BAB II
STUDI LITERATUR

Pada bab ini berisi penelitian terkait dan landasan teori sebagai parameter rujukan untuk dilaksanakannya penelitian ini. Pada subbab penelitian terkait, berisi pembahasan penelitian terkait sebelumnya. Pada subbab landasan teori, berisi penjelasan teori terkait *chatbot, AIML* dan *CBR.*

II.1. Penelitian Terkait

Studi yang berkaitan dengan aplikasi *chatbot,* representasi pengetahuan *chatbot* dengan *Artificial Intelligence Markup Language (AIML)*, serta aplikasi yang menerapkan *Case Based Reasoning (CBR)*, dapat diperoleh dalam jurnal – jurnal ilmiah yang penjelasannya sebagai berikut :

a. Chatbot

Pada penelitian yang telah dilakukan oleh Benedictus, dkk [7], menghasilkan *chatbot* yang berfungsi untuk menjawab pertanyaan – pertanyaan seputar penggunaan aplikasi – aplikasi dalam Sistem Informasi Terpadu Universitas Sam Ratulangi [7]. *Chatbot* yang dikembangkan merupakan aplikasi berbasis *website*. Untuk pencocokan pola kalimat digunakan algoritma *bigram*, sedangkan untuk penalaran jawaban berdasarkan pertanyaan yang diberikan, menggunakan algoritma *forward chaining*. Selain itu, terdapat juga penelitian lainnya yang masih berhubungan dengan *chatbot* yang dikembangkan oleh Domarco dan Iswari [8], menghasilkan *chatbot* yang berfungsi sebagai media interaktif dalam mendapatkan informasi seputar *anime* berbasis teks [8]. Aplikasi berbasis *website* ini menggunakan POS *Tagging (Part of Speech Tagging)* dalam mengidentifikasi tata bahasa pada kalimat pertanyaan. Untuk menelusuri jawaban setiap pertanyaan, digunakan *regular expression pattern matching.*

b. Artificial Intelligence Modelling Language (AIML)

Pada pengembangan *chatbot,* terdapat beragam cara untuk merepresentasikan pengetahuan, salah satunya dengan *AIML*. Pada penelitian yang telah dilakukan Suryani dan Amalia [4] yang menerapkan *AIML* pada *chatbot* yang dikembangkan, dapat menghasilkanaplikasi *chatbot* berbasis *website* sebagai media informasi pariwisata Jawa Timur [4]. Dalam membangun *chatbot*, digunakan *AIML* untuk menyimpan pola pertanyaan dengan setiap pola tersebut memiliki jawabannya. Dalam mengelola pengetahuan *chatbot* yang meliputi pengubahan pengetahuan serta penambahan pengetahuan dari dokumen *AIML* eksternal, digunakan interpreter *Program-O*. Di akhir, penulis jurnal memberikan saran agar dapat membuat *chatbot* mampu belajar sendiri. Selain itu, terdapat penelitian lainnya yang membahas tentang *AIML* pada sebuah aplikasi *chatbot*, yang dilakukan oleh Mahdiyah dan Andriyani [3]. Pada penelitian tersebut, membahas bagaimana ALICE ChatBot dapat memahami kalimat pertanyaan dengan menggunakan algoritma *AIML* [3]. Dijelaskan bahwa *AIML* menyediakan *tag* bagi *chatbot* agar dapat melakukan proses *input output. AIML* juga menyediakan *tag* bagi *chatbot* agar dapat melalukan normalisasi kalimat *input*, yang nantinya dapat disesuaikan dengan *knowledge base. AIML* juga menyediakan *tag* jawaban untuk setiap *tag pattern* pertanyaan.

c. Case Based Reasoning (CBR)

