JTKSI, Vol.04 No.03 September 2021 ISSN : 2620-3022
Hal. 101-108

Frequenly Ask Question (FAQ) Chatbot for New Student Admission System Using Natural Language Processing at Politeknik Aisyiyah Pontianak

Safri Adam[1], Eliyana Lulianthy[2]
[1]Information Technology Study Program, Politeknik Aisyiyah Pontianak, Pontianak
[2]Midwifery Study Program, Politeknik Aisyiyah Pontianak, Pontianak
[1,2]Ampera Street No.09, Pontianak, West Kalimantan, Indonesia
Email: safriadam@polita.ac.id[1], eliyana.lulianthy@gmail.com[2]

Received: 30 Juni 2021 Revised: 31 Juli 2021 Accepted: 5 Agustus 2021

Abstract—CS (customer service) is a crucial part in an agency that deals with customers, including universities. Especially on the part of the new student admissions committee. However, the ability of CS handled by humans is very limited. In addition, the questions asked are often repeated. So we need an AI-based chatbot (Artificial intelligence) to answer questions from prospective new students. The purpose of this research is to design and build a chatbot system using NLP (natural language processing) to assist the CS of the new student admissions committee at Aisyiyah Polytechnic Pontianak in handling questions from prospective new students. This study uses the NLP method to recognize natural language input from the user. For the weighting of the answers using TF-IDF and cosine similarity. The chatbot system is implemented in the python programming language. The messaging application used is Telegram using the AIOGRAM library and BotFather as a provider of bots on Telegram. This research resulted in a chatbot application using Telegram which was built using the Python programming language and has been trained to recognize the natural language input from the user. From the test results obtained a recall rate of 88% and a precision of 66%. The chatbot application that was built can help the new student admissions committee in serving questions for prospective new students. The evaluation results show that the built chatbot is able to answer questions from users well.

Keywords: Chatbot, NLP, Telegram, TF-IDF, cosine similarity.

I. INTRODUCTION

The development of technology and information is currently a demand in agencies, organizations and companies, especially information technology which is carried out through internet media. This is realized by the integration of computers that are able to store and manage various kinds of data quickly, precisely and accurately. Along with the development of information technology, a company, organization or agency is required to be able to mutually improve the quality and quality of information systems in order to compete in the current era of globalization, like universities. According to PDDIKTI data in 2020, the number of universities in Indonesia reached 4645 universities [1]. There are polytechnics, academies, high schools, institutes, universities, community academies. Competition between universities, especially those managed by the private sector, in addition to improving the quality of education and teaching is in increasing the number of students. To recruit students at universities, it is necessary to manage new student admissions quickly and easily. So that prospective students do not need much effort to register themselves.

One of the efforts to increase the number of student interest is by increasing the facilities for new student admissions. This is done so that prospective students are easy in the registration process. Some efforts to increase the number of students is by conducting campus promotions by creating web pages, soft file posters there is a CS (customer service) facility information that handles questions from prospective students who want to register. This CS usually uses social media to communicate with each other. In handling questions from prospective students, CS must always be available and fast response in serving questions from prospective students 24 hours a day [2]. However, the ability of CS is limited due to the large number of prospective students who ask questions so that CS is not handled properly. In addition, the questions posed to CS are often repeated on the same topic. So that CS is overwhelmed and feels bored answering questions in the same category [3].

A chatbot research to solve a similar problem was conducted by Halimatus in 2020 [4]. In this study, a chatbot was built using Telegram messaging. One of the advantages of Telegram is its open API for development. Chatbot applications are built using certain patterns or patterns that are stored in the chatbot system. Users cannot send messages by typing like replying to messages with humans, but only pressing a button where each button represents a question and when clicked it produces a prepared answer. So that some questions cannot be asked directly to the chatbot.

