



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. *Artificial Intelligence*

Kecerdasan buatan adalah kemampuan komputer digital atau robot yang dikendalikan komputer untuk melakukan tugas-tugas yang umumnya terkait dengan makhluk yang cerdas. Istilah ini sering diterapkan pada proyek pengembangan sistem yang memanfaatkan kemampuan untuk berpikir, menemukan makna dari suatu kalimat, menggeneralisasikan, atau belajar dari pengalaman (Copeland, 2018).

Kecerdasan buatan merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia (Rich, Knight, & Nair, 2009).

Kecerdasan buatan ialah cabang dari ilmu komputer yang memproses informasi berdasarkan metode heuristic atau dengan berdasarkan sejumlah aturan yang berfungsi untuk memudahkan manusia dalam melakukan tugas-tugas yang biasanya dilakukan oleh manusia.

2.2. *Chatbot*

Program *chatbot* pertama ditulis oleh Joseph Weizenbaum, profesor MIT pada tahun 1966. Pada waktu itu tentu saja *chatbot* yang dibuat masih sangat sederhana. Meskipun perkembangan kecerdasan buatan saat ini sangat pesat dan canggih, namun *chatbot* tetap mempertahankan kedudukannya dalam dunia

Artificial Intelligence. (Campbell-Kelly, 2008) *Chatbot* adalah sebuah simulator komunikasi interaktif yang dapat berupa program komputer yang mampu berdialog dengan penggunanya dalam bahasa alami dengan memanfaatkan *natural language processing*. Karena *chatbot* adalah sebuah program, dan bukan robot (*chatbot* tidak memiliki tubuh dan tidak memiliki mulut fisik sehingga tidak dapat berbicara seperti manusia), maka yang dimaksud dengan dialog antar manusia sebagai pengguna *chatbot* dilakukan dengan cara mengetik apa yang akan dibicarakan dan *chatbot* akan memberikan *feedback* kepada pengguna atas pertanyaannya. (Wikipedia: Chatbot, 2016)

2.3. Text Editor

Text Editor adalah suatu program komputer yang yang memungkinkan penggunanya untuk membuat, mengubah, atau mengedit *plain-text*. *Text editor* dapat digunakan untuk membuat program-program komputer, mengedit *source code* bahasa pemrograman, serta membuat halaman web, dan pembuatan aplikasi tertentu. (Haryati, 2012) Secara umum *Text Editor* sering digunakan untuk tujuan pemrograman.

2.3.1. Visual Studio Code

Visual Studio Code adalah *source code editor* yang ringan untuk diproses oleh ram namun dapat digunakan untuk banyak keperluan yang berupa *desktop application* dan tersedia untuk Windows, macOS dan Linux. Hadir dengan dukungan *built-in* untuk *JavaScript*, *TypeScript* dan *Node.js* dan memiliki ekosistem ekstensi untuk bahasa lain (seperti *C ++*,

C #, Java, Python, PHP, Go) dan *runtimes* seperti *.NET* dan *Unity* (Documentation: Visual Studio Code).

2.4. Node.js

Node.js adalah *cross-platform JavaScript run-time environment* yang bersifat *open-source* dan berfungsi untuk menjalankan kode *JavaScript* pada sisi *server*. Node.js memungkinkan *developer* untuk menggunakan *JavaScript* untuk melakukan proses *scripting* di sisi *server*, yang berguna untuk membuat konten halaman web dinamis sebelum halaman dikirim ke *web browser* pengguna. Maka dari itu, Node.js dapat menyatukan pengembangan aplikasi *web* menggunakan satu bahasa pemrograman, daripada bahasa yang berbeda untuk *server-side* dan *client-side scripting*. (Patel, 2018)

2.5. AngularJS

AngularJS adalah *open-source front-end web application framework* yang berbasis *JavaScript*. Komponen *JavaScript* melengkapi kebutuhan *Apache Cordova*, yaitu sebuah *framework* yang digunakan untuk mengembangkan *cross-platform mobile application*. Yang bertujuan untuk memudahkan proses pengembangan dan *testing* dari aplikasi dengan menyediakan sebuah *framework* bagi *client-side model-view-controller (MVC)* dan *model-view-viewmodel (MVVM)*. (Docs: FAQ)

2.6. Apache Cordova

Apache Cordova adalah sebuah *open-source mobile development framework*. Yang dapat memungkinkan pengguna untuk menggunakan *standard web technologies* seperti HTML5, CSS3, dan JavaScript untuk melakukan *cross-platform development*. Aplikasi dijalankan melalui *wrappers* yang ditujukan kepada tiap *platform*, dan mengikuti standarisasi dari *API bindings* untuk mengakses tiap kemampuan perangkat seperti sensor, data, status jaringan, dan lain-lain.

