

OTOMATISASI LAYANAN FREQUENTLY ASK QUESTIONS BERBASIS NATURAL LANGUAGE PROCESSING PADA TELEGRAM BOT

Hani Husamuddin⁽¹⁾, Dessyanto Boedi Prasetyo⁽²⁾, Heru Cahya Rustamadji⁽³⁾
Jurusan Teknik Informatika Fakultas Teknik Industri UPN "Veteran" Yogyakarta
Jl. Babarsari 2 Tambakbayan Yogyakarta
email: hani.husam@gmail.com⁽¹⁾, dess@upnyk.ac.id⁽²⁾, herucr@upnyk.ac.id⁽³⁾

Abstract

Kitabisa or kitabisa.com is the most popular online fundraising and donating platform in Indonesia. Kitabisa.com provides frequently asked questions (FAQs) to help visitors find out about Kitabisa.com and how to use services that Kitabisa.com provides. In this study, it will be proposed a solution to maximize the automation of the FAQ service of Kitabisa.com using Natural Language Processing (NLP). This service is carried out with the bot feature on Telegram messenger which is designed based on NLP using TensorFlow technology. Thus, this chatbot acts as a customer service that will answer the questions asked. The results of this study are chatbots using the NLP method to respond according to the context of what the user asks with an accuracy of 73%.

Keywords : *Chatbot, Natural Language Processing, TensorFlow, Telegram*

Abstrak

Kitabisa atau kitabisa.com adalah *platform* untuk menggalang dana dan berdonasi secara online terpopuler di Indonesia. Kitabisa.com menyediakan *frequently ask questions* (FAQ) untuk membantu visitor mengetahui mengenai Kitabisa.com dan bagaimana menggunakan layanan yang Kitabisa.com sediakan. Pada penelitian ini akan mengusulkan satu solusi untuk memaksimalkan otomatisasi layanan FAQ Kitabisa.com berbasis *Natural Language Processing* (NLP). Layanan ini dilakukan dengan fitur *bot* pada *messenger* Telegram yang dirancang berbasis NLP menggunakan teknologi TensorFlow. Dengan demikian, *chatbot* ini bertindak sebagai *customer service* yang akan menjawab pertanyaan-pertanyaan yang diajukan. Hasil dari penelitian ini adalah *chatbot* dengan menggunakan metode NLP dapat memberikan respon sesuai dengan konteks atas apa yang *user* tanyakan dengan akurasi 73%.

Kata kunci: *Chatbot, Natural Language Processing, TensorFlow, Telegram*

1. PENDAHULUAN

Kitabisa atau kitabisa.com adalah *platform* untuk menggalang dana dan berdonasi secara *online (crowdfunding)* terpopuler di Indonesia. Kitabisa.com menyediakan media informasi *help center* untuk membantu *visitor* mengetahui mengenai Kitabisa.com dan bagaimana menggunakan layanan yang Kitabisa.com sediakan. *Help center* Kitabisa.com berisi tentang *list* pertanyaan yang sering *visitor* tanyakan (*Frequently Ask Questions*) dan juga fitur *search engine* yang akan memudahkan *visitor* untuk mencari pertanyaan yang sesuai dengan yang ingin *visitor* tanyakan. Namun hasil pencarian dari *search engine* ini masih berupa *list* pertanyaan yang mendekati dengan masukan dari *visitor*, yang mana *visitor* masih harus melakukan penelusuran untuk mencari pertanyaan yang mana yang sesuai dengan yang ingin *visitor* tanyakan.

Pada penelitian ini akan mengusulkan satu solusi untuk memaksimalkan otomatisasi layanan media informasi *Frequently Ask Questions* (FAQ) pada *help center* Kitabisa.com berbasis *Natural Language Processing* (NLP). Layanan ini dilakukan dengan *messenger* Telegram. Orang-orang akan berkomunikasi dengan Bot Telegram yang dirancang berbasis NLP menggunakan teknologi TensorFlow. Dengan demikian, *chatbot* ini bertindak sebagai *customer service* yang akan

menjawab pertanyaan-pertanyaan yang diajukan mengenai informasi pada *help center*. TensorFlow digunakan untuk membuat *neural* model yang mana *bot* akan dilatih berdasarkan *intent file* (*help center* Kitabisa.com) sehingga *bot* ini dapat mempertahankan konteks dan dapat memberikan respon berdasarkan konteks.

Chatbot (Robot chatting) yaitu program komputer yang mampu mensimulasikan percakapan dengan menggunakan bahasa alami (*natural language*), *chatbot* mengadopsi pengetahuan manusia ke komputer, agar komputer dapat melakukan percakapan dengan pengguna. *Chatbot* dapat diimplementasikan untuk bidang komersial, pendidikan, *entertainment* dan sektor pelayanan publik (Kerlyl, et al., 2007).

Telegram merupakan alternatif layanan aplikasi perpesanan untuk ponsel (*mobile*) maupun dekstop yang berbasis *cloud* dengan tingkat keamanan tinggi serta kecepatan aksesnya. Aplikasi *instant messaging* tersebut tersedia untuk berbagai *device* seperti ponsel yang memiliki sistem operasi Android, iOS, Windows Phone, Ubuntu Touch. Telegram tidak hanya dapat digunakan pada perangkat *mobile*, tetapi juga dapat berjalan pada sistem *desktop* seperti Windows, OS X, dan Linux. Meskipun Telegram terlihat sederhana, tetapi merupakan aplikasi non-berbayar (*free*) dan memiliki fitur yang lebih unggul dibandingkan dengan aplikasi *instant messaging* lainnya. Telegram diklaim sebagai aplikasi yang aman karena salah satunya memiliki fitur dimana menyediakan pilihan pesan *end to end* yang dienkripsi serta dapat hancur dengan sendirinya dalam jangka waktu tertentu (Telegram, n.d.). Bot yang dimiliki Telegram berdasar pada AI dan *mechine learning* dalam skala atau level rendah. Pengguna dapat berinteraksi dengan *bot* dengan mengirimkan perintah atau *command* yang kemudian *bot* akan membalas dengan tepat sesuai dengan informasi yang tersimpan pada *database*.

