BAB II
LANDASAN TEORI

2.1 Chatbot

Program *chatbot* pertama ditulis oleh Joseph Weizenbaum, profesor MIT pada tahun 1966. Pada waktu itu *chatbot* dibuat masih sangat sederhana. Meskipun perkembangan kecerdasan buatan saat ini sangat pesat dan canggih, namun *chatbot* tetap mempertahankan kedudukannya dalam dunia *Artificial Intellegence* (AI). *Chatbot* merupakan salah satu bentuk aplikasi *Natural Language Processing* (NLP) (Nila & Afrianto, 2015).

Percakapan yang terjadi antara komputer dengan manusia merupakan bentuk respon dari program yang telah dideklarasikan pada database program. Respon yang dihasilkan merupakan hasil pemindaian kata kunci pada *input user* dan menghasilkan respon balasan yang dianggap paling cocok, atau pola kata-kata yang dianggap paling mendekati (Ridwan, 2013).

Chatbot terdiri dari komponen *bot program* dan komponen *brain file*. *Bot program* merupakan program utama pada *chatbot* yang akan mengakses *input* dari pengguna, melakukan *parsing* dan kemudian membawanya ke *brain file* untuk kemudian diberikan respon. *Bot program* terdiri dari komponen *scanner* dan *parser*. *Brain file* merupakan otak dari *chatbot* itu sendiri yang menentukan bagaimana cara *chatbot* berpikir dan akan memberikan respon. *Brain file* biasanya berupa file *plain text*. *Brain file* berfungsi sebagaimana tabel informasi pada kompilator bahasa pemrograman tingkat tinggi. Di dalam *brain file* inilah disimpan semua kosakata, kepribadian, dan pengetahuan (*knowledge*) dari *chatbot*. Semakin

banyak pengetahuan yang dimiliki *chat bot* maka akan semakin besar ukuran file dari *brain file* tersebut (Rudiyanto, 2005).

2.2 Natural Language Processing

Natural Language Processing (NLP) merupakan bagian dari *Artificial Intelligence* (AI). NLP mempelajari komunikasi antara manusia dengan komputer melalui bahasa alami (Chopra, Prashar, & Sain, 2013). Tahap-tahap dalam NLP adalah sebagai berikut.

1. *Morphological and Lexical Analysis*

   Morfologi merupakan tentang kata dan bentuknya agar dapat dibedakan antara satu dengan yang lainnya. *Lexical analysis* merupakan pembagian teks ke dalam paragraf, kata dan kalimat (Chopra, Prashar, & Sain, 2013).

2. *Syntactic Analysis*

   Syntactic analysis merupakan analisis terhadap urutan kata dalam pembentukan kalimat (Chopra, Prashar, & Sain, 2013).

3. *Semantic Analysis*

   Semantic analysis merupakan analisis terhadap arti suatu kata dan bagaimana arti kata tersebut membentuk suatu arti kata dari kalimat yang utuh (Chopra, Prashar, & Sain, 2013).

4. *Discourse Integration*

Discourse integration merupakan hubungan antar kalimat, apakah suatu kalimat yang telah dikenali mempengaruhi kalimat selanjutnya (Chopra, Prashar, & Sain, 2013).

5. *Pragmatic Analysis*

Pragmatic analysis merupakan analisis terhadap konteks kata/kalimat yang berhubungan dengan sebuah keadaan atau topik (Chopra, Prashar, & Sain, 2013).

2.3 Text Preprocessing

Text preprocessing merupakan tahapan awal dalam mengolah data *input* sebelum memasuki proses tahapan utama (Ramdhan, 2015). Pada tahapan ini keberadaan digit angka, huruf kapital, atau karakter-karakter yang lainnya akan dihilangkan dan diubah (Marfian, 2015). Berikut adalah tahapan *text preprocessing.*

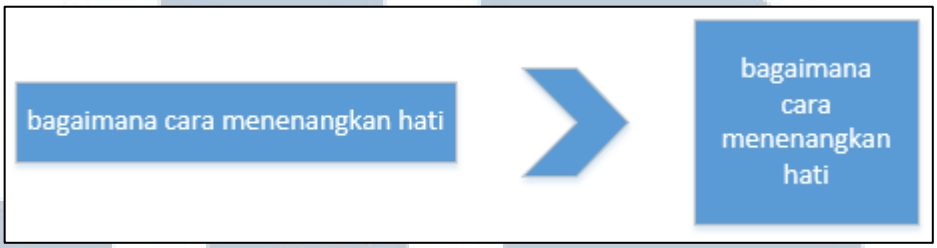
1. Case Folding

Case folding adalah tahap yang merubah semua huruf menjadi huruf kecil. Tahap ini hanya menerimaa huruf ‘a’ sampai dengan ‘z’. Karakter selain huruf akan dihilangkan dan dianggap sebagai pembatas.

Gambar 2.1 Proses Case Folding (Marfian, 2015)

2. Tokenizing

Tahap *tokenizing* adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Spasi digunakan untuk memisahkan antar kata tersebut.



Gambar 2.2 Proses Tokenizing (Ramdhan, 2015)

2.4 Text Transformation

Text transformation tahapan yang tujuannya mengubah kata-kata ke dalam bentuk dasar dan mengurangi jumlah kata-kata tersebut. Berikut adalah tahapan *text transformation.*

1. Stopwords Removal/Filtering

Filtering adalah mengambil kata-kata penting dari hasil *tokenizing.* Proses ini dapat dilakukan dengan membuang kata yang kurang penting (*stopwords*/*stoplist*) dari hasil *tokenizing*. *Stopwords*/*stoplist* adalah kata-kata yang dapat dibuang. Contoh *stopwords* adalah “yang”, “dan”, “di”, “dari”, dan lain-lain.

Gambar 2.3 Proses Stopwords Removal (Ramdhan, 2015)

2. Stemming

Stemming adalah proses untuk mencari akar dari kata hasil proses *filtering*. Pencarian akar sebuah kata atau kata dasar dapat memperkecil hasil indeks tanpa harus menghilangkan makna (Utomo, 2013). Proses *stemming* juga bertujuan untuk melakukan pengelompokan kata-kata yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk yang berbeda karena mendapatkan imbuhan yang berbeda (Tahitoe & Purwitasari, 2010).

Gambar 2.4 Proses Stemming (Ramdhan, 2015)

2.5 Algoritma Nazief & Adriani

Algoritma Nazief & Adriani dikembangkan pertama kali oleh Bobby Nazief dan Mirna Adriani. Algoritma ini berdasarkan pada aturan morfologi bahasa Indonesia yang luas, yang dikumpulkan menjadi satu grup dan dienkapsulasi pada imbuhan/*affixes* yang diperbolehkan (*allowed affixes*) dan imbuhan/*affixes* yang tidak diperbolehkan (*disallowed affixes*). Algoritma ini menggunakan kamus kata dasar dan mendukung *recoding*, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih (Wahyudi, Susyanto, & Nugroho, 2017).

Aturan morfologi Bahasa Indonesia mengelompokan imbuhan ke dalam beberapa kategori sebagai berikut.

1. Inflectional suffixes

Kelompok akhiran yang tidak merubah bentuk kata dasar. *Inflectional suffixes* ini dibagi menjadi dua.

a. Particle (P), yakni termasuk di dalamnya "-lah", "-kah", "-tah", dan "-pun".

b. Possesive pronoun (PP), kata ganti kepunyaan, termasuk di dalamnya adalah "-ku" , "- mu", dan "-nya".

2. Derivational suffixes (DS)

Kumpulan akhiran asli Bahasa Indonesia yang secara langsung ditambahkan pada kata dasar yaitu akhiran "-i", "-kan", dan "-an".

3. Derivational Prefix (DP)

Kumpulan awalan yang dapat langsung diberikan pada kata dasar murni, atau pada kata dasar yang sudah mendapatkan penambahan sampai dengan 2 awalan, yaitu awalan yang dapat bermorfologi ("me-", "be-", "pe-", dan "te-") dan awalan yang tidak bermorfologi ("di-", "ke-" dan "se-").

Berdasarkan pengklasifikasian imbuhan-imbuhan di atas, maka bentuk kata berimbuhan dalam Bahasa Indonesia dapat dimodelkan sebagai [DP+ [DP+ [DP+]]] Kata Dasar [[+DS] [+PP] [+P]] (Tahitoe & Purwitasari, 2010).

Langkah-langkah algoritma Nazief & Adriani adalah sebagai berikut.

1. Kata dicari pada kamus. Jika ditemukan, kata tersebut adalah kata dasar dan algoritma berhenti (Asian, 2007).
2. Hilangkan *Inflectional suffixes*, yaitu dengan menghilangkan *particle* dan *possessive pronoun*. Kata dicari pada kamus. Jika ditemukan, kata tersebut adalah kata dasar dan algoritma berhenti, jika tidak lanjut ke langkah 3 (Asian, 2007).

3. Hapus *Derivational Suffix*. Kata dicari pada kamus. Jika ditemukan, kata tersebut adalah kata dasar dan algoritma berhenti. Jika tidak, maka lanjut ke langkah 3a.

a. Jika akhiran "-an" telah dihapus dan huruf terakhir dari kata tersebut adalah "-k", maka "-k" juga dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah 3b (Asian, Williams, & Tahaghoghi, 2005).

b. Akhiran yang dihapus ("-i", "- an" atau "-kan") dikembalikan, lanjut ke langkah 4 (Asian, Williams, & Tahaghoghi, 2005).

