



### **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. Bot

*Bot* adalah perangkat lunak kecerdasan buatan yang dirancang untuk melakukan serangkaian tugas sendiri dan tanpa bantuan manusia. Tugas yang dapat dilakukan bot dapat bervariasi dari hal-hal seperti membuat reservasi di restoran, menandai tanggal di kalender atau mengumpulkan dan menampilkan informasi kepada penggunanya, dan memberi tahu pengguna tentang cuaca, dll.

#### 2.2. Chatbot

*Chatbot* adalah program komputer yang memproses *input* bahasa alami dari pengguna dan menghasilkan respons relatif dan cerdas yang kemudian dikirim kembali ke pengguna. Saat ini, *chatbots* ditenagai oleh aturan *rules-driven* atau mesin *artificial intelligent* (AI) yang berinteraksi dengan pengguna terutama melalui antarmuka berbasis teks

Ini adalah program komputer independen yang dapat diimplementasikan ke salah satu dari beberapa *platform messaging* yang telah dibuka untuk pengembang melalui API seperti Facebook Messenger, Slack, LINE, dan lain-lain (Khan & Das, 2018).

Arsitektur *chatbot* mengintegrasikan model bahasa dan algoritma komputasi untuk meniru komunikasi obrolan informal antara pengguna manusia dan komputer menggunakan bahasa alami. Pengguna dapat mengobrol *input* teks atau suara

melalui layar komputer dengan teks *chatbot output* atau *output* audio / suara (Wang & Petrina, 2013).

### **2.2.1. Kecerdasan Buatan (Artificial Intelligence)**

Kecerdasan buatan adalah seperangkat teknologi yang memungkinkan komputer untuk membantu dan memecahkan masalah dengan cara yang mirip dengan manusia dengan memersepsikan, belajar, dan bernalar. Ini memungkinkan komputer untuk belajar dari sejumlah besar data, dan berinteraksi lebih alami dan responsif dengan dunia, daripada mengikuti rutinitas yang telah diprogram (Raman & Tok, 2018).

Teknologi ini sudah banyak digunakan oleh orang-orang untuk bersaing di berbagai sektor bisnis. Bahkan para raksasa seperti Facebook, Google, dan Apple berlomba-lomba mengaplikasikannya dalam berbagai aspek. Mulai dari mesin pencari, mobil pintar, hingga periklanan.

Teknologi kecerdasan buatan dapat dipelajari dalam berbagai bidang-bidang seperti robotika (*robotics*), penglihatan komputer (*computer vision*), pengolahan bahasa alami (*natural language processing*), pengenalan pola (*pattern recognition*), sistem syaraf buatan (*artificial neural system*), pengenalan suara (*speech recognition*) dan sistem pakar (*expert system*) (Simarmata, 2006).

### **2.2.2. Natural Language Processing (NLP)**

NLP adalah bidang dari *artificial intelligence* dan linguistik, yang ditujukan untuk membuat komputer memahami pernyataan atau kata-kata yang

ditulis dalam bahasa manusia. NLP pada dasarnya adalah kemampuan untuk mengambil tubuh teks yang dihasilkan manusia dan membuatnya menjadi bahasa yang dapat dibaca mesin. NLP juga menganalisis dan mengekstrak *metadata* utama dari teks.

NLP mendapat banyak perhatian untuk mewakili dan menganalisis bahasa manusia secara komputasi. Ini telah menyebar aplikasinya di berbagai bidang seperti dapat diterapkan ke berbagai bidang seperti, *machine translation*, *email spam detection*, *information extraction*, *summarization*, *medical*, *question answering* (Khurana, Koli, Khatter, & Singh, 2017) (Raman & Tok, 2018).

### 2.2.3. Pattern Matching

*Pattern matching* adalah pencocokan dengan mencocokkan kata kunci dalam suatu kalimat, kemudian itu akan dicari dan dicocokkan ke dalam *database* serta memberi balasan dengan jawaban yang sesuai (Mhatre, Motani, Shah, & Mali, 2016). Dalam pengembangan *chatbot*, *pattern matching* ini adalah suatu teknik kecerdasan buatan yang digunakan dalam merancang suatu *chatbot*. *Input* yang diberikan dicocokkan dengan *input* yang telah disimpan dalam *database* kemudian *response* akan dikembalikan (Dahiya, 2017).

## 2.3. LINE

LINE adalah suatu aplikasi dari Jepang yang digunakan untuk kegiatan berkirim pesan (*messenger / chatting*) secara gratis di perangkat *smartphone*. Ini dioperasikan oleh perusahaan layanan Internet LINE Co., Ltd. (nama sebelumnya:

NHN Japan Co., Ltd.), dan telah terdaftar di Tokyo dan New York pada saat yang sama pada Juli 2016.

LINE menggunakan sistem nomor telepon seluler penggunaanya sebagai basis untuk saling berhubungan. LINE saat ini tersedia untuk *gadget* yang memiliki sistem operasional iOS dan Android. LINE dapat diunduh secara gratis di App Store dan Google Play. LINE berbeda dari aplikasi IM lainnya, karena ada *emoticon* yang bervariasi. Ada *emoticon* yang menggambarkan kepala dengan bermacam ekspresi, lalu *emoticon* berupa susunan karakter teks yang juga membentuk ekspresi, serta ada *stickers*. *stickers* ini yang cukup unik untuk LINE, karena gambar ikonnya lucu-lucu, berukuran besar dan lebih ekspresif (Irianto, Yusanto, & Putri, 2015).

#### **2.4. API (Application Programming Interface)**

API adalah protokol yang harus diikuti oleh *programmer* untuk memanfaatkan beberapa fungsionalitas inti dari bahasa pemrograman. Dalam istilah yang sederhana, sebuah API adalah cara bagi dua buah perangkat lunak untuk berkomunikasi. Tujuan dari API adalah untuk memfasilitasi interaksi antara kedua program ini (Wagner, Eichner, & Graham, 2013).

#### **2.5. Messaging API LINE**

*Messaging API LINE* memungkinkan data dilewatkan antara *server* aplikasi *bot* dan *Platform LINE*. Ketika seorang pengguna mengirimkan pesan ke aplikasi *bot*, *webhook* akan terpicu dan *Platform LINE* akan mengirimkan permintaan ke URL *webhook* tersebut. Setelah itu, *server* akan mengirim permintaan ke *Platform*

LINE untuk menanggapi pengguna. Permintaan dikirim melalui HTTPS dalam format JSON (Documentation : Messaging API, n.d.).

## 2.6. Platform

*Platform* (komputasi) adalah kerangka kerja teknis di mana satu atau lebih aplikasi dapat dijalankan dan di mana data disimpan. Untuk tujuan dari interaksi pengguna (UI) dan pengalaman pengguna (UX), istilah "*website*" bukan platform yang akan digunakan (Defining principles and guidelines for mobile apps and platform, 2016).

### 2.6.1. Platform Kata.ai

*Platform* Kata.ai menyediakan semua elemen yang dibutuhkan untuk menciptakan *chatbot* dalam satu platform yang terintegrasi. Sistem dari *Platform* Kata.ai memungkinkan anda untuk mendesain, melatih, serta mengelola *chatbot* tanpa hambatan (All-in-One Chatbot Platform, n.d.).

### 2.6.2. Bot Studio

*Bot studio* membantu untuk mempercepat proses pengembangan *chatbot* dengan berkolaborasi. Dengan *Bot Studio*, pemilik bisa dengan mudah membuat desain percakapan dengan fitur "*drag & drop*". Di waktu yang sama, *developer* bisa melakukan pemrograman menggunakan sistem berbasis YAML (*Bot Studio*, n.d.).

### 2.6.3. Smart Deployment Manager

Fitur *Smart Deployment Manager* memudahkan *chatbot* yang telah dibangun untuk terhubung secara instan ke *messaging platform* yang ingin digunakan, sekaligus memastikan hanya *chatbot* berkualitas tinggi yang tersaji ke pengguna (Smart Deployment Manager, n.d.).

### 2.6.4. Natural Language Understanding (NLU)

NLU adalah objek yang mengubah pesan teks menjadi objek terstruktur. NLU di NL Studio merupakan kumpulan model pembelajaran mesin yang digunakan untuk mengubah bahasa manusia menjadi bentuk yang lebih mudah diproses oleh mesin. Model itu sendiri adalah bentuk dan kumpulan aturan yang dihasilkan dari data pelatihan menjadi yang berguna sebagai pengambil keputusan untuk *input*. Satu NLU mungkin memiliki beberapa entitas (NL Studio : NLU, n.d.).

### 2.6.5. Flow

*Flow* adalah aliran atau arus utama dari *bot* sesuai dengan wacana dalam percakapan. *Flow* terdiri dari *intent*, *state*, dan *action*. Pengolahan pesan dari pengguna dibagi menjadi 3 langkah :

#### 1) *Intent Recognition*

*Intent Recognition* ini mengubah pesan mentah (*input user*) menjadi informasi yang terstruktur (dalam bentuk *intents* dan *attributes*).

## 2) *State Mapping*

*State Mapping* ini menentukan dimana keadaan *bot* dan *variable context*.

## 3) *Chosing Action*

*Chosing Action* ini akan menghasilkan respons dari hasil pengeksekusian *action* yang telah ditentukan (Concepts : Flow, n.d.).

### 2.6.6. Intent

*Intent* dapat dipahami sebagai esensi / makna dari suatu ucapan. Dalam sebuah wacana kita dapat membedakan antara *initial intent* dan *follow up intent*.

Ada beberapa tujuan dalam proses pemilihan *intent*:

- 1) Ini akan memilih *intent* dengan mencocokkan kondisi dan jenisnya.
- 2) Jika *intent* mempunyai *classifier* maka akan dijalankan dan akan membatalkan pilihan jika *classifier* tidak cocok.
- 3) Ini akan memilih 1 *intent* yang cocok dengan prioritas tertinggi (Concepts : intent, n.d.)

### 2.6.7. State

*State* mempresentasikan tahapan dalam pembicaraan. Setiap *flow* yang baru dibuat, akan dimulai dari *initial state*. Setiap *flow* setidaknya memiliki 1 *end state*. *End state* akan menyebabkan suatu *flow* berakhir / ditutup dan itu berarti semua *context* yang terkait dengan *flow* tersebut akan dihapus. Semua *flow* yang telah ditutup dapat dibuka kembali dengan suatu pesan yang memicu *initial state*-nya (Concepts : State, n.d.).



### 2.6.8. Action

*Action* adalah bagian yang memicu respons terhadap pengguna. *Action* ini biasa dilekatkan dengan suatu *state*, sehingga *action* akan terpicu ketika suatu *flow* mencapai *state* tertentu. Jenis *action* ini bisa berupa:

- 1) *Text* (Mengembalikan 1 / lebih *bubble text*).
- 2) *Image* (Mengembalikan gambar).
- 3) *Template* (*Rich action*).
- 4) *Command* (Memicu *command*).
- 5) *API* (Memicu untuk memanggil API) (Concepts : Action, n.d.).

### 2.7. Web Hosting

*Web Hosting* adalah suatu bentuk layanan yang telah menyediakan suatu *server* penyimpanan (*storage*) data untuk penempatan situs *web* dimana dapat diakses umum melalui koneksi ke jaringan internet dari *personal computer* (PC) di seluruh penjuru dunia. *Web hosting* berarti internet *hosting* yang memungkinkan bisnis dan individu untuk membuat kehadiran mereka *online* dalam bentuk *website*, dapat diakses oleh publik melalui internet (Aditya & Handoyo, 2011).

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

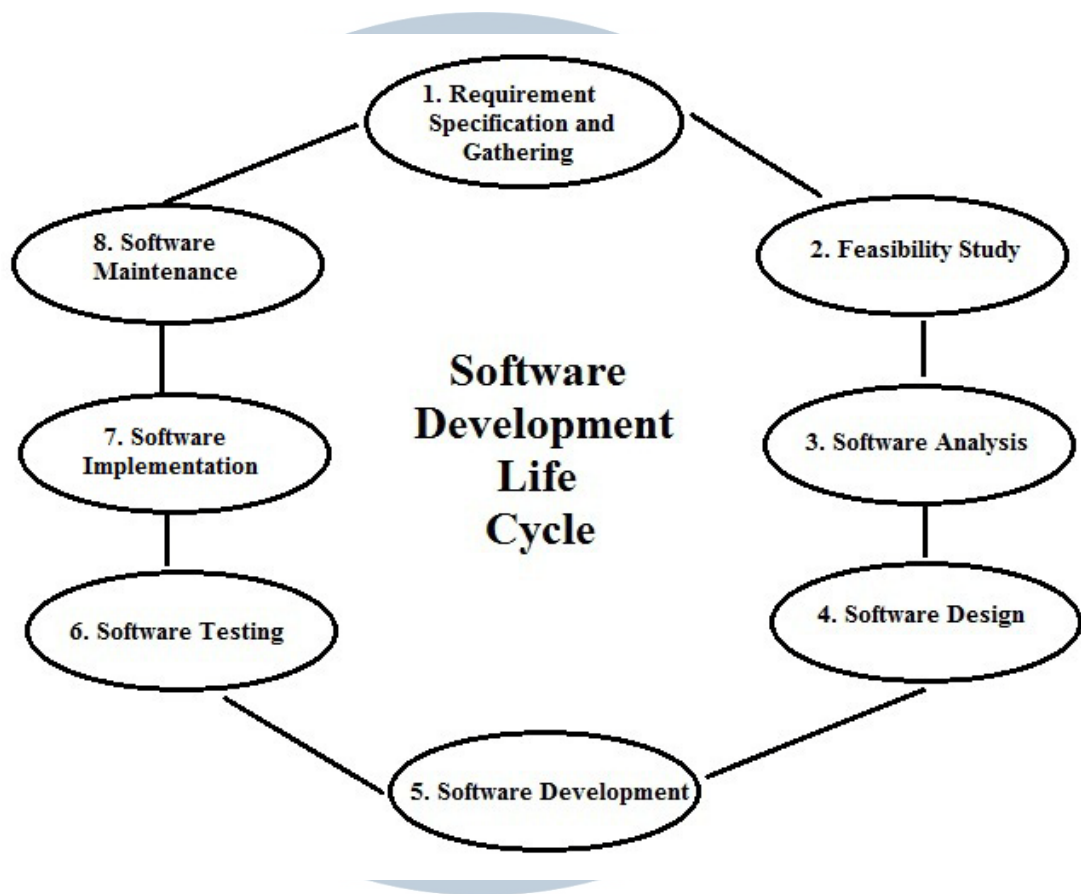
## 2.8. System Development Life Cycle (SDLC)

SDLC adalah konsep yang digunakan dalam rekayasa perangkat lunak untuk menggambarkan prosedur untuk perencanaan, pembuatan, pengkodean, pengujian dan implementasi spesifikasi kebutuhan pengguna. Fase- fase yang ada pada SDLC adalah sebagai berikut:

- 1) *Requirement specification and gathering*
- 2) *Feasibility Study*
- 3) *Analysis*
- 4) *Design*
- 5) *Development or Coding*
- 6) *Testing*
- 7) *Implementation*
- 8) *Maintenance*

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



**Gambar 2.1. Fase dari *System Development Life Cycle***

**Sumber:** (Sharma, 2017)

SDLC berlaku untuk berbagai konfigurasi perangkat keras dan perangkat lunak. SDLC adalah proses langkah demi langkah untuk membuat perangkat lunak berkualitas bagi pengguna (Sharma, 2017).

## **2.9. Rapid Application Development (RAD)**

RAD adalah kumpulan metodologi yang muncul sebagai tanggapan terhadap kelemahan pengembangan menggunakan *waterfall* dan variasinya. RAD menggabungkan teknik dan komputer khusus alat untuk mempercepat fase analisis,

desain, dan implementasi untuk mendapatkan beberapa bagian dari sistem dikembangkan dengan cepat dan ke tangan pengguna untuk evaluasi dan *feedback* (Dennis, Wixom, & Roth, 2012).

Metode ini juga merupakan metodologi untuk mengembangkan perangkat lunak yang memerlukan perencanaan minimum untuk pembuatan prototipe yang cepat (Naz & Khan, 2015). RAD menekankan pada siklus pembangunan jangka pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini. Metode RAD ini bisa menangani perubahan-perubahan yang terjadi dengan cepat.

Fase-fase dalam metode *Rapid Application Development* (RAD) yaitu:

1. *Business modeling*

*Business modeling* adalah fase yang digunakan untuk mendapatkan informasi yang penting dalam bisnis tersebut, mulai dari datangnya informasi, siapa yang membuat informasi, siapa yang memproses, dan lain-lain.

2. *Data modeling*

*Data modeling* adalah fase yang digunakan untuk menyaring lagi semua informasi yang telah dikumpulkan dari fase sebelumnya. Fase ini berfungsi untuk mendapatkan objek data yang dibutuhkan untuk menopang bisnis tersebut. Masing-masing objek tersebut akan diidentifikasi dan didefinisikan hubungan antar objek-objeknya.

