

## CHATBOT VIRTUAL ASSISTANT HELP DESK DENGAN METODE AIML

Wendy Bunandi<sup>1</sup>, Viny Christanti Mawardi<sup>2</sup>

<sup>1</sup>Program Studi Teknik Informatika, Universitas Tarumanagara Jakarta  
Email: wendy.535200066@stu.untar.ac.id

<sup>2</sup>Program Studi Teknik Informatika, Universitas Tarumanagara Jakarta  
Email: viny@fti.untar.ac.id

Masuk : 27-11-2023, revisi: 07-12-2023, diterima untuk diterbitkan : 15-12-2023

---

### ABSTRAK

*Chatbot virtual assistant help desk* merupakan aplikasi yang dapat digunakan untuk melakukan percakapan virtual mengenai pertanyaan yang sering ditanyakan seperti *Frequently Asked Questions (FAQ)*. Aplikasi ini berguna untuk membantu menjawab *user* jika memiliki pertanyaan diluar jam kerja selama 24 jam. Metode penelitian yang digunakan dalam *chatbot virtual assistant help desk* ini adalah metode *artificial intelligence markup language (AIML)*. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *python* dan dengan menggunakan *library python flask*. Data yang digunakan untuk pengujian adalah data pertanyaan dan jawaban *Frequently Asked Questions (FAQ)* yang umumnya sering ditanyakan oleh *user* yang dikumpulkan kemudian dijadikan ke dalam bentuk format AIML agar dapat digunakan dalam aplikasi *chatbot virtual assistant help desk*. Pada pembuatan aplikasi ini dilakukan metode pengujian *artificial intelligence markup language (AIML)* untuk mengetahui kesesuaian hasil jawaban yang diberikan *chatbot* dengan dataset yang ada menggunakan 3 skenario, yaitu skenario pertanyaan yang sama persis dengan dataset, skenario pertanyaan yang 70% sama/mirip dengan dataset, dan skenario pertanyaan yang berbeda atau tidak ada pada dataset kemudian didapatkan kesimpulan bahwa hasil pengujian AIML dengan *blackbox testing* pada 3 skenario yang telah diujikan didapatkan bahwa semakin mirip masukan pertanyaan dengan pertanyaan pada dataset maka akan semakin tinggi hasil persentase kesesuaian jawaban *chatbot* dengan jawaban yang ada pada dataset ini dikarenakan *chatbot* merupakan *chatbot* berjenis *retrieval* yang mengeluarkan output sesuai dengan aturan atau *input* yang telah ditentukan sebelumnya. *Chatbot* ini juga berkontribusi dalam membantu menjawab pertanyaan pengguna diluar jam kerja.

**Kata Kunci:** Chatbot; Asisten Virtual; Help Desk; AIML; Flask

### ABSTRACT

*The help desk virtual assistant chatbot is an application that can be used to have virtual conversations about frequently asked questions such as Frequently Asked Questions (FAQ). This application is useful for helping users answer questions if they have questions outside of 24 hour working hours. The research method used in this virtual assistant help desk chatbot is the artificial intelligence markup language (AIML) method. The programming language used is the Python programming language and the Python Flask library is used. The data used for testing is Frequently Asked Questions (FAQ) question and answer data which is generally frequently asked by users which is collected and then converted into AIML format so that it can be used in the virtual assistant help desk chatbot application. In making this application, the artificial intelligence markup language (AIML) testing method was carried out to determine the suitability of the answers given by the chatbot with the existing dataset using 3 scenarios, namely question scenarios that are exactly the same as the dataset, question scenarios that are 70% the same/similar to the dataset, and question scenarios that are different or do not exist in the dataset, then it is concluded that the results of AIML testing with black box testing on the 3 scenarios that have been tested show that the more similar the input questions are to the questions in the dataset, the higher the results of the percentage of correspondence between the chatbot answers and the answers in the dataset. This dataset is because the chatbot is a retrieval type chatbot that produces output according to predetermined rules or input. This chatbot also contributes to helping answer user questions outside of working hours.*