Natural Language Processing atau pengolahan bahasa alami, biasanya disingkat dengan NLP, merupakan bidang kecerdasan buatan dimana komputer didesain untuk dapat berkomunikasi dengan manusia menggunakan bahasa alami, seperti Bahasa Indonesia. NLP tidak bertujuan untuk mentransformasikan bahasa yang diterima dalam bentuk teks atau suara menjadi data digital dan/atau sebaliknya, melainkan bertujuan untuk memahami arti dari kalimat yang diberikan dalam bahasa alami dan memberikan respon yang sesuai, misalnya dengan melakukan suatu aksi tertentu atau menampilkan data tertentu (Ridwan, 2013).

TensorFlow adalah *open source library* dari Google untuk implementasi pembelajaran mesin berskala besar. TensorFlow, dalam arti sebenarnya, adalah penerus DistBelief, yang merupakan kerangka kerja perangkat lunak sebelumnya yang dirilis oleh Google yang mampu memanfaatkan cluster komputasi dengan ribuan mesin untuk melatih model-model yang besar. TensorFlow menawarkan pergerakan tugas komputasi yang sangat intensif dari CPU ke *platform* berorientasi GPU yang heterogen, dengan perubahan kode yang sangat kecil. Selain itu, model yang dilatih pada satu mesin dapat digunakan pada perangkat ringan lain, seperti perangkat seluler yang mendukung Android, untuk tujuan implementasi akhir (Goyal, Pandey, & Jain, 2018).

TF.Learn adalah sebuah antarmuka yang lebih spesifik dari TensorFlow yang membantu orang-orang untuk memulai menganalisa prediktif dan *data mining*. TF.Learn membantu memperlancar proses transisi dunia Scikit-Learn (*Machine Learning Library* menggunakan Python) *one-liner machine learning* ke dunia yang lebih terbuka dalam membangun berbagai bentuk model *machine learning* (Tang, 2016).

Natural Language Toolkit (NLTK) adalah serangkaian modul program sumber terbuka, tutorial dan set masalah, yang menyediakan courseware linguistik komputasi yang siap pakai. NLTK mencakup pemrosesan bahasa alami simbolis dan statistik, dan dihubungkan dengan korpora beranotasi (Bird, 2007).

Algoritma Nazief Adriani merupakan salah satu algoritma yang digunakan untuk proses *stemming* dalam bahasa Indonesia. *Stemming* merupakan proses transformasi kata menjadi kata dasarnya (*root*) dengan menghilangkan semua imbuhan kata (*affixes*) meliputi awalan kata (*prefixes*), sisipan kata (*infixes*), akhiran kata (*suffixes*) dan atau menghilangkan awalan dan akhiran kata (*confixes*) pada kata turunan.

Neural Network (NN) atau *Artificial Neural Network* (ANN) adalah bidang pembelajaran mesin yang diperluas yang telah terbukti sangat bermanfaat, terutama dalam bidang teks, gambar, dan ucapan. Kumpulan algoritma yang diterapkan pada NN memiliki kesamaan dengan hubungan antara rangsangan dan *neuron* di otak manusia. NN memiliki aplikasi yang luas dalam bidang komputer, terjemahan bahasa, pengenalan suara, pembuatan gambar, dan sebagainya. NN merupakan cara menghitung *output* dari *input* (klasifikasi) menggunakan koneksi berbobot (sinapsis) yang dihitung dari pengulangan berulang melalui data pelatihan. Setiap melewati data pelatihan hal ini akan terus menerus mengubah bobot sehingga NN menghasilkan *output* dengan akurasi yang lebih besar (tingkat kesalahan yang lebih rendah). NN terinspirasi dari kesadaran atas *complex learning system* pada otak yang terdiri dari beberapa set neuron yang saling berhubungan secara dekat. Jaringan neuron mampu melakukan tugas yang sangat kompleks seperti klasifikasi dan pemahaman pola. NN dapat memperkirakan rentang yang cukup luas suatu model statistika dan fleksibel dalam menggambarkan model (linier maupun nonlinier) (Bar-Yam, 1997).

2. TINJAUAN PUSTAKA

Penelitian yang dilakukan oleh (Wiranda, 2018) dengan judul **Implementasi *Natural Language Processing* Pada *Chatbot* Peribahasa** dimana penelitian ini menggunakan NLP dengan tujuan melakukan proses pembuatan model komputasi dari bahasa sehingga dapat terjadi suatu interaksi antara manusia dengan komputer dengan perantara bahasa alami. Inputan pengguna akan diproses menggunakan metode text mining. Respon yang dihasilkan merupakan pencocokan *pattern* pada *knowledge base* dengan menggunakan Algoritma *Boyer-moore* dan menghasilkan respon balasan yang dianggap paling cocok. Algoritma *Boyer-moore* hanya dapat menangani *keyword* yang tepat sesuai *knowledge base*. Selain itu digunakan Algoritma *Levenshtein Distance* sebagai *spelling correction* yang digunakan jika terjadi kesalahan pengetikan pada masukan pengguna dengan cara membandingkan dua buah *string* dan mengambil nilai perbedaan yang terkecil. Pada penelitian ini dapat disimpulkan bahwa sistem ini dapat digunakan sebagai media interaktif untuk mendapatkan informasi peribahasa Indonesia dengan persentase keberhasilan *chatbot* sebesar 83,3% dimana hasil yang diperoleh dari respon *chatbot* tergantung pada kelengkapan kata kunci.

Penelitian yang dilakukan oleh (Singh, Paste, Shinde, Patel, & Mishra, 2018) dengan judul ***Chatbot Using TensorFlow for Small Businesses***. Penelitian yang diusulkan adalah untuk membuat sebuah *chatbot* yang menunjukkan metode pengembangan *chatbot* yang dapat mengikuti konteks percakapan. Metode ini menggunakan TensorFlow untuk mengembangkan model *neural network* dari *chatbot* dan menggunakan teknik NLP untuk mempertahankan konteks percakapan. *Chatbot* ini nantinya dapat digunakan dalam industri kecil atau bisnis untuk mengotomatiskan layanan pelanggan karena permintaan pengguna akan ditangani oleh *chatbot* sehingga mengurangi kebutuhan tenaga kerja dan pengeluaran. Hasil dari penelitian ini adalah dari 10 pertanyaan yang diajukan, aplikasi menjawab 6 hingga 7 jawaban benar. Namun hal tersebut juga tergantung pada bagaimana pertanyaan disusun. Diasumsikan pengguna memiliki pengetahuan bahasa Inggris yang baik dan dapat berinteraksi dengan aplikasi dengan baik.

Penelitian yang dilakukan oleh (Massaro, Maritati, & Galiano, 2018) dengan judul ***Automated Self-Learning Chatbot Initially Built as a FAQs Database Information Retrieval System: Multi-level and Intelligent Universal Virtual Front-Office Implementing Neural Network Case Study*** dimana penelitian ini melakukan perbandingan antara AIML dengan metode *Neural Network* (NN) untuk mendapatkan seberapa besar akurasi yang diberikan oleh masing-masing metode untuk merespon *request* dalam percakapan dengan *user*. Penelitian ini bertujuan untuk mengimplementasikan *smart virtual front-office* atau sering disebut *customer service* cerdas, yang akan mensinergikan antara daftar FAQ dan *chatbot* dalam menangani *user*. Hasil dari penelitian ini adalah metode AIML mendapatkan 85% ketepatan, 5% jawaban salah, dan 10% tak terjawab. Sedangkan NN mendapatkan 99% ketepatan, 0% jawaban salah, dan 1% tak terjawab.

Penelitian sebelumnya mengenai *chatbot* berbasis NLP banyak di antaranya masih menggunakan metode berbasis *comparing*, membandingkan dan mencari kemiripan, antara *input* atau permintaan *user* dengan informasi atau pengetahuan yang dimiliki *chatbot*, seperti

penggunaan AIML, Algoritma Bigram, Algoritma Boyer Moore, dan lain-lain. Kekurangan dari metode tersebut adalah tidak optimalnya akurasi antara apa yang dimaksud *user* pada *input* dengan respon yang diberikan oleh *chatbot*. Adapun penelitian sebelumnya yang menggunakan metode *Deep Neural Network* telah memberikan hasil dengan akurasi yang cukup baik. Selain itu *chatbot* juga bisa semakin belajar (*self learning*) dari banyaknya variasi masukan yang didapatkan dari *user*.

Berdasarkan uraian di atas dapat disimpulkan bahwa penelitian ini akan membangun sebuah *chatbot* berbasis NLP menggunakan metode *Deep Neural Network* atau *Deep Neural Network* sehingga *chatbot* dapat memberikan respon berdasarkan konteks dan dapat mempertahankan konteks. *Chatbot* dibangun dengan *messenger* Telegram supaya memudahkan tidak lagi berfokus membangun *interface* aplikasi *chat*. Telegram merupakan *platform instant messenger* yang memiliki tingkat keamanan yang tinggi serta kecepatan aksesnya. Telegram juga memiliki Bot API yang terus berkembang sehingga memudahkan pertukaran data *cross platform*.

3. METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan pada penelitian ini adalah kuantitatif yang terdiri dari 3 tahap. Tahap pertama adalah studi literatur penelitian terkait. Mengumpulkan dan mempelajari referensi yang terkait dengan penelitian sehingga dapat memperkaya informasi dan juga ilmu untuk mendasari dan menunjang penelitian yang dilakukan. Tahap kedua adalah mengumpulkan data. Data yang dikumpulkan adalah daftar FAQ yang didapat dari halaman *help desk* Kitabisa.com. Pengumpulan data tersebut sudah mendapatkan izin kerjasama dari pihak Kitabisa.com. Tahap ketiga adalah analisa data dari daftar FAQ yang sudah terkumpul berdasarkan studi literatur yang telah dilakukan. Proses analisa data akan dibahas pada tahap pengembangan.

3.1. Pengumpulan Kebutuhan

Pengumpulan kebutuhan merupakan segmen atau tahapan untuk menganalisis masalah, fungsi dan komponen produk yang akan dibuat dan hasil dari segmen ini adalah sebuah dokumen kebutuhan yang meliputi analisis kebutuhan pengguna, analisis domain, dan identifikasi sistem.

Kebutuhan Pengguna

Analisis kebutuhan *user* pada Otomatisasi Layanan Media Informasi FAQ Berbasis NLP pada Telegram Bot ini berupa:

1. Melakukan *input* pertanyaan dengan bahasa alami (*natural language*) pada sistem.
2. Mendapatkan respon jawaban sesuai yang diinginkan.

Domain

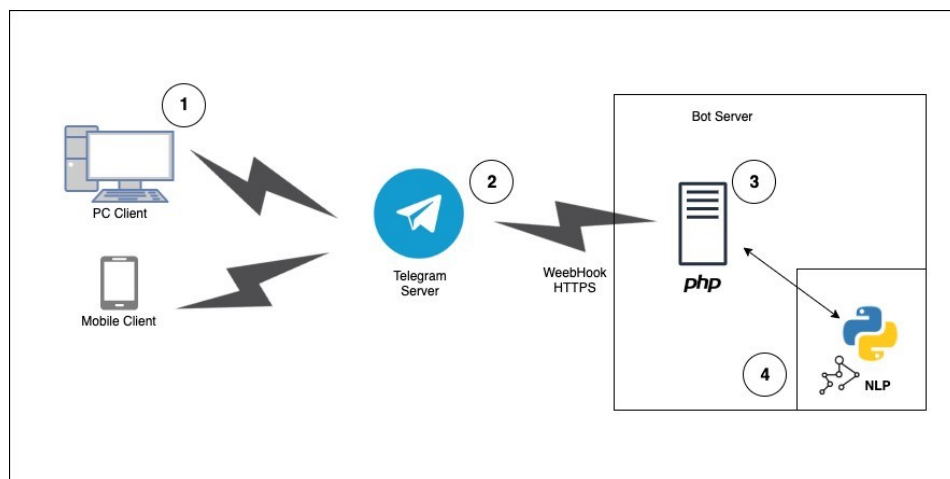
Analisis domain pada penelitian ini adalah aplikasi yang akan dibangun berbasis *mobile* dan *desktop* yang mana akan menggunakan fitur *bot* pada *platform* Telegram. Ada 2 sisi dalam pembuatan aplikasi ini. Sisi Telegram sebagai *server* dan *interface* dengan *user* dibangun menggunakan *framework* Laravel bahasa pemrograman PHP dan sisi Telegram *bot* sebagai AI *engine* dibangun menggunakan bahasa pemrograman *Python*. Aplikasi dapat diakses melalui berbagai sistem operasi; untuk *mobile* dapat diakses melalui sistem operasi Android, iOS, Windows Phone, dan Ubuntu Tocrh. Sedangkan *desktop* dapat diakses melalui sistem operasi Windows, OS X, dan Linux. Aplikasi ini berfungsi sebagai *chatbot* yang dapat merespon jawaban dari pertanyaan *user* yang masuk di manapun dan kapanpun.

3.2. Analisis

Dalam tahap ini membahas tentang analisis perancangan sistem yang akan dibangun menggunakan UML sebagai bahasa spesifikasi standar untuk mendokumentasi, menspesifikasi, dan membangun sebuah sistem dengan menggunakan beberapa diagram.

Identifikasi Sistem

Identifikasi sistem merupakan gambaran model diagram untuk menunjukkan tata letak sebuah sistem secara fisik dengan menampilkan bagian-bagian perangkat lunak yang berjalan pada perangkat keras (*hardware*). Perangkat keras yang digunakan adalah ponsel dan *Personal Computer* (PC) yang akan menjalankan salah satu fitur dari Telegram *messenger*, yaitu *bot*. *Bot* ini nanti yang akan merespon atau menjawab pertanyaan yang masuk dari *user* atau *client*. Arsitektur sistem dapat dilihat di Gambar 1. *User* mengirimkan pesan teks ke *account bot* melalui Telegram *server* dan diteruskan ke *bot server*. *Bot server* akan memproses pesan tersebut untuk dapat memberikan respon yang tepat ke *user* berupa pesan teks. Respon jawaban dikirimkan ke *user* melalui Telegram *server*.



Gambar 1. Arsitektur Aplikasi

Keterangan:

1. *User/Client* mengirimkan pesan kalimat (pertanyaan) dari *desktop* atau *mobile*.
2. Pesan diterima oleh server Telegram yang kemudian diteruskan melalui Telegram API menuju *Bot Server*.
3. *Bot Server* yang berbasis PHP menerima terusan dari Telegram API untuk kemudian dilanjutkan ke sisi *Python* untuk dilakukan NLP pada data masukan.
4. Sisi *Python* melakukan NLP pada data sehingga mendapatkan hasil untuk diteruskan kembali ke *user/client* sebagai respon (jawaban pertanyaan).

Setiap pesan akan ditangkap oleh sisi PHP yang kemudian pesan tersebut akan diteruskan ke sisi *Python* untuk dilakukan pemrosesan NLP, yaitu *text preprocessing* berupa *tokenizing* dan *stemming*. Hasil proses tersebut akan dilakukan proses konversi menjadi *bag of words* yang kemudian dilakukan proses klasifikasi teks dengan ANN.

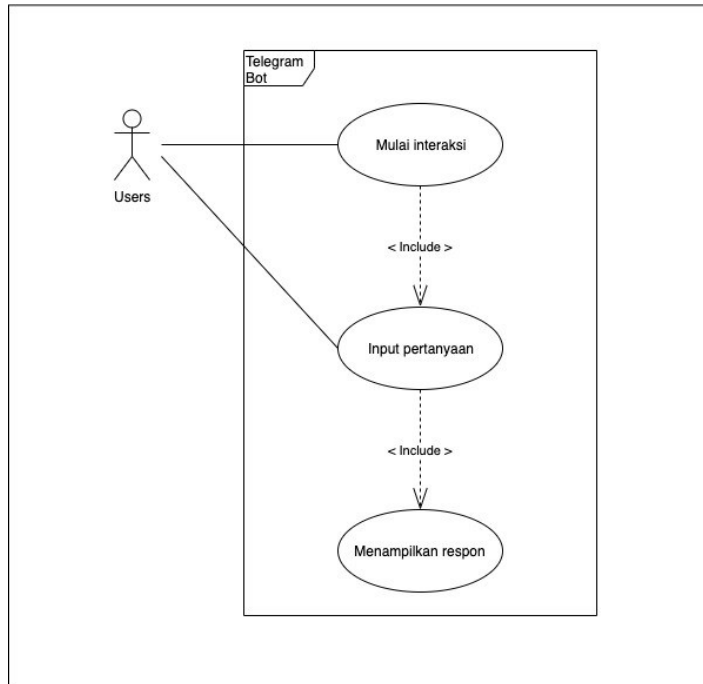
Identifikasi Kebutuhan Sistem

Aksi identifikasi kebutuhan sistem ini menghasilkan produk *use case diagram*, dan *sequence diagram*.

a. Use Case Diagram

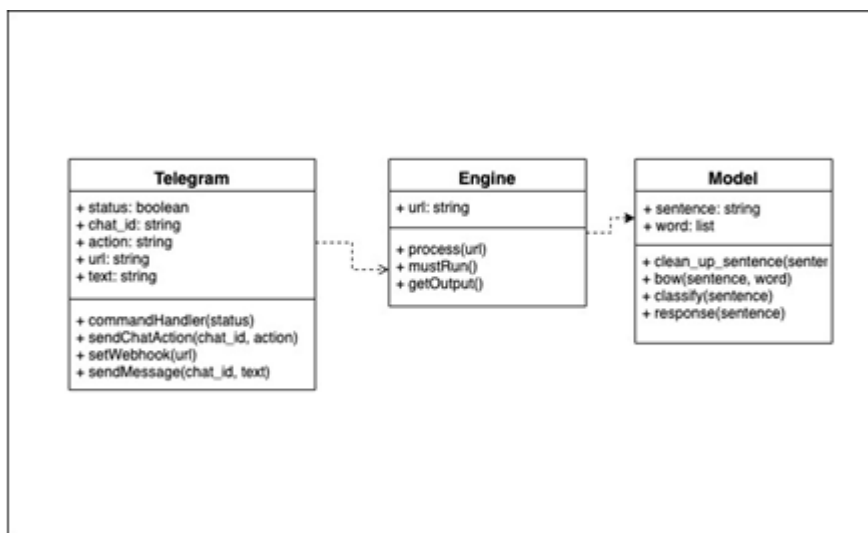
Use case diagram pada Gambar 2 memiliki 1 aktor yaitu *user*. Dari diagram tersebut dapat dijelaskan bahwa *user* dapat melakukan 2 tindakan awal terhadap Telegram Bot. Apabila *user* baru pertama kali melakukan interaksi dengan *bot* maka *use case* "Mengirimkan perintah /start" adalah tindakan pertama yang harus dilakukan oleh *user* untuk memulai interaksi. Namun apabila sudah pernah melakukan interaksi sebelumnya

maka *user* dapat langsung melakukan *use case* "Input pertanyaan". Ketiga *use case* yang ada pada Telegram Bot merupakan satu kesatuan fungsionalitas *use case*.



Gambar 2. Use Case Diagram

b. Class Diagram



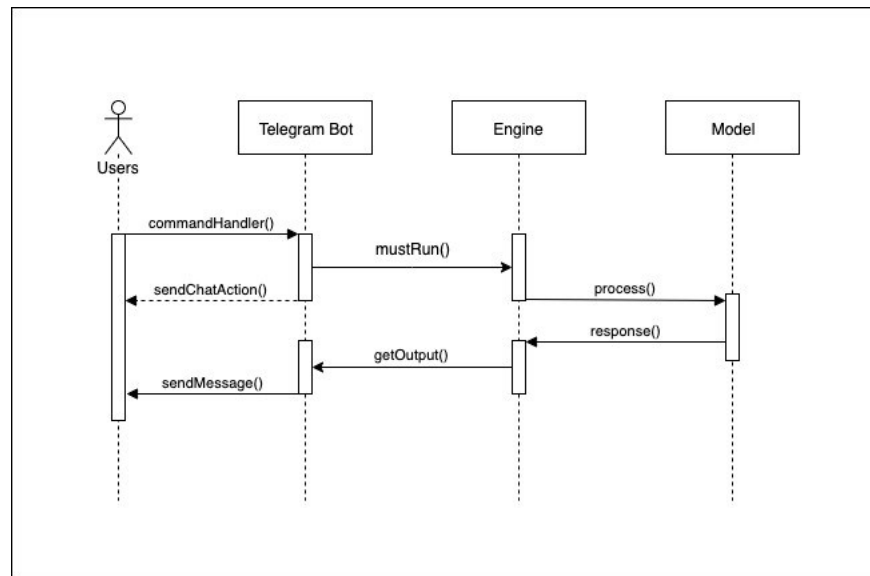
Gambar 3. Class Diagram

Pada Gambar 3 di atas terlihat hubungan antar *class* pada aplikasi Telegram Bot untuk menangani masukan pertanyaan *user* sehingga dapat memberikan respon jawaban. Setiap *class* memiliki *attributes* dan *operations* masing masing. Contohnya *method* `commandHandler()` membutuhkan parameter atribut `status` dengan tipe `boolean`. *Method*

ini selalu bernilai *true* sebab berfungsi untuk menangkap setiap perintah atau masukan dari *user* yang kemudian data tersebut digunakan untuk operasi *method* lainnya.

c. Sequence Diagram

Gambar 4 menunjukkan alur aksi dari aplikasi bahwa aksi pertama yang dilakukan adalah menangkap setiap perintah yang *user* lakukan terhadap *bot*. Telegram *bot* melakukan respon balik sementara kepada *user* selagi menjalankan proses lanjutan dari perintah *user* menuju *engine*. Dilakukan proses pengartian perintah sesuai dengan pada sisi model kemudian hasilnya dikembalikan kepada Telegram *bot* untuk dikirimkan sebuah pesan kepada *user* sebagai bentuk respon jawaban atas perintah yang masuk.



Gambar 4. Sequence Diagram

Analisis Sumber Data

```

1  {
2  "intents": [
3  {
4    "tag": "sapaan",
5    "patterns": [
6      "Selamat *",
7      "Hello min",
8      "Hallo min",
9      "Hi. Mau tanya dong",
10     "Permisi min, mau nanya",
11     "Hai",
12     "Hi min",
13     "Pagi",
14     "Siang",
15     "Sore",
16     "Malam"
17   ],
18   "responses": [
19     "Hallo #orangbaik. Ada yang bisa saya bantu?",
20     "Hai #orangbaik. Ada yang ingin Anda tanyakan?",
21     "Selamat datang. Ingin menanyakan apa?"
22   ],
23   "context_set": ""
24 },
25 {
26   "tag": "salam",
27   "patterns": [
28     "Assalamu'alaikum",
29     "Assalamualaikum",
30     "Assalamu'alaikum. Mau nanya min"
31   ],
32   "responses": [
33     "Wa'alaikumussalam. Hallo #orangbaik. Ingin menanyakan apa?",
34     "Wa'alaikumussalam, ada yang ingin Anda tanyakan?"
35   ],
36   "context_set": ""
37 },
38 ],
39 }
  
```

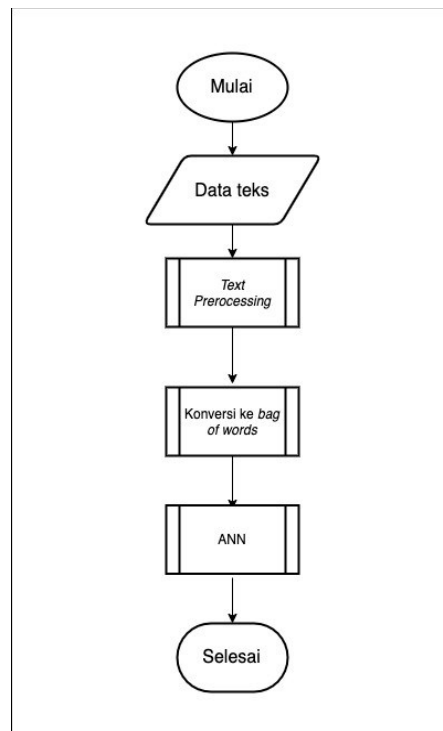
Gambar 5. Chatbot Intents

Data yang digunakan pada penelitian ini diambil dari daftar FAQ yang terdapat di halaman FAQ *website* Kitabisa.com (kitabisa.com/faq). Daftar FAQ kemudian disusun seperti pada Gambar 5 dengan maksud untuk membuat kerangka *chatbot* yang terstruktur di mana *intent* (maksud) percakapan didefinisikan. Setiap daftar yang berisi pertanyaan dan jawaban diklasifikasikan secara manual berdasarkan atau ditandai dengan *tag* sebagai identitas unik.

Setiap pertanyaan dikembangkan menjadi beberapa variasi pertanyaan yang kemudian ditandai dengan *patterns*, pola kalimat yang nantinya untuk pengklasifikasian teks dengan NN. Setiap jawaban juga bisa dikembangkan menjadi beberapa variasi jawaban atau tetap 1 yang kemudian ditandai dengan *responses*. *File* berisi daftar FAQ yang sudah disusun disimpan dalam 1 *file* dengan tipe JSON.

3.3. Klasifikasi Teks

Klasifikasi teks untuk memproses masukan berupa teks kalimat terdapat 3 tahapan proses. Diagram alur proses klasifikasi teks ada di Gambar 6.



Gambar 6. Flowchart Klasifikasi Teks

Text preprocessing merupakan salah satu langkah yang penting untuk *text classification* daftar FAQ untuk pengetahuan dasar *chatbot*. Tahap *text processing* yang dilakukan ada 2, yaitu *tokenizing* dan *stemming*.

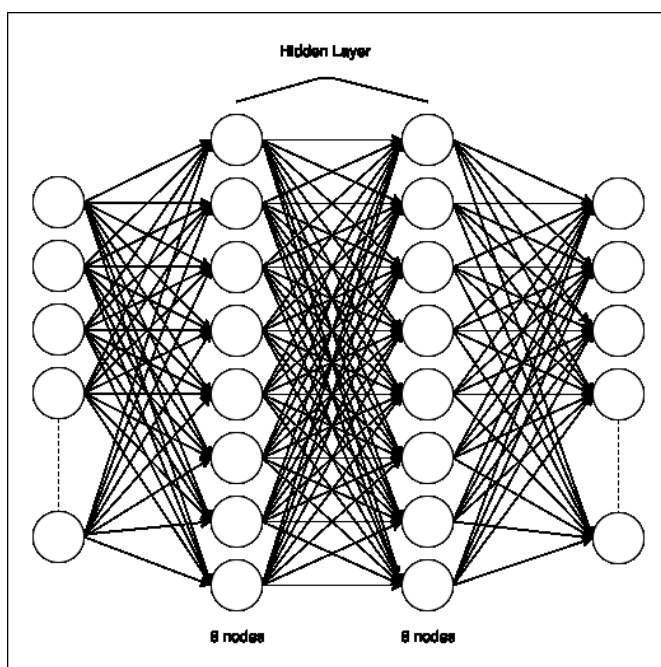
Tahap *tokenizing* adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya (Ratniasih, et al., 2017) dengan menggunakan karakter spasi dan tanda baca sebagai pemisah. Perintah *tokenizing* yang digunakan untuk penelitian ini adalah fungsi `word_tokenize()` pada *library* yang disediakan oleh NLTK. Fungsi tersebut melakukan langkah-langkah berikut:

1. Memperlakukan sebagian besar karakter tanda baca sebagai *token* terpisah.
2. Memisahkan koma dan kutipan tunggal, bila diikuti oleh karakter spasi.
3. Memisahkan tanda titik yang muncul di akhir baris.

Pada penelitian ini proses *steeming* menggunakan *Library* Python Sastrawi. *Library* ini merupakan pengembangan dari *Library* PHP Sastrawi dimana *library* tersebut menerapkan algoritma Algoritma Nazief dan Adriani.

Dibutuhkan sebuah cara untuk merepresentasikan data teks agar dapat diolah oleh algoritma *machine learning* dan model *bag of words* lah yang dapat menjalankan tugas tersebut. Ini adalah cara mengekstraksi fitur dari teks untuk digunakan ke dalam algoritma *machine learning*. Hal pertama yang dilakukan pada tahap konversi ke *bag of words* adalah melakukan *text preprocessing* berupa *tokenizing* dan *stemming* untuk memecah kalimat menjadi kata dan

Arsitektur NN yang akan digunakan pada tahap ini adalah *multilayer* NN, yang meliputi *input layer*, 2 *hidden layers*, dan *output layer*. Pendekatan “*bag of words*” digunakan juga untuk merapikan data *training* dengan mengubah teks menjadi *array* 0 dan 1. Masukan yang digunakan dalam *input layer* adalah kata-kata unik dalam korpus hasil dari proses *stemming* dan konversi *bag of words*. Jumlah *node* yang digunakan pada masing-masing *hidden layer* adalah 8. Jumlah tersebut adalah jumlah yang sudah umum digunakan berdasarkan (Hirst et al., 2015). Gambar arsitektur NN ada di Gambar 7.



Gambar 7. Arsitektur NN

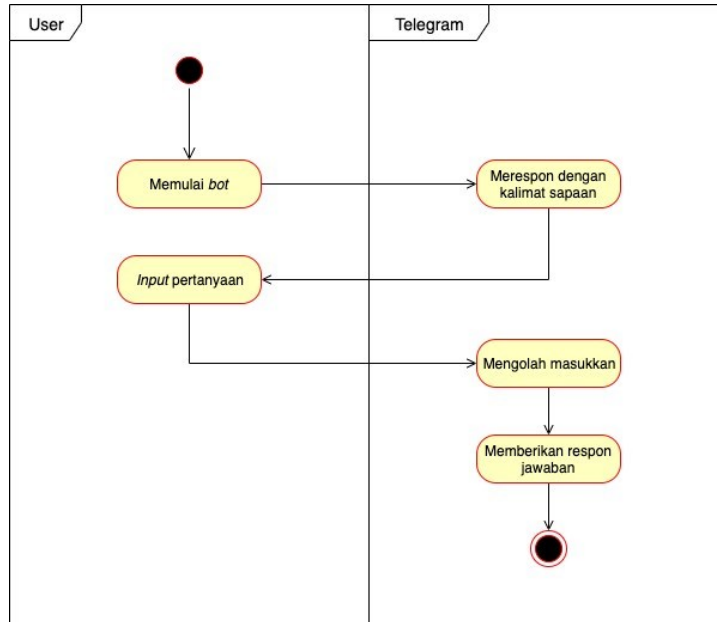
Proses pada NN menggunakan *library* dari TF.Learn. Mulai dari proses *feeding*, melewati *hidden layer*, regresi/evaluasi, hingga mendapatkan *output*. Jumlah *node* pada *output layer* berdasarkan jumlah kelas/*intent* dari data latih. Untuk memudahkan mendapatkan *output* yang sesuai pada hasil proses perhitungan *hidden layer* dioptimalkan menggunakan teknik Adam (*Adaptive Moment Estimation*). Teknik Adam merupakan setelan *default* dari *library* TF.Learn untuk proses optimasi. Hasil dari optimasi kemudian dilanjutkan dengan proses fungsi aktivasi *softmax* untuk normalisasi nilai agar mendapatkan *output* yang tepat.

3.4. Desain

Pada tahap perancangan sistem ini akan memodelkan urutan atau alur kerja suatu proses yang dirancang menggunakan diagram *activity*. Dari pemodelan tersebut kemudian dikembangkan sehingga menghasilkan antarmuka aplikasi.

Activity Diagram

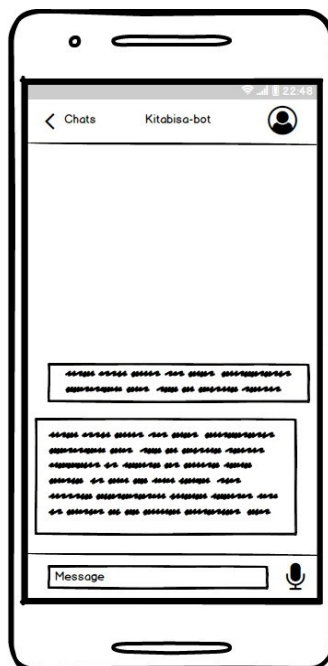
Pada Gambar 8 terlihat saat *user* membuka halaman *chat* dengan *bot* untuk pertama kalinya maka akan muncul tombol “/start” yang digunakan untuk memulai percakapan awal dengan *bot*. *Bot* pertama kali akan merespon dengan sapaan. Selanjutnya *user* dapat memasukkan kalimat teks pertanyaan kepada *bot*. Ada momen sejenak *bot* memproses masukan yang nanti kemudian akan membalas dengan *respon* jawaban kepada *user*.



Gambar 8. Activity Diagram

Perancangan Aplikasi

Perancangan aplikasi antarmuka untuk aplikasi media informasi FAQ pada Telegram Bot, yang diberi nama Kitabisa-bot berupa antarmuka untuk *mobile* dapat dilihat pada Gambar 9.



Gambar 9. Perancangan Antarmuka Mobile

Perancangan Pengujian

Pengujian yang dilakukan hanyalah pengujian *chatbot* yang mana dilakukan untuk menilai apakah *chatbot* memiliki kemampuan untuk menjawab pertanyaan yang diajukan oleh pelanggan. Teknis pengujiannya yaitu dengan cara Dibagikan daftar 10 tema pertanyaan kepada 10 koresponden untuk dilakukan pengujian oleh masing-masing koresponden dengan memasukkan kalimat tanya dengan bahasa alamiah untuk masing-masing tema. Kalimat tanya bersifat variatif berdasarkan masing-masing koresponden. Dari total jumlah pertanyaan yang diajukan kemudian dihitung berapa persen pertanyaan yang sesuai dengan jawaban dan mana yang tidak.

4. HASIL DAN PEMBAHASAN

Setelah melakukan penelitian dan perancangan, pada tahap ini merupakan implementasi dan pengujian hasil implementasi. Tahap pertama adalah dilakukan proses pembentukan model dari data FAQ untuk dijadikan *knowledge base* sebagai acuan *chatbot* dalam memahami *request user*. Pembentukan model dilakukan mulai tahap *text preprocessing*, konversi ke *bag of words*, hingga iterasi oleh proses NN. Tahap kedua adalah dilakukan proses pencocokan model, yaitu masukan kalimat pertanyaan dari *user* dengan *knowledge base* yang dimiliki oleh *chatbot* guna mendapatkan hasil respon jawaban yang sesuai dengan konteks. Proses pencocokan model tersebut dimulai dari pengolahan masukan dengan *text preprocessing* dan konversi ke *bag of words* yang kemudian hasil tersebut dilakukan proses NN dengan mencocokkan hasil perhitungan masukan dengan bobot yang telah dimiliki model untuk mendapatkan keluaran yang menunjukkan masukan *user* memiliki *intent* yang nantinya menjadi acuan untuk mengembalikan respon yang sesuai dengan *intent* tersebut.

Hasil dari pengujian yang dilakukan adalah 100 pertanyaan yang berasal dari 10 koresponden yang mengajukan masing-masing 10 pertanyaan mendapatkan 73 pertanyaan yang direspon sesuai dengan konteks dan 27 pertanyaan direspon tidak sesuai dengan konteks. Apabila dipersentasekan mendapatkan akurasi sebesar 73%.

Kendala yang didapat selama pengujian adalah sebagian koresponden salah mengartikan contoh pertanyaan yang diberikan sehingga variasi pertanyaan yang diajukan oleh koresponden tidak sesuai dengan konteks yang seharusnya. Hal itu menyebabkan *chatbot* merespon dengan *konteks* yang salah. Salah pengetikan huruf juga dilakukan oleh sebagian koresponden sehingga ada beberapa pertanyaan yang diajukan tidak mendapatkan respon yang sesuai. Hasil tampilan antarmuka aplikasi Telegram pada *mobile* dapat dilihat pada Gambar 10.



Gambar 10. Antarmuka Mobile

5. KESIMPULAN

Berdasarkan hasil dari analisis dan perancangan yang telah dibuat, didapatkan beberapa kesimpulan, yaitu:

- a. Aplikasi pemesanan otomatis untuk media informasi FAQ di Kitabisa.com menggunakan Telegram bot berhasil dibangun.
- b. Telegram bot berbasis NLP menggunakan teknologi TensorFlow berhasil dibangun. Sisi antarmuka Telegram bot dibangun menggunakan Laravel, PHP framework. Sedangkan untuk proses NLP berdasar bahasa pemrograman Python. Kedua bahasa pemrograman tersebut berfungsi dengan baik di dalam satu kesatuan aplikasi.
- c. Hasil pengujian data yang didapatkan dari tes aplikasi chatbot yang telah dibuat dengan basis NLP pada penelitian ini memiliki persentase nilai akurasi yang dihasilkan oleh aplikasi adalah sebesar 73%. Kesalahan respon yang tidak sesuai konteks dapat disebabkan oleh kesalahan koresponden memahami konteks contoh pertanyaan dan masukan koresponden yang tidak menggunakan kalimat formal dan salah pengejaan huruf.

DAFTAR PUSTAKA

- Bar-Yam, Y. (1997). *Dynamics of complex systems* (Vol. 213). Addison-Wesley Reading, MA.
- Bird, S. (2007). Nltk, 69–72.
- Goyal, P., Pandey, S., & Jain, K. (2018). Deep Learning for Natural Language Processing Creating Neural Networks with Python Deep Learning for Natural Language Processing: Creating Neural Networks with Python. <https://doi.org/10.1007/978-1-4842-3685-7>
- Hirst, G., Goldberg, Y., Williams, P., Sennrich, R., Post, M., Koehn, P., ... Van Zaanen, M. (2015). *Synthesis Lectures on Human Language Technologies* Editor Neural Network Methods for Natural Language Processing Syntax-based Statistical Machine Translation Domain-Sensitive Temporal Tagging Linked Lexical Knowledge Bases: Foundations and Applications Bay.
- Massaro, A., Maritati, V., & Galiano, A. (2018). Automated Self-Learning Chatbot Initially Built as a FAQs Database Information Retrieval System : Multi-level and Intelligent Universal Virtual Front-Office Implementing Neural Network Case study, 42, 515–525.
- Meinanda, M. H., Annisa, M., Muhandri, N., & Suryadi, dan K. (2009). Prediksi masa studi sarjana dengan artificial neural network. *Internetworking Indonesia Journal*, 1(2), 31–35.
- Ratniasih, N. L., Sudarma, M., & Gunantara, N. (2017). Penerapan Text Mining dalam Spam Filtering untuk Aplikasi Chat. *Majalah Ilmiah Teknologi Elektro*, 16(3), 13–18.
- Ridwan, F. (2013). *Pembangunan Aplikasi Natural Language Procesing Untuk Pembuatan Chat Bot Berbasis Web* (Di Universitas Komputer Indonesia).
-

Universitas Komputer Indonesia. Retrieved from <https://repository.unikom.ac.id/23032/>

Singh, R., Paste, M., Shinde, N., Patel, H., & Mishra, N. (2018). Chatbot using TensorFlow for small Businesses. 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), (Icicct), 1614–1619.

Tang, Y. (2016). TF.Learn: TensorFlow's High-level Module for Distributed Machine Learning, 1–7. <https://doi.org/10.1111/j.1528-1167.2010.02793.x>

Tim Pengembang Pedoman Bahasa Indonesia. (2016). Pedoman Umum Ejaan Bahasa Indonesia Edisi Keempat.

Wiranda, D. (2018). Implementasi Natural Language Processing pada Chatbot Peribahasa Indonesia.