Terdapat beberapa penilitian terkait penggunaan *CBR* terhadap pengembangan suatu aplikasi, salah satunya yaitu yang dilakukan oleh Kartikasari, dkk [9]. Pada penelitian tersebut, metode *CBR* digunakan untuk menyelesaikan masalah – masalah baru yang berkenaan dengan complain penyewa mall, dengan memanfaatkan masalah lama yang telah terselesaikan [9]. Dalam menemukan kemiripan antara masalah lama dengan masalah baru, digunakan algoritma *Nearest Neighbor.* Pada penelitian lainnya, Adawiyah [10] berhasil mengembangkan

aplikasi yang berguna untuk mendiagnosis penyakit demam berdarah [10]. Pada penelitian tersebut, membahas penggunaan *CBR* yang digunakan untuk memasukkan permasalahan baru lalu dibandingkan dengan kasus lama lalu dihitung nilai similaritansya. Untuk mendapatkan nilai similaritasnya, digunakan algoritma *Nearest Neighbor*.

Merujuk pada uraian di atas, terdapat poin – poin penting terkait penelitian sebelumnya sebagai pembanding dengan penelitian ini. Salah satu poin penting pada penelitian sebelumnya yaitu saran pada penelitian Suryani dan Amalia [4], bahwa kedepannya *chatbot* dapat belajar sendiri. Penulis pun mencoba untuk membangun *chatbot* dengan *AIML*, namun tetap bisa belajar sendiri dengan memanfaatkan metode *CBR*. Rangkuman penelitan terkait sebelumnya dapat di tuliskan dalam Tabel 2.1.

Tabel 2. 1. Rangkuman Penelitian Terkait.

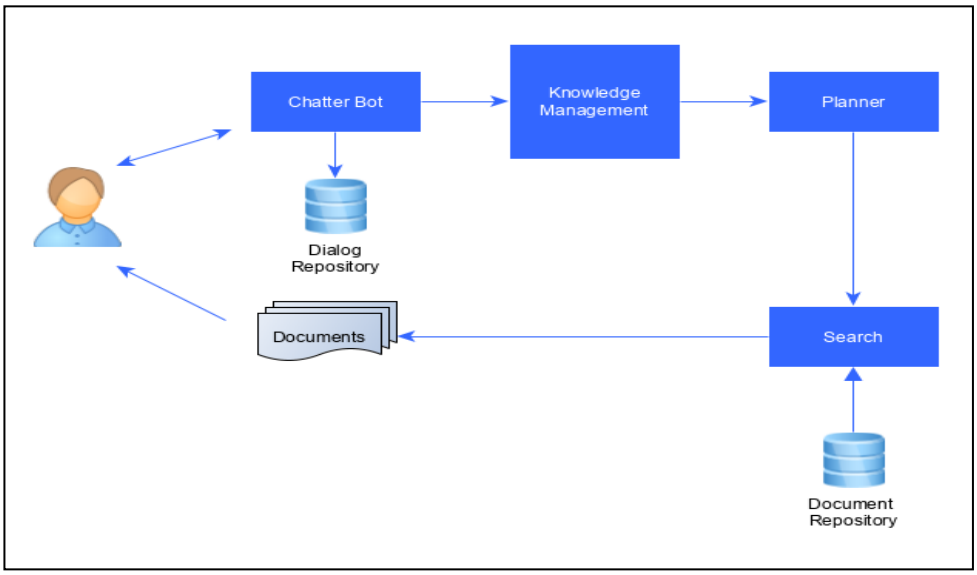
Nama Penulis	Tahun	Topik Penelitian	Hasil
Benedictus, dkk [7]	2017	Rancang Bangun *Chatbot* Helpdesk Universitas Sam Ratulangi	Aplikasi *chatbot* yang menjawab pertanyaan seputar Sistem Informasi Terpadu, dengan menggunakan algoritma *bigram* dan *forward chaining*
Iswari [8]	2017	Rancang Bangun Aplikasi *Chatbot* Sebagai Media Pencarian Anime	Aplikasi *chatbot* sebagai media dalam mendapatkan informasi seputar *anime* berbasis *website*, dengan menggunakan POS *Tagging* dan *regular expression pattern matching*
Suryani dan Amalia [4]	2017	Aplikasi *Chatbot* Objek Wisata Jawa	Aplikasi *chatbot* berbasis *website* sebagai media informasi pariwisata Jawa Timur,