**Keywords:** Chatbot; Virtual Assistant; Help Desk; AIML; Flask

## 1. PENDAHULUAN

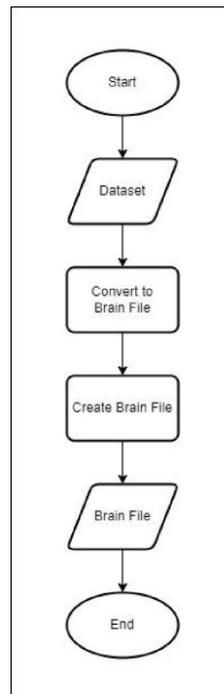
*Help Desk* adalah sistem yang dapat digunakan untuk membantu organisasi menanggapi kebutuhan pengguna untuk pertanyaan, layanan teknis, atau keluhan. Layanan yang dapat disediakan oleh *Help Desk* adalah instalasi komputer, pemeliharaan, dan pemecahan masalah. Aplikasi *Help Desk* dapat dengan cepat mengirimkan keluhan pengguna kepada pihak yang bertanggung jawab agar permasalahan dapat cepat teratasi. Fungsi *Help Desk* adalah untuk menghubungkan teknisi dan pengguna mengenai masalah layanan yang dilaporkan. *Help Desk* dapat digunakan untuk mengevaluasi kinerja departemen TI berdasarkan laporan yang dihasilkan, dan penggunaan *Help Desk* dalam perusahaan dapat meningkatkan produktivitas (Fauzi, M. Masrizal and Sihombing, V. 2021).

Akan tetapi *Help Desk* memiliki keterbatasan dalam waktu karena *Help Desk* menghubungkan langsung pengguna dan layanan pengguna yang dikelola manusia, sehingga ketika diluar dari jam kerja layanan pengguna yang dikelola manusia tidak dapat melayani dan tidak dapat membantu *user* Santoso, R. B. (2011). Seperti halnya *Help Desk* pada PT. Computrade Technology International yang tidak dapat melayani pengguna diluar jam kerja sehingga dibutuhkan lagi sebuah *chatbot virtual assistant* yang dapat melayani pengguna selama 24 jam sehari.

Karena itu penulis bertujuan untuk merancang dan membuat *chatbot virtual assistant help desk* dengan metode AIML sebagai asisten yang dapat membantu menjawab pertanyaan umum yang berkaitan dengan sistem selama 24 jam. *Chatbot virtual assistant* ini dirancang untuk mengatasi pertanyaan *user* diluar waktu jam kerja dan juga dapat membantu menjawab pertanyaan yang umum mengenai sinta dan sicti dari *user* sehingga tidak perlu menunggu respon dari manusia yang dapat mengefisiensi waktu untuk merespon *user* lain dengan pertanyaan yang rumit lebih cepat. Sistem *chatbot virtual assistant* ini dirancang menggunakan bahasa pemrograman *python* dan *framework flask* untuk diterapkan pada *website*. *Chatbot virtual assistant* ini juga bermanfaat untuk membantu tim *help desk* agar dapat melayani pertanyaan selain diluar jam kerja namun juga dapat melayani pada saat jam sibuk atau pada saat banyaknya pertanyaan kepada tim *help desk* sehingga *chatbot virtual assistant* ini dapat membantu menjawab *user* agar tidak terlalu lama merespon karena banyaknya pertanyaan sehingga dapat bertanya terlebih dahulu kepada *chatbot* dan menyelesaikan masalah. *Chatbot virtual assistant* berkontribusi untuk memudahkan perusahaan dan pengguna untuk mencari informasi menggunakan *chatbot* tanpa batasan waktu.

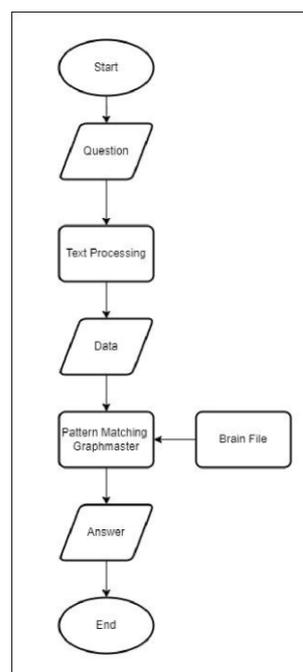
## 2. METODE PENELITIAN

Metode penelitian yang dilakukan adalah pada tahap pertama adalah melakukan perancangan tampilan antarmuka untuk login pengguna. Kemudian perancangan tampilan antarmuka *virtual assistant* dan juga melakukan perancangan *backend* dengan *Flask* pada sistem *chatbot virtual assistant help desk* agar dapat memproses pertanyaan dari *user* dan dapat memberikan respon jawaban dari dataset. Selanjutnya dibuat perancangan tampilan antarmuka untuk admin agar dapat menambah dan mengurangi atau menghapus pengguna dan juga untuk menambahkan data pertanyaan dan jawaban. Kemudian dibuat juga perancangan tampilan antarmuka tentang yang berisi informasi mengenai perancang dan pembuat *chatbot virtual assistant help desk* dengan metode AIML untuk PT. Computrade Technology International. Pada saat aplikasi *chatbot virtual assistant help desk* dijalankan maka sistem akan memproses dataset *aiml* yang ada untuk dilatih/training. Ketika dataset ini dilatih/*training* sistem akan membaca dataset yang sudah dalam bentuk *aiml* kedalam bentuk *binary* kemudian akan disimpan hasil latihan/*training* nya ke dalam *brain file*. Proses pelatihan/*training* ini dapat dilihat pada Gambar 1.



Gambar 1. Training

Setelah dataset dilatih/*training*, ketika pengguna memasukkan pertanyaan maka sistem *chatbot virtual assistant help desk* akan melakukan pengujian atau *testing*. Pada proses ini ketika pertanyaan dari pengguna masuk maka pada AIML itu sendiri akan melakukan *Preprocessing text* seperti *remove punctuation, uppercase, tokenisasi*. Kemudian data hasil preprocessing ini akan dicocokkan dengan algoritma graphmaster pattern matching dengan dan *brain file* atau data hasil latih/*training* yang sudah dilakukan. Setelah proses algoritma graphmaster pattern matching maka akan dikeluarkan jawaban. Proses pengujian atau testing ini dapat dilihat pada Gambar 2.



Gambar 2. Testing

## Preprocessing Text

Tahap *preprocessing text* atau tahap pra-pemrosesan teks merupakan langkah penting dalam *Natural Language Processing* (NLP) Chopra, A. Prashar, A. and Sain, C. (2013). Tahap *preprocessing text* atau tahap pra-pemrosesan teks ini adalah tahapan dimana aplikasi melakukan seleksi data teks yang akan diproses pada setiap dokumen. Tahap ini dilakukan agar input dan data dalam bentuk teks dapat memiliki format yang sesuai dengan kebutuhan. Tahap-Tahap *preprocessing text* atau pra-pemrosesan teks antara lain seperti *remove punctuation*, *uppercase*, *tokenization*. *Remove Punctuation* merupakan tahapan yang mencakup pembuangan tanda baca atau delimiter. *Uppercase* adalah proses mengubah karakter biasa menjadi karakter dalam huruf besar/kapital. *Tokenization* atau disebut juga sebagai tokenisasi dalam Bahasa Indonesia. Pada tahap ini dilakukan pemecahan string input dari yang awalnya berbentuk kalimat utuh menjadi kata-kata yang menyusunnya.

## AIML

*Artificial Intelligence Markup Language* (AIML) adalah bahasa yang mendeskripsikan objek data dan perilaku *program komputer* yang memprosesnya. AIML sendiri merupakan turunan dari *Extensible Markup Language* (XML). AIML berisi kumpulan pola dan respon yang dapat digunakan oleh *chatbot* untuk penelusuran jawaban setiap kalimat yang diberikan. *Interpreter* AIML diperlukan untuk menerima masukan pertanyaan dan melakukan penelusuran jawaban pada dokumen AIML Bahartyan, E. Bahtiar, N. and Waspada, I. (2015). Bagian-bagian penting dari AIML adalah sebagai berikut:

### 1. Category

Pada AIML, *category* adalah unit dasar dari pengetahuan. *Category* paling sedikit terdiri dari dua element AIML yaitu *pattern* dan *template* Pada AIML, *category* merupakan unit dasar dari pengetahuan. *Category* minimal terdiri dari dua element AIML yaitu *pattern* dan *template*. Contoh *category* adalah seperti pada Gambar 3.

```
<category>
<pattern>CARA MEMBUAT CLAIM</pattern>
<template>cara membuat atau mencreate claim dapat dilakukan dengan membuka modul
claim kemudian click create kemudian mengisi data claim kemudian save </template>
</category>
```

Gambar 3. Category

### 2. Pattern

*Pattern* adalah rangkaian huruf yang dimaksudkan untuk mencocokkan satu atau lebih masukan pengguna. *Pattern* dapat menggunakan *wildcard* yang cocok dengan satu atau lebih masukan pengguna. Contoh tag `<pattern>` adalah seperti pada Gambar 4.

```
<pattern>CARA MEMBUAT CLAIM</pattern>
```

Gambar 4. Pattern

### 3. Template

*Template* adalah jawaban atau respon terhadap suatu pertanyaan yang ditanyakan oleh pengguna. Suatu *template* menentukan respon berdasarkan *pattern* yang sesuai. Contoh tag `<template>` adalah seperti pada Gambar 5.

```
<template>cara membuat atau mencreate claim dapat dilakukan  
dengan membuka modul claim kemudian click create kemudian  
mengisi data claim kemudian save </template>
```

Gambar 5. Template

#### 4. Srai

Tag srai menunjukkan karakteristik penting dari AIML. Dengan menggunakan <srai> kita dapat menargetkan banyak <pattern> untuk satu <template>. Jadi akan menjawab secara efisien untuk berbagai hal input pengguna memiliki arti serupa. Contoh tag <srai> adalah seperti pada Gambar 6.

```
<category>  
  <pattern>TERIMA KASIH</pattern>  
  <template>SAMA-SAMA CITIZEN</template>  
</category>  
  
<category>  
  <pattern>TERIMA KASIH *</pattern>  
  <template><srai>TERIMA KASIH</srai></template>  
</category>
```

Gambar 6. Srai

### Grpmaster Pattern Matching

Grpmaster Pattern Matching adalah metode pencocokan pola yang digunakan dalam sistem chatbot. Metode ini digunakan untuk mencari pola-pola dalam teks input pengguna dan mencocokkannya dengan pola-pola yang telah ditentukan sebelumnya. Dengan menggunakan algoritma pencocokan pola, Grpmaster Pattern Matching membantu *chatbot* dalam mencari jawaban yang paling sesuai dengan pertanyaan atau pernyataan pengguna Triono, A. (2017). Metode Grpmaster Pattern Matching bekerja dengan membangun graf yang merepresentasikan pola-pola yang telah ditentukan sebelumnya. Graf ini terdiri dari simpul-simpul yang mewakili kata-kata atau frasa-frasa yang relevan. Setiap simpul dalam graf memiliki koneksi dengan simpul-simpul lain yang memiliki hubungan semantik atau struktural. Graf ini membantu *chatbot* dalam mencari pola-pola yang mirip dengan input pengguna. Ketika ada input teks baru, metode Grpmaster Pattern Matching mencocokkan pola input dengan pola-pola yang ada dalam graf.

Proses pencocokan pola ini melibatkan analisis struktur dan makna dari kata-kata atau frasa-frasa dalam input. Jika ada kesamaan pola yang cukup tinggi antara input dan pola yang ada dalam graf, maka jawaban yang sesuai akan ditemukan dan diberikan kepada pengguna Triono, A. (2017).

### 3. HASIL DAN PEMBAHASAN

Tabel dan gambar diletakkan di tengah halaman. Judul tabel ditulis di atas tabel, sedangkan judul Pada tahap pertama dilakukan pembuatan modul *login*, pada modul *login* terlebih dahulu dibuat *splash screen* yang akan muncul saat pertama kali *chatbot virtual assistant help desk* ini diakses. Tampilan *splash screen* ini terdapat pada Gambar 7.



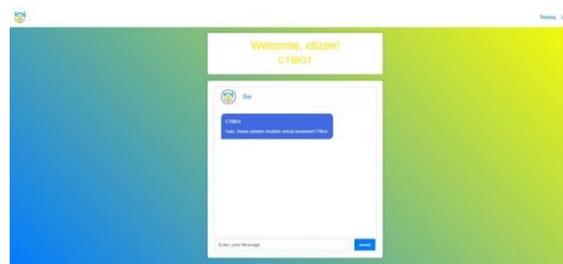
Gambar 7. *Splash screen*

kemudian dibuat tampilan *login* untuk melakukan *login* ke *chatbot virtual assistant help desk*. Tampilan *login* ini terdapat pada Gambar 8.



Gambar 8. *Login*

Kemudian dilakukan pembuatan modul *chatbot virtual assistant help desk* yang digunakan untuk melakukan komunikasi dan pada tampilan *chatbot* ini akan berbeda pada saat *login* sebagai *user* dan *admin*, dimana pada *admin* terdapat *navbar* untuk mengalihkan ke modul *admin*. Tampilan modul *chatbot virtual assistant help desk* saat *login* sebagai *user* ini terdapat pada Gambar 9.



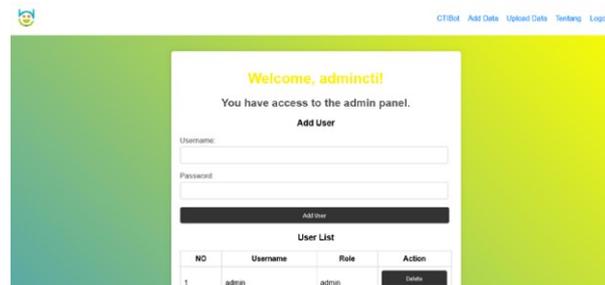
Gambar 9. *Chatbot sebagai user*

Tampilan modul *chatbot virtual assistant help desk* saat *login* sebagai *admin* ini terdapat pada Gambar 10.



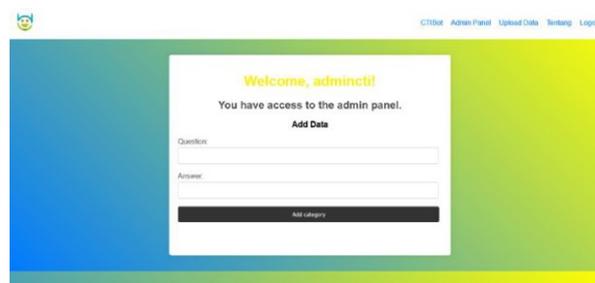
Gambar 10. *Chatbot sebagai admin*

Kemudian dilakukan pembuatan modul admin, pada modul admin ini pertama dibuat tampilan *admin panel* untuk melihat, menambah atau menghapus *user* yang dapat mengakses *chatbot virtual assistant help desk*. Tampilan *admin panel* terdapat pada Gambar 11.



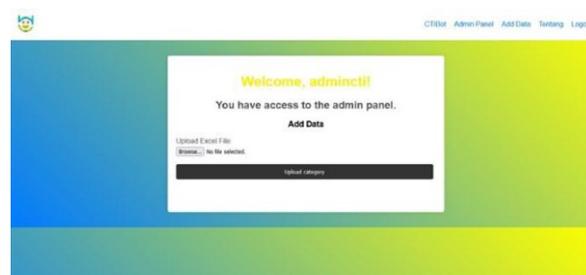
Gambar 11. Admin panel

kedua dibuat tampilan *add data* yang digunakan untuk menambahkan data secara manual yang terdapat pada Gambar 12.



Gambar 12. Add data

ketiga dibuat tampilan tambahan menambahkan data dengan cara *upload*, tampilan *upload data* ini digunakan untuk mengupload data menggunakan excel yang terdapat pada Gambar 13.



Gambar 13. Upload data

Kemudian dilakukan pembahasan pengujian AIML dengan *blackbox testing* untuk mengetahui kesesuaian hasil jawaban yang diberikan oleh *chatbot* berdasarkan skenario seperti skenario pertanyaan yang sama persis dengan dataset, skenario pertanyaan yang 70% sama/mirip dengan dataset, dan skenario pertanyaan yang berbeda atau tidak ada pada dataset. Contoh pengujian menggunakan *blackbox* dapat dilihat pada Tabel 1.

Tabel 1. contoh *blackbox testing*

no	pertanyaan	jawaban chatbot	jawaban dataset	kesesuaian
1	siapa bisa create company?	untuk create company hanya dapat dilakukan oleh tim CAM, Admin, TL Up.	untuk create company hanya dapat dilakukan oleh tim CAM, Admin, TL Up.	sesuai
2	lama data contact invalid?	untuk bagian primary dianggap invalid 6 bulan, untuk bagian secondary dianggap invalid 12 bulan.	untuk bagian primary dianggap invalid 6 bulan, untuk bagian secondary dianggap invalid 12 bulan.	sesuai
3	cara update data company?	update company bisa dilakukan dengan mencentang checkbox sebelum melakukan update pada field yang diinginkan.	update company bisa dilakukan dengan mencentang checkbox sebelum melakukan update pada field yang diinginkan.	sesuai
4	lama data company invalid?	untuk bagian primary dianggap invalid 6 bulan, untuk bagian secondary dianggap invalid 12 bulan.	untuk bagian primary dianggap invalid 6 bulan, untuk bagian secondary dianggap invalid 12 bulan.	sesuai
5	bisa set pic di company?	yang dapat set PIC di company/contact adalah: CDM, Developer SINTA, Admin SILO, TL/Manager/GM SILO hanya bisa set membeinya (1 layer below).	yang dapat set PIC di company/contact adalah: CDM, Developer SINTA, Admin SILO, TL/Manager/GM SILO hanya bisa set membeinya (1 layer below).	sesuai
6	membuat claim?	Maaf, saya tidak mengerti pertanyaan Anda.	cara membuat atau mencreate claim dapat dilakukan dengan membuka modul claim kemudian click create kemudian mengisi data claim kemudian save	tidak sesuai
7	buat claim?	Maaf, saya tidak mengerti pertanyaan Anda.	cara membuat atau mencreate claim dapat dilakukan dengan membuka modul claim kemudian click create kemudian mengisi data claim kemudian save	tidak sesuai
8	siapa dapat delete pipeline?	yang dapat melakukan delete pipeline adalah TL Up (TL, Manager, GM, BOD), namun jika pipeline tersebut merupakan pipeline synergy maka perlu melakukan Request To Update (RTU) ke PIC Synergy.	yang dapat melakukan delete pipeline adalah TL Up (TL, Manager, GM, BOD), namun jika pipeline tersebut merupakan pipeline synergy maka perlu melakukan Request To Update (RTU) ke PIC Synergy.	sesuai
9	bisa delete pipeline?	yang dapat melakukan delete pipeline adalah TL Up (TL, Manager, GM, BOD), namun jika pipeline tersebut merupakan pipeline synergy maka perlu melakukan Request To Update (RTU) ke PIC Synergy.	yang dapat melakukan delete pipeline adalah TL Up (TL, Manager, GM, BOD), namun jika pipeline tersebut merupakan pipeline synergy maka perlu melakukan Request To Update (RTU) ke PIC Synergy.	sesuai
10	siapa dapat update pipeline?	yang dapat melakukan update status pipeline adalah TL Up dan Admin SILO apabila status pipeline sudah on going, po receive, delivery.	yang dapat melakukan update status pipeline adalah TL Up dan Admin SILO apabila status pipeline sudah on going, po receive, delivery.	sesuai

Pembahasan pada 3 skenario yang telah diujikan didapatkan bahwa semakin mirip masukan pertanyaan maka akan semakin tinggi persentase kesesuaian jawaban dari *chatbot* ini dikarenakan *chatbot* CTIBOT merupakan *chatbot* berjenis *retrieval* yang mengeluarkan *output* sesuai dengan aturan atau *input* yang telah ditentukan sebelumnya. Masukan pertanyaan yang kemiripannya <70%

atau tidak mirip dengan dataset kemungkinan besar menghasilkan jawaban yang tidak sesuai, karena AIML menggunakan graphmaster pattern matching untuk mencari kemiripan masukan pertanyaan dengan pertanyaan yang ada pada dataset untuk mengeluarkan jawaban dari dataset sehingga semakin tidak mirip dengan dataset maka kemungkinan semakin tidak sesuai jawaban yang di berikan chatbot semakin besar.

#### 4. KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan didapatkan kesimpulan bahwa hasil pengujian AIML dengan *blackbox testing* didapatkan bahwa semakin mirip masukan pertanyaan dengan pertanyaan pada dataset maka akan semakin tinggi hasil persentase kesesuaian jawaban *chatbot* dengan jawaban yang ada pada dataset. Pengujian dengan skenario kedua mulai terdapat jawaban yang tidak sesuai dan pada skenario ketiga semakin banyak jawaban yang tidak sesuai, jika diatas 70% kemiripan pertanyaan dengan dataset maka masih dapat memberikan jawaban yang sesuai, sedangkan ketika dibawah 70% kemiripan masukan pertanyaan dengan dataset maka jawaban kemungkinan tidak sesuai. Terdapat juga saran yaitu Memperbanyak lagi dataset pertanyaan dan jawaban dan dengan memperbanyak komponen AIML untuk di implementasikan pada dataset yang digunakan seperti tag *srai*, *star*, dan tag lainnya dan juga Meningkatkan tampilan aplikasi agar lebih responsif dan lebih *modern* seperti dapat juga menggunakan *framework* agar lebih menarik dan lebih bagus khususnya tampilan *frontend* dengan mempertimbangkan *ui/ux*.

#### Ucapan Terima Kasih

Ucapan Terima kasih disampaikan kepada Bapak Boy sebagai mentor/*leader* pada perusahaan CTI dan juga kepada keluarga dan teman-teman yang telah mendukung dan rekan-rekan kerja yang telah membantu.

#### REFERENSI

- Bahartyan, E. Bahtiar, N. and Waspada, I. (2015). "Integrasi Chatbot Berbasis AIML Pada Website E-Commerce Sebagai Virtual Assistant Dalam Pencarian Dan Pemesanan Produk (Studi Kasus Toko Buku Online edu4indo.com)". JURNAL MASYARAKAT INFORMATIKA, vol. 5, no. 10, pp. 34-43.
- Bariah, S. H. Pratiwi, W. and Imania, K. A. N. (2022). "Pengembangan Virtual Assistant Chatbot Berbasis Whatsapp Pada Pusat Layanan Informasi Mahasiswa Institut Pendidikan Indonesia-Garut". Jurnal Pendidikan Teknologi Informasi Dan Komunikasi, vol. 8, no. 1, pp. 66-79.
- Chopra, A. Prashar, A. and Sain, C. (2013). "Natural language processing," International journal of technology enhancements and emerging engineering research, vol. 1, no. 4, pp. 131-134.
- Fauzi, M. Masrizal and Sihombing, V. (2021). "Sistem Informasi It-Helpdesk Universitas Labuhanbatu Berbasis Web". JURTEKSI (Jurnal Teknologi dan Sistem Informasi), vol. 7, no. 3, pp. 259-266.
- Irsyad, R. (2018). "Penggunaan Python Web Framework Flask Untuk Pemula"
- Ivanedra, K, and Mustikasari, M, (2019). "Implementasi Metode Recurrent Neural Network Pada Text Summarization Dengan Teknik Abstraktif,". Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), vol. 6, no. 4, pp. 377-382.
- Maskur, Y. Afandi, A. Waris and Afirianto, T. (2022). "Prototipe Virtual Assistant Chatbot Sebagai Pusat Layanan Informasi Mahasiswa". Jurnal Teknik Ilmu dan Aplikasi, vol. 3, no. 1, pp. 35-39.
- Santoso, R. B. (2011). "Rancang Bangun Prototype Chat Bot Customer Service System Berbasis Web," Skripsi. Fak. Sains dan Teknol. Tek. Inform. Univ. Islam Negeri Sultan Syarif Kasim, Riau.

- Triono, A. (2017). "Rancang Bangun Aplikasi Chatbot Berbasis AIML Dengan Metode Pattern Matching (Studi Kasus: Akademik UIN SUSKA RIAU)," Doctoral dissertation, Universitas Islam Negeri Sultan Syarif Kasim Riau.
- Vyshnavi, V. R. and Malik, A. (2019). "Efficient Way of Web Development Using Python and Flask", International Journal of Recent Research Aspects, vol. 6, no. 2, pp. 16-19.