Apache Cordova digunakan untuk:

- a. *Mobile Developer* yang ingin memperluas jangkauan aplikasi yang dimiliki ke *platform* lain, tanpa harus melakukan implementasi ulang dengan Bahasa dan *tool set* yang digunakan oleh *platform* tersebut.
- b. Seorang *Web Developer* yang ingin menerapkan aplikasi web yang dikemas untuk didistribusikan di berbagai portal aplikasi.
- c. Seorang *Mobile Developer* yang tertarik untuk mencampure komponen aplikasi *native* dengan *WebView* yang dapat mengakses *API* pada tingkat perangkat, atau jika pengembang ingin mengembangkan tampilan *plugin* antara komponen *native* dan *WebView*. (Documentation: Apache Cordova)

2.7. Syntatically Awesome Style Sheets (SASS)

Syntatically Awesome Style Sheets (SASS) adalah *preprocessor scripting language* yang dapat menafsirkan atau menyatukan *scripting language* ke dalam

Cascading Style Sheets (CSS). *SassScript* adalah *scripting language*. *Sass* terdiri dari dua buah *syntax*. *Syntax* asli yang disebut "*the indented syntax*" yang menggunakan *syntax* mirip dengan *HAML*. *SASS* bekerja menggunakan *indentation* untuk memisahkan *code blocks* dan *newline characters* untuk memisahkan "*rules*". *Syntax* yang kedua ialah "*SCSS*", *SCSS* bekerja menggunakan "*block formatting*" seperti *CSS*, yang menggunakan simbol *braces* untuk menunjukkan perbedaan antara *code blocks* dan titik koma untuk memisahkan garis dalam satu blok. (SASS Lang: Guide, 2009)

2.8. Ionic Framework

Ionic adalah *Software Development Kit* yang bersifat *open-source* untuk pengembangan *hybrid mobile application*. *Ionic Framework* di-rilis pada tahun 2013 dan dibangun diatas *AngularJS* dan *Apache Cordova*. *Ionic* menyediakan layanan dan perangkat untuk pengembangan *hybrid mobile application* menggunakan teknologi *web* seperti *CSS*, *HTML5*, dan *SASS*. Aplikasi dapat dibuat dengan menggunakan teknologi *web* tersebut dan di distribusikan melalui *native app stores* yang dapat di-*install* pada perangkat dengan memanfaatkan *Cordova*. (*Ionic Framework: Docs*)

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

2.9. Database

Database merupakan kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi. (Sultan, 2015)

2.11.1. Firebase

Firebase adalah suatu layanan dari Google yang digunakan untuk mempermudah para pengembang dengan menyediakan layanan berupa *cloud messaging, authentication, remote config, realtime database, storage, dan hosting*. (Tolle, 2017)

2.11.2. MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (DBMS) yang *multithread*, dan *multi-user*. MySQL adalah implementasi dari sistem manajemen basisdata relasional (RDBMS).

2.12. Dialogflow

Dialogflow adalah *Google-owned developer* untuk teknologi interaksi manusia dengan komputer yang didasarkan pada percakapan *natural language*.

Perusahaan ini dikenal karena menciptakan *assistant*, teman virtual untuk smartphone Android, iOS, dan Windows Phone yang melakukan tugas dan menjawab pertanyaan pengguna dalam bahasa alami. Dialogflow juga telah menciptakan *natural processing engine* yang menggabungkan konteks percakapan

seperti riwayat dialog, lokasi, dan preferensi pengguna. (Wikipedia: Dialogflow, 2018)

2.13. Agile Development Method

Agile Development Method adalah *software development methodologies* yang pengembangannya bersifat *iterative*, dimana persyaratan dan solusinya dapat berubah-ubah sesuai dengan kolaborasi antar anggota dalam tim. *Agile methods* merupakan metode yang mengembangkan proses manajemen proyek yang mendorong seringnya diadakan inspeksi dan adaptasi, filosofi kepemimpinan yang mengangkat kerja tim, pengaturan diri, dan akuntabilitas.

