* yang di inputkan dianggap sebagai bentuk dasar (Asian dan William, 2007).

Aturan morfologi Bahasa Indonesia mengelompokan imbuhan menjadi beberapa kategori sebagai berikut (Anggara dkk., 2013).

1. *Inflection Suffixes* (IS), yaitu kelompok-kelompok akhiran yang tidak mengubah kata dasar. Kelompok akhiran ini dapat dibagi lagi menjadi 2, yaitu:
   a. *Particle* (P), atau partikel, diantaranya "-lah", "-kah", "-tah" dan "-pun".
   b. *Possessive Pronoun* (PP), atau akhiran yang menambahkan arti sebagai kepunyaan, diantaranya "-ku", "-mu" dan "-nya".
2. *Derivation Suffixes* (DS), yaitu kumpulan akhiran yang secara langsung ditambahkan pada kata dasar, diantaranya "-i", "-kan", dan "-an".
3. *Derivation Prefixes* (DP), yaitu kumpulan awalan yang dapat langsung diberikan pada kata dasar murni, atau pada kata dasar yang sudah mendapatkan penambahan sampai dengan 2 awalan. Berdasarkan perubahan bentuknya awalan dibagi menjadi 2 :
   a. Awalan yang bermorfologi, diantaranya "me-", "be-", "pe-", dan "te-"
   b. Awalan yang tidak bermorfologi , diantaranya "di-", "ke-" dan "se-".

Gabungan awalan dan akhiran menjadi imbuhan membentuk kata dasar menjadi kata baru yang memiliki makna berbeda dari kata dasar sebelumnya, tetapi tidak semua kombinasi diperbolehkan. Kombinasi yang tidak diperbolehkan diantaranya "be- i","di- -an","ke- -i", "ke- -kan", "me- -an", "se- -i", "se- -kan", dan "te- -an" (Anggara dkk., 2013). Kombinasi terlarang menyebabkan kata dasar

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

menjadi kata baru yang tidak baku. Dengan model kata bahasa Indonesia, aturan-aturan dasar morfologi pada bahasa Indonesia serta pengklasifikasian imbuhan-imbuhan seperti di atas, maka bentuk kata berimbuhan dalam bahasa Indonesia dapat dimodelkan sebagai berikut :

[DP+ [DP+ [DP+]]] Kata Dasar [[+DS] [+PP] [+P]] ... (2.1)

Skema algortima *Enhanced Confix Stripping Stemmer* adalah sebagai berikut (Anggara dkk., 2013).

1. Kata dalam dokumen uji sudah mengalami proses *preprocessing* dan berbentuk kata yang tunggal. Lakukan pengecekan terhadap kata, cari kata dalam kamus, jika kata ditemukan sebagai kata yang *valid* dalam kamus maka kata tesebut dianggap sudah berbentuk kata dasar, jika tidak maka lakukan langkah 2.
2. Hapus *Inflection Suffixes*. Akhiran yang pertama dihilangkan adalah akhiran yang termasuk Particles (P) yaitu “-lah”, “-kah”, “-tah” atau “-pun”. Setelah itu langkah selanjutnya penghilangan akhiran yang termasuk *Possesive Pronouns* (PP) yaitu “ku”, “-mu” atau “-nya”.
3. Hapus *Derivation Suffixes* (DS) yaitu “-i”, “-kan”, atau “-an”.
4. Hapus *Derivation Prefixes* (DP) yaitu “di-”, “ke-”, “se-”,“me-”, “be-”, “pe-”, atau “te-”. Langkah empat memiliki keadaan khusus dimana jika mengalami keadaan khusus maka langkah tambahan dilakukan. Langkah tambahan adalah sebagai berikut:
   a. Langkah 4 akan terhenti jika penghilangan awalan sudah 3 kali.
   b. Proses penghilangan DP:

UNIVERSITAS MULTIMEDIA NUSANTARA

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

- Standar, penghilangan awalan yang tidak bermorfologi, awalan dapat langsung dihilangkan karena tidak merubah kata dasar.
- Kompleks, penghilangan awalan yang bermorfologi, awalan yang mengalami perubahan bentuk sesuai dengan huruf awal kata dasar, penghilangan awalan sesuai dengan tabel aturan pemenggalan imbuhan.

c. Cek kombinasi terlarang. Jika kata memiliki kombinasi terlarang maka proses *stemming* berhenti dan kata dikelompokkan menjadi kata tak terdeteksi karena kata tersebut tidak baku.

d. Cari kata dalam kamus, apabila ditemukan maka proses dihentikan dengan status kata sudah ditemukan, apabila tidak ditemukan maka langkah 4 diulang kembali sebanyak 3 kali. Jika proses berhenti tetapi kata belum ditemukan maka dilanjutkan pada langkah 5.

5. Lakukan *Recoding*, proses mengacu pada tabel aturan pemenggalan imbuhan. *Recoding* adalah penambahan karakter di awal kata yang dipenggal, karakter *recoding* adalah karakter serelah tanda hubung ("-") atau bentuk lain setelah tanda (" | "). Sebagai contoh kata "menunda" setelah dipenggal menjadi "nunda", karena kata "nunda" tidak valid maka di-*recoding* sehingga kata "nunda" menjadi "tunda" dan valid ketika dicek pada kamus kata dasar.
6. Menambahkan suatu fungsionalitas tambahan untuk mengatasi kesalahan pemenggalan akhiran yang seharusnya tidak dilakukan. Fungsionalitas ini disebut *loop* Pengembalian Akhiran. Fungsionalitas Loop Pengembalian Akhiran dideskripsikan sebagai berikut:

a. Kembalikan seluruh awalan yang telah dihilangkan sebelumnya, sehingga menghasilkan model kata seperti berikut:

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

[DP+[DP+[DP]]] + Kata Dasar ... (2.2)

Pemenggalan awalan dilanjutkan dengan proses pencarian di kamus kemudian dilakukan pada kata yang telah dikembalikan menjadi model tersebut.

b. Kembalikan akhiran sesuai dengan urutan model pada bahasa Indonesia. Ini berarti bahwa pengembalian dimulai dari DS (“-i”, “- kan”, “-an”), lalu PP (“-ku”, “-mu”, “-nya”), dan terakhir adalah P (“- lah”, “-kah”, “-tah”, “-pun”). Khusus untuk akhiran “-kan”, pengembalian pertama dimulai dengan “k”, baru kemudian dilanjutkan dengan “an”.
c. Lakukan pengecekan di kamus kata dasar. Apabila ditemukan, proses dihentikan. Apabila gagal, maka lakukan proses pemenggalan awalan berdasarkan tabel aturan pemenggalan imbuhan.
d. Lakukan *recoding* apabila diperlukan.
e. Apabila semua langkah diatas telah dilakukan lalu kata tidak ditemukan setelah pengecekan pada kamus, maka kata dikembalikan dengan semua imbuhan dan kata dimasukkan dalam kelompok kata tak terdeteksi.

7. Jika kata tidak ditemukan setelah semua langkah dilakukan maka kata tersebut dikelompokan pada kata tak terdeteksi. Pada tabel 2.2 dapat dilihat aturan ECS.

Tabel 2.2 Aturan Pemenggalan Awalan ECS Stemmer (Anggara dkk., 2013)

Aturan	Format Kata	Pemenggalan
1	berV...	ber-V... | be-rV...
2	berCAP...	ber-CAP... dimana C!=”r”& P!=”er”
3	berCAerV...	ber-CaerV... dimana C!=”r”
4	Beajar	bel-ajar
5	beClerC2...	be-C!erV... dimana C1!={“r”|”1”}
6	terV...	ter-V... | te-rV...
7	terCerV...	ter-CerV.. dimana C!=”r”
8	terCP...	terCP... dimana C!=”r” dan P!=”er”
9	terC!erC2...	te-ClerC2... dimana C1!=”r”

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

Tabel 2.2 Aturan Pemenggalan Awalan ECS Stemmer (Lanjutan)
(Anggara dkk., 2013)

10	me{1|r|w|y|}V…	me-{1|r|w|y|}V…
11	mem{b|f|v}…	mem-{b|f|v}…
12	mempe…	mem-pe…
13	mem{rV|V}…	me-m{rV|V}… | me-p{rV|V}…
14	men{c|d|j|t|z}…	men-{c|d|j|t|z}…
15	menV…	me-nV… | me-tV…
16	meng{g|h|q|k}…	meng-{g|h|q|k}…
17	mengV…	me-ngV.. | meng-V… | meng-kV… | mengV…jika v=”e”
18	menyV…	me-nyV… | meny-sV…
19	mempA…	mem-pA dimana A!=”e”
20	pe{w|y}V…	pe-{w|y}V…
21	perV…	per-V… | pe-rV…
22	perCAP…	per-CAP… dimana C!=”r” dan P!=”er”
23	perCAerV…	per-CAerV… dimana C!=”r”
24	pem{b|f|v}…	pem-{b|f|v}…
25	pem{rV|V}…	pe-m{rV|V}… | pe-p{rV|V}…
26	pen{c|d|j|z}…	pen-{c|d|j|z}…
27	penV…	pen-V… | pe-tV…
28	pengC…	peng-C…
29	pengV…	Pe-ngV..| peng-V…| peng-kV…| pengV…jika V=”e”
30	penyV…	pe-nyV…| peny-sV…
31	pelV…	pe-1V… kecuali “pelajar” yang menghasilkan “ajar”
32	peCerV…	peC-erV… dimana C!={r|w|y|1|m|n}
33	peCP…	pe-CP… dimana C!={r|w|y|1|m|n} dan P!=”er”
34	terC1erC2..	ter-C1erC2.. dimana C1!=”r”
35	peC1erC2..	pe-C1erC2.. dimana C!={r|w|y|1|m|n}
36	CerV…	CerV…| CV…
37	CelV…	CelV…| CV…
38	CemV…	CemV…|CV…

Keterangan dari Tabel 2.2 dan Tabel 2.3 adalah :

Huruf “C” menandakan huruf konsonan

Huruf “V” menandakan huruf vokal

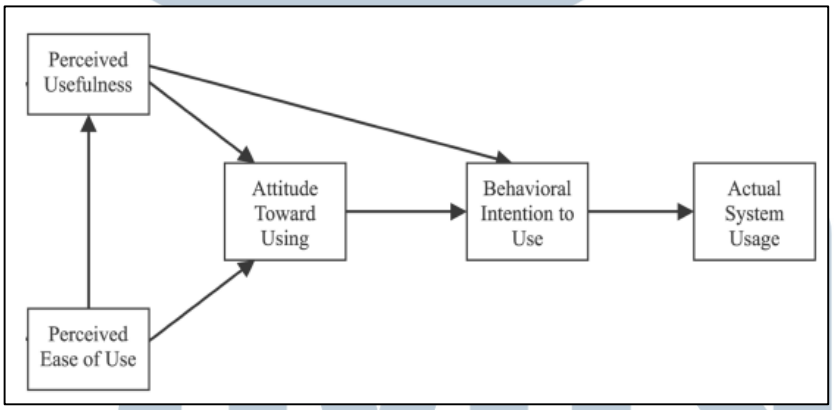
Huruf “A” menandakan huruf apa saja

Huruf “P” menandakan fragmen kata yang pendek seperti “er”

UNIVERSITAS MULTIMEDIA NUSANTARA

2.6 Metode Technology Acceptance Model

Technology Acceptance Model (TAM) merupakan *model* evaluasi sistem yang digunakan untuk mengetahui apakah *user* menerima dan berkehendak untuk menggunakan teknologi baru (Davis, 1989). TAM menggunakan 2 (dua) variabel utama sebagai faktor yang mempengaruhi *user*, yaitu *perceived ease of use* dan *perceived usefulness*. *Perceived ease of use* merupakan tingkat kepercayaan *user* bahwa sistem dapat dipahami dan digunakan tanpa membutuhkan banyak usaha (*effort*) untuk mempelajarinya (Davis, 1989). *Perceived usefulness* merupakan tingkat kepercayaan *user* bahwa penggunaan sistem dapat meningkatkan kinerja (Davis, 1989). *Ease of use* dan *usefulness* merupakan faktor penting dalam proses evaluasi kualitas dari *online service* (Rod dkk., 2009). TAM telah banyak dipelajari dan dikembangan (Sharma dkk., 2014). TAM dapat dilihat pada Gambar 2.26.



Gambar 2.26 Technology Acceptance Model (Davis, 1989)

Beberapa penerapan metode modifikasi TAM (*Extended Technology Acceptance Model*) berhubungan dengan *trust* dan *user satisfaction* telah dilakukan oleh Al-Gahtani (2011), Shawar dkk. (2014), dan Aufar (2009). Menurut Zeithaml (2002), *user satisfaction* merupakan tingkat atau tolak ukur dari *product service* apakah *product* atau *service* tersebut telah memenuhi kebutuhan dan ekspektasi

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

user. *Trust* merupakan tingkat kepercayaan *user* untuk menggunakan sistem dengan mempertimbangkan karakteristik sistem (Al-Gahtani, 2011).

2.7 Skala Likert

Menurut Sugiyono (2014) skala Likert merupakan sebuah metode untuk mengukur data kuantitatif dan kualitatif. Skala ini menggunakan model *rating* dalam pengukurannya. Model *rating* menyediakan jawaban yang bersifat kuantitatif. Dengan demikian, jawaban dari pertanyaan yang bersifat kualitatif dapat disajikan dalam sebuah data kuantitatif. Skala Likert menggunakan nilai positif dan negatif sebagai batas pengukuran. Pemodelan rating dalam skala Likert secara umum menggunakan lima buah poin penilaian, dimulai dari nilai tidak setuju sampai setuju dengan nilai tengahnya berupa nilai netral. Perhitungan skor pada Likert (1932) dibagi menjadi lima kriteria yang ditunjukkan pada Tabel 2.4 dengan 'X' sebagai nilai.

Tabel 2.3 Kriteria Skala Likert (Sugiyono, 2014)

Kategori	Kritera	Syarat
SA	*Strongly Agree*	X ≥ 80%
A	*Agree*	60% ≤= X < 80%
N	*Neither*	40% ≤ X < 60%
D	*Disagree*	20% ≤= X < 40%
SD	*Strongly Disagree*	0% ≤= X < 20%

Berdasarkan Tabel Kriteria Skala Likert, rumus yang digunakan untuk menghitung skor adalah sebagai berikut.

$$X = \frac{(SA * nSA) + (A * nA) + (N * nN) + (D * nD) + (SD * nSD)}{\text{Jumlah Pertanyaan} * \text{Jumlah Sampel}} \quad \ldots (2.3)$$

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

2.8 Objek Wisata DKI Jakarta

Pembangunan merupakan suatu atau rangkaian usaha pertumbuhan dan perubahan yang dilakukan secara sadar oleh suatu bangsa (Siagian, 2005). Pelaksanaan pembangunan bisa dari bermacam-macam aspek kehidupan masyarakat, salah satunya adalah pariwisata. Pariwisata merupakan salah satu program pemerintah yang dapat menambah pemasukan devisa yang besar bagi pemerintah dalam rangka menunjang berhasilnya pembangunan nasional, khususnya dalam pengelolaan dan pengembangan zona industri, apalagi industri-industri yang ada semakin berkembang dengan pesat mulai dari industri rumah tangga hingga industri yang berskala besar (Santosa dkk., 2015).

Terdapat beberapa definisi dari berbagai pakar mengenai konsep pariwisata:

- Menurut Wahab (2003), Pariwisata adalah salah satu industri dari gaya baru, yang mampu meyediakan pertumbuhan ekonomi yang cepat dalam hal kesempatan kerja, pendapatan, taraf hidup dan dalam mengaktifkan sektor produksi lain di dalam negara penerima wisatawan.
- Secara etimologi, kata pariwisata berasal dari bahasa sansekerta yang terdiri atas dua kata yaitu pari dan wisata. Pari berarti “banyak” atau “berkeliling”, sedangkan wisata berarti “pergi“ atau “bepergian”. Sehingga pariwisata adalah perjalanan yang dilakukan berkali-kali untuk berputar-putar, dari suatu tempat ketempat yang lain (Suwena dkk., 2010).
- Pariwisata sebagai segala kegiatan dalam masyarakat yang berkaitan dengan wisatawan (Soekadijo, 2000).

UNIVERSITAS MULTIMEDIA NUSANTARA

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

2.8.1 Faktor Pendorong Pariwisata

Faktor-faktor yang mendorong pariwisata diantaranya sebagai berikut (Fandeli, 1995) :

1. Keinginan untuk melepaskan diri dari tekanan hidup sehari-hari di kota, keinginan untuk mengubah suasana dan memanfaatkan waktu senggang.
2. Kemajuan pembangunan dalam bidang komunikasi dan transportasi.
3. Keinginan untuk melihat dan memperoleh pengalaman-pengalaman baru mengenai budaya masyarakat dan di tempat lain.

2.8.2 Jenis-Jenis Pariwisata

Jenis-jenis pariwisata terbagi menjadi 7, yaitu (Pendit, 1994) :

1. Wisata Budaya, ini dimaksudkan dengan perjalanan yang dilakukan atas dasar keinginan unutk memperluas pandangan hidup seseorang, mempelajari keadaan rakyat, kebiasaan, dan adat istiadat mereka, cara hidup mereka budaya, dan seni mereka (seni tari, drama, musik, dan seni suara).
2. Wisata Konvensi, Berbagai negara dewasa ini membangun wisata konvensi dengan menyediakan fasilitas bangunan dengan ruangan-ruangan tempat bersidang bagi para peserta suatu konfrensi. Misalnya Indonesia memiliki Balai Sidang Senayan di Jakarta untuk penyelenggaraan sidang-sidang pertemuan yang besar dengan perlengkapan yang modern.
3. Wisata Pertanian (Agrowisata) adalah pengorganisasian perjalanan yang dilakukan ke proyek-proyek pertanian, perkebunan, ladang pembibitan, dan sebagainya dimana wisatawan dapat mengadakan kunjungan dan peninjauan untuk tujuan studi maupun untuk sekedar menikmati aneka macam tanaman.

UNIVERSITAS MULTIMEDIA NUSANTARA

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

4. Wisata Maritim (Bahari), jenis wisata ini biasanya dikaitkan dengan kegiatan oleh raga di air, danau, pantai, teluk, dan laut seperti memancing, berlayar, menyelam sambil melakukan pemotretan, kompetisi berselancar, balapan mendayung, melihat–lihat taman laut dengan pemandangan indah di bawah permukaan air serta berbagai rekreasi perairan yang banyak dilakukan di daerah–daerah atau negara–negara maritim.
5. Wisata Cagar Alam (Taman Konservasi), wisata diselenggarakan oleh agen atau biro perjalanan yang mengkhusukan usaha-usaha dengan jalan mengatur wisata ke tempat atau daerah cagar alam, taman lindung dan sebagainya yang kelestariannya dilindungi oleh undang-undang.
6. Wisata Buru, jenis wisata ini banyak dilakukan di negeri-negeri yang banyak memiliki daerah atau hutan berburu yang diperbolehkan oleh pemerintah dan digalakkan oleh berbagai agen atau biro perjalanan. di Indonesia pemerintah membuka wisata buru untuk daerah Baluran di Jawa Timur dimana wiatwan boleh menembak banteng dan babi hutan.
7. Wisata Ziarah (Pilgrim), jenis wisata ini sedikit banyak dikaitkan dengan agama, sejarah, adat istiiadat dan kepercayaan umat atau kelompok dalam masyarakat. Wisata ziarah banyak dilakukan oleh perorangan atau rombongan ke tempat-tempat suci, ke makam-makam orang besar atau pemimpin yang diagungkan, ke bukit atau gunung yang dianggap keramat, tempat pemakaman tokoh atau pemimpin sebagai manusia ajaib penuh legenda.

UNIVERSITAS MULTIMEDIA NUSANTARA

Faktor-faktor yang diperhatikan dalam suatu objek wisata menurut Dwiputra (2013) dan Warlan (2015):

1. Faktor Daya Tarik
2. Faktor Aksesibilitas
    a. Kondisi Jalan Yang Dilalui
    b. Jaringan Transportasi
    c. Waktu Tempuh
    d. Biaya Perjalanan
    e. Frekuensi Kendaraan Angkutan
    f. Jarak tempuh dari tempat tinggal menuju objek wisata
3. Faktor Fasilitas
4. Sarana Wisata
    a. Tempat Makan
    b. Tempat Belanja

2.8.3 Daya Tarik Wisata di Jakarta

Sebagai Ibu kota negara, Jakarta memiliki nilai sejarah dan budaya yang sangat penting bagi perjalanan sejarah negara Indonesia. Daya tarik wisata di Jakarta memiliki beragam karakteristik dan keunikan tersendiri, yang terdiri dari daya tarik wisata bahari, sejarah, budaya, religi, rekreasi dan hiburan, kuliner, olahraga dan kebugaran (Ginting, 2016).

1. Kota Administrasi Jakarta Selatan

Selain sebagai wilayah dengan jasa usaha pelayanan makanan dan minuman terbanyak, Jakarta Selatan juga merupakan wilayah dengan daya tarik wisata dan pusat belanja terbanyak di Provinsi DKI Jakarta yaitu terdapat 35 lokasi, di mana

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

74,28% (26 lokasi) merupakan pusat belanja dan kuliner (ITC dan Mall). Dengan banyaknya hotel yang tersedia baik bintang maupun non bintang yang berlokasi di Jakarta Selatan, maka wisatawan yang berwisata di Jakarta memiliki banyak motivasi dan kesempatan berwisata belanja dan menikmati kuliner di wilayah tempat mereka menginap. Taman Margasatwa Ragunan yang terdapat di wilayah Jakarta Selatan merupakan daya tarik wisata dengan jumlah pengunjung terbanyak ketiga di Provinsi DKI Jakarta yaitu 4.090.567 orang berdasarkan data tahun 2015 (Januari – Desember). Daya tarik wisata dengan produk unggulan Taman Rekreasi, Kebon Binatang (270 jenis satwa jumlah 3.500) dan Atraksi (pentas satwa) didirikan tahun 1864 di Cikini kemudian dipindah ke kawasan Ragunan Jakarta Selatan (Ginting, 2016).

2. Kota Administrasi Jakarta Timur

Wilayah Jakarta Timur memiliki 35 daya tarik wisata, di mana 8 lokasi merupakan pusat belanja, 8 daya tarik wisata pendidikan dan sejarah, serta memiliki 1 daya tarik wisata minat khusus yaitu Pacuan Kuda Pulomas. Pacuan kuda ini masih perlu ditingkatkan kualitasnya dan perlu untuk diadakan event-event pacuan kuda untuk meningkatkan minat generasi muda terhadap olahraga berkuda. Pada wilayah ini, daya tarik wisata yang paling memiliki daya tari bagi wisatawan adalah daya tarik wisata Taman Mini Indonesia Indah (TMII) yang merupakan miniatur Indonesia (Indonesia Mini) dengan jumlah pengunjung sebanyak 5.186.465 orang, kondisi menurun dibandingkan tahun 2014 dengan kunjungan sebanyak 5.298.719 orang, namun demikian masih tetap menduduki posisi kedua daya tarik wisata terbanyak setelah Ancol. Di dalam kawasan wisata Taman Mini Indonesia Indah selain terdapat rumah adat dari 33 provinsi provinsi di Indonesia, berbagai museum

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

juga melengkapi informasi tentang transportasi, keprajuritan, flora fauna dan lain-lain (Ginting, 2016).

3. Kota Administrasi Jakarta Utara

Berdasarkan jenis daya tarik wisata, wilayah ini memiliki jenis daya tarik wisata yang paling variatif. Dari 20 daya tarik wisata yang ada terdiri dari Wisata Pendidikan, Wisata Sejarah dan Budaya, Wisata Religi. Wisata Bahari, Wisata Rekreasi dan Hiburan, Wisata Alam, Wisata Belanja dan Kuliner. Keragaman ini tentu menjadi daya tarik tersendiri bagi wisatawan yang datang ke Jakarta Utara baik wisatawan mancanegara maupun wisatawan nusantara. Hal ini diperkuat dengan dipublikasikannya 12 jalur wisata pesisir utara yang terdiri dari daya tarik wisata, taman suaka margasatwa Muara Angke, pasar ikan Muara Angke, kawasan Sunda Kelapa, Jakarta Islamic Center, Sentra Belanja Mangga Dua, Taman Impian Jaya Ancol, Bantera Jaya Ancol, Kampung Tugu, Kampung Marunda, dan Rumah Si Pitung, Sentra Belanja dan Kuliner Kelapa Gading. Taman Impian Jaya Ancol (TIJA Ancol) sebagai Theme Park untuk rekreasi hiburan yang besar di daerah utara Jakarta, menjadi daya tarik wisata terfavorit bagi wisatawan nusantara. Jumlah pengunjung/wisata nusantara yang mengunjungi Ancol pada tahun 2015 yaitu 21.354.785 orang (Januari-Desember), dan merupakan pengunjung terbesar dibandingkan daya tarik wisata lain di DKI Jakarta (Ginting, 2016).

4. Kota Administrasi Jakarta Barat

Wilayah barat Jakarta ini memiliki 22 daya tarik wisata yang terdiri dari Wisata Religi (5), Wisata Belanja (7) dan sebanyak 10 lokasi yang merupakan daya tarik wisata sejarah dan pendidikan. Di wilayah Jakarta Barat memiliki daya tarik wisata Jembatan Kota Intan dan Museum Bahari yang memberikan gambaran Jakarta

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

sebagai pelabuhan besar dan memiliki kanal-kanal yang menghubungkan laut dengan sungai di Jakarta. Dalam wilayah Jakarta Barat memiliki museum sejarah Jakarta dengan jumlah pengunjung terbanyak keempat di Provinsi Jakarta sebanyak 437.040 orang. Kunjungan ke Museum Sejarah Jakarta mengalami penurunan dibandingkan tahun 2014 sebanyak 448.501 orang. Daya tarik wisata Museum Sejarah yang terletak di Jalan Taman Fatahilah, merupakan wisata sejarah budaya dan wisata pendidikan. Tingginya minat wisatawan untuk melakukan wisata baik dalam komunitas sejarah, Sahabat Museum yang sering melakukan kegiatan wisata hingga saat ini minat pengunjung ke kawasan Kota Tua semakin tinggi. Dalam perkembangannya kawasan wisata kota tua semakin ramai dikunjungi wisatawan dan masyarakat Jakarta, di satu sisi kondisi ini menggembirakan namun dampak dari banyaknya pengunjung juga mengakibatkan semrawutnya kawasan dengan pedagang kaki lima dan asongan (Ginting, 2016).

5. Kota Administrasi Jakarta Pusat

Wilayah ini memiliki 42 daya tarik wisata (wilayah dengan daya tarik wisata terbanyak), salah satu andalan daya tarik wisata di Provinsi DKI Jakarta dan menjadi kebanggaan yaitu Monumen Nasional (MONAS). Daya tarik wisata dikunjungi sebanyak 1.156.153 orang meningkat 17,33% dibandingkan kunjungan tahun 2014 sebanyak 1.253.266 pengunjung. Jakarta Pusat merupakan saksi sejarah perjuangan Kemerdekaan Republik Indonesia, antara lain Museum Joang ’45, Museum Kebangkitan Nasional, Museum Perumusan Naskah Proklamasi, Museum Sumpah Pemuda, Monumen Proklamasi. Keberadaan daya tarik wisata tersebut hendaknya lebih sering disosialisasikan pada generasi muda tidak saja melalui

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

program Wajib Kunjung Museum tetapi juga melalui event yang melibatkan generasi muda dan dengan media peliputan yang intensitas tinggi (Ginting, 2016).

6. Kota Administrasi Kepulauan Seribu

Kepulauan seribu merupakan gugusan pulau terbentang di perairan Teluk Jakarta yang terdiri dari pulau besar dan kecil berjumlah sekitar 110 pulau. Pulau-pulau tersebut terbagi atas pulau wisata, konservasi sejarah dan pemukiman sesuai karakteristik. Kepulauan seribu merupakan wilayah Kabupaten Administrasi dengan luas wilayah mencapai 11,81 km2 yang terbentang dari pantai utara Jakarta hingga 100 mil laut ke arah utara. Dibutuhkan waktu tempuh berkisar 20 menit ke pulau terdekat hingga 3 jam untuk mencapai pulau terjauh. Secara umum keadaan laut di wilayah Kepulauan Seribu memiliki kedalaman 0-40 m, dan ada dua tempat yang memiliki kedalaman lebih dari 40 m yaitu di sekitar Pulau Payung dan Pulau Pari. Cuaca baik di Kepulauan Seribu sekitar bulan-bulan Maret, April sampai Mei. Curah hujan tertinggi terjadi di bulan Januari, dan bulan kering di bulan Juni sampai September. Ada dua angin musim yang dominan di Kepulauan Seribu yaitu angin musim barat disertai hujan lebat dan angin musim timur yang kering. Data kunjungan wisatawan ke Kepulauan Seribu berikut data daya tarik wisata di beberapa pulau di Kepulauan Seribu, begitu pula dengan data daya tarik wisata di lima wilayah Kota Administrasi di Provinsi DKI Jakarta. Dengan perkembangan teknologi informasi dan gencarnya program televisi tentang kegiatan wisata maka semakin marak kegiatan wisata yang dilakukan oleh masyarakat Jakarta baik di dalam kota maupun hingga ke Kepulauan Seribu, begitu juga dengan kunjungan dari wisatawan nusantara. Selain itu generasi muda saat ini mulai banyak yang tertarik untuk mengunjungi dan mempelajari sejarah kota Jakarta tidak saja di

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019

sekitar kawasan Kota Tua namun hingga potensi wisata sejarah di Kepulauan Seribu antara lain di Pulau Onrust, Kelor, Pula Cipir dan Pulau Bidadari. Pengaruh jejaring sosial seperti twitter, facebook, friendster juga memotivasi wisatawan untuk saling memberikan informasi mengenai layak tidaknya suatu daya tarik wisata dikunjungi, Trend pariwisata dunia yang lebih mengarah kepada "green tourism" dan "eco tourism" juga berdampak positif bagi wisatawan produktif/muda untuk wisata petualangan. Selain itu muncul pula trend wisata dalam kemasan paket wisata seperti "dark tourism" yang artinya wisata untuk mengunjungi daya tarik wisata atau daerah yang pernah mengalami kekelaman atau keharuan yang dalam seperti; penjara bawah tanah di Museum Sejarah Jakarta, Lubang Buaya dan sebagainya. Hal yang masih menjadi perdebatan baru untuk kemasan paket wisata di Jakarta adalah dengan munculnya istilah Jakarta Hidden Tour yang awalnya merupakan perjalanan yang dikemas oleh LSM Lingkungan dengan mengunjungi tempat kumuh di Jakarta dan kemudian peserta tour memberikan kontribusi kepada daerah yang dikunjungi untuk membantu perbaikan fasilitas dan sebagainya. (Ginting, 2016).

UNIVERSITAS MULTIMEDIA NUSANTARA

Rancang bangun chatbot..., Richaer Quamala, FTI UMN, 2019