3. *Process modeling*

*Process modeling* adalah fase yang digunakan untuk menetapkan arus informasi bisnis yang diperlukan untuk mencapai tujuan bisnis yang sesuai

model bisnisnya. Objek data pada fase tersebut akan diimplementasikan pada fungsi bisnis tersebut. Proses ini dibangun untuk menambahkan, menghapus, mengambil atau memodifikasi objek data yang diberikan.

#### 4. *Application generation*

*Application generation* adalah fase yang digunakan untuk merancang dan membangun sistem dengan menggunakan *automatic tools* untuk mengubah model proses dan data menjadi prototipe yang aktual.

#### 5. *Testing and turnover*

*Testing and turnover* adalah fase yang digunakan untuk menguji keandalan sistem, komponen dan semua yang telah dirancang. Proses RAD menekankan pada pemakaian kembali, sehingga banyak komponen program yang telah diuji. Hal ini akan mengurangi keseluruhan waktu pengujian sistem.

Dengan menggunakan RAD dapat membuat waktu yang lebih singkat daripada membutuhkan model lama siklus pengembangan sistem antara merancang dan mengimplementasikan sistem informasi. Intinya adalah RAD mencoba untuk memenuhi persyaratan bisnis yang berubah dengan cepat. Setiap prototipe diuji oleh pengguna dan *feedback* akan dikumpulkan. *Feedback* ini digunakan untuk memodifikasi struktur proyek yang ada dan menerapkan perubahan sesuai dengan interaksi pengguna dengan prototipe. Proses uji ini diulang untuk setiap prototipe yang menyebabkan *bug* menjadi lebih sedikit pada aplikasi akhir (Wibowo & Delima, 2017) (Rico, 2012).

## 2.10. Penelitian Sebelumnya

**Tabel 2.1. Penelitian Sebelumnya**

	Nama Peneliti	Namita Mhatre, Karan Motani, Maitri Shah dan Swati Mali.
	Judul Penelitian	<i>Donna Interactive Chat-bot acting as a Personal Assistant</i>
	Tahun	2017
	Objek Penelitian	<i>Chatbot Donna, Chatbot A.L.I.C.E, dan Chatbot x.ai (AMY).</i>
1	Metode Penelitian	<p>Pada penelitian tersebut, peneliti menggunakan tipe dari NLP yaitu <i>pattern matching</i> dimana sistem mencari kata kunci dalam suatu kalimat dan berdasarkan kata kunci tersebut, itu akan dicari ke dalam <i>database</i> untuk dicocokkan dan dibalas dengan jawaban yang sesuai.</p> <p>Ada 4 modul yang digunakan dalam sistem arsitektur dalam perancangan <i>Chatbot Donna</i>, yaitu <i>presentation layer, service layer, data access layer, database layer</i>. Saat tahap implementasi, <i>Donna</i> menggunakan beberapa <i>web server</i> seperti akses internet, PHP 5+, MySQL, XAMPP, dan lain-lain.</p> <p>Penulis juga membandingkan aplikasi <i>chatbot</i> lainnya seperti A.L.I.C.E dan juga x.ai (AMY). Ide dibuatnya <i>Donna</i> ini karena terinspirasi dari <i>Chatbot x.ai</i>, tapi <i>Donna</i> lebih baik daripada x.ai karena x.ai juga masi dalam versi beta.</p> <p>Pada tahap eksperimen, dilakukan tes untuk mengetahui cara menjawab yang dilakukan oleh <i>Donna</i> di mana sebelum di ajarkan dan setelah di ajarkan. Perbedaannya cukup terlihat karena ternyata didapatkan fakta bahwa <i>Donna</i> adalah <i>bot</i> yang mampu belajar dan modul ini membuat kecerdasan buaatannya terlihat.</p>
	Hasil Penelitian	<p>Hasil dari penelitian tersebut terlihat bahwa <i>Chatbot Donna</i> adalah <i>chatbot</i> interaktif yang terdiri dari elemen yang dapat belajar dengan sendirinya, sehingga setiap kali pengguna menemukan jawaban yang tidak benar dari jawaban <i>Donna</i>, pengguna akan mengajarkan jawaban yang benar kepada <i>Donna</i> untuk pertanyaan yang sama dan <i>Donna</i> mempelajari itu.</p> <p>Hasil lain dari penelitian tersebut adalah <i>Chatbot Donna</i> dapat menjawab dengan kualitas rata-rata sekitar 60% dari 700 pesan yang di uji kepada <i>Donna</i>.</p> <p>Manusia juga sering kali membuat kesalahan dalam memahami pernyataan atau pertanyaan dan mungkin menjawabnya dengan tidak relevan. Kecerdasan buatan yang ada di dalam <i>Donna</i> ini membuat kesalahan yang dibuat oleh sistem tidak dapat dibedakan dengan yang dibuat oleh asisten manusia.</p>

2	Nama Peneliti	Rishabh Shah, Siddhant Lahoti, dan Prof. Lavanya. K
	Judul Penelitian	<i>An Intelligent Chat-bot using Natural Language Processing</i>
	Tahun	2017
	Objek Penelitian	<i>Intelligent chatbot</i>
	Metode Penelitian	<p>Pada penelitian ini, penelitian ini menggunakan beberapa metode untuk membuat suatu <i>chatbot</i> yang pintar, yaitu AIML, NLP, <i>query extraction</i>, <i>pattern matching</i>, <i>splitting / classifying the word</i>, <i>dataset</i>, <i>knowledge base</i> dan LSTM (<i>Long Short Term Memory</i>).</p> <p>Proses model yang akan digunakan yaitu model <i>incremental</i>. Secara perlahan akan menambahkan dan meningkatkan fungsionalitas untuk mengembangkan modul perangkat lunak.</p>
Hasil Penelitian	<p>Hasil dari penelitian tersebut adalah pembuatan <i>chatbot</i> menggunakan AIML akan lebih baik ketika <i>chatbot</i> dapat bekerja dengan pertanyaan umum dimana dapat langsung disimpan dalam <i>database</i> dengan <i>pattern matching</i>.</p> <p><i>Chatbot</i> dapat bekerja pada <i>template</i> yang tetap sehingga akan memberikan hasil yang lebih baik, kecuali algoritma <i>machine learning</i> telah masuk, karena <i>machine learning</i> dapat bekerja untuk menyediakan suatu aturan sehingga dapat membuat aturan sederhana berdasarkan dalam model yang kompleks.</p> <p>Kecenderungan hasil menjanjikan dalam NLP dan teknik belajar mandiri akan menjadi tambahan yang bagus untuk <i>chatbot</i> di mana dapat membuatnya lebih seperti manusia.</p>	
3	Nama Peneliti	Ameya Vichare, Ankur Gyani, Yashika Shrikhande, dan Nilesh Rathod
	Judul Penelitian	<i>A chatbot system demonstrating Intelligent Behaviour using NLP</i>
	Tahun	2015
	Objek Penelitian	<i>Chatbot ELIZA, Chatbot ALICE, dan Siri</i>
	Metode Penelitian	<p>Pada penelitian ini akan membahas tentang sistem yang dapat berfungsi sebagai aplikasi dan dapat memberikan informasi kepada pengguna tentang berbagai jenis olahraga yang ada di dunia. Peneliti juga menggunakan <i>Chatbot ELIZA, Chatbot ALICE, dan Siri</i> dalam membantu pembuatan sistem mereka. Secara singkat, ELIZA merupakan salah satu <i>chatbot</i> pertama yang dirancang tahun 1966. ELIZA bekerja dengan <i>simple parsing</i> dan substitusi kata kunci ke dalam frasa. ALICE adalah <i>chatbot</i> yang terinspirasi dari ELIZA. ALICE menggunakan NLP, yaitu program yang terlibat dalam percakapan dengan manusia yang menerapkan beberapa aturan pencocokan pola dari</p>

