

BAB II

LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 *Chatbot*

Chatbot merupakan sebuah aplikasi obrolan yang dirancang dan dibangun agar dapat melakukan interaksi dengan manusia sebagai pengguna dengan penggunaan media pesan atau teks, suara, yang telah disisipkan Kecerdasan Buatan di dalamnya sehingga dapat membalas pesan-pesan atau pertanyaan yang diberikan oleh manusia secara tepat melalui respon yang sebelumnya sudah ditetapkan pada basis data yang ada di dalam aplikasi *Chatbot* [6].

Aplikasi *chatbot* sendiri terdiri dari 3 macam kombinasi yaitu Tampilan Antarmuka (*User Interface*), Kecerdasan Buatan (*Artificial Intelligence*), dan Integrasi. *User Interface* merupakan tampilan antarmuka dalam aplikasi *chatbot* yang menjadi jembatan penghubung antara *sistem chatbot* dan pengguna (*user*) agar dapat saling berinteraksi. Kecerdasan Buatan dalam *chatbot* berfungsi untuk memahami setiap interaksi yang terjadi antara *chatbot* dengan pengguna. Fitur Integrasi dalam aplikasi *chatbot* memungkinkan sistem *chatbot* untuk terhubung dengan sistem yang lain seperti *platform* media sosial *Whatsapp*, *Telegram*, *Facebook*, *Slack*, dan lainnya.[7]

Dari segi pemahaman obrolan, aplikasi *Chatbot* terbagi dalam 2 jenis yaitu ***Open-Domain Chatbot*** dan ***Closed-Domain Chatbot***.

2.1.1.1 *Open-Domain Chatbot*

Chatbot yang bersifat *open-domain* adalah *chatbot* dengan sistem yang dapat melakukan interaksi dengan manusia dengan topik sehari-hari atau apapun dan sistem *chatbot* dapat membalasnya, dengan begitu pengguna dapat bertanya tentang topik apapun dan sistem *chatbot* akan diharapkan untuk dapat menjawabnya secara masuk akal dan logis. [8]

2.1.1.2 Closed-Domain Chatbot

Chatbot yang bersifat *closed-domain* merupakan *chatbot* yang dibangun untuk untuk menyelesaikan suatu permasalahan bisnis dalam sektor apapun secara khusus, sehingga hanya sistem *chatbot* hanya dapat menjawab kepada topik-topik tertentu [9]. Pada penelitian ini akan berfokus kepada pembangunan aplikasi *Chatbot* yang bersifat *closed-domain* karena bertujuan untuk menjawab dan memberikan informasi hanya seputar jurnal saja seperti informasi untuk penulis artikel, pembaca artikel, dan pengulas.

2.1.2 PHP (Hypertext Processor)

PHP atau kepanjangan dari *Hypertext Preprocessor* merupakan sebuah bahasa pemrograman yang bersifat *server-side scripting* yang digunakan untuk membuat situs web yang bersifat dinamis dengan berjalan di sebuah *web server* yang bertugas untuk memproses hasil dari kode php dan ditampilkan di sisi klien [10]

2.1.3 MySQL

MySQL merupakan salah satu jenis basis data yang bersifat relasional yang di mana *SQL*-nya merupakan singkatan dari *Structured Query Language*. Basis data merupakan sebuah kumpulan dari data-data yang bersifat kompleks dan disusun dan dinormalisasikan sehingga setiap datanya dapat saling terhubung dan mudah diakses [11].

2.1.4 Open Journal Systems

Open Journal Systems atau yang disingkat *OJS* merupakan sebuah sistem manajemen untuk pengelolaan artikel-artikel jurnal secara daring yang dibuat dan dikembangkan oleh *Public Knowledge Project*, sistem manajemen tersebut dibuat dengan berbasis *PHP* dan dapat digunakan secara gratis oleh siapapun [12].



OJS

OPEN
JOURNAL
SYSTEMS

Gambar 2.1 Logo Open Journal Systems

Situs *Journal of Multidisciplinary Issues* sendiri telah menggunakan sistem *Open Journal Systems* versi 3.3 dan mendukung penggunaan basis data MySQL.

