

Penerapan Deep Learning dalam Pembuatan Chatbot untuk Service Desk ITS

Naufal Sahat Pratama, Supeno Mardi S. Nugroho

Dept. Teknik Komputer Institut Teknologi Sepuluh Nopember Surabaya, Indonesia

Email: naufal.sahat16@mhs.te.its.ac.id, mardi@its.ac.id

Abstrak

Artificial intelligence sudah menjadi sesuatu yang tidak bisa dipisahkan dari kehidupan manusia sekarang ini. Salah satu bentuk dari *artificial intelligence* yang sering ditemui dalam kehidupan sehari-hari adalah chatbot. Belakangan ini chatbot banyak sudah digunakan oleh perusahaan untuk menunjang bisnis mereka. Alasan penggunaan chatbot yang semakin marak di dunia bisnis adalah chatbot dapat digunakan sebagai layanan pelanggan atau *customer service* yang biaya operasionalnya lebih murah dan layanannya yang tersedia 24 jam. Meskipun sudah banyak digunakan di beberapa bidang, penggunaan chatbot di bidang akademik dirasakan masih kurang. Dalam ruang lingkup Institut Teknologi Sepuluh Nopember (ITS) sendiri, layanan Service Desk ITS masih menggunakan metode manual dalam menjawab pertanyaan yang disampaikan, yaitu dijawab langsung oleh admin. Oleh karena itu dibutuhkan chatbot yang dapat digunakan untuk mengatasi masalah tersebut.

Kata kunci: *Chatbot, Deep Learning, Natural language processing*

Diterima Redaksi: 1-Nov-2024 Selesai Revisi: 15-Nov-2024 Diterbitkan Online: 15-Des-2024
DOI: <https://doi.org/10.59378/jcenim.v2i3.57>

I. Latar Belakang

Kecerdasan buatan atau dalam bahasa Inggris biasa disebut *Artificial Intelligence* adalah kecerdasan yang diciptakan oleh manusia yang kemudian ditanamkan ke dalam suatu mesin sehingga membuat mesin tersebut seolah-olah mampu berpikir sendiri untuk mengambil sebuah keputusan. Salah satu bentuk dari *artificial intelligence* yang kini sering ditemui di kehidupan sehari-hari adalah chatbot. Chatbot merupakan sebuah program komputer yang dapat berkomunikasi dengan manusia menggunakan bahasa yang *natural*. Pada awalnya chatbot hanya digunakan sebagai medium automasi pesan saja. Namun, seiring perkembangannya chatbot banyak digunakan oleh berbagai perusahaan untuk menunjang usaha mereka. Salah satu pemanfaatannya adalah sebagai layanan informasi dan layanan pelanggan atau *customer service* [1].

Di Indonesia sendiri, chatbot sudah banyak diimplementasikan dalam berbagai bidang usaha. Dari bidang perbankan Bank Mandiri merilis Mandiri Intelligence Assistant atau Mita dan Bank BRI merilis Smart BRI New Assistant atau Sabrina. Sementara dari bidang telekomunikasi Telkomsel memiliki Veronika dan XL Axiata memiliki Maya. Keuntungan dari penggunaan chatbot sehingga banyak digunakan oleh berbagai perusahaan adalah karena dapat menghemat biaya operasional. Dengan menggunakan chatbot maka perusahaan tidak perlu mempekerjakan banyak karyawan sebagai *customer service* karena satu chatbot dapat menangani banyak pelanggan dalam satu waktu sedangkan satu karyawan hanya bisa menangani satu pelanggan dalam satu waktu. Keuntungan lain dari penggunaan chatbot adalah layanannya dapat tersedia 24 jam [2].

II. Dasar Teori

A. Chatbot

Chatbot merupakan sebuah program komputer yang dapat berkomunikasi dengan manusia menggunakan bahasa yang *natural* [3]. Chatbot dibekali oleh *artificial intelligence* atau kecerdasan buatan sehingga dapat menjawab pertanyaan yang diajukan oleh manusia. Setelahnya semakin banyak orang yang meneliti tentang chatbot sehingga terjadi perkembangan yang pesat dalam penelitian di bidang ini.