	<p><i>input</i> pengguna dan akan menghasilkan tanggapan yang sesuai. Siri adalah program komputer yang berfungsi sebagai asisten pribadi virtual dan navigator pengetahuan. Fitur ini menggunakan <i>natural language user interface</i> untuk menjawab pertanyaan, memberikan saran, dan melakukan tindakan dengan menggunakan suatu <i>web service</i>.</p> <p>Pembuatan sistem tersebut akan menggunakan algoritma <i>pattern matching</i> menggunakan <i>depth first search</i> (DFS). Ini akan mengkonversi <i>user's query</i> dari <i>speech</i> ke teks menggunakan konverter <i>speech-to-text</i> dan <i>back to speech</i> ke sebagai <i>output</i> ke pengguna.</p> <p>Sistem menemukan apa yang sebenarnya diinginkan pengguna dengan mengambil makna semantik dari <i>inputan</i>. Sistem akan meneruskan teks semantik yang dimasukkan pengguna sebagai masukan ke algoritma <i>pattern matching</i>. Sistem <i>pattern matching</i> akan memanfaatkan pengetahuan yang sudah dimasukkan ke dalam <i>database</i> untuk menghasilkan <i>response</i>. <i>Response</i> yang telah dihasilkan, akan diubah kembali menjadi ucapan dan aplikasi akan membaca isinya.</p>
<p>Hasil Penelitian</p>	<p>Hasil dari penelitian ini adalah peneliti telah belajar dari semua sistem yang ada termasuk ELIZA, ALICE, dan Siri. ELIZA membantu dalam memahami bagaimana menata ulang pertanyaan-pertanyaan yang dapat membuat percakapan seperti manusia. ALICE membantu dalam memahami bagaimana memanfaatkan AIML dalam sistem. Siri membantu dalam memahami keterbatasan pemrosesan bahasa dari <i>speech</i> ke <i>speech agents</i>. Sistem tersebut mengimplementasikan NLP dengan algoritma <i>pattern matching</i> menggunakan DFS untuk mencocokkan <i>query</i> dan berlanjut sampai kecocokan ditemukan di Android OS.</p> <p>Dalam sistem ini, pengguna akan diminta untuk menanyakan pertanyaan yang terkait dengan olahraga. Pengguna akan mengucapkan suatu kalimat di ponselnya menggunakan UI aplikasi. Berbagai pertanyaan terkait olahraga dan tanggapan mereka akan dikodekan ke AIML dan disimpan ke dalam <i>database</i>. Ketika pengguna akan mengajukan permintaan, permintaan ini akan dicocokkan dengan berbagai pola yang ada dalam <i>database</i> dan <i>template</i> yang sesuai dengan pola itu akan dikembalikan dalam bentuk ucapan kepada pengguna.</p>



Pada penelitian pertama yang berjudul “*Donna Interactive Chat-bot acting as a Personal Assistant*”, peneliti menggunakan *pattern matching* yang merupakan tipe dari NLP dan terbukti bahwa *Chatbot* Donna dapat menjawab dengan kualitas rata-rata sekitar 60% dari 700 pesan yang di uji kepada Donna.

Pada penelitian kedua yang berjudul “*An Intelligent Chat-bot using Natural Language Processing*”, peneliti menggunakan beberapa metode untuk membuat suatu *chatbot* yang pintar, yaitu AIML, NLP, *query extraction*, *pattern matching*, *splitting / classifying the word*, *dataset*, *knowledge base* dan LSTM (*Long Short Term Memory*). Penelitian ini membuktikan bahwa pembuatan *chatbot* menggunakan AIML akan lebih baik ketika *chatbot* dapat bekerja pada *template* yang tetap dengan pertanyaan umum dimana dapat langsung disimpan dalam *database* dengan *pattern matching*, kecuali algoritma *machine learning* telah masuk, karena *machine learning* dapat bekerja untuk menyediakan suatu aturan sehingga dapat membuat aturan sederhana berdasarkan dalam model yang kompleks.

Pada penelitian ketiga yang berjudul “*A chatbot system demonstrating Intelligent Behaviour using NLP*”, peneliti mengimplementasikan metode NLP dengan algoritma *pattern matching* menggunakan *depth first search* (DFS) dan didapatkan hasil bahwa pertanyaan dan tanggapan pengguna akan dikodekan ke AIML dan disimpan ke dalam *database*. Ketika pengguna akan mengajukan permintaan, permintaan ini akan dicocokkan dengan berbagai pola yang ada dalam *database* dan *template* yang sesuai dengan pola itu akan dikembalikan dalam bentuk ucapan kepada pengguna.

Dari penelitian-penelitian sebelumnya yang telah disajikan di atas, akan diimplementasikan algoritma *pattern matching* berdasarkan kata kunci yang dilakukan oleh (Mhatre, Motani, Shah, & Mali, 2016), karena dalam pengimplementasian *chatbot* di Kata.ai akan menggunakan pencocokan kata kunci juga. Kata kunci dari pengguna akan dicocokkan dengan model *keyword* yang akan digunakan dalam Kata.ai, sehingga nantinya akan menghasilkan respons yang sesuai dan dikembalikan ke pengguna.

A large, light blue circular logo with a white smiley face (:) in the center, positioned behind the main text.

# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA