

IMPLEMENTASI CHATBOT PADA PENERIMAAN MAHASISWA BARU DI POLITEKNIK TEDC BANDUNG MENGGUNAKAN NATURAL LANGUAGE PROCESSING

Hani Rahmawati^{1*}, Ari Sudrajat²

^{1,2} Program Studi Teknik Informatika, Politeknik TEDC Bandung; Jl. Politeknik - Pesantren Km 2 Cibabat, Cimahi Utara; (022) 6645951

Received: 12 Oktober 2024
Accepted: 14 Januari 2025
Published: 20 Januari 2025

Keywords:

Chatbot;
PMB;
Natural Language Processing;
Neural Network.

Correspondent Email:

hanirahmawati8910@gmail.com

Abstrak. Chatbot adalah program komputer yang mampu melakukan percakapan seperti halnya manusia. Model percakapan menggunakan kecerdasan buatan untuk memahami pertanyaan pengguna dan memberikan respons yang sesuai dengan topik yang dibahas. Proses penerimaan mahasiswa baru memerlukan layanan informasi yang cepat dan efisien, website penerimaan mahasiswa baru di Politeknik TEDC Bandung masih sebatas berisi informasi umum. Penelitian ini bertujuan untuk membuat suatu aplikasi chatbot yang dapat menjawab pertanyaan dari calon mahasiswa seputar penerimaan mahasiswa baru di Politeknik TEDC Bandung. Aplikasi chatbot diimplementasikan menggunakan metode Natural Language Processing (NLP). Tahapan pada NLP terdiri dari pembuatan dataset, text preprocessing, bag of word (BoW), dan model neural network. pelatihan pada model dilakukan sebanyak 450 epoch dan mendapatkan hasil dengan final loss 36,85% dan accuracy sebesar 90,77%. Hasil pengujian User Acceptance Test (UAT) pada Aplikasi Chatbot Penerimaan Mahasiswa Baru di Politeknik TEDC Bandung terbukti dapat diterima dengan baik oleh pengguna, dengan nilai persentase mencapai 89,78%.

Abstract. A chatbot is a computer programme that is capable of having human-like conversations. The conversation model utilizes artificial intelligence to understand user questions and provide appropriate responses to the topics discussed. The new student admission process requires fast and efficient information services, the new student admission website at TEDC Polytechnic Bandung is still limited to containing general information. This research aims to create a chatbot application that can answer questions from prospective students regarding new student admissions at the TEDC Polytechnic Bandung. The chatbot application is implemented using the Natural Language Processing (NLP) method. Stages in NLP include dataset creation, text preprocessing, bag of word (BoW), and model. training on the model was carried out as many as 450 epochs and obtained results with a final loss of 36.85% and accuracy of 90.77%. The results of User Acceptance Test (UAT) testing on the New Student Admission Chatbot Application at TEDC Polytechnic Bandung proved to be well received by users, with a percentage value reaching 89.78%.

1. PENDAHULUAN

Penerimaan Mahasiswa Baru (PMB) merupakan salah satu agenda rutin yang

dilakukan semua perguruan tinggi di Indonesia. Informasi mengenai PMB menjadi hal yang penting bagi lembaga pendidikan maupun calon

mahasiswa. Lembaga Pendidikan harus menyajikan informasi PMB dengan jelas dan transparan. Komunikasi yang baik dan akses mudah terhadap informasi dapat membantu memastikan bahwa calon mahasiswa dapat mengambil keputusan yang tepat dan siap mengikuti proses penerimaan dengan baik.

Saat ini Politeknik TEDC Bandung menggunakan *website* sebagai *platform* penyedia informasi mengenai penerimaan mahasiswa baru, namun pada *website* tersebut masih belum adanya komunikasi dua arah antara calon mahasiswa dan petugas PMB. Calon mahasiswa membutuhkan komunikasi dua arah untuk menanyakan informasi yang lebih detail. Salah satu layanan yang tersedia untuk menjawab pertanyaan dari calon mahasiswa adalah layanan menggunakan *whatsapp*. Namun layanan tersebut belum dapat dilakukan selama 24 jam karena keterbatasan waktu kerja sehingga pertanyaan yang disampaikan tidak dapat dijawab dengan instan.

Pesatnya perkembangan teknologi membantu layanan informasi yang pada umumnya dikerjakan oleh manusia kini dapat dikerjakan oleh program komputer, salah satunya yaitu *chatbot*. *Chatbot* adalah program komputer yang mampu melakukan percakapan layaknya manusia [1]. *Chatbot* dapat beroperasi selama 24 jam tanpa pengawasan manusia, *chatbot* berguna untuk membantu merespon pertanyaan yang diajukan oleh calon mahasiswa dengan instan dan cepat.

Chatbot dapat diimplementasikan menggunakan metode *Natural Language Processing* (NLP). NLP merupakan salah satu cabang dari kecerdasan buatan yang berfokus pada pengolahan bahasa alami atau bahasa yang secara umum digunakan oleh manusia untuk berkomunikasi satu sama lain [2]. NLP memungkinkan *chatbot* untuk memahami pesan dari pengguna dan memberikan respons dengan cepat dan tepat. Agar *chatbot* menjadi lebih komunikatif serta memahami dan memberikan respons layaknya manusia dapat menggunakan algoritma [3].

Berdasarkan permasalahan yang telah diuraikan, penulis akan menerapkan metode *Natural Language Processing* untuk membangun aplikasi *chatbot* penerimaan mahasiswa baru di Politeknik TEDC Bandung. Penelitian ini bertujuan untuk membangun suatu aplikasi *chatbot* yang mampu menjawab

pertanyaan dari calon mahasiswa seputar penerimaan mahasiswa baru. Penelitian ini diharapkan dapat memberikan kemudahan kepada calon mahasiswa dalam mencari informasi dan mempermudah pihak PMB dalam menyampaikan informasi mengenai penerimaan mahasiswa baru.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Terdapat beberapa penelitian-penelitian sebelumnya yang terkait sebagai referensi dalam penelitian ini, antara lain :

- a. Penelitian yang berjudul “Aplikasi *Chatbot* Untuk Penerimaan Mahasiswa Baru Universitas Indo Global Mandiri Menggunakan *Deep Learning*”. Penelitian ini bertujuan untuk membangun aplikasi *chatbot* yang dapat memberikan informasi mengenai penerimaan mahasiswa baru di Indo Global Mandiri kepada pengguna. *chatbot* diimplementasikan menggunakan bahasa pemrograman *python* dengan *deep learning* [2].
- b. Penelitian yang berjudul “Implementasi *Natural Language Processing* Pada *Chatbot* Untuk Layanan Akademik”. Penelitian ini bertujuan untuk mempermudah proses layanan informasi pada bidang akademik. Pada penelitian ini diterapkan metode *Natural Language Processing* (NLP) untuk membangun sistem *chatbot* layanan akademik berbasis web. Algoritma yang digunakan dalam membangun *chatbot* pada penelitian ini adalah [3].
- c. Penelitian yang berjudul “Penerapan *Natural Language Processing* Pada Aplikasi *Chatbot* Info Layanan Kantor Menggunakan *Naïve Bