

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Chatterbot (disebut juga *chatbot* atau *bots*) adalah sebuah program komputer yang dirancang untuk mensimulasikan percakapan intelektual dengan satu atau lebih manusia baik secara audio maupun teks. Pada mulanya, program komputer (*bots*) ini diuji melalui Turing Test, yaitu dengan merahasiakan identitasnya sebagai mesin sehingga dapat mengelabui orang yang bercakap-cakap dengannya. Jika pengguna tidak dapat mengidentifikasi *bots* sebagai suatu program komputer, maka *chatterbot* tersebut dikategorikan sebagai kecerdasan buatan (atau *Artificial Intelligence*). Dewasa ini, *chatterbot* 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 (atau *conversational agent*). Yang membedakan *chatterbot* dengan sistem pemrosesan bahasa alami (atau *Natural Language Programming*) adalah kesederhanaan algoritma yang digunakan.

Meskipun banyak *chatbot* yang tampaknya dapat mengartikan dan menanggapi *input* manusia, sebenarnya *chatbot* 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.

Penelitian tentang penggunaan chatbot pernah dilakukan oleh Balbir Singh Bani dan Ajay Pratap Singh yaitu penggunaan *chatbot* A.L.I.C.E untuk memudahkan mahasiswa baru yang sedang mencari informasi pada saat proses admisi, dengan adanya *chatbot* ini mahasiswa baru dapat dengan mudah mendapatkan segala informasi yang berkaitan dengan jurusan atau program studi yang dipilih.

Penelitian lain tentang penggunaan *chatbot* juga pernah dilakukan oleh Mr. Sathis Kumar .T, N. Vijay Kumar, R. R. Vinodh, U. Vinoth Kumar dan T. Vivekananthan dari Sarathan College of Engineering, India. Pada penelitian tersebut *chatbot* yang dibangun digunakan untuk membantu mahasiswa untuk me-update informasi tentang kegiatan kuliah dengan cara mahasiswa bertanya melalui *bot* yang digunakan untuk mengobrol, mahasiswa dapat mengobrol menggunakan format apapun, tidak ada format khusus yang harus diikuti oleh pengguna.

Adapun ringkasan dari perbandingan dengan peneliti sebelumnya bisa dilihat pada tabel 2.1.

Tabel 2.1 Daftar Penelitian Sebelumnya

Penulis / Parameter	Judul	Metode	Bahasa Pemrograman	Interface
Bahartyan, Egga, Bahtiar, Nurdin, Waspada, Indra (2015)	Integrasi <i>Chatbot</i> Berbasis AIML Pada Website	Ripple	PHP	GUI

Parameter Penulis	Judul	Metode	Bahasa Pemrograman	Interface
	<i>E-Commerce</i> sebagai <i>Virtual Assistant</i> Dalam Pencarian Dan Pemesanan Produk (Studi Kasus Toko Buku <i>Online</i> edu4indo.mom)			
Hormansyah, Dhebys Suryani, Utama, Yoga Putera (2018)	Aplikasi <i>Chatbot</i> Berbasis Web Pada Sistem Informasi Layanan Publik Kesehatan Di Malang Dengan Menggunakan Metode TF-IDF	TF-IDF	PHP	GUI
Maskur, Maskur (2016)	Perancangan <i>Chatbot</i> Pusat Informasi Mahasiswa Menggunakan AIML Sebagai <i>Virtual Assistant</i> Berbasis Web	TF-IDF	PHP	GUI
Jati, Dimas Pratama, Maarif, Muhammad Rifqi (2018)	The Development Of <i>Chatbot</i> Application On Line	NLP (<i>Natural Language Processing</i>)	PHP	GUI

Parameter Penulis	Judul	Metode	Bahasa Pemrograman	Interface
	Messaging Platform For Customer Service In Jogja Sewa Kamera			
Balbir Singh Bani, Ajay Pratap Singh (2017)	College Enquiry Chatbot Using A.L.I.C.E	Pattern Matching	AIML	Teks
Mr. Sathis Kumar .T, N. Vijay Kumar, R. R. Vinodh, U. Vinoth Kumar dan T. Vivekananthan (2018)	Chatbot for College Related Queries	Decision Trees	Python, MongoDB	Teks

2.2 Dasar Teori

Chatbot HiBuddy yang dibuat merupakan aplikasi berbasis *command line* sehingga aplikasi chatbot ini tidak memiliki tampilan, dan aplikasi *chatbot* ini bertipe *Question Answering*, sehingga bantuan yang ditawarkan adalah menjawab pertanyaan *User*. *Chatbot* ini menggunakan teknik *Regular Expression* sebagai algoritma untuk *pattern matching*, yaitu melakukan pencarian dengan mencocokkan pola yang diberikan dengan tabel indeks (teks) yang tersedia.

Chatbot yang dibuat merupakan *chatbot* dengan pengetahuan yang bersifat *closed domain*. Proses pengembangan *software* menggunakan metode *Prototyping* dan pengujian menggunakan metode *Black Box*.

Untuk mendukung penelitian ini, maka perlu dikemukakan hal-hal atau

teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai referensi.

2.2.1 Chatbot

Chatbot merupakan program komputer yang berinteraksi dengan user menggunakan bahasa natural (*Natural Language Processing*). Teknologi *chatbot* pertama dimulai pada tahun 1960an. Tujuan pembuatan *chatbot* ini adalah pengujian apakah *chatbot* dapat menipu *user* agar mereka mengira sedang berkomunikasi dengan manusia. Pengujian ini dikenal dengan nama "*Turing Test*" (Mauldin, 1994).

Chatbot adalah sebuah program yang dapat melakukan obrolan dalam bahasa alami mengenai sebuah topik yang ada dalam model pengetahuan *chatbot* tersebut. Artinya, *chatbot* harus bisa mengenali dan merespon kata-kata yang diberikan *user*. Kemampuan *chatbot* dalam mengenali dan memberikan respon ditentukan oleh ruang lingkup dari pengetahuan *chatbot*. Terdapat 2 macam ruang lingkup dari *chatbot*, yaitu:

1. *Open Domain*. Pengguna dapat membawa arah percakapan kemana saja. Jumlah topik yang dibahas tidak terbatas. Dibutuhkan pengetahuan tentang dunia luas dalam basis pengetahuan agar bisa mengenali dan memberikan respon terhadap masukan *user*.
2. *Close Domain*. *Chatbot* dirancang untuk memenuhi tujuan yang

spesifik. Basis pengetahuan *chatbot* hanya mencakup topik seputar tujuan *chatbot*. Dalam penelitian ini, dibuat *chatbot* bertipe *close domain* sehingga *chatbot* hanya bisa mengenali dan merespon masukan seputar tipe-tipe pesan yang bisa diproses oleh aplikasi pesan HiApp.

2.2.2 Regular Expression

Regular expression merupakan notasi yang digunakan untuk mendeskripsikan himpunan karakter *string*. Sebagai contoh, himpunan yang terdiri dari string “*Handel*”, “*Händel*”, dan “*Haendel*” dapat dideskripsikan dengan “*H(ä|ae?)ndel*” (atau alternatif lainnya, notasi ini menunjukkan bahwa notasi tersebut sesuai dengan string string yang diberikan) (Thompson, 1968).

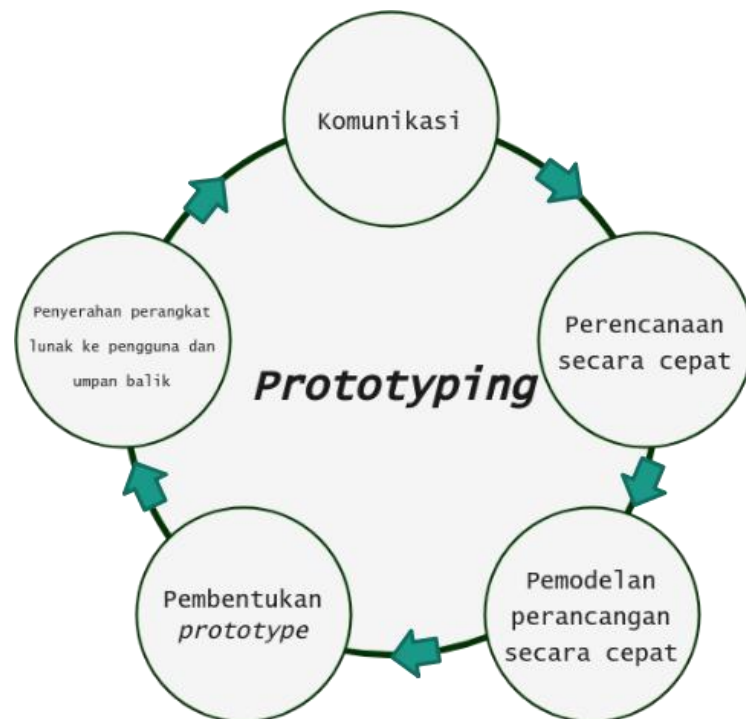
Regular expression didefinisikan berdasarkan aturan teori bahasa formal. *Regular expression* terdiri dari konstanta dan operator yang menunjukkan himpunan-himpunan string dan operasi antara himpunan *string* tersebut secara berurutan.

2.2.3 Pattern Matching

Pattern matching dalam ilmu komputer adalah kegiatan pemeriksaan serangkaian *string* atau karakter yang ada untuk menemukan pola yang konstituen antar *string* (Hak & Dul, 2008). Salah satu teknik dalam *pattern matching* adalah *regular expression*.

2.2.4 Prototyping

Prototyping merupakan sebuah paradigma pengembangan perangkat lunak yang menawarkan pendekatan paling baik dan membantu pengembang dan *stakeholder* untuk memahami lebih baik apa yang akan dikembangkan saat spesifikasi kebutuhan belum jelas (Pressman, 2012). Spesifikasi kebutuhan yang belum jelas biasanya terjadi ketika *stakeholder* maupun pengembang masih belum bisa mengidentifikasi secara detail spesifikasi yang rinci untuk fungsi-fungsi dan fitur-fitur, algoritma yang akan digunakan, maupun komponen-komponen lainnya seperti desain tampilan dan spesifikasi sistem.



Gambar 2.1 Metode *Prototype*

Metode *prototyping* yang diterapkan dalam pengembangan sebuah

perangkat lunak terdiri dari 5 (lima) tahap, yaitu Tahap Komunikasi, Tahap Perencanaan secara cepat, Tahap Pemodelan Perancangan secara cepat, Tahap Pembentukan *Prototype*, dan Tahap Penyerahan Perangkat Lunak ke Pengguna.

1. Tahap komunikasi

Pada tahap ini dikumpulkan data melalui diskusi tentang fitur apa saja pada aplikasi chatting HiApp yang bisa diterima, dan bagaimana format pesan yang bisa diterima sehingga pesan yang dikirimkan melalui chatbot Hi Buddy bisa diproses oleh aplikasi HiApp. Pada tahap ini juga didiskusikan format seperti apa yang akan diproses oleh chatbot sehingga aplikasi HiApp bisa memanfaatkan chatbot untuk kebutuhan testing.

2. Tahap perencanaan secara cepat

Pada tahap ini dilakukan perancangan secara cepat tentang alur kerja chatbot berdasarkan informasi yang sudah didapatkan dan disepakati pada tahap sebelumnya.

3. Tahap pemodelan perancangan secara cepat

Pada tahap ini dilakukan pemodelan perangkat lunak berdasarkan hasil desain dari tahap dua. Pemodelan tersebut mencakup Use Case Diagram, Flowchart dan Activity Diagram.

4. Tahap pembentukan *prototype*

Pada tahap ini hasil pemodelan dituangkan dalam kode sumber aplikasi.

5. Tahap penyerahan aplikasi (*deployment*)

Prototype yang berhasil dibangun akan diinstall di server yang sudah disediakan, dalam hal ini implementasi docker image ke AWS ECS sehingga *chatbot* bisa berjalan dan bisa berinteraksi dengan user melalui aplikasi HiApp. *User* akan memberikan umpan balik mengenai kekurangan aplikasi. Perbaikan *chatbot* dilakukan berdasarkan hasil umpan balik dari *user*.

2.2.5 Black Box

Berdasarkan IEEE (1990), definisi *Black Box Testing* adalah (1) proses pengujian dimana mekanisme internal dari sebuah komponen atau sistem diabaikan dan berfokus kepada kondisi eksekusi serta nilai keluaran yang dihasilkan sebagai respon terhadap nilai input yang dipilih. (2) Proses pengujian yang dilakukan untuk mengevaluasi pemenuhan sistem atau komponen dengan kebutuhan fungsional tertentu (IEEE, 1991). Kelebihan dari pengujian *black box* adalah:

1. Penguji tidak memerlukan pengetahuan akan bahasa pemrograman yang digunakan.
2. Membantu menemukan ambiguitas dan ketidakkonsistenan yang terjadi antara penguji dan pengembang.

2.2.6 Message Stanza

Extensible Messaging and Presence Protocol (XMPP) adalah protokol

untuk streaming elemen XML untuk bertukar pesan dan informasi keberadaan dalam waktu yang hampir bersamaan. XMPP Server dan client berkomunikasi melalui pengiriman Message Stanza, yaitu kode XML yang fleksibel (sesuai spesifikasi) yang diencode menggunakan Base64 sebelum dikirimkan. Message Stanza yang dikirimkan disebut packet yang bisa berupa *Message*, *Presence* dan IQ (xmpp.org).

Beberapa contoh sederhana dari XML Stanza:

```
<iq to='example.com'  
  type='set'  
  id='sess_1'>  
  <session  
xmlns='urn:ietf:params:xml:ns:xmpp-session' />  
</iq>
```

Gambar 2.2 Perintah *Request* Client ke Server

```
<iq from='example.com'  
  type='result'  
  id='sess_1' />
```

Gambar 2.3 *Response* Server ke Client

```
<presence xml:lang='en'>  
  <show>dnd</show>  
  <status>Menunggu waktu Sholat</status>  
</presence>
```

Gambar 2.4 *Presence*

```
<message
  to='bejo@example.net'
  from='susi@example.com/balcony'
  type='chat'
  xml:lang='en'>
  <body>Dimana kamu, Bejo?</body>
</message>
```

Gambar 2.5 *Message*

2.2.7 Elixir Language

Elixir adalah *General Purpose Programming Language*, dengan pemrosesan *concurrent* dan fungsional. Elixir adalah Bahasa pemrograman yang sangat powerfull, dan sangat mudah untuk penulisannya, dibuat oleh José Valim yang berjalan pada mesin virtual Erlang yaitu (BEAM). Elixir dibangun diatas bahasa pemrograman yang bernama Erlang, yang dikembangkan oleh perusahaan raksasa yang bernama Ericsson di Denmark. (Fachrizal Rifahdi, 2018)

2.2.8 Docker

Dalam membangun program, pengembang biasanya menjalankan virtualisasi pada server sehingga proses pembuatan program dapat berjalan diberbagai platform maupun konfigurasi hardware. Masalah yang dihadapi dengan virtualisasi adalah perlunya menyiapkan satu sistem operasi secara utuh, termasuk berbagai aplikasi yang dibawa sistem tersebut. Bisa dibayangkan dengan banyaknya virtualisasi yang berjalan di sebuah server akan memberatkan sistem tersebut.

Container kemudian datang dan membawa beberapa perubahan. Dengan

container, sebuah program ‘diikat’ beserta *library*-nya, file konfigurasi, dan seluruh hal yang dibutuhkannya. Perbedaan yang sangat terlihat dibandingkan dengan virtualisasi adalah container memiliki ukuran file yang jauh lebih kecil karena tidak perlu menyiapkan sistem operasi secara penuh. Dalam hal ini, pengembang biasa menyebutnya sebagai ‘*lightweight*’ platform. Aplikasi yang berjalan menggunakan container pun jauh lebih cepat dan lebih efisien.

Docker adalah salah satu platform yang dibangun berdasarkan teknologi container. Docker merupakan sebuah project open-source yang menyediakan platform terbuka untuk developer maupun sysadmin untuk dapat membangun, mengemas, dan menjalankan aplikasi dimanapun sebagai sebuah wadah (container) yang ringan. Dengan sangat populernya docker, sebagian orang sering menganggap docker adalah sebutan lain untuk container.

2.2.9 Amazon Web Service (AWS)

Amazon Web Services (AWS) adalah penyedia layanan cloud yang aman, AWS menawarkan tenaga komputasi, ruang penyimpanan database, "*content delivery network*" dan fungsionalitas lainnya yang membantu banyak bisnis untuk berkembang dan menjalankan aplikasi dengan baik.

AWS memberikan banyak pilihan produk yang sangat memudahkan dalam membangun bisnis. *Infrastructure as a service*, adalah istilah yang tepat untuk AWS. Istilah lainnya ada juga *Software as a service* dan *Platform as a service*.

Amazon Elastic Container Service (Amazon ECS) adalah layanan AWS dalam hal orkestrasi paket berkinerja tinggi dan mendukung paket Docker. Dengan layanan ECS dari AWS dapat mengurangi proses setup server dan instalasi sistem operasi yang akan digunakan untuk menjalankan *ChatBot* ini.