A similar study was also conducted by Hormansyah in 2018 [2]. In this study, a web-based chatbot application was built for public health services in the city of Malang using the TF-IDF method. However, the weakness of web-based applications is that they are not mobile-friendly, where one of the current human needs is always on the move. The method used is by making questions from users as keywords to search for documents related to public health services. So that the interaction between the chatbot and the user is only limited to searching for specific documents. Unable to handle queries in the natural language of the user.

AI (artificial intelligence) are currently developed technology based that is used to solve human problems. Until now, AI is widely used in various fields and is able to solve more complex problems, including in the field of public services [5].

Therefore, an application using AI technology is proposed that can handle questions from users who can serve 24 hours a day in the form of a chatbot application. Chatbot is an AI-based application designed and trained to automatically reply to human messages via text messages [6]. In its design, chatbots are equipped with artificial intelligence capabilities that can answer questions from humans that are built on certain topics that have been modeled in the knowledge base. In processing chat from the user in the form of text, the AI is built using a natural language processing-based method (NLP: natural language processing) where this method is used to process natural language input from the user [7].

II. THEORETICAL

A. *Chatbot*

Chatbot applications are made for various purposes, such as customer service for companies, agencies, and organizations or just for regular conversations [8]. Input from the chatbot user then provides a response that is in accordance with the knowledge possessed. The response given depends on the knowledge that has been trained (training) owned by the chatbot itself [9]. Knowledge is stored in various forms such as simple text document files or in database systems. The messaging application used is the Telegram application. Telegram is a cloud-based instant messaging, videocall and voice over IP service with end-to-end encrypted chat only for secret chats [10].

B. *TF-IDF*

The most popular term weighting is TF (term frequency). TF counts the number of occurrences of a term in a document. The indicator is that the more often the word appears in the document, the more it represents the label of the document [11]. The formula for calculating TF is according to equation 1. Where *Tf* (term frequency) is obtained from the sum of 1 and the *log* frequency of the calculated word..

$$TF = 1 + \log(TF) \tag{1}$$

The TF-IDF (Term Frequency Inverse Document Frequency) method is used to weight the relationship between a term and a document. The TF-IDF method combines two concepts for weight calculation, namely, the frequency of occurrence of a word in a particular document and the inverse frequency of the document containing the word. The formula for calculating IDF can be seen from equation 2.

$$Idf = \log \frac{D}{Df} \tag{2}$$

Where *IDF* is the inverse document frequency obtained from log *D* (total documents) divided by *Df* (document frequency of words/terms). Then the weight is calculated by multiplying the results of *TF* and *IDF*, so that the results of the weight of the document against the query in this case are the question document from the user.

C. *Cosine Similarity*

Measurement of similarity between documents can be calculated using the cosine similarity method. Cosine similarity calculates the angular distance between 2 documents that have a small angle number which means that the two documents are similar. The formula for calculating similarity using cosine similarity can be seen in equation 3.

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \bullet \vec{d}}{|\vec{q}||\vec{d}|} \tag{3}$$

Where *cos(q,d)* is the cosine equation of *q* and *d* or, equivalently, the cosine of the angle between *q* and *d* obtained from the dot product between (weight *tf-idf* of term *i* in the query) with (is the weight of *tf-idf* of term i in the document) divided by the absolute product of and . [12], [13]

III. METHOD

A. System architecture

The architecture of the chatbot system in this study can be seen in Fig. 1. Chat from user (users) is inputted into the system is then processed using NLP (natural language processing) where the output is data in the form of text that has been cleaned and the information extracted in it. Then, based in this information, decisions are made to take answers from the system. In decision making, the system accesses the FAQ (frequently ask and question) document to measure the similarity with the input text from the user. After obtaining the relevant documents, the documents are then forwarded to the message generator, in this case the AIOGRAM framework which provides an instant system to receive and send messages automatically through the telegram instant messaging application.