2.14. Scrum Framework

Scrum adalah bagian dari *agile methods*, yang merupakan *framework* dari proses *agile development*, dan *Scrum Framework* adalah *framework* yang paling sering digunakan. *Scrum framework* digunakan untuk meningkatkan kualitas dari penyampaian pesan kepada antar anggota dan juga klien, memudahkan perubahan yang terjadi, menyediakan perkiraan yang lebih baik dengan waktu yang lebih singkat, dapat mengontrol jadwal proyek lebih baik.

1. Peran-peran dalam *Scrum Framework* :

a. *Product Owner*

Product Owner bertanggung jawab untuk memaksimalkan pengembalian investasi (*Return on Investment*) dengan

mengidentifikasi fitur produk , menerjemahkannya ke dalam daftar yang berisi tingkat prioritas suatu fitur, menentukan daftar utama untuk *sprint* berikutnya , memprioritaskan dan memperbaiki daftar.

b. Team

Team bertugas untuk membangun produk yang akan klien gunakan, contohnya membuat website atau situs web. *Team* dalam *scrum* adalah lintas fungsional dan mencakup semua keahlian yang diperlukan untuk membuat suatu produk yang berpotensi untuk dikirim setiap *sprint*. *Team* khusus biasanya terdapat pada produk perangkat lunak termasuk *programmer*, merancang interface dan penguji. *Team* juga bertugas untuk membangun produk dan memberikan ide-ide untuk pemilik produk sehingga produk bisa semakin lebih baik.

c. Scrum Master

Scrum Master membantu *team* dalam membuat produk dan menerapkan *scrum* untuk mencapai *business value*. *Scrum Master* bukanlah seorang *manager* dalam sebuah *team* tetapi melayani *team*, menjaga *team* dari gangguan luar, mendidik serta membimbing *product owner* dan *team* dalam keahlian menggunakan *scrum*. *Scrum Master* memastikan setiap orang dalam *team* (termasuk *product owner*, dan para pihak manajemen) memahami dan mengikuti praktik-praktik *Scrum*.

2. Aktivitas dalam *Scrum Framework*:

a. *Product Backlog*

Product backlog adalah daftar dari pekerjaan yang akan dilakukan dan daftar yang dibuat tersebut dapat selalu bertambah, diatur kembali tingkat prioritasnya, atau dihapus oleh *Product Owner* tergantung dari kebutuhan *user/client*, ide baru, perubahan kompetisi, dan sebagainya. Terdapat dua variable dalam membantu memprioritaskan *Product Backlog* yaitu *business value* dan *effort*. *Product Backlog* biasa disebut juga sebagai “*User Stories*”.

Backlog ini berguna untuk menentukan *Release Backlog* yaitu daftar pekerjaan yang diperuntukan untuk produk yang akan di rilis. Dalam hal ini pekerjaan yang terdapat dalam *Release Backlog* harus diambil dari *Product Backlog*. Dalam menentukan *Release Backlog*, *Product Owner* harus menimbang pekerjaan mana yang menghasilkan *business value* yang lebih menguntungkan dan *effort* yang lebih sedikit guna memaksimalkan *return of investment*. Dalam *release backlog scrum master* harus mengestimasi waktu dan memprioritaskan pekerjaan yang harus dilakukan terlebih dahulu guna memperkirakan tanggal produk tersebut dirilis.

b. *Sprint Planning*

Sprint Planning adalah aktivitas yang dihadiri oleh *product owner*, *scrum master*, dan *scrum team*. Dalam aktifitas ini *product owner* akan menjelaskan prioritas-prioritas dari pekerjaan yang akan dilakukan.

c. *Sprints*

Sprints merupakan aktivitas pengerjaan produk pada metodologi *Scrum*. Umumnya setiap *Sprints* dilakukan dalam kurun waktu 30 hari dan *sprints* tidak dapat ditambah.

d. *Daily Scrum Meetings*

Scrum Meeting merupakan pertemuan yang rutin dilakukan perhari untuk mengevaluasi apa yang dikerjakan, hambatan yang ada, dan target penyelesaian untuk bahan *meeting* selanjutnya.

Scrum Meeting memiliki beberapa tipe yaitu *Sprint Planning Meeting*, *Daily Scrum Meeting*, *End Meeting*.