Chatbot dapat bekerja dengan menggunakan beberapa teknologi pendukung. Di antara teknologi pendukung tersebut adalah *machine learning* dan *natural language processing* [2]. Pada awalnya chatbot hanya digunakan sebagai medium automasi pesan saja. Namun, seiring perkembangannya chatbot banyak digunakan oleh berbagai perusahaan untuk menunjang usaha mereka.

B. Natural Language Processing

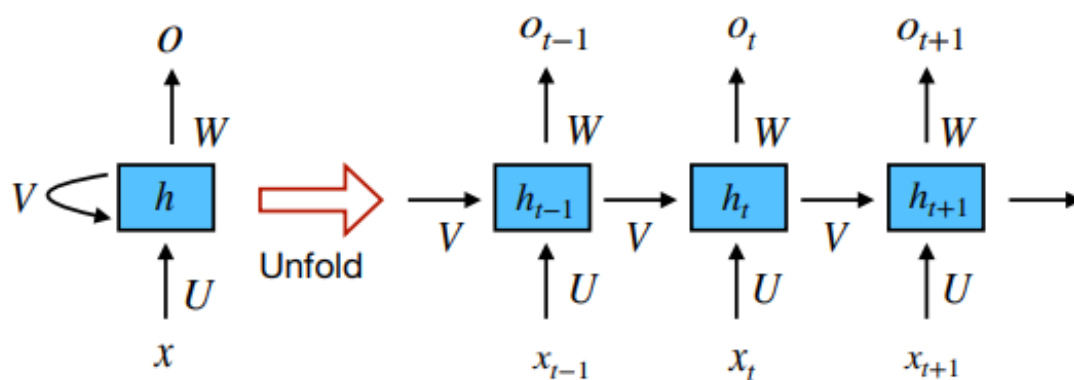
Natural language processing (NLP) atau pemrosesan bahasa alami adalah teori yang termotivasi dari teknik komputasi untuk analisa otomatis dan representasi bahasa manusia. NLP memungkinkan komputer untuk melakukan tugas bahasa alami, seperti penguraian dan pelabelan kelas kata, terjemahan mesin, hingga yang populer saat ini adalah sistem dialog. Sistem dialog memungkinkan manusia sebagai aktor untuk berinteraksi dengan mesin atau lebih dikenal sebagai chatbot [4]. Beberapa contoh lain dari penggunaan NLP adalah *stemming* (proses untuk menemukan kata dasar dari sebuah kata), *summarization* (ringkasan dari sebuah bacaan), dan *translation tools* (menerjemahkan bahasa).

C. Deep Learning

Deep learning adalah salah satu cabang dari *machine learning* yang menggunakan arsitektur *Multi Layer Perceptron* (MLP) dan algoritma *backpropagation*. Algoritma *backpropagation* digunakan untuk mengatasi keterbatasan MLP dalam mengolah data dalam jumlah besar [5]. Arsitektur dan algoritma *deep learning* tersebut telah membuat kemajuan yang mengesankan dalam bidang visi komputer dan pengenalan pola. Mengikuti tren ini, penelitian di bidang NLP belakangan ini semakin berfokus pada penerapan *deep learning* dalam penelitiannya. *Deep learning* memungkinkan fitur otomatis multi-level *representation learning*. Hal ini membuat *deep learning* lebih efektif untuk menyelesaikan permasalahan NLP dibandingkan dengan metode NLP tradisional [6].

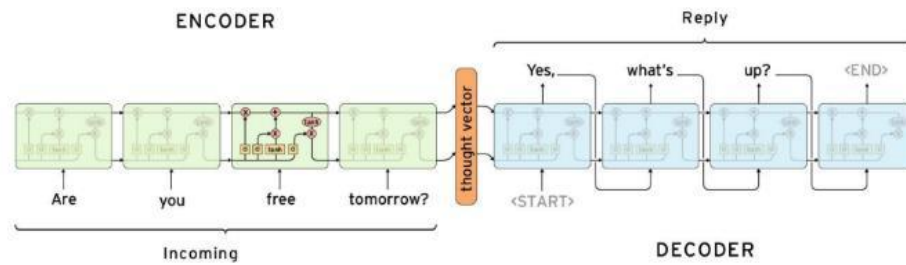
D. Recurrent Neural Network

Recurrent Neural Network (RNN) adalah salah satu arsitektur dari *artificial neural network* yang digunakan untuk memproses data sekuensial dan pemrosesannya dilakukan berulang-ulang. RNN banyak digunakan untuk menyelesaikan masalah yang berkaitan dengan *Natural language processing* (NLP) karena memiliki akurasi keberhasilan yang tinggi [5]. Beberapa bentuk aplikasi RNN pada NLP adalah *word-level classification*, *sentence-level classification*, dan *language modeling*.



Gambar 1: Ilustrasi RNN

Salah satu implementasi RNN yang sering digunakan dalam NLP adalah *Seq2Seq Model* (*Sequence to Sequence Model*). *Seq2Seq Model* memiliki dua RNN, yaitu satu untuk *encoder network* yang memproses input dan satu untuk *decoder network* untuk menghasilkan output. Penelitian arsitektur *seq2seq model* mengalami kemajuan untuk menghasilkan model dengan akurasi yang tinggi. Pada awalnya *seq2seq model* digunakan untuk menyusun ulang peringkat kalimat [7]. Namun, seiring perkembangannya *seq2seq model* digunakan untuk menghasilkan kalimat baru [10].

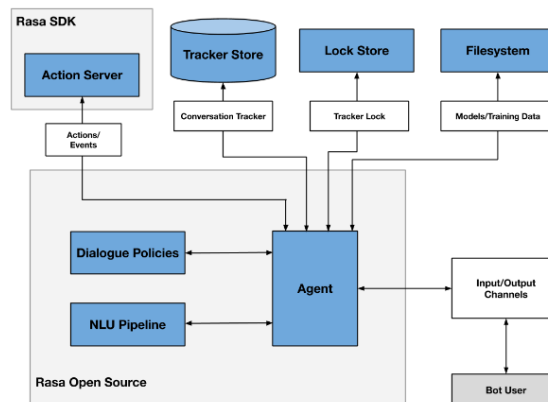


Gambar 2: Ilustrasi Seq2seq Model

E. Rasa Open Source

Rasa open source merupakan sebuah *framework* yang digunakan untuk automasi pesan dan bersifat *open source*. *Rasa open source* bertujuan untuk menjembatani kesenjangan antara penelitian dan aplikasi dan membawa kemajuan terbaru dalam sistem percakapan *artificial intelligence* dan *Natural language processing* kepada masyarakat pada umumnya yang ingin belajar bidang tersebut. *Rasa open source* sudah digunakan oleh ribuan developer di seluruh dunia [11].

Rasa open source memiliki dua komponen utama, yaitu *Natural language understanding* (NLU) dan manajemen dialog. NLU adalah bagian yang menangani klasifikasi *intent*, ekstraksi entitas, dan pengambilan respons. Tujuan dari NLU adalah untuk mengekstrak informasi terstruktur dari pesan pengguna. Ini biasanya mencakup maksud pengguna dan entitas apa pun yang terkandung dalam pesan mereka. Sedangkan komponen manajemen dialog memutuskan tindakan selanjutnya dalam percakapan berdasarkan konteksnya [12].



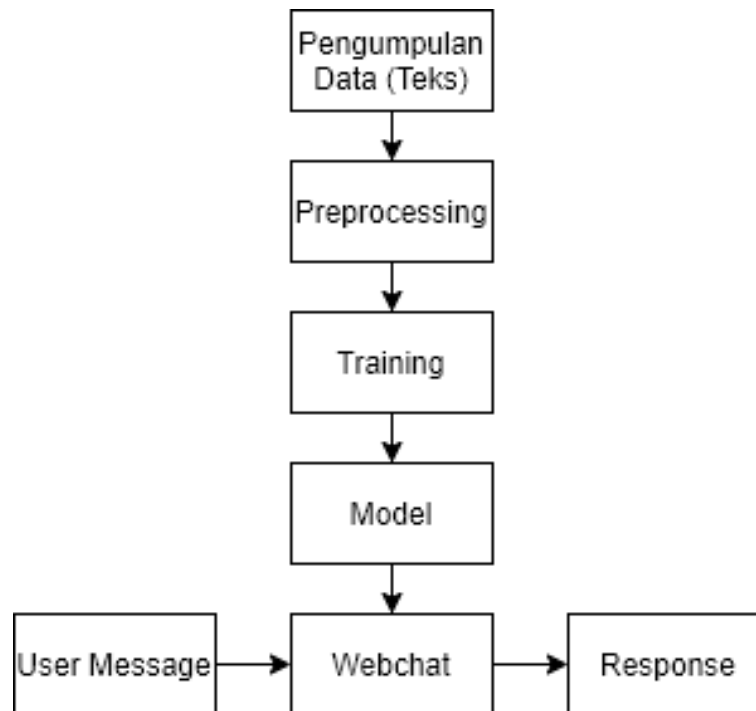
Gambar 3: Ilustrasi Arsitektur Rasa Open Source

Untuk mengaplikasikan *Rasa Open Source*, terdapat beberapa komponen yang perlu diperhatikan. Beberapa komponen tersebut adalah:

1. *Intent*: *Intent* merupakan sebuah representasi dari sebuah hal yang diinginkan oleh pengguna dalam sebuah percakapan atau maksud dari pesan yang dikirimkan oleh pengguna. Pada *Rasa Open Source intent* perlu didefinisikan dan diisi dengan variasi kalimat yang merepresentasikan *intent* itu yang nantinya hal ini akan digunakan sebagai training data.
2. *Rules*: *Rules* merupakan sebuah training data yang digunakan untuk mengelola sistem percakapan yang akan digunakan nantinya. *Rules* mendefinisikan percakapan singkat dan sederhana yang harus selalu mengikuti alur percakapan yang sama.
3. *Stories*: *Stories* merupakan sebuah training data yang digunakan untuk mengelola sistem percakapan yang akan digunakan nantinya. Berbeda dengan *rules*, *stories* dapat digunakan untuk mendefinisikan alur percakapan yang lebih kompleks dan beragam.
4. *Responses*: *Responses* merupakan balasan dari pesan yang dikirimkan oleh pengguna.

III. Desain Sistem

Penelitian ini bertujuan untuk menerapkan *deep learning* dalam pembuatan chatbot yang dapat menjawab pertanyaan yang disampaikan kepada Service Desk ITS. Adapun tahapan proses yang akan dikerjakan dapat dilihat pada Gambar 4.



Gambar 4: Blok Diagram Penelitian

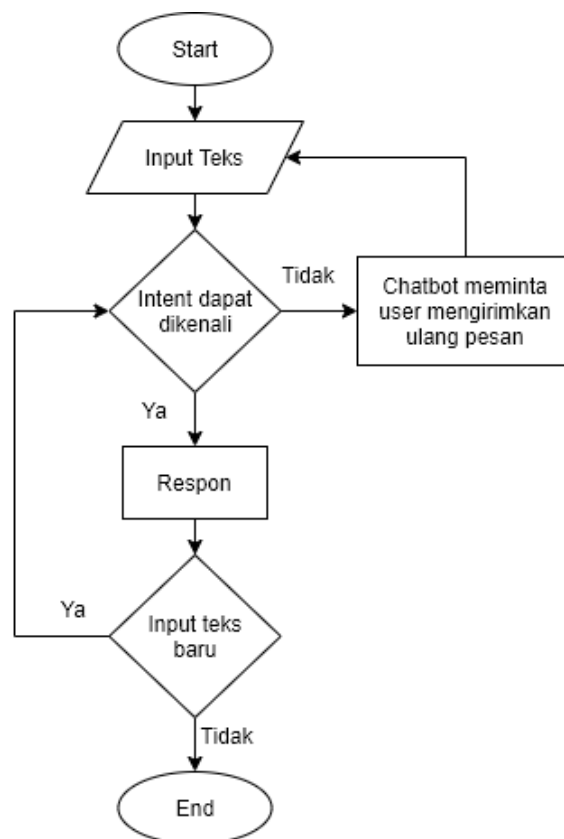
Berdasarkan Gambar 4, desain sistem pada pengerjaan penelitian ini dimulai dari input data berupa kumpulan teks percakapan. Dari data yang sudah dikumpulkan maka perlu dilakukan *preprocessing* terlebih dahulu agar data sesuai dengan format yang dapat diterima untuk tahap selanjutnya. Setelah *preprocessing* dilakukan maka tahap berikutnya adalah training data yang sudah disiapkan dan pembuatan model percakapan. Setelah training selesai maka chatbot akan diimplementasikan melalui webchat. Tahap terakhir adalah melakukan testing atau pengujian terhadap model yang diperoleh. Output yang diharapkan adalah jawaban yang sesuai dengan pertanyaan yang diajukan oleh pengguna kepada chatbot.

A. Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa FAQ (*Frequently Asked Questions*) dan beberapa panduan penggunaan layanan Service Desk ITS. Data yang digunakan dalam bentuk teks yang didapatkan dari Service Desk ITS. Data FAQ tersebut kemudian disimpan sebagai data NLU (*Natural Language Understanding*) pada *Rasa Open Source*. Data NLU diisi oleh kata kunci dan FAQ serta variasinya yang sekiranya akan dikatakan oleh pengguna. Data NLU ini dikategorikan berdasarkan *intent*. Setelah itu, untuk jawaban atau respon dari FAQ juga perlu dimasukkan. Karena merupakan FAQ, jawaban dari setiap pertanyaan yang diajukan, hanya memiliki satu jenis jawaban saja. Setelah data pertanyaan dan jawabannya dimasukkan, maka langkah selanjutnya yang perlu dilakukan pengaturan model sistem dialog yang akan digunakan dengan mendefinisikan *rules*. *Rules* menjelaskan bagian percakapan singkat yang harus selalu mengikuti jalur yang sama. Adapun alur percakapan yang digunakan dapat dilihat pada Gambar 5.

B. Proses Natural Language Processing

Proses berikutnya adalah *Natural Language Processing*. Pada tahap ini akan dilakukan proses NLP sebagai berikut. Yang pertama adalah mendefinisikan bahasa yang digunakan, yaitu bahasa Indonesia. Setelah itu akan dilakukan tokenisasi. Tokenisasi yang digunakan adalah *White space tokenization*.



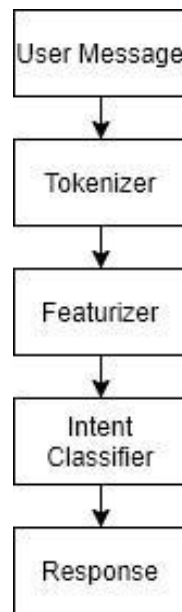
Gambar 5: Alur Percakapan yang Digunakan

White space tokenization akan memisahkan kalimat yang didapatkan dari pesan pengguna menjadi rangkaian kata atau token berdasarkan *white space* (ruang putih) atau spasi yang terdapat pada kalimat tersebut. Setelah itu akan dibuat *bag-of-words* yang berisi serangkaian token yang merepresentasikan pesan dari pengguna, *intent*, dan respon yang akan dikirimkan.

Proses selanjutnya adalah *featurizer*. Pada proses *featurizer* akan menggunakan *CountVectors Featurizer*. *CountVectors Featurizer* akan mengubah huruf-huruf dari rangkaian token yang didapatkan menjadi huruf kecil. Setelah itu, rangkaian token yang didapatkan akan direpresentasikan dalam bentuk vektor. Proses selanjutnya akan dilakukan klasifikasi *intent* untuk menyocokkan *intent* dari pesan yang dikirim oleh pengguna dengan data yang sudah dimasukkan sebelumnya. Klasifikasi *intent* yang digunakan pada proses ini adalah DIET (*Dual Intent and Entity Transformer*) *Classifier*. DIET *Classifier* merupakan sebuah arsitektur *multi-task* yang digunakan untuk melakukan klasifikasi *intent* dan pengenalan entity. Pada proses klasifikasi ini, pengenalan entity diprediksi melalui *conditional random field* pada layer atas output yang dihasilkan oleh transformer dengan input serangkaian token. Sedangkan untuk *intent label*, output dari transformer yang berupa ucapan lengkap dan *intent label* akan digabungkan ke dalam satu ruang vektor semantik. Setelah itu akan dilakukan *dot product loss* untuk melihat kemiripan *intent label*. Jika *intent* dari pengguna dapat dikenali maka chatbot akan memberikan respon yang sesuai. Sedangkan jika *intent* tidak dikenali maka chatbot akan meminta pengguna mengirimkan ulang pesan atau mengirimkan pesan bahwa chatbot tidak mengerti maksud dari pesan yang dikirimkan pengguna.

C. Training

Pada proses ini akan dilakukan training pada data NLU. Data NLU berisi kumpulan kalimat dan FAQ yang dikategorikan berdasarkan *intent*. Pada proses training menggunakan dataset kalimat sejumlah 227. Pada klasifikasi *intent* menggunakan epoch sebesar 100, didapatkan hasil train loss sebesar 0.66%. Dari proses training dihasilkan sebuah model yang nantinya akan digunakan untuk proses selanjutnya.



Gambar 6: Proses NLP

D. Webchat

Setelah proses training selesai maka chatbot akan diimplementasikan pada webchat. Webchat digunakan sebagai medium antara chatbot dan pengguna untuk saling berinteraksi melalui *web browser*. Webchat yang digunakan dapat mendukung interaksi antara chatbot dan pengguna dalam bentuk teks. Adapun tampilan *user interface* webchat yang digunakan dapat dilihat pada Gambar 7.



Gambar 7: Tampilan User Interface Webchat

E. Testing

Tahap terakhir adalah melakukan testing, yaitu melakukan pengujian terhadap model yang diperoleh dari tahap training. Pengujian akan dilakukan dengan mencoba klasifikasi *intent* dengan variasi input yang berbeda dan akan dilakukan dengan teks yang tidak ada di training data. Hal ini ditujukan untuk mengetahui kesesuaian balasan dari chatbot terhadap pesan yang dikirimkan oleh pengguna.

IV. Hasil dan Pengujian

Pada bagian ini akan dilakukan pengujian sebagai berikut:

A. Pengujian Klasifikasi Intent

Pengujian dilakukan untuk menilai klasifikasi *intent* yang dilakukan chatbot sudah sesuai terhadap pertanyaan yang diajukan oleh pengguna. Pengujian dilakukan dengan memasukkan pesan dengan tiga jenis variasi input yang berbeda dari training data untuk setiap *intent*, kemudian dilakukan penilaian hasil klasifikasi *intent* yang dilakukan oleh chatbot berdasarkan respon yang dikirimkan terhadap pesan yang dimasukkan sebelumnya. Jika hasilnya sudah sesuai dengan maksud dari pengguna maka akan ditulis “Sesuai” dan jika hasilnya tidak sesuai maka akan ditulis “Tidak Sesuai”. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1: Pengujian Klasifikasi Intent

No.	Intent	Pertanyaan 1	Pertanyaan 2	Pertanyaan 3
1	Menanyakan apakah ini bot	Sesuai	Sesuai	Sesuai
2	Video conference	Sesuai	Sesuai	Sesuai
3	Install dan aktivasi office365	Sesuai	Sesuai	Sesuai
4	Lupa password vps	Sesuai	Sesuai	Sesuai
5	Lupa password integra	Sesuai	Sesuai	Sesuai
6	Lupa username dan password Email ITS	Sesuai	Sesuai	Sesuai
7	Perhitungan IKITS	Sesuai	Sesuai	Sesuai
8	Realisasi IKITS	Sesuai	Sesuai	Sesuai
9	Forlap	Sesuai	Sesuai	Sesuai
10	Microsoft teams for education essentials	Sesuai	Tidak Sesuai	Tidak Sesuai
11	Lembar kerja microsoft teams for education essentials	Sesuai	Tidak Sesuai	Tidak Sesuai
12	Sharepoint	Sesuai	Sesuai	Sesuai
13	Onedrive	Sesuai	Sesuai	Sesuai
14	Power BI	Sesuai	Sesuai	Sesuai
15	myITS Classroom	Sesuai	Sesuai	Sesuai
16	Host webinar zoom	Sesuai	Sesuai	Sesuai
17	Pengambilan toga online	Sesuai	Sesuai	Sesuai
18	Pengembalian toga online	Sesuai	Sesuai	Sesuai
19	Memulai webinar dengan zoom	Sesuai	Tidak Sesuai	Sesuai
20	Aplikasi booking webinar	Sesuai	Sesuai	Sesuai
21	Publikasi internasional	Sesuai	Sesuai	Sesuai
22	Microsoft forms	Sesuai	Sesuai	Sesuai
23	Blog ITS	Sesuai	Sesuai	Sesuai
24	Wordpress	Sesuai	Sesuai	Sesuai
25	Pengamanan zoom untuk webinar	Sesuai	Sesuai	Tidak Sesuai
26	Live youtube	Sesuai	Sesuai	Sesuai
27	Huawei e-learning untuk dosen	Sesuai	Sesuai	Sesuai
28	Huawei e-learning untuk mahasiswa	Sesuai	Sesuai	Tidak Sesuai
29	Penggunaan dan Instalasi Matlab	Sesuai	Sesuai	Sesuai
30	Online meeting	Sesuai	Sesuai	Sesuai
31	Migrasi office365	Sesuai	Sesuai	Sesuai
32	Akses email office365	Sesuai	Sesuai	Sesuai
33	Mailing list office 365	Sesuai	Sesuai	Sesuai
34	Setting email pimpinan ITS	Sesuai	Sesuai	Sesuai
35	Aktivasi windows menggunakan kms	Sesuai	Sesuai	Sesuai
36	Setting web departemen	Sesuai	Sesuai	Sesuai
37	Akses internet tamu	Sesuai	Sesuai	Sesuai
38	Akses vpn	Sesuai	Sesuai	Tidak Sesuai
39	E-perkantoran untuk pegawai ITS	Sesuai	Sesuai	Sesuai
40	Panduan service desk	Sesuai	Sesuai	Sesuai

Berdasarkan hasil pengujian di atas, dapat diketahui bahwa dari 120 sampel uji klasifikasi *intent* didapatkan hasil 112 sampel yang hasilnya sesuai dengan maksud dari pesan yang dikirimkan pengguna.

Sedangkan 8 sampel yang hasilnya tidak sesuai disebabkan oleh kurangnya variasi input pada training data sehingga chatbot kesulitan untuk mengenali *intent* dari pesan yang dikirimkan oleh pengguna. Adapun untuk pesan dengan *intent* yang tidak dapat dikenali oleh chatbot, akan diberikan balasan sebagaimana pada Gambar 7.

Untuk akurasi keberhasilan, didapatkan dengan menggunakan Persamaan 1:

$$\text{Akurasi} = \frac{\text{Jumlah Keberhasilan}}{\text{Jumlah Pengujian}} \times 100\% \quad (1)$$

Dengan jumlah keberhasilan sebesar 112 sampel dan jumlah pengujian sebanyak 120 sampel, maka didapatkan presentase keberhasilan chatbot ini, yaitu sebesar 93%.

V. Kesimpulan

Dari pelaksanaan dan pengujian sistem yang sudah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Chatbot yang dihasilkan dapat melakukan klasifikasi *intent* dan memberikan respon dengan baik.
2. Dari 120 sampel pesan yang diujikan, didapatkan jumlah keberhasilan sebanyak 112 sampel. Presentase keberhasilan yang didapatkan sebesar 93%.
3. Kegagalan disebabkan oleh kurangnya variasi training data yang dimasukkan sebelum melakukan training.

Daftar Pustaka

- [1] N. K. Manaswi, N. K. Manaswi, and S. John, *Deep Learning with Applications Using Python*. Springer, 2018. .
- [2] R. Astuti, "Chatbot: Semua tentang chatbot," Oktober 2020.
- [3] R. Csaky, "Deep learning based chatbot models," arXiv preprint arXiv:1908.08835, 2019.
- [4] M. J. Garbade, "A simple introduction to natural language processing," Oktober 2018.
- [5] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. MIT Press Cambridge, 2016.
- [6] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.
- [7] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.
- [8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [9] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.
- [10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 27, pp. 3104–3112, 2014.
- [11] T. Bocklisch, J. Faulkner, N. Pawlowski, and A. Nichol, "Rasa: Open source language understanding and dialogue management," arXiv preprint arXiv:1712.05181, 2017.
- [12] G. Cukier, "Rasa architecture overview," Desember 2020.