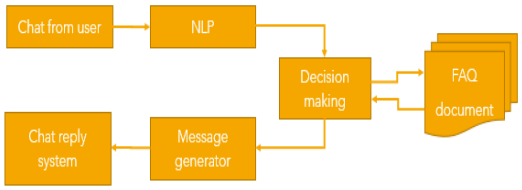

Fig 1. Chatbot system architecture

The data used in this study is primary data obtained from the results of collecting chat messages from prospective students who were asked to the CS committee of the PMB Aisyiyah Polytechnic Pontianak. Chat questions are collected from July 2020 to August 2020. The chat collection is then entered into a *.txt* formatted document. From the results of data collection, 16 pairs of questions and answers were often asked by prospective students. The data then pre-processed which aims to clean the data so that it can be processed by the system. Text processing is done in 2 technique, manually and automatically. Automatic text preprocessing is done by correcting typos or typos [14]. Cleaning this data is useful for important words in the document are not eliminated when processed by the system. So, it is hoped that the system will provide a more accurate answer. While text preprocessing is automatically carried out by the system where it aims to convert text into an executable form in the system. The text pre-processing process automatically can be seen in Fig 2.

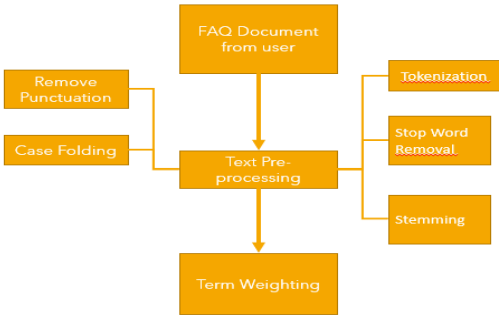

Fig 2. Text processing

B. Answering retrieval method

After weighting each word, then the next step is answer retrieval. A complete chart of the method of drawing explanations can be seen in Figure 3. In the figure it is explained that each question from the user is measured using *TF-IDF* documents and the similarities described previously. Then the results of the assessment are sorted by the documents with the highest scoring figures, then that document is chosen as the answer. But set a threshold where this is a weight limit. If the weight is below the threshold, the system will issue an error message in the form of text: "maaf anda tanya apa ?, mohon bertanya lebih spesifik". A weight below the threshold means that the query from the user does not match the answer document prepared by the system at all.

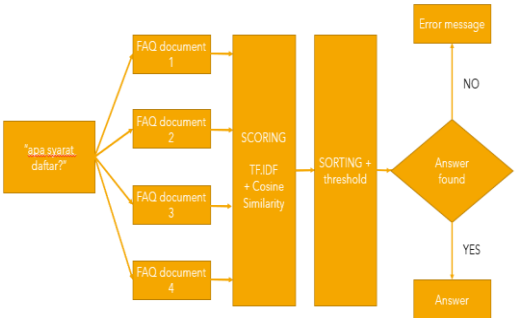

Fig 3. The method of answers retrieval from the chatbot system

C. Experiment Equipment Specification

In testing the system, computer hardware and software are needed that can support the development of a chatbot system. The specifications of the test equipment can be seen in Table 1.

Table 1. Experiment Equipment Specification

	Developer specifications	User and third party specifications
Hardware	Computer laptop, *Processor Intel®Core ™ i57200U CPU @ 2.50GHz, RAM 4 GB.*	*Processor Intel®Core ™ i5-7200U CPU @ 2.50GHz, RAM 12 GB.*

Software	*OS Windows 10, Python 3.8.3, Telegram API, @BotFather, Library AIOGRAM, Wireless Communication protocol 802.11n WPA2Personal 2,4 GHz*	*OS Android, OS Windows, Telegram, Wireless Communication protocol 802.11n WPA2Personal 2,4 GHz*

D. Evaluation Method

Evaluation are carried out to ensure that the system that has been designed and built functions according to its purpose. Then an evaluation is carried out to measure how well the system is built. The test scenario was carried out by giving 15 questions to the chatbot. then measured using a confusion matrix. After measuring, it is calculated using the recall and precision formulas. Recall is the ratio of the number of documents that can be provided by a search process in the system which can be calculated using equation 4.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

While precision is the ratio of documents found and considered relevant by the user which can be calculated by equation 5.

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Where TP (*true positive*) is the number of chatbot system giving correct answers, FP (*False negative*) is the number of questions answered by the chatbot system but not appropriate. While FN (*false negative*) is the number of questions that the chatbot system cannot answer [15].

IV. DISCUSSION

The chatbot program is implemented into the Python programming language version 3.8.3. In implementing the program, there are several important functions in running the program, including the function to calculate *TF-IDF* , *Cosine similarity*, and answers retrieval. The line of code for measuring *TF-IDF* weights is as follows:

```
def tfidf(query, data, stopWords, N):
    #build a list of tokenized docs
    words_list = []
    for doc in data:
        doc = re.sub('[^A-Za-z0-9]+',' ',doc)
# you may want to do more pre-processing here...
     words_list.append([word.lower()
     for word in
     nltk.word_tokenize(doc) if word
     not in stopWords])
      # a list of all words, with duplicates
      all_words = list(itertools.chain(*words_list))
     # a list of all words, without duplicates - for vector bases
      word_set = list(set(all_words))
     # construct tf vectors
      tf_veCS  = [0 for i in range(N)]
      for i in range(N):
          tf_veCS [i] = [words_list[i].count(w) for w in word_set]
     # compute idf values
      idf_all_words = list(itertools.chain(*[set(doc_words) for doc_words in words_list]))
     idfs = [math.log(float(N) / idf_all_words.count(w), N) for w in word_set]
     # compute tf-idf & normalize
     tfidf = [0 for i in range(N)]
     nomDc=[];
     for i in range(N):
         tfidf[i] = [tf * idf for tf, idf in zip(tf_veCS [i], idfs)]
         nomD = math.sqrt(sum(x**2 for x in tfidf[i]))
         tfidf[i] = [x / nomD for x in tfidf[i]]
         nomDc.append(tfidf[i])
     # now let's work on the query vector
     query = re.sub('[^A-Za-z0-9]+',' ', query)
     qwords = [word.lower() for word in query.split() if word not in stopWords]
     # tf vector
     qvec = [qwords.count(w) for w in word_set]
     # tf-idf vector
     qvec = [tf * idf for tf, idf in zip(qvec, idfs)]
     # normalize
     nomQ = math.sqrt(sum(x**2 for x in qvec))
     try:
       qvec = [x / nomQ for x in qvec]
     except ZeroDivisionError:
       print('divide by zero')
     #query
     return qvec, nomDc
```

```
```

The line of code for the function to calculate cosine similarity is as follows:

```
def cosinsimiliraty(nDoc,nQue):
    result = []
    for x in range(len(nDoc)):
        cosin=[]
        for y in range(len(nDoc[0])):

cosin.append(nDoc[x][y]*nQue[y])
        count = (sum(cosin))
        result.append(count)
    return result
```

The sample code line of the function for retrieving answers is as follows:

```
def cosinsimiliraty(nDoc,nQue):
    result = []
    for x in range(len(nDoc)):
        cosin=[]
        for y in range(len(nDoc[0])):

cosin.append(nDoc[x][y]*nQue[y])
        count = (sum(cosin))
        result.append(count)
    return result
```

All functions call each other in running the chatbot system and are stored in 1 python file and named *app.py*. Then to run the command then run the following command in the command prompt window:

```
>>> python app.py
```

The chatbot implemented for telegram messaging is named '@Sipenmaru_Polita_Bot' which can be accessed at http://t.me/sipenmaru_polita_bot. The chatbot display implemented in telegram messaging can be seen in Fig. 4.

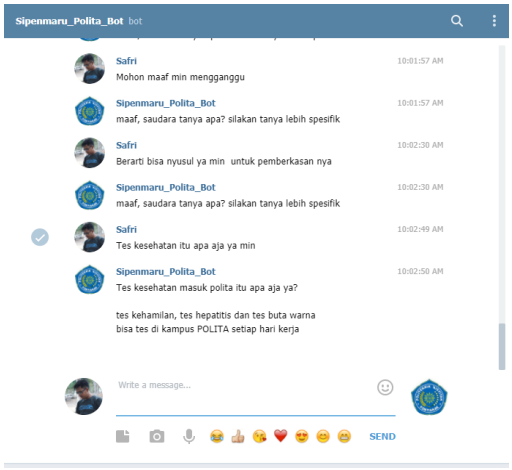

Fig 4. Chatbot on telegram

A. The results of the measurement of the similarity of the answers of TF-IDF and Cosine Similarity

When measuring the similarity of the answer to the question document, a certain similarity weight will be accepted as the most similar document and used as an answer. From the test results obtained several pairs of questions and answers that were asked to the chatbot. The results of the similarity and weight measurements can be seen in Table 2.

Table 2. Weight of *COSINE SIMILARITY*

No	Question	*Cosine weight*	Answer chatbot
1	Kak saya mau nanya saya mau daftar di aisyiyah kapan pembukakan nya kak?	0.182391375	bebas_test_biaya
2	Gini kak saya mau daftar di aisyiyah kebidanan saya dari smk keperawatan persyaratan nya apa2 aja kak?	0.254767111	syarat_bidan
3	Kak kan klau online di situ kan ada data orang tua, orang tua saya udah gk ada jadi boleh di isi nama wali gk kak	0.747723541	ortu_wali
4	Wah kak untuk biaya spp nya kebidanan berapa ya kak?	0.515565037	uang_spp_bidan
5	Daftar ulang nanti lansung ke kampus kah kak?	0.257885777	bebas_test_biaya
6	Kak ada asrama juga ya	0.55869108	asrama
7	kak fasilitas asrama apa aja ya kak?	0.384073202	asrama

8	Oh ya kak di aisyiyah ada beasiswa gk?	0.287012086	beasiswa
9	Permisi ibu, setelah melengkapi berkas selanjutnya apa ya bu?	0.185754662	uang_spp_ti
10	Perkuliahan tatap muka kapan dimulai ya kak?	0.195739394	perkuliahan
11	Test kesehatannya kapan kak	0.260530285	bebas_test_biaya
12	assalamualaikum	0	maaftanyaapa
13	Mohon maaf min mengganggu	0.362958368	maaftanyaapa
14	Berarti bisa nyusul ya min untuk pemberkasan nya	0.075862248	maaftanyaapa
15	Tes kesehatan itu apa aja ya min	0.681046898	test_kesehatan_masuk_polita

Based on Table 2, that similar documents have a large weight. From this weight, a threshold is set that each question that has a weight of <=0.15 means that it does not have a suitable answer document. So, it returns an answer with the label 'maaftanyaapa'.

B. Hasil Evaluasi

Evaluation is done by asking questions to the chatbot, then analyzing the results of the answers and then assessing them which is represented in the confusion matrix. The confusion matrix can be seen in table 3.

Table 3. CONFUSSION MATRIX

No	Question	Chatbot's answer (label)	TP	FP	FN	TN
1	Kak saya mau nanya saya mau daftar di aisyiyah kapan pembukakan nya kak?	bebas_test_biaya		1		
2	Gini kak saya mau daftar di aisyiyah kebidanan saya dari smk keperawatan persyaratan nya apa2 aja kak?	syarat_bidan	1			
3	Kak kan klau online di situ kan ada data orang tua, orang tua saya udah gk ada jadi boleh di isi nama wali gk kak	ortu_wali	1			
4	Wah kak untuk biaya spp nya kebidanan berapa ya kak?	uang_spp_bidan	1			
5	Daftar ulang nanti lansung ke kampus kah kak?	bebas_test_biaya		1		
6	Kak ada asrama juga ya	asrama	1			
7	kak fasilitas asrama apa aja ya kak?	asrama	1			
8	Oh ya kak di aisyiyah ada beasiswa gk?	beasiswa	1			
9	Permisi ibu, setelah melengkapi berkas selanjutnya apa ya bu?	uang_spp_ti		1		
10	Perkuliahan tatap muka kapan dimulai ya kak?	perkuliahan	1			
11	Test kesehatannya kapan kak	bebas_test_biaya		1		
12	assalamualaikum	maaftanyaapa				1
13	Mohon maaf min mengganggu	maaftanyaapa				1
14	Berarti bisa nyusul ya min untuk pemberkasan nya	maaftanyaapa			1	
15	Tes kesehatan itu apa aja ya min	test_kesehatan_masuk_polita	1			
	TOTAL		**8**	**4**	**1**	**2**

In the table, the question column is all the test questions asked by the user to the chatbot. In the chatbot answer column, the answer from the chatbot is where the label is enough to be written in the table. In that column there is a label 'maaftanyaapa', this is a label for the chatbot's answer if you don't find the answer asked by the user.

Evaluation of research results using the confusion matrix method and the parameters used are recall and precession. Recall calculates the ratio of the number of relevant answers that have been successfully retrieved by the chatbot to the total number of answers in the question-and-answer document that are considered relevant. While precision is the ratio of the number of relevant answers found to the total number of question and

answer documents. The evaluation results in this study obtained a recall value of 88% and precision of 66%.

The evaluation number obtained is above 50% which shows that the chatbot application that has been built is quite good. The recall number is obtained because it still produces False Negative (FN) answers which can affect the divisor when calculating the recall formula. A false negative answer means that the chatbot does not provide an answer even though the answer is already available in the question and answer document. Some of the limitations that exist in this study are that the system still gives wrong answers. This is evidenced by the recall results of 88%, so that is mean 12% possibility the system will return the wrong answer. This is happen because the answer document data prepared by the system is still limited, so the system cannot reply to chat in the form of a greeting. This can be proven by the results of the chatbot's answer in Table 3 row number 12, when the user greets with the sentence "Assalamualaikum", the chatbot replies with a response that does not find an answer.

With the results of the recall and precession numbers obtained, the results of this study in the form of a chatbot can be used to assist the task of the new student admissions committee in serving questions from prospective new students who want to register at the Aisyiyah Polytechnic Pontianak. The chatbot application that was built has the advantage that it can be accessed anywhere compared to research [2]. In addition, this study only uses user questions as queries (keywords), so that the answers given by the chatbot do not focus on user questions, but provide answers that are similar to queries. The problem in research [2] can be overcome in the research that the author did by adding NLP (natural language processing). In the process, NLP helps the system sort out words that are not needed in the weight measurement, so that when measuring the similarity, they can find answer documents that are relevant to the user's questions.

In research [4], namely building a telegram bot to meet the information needs of the academic community of the UIN Imam Bonjol Padang campus. They have limitations, namely only using pattern matching to match questions with answers. So that the questions that the user wants to ask are very narrow because they are only limited to patterns or patterns provided by the system. The limitations of this research can be overcome by the our research.

V. CONCLUSION AND SUGGESTION

Based on the results of research and discussion, the following conclusions can be drawn. A chatbot system has been built using NLP (natural language processing) for Sipenmaru Aisyiyah Polytechnic Pontianak which can process natural language inputted by the user. The limitations of previous research can be overcome by adding NLP to the chatbot so that the answers given by the system are relevant to the user's questions. The combined method of TF-IDF and Cosine similarity is able to handle the weighting and measurement of similarity between the questions and answers returned by the chatbot. The results of the evaluation of the chatbot system obtained a recall rate of 88% and precision 66%. This figure shows that the built chatbot is able to handle questions from users quite well.

Based on the results of the research and the shortcomings in this study, several suggestions were obtained for further research. Chatbots can be developed to handle questions in the form of photos and other media. The chatbot system can be combined with similarity measurement methods to increase recall and precision. The FAQ (frequently ask and question) document data used can be reproduced and added documents to overcome chatbot errors in answering greetings.