4. Hapus *Derivational Prefix*. Kata dicari pada kamus. Jika ditemukan, kata tersebut adalah kata dasar dan algoritma berhenti, jika tidak, maka lakukan *recoding*. Tahapan ini dihentikan jika memenuhi beberapa kondisi berikut.

a. Terdapat kombinasi awalan dan akhiran yang tidak diijinkan (Asian, Williams, & Tahaghoghi, 2005).

Tabel 2.1 Kombinasi Awalan dan Akhiran yang Tidak Diizinkan (Asian, 2007)

Awalan	**Akhiran**
Be-	-i
Di-	-an
Ke-	-i, -kan
Me-	-an
Se-	-i, -kan

b. Awalan yang dideteksi sama dengan awalan yang dihilangkan sebelumnya.

Tabel 2.2 Jenis Awalan Berdasarkan Tipe Awalan (Asian, 2007)

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	belajar…	bel-ajar…
5	beC1erC2…	be-C1erC2 … | dimana C1!={'r'| 'l'}

Tabel 2.1 Jenis Awalan Berdasarkan Tipe Awalan (Asian, 2007) (Lanjutan)

6	terV…	ter-V… | te-rV…
7	terCerV…	ter-CerV… | dimana C!='r'
8	terCP…	ter-CP… | dimana C!='r' and P!='er'
9	teC1erC2…	te-C1erC2… | dimana C1!='r'
10	me{l|r|w|y}V…	me-{l|r|w|y}V…
11	mem{b|f|v}…	mem-{b|f|v}…
12	mempe{r|l}…	mem-pe…
13	mem{rV|V}…	me-m{rV|V}… | me-p{rV|V}…
14	men{c|d|j|z}…	men-{c|d|j|z}…
15	menV…	me-nV… | me-tV…
16	meng{g|h|q}…	meng-{g|h|q}…
17	mengV…	meng-V… | meng-kV…
18	menyV…	meny-sV…
19	mempV…	mem-pV… | dimana V!='e'
20	pe{w|y}V…	pe-{w|y}V…
21	perV…	per-V… | pe-rV…
23	perCAP…	per-CAP… | dimana C!='r' and P!='er'
24	perCAerV…	per-CAerV… | dimana C!='r'
25	pem{b|f|v}…	pem-{b|f|v}…
26	pem{rV|V}…	pe-m{rV|V}… | pe-p{rV|V}…
27	pen{c|d|j|z}…	pen-{c|d|j|z}…
28	penV…	pe-nV… | pe-tV…
29	peng{g|h|q}…	peng-{g|h|q}…
30	pengV…	peng-V… | peng-kV…
31	penyV…	peny-sV…
32	pelV…	pe-lV… | Exception: for "pelajar", return ajar
33	peCerV…	per-erV… | dimana C!={r|w|y|l|m|n}
34	peCP…	pe-CP… | dimana C!={r|w|y|l|m|n} and P!='er'

Keterangan:
C : huruf konsonan
V : huruf vokal
A : huruf vokal atau konsonan
P : partikel atau fragmen dari suatu kata, misalnya "er"

c. Cari kata yang telah dihilangkan awalannya ini di dalam kamus. Apabila tidak ditemukan, maka langkah 4 diulangi kembali. Apabila ditemukan, maka keseluruhan proses berhenti (Asian, Williams, & Tahaghoghi, 2005).

5. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai kata dasar. Proses selesai (Asian, Williams, & Tahaghoghi, 2005).

2.6 Artificial Intelligence Markup Language (AIML)

Artificial Intelligence Markup Language (AIML) adalah bahasa *scripting* turunan dari *Extensible Markup Language* (XML) dengan fungsi yang lebih spesifik. Salah satu fungsinya adalah membuat sistem *stimulus-respond* berbasis pengetahuan. AIML terdiri dari objek-objek yang dipisahkan oleh *tag-tag* tertentu seperti layaknya dokumen XML atau HTML (Azwary, Indriani, & Nugrahadi, 2016).

Terdapat tiga *tag* utama dalam objek AIML, yaitu *category* yang mendefinisikan satu *unit knowledge* dari suatu percakapan, *pattern* yang mengidentifikasi *input* dari *user*, dan *template* yang digunakan untuk mencatat respon yang akan diberikan berdasarkan *input* tertentu dari user. *Pattern* ditulis dalam huruf kapital untuk standarisasi dan simplifikasi proses *pattern matching* dari *input* user (SANTOSO, 2011).

Tag yang digunakan pada AIML dijelaskan sebagai berikut.

a. Tag <aiml>

Merupakan *tag* yang menandai awal dan akhir suatu dokumen AIML (Bahartyan, Bahtiar, & Waspada, 2014). *Tag* ini mengandung *atribute version* dan *encoding*.

```
<?xml version="1.0" encoding="utf-8"?>
<aiml>
        <category>
                <pattern>NAMA LENGKAP KAMU APA</pattern>
                <template>
                        Nama lengkap saya adalah Chatbot Popoyo.
                </template>
        </category>
</aiml>
```

Gambar 2.5 Penggunaan Tag <aiml>, <category>, <pattern>, dan <template>

(Setiawan, 2017)

b. Tag <category>

Setiap *tag category* merupakan unit dasar pengetahuan yang terkandung dalam *chatbot knowledge base* (Marietto, et al., 2013).

c. Tag <pattern>

Tag ini berisi *input user* yang telah melalui proses dengan *text preprocessing* dan *text transformation*. Hanya ada satu <pattern> pada satu <category>. Kata-kata di dalam *tag* dipisahkan dengan satu spasi dan penggunaan *wildcards* dapat mensubtitusi bagian dari kalimat tersebut (Marietto, et al., 2013).

d. Tag <template>

Tag ini 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 (Marietto, et al., 2013).

e. Tag <star index = “n”>

Tag ini menyimpan dan memetakan komponen bagian dari kalimat pada *user input*. Indeks “n” menunjukkan frase yang dipetakan oleh *star index*. *Tag* ini merupakan konteks opsional pada *category*. Untuk memetakan *input user* pada *tag pattern*, digunakan simbol atau *wildcard* “*”. *Tag* <star/> mempunya arti yang sama dengan <star index=”1”>.

```
<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 2.6 Penggunaan Tag <star index = “n”> (Marietto, et al., 2013)

f. Tag <srai>

```
<?xml version="1.0" encoding="utf-8"?>
<aiml>
	<category>
		<pattern>ANGELICA AGNESIA DOSEN APA</pattern>
		<template>
			Wah, mantap jiwa nih. Aku kurang tahulah. Mudah-mudah aja masuk mata kuliah Pemrograman Berorientasi Objek di kelas kita ya.
		</template>
	</category>
	<category>
		<pattern>YUDI SETIAWAN DOSEN APA</pattern>
		<template>
			Tahulah. Anak kelas sebelah pernah cerita kalau dia itu dosen mata kuliah pemrograman mobile.
		</template>
	</category>
	<category>
		<pattern>NIA AMELIZA DOSEN APA</pattern>
		<template>
			Oh, dia itu dosen mata kuliah TechnoPreneurship.
		</template>
	</category>
	<category>
		<pattern>KAMU TAHU * NGAJAR MATA KULIAH APA</pattern>
		<template>
			<srai><star /> DOSEN APA</srai>
		</template>
	</category>
</aiml>
```

Gambar 2.7 Penggunaan Tag <srai> (Setiawan, 2017)

Salah satu properti yang sangat berguna dalam AIML di mana kemampuannya untuk menargetkan <pattern> yang berbeda untuk satu <template>. Dengan demikian, interpreter AIML dapat mencari jawaban dari *input user* yang berbeda-beda. Hal ini dilakukan dengan menggunakan *tag* <*srai*>.

g. Tag <random> dan

Tag <*random*> digunakan untuk memberikan respon ke *user* dengan cara yang berbeda. Setiap kalimat respon yang memungkinkan ditulis dalam *tag* . Respon *chatbot* ditangani sebagai *list* dan jawaban akan dipilih secara acak oleh AIML *interpreter*.

```
1  <category>
2     <pattern> HI </pattern>
3     <template>
4        <random>
5           <li> Hi! Nice to meet you </li>
6           <li> Hello, How are you? </li>
7           <li> Hello! </li>
8        </random>
9     </template>
10 </category>
```

Gambar 2.8 Tag <random> dan <li> (Marietto, et al., 2013)

h. 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>.

```
1  <category>
2     <pattern> MY NAME IS * </pattern>
3     <template>
4        Hello <set name="nameUser"> <star/> </set>
5     </template>
6  </category>
```

Gambar 2.9 Penggunaan Tag <set> (Marietto, et al., 2013)

Tag <get> digunakan untuk mengembalikan nilai yang sudah disimpan pada <set>.

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

Gambar 2.10 Penggunaan Tag <get> (Marietto, et al., 2013)

i. Tag <that>

Tag <that> memungkinkan sistem untuk menganalisis kalimat terakhir dari respon *chatbot* sebelumnya. Percakapan dapat berjalan sesuai dengan konteks yang sedang dibicarakan. *Tag* ini harus berada di dalam *tag* <category>.

