



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 Chatting Robot

Chatting Robot (Chatbot) adalah sebuah *program* yang memungkinkan manusia dapat berinteraksi atau berdialog dengan mesin (komputer) dengan menggunakan bahasa alami. *Chatbot* akan memindai kata kunci (*keyword*) dalam input manusia 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 (Ribalta, 2014). Hal ini memberikan *chatbot* ruang untuk berkembang dalam fungsi-fungsi sistem interaktif dalam memberikan informasi pada pengguna.

2.1.1 Tipe Percakapan Chatting Robot

Menurut seorang profesor Ilmu Komputer dan *Co-Founder* Narrative Science, Kris Hammond, terdapat beberapa tipe percakapan dalam *Chatbot*. Berikut tipe-tipe percakapan dalam *Chatbot* (Hammond, 2016):

1) *Search*

Setelah pencarian selesai dan menampilkan respon, sistem akan melupakan apa yang baru dikatakan kepada pengguna. Pada tipe ini, interaksi yang diberikan terbatas. Sistem tidak dapat melakukan tugas-tugas yang lebih kompleks, hanya dapat menampilkan informasi kemudian selesai.

2) *Chat as Chat*

Pada tipe ini, sistem *chat* lebih terfokus untuk hiburan. Sistem ini menghasilkan tanggapan didasarkan pada model statistik dari seberapa relevan tanggapan sistem dari input teks pengguna. Pada akhirnya, sistem tidak tahu apa

yang pengguna katakan atau yang sistem katakan. Sistem hanya tahu bahwa mereka membuat hubungan yang baik dengan penggunanya.

3) *Triggered Task Model*

Input yang dimasukkan pengguna akan diproses untuk memanggil fungsi lain untuk melakukan suatu tugas berikutnya. Dibutuhkan semua pilihan yang memungkinkan informasi lain akan muncul atau timbul dan kemudian menggunakannya untuk mengelola interaksi berikutnya.

4) *Complex Task Interactions*

Sistem memiliki pengetahuan tentang tugas, informasi yang dibutuhkan untuk melakukan sesuatu, dan kemampuan untuk melacak informasi pengguna. Pengetahuan tersebut akan digunakan untuk membantu pengguna dan kemudian menggunakan pengetahuan tersebut untuk mendukung percakapan.

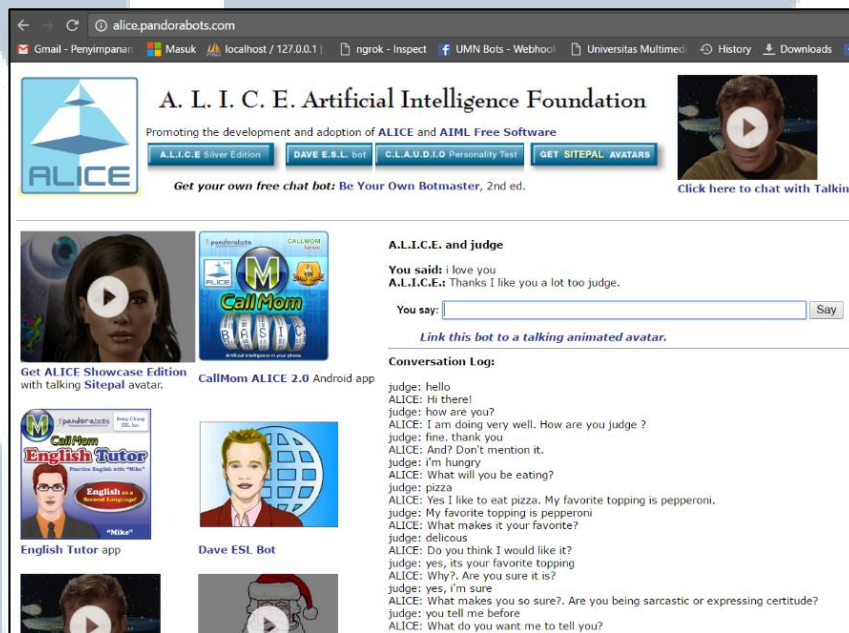
5) *Interactive Information Access*

Pada model *interactive information access*, sistem berusaha untuk memahami penggunanya dengan memiliki akses informasi mengenai kehidupan penggunanya. Sistem dapat melakukan interaksi percakapan dengan memetakan data ke makna dan bahasa untuk menghasilkan narasi. Data yang dimasukkan pengguna akan dianalisis. Kemudian hasil analisis digabungkan untuk menentukan fakta-fakta yang didefinisikan oleh data dengan meregenerasi bahasa alami untuk menghasilkan interaksi yang lebih manusiawi. Sistem benar-benar tahu apa yang pengguna tanyakan.

2.1.2 **Perkembangan Chatting Robot**

Chatbot pertama kali diperkenalkan oleh Profesor Joseph Weizenbaum di MIT (*Massachusetts Institute of Technology*), yang bernama ELIZA, pada tahun

1964 sampai 1966 (Norman, 2017). ELIZA dibangun dengan tujuan untuk mempelajari komunikasi *natural language* antara manusia dengan mesin. ELIZA bertindak seolah-olah seorang dokter dengan mempelajari laporan pasien sebagai pertanyaan dan menjawab pertanyaan dari pasien (Chatbots Org, 2017). Selain ELIZA, terdapat *chatbot* lainnya seperti ALICE. ALICE (*Artificial Linguistic Internet Computer Entity*) yang diciptakan oleh Dr. Richard Wallace. ALICE dapat menciptakan suasana seperti percakapan antar dua manusia seperti biasanya, bahkan ALICE dapat berbohong, menyebarkan gosip, dan mengetahui banyak kejadian yang terjadi di dunia nyata.



Gambar 2.1 Contoh *Chatting Robot* ALICE (Alice Artificial Intelligence Foundation, 2001)

Aplikasi dirancang untuk memberikan jawaban paling akurat dari semua pertanyaan yang disampaikan oleh pengguna. Tidak hanya sebatas teman berbicara, aplikasi ini dapat membantu hubungan antara perusahaan dengan konsumen. Selain itu, *chatbot* juga diimplementasikan untuk bidang komersial, pendidikan,

entertainment, dan sektor pelayanan publik (Kerly, Hall dan Bull, 2006). Berikut, contoh pengimplementasian *chatbot*:

1. Sebagai pendamping berbelanja
2. Sebagai *customer service*
3. Sebagai asisten pribadi
4. Sebagai dokter pribadi
5. Sebagai pembaca berita pribadi
6. Sebagai guru pribadi

2.1.3 Penggunaan Brain File

Brain file merupakan otak atau pola dari *chatbot* yang menentukan bagaimana cara *chatbot* berpikir dan memberikan respon (Ribalta, 2014). Semua kosakata, kata kunci, kepribadian dan pengetahuan akan disimpan di *brain file* sebagai tabel informasi (*knowledge base*). Biasanya *brain file* berupa teks. Semakin banyak pengetahuan yang dimiliki *chatbot* maka akan semakin besar ukuran *file*.

2.2 String Matching

String Matching merupakan suatu algoritma pencocokkan *string* untuk memecahkan masalah pencocokkan suatu teks terhadap teks lain. Masalah utama dalam pencocokkan *string* adalah untuk menemukan semua kejadian (atau kejadian pertama) dari sebuah pola (*pattern*) dalam sebuah teks, dimana pola dan teksnya berupa *string* yang terdiri dari beberapa alfabet (He, Fang, dan Sui, 2004). Berdasarkan arah pencariannya, algoritma *string matching* mempunyai tiga cara. Algoritma yang menggunakan arah dari kiri ke kanan adalah Algoritma Brute Force dan Algoritma Knuth Morris Pratt. Kemudian cara dari arah kanan ke kiri dengan menggunakan Algoritma Boyer Moore, yang secara praktiknya merupakan hasil

terbaik (Sagita dan Prasetyowati, 2013). Dan secara teoritis, menggunakan arah yang ditentukan oleh algoritma itu sendiri, ada Algoritma Colussi dan Algoritma Crochemore-Perrin.

Dalam membangun *chatting robot* dibutuhkan metode pencocokkan *string* (*string matching*). Pertanyaan yang masuk akan dilakukan proses pencocokkan *string* dengan kata kunci. Jika kata kunci ditemukan akan dilakukan proses *reasoning* sebagai bentuk respon. Untuk implementasinya, proses pencocokkan kata kunci digunakan Algoritma Boyer-Moore.

2.2.1 Algoritma Boyer-Moore

Algoritma Boyer-Moore diperkenalkan oleh R.M. Boyer dan J.S. Moore pada tahun 1977. Algoritma Boyer-Moore merupakan salah satu algoritma yang paling efisien untuk pencarian *string matching* dibandingkan Algoritma Knuth-Morris-Pratt (KMP), dan Algoritma Rabin-Karp (Saputra dan Rahman, 2013). Algoritma Boyer-Moore memiliki waktu pencarian tercepat daripada Algoritma Turbo Boyer-Moore dan Algoritma Tuned Boyer-Moore (Sagita dan Prasetyowati, 2013). Rabin-Karp memiliki waktu pencarian *string* 3.46 detik, KMP 0.99 detik, Brute Force 0.98 detik dan Boyer Moore 0.92 detik (Utomo, Harjo, dan Handoko, 2008).

2.2.2 Cara Kerja Algoritma Boyer Moore

Sederhananya Algoritma Boyer-Moore melakukan pencocokkan *string* atau karakter dari arah kanan ke kiri (Boyer dan Moore, 1977). Jika karakter tidak cocok ada dua kondisi, yaitu:

1. Jika karakter pada teks tidak terdapat pada *pattern*, *pattern* akan bergeser ke kanan sejauh panjang *pattern* (pergeseran *window*).

2. Jika karakter pada teks terdapat pada *pattern*, *pattern* akan bergeser sebanyak satu karakter ke kanan.

Jumlah pergeseran *pattern* akan ditentukan dengan menggunakan dua cara yaitu *BadCharacter* dan *GoodSuffixes*. Dari hasil hitung *BadCharacter* dan *GoodSuffixes* akan dibandingkan untuk diambil nilai paling besar. Kemudian lakukan pergeseran *window* menggunakan nilai tersebut (Sagita dan Prasetyowati, 2012). Arah yang digunakan untuk pergeseran *window* adalah dari arah kiri ke kanan (Argakusumah dan Hansun, 2014). Jika karakter pada teks memiliki kesamaan, pemeriksaan akan bergeser ke kiri dan melakukan pencocokkan pada karakter berikutnya. Lakukan langkah dua sampai paling kiri dari *pattern*.

2.2.3 *BadCharacter* dan *GoodSuffixes*

Tabel *BadCharacter Shift (Occurance Heuristic)* digunakan untuk menyimpan nilai-nilai pergeseran berdasarkan karakter apa yang menyebabkan ketidakcocokan dan seberapa jauh karakter tersebut dari karakter paling akhir (Argakusumah dan Hansun, 2014). Setiap karakter yang ada di *pattern* diberi nilai sesuai dengan ukuran jauhnya karakter tersebut dari karakter paling terakhir dari *pattern*. Karakter yang tidak terdapat pada *pattern* akan diberikan nilai sama dengan jumlah panjang dari *pattern*.

Tabel *GoodSuffixes Shift (Match Heuristic)* digunakan untuk menyimpan nilai-nilai pergeseran ketika ketidakcocokan ditemukan berdasarkan karakter pada posisi keberapa yang menyebabkan ketidakcocokan (Sagita dan Prasetyowati, 2012). Tabel *Suffix* digunakan untuk menyimpan hasil hitung banyaknya perulangan pada suatu karakter. Tabel *GoodSuffixes* ditentukan melalui tabel *suffix*. Semakin banyak perulangan, semakin kecil nilai pergeseran.

Berikut contoh cara kerja Algoritma Boyer-Moore:

Teks: apa syarat untuk mendaftar

Pattern: daftar

	A	P	A	S	Y	A	R	A	T	U	N	T	U	K	M	E	N	D	A	F	T	A	R		
1	D	A	F	T	A	R																			
2						D	A	F	T	A	R														
3											D	A	F	T	A	R									
4																D	A	F	T	A	R				
5																				D	A	F	T	A	R

Gambar 2.2 Contoh Pencocokkan *String* Algoritma Boyer-Moore (Jaffray, 2013)

Saat pencocokkan karakter 'R' dan 'Y', karena karakter 'Y' tidak memiliki kecocokan dengan karakter pada *pattern*, nilai pergeseran menggunakan BadCharacter (Kasus 1). Pencocokkan karakter 'R' dan 'U', juga menggunakan pergeseran BadCharacter (Kasus 2). Kemudian pencocokkan karakter 'R' dan 'M', karakter 'M' juga tidak terkandung dalam *pattern*, geser menggunakan nilai *BadCharacter* (Kasus 3). Pencocokan karakter 'R' dengan 'T' menggunakan nilai pergeseran GoodSuffixes, dimana *pattern* memiliki karakter 'T' (Kasus 4). Setelah melakukan pergeseran, cocokkan karakter 'R' pada *pattern* dengan karakter 'R' pada teks. Jika karakter pada teks sama dengan karakter pada *pattern*, lakukan pencocokan karakter sebelah kiri dari karakter paling terakhir (karakter A). Setelah semua karakter pada *pattern* diperiksa dan cocok, *pattern* ditemukan dalam teks (Kasus 5).

2.3 Facebook Messenger

Facebook merupakan sebuah layanan jejaring sosial yang dapat menghubungkan pengguna dengan pengguna lainnya, dengan menggunakan jaringan internet sebagai penghubungnya. Misi Facebook adalah memberikan penggunanya kekuatan untuk membangun komunitas dan mendekatkan dunia (Facebook, 2017). Pengguna dapat mengubah profil pribadi, menambahkan pengguna lain sebagai teman, mengirimkan pesan, dan berbagi informasi baik dalam bentuk foto maupun *video*. Selain itu, pengguna dapat bergabung dalam grup dengan pengguna lain yang memiliki ketertarikan yang sama. Orang menggunakan Facebook untuk tetap terhubung dengan teman dan keluarga, untuk menemukan apa yang sedang terjadi di dunia, dan untuk berbagi dan mengungkapkan hal-hal penting bagi pengguna (Facebook, 2017).

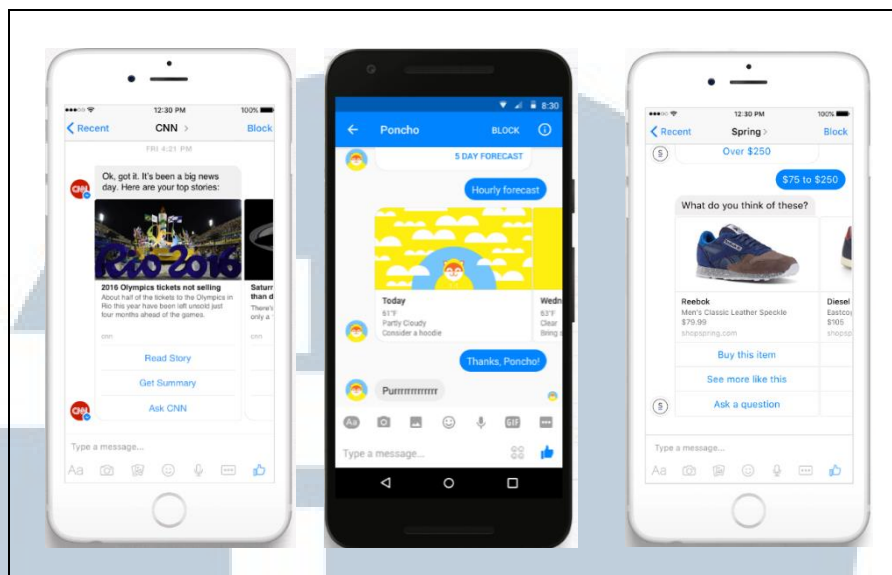
Pembuatan situs Facebook ini berawal dari beberapa mahasiswa Harvard University, yaitu Mark Zuckerberg, Eduardo Saverin, Dustin Moskovitz, dan Chris Hughes yang membangun sebuah situs jejaring sosial untuk mahasiswa Harvard pada Februari 2004 dengan nama situs domain thefacebook.com (Facebook, 2017). Pada Maret 2004, Facebook memperluas jaringan ke Stanford, Columbia dan Yale University. Kemudian situs domain thefacebook.com diubah menjadi Facebook. Setelah Facebook memperkenalkan Facebook Wall pada September 2004, dimana pengguna diberikan tempat untuk mengirim pesan kepada teman mereka, pengguna aktif Facebook mencapai satu juta pengguna (Facebook, 2017).

Facebook telah banyak meluncurkan fitur-fitur yang dapat digunakan penggunanya dan memperluas jaringannya hingga seluruh dunia. Pada 26 September 2006, Facebook membuka peluang bagi siapa saja dapat mendaftar

sebagai anggota Facebook. Dan sejak dibukanya peluang tersebut, pengguna aktif Facebook mencapai 12 juta. Berdasarkan News Room Facebook pada Maret 2017 didapatkan rata-rata pengguna aktif Facebook mencapai 1.94 miliar perhari (Facebook, 2017).

Facebook Messenger adalah aplikasi *instant messaging* pada *smartphone*, yang memungkinkan pengguna dapat menjangkau orang-orang dengan mudah menggunakan telepon (Messenger, 2015). Dengan Messenger pengguna dapat mengirim pesan pribadi dan stiker, *chatting* dalam grup, dan panggilan suara dan *video* secara gratis, bahkan untuk orang di negara lain. Pesan yang dikirim dapat berupa teks, gambar, stiker, maupun dokumen. Facebook Messenger tersedia untuk *smartphone* dan *PC Platform* (Android, iOS, dan Windows Phone). Pada 2016, lebih dari satu miliar pengguna menggunakan Messenger setiap bulan (Messenger, 2016).

Facebook memberikan kesempatan kepada *developer* untuk membuat, mengembangkan, dan menghasilkan uang dari aplikasi yang dibangun untuk Facebook. Facebook telah menyediakan *tools* yang dapat digunakan bagi *developer* salah satunya *Log In with Facebook*. Messenger juga telah membuka *platform* bagi *developer* dan pebisnis yang ingin membangun aplikasi dengan menggunakan Messenger. *Bots* yang dibangun dapat digunakan sebagai *personal assistant*, pusat informasi, dan tempat transaksi jual beli. Beberapa *bots* yang telah berhasil dibangun dengan menggunakan Messenger Platform adalah Spring, Poncho, dan CNN. Berikut tampilan aplikasi bots pada Facebook Messenger:



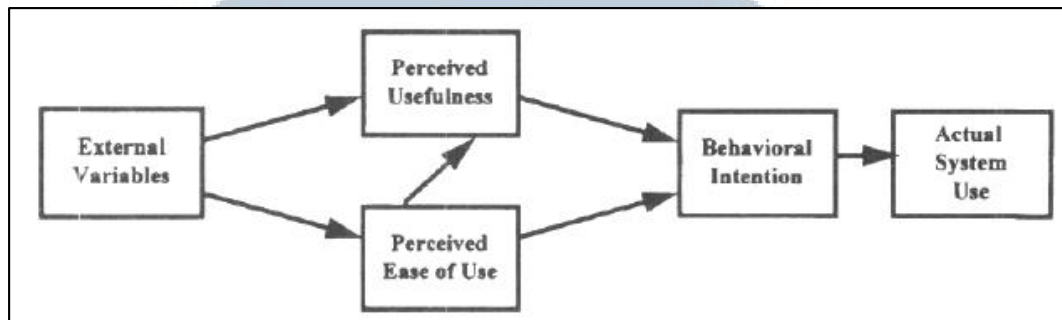
Gambar 2.3 *Mock Up* Aplikasi *Bots* pada Facebook Messenger (Messenger, 2016)

2.4 Technology Acceptance Model (TAM)

Technology Acceptance Model (TAM) diperkenalkan pertama kali oleh Davis pada tahun 1986, yang merupakan hasil pengembangan dari *Theory of Reasoned Action* (TRA) yang dikembangkan oleh Fishbein dan Ajzen pada tahun 1980. TAM merupakan salah satu model yang dibangun untuk menganalisis dan memahami faktor-faktor yang mempengaruhi diterimanya oleh penggunaan teknologi komputasi (Davis, 1989). Tujuan utama adanya TAM adalah untuk memberikan penjelasan mengenai perilaku pengguna di berbagai teknologi komputasi (*computer usage behavior*) sehingga peneliti dan praktisi dapat mengidentifikasi suatu sistem mungkin tidak dapat diterima dan memberikan langkah perbaikan (Davis, Bagozzi, dan Warshaw, 1989).

TAM memposisikan dua kepercayaan (*beliefs*), yaitu *perceive usefulness* (Persepsi Kegunaan Penggunaan) dan *perceive ease of use* (Persepsi Kemudahan Penggunaan). Kemudian menjadi niat perilaku untuk menggunakan (*behavioral*

intention system use) sehingga pada akhirnya menunjukkan penggunaan nyata dari sistem (*actual system use*).



Gambar 2.4. Pemodelan Technology Acceptance Model (Venkatesh dan Davis, 1996)

2.5 Proportionate Stratified Random Sampling

Terdapat dua metode dasar penarikan sampel yaitu *Probability Sampling* dan *Non-Probability Sampling*. Perbedaan dua teknik tersebut adalah *Probability Sampling* memberikan peluang yang sama bagi setiap unsur (anggota) populasi untuk dipilih menjadi anggota sampel, sedangkan *Non-Probability Sampling* tidak memberikan peluang atau kesempatan yang sama bagi setiap unsur populasi untuk dipilih menjadi anggota sampel (Guritno, Sudaryono, dan Rahardja, 2011). *Proportionate Stratified Random Sampling* merupakan bagian dari *Probability Sampling*. *Proportionate Stratified Random Sampling* digunakan bila populasi memiliki anggota (unsur) yang tidak homogen dan berstrata secara proposional, sehingga jumlah sampel yang diambil harus mewakili setiap strata yang ada dalam populasi (Sugiyono, 2012).

2.6 Skala Likert

Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang kejadian atau gejala sosial (Guritno, Sudaryono, dan Rahardja, 2011). Dalam penelitian gejala sosial, variabel penelitian

telah ditetapkan secara spesifik oleh peneliti. Variabel penelitian yang diukur dijabarkan hingga menjadi indikator-indikator yang dapat diukur. Dari indikator tersebut digunakan sebagai titik tolak untuk membuat item instrumen berupa pertanyaan atau pernyataan yang perlu dijawab oleh responden (Guritno, Sudaryono, dan Rahardja, 2011). Jawaban dari setiap instrument memiliki gradasi dari sangat tidak positif sampai sangat negatif.

Tabel 2.1 Skala Penilaian untuk Pertanyaan Positif dan Negatif

Keterangan	Pernyataan Positif	Pertanyaan Negatif
Sangat Tidak Setuju (STS)	1	5
Tidak Setuju (TS)	2	4
Netral (N)	3	3
Setuju (S)	4	2
Sangat Setuju (SS)	5	1

(Sugiyono, 2010)

Jumlah skor terendah dan skor tertinggi akan digunakan sebagai batasan. Kemudian dari total skor setiap variabel dianalisis apakah variabel tersebut sangat lemah atau sangat kuat, berdasarkan kriteria interpretasi skor pada Tabel 2.2.

Tabel 2.2 Kriteria Interpretasi Skor

Kategori	Persentase Skor
Sangat Lemah	20% > Skor >= 0%
Lemah	40% > Skor >= 20%
Cukup	60% > Skor >= 40%
Kuat	80% > Skor >= 60%
Sangat Kuat	Skor >= 80%

(Guritno, Sudaryono, dan Rahardja, 2011)