Reference

[1] KEMENRISTEKDIKTI, “Grafik Jumlah Perguruan Tinggi,” *PANGKALAN DATA PENDIDIKAN TINGGI*, 2020. [Online]. Available: https://forlap.ristekdikti.go.id/perguruantinggi/homegraphpt.

[2] D. S. Hormansyah, Y. P. Utama, P. Studi, T. Informatika, J. T. Informasi, and K. Malang, “APLIKASI CHATBOT BERBASIS WEB PADA SISTEM INFORMASI LAYANAN PUBLIK KESEHATAN DI MALANG DENGAN MENGGUNAKAN METODE TF-IDF,” vol. 4, pp. 224–228, 2018.

[3] E. Tanuwijaya, S. Adam, M. Fatoni, and A. Zainal, “Query Expansion menggunakan Word Embedding dan Pseudo Relevance Feedback,” vol. 5, no. 34, pp. 47–54, 2019.

[4] H. Sa’adah, S. Doni, and Syamsir, “Pemanfaatan Bot Telegram Untuk Pemenuhan Kebutuhan Informasi Sivitas Akademika UIN Imam Bonjol Padang Selama Pandemi Korona,” *Tadwin J. Ilmu Perpust. dan Informasi*, vol. 1, no. 1, pp. 32–43, 2020.

[5] N. Fransisko Agusning, “Pembangunan Aplikasi Sistem Pendaftaran Poliklinik Rumah Sakit Berbasis Android Pada Rumah Sakit Umum Bina Sehat.,” 2018.

[6] A. Iswandi, “IMPLEMENTASI CHATBOT PADA ORDER MANAGEMENT SYSTEM USAHA MIKRO KECIL MENENGAH,” Universitas Islam Indonesia, 2018.

[7] A. S. A Ferico Octaviansyah, Dedi Darwis,

"SISTEM PENCARIAN LOKASI BENGKEL MOBIL RESMI BAHASA ALAMI," *TEKNOINFO*, vol. 13, no. 2, pp. 71–77, 2019.

[8] A. Dewi and B. Setiaji, "PEMANFAATAN SENTENCE-SIMILARITY MEASUREMENT UNTUK PROSES PENCARIAN POLA PADA CHATBOT BERBASIS PATTERN-MATCHING," pp. 39–44, 2014.

[9] R. Singh, M. Paste, N. Shinde, H. Patel, and N. Mishra, "Chatbot using TensorFlow for small Businesses," *2018 Second Int. Conf. Inven. Commun. Comput. Technol.*, no. Icicct, pp. 1614–1619, 2018.

[10] C. Anglano, M. Canonico, and M. Guazzone, "Forensic Analysis of Telegram Messenger on Android Smartphone," *Digit. Investig.*, 2017.

[11] R. T. Wahyuni, D. Prastiyanto, and E. Supraptono, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," vol. 9, no. 1, 2017.

[12] A. R. Lahitani, A. E. Permanasari, and N. A. Setiawan, "Cosine Similarity to Determine Similarity Measure : Study Case in Online Essay Assessment," *2016 4th Int. Conf. Cyber IT Serv. Manag.*, 2016.

[13] B. Li and L. Han, "Distance Weighted Cosine Similarity Measure for Text Classification," pp. 611–612, 2013.

[14] R. Ludviani and K. Hayati, "Optimasi Pembobotan pada Query Expansion dengan Term Relatedness to Query-Entropy based (TRQE)," *J. Buana …*, vol. 6, no. 3, pp. 203–212, 2015.

[15] D. Domarco, N. Made, and S. Iswari, "Rancang Bangun Aplikasi Chatbot Sebagai Media Pencarian Informasi Anime Menggunakan Regular Expression Pattern Matching," *Ultimatics*, vol. IX, no. 1, pp. 19–24, 2017.