```
<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 2.11 Penggunaan Tag <that> (Marietto, et al., 2013)

j. Tag <topic>

Tag <topic> digunakan untuk menyatakan topik atau subjek pembicaraan *chatbot*. *Tag* <topic> ini mengelompokkan *tag category* sehingga mempercepat proses pencarian respon.

```
<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 2.12 Penggunaan Tag <topic> (Marietto, et al., 2013)

k. Tag <think>

Isi dari *tag* <think> merupakan proses yang dilakukan *chatbot* tapi tidak ditampilkan ke *user*. *Tag* ini digunakan untuk memproses data, *conditional statements* dan *tests*.

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

Gambar 2.13 Penggunaan Tag <think> (Marietto, et al., 2013)

l. Tag <condition>

Tag <condition> digunakan ketika beberapa respon yang mungkin diberikan kepada *user* dan pemilihan respon dilakukan berdasarkan analisis *value* dari suatu *variabel* yang berubah seiring dengan berjalannya proses *chatting*. *Tag* <condition> dapat dikatakan mempunyai nilai yang sama dengan *command* "Case" pada bahasa pemrograman.

```
<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 2.14 Penggunaan Tag <condition> (Marietto, et al., 2013)

m. Tag <bot>

AIML mengijinkan pengembang untuk mendefinisikan properti *chatbot* dan properti ini dapat dilihat oleh *user* saat percakapan berlangsung.

```
<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>
```

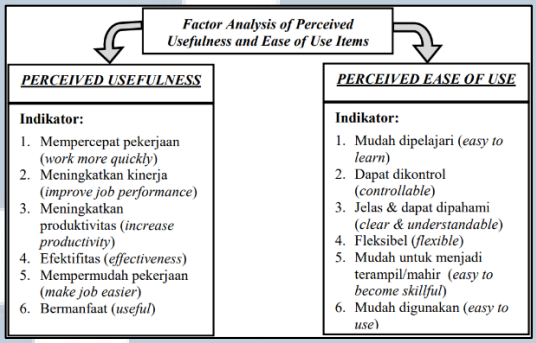
Gambar 2.15 Penggunaan Tag <bot> (Marietto, et al., 2013)

2.7 Technology Acceptance Model (TAM)

Pada tahun 1989 *Technology Acceptance Model* (TAM) pertama kali diperkenalkan oleh Davis sebagai pemodelan adopsi pengguna sistem informasi. Tujuan TAM adalah mendirikan dasar penelusuran pengaruh faktor ekternal terhadap kepercayaan, sikap, dan tujuan pengguna komputer (Wijayanti, 2009).

TAM menggunakan dua variabel persepsi ketika *user* akan menggunakan sistem informasi baru, yaitu *perceived ease of use* dan *perceived usefulness*. *Perceived ease of use* adalah sejauh mana orang percaya bahwa menggunakan

sistem tertentu akan bebas dari usaha. *Perceived usefulness* adalah sejauh mana orang percaya bahwa menggunakan sistem tertentu akan meningkatkan kinerjanya dan menggambarkan manfaat dari suatu sistem tersebut (Fatmawati, 2015).



Gambar 2.16 Factor Analysis of Perceived Usefulness and Ease of Use Questions (Davis, 1989)

2.8 Skala Likert

Skala likert adalah skala yang digunakan untuk sikap, pendapat, dan persepsi seseorang mengenai kejadian atau gejala sosial. Cara yang digunakan untuk mengukur adalah menghadapkan seorang responden dengan sebuah pertanyaan dan kemudian responden diminta untuk menjawab dari 5 pilihan jawaban dimana nilai jawaban memiliki nilai jawaban yang berbeda (Janti, 2014). Pilihan yang dapat responden pilih adalah Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), dan Sangat Tidak Setuju (STS).

Tabel 2.3 Tingkat Skala Likert

Pertanyaaan	Skor
Sangat Setuju	5
Setuju	4
Netral / Biasa Saja	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Menurut Sugiyono (2012), persentasi skor pada suatu kuisioner dihitung dengan menggunakan Rumus (2.1).

Persentase Skor = (((Sangat Setuju * 5) + (Setuju * 4) + (Netral * 3) + (Tidak Setuju * 2) + (Sangat Tidak Setuju * 1)) / (5 * Jumlah Responden)) * 100% ...(2.1)

2.9 Kateglo API

Kateglo adalah aplikasi dan layanan web sumber dan isi terbuka untuk kamus,tesaurus, dan glosarium bahasa Indonesia. Namanya diambil dari akronim unsur layanannya: ka(mus), te(saurus), dan glo(sarium). Data dari Pusat Bahasa Departemen Pendidikan Nasional Indonesia (Pusba) merupakan hak cipta dari Pusba dan dipergunakan di Kateglo dengan seizin Pusba (Kateglo, 2009).

Keterangan untuk mengakses Kateglo API adalah sebagai berikut.

Tabel 2.4 Tabel Keterangan Kateglo API

Keterangan	Nilai
URL Akses	http://kateglo.com/api.php?format=[xml|json]&phrase=[kata_yang_dicari]
Parameter	format=[xml|json]; phrase=[kata_yang_dicari]
Return Value	JSON atau XML

Gambar 2.17 dan Gambar 2.18 menggambarkan hasil *response* dari mengakses URL API Kateglo. Hasil yang menandakan kata dasar adalah ketika objek “root” memiliki hasil “[]” yang artinya sudah tidak ada kata dasar dari kata yang dicari dan ditunjukkan pada Gambar 2.18. Hasil yang menandakan bukan kata dasar adalah ketika object “root” meliki hasil “root_phrase” yang artinya ada kata dasar dari kata yang dicari dan ditunjukkan pada Gambar 2.17.


```
{"kategloc":
{"phrase":"meyakinkan","phrase_type":"d","lex_class":"v","roget_class":null,"pronounciation":null,"etymology":null,"ref_source":
"KBBI3","def_count":"4","actual_phrase":null,"info":null,"notes":null,"updated":"2009-06-14
08:42:26","updater":null,"created":"2009-06-14
08:42:26","creator":null,"proverb_updated":null,"wikipedia_updated":null,"kbbi_updated":"2009-06-14
08:42:26","lex_class_name":"Verba","roget_name":null,"ref_source_name":"KBBI III","lex_class_ref":"verba","root":
[{"root_phrase":"yakin","rel_type":"d"}],"definition":
[{"def_uid":"167091","phrase":"meyakinkan","def_num":"1","lex_class":null,"def_text":"menyaksikan sendiri supaya yakin;
memastikan","discipline":null,"sample":"supaya tidak salah mengambil putusan, baiklah kita ~ sendiri
kebenarannya","see":null,"updated":null,"updater":null,"wikipedia_updated":null,"lex_class_ref":null},
{"def_uid":"167092","phrase":"meyakinkan","def_num":"2","lex_class":null,"def_text":"menjadikan (menyebabkan dsb)
yakin","discipline":null,"sample":"ia berusaha ~ ayahnya bahwa uang itu benar didapat di jalan, bukan hasil
curian","see":null,"updated":null,"updater":null,"wikipedia_updated":null,"lex_class_ref":null},
{"def_uid":"167093","phrase":"meyakinkan","def_num":"3","lex_class":null,"def_text":"melakukan sesuatu dengan
sungguh","discipline":null,"sample":"~
pengajiannya","see":null,"updated":null,"updater":null,"wikipedia_updated":null,"lex_class_ref":null},
{"def_uid":"167094","phrase":"meyakinkan","def_num":"4","lex_class":"adj","def_text":"sungguh-sungguh (dapat dipercaya, dapat
diandalkan, dsb)","discipline":null,"sample":"bagaimana kita tidak terpikat, perkataannya begitu
~","see":null,"updated":null,"updater":null,"wikipedia_updated":null,"lex_class_ref":"adjektiva"}],"reference":[],"proverbs":
[],"translations":[{"lemma":"meyakinkan","ref_source":"ebsoft","translation":"1 convince s.o. 2 make convincing."},
{"lemma":"meyakinkan","ref_source":"gkamus","translation":"1 convince s.o. 2 make convincing."}],"relation":
{"relation_direct":3,"relation_reverse":7,"s":{"0":
{"rel_uid":"142341","root_phrase":"meyakinkan","related_phrase":"memastikan","rel_type":"s","updated":null,"updater":null,"rel_t
ype_name":"Sinonim","lex_class":"v"},"1":
{"rel_uid":"191313","root_phrase":"meyakinkan","related_phrase":"tepercaya","rel_type":"s","updated":null,"updater":"TESAURUS","
rel_type_name":"Sinonim","lex_class":"adj"},"name":"Sinonim","2":
{"root_phrase":"meyakinkan","related_phrase":"bonafide","rel_type":"s","rel_type_name":"Sinonim","lex_class":"adj"},"3":
{"root_phrase":"meyakinkan","related_phrase":"mencamkan","rel_type":"s","rel_type_name":"Sinonim","lex_class":"v"},"4":
{"root_phrase":"meyakinkan","related_phrase":"mengesahkan","rel_type":"s","rel_type_name":"Sinonim","lex_class":"v"},"5":
{"root_phrase":"meyakinkan","related_phrase":"mengabsahkan","rel_type":"s","rel_type_name":"Sinonim","lex_class":"v"},"6":
{"root_phrase":"meyakinkan","related_phrase":"aman","rel_type":"s","rel_type_name":"Sinonim","lex_class":"adj"},"7":
{"root_phrase":"meyakinkan","related_phrase":"andal","rel_type":"s","rel_type_name":"Sinonim","lex_class":"adj"}},"a":{"0":
{"rel_uid":"191314","root_phrase":"meyakinkan","related_phrase":"menyangsikan","rel_type":"a","updated":null,"updater":"TESAURUS
","rel_type_name":"Antonim","lex_class":"v"},"name":"Antonim","1":
{"root_phrase":"meyakinkan","related_phrase":"meragukan","rel_type":"a","rel_type_name":"Antonim","lex_class":"v"}},"r":
{"name":"Berkaitan"},"d":{"name":"Turunan"},"c":{"name":"Gabungan kata"},"pb":
{"name":"Peribahasa"},"relation_all":10},"all_relation":
[{"rel_uid":"142341","root_phrase":"meyakinkan","related_phrase":"memastikan","rel_type":"s","updated":null,"updater":null,"rel_
type_name":"Sinonim","lex_class":"v"},
{"rel_uid":"191313","root_phrase":"meyakinkan","related_phrase":"tepercaya","rel_type":"s","updated":null,"updater":"TESAURUS","
rel_type_name":"Sinonim","lex_class":"adj"},
{"rel_uid":"191314","root_phrase":"meyakinkan","related_phrase":"menyangsikan","rel_type":"a","updated":null,"updater":"TESAURUS
","rel_type_name":"Antonim","lex_class":"v"}]}}
```