2.1.5 *Natural Language Processing*

Natural Language Processing atau yang disingkat dengan *NLP* merupakan sebuah proses untuk pengolahan bahasa alami. Sistem yang menerapkan metode *Natural Language Processing* memiliki tujuan untuk dapat mengerti konsep-konsep dan arti dari bahasa yang digunakan oleh manusia secara alami [13]. Bahasa alami merupakan sebuah bahasa yang digunakan untuk berkomunikasi oleh manusia pada satu sama lain.

Terdapat beberapa situs yang menyediakan layanan *NLP* seperti Prosa.ai yang dapat menyediakan metode *NLP* untuk bahasa Indonesia [14] Pada penelitian ini, akan digunakan *platform* yang bernama *Dialogflow* yang merupakan layanan dari *Google* dalam membangun sebuah sistem *Chatbot* yang dapat mendukung penggunaan dari metode *Natural Language Processing*.

2.1.6 *Dialogflow*

Dialogflow yang sebelumnya bernama API.ai merupakan layanan atau *platform* yang menyediakan teknologi *Natural Language Processing* atau *Natural Language Understanding* yang dapat memudahkan penggunaannya untuk membangun aplikasi seperti *chatbot* dan sistem interaksi menggunakan suara yang berbasis aplikasi *mobile* ataupun berbasis situs web [15]. Perusahaan *Google* mengakuisisi perusahaan API.ai pada tahun 2016 dan merubah namanya menjadi *Dialogflow* [16].



Gambar 2.2 Logo Dialogflow

Penggunaan *Dialogflow* memiliki beberapa keuntungan seperti sistem mudah untuk dibangun dan dikelola, didukung oleh layanan *Machine Learning* oleh *Google*, mendukung berbagai *platform* dan berbagai bahasa. Terdapat 2 versi yang dimiliki *Dialogflow*, yaitu *Dialogflow Essentials* dan *Dialogflow CX*. Pada penelitian ini akan menggunakan versi *Dialogflow Essentials* dengan *Trial Edition*, yaitu versi yang menyediakan fitur-fitur yang standar dan cocok untuk proyek berskala kecil serta gratis dalam penggunaannya.

Dialogflow terdiri dari beberapa istilah penting dan mendasar dalam membangun sistem *Chatbot*, yaitu sebagai berikut[16] :

- *Agent*

Agent merupakan sebuah istilah untuk penamaan sistem aplikasi *Chatbot* yang akan dibuat untuk menangani obrolan-obrolan dengan pengguna.

- *Intent*

Intent merupakan sebuah objek yang menggambarkan maksud atau tujuan dari pesan yang diberikan oleh pengguna, ketika seorang pengguna mengirim pesan ke *Chatbot, Dialogflow* akan memasangkan pesan yang dikirim dengan *Intent* yang paling cocok berdasarkan *training phrases*. Setelah pesan tersebut cocok dengan *Intent*, maka *chatbot* akan mengirim jawaban di dalam *Intent* tersebut sebagai respon dari *chatbot* kepada penggunanya.

- *Responses*

Responses merupakan jawaban yang terdapat dalam *Intent* yang akan disampaikan oleh aplikasi *Chatbot* kepada penggunanya apabila pengguna telah mengirimkan pesan atau pertanyaan yang cocok dengan *Intent* tersebut.