Dalam aktivitas ini kinerja tim akan diukur dengan menggunakan *burndown charts* dimana jika terjadi hambatan yang mengakibatkan pekerjaan harus dilakukan lebih lama maka garis *burndown* akan naik dan jika suatu pekerjaan selesai maka garis *burndown* akan turun sampai ke titik 0 yang menandakan seluruh pekerjaan telah selesai. Pada *burndown charts* terdapat *idealized line* yang mengindikasikan idealnya sisa waktu total pekerjaan yang harus dilakukan perharinya.

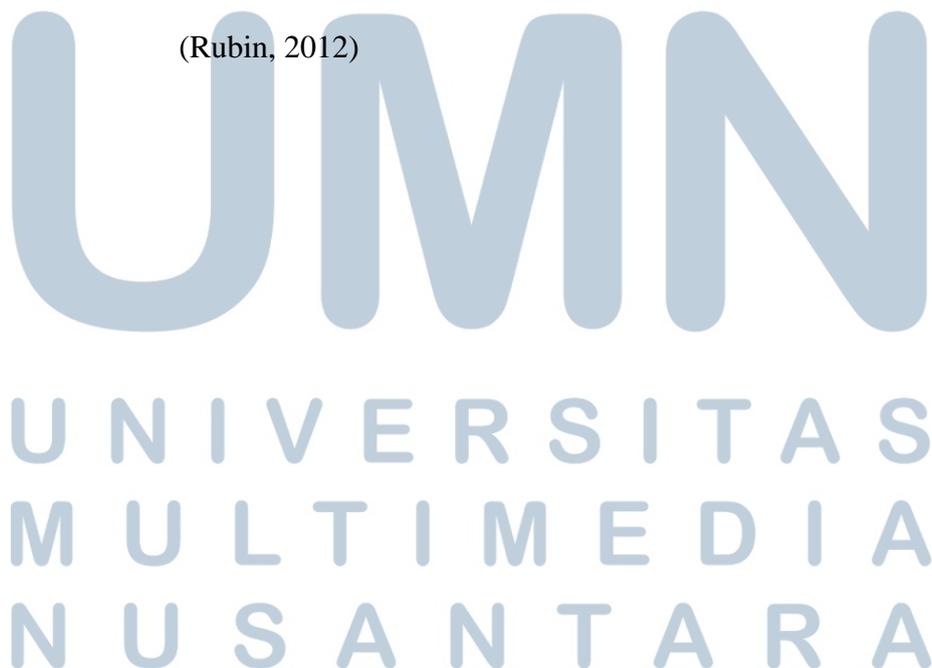
e. *Sprint Review and Retrospective*

Sprint review and retrospective adalah aktivitas melakukan review dan menginspeksi sebuah produk.

Sprint review melibatkan *scrum team*, *sponsor*, *stakeholder*, dan *customer*. Dalam *sprint review* ini *scrum team* akan mendemokan hasil pekerjaan mereka. Dari demo ini akan dihasilkan *feedback* yang berguna untuk proses *scrum* selanjutnya.

Sprint retrospective adalah aktivitas mendiskusikan apa saja yang tidak berjalan pada *Scrum* dan terkait pembahasan secara teknis. Tujuannya adalah untuk melakukan evaluasi terhadap *scrum team*. Setelah *sprint retrospective* selesai seluruh fase dulangi kembali dimuali dengan *sprint planning* untuk *sprint* berikutnya. Seluruh fase dapat diulangi sampai project selesai.

(Rubin, 2012)



2.15. Penelitian Sebelumnya

Tabel 2.1. Referensi 1 Penelitian Sebelumnya

Nama Peneliti	Anirudh Khanna, Bishwajeet Pandey, Kushagra Vashishta, Kartik Kalia, Bhale Pradeepkumar dan Teerath Das
Judul Penelitian	A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence
Tahun	2015
Sumber	<i>International Journal of u- and e-Service, Science and Technology</i> , 277-284
Objek Penelitian	<i>ALICE Bot, FUTURE Bot, dan SARANG Bot</i>
Metode Penelitian	Dalam penelitian tersebut, penelitian dilakukan menggunakan bahasa pemrograman C++ dalam proses rancang bangun <i>chatbot FUTURE</i> , dan <i>Artificial Intelligence Markup Language</i> untuk merancang dan membangun <i>chatbot SARANG</i> . Teknik yang digunakan oleh peneliti adalah <i>pattern matching</i> dengan metode <i>approximate string matching</i> , dimana <i>chatbot</i> akan mencari suatu pola dalam kalimat dan menganalisa pola tersebut dan digunakan kembali untuk kalimat yang memiliki kemiripan pola untuk membantu <i>chatbot</i> dalam memberikan <i>response</i> . Dalam perancangan <i>bot</i> , peneliti membuat 7 <i>layer</i> pada aplikasi, yaitu <i>Introductory Screen</i> yang merupakan <i>layer</i> dimana akan memperkenalkan program kepada pengguna yang berfungsi untuk menanyakan informasi umum mengenai diri pengguna dan tujuan dari pembicaraan yang akan dilakukan. Hal tersebut dilakukan guna untuk disimpan dalam <i>database</i> dalam bentuk <i>datafile</i> dan dipergunakan pada tahap <i>Data Handling</i> . Selanjutnya <i>layer Chat Interface dan Manager</i> , pada <i>layer</i> ini pengguna dapat memberikan <i>input/trigger</i> kepada <i>chatbot</i> dan <i>text-</i>

<p>Metode Penelitian</p>	<p><i>input</i> akan di-<i>fetch</i> untuk mencari jawaban terbaik dari <i>database</i>. Lalu <i>layer Database</i>, <i>database</i> merupakan otak dari <i>chatbot</i> yang berisikan daftar jawaban yang dibuat secara manual dan tersusun. <i>Layer Productivity Application</i>, merupakan fitur tambahan pada <i>chatbot</i> yang berisikan program sederhana seperti kalkulator, kamus, dan juga permainan sederhana. <i>Layer Ambiguity Handling</i>, seperti pada namanya <i>layer</i> ini berfungsi untuk menangani pertanyaan yang tidak memiliki jawaban pada <i>database</i>. <i>Layer Data Handling</i>, <i>layer</i> ini berfungsi agar pengguna tidak perlu memberikan informasi umum mengenai dirinya setiap kali menggunakan program <i>chatbot</i>, karena data tersebut akan disimpan dalam <i>database</i> untuk memodifikasi jawaban yang dapat menyesuaikan pengguna. <i>Layer Error Handling</i>, <i>layer</i> ini berfungsi untuk menanggulangi pesan <i>error</i> agar <i>response chatbot</i> tetap dapat memuaskan pengguna, seperti jika pengguna memasukan perhitungan pada <i>layer chat interface</i> maka <i>input</i> dari pengguna akan dimasukan ke <i>layer productivity application</i> yang berupa kalkulator dan hasil perhitungan akan dikembalikan kepada pengguna di <i>layer chat interface</i>. Pada tahap implementasi, <i>FUTURE Bot</i> menggunakan Bahasa C++ dan <i>SARANG Bot</i> menggunakan <i>AIML</i>.</p>
<p>Hasil Penelitian</p>	<p>Hasil dari penelitian tersebut memperlihatkan bahwa <i>chatbot</i> yang dibuat dengan <i>Artificial Intelligence Markup Language (SARANG Bot)</i> memiliki waktu pengembangan yang lebih singkat, dengan tingkat akurasi jawaban sebesar 80%. Hal tersebut mengindikasikan <i>chatbot FUTURE</i> tidak lebih unggul dari <i>chatbot SARANG</i> karena pengembangan <i>chatbot SARANG</i> memanfaatkan <i>AIML</i> dengan <i>pattern matching method</i>, sedangkan <i>chatbot</i></p>

Hasil Penelitian	<i>FUTURE</i> menggunakan bahasa C++ yang merupakan <i>object-oriented programming language</i> . Dan hasil penelitian akan keandalan <i>Turing Test</i> adalah <i>Turing Test</i> bukan merupakan metode pengukuran yang tepat untuk menilai apakah sebuah mesin atau <i>bot</i> dapat dianggap entitas yang cerdas, karena pada test yang dilakukan terkadang mesin tanpa kecerdasan buatan dapat melakukan tugas-tugas manusia lebih cepat dan akurat daripada entitas cerdas itu sendiri (manusia).
------------------	---

Sumber: (Khanna, Pandey, Vashishta, Kalia, Pradeepkumar, & Das, 2015)

Tabel 2.2. Referensi 2 Penelitian Sebelumnya

Nama Peneliti	Zifora Nur Baiti, Fresy Nugroho, ST.,MT
Judul Penelitian	Aplikasi Chatbot “MI3” Untuk Informasi Jurusan Teknik Informatika Berbasis Sistem Pakar Menggunakan Metode <i>Forward Chaining</i>
Tahun	2016
Sumber	<i>Paper Saintek</i> UIN, 178-183
Objek Penelitian	Eliza Bot dan MI3
Metode Penelitian	Dalam penelitian tersebut, hasil uji coba di atas, membuktikan bahwa metode <i>forward chaining</i> mampu menyelesaikan permasalahan untuk mengenali kalimat yang dimasukkan, apakah kalimat tersebut merupakan bentuk dari salah satu kata kunci atau bukan. Meskipun hampir mendekati sempurna, namun hasil persentase sebesar 92,5% juga membuktikan bahwa aplikasi ini masih memerlukan penyempurnaan. Kelengkapan kata kunci sangat berpengaruh pada keberhasilan aplikasi ini. hasil implementasi dan uji coba yang telah peneliti lakukan, memperoleh kesimpulan bahwa algoritma Sistem Pakardengan metode <i>forward chaining</i> merupakan algoritma yang baik sebagai penyelesaian proses pencarian jawaban berdasarkan kata kunci dari pertanyaan

	user.
Hasil Penelitian	Hasil dari penelitian tersebut membuktikan bahwa aplikasi chatbot sangat cocok untuk menjadi media informasi, sebagaimana terlihat dari hasil uji coba persentase sebesar 92%. Dari hasil penelitian penulis menyimpulkan bahwa aplikasi ini sangat membantu mahasiswa sebagai media informasi jurusan Teknik Informatika.

Sumber: (Baiti, 2016)

Tabel 2.3. Referensi 3 Penelitian Sebelumnya

Nama Peneliti	Bibek Behera
Judul Penelitian	<i>Chappie - A Semi-Automatic Intelligent Chatbot</i>
Tahun	2016
Sumber	<i>Chatbot Paper and Learning Journal:</i> https://www.chatbots.org/paper/chappie_-_a_semi-automatic_intelligent_chatbot/
Objek Penelitian	Chappie Bot
Metode Penelitian	Dalam penelitian tersebut, Chappie bot merupakan bot yang diciptakan untuk menjadi asisten pribadi pengguna. Karena banyak kegiatan yang jika dilakukan oleh manusia menjadi tidak efisien dan membutuhkan waktu lama untuk memproses satu permintaan seperti memesan tiket, memesan makanan, mendapatkan layanan kecantikan dll. Ini mengurangi throughput dan kinerja bisnis secara drastis. Jelas ada peningkatan permintaan otomatisasi obrolan karena a) menghilangkan faktor manusia dan b) dapat memberikan layanan 24 jam yang akan memiliki efek multiplikatif pada pendapatan. Chatbot, Chappie lahir sebagai persyaratan bisnis untuk mengotomatisasi asisten pribadi atau petugas. Saat ini Chappie digunakan sebagai agen perutean di mana Chappie dapat mengklasifikasikan kebutuhan pengguna menjadi salah satu layanan yang disediakan oleh bisnis berdasarkan beberapa obrolan pertama dan kemudian

Metode Penelitian	mentransfernya ke pakar agen dalam layanan itu. Ia menggunakan pemrosesan bahasa alami (nlp) dengan metode <i>multi labelling classification</i> untuk menganalisis obrolan dan mengekstrak maksud pengguna dengan skor yang mirip dengan WIT. Kemudian ia menggunakan informasi ini dan AIML (Bahasa Mark-up Inteligensi Buatan) untuk melakukan percakapan dengan pengguna. Ini adalah perbedaan nyata antara Chappie dan <i>chatbot</i> yang ada seperti ALICE, yang hanya berfungsi pada AIML.
Hasil Penelitian	Hasil dari penelitian tersebut membuktikan bahwa aplikasi chatbot chappie dapat melakukan obrolan semi-otomatis yang beralih ke mode manual setelah mampu memahami maksud pengguna dan jenis layanan yang diperlukan olehnya. Saat ini kami membiarkan Chappie ditayangkan satu jam sekali setiap hari dan kami mencatat respons serta menguji kasus untuk klasifikasi. Sejauh ini hasilnya menggembirakan karena orang belum dapat membedakan Chappie dari manusia. Juga telah dapat rute dengan benar pada beberapa kesempatan. Setelah kami memperoleh jumlah akurasi yang memadai dalam perutean dan koherensi obrolan, itu akan digunakan setiap 24 jam.

Sumber: (Behera, 2016)

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