Gambar 2.17 Response Kategloc API Bukan Kata Dasar


```
{"kategloc":
{"phrase":"yakin","phrase_type":"r","lex_class":"adj","roget_class":null,"pronounciation":null,"etymology":null,"ref_source":"KBBI3","def_count":"2","actual_phrase
":null,"info":null,"notes":null,"updated":"2009-06-14 08:42:26","updater":null,"created":"2009-06-14
08:42:26","creator":null,"proverb_updated":null,"wikipedia_updated":"2009-06-02 18:42:21","kbbi_updated":"2009-06-14
08:42:26","lex_class_name":"Adjektiva","roget_name":null,"ref_source_name":"KBBI III","lex_class_ref":"adjektiva","root":[],"definition":
[{"def_uid":"167087","phrase":"yakin","def_num":"1","lex_class":null,"def_text":"percaya (tahu, mengerti) sungguh-sungguh; (merasa) pasti (tentu, tidak salah
lagi)","discipline":null,"sample":"hakim -- akan kesalahan terdakwa itu; ia berkata dengan -- nya, berkata dengan pasti; pada -- ku, pada
pendapatku","see":null,"updated":null,"updater":null,"wikipedia_updated":null,"lex_class_ref":null},
{"def_uid":"167088","phrase":"yakin","def_num":"2","lex_class":null,"def_text":"sungguh; sungguh-sungguh","discipline":null,"sample":"-- bukan saya yang mengambil,
kalau perlu saya berani bersumpah; dengan -- belajar, belajar sungguh-
sungguh","see":null,"updated":null,"updater":null,"wikipedia_updated":null,"lex_class_ref":null}],"reference":[{"ext_uid":"1675","phrase":"yakin","label":"Yakin -
Wikipedia bahasa Indonesia","url":"http:\/\/id.wikipedia.org\/wiki\/Yakin","updated":"2009-06-02 18:42:21","updater":null}],"proverbs":[],"translations":
[{"lemma":"yakin","ref_source":"ebsoft","translation":"sure, certain, convinced."},{"lemma":"yakin","ref_source":"gkamus","translation":"sure, certain,
convinced."}],"relation":{"relation_direct":9,"relation_reverse":7,"s":{"0":
{"rel_uid":"142335","root_phrase":"yakin","related_phrase":"sungguh","rel_type":"s","updated":null,"updater":null,"rel_type_name":"Sinonim","lex_class":"adj"},"1":
{"rel_uid":"142336","root_phrase":"yakin","related_phrase":"sungguh-
sungguh","rel_type":"s","updated":null,"updater":null,"rel_type_name":"Sinonim","lex_class":"adv"},"2":
{"rel_uid":"208740","root_phrase":"yakin","related_phrase":"tetap","rel_type":"s","updated":null,"updater":"TESAURUS","rel_type_name":"Sinonim","lex_class":"v"},"n
ame":"Sinonim","3":{"root_phrase":"yakin","related_phrase":"positif","rel_type":"s","rel_type_name":"Sinonim","lex_class":"adj"},"4":
{"root_phrase":"yakin","related_phrase":"teguh","rel_type":"s","rel_type_name":"Sinonim","lex_class":"adj"},"5":
{"root_phrase":"yakin","related_phrase":"optimistis","rel_type":"s","rel_type_name":"Sinonim","lex_class":"adj"},"6":
{"root_phrase":"yakin","related_phrase":"berpengharapan","rel_type":"s","rel_type_name":"Sinonim","lex_class":"v"},"7":
{"root_phrase":"yakin","related_phrase":"percaya","rel_type":"s","rel_type_name":"Sinonim","lex_class":"v"}},"a":{"0":
{"rel_uid":"208741","root_phrase":"yakin","related_phrase":"ragu-
ragu","rel_type":"a","updated":null,"updater":"TESAURUS","rel_type_name":"Antonim","lex_class":"adj"},"name":"Antonim","1":
{"root_phrase":"yakin","related_phrase":"sangsi","rel_type":"a","rel_type_name":"Antonim","lex_class":"adj"},"2":
{"root_phrase":"yakin","related_phrase":"skeptis","rel_type":"a","rel_type_name":"Antonim","lex_class":"adj"}},"r":{"name":"Berkaitan"},"d":{"0":
{"rel_uid":"142345","root_phrase":"yakin","related_phrase":"berkeyakinan","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"v"},"
1":
{"rel_uid":"142342","root_phrase":"yakin","related_phrase":"keyakinan","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"n"},"2":
{"rel_uid":"142338","root_phrase":"yakin","related_phrase":"meyakin-
yakini","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"v"},"3":
{"rel_uid":"142337","root_phrase":"yakin","related_phrase":"meyakini","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"v"},"4":
{"rel_uid":"142340","root_phrase":"yakin","related_phrase":"meyakinkan","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"v"},"na
me":"Turunan"},"c":{"name":"Gabungan kata"},"pb":{"name":"Peribahasa"},"relation_all":16},"all_relation":
[{"rel_uid":"142335","root_phrase":"yakin","related_phrase":"sungguh","rel_type":"s","updated":null,"updater":null,"rel_type_name":"Sinonim","lex_class":"adj"},
{"rel_uid":"142336","root_phrase":"yakin","related_phrase":"sungguh-
sungguh","rel_type":"s","updated":null,"updater":null,"rel_type_name":"Sinonim","lex_class":"adv"},
{"rel_uid":"208740","root_phrase":"yakin","related_phrase":"tetap","rel_type":"s","updated":null,"updater":"TESAURUS","rel_type_name":"Sinonim","lex_class":"v"},
{"rel_uid":"208741","root_phrase":"yakin","related_phrase":"ragu-
ragu","rel_type":"a","updated":null,"updater":"TESAURUS","rel_type_name":"Antonim","lex_class":"adj"},
{"rel_uid":"142345","root_phrase":"yakin","related_phrase":"berkeyakinan","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"v"},
{"rel_uid":"142342","root_phrase":"yakin","related_phrase":"keyakinan","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"n"},
{"rel_uid":"142338","root_phrase":"yakin","related_phrase":"meyakin-
yakini","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"v"},
{"rel_uid":"142337","root_phrase":"yakin","related_phrase":"meyakini","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"v"},
{"rel_uid":"142340","root_phrase":"yakin","related_phrase":"meyakinkan","rel_type":"d","updated":null,"updater":null,"rel_type_name":"Turunan","lex_class":"v"}]}}
```


Gambar 2.18 Response Kategloc API Adalah Kata Dasar

2.10 Program O

Program O merupakan aplikasi *open source* dan dapat digunakan secara gratis. Program O menggunakan bahasa pemrograman PHP dengan *database* MySQL. Fungsi yang diberikan Program O mendukung aplikasi untuk membaca bahasa AIML (Program-O, 2008). Fungsi dari Program O yang digunakan dalam penelitian ini adalah yang terdapat pada Tabel 2.5. $msg pada fungsi processUpload dan parseAIML merupakan *return value* yang menandakan bahwa proses yang dilakukan berhasil atau tidak. $outFile pada fungsi getAIMLByFileName merupakan isi *string* dari data *file* AIML terpilih yang terdapat pada *database* Aplikasi Chatbot Penyedia Informasi Bengkel.

Tabel 2.5 Fungsi yang Digunakan pada Program O

Nama Fungsi	Manfaat	Parameter	Return Value
processUpload	Melakukan proses *uploading* dari *file* AIML input	$_FILES['aimlfile']	$msg
parseAIML	Membaca isi file .aiml untuk dimasukan ke dalam tabel aiml dan tabel srai_lookup	$fn, $aimlContent, $from_zip = false	$msg
download.php	Mendapatkan *file* AIML yang terdapat pada tabel aiml dan srai_lookup sampai menjadi sebuah file .zip	$_POST['type'], $_POST['filenames']	File AIML.zip

Tabel 2.6 Fungsi yang Digunakan pada Program O (Lanjutan)

getAIMLByFileName	Mendapatkan String berupa isi di dalam *file* AIML	$filename	$outFile
conversation_start.php	Melakukan proses pencocokan pola dan mendapatkan respon atas pertanyaan *user*	$_POST[‘convo_id’], $_POST[‘bot_id’], $_POST[‘say’], $_POST[‘format’], $_POST[‘debug_mode’], $_POST[‘debug_level’], $_POST[‘name’]	$convoArr