- *Training Phrases*

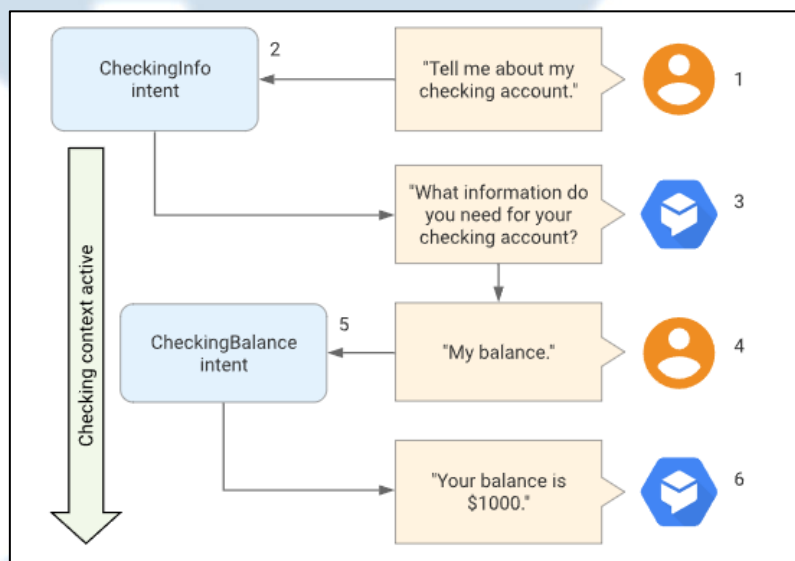
Training Phrases merupakan kumpulan-kumpulan kalimat yang digunakan untuk melatih *chatbot* agar ketika pengguna mengirimkan pesan, pesan tersebut akan dicocokkan kepada daftar kalimat tersebut meskipun terdapat salah pengejaan, teknologi dari *Dialogflow* yaitu *Natural Language Processing* atau *Natural Language Understanding* akan memahami apa yang dimaksud oleh pengguna dan *Intent* yang cocok untuk pesan tersebut.

- *Fallback Intent*

Fallback Intent merupakan sebuah *Intent* yang akan dituju apabila tidak terdapat kecocokan antara pesan yang dikirim oleh pengguna dengan daftar *Intent* yang ada, dan mengirim isi respon yang ada pada *Fallback Intent* kepada pengguna. Contohnya seperti “Maaf, saya tidak mengerti apa yang kamu maksud”.

- *Contexts*

Contexts merupakan sebuah fitur yang dapat melakukan kontrol terhadap alur percakapan antara pengguna dengan aplikasi *Chatbot*. Ketika sebuah *intent* dipanggil dan cocok, maka semua *output context* untuk *intent* tersebut akan menjadi aktif, sehingga *Dialogflow* dapat lebih cenderung untuk mencocokkan *intent* yang telah dikonfigurasi dengan *input contexts* yang sesuai dengan *intent* yang sedang aktif. Berikut adalah diagram dari penggunaan *context* yang dapat dilihat pada gambar 2.3.



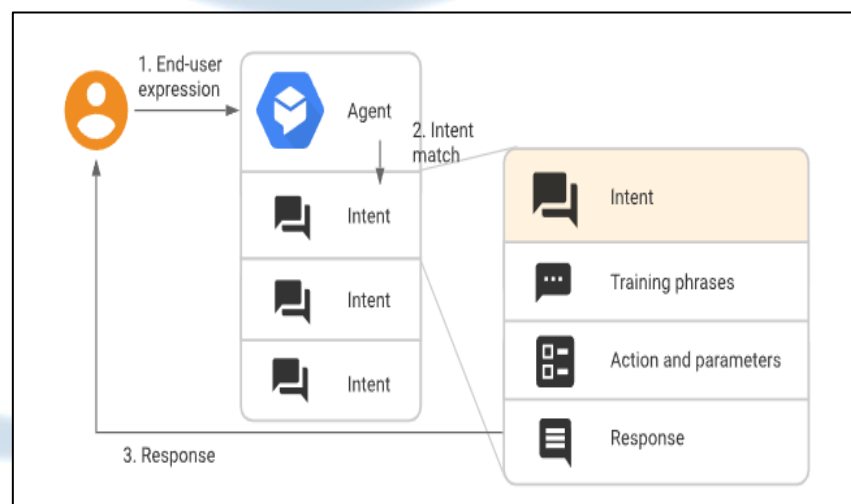
Gambar 2.3 Diagram Contoh Penggunaan Context pada *Dialogflow*

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2.1.6.1 Intent Matching

Intent Matching atau pencocokan *Intent* merupakan sebuah proses di mana *Dialogflow* melakukan pencocokan antara pesan yang dikirim oleh *user* (*end-user expression*) dengan daftar *training phrases* yang ada dalam setiap *Intent* yang ada untuk mendapatkan kecocokan terbaik. *Dialogflow* memiliki 2 macam algoritma yang digunakan untuk melakukan pencocokan tersebut, yaitu algoritma *ML Matching* dan *Rule-based Grammar Matching* [15]. Kedua algoritma ini dipakai secara bersamaan dan hasil pencocokan terbaik di antara kedua algoritma tersebut yang akan dipakai.

Berikut merupakan alur diagram percakapan dengan menggunakan metode pencocokan *Intent* yang dimiliki oleh *Dialogflow* [15].



Gambar 2.4 Alur Percakapan Dialogflow dengan Intent Matching

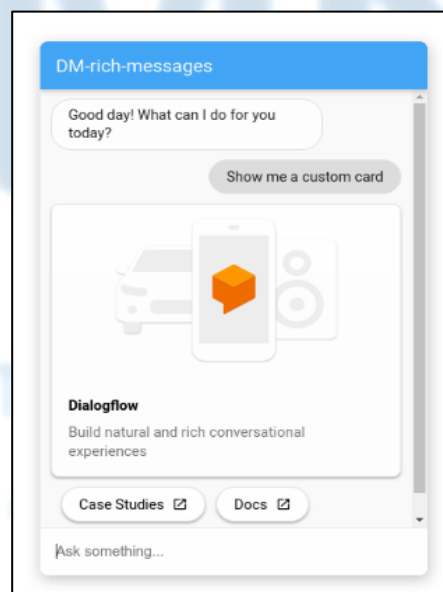
Sebuah *Agent Chatbot* akan terdiri dari beberapa topik percakapan yang termasuk ke dalam *Intent*, yang artinya dalam sebuah *Agent* akan terdiri dari beberapa macam *Intent*. Di dalam *Intent*, terdapat daftar pertanyaan atau pesan yang dapat

dimasukkan ke dalam *training phrases*, dan jawaban yang akan diberikan di dalam *response*. Ketika pengguna mengirimkan pesan atau pertanyaan yang cocok dengan daftar *Intent* yang ada, maka *Intent* tersebut akan dipanggil dan mengirim pesan *response* yang terdapat pada *Intent* tersebut kepada pengguna.

2.1.6.2 Dialogflow Messenger

Layanan *Google Dialogflow* menyediakan beberapa macam cara untuk mengintegrasikan aplikasi *chatbot* di berbagai aplikasi atau platform seperti situs web, *Slack*, *Facebook Messenger*, *LINE*, *Telegram*, dan lainnya. Untuk mengintegrasikan aplikasi *Chatbot* ke situs web, *Google Dialogflow* menyediakan cara integrasi seperti *Dialogflow Messenger* dan *Dialogflow Web Demo*.

Penelitian ini akan menggunakan cara integrasi yang bernama *Dialogflow Messenger* karena telah mendukung berbagai macam tipe respon jawaban seperti gambar, link, tombol, dan lainnya yang disebut dengan *rich response messages* [15], serta memiliki desain tampilan yang lebih modern dibandingkan *Dialogflow Web Demo*.

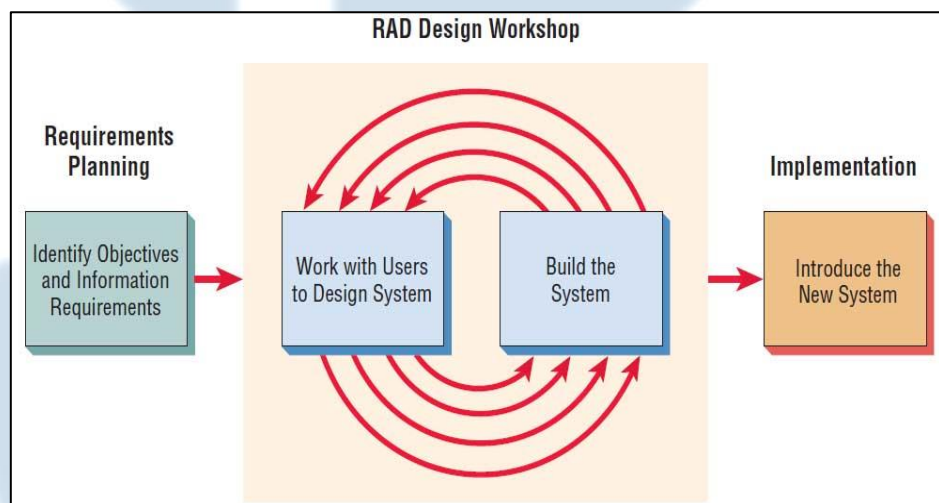


Gambar 2.5 Tampilan Dialogflow Messenger

2.1.7 Rapid Application Development

Rapid Application Development atau yang disingkat dengan *RAD* merupakan salah satu metode dalam pengembangan sistem aplikasi atau *SDLC (System Development Life Cycle)* yang digunakan untuk mengembangkan suatu perangkat lunak tambahan yang menggunakan siklus yang relatif singkat [17]

Penggunaan Siklus *Rapid Application Development* membutuhkan waktu dari 30 hari sampai dengan 90 hari dengan melakukan pembangunan atau pembuatan sistem perangkat lunak yang dimana proses analisa dan perancangan dilakukan dengan klien hingga mencapai kesepakatan [18].



Gambar 2.6 Tahapan Siklus Rapid Application Development

2.1.8 Alpha Testing

Alpha Testing merupakan sebuah proses pengujian yang termasuk ke dalam metode pengujian *User Acceptance Test*. Tujuannya adalah untuk dapat mengidentifikasi permasalahan-permasalahan yang mungkin dapat muncul [19].

Pengujian *Alpha Testing* umumnya masih dilakukan oleh tim pengembang dan masih bersifat *internal*. Pengujian *Alpha Testing* dapat menggunakan 2 macam metode yakni pengujian *Black-Box Testing* dan *White-Box Testing*, penelitian ini akan menggunakan pengujian *Black-Box Testing* untuk menguji *Intent* yang telah dibuat apakah dapat memberikan respon jawaban yang tepat dengan berbagai skenario pertanyaan.

2.1.9 Closed Beta Testing

Closed Beta Testing merupakan salah satu cara dalam melakukan *Beta Testing* yang termasuk ke dalam metode pengujian *User Acceptance Test* yang dilakukan oleh *real user* untuk melakukan uji coba dari aplikasi secara langsung [19].

Tujuan dari *Beta Testing* adalah untuk mendapatkan *feedback* terkait kepuasan dan perspektif dari *real user* yang telah menguji aplikasi. Pengujian akan dilakukan dengan cara mengajak beberapa *real user* yang dalam kasus ini adalah para mahasiswa yang sedang atau pernah menggunakan situs jurnal untuk pencarian artikel jurnal pada keperluan tugas kuliah.

Setelah pengujian, *real user* akan diminta untuk mengisi kuesioner yang berisi pertanyaan-pertanyaan agar *real user* dapat memberikan penilaian terhadap aplikasi yang telah dibangun. Jawaban untuk pertanyaan dari kuesioner akan menggunakan metode Skala *Likert* yang merupakan skala untuk dapat melakukan pengukuran terkait pendapat dan penilaian seseorang terhadap sesuatu [20]

2.2 Penelitian Terdahulu

Berikut adalah 5 penelitian terdahulu yang serupa mengenai pembuatan aplikasi *chatbot* yang menggunakan layanan *Google Dialogflow* yang dapat dilihat pada tabel 2.1 :

Tabel 2.1 Penelitian Terdahulu

Judul Jurnal	Penulis	Nama Jurnal, Tahun	Hasil Penelitian
Perancangan Chatbot Hotel dengan Model Natural Language Processing Chatbot dan Button Based Chatbot [21]	I Gede Ryoga Kusnanda, I Made Sukarsa, Anak Agung Ngurah Hary Susila	JITTER- Jurnal Ilmiah Teknologi dan Komputer, 2022	Hasil dari penelitian ini adalah perancangan dan implementasi aplikasi <i>chatbot</i> yang bersifat <i>Button-Based</i> tentang hotel dengan model <i>NLP</i> memberikan hasil yang baik karena mampu untuk memproses seluruh pesan masukan dari pengguna meskipun terdapat kesalahan dalam pengetikan.
Perancangan Chatbot Menggunakan Dialogflow Natural Language Processing (Studi Kasus: Sistem Pemesanan pada Coffee Shop) [22]	Albert Yakobus Chandra, Didik Kurniawan, Rahmat Musa	Jurnal Media Informatika Budidarma, 2020	Hasil dari penelitian ini adalah aplikasi <i>chatbot</i> yang diintegrasikan menggunakan <i>Dialogflow Web Demo</i> dibuat menggunakan metode pengembangan <i>Waterfall</i> yang dimana pelayanan yang diberikan oleh <i>chatbot</i> akan optimal selama pengguna mengirim pesan yang sesuai dengan pemetaan jawaban yang ada di daftar <i>intent</i> yang telah dibuat.
Rancang Bangun Fitur Chatbot Customer Service Menggunakan Dialogflow [23]	Jansen Wiratama, Samuel Ady Sanjaya, Victor Ilyas Sugara	KOMPUTA SI: Jurnal Ilmiah Ilmu Komputer dan Matematika, 2022	Hasil dari penelitian ini adalah aplikasi <i>chatbot</i> yang telah diterapkan dapat membuat tim <i>customer service</i> untuk bekerja lebih efektif karena adanya proses otomatisasi untuk membalas pertanyaan-pertanyaan sehingga tidak menumpuk atau terabaikan oleh tim <i>customer service</i> .

Judul Jurnal	Penulis	Nama Jurnal, Tahun	Hasil Penelitian
Pengembangan Chatbot Percakapan Bahasa Inggris Menggunakan Dialogflow [24]	Aliv Faizal Muhammad, Farah Adila	JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), 2021	Hasil dari penelitian ini adalah semua <i>intent</i> yang dimiliki <i>chatbot</i> berbahasa Inggris yang diintegrasikan dengan <i>Dialogflow Web Demo</i> mencapai semua indikator dan memiliki tingkat akurasi 100%. Tingkat akurasi akan mengalami peningkatan apabila terdapat variasi dari kumpulan-kumpulan data sebagai pertanyaan (<i>training phrases</i>).
Pengembangan Chatbot Yanies Cookies Untuk Pemesanan Kue Kering Berbasis Dialogflow[25]	Sultan Saladdin, Issa Arwani, Bayu Rahayudi	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, 2020	Hasil dari penelitian ini adalah implementasi aplikasi <i>chatbot</i> pada platform <i>Facebook Messenger</i> dan mendapat skor 86 dari 100 untuk uji kelayakan yang diuji oleh 5 responden.

Pada penelitian Kusnanda dkk., aplikasi *Chatbot* yang dibuat menggunakan *Dialogflow* untuk layanan informasi tentang hotel secara *Button-Based* [21] Hal ini serupa dengan apa yang akan dilakukan pada penelitian ini yaitu menggunakan *Button-Based* sebagai input yang dapat digunakan pengguna yaitu dengan cara memilih tombol yang tersedia. Pada penelitian yang dilakukan oleh Chandra dkk., aplikasi *Chatbot* dibuat menggunakan metode *Waterfall* serta cara integrasi menggunakan *Dialogflow Web Demo* [22] yang hanya mendukung respon jawaban yang bersifat teks. Hal ini berbeda pada penelitian ini karena penelitian ini menggunakan metode *Rapid Application Development* serta cara integrasi yang baru yaitu menggunakan *Dialogflow Messenger* yang lebih mendukung berbagai macam tipe respon jawaban seperti teks, suara, gambar, tombol, link. Pada penggunaan bahasa, aplikasi *Chatbot*

yang akan dibuat akan mendukung penggunaan bahasa Inggris. Hal ini dikarenakan situs jurnal yang memang didasari oleh penggunaan bahasa Inggris dan juga sesuai dengan permintaan dari pengelola situs.