Nama Penulis	Tahun	Topik Penelitian	Hasil
		Timur Berbasis *AIML*	menggunakan *AIML* sebagai representasi pola pertanyaan dan jawaban, dengan catatan diharapkan *chatbot* dapat belajar sendiri
Mahdiyah dan Andriyani [3]	2013	Analisa Algoritma Pemahaman Kalimat Pada ALICE ChatBot	*AIML* menyediakan *tag – tag* yang dapat digunakan untuk memetakan pola *input* dan *output*
Kartikasari, dkk [9]	2015	Penerapan *CBR* Pada Sistem Pendukung Keputusan Penangan Komplain Penyewa Mall	Metode *CBR* dapat menyelesaikan masalah – masalah baru dengan memanfaatkan masalah lama yang telah terselesaikan
Adawiyah [10]	2017	*Case Based Reasoning* Untuk Diagnosis Penyakit Demam Berdarah	Sebuah sistem dengan memanfaatkan metode *CBR* sebagai media untuk memasukkan permasalahan baru yang dibandingkan dengan kasus lama, lalu dihitung similaritasnya dengan menggunakan algoritma *nearest neighbor*

II.2. Landasan Teori

a. Chatbot

Definisi *chatbot* dapat dipahami berdasarkan komponen penyusun kata yaitu berasal dari kata *chat* dan *bot*. Kata *chat* dalam dunia komputer dapat diartikan sebagai obrolan yang menggunakan sarana tulisan. Sedangkan kata *bot* dapat diartikan sebagai program otomatis yang bila diberi sebuah *input* maka akan memberikan *output*. *Chatbot* adalah program komputer yang dapat melakukan percakapan dengan manusia atau *chatbot* lain dengan melalui media tulisan [4]. Teknologi *chatbot* pertama kali dimulai pada tahun 1966 oleh Joseph Weizenbaum [11], profesor MIT. Pada kala itu *chatbot* masih sangat sederhana. Tujuan pembuatan *chatbot* tersebut untuk meneliti apakah *user* dalam hal ini manusia tertipu mengira melakukan percakapan dengan manusia. Pengujian ini dikenal sebagai "Turing *Test*".



Gambar 2.1. Konsep *Chatbot* [1]

Chatbot adalah sebuah program komputer yang dapat melakukan percakapan dengan penggunanya mengenai sebuah topik, sesuai dengan pengetahuan yang dimiliki *chatbot* tersebut [7]. *Chatbot* yang telah dikembangkan memiliki pengetahuan atau kamus mengenai kata–kata apa saja yang digunakan oleh penggunanya dalam percakapannya

nanti. Setiap kata tersebut tentu memiliki respon – respon yang juga telah dimilliki *chatbot*. Secara sederhana, konsep *chatbot* dapat digambarkan pada Gambar 2.1.

Seperti yang telah dipaparkan sebelumnya, setiap *chatbot* memiliki basis pengetahuan mengenai topik tertentu sesuai dengan tujuan pembuatan *chatbot* tersebut. Sehingga *chatbot* memiliki ruang lingkup tertentu. Terdapat 2 macam ruang lingkup *chatbot* yaitu [7]:

1. *Open Domain*

   Pengguna *chatbot* dapat membahas berbagai macam topik ke *chatbot* ini. Jumlah topik yang dikuasai oleh *chatbot* ini tidak terbatas. Sehingga *chatbot* ini membutuhkan basis pengetahuna mengenai dunia yang luas, agar bisa mengenal topik yang sedang dibahas dan dapat memberikan respon yang sesuai.
2. *Close Domain*

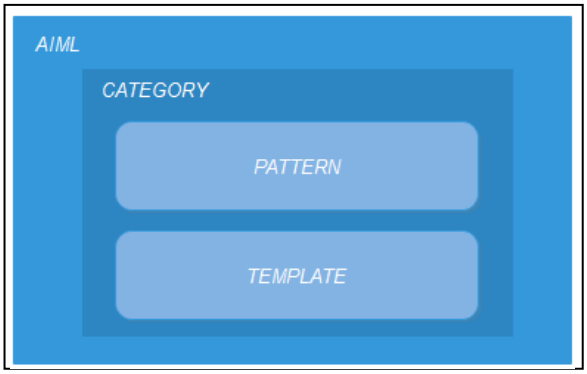
   Pengguna *chatbot* pada domain ini hanya dapat membahas topik tertentu saja. *Chatbot* dengan lingkup *close domain* telah dirancang hanya untuk membahas satu topik tertentu saja sesuai dengan tujuan pembuatan *chatbot* tersebut. Dalam penelitian ini, *chatbot* yang akan dikembangkan memiliki ruang lingkup *close domain* dengan topik yaitu informasi pengisian Data Induk Mahasiswa Institut Teknologi Sumatera (DIM ITERA).

b. Artificial Intelligence Markup Language

Artificial Intelligence Markup Language atau dapat disingkat *AIML* adalah bahasa *scripting interpreter* yang merupakan turunan dari *Extensible Markup Languange (XML)* dengan fungsi lebih spesifik [12]. Salah satu fungsinya adalah membuat sistem tanya-jawab (*stimulus-response)* berbasis pengetahuan. *AIML* mendeskripsikan objek data dan perilaku program komputer yang memprosesnya [2]. Objek *AIML* terdiri atas unit – unit yang disebut *topics* dan *categories*, berisi data yang telah terurai (*parsing*). Data yang ter-*parsing* berisi

karakter – karakter, beberapa diantaranya merupakan data karakter, sedangkan yang lainnya dapat berupa elemen *AIML*. Elemen *AIML* mengkapsulasi pengetahuan dalam bentuk *stimulus-response* di dokumen. Struktur *AIML* digambarkan pada Gambar 2.2.

Category merupakan unit dasar pengetahuan pada *AIML*, yang mana terdiri dari dua elemen yaitu *pattern* (*stimulus* atau pertanyaan) dan *template* (*response* atau jawaban). Gambar 2.3 merupakan contoh dari *categories*.



Gambar 2.2. Struktur Dokumen *AIML* [2]

```
1  <category>
2  <pattern>WHAT IS YOUR NAME</pattern>
3  <template>MY NAME IS MARTIN</template>
4  </category>
```

Gambar 2.3. Contoh *Categories*

Pattern adalah sebuah rangkaian huruf yang diharapkan dapat sesuai dengan satu atau bahkan lebih dari masukan pengguna. Sebuah *pattern* dapat menggunakan *wildcard* yang akan cocok dengan satu atau lebih masukan pengguna. Suatu *pattern* dapat dituliskan seperti pada baris 1 Gambar 2.3.

Pattern di atas cocok dengan pertanyaan, "*What is your name*", "*what is your father name*" dan sebagainya. *Template* menentukan respon dari *pattern* yang sesuai. Sebuah *template* bisa berupa teks harfiah, bisa juga dapat berupa teks variabel, seperti pada Gambar 2.4.

```
1  <template>MY NAME IS MARTIN</template>
2  <template>MY NAME IS <bot name="name"/></template>
```

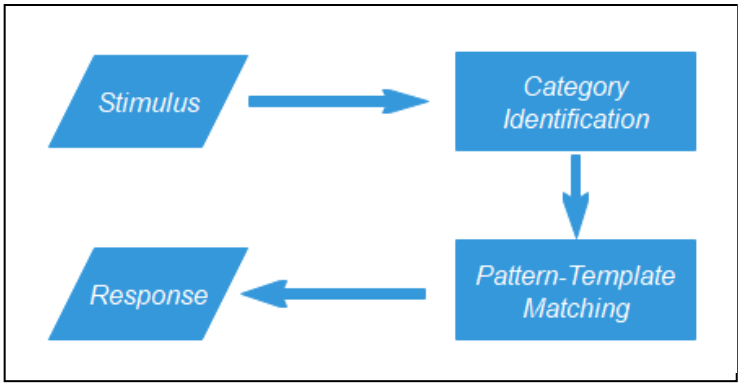
Gambar 2.4. Contoh Teks Variabel

Ada beberapa *tag* pada *AIML* yang tidak dimiliki *XML*, yang berguna untuk membantu menyusun pengetahuan *chatbot*. Salah satu contohnya adalah *tag* <*random*>. *Tag* <*random*> ini memungkinkan *chatbot* untuk memilih satu secara acak dari pilihan *response* yang disediakan. *Tag* <*random*> menggunakan *substag* <*li*> untuk menandai pilihan yan tersedia. Gambar 2. 5. Menunjukkan contoh penggunaan *tag* <*random*>.

```
278  <category><pattern>PURPOSE</pattern>
279  <template>
280  <random>
281  <li>I'm here to help you in any way I can.</li>
282  <li>I am a mobile virtual assistant.</li>
283  <li>I'm here to help.</li>
284  </random>
285  </template>
286  </category>
```

Gambar 2. 5. Contoh *Tag* <*Random*>

Pada *AIML* terdapat isitilah yang disebut dengan interpreter *AIML*. Interpreter *AIML* adalah perangkat lunak yang dapat membaca set *AIML*, mencocokkan i*nput user* dengan *category*¸ memproses isi *template* dalam *category* dan mengembalikannya kepada *user*[2]. Sebuah interpreter *AIML* dapat menggunakan *service XML processor* tetapi tidak boleh melanggar batasan – batasan yang sudah ditetapkan sebelumnya. Berikut ini gambaran alur kerja interpreter *AIML* pada Gambar 2. 6.



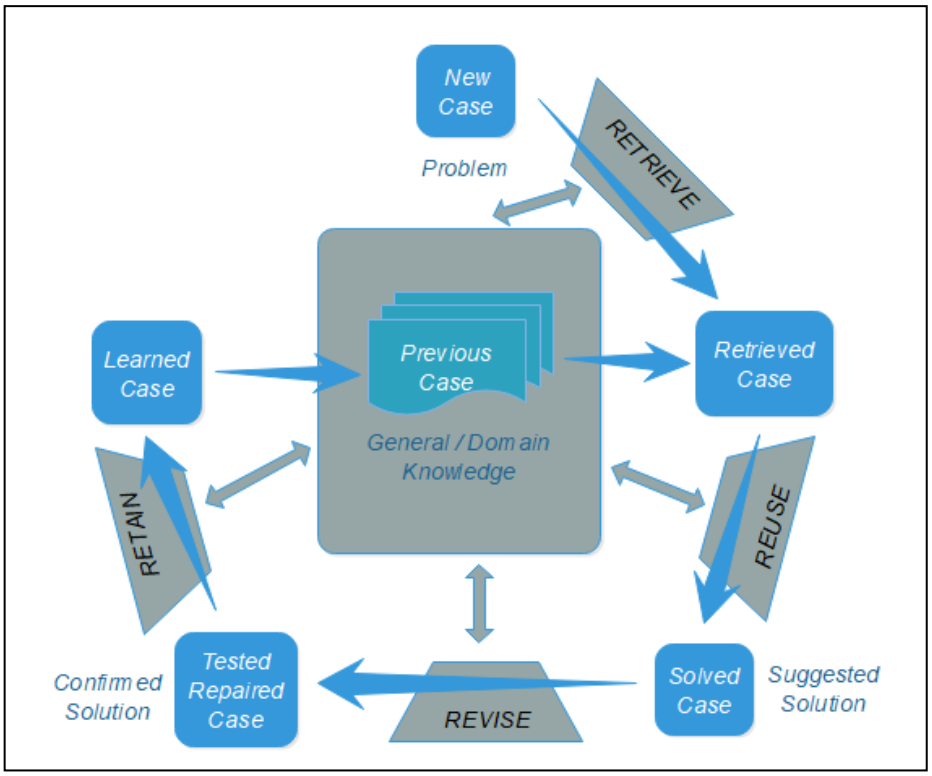
Gambar 2.6. Alur Kerja Interpreter AIML [2]

Pembuatan *chatbot* pada penelitian ini, menggunakan *AIML* sebagai representasi pengetahuan dari pertanyaan dan jawaban yang tersedia. *AIML* ini memiliki struktur yang sederhana, seperti yang telah digambarkan pada gambar 2.2. Pada penelitian Suryani dan Amalia [4], berhasil menghasilkan *chatbot* dengan *AIML* sebagai representasi pola pertanyaan dan jawaban, namun disarankan untuk memberikan kemampuan 'belajar sendiri' pada *chatbot* tersebut. Sehingga pada penilitian ini ingin mencari tahu bagaimana membuat *chatbot* dengan *AIML* sebagai representasi pola pertanyaan dan jawaban serta memiliki kemampuan 'belajar sendiri'. 'Belajar sendiri' yang dimaksudkan adalah kemampuan untuk menjawab pertanyaan baru diluar *pattern* yang telah didefinisikan pada dokumen *AIML*, serta memperbaiki jawaban yang dirasa masih kurang tepat dengan pertanyaan yang diberikan.

c. Case Based Reasoning

Case Based Reasoning atau bisa disebut *CBR*, merupakan salah satu metode yang mampu melakukan penalaran atau memecahkan permasalahan berdasarkan kasus yang telah ada sebagai solusi masalah baru [10]. Metode ini memungkinkan untuk menyelesaikan masalah dengan mengingat kejadian – kejadian yang sama atau sejenis yang pernah terjadi di masa lalu, kemudian menggunakan pengetahuan tersebut untuk menyelesaikan masalah yang baru. Dengan kata lain, *CBR* dapat menyelesaikan masalah dengan mengadopsi solusi – solusi

yang pernah dilakukan sebelumnya. Gambaran alur metode *CBR* ditampilkan pada Gambar 2. 7.



Gambar 2.7. Metode *CBR* [14]

1. *Retrieve*

   Pada proses ini didapatkan atau diperoleh kembali kasus yang paling menyerupai atau relevan dengan kasus yang baru. Tahap ini dimulai dengan menggambarkan atau menguraikan sebagian masalah dan diakhiri jika ditemukannya kecocokan terhadap masalah sebelumnya yang tingkat kecocokannya paling tinggi. Bagian ini mengacu pada segi identifikasi, kecocokan awal, pencarian dan pemilihan serta eksekusi.

2. *Reuse*

   Setelah tahapan *retrieve,* selanjutnya masuk ke tahapan pemodelan atau penggunaan kembali pengetahuan dan informasi kasus lama berdasarkan bobot kemiripan yang paling relevan ke dalam kasus

yang baru, sehingga menghasilkan usulan solusi dimana mungkin diperlukan suatu adaptasi dengan masalah yang baru tersebut.

3. *Revise*

   Setelah mendapatkan model sebagai hasil dari tahapan *reuse,* pada proses ini akan dilakukan peninjauan kembali terhadap solusi (model) yang diusulkan, kemudian diuji coba pada kasus nyata (simulasi). Jika diperlukan, maka solusi tersebut akan diperbaiki agar cocok dengan kasus yang baru.

4. *Retain*

   Pada tahap ini, terjadi proses integrasi yaitu proses untuk menyimpan kasus baru yang telah berhasil mendapatkan solusi, agar dapat digunakan oleh kasus – kasus selanjutnya yang mirip dengan kasus tersebut. Tetapi jika solusi baru tersebut gagal, maka menjelaskan kegagalannya, memperbaiki solusi yang digunakan dan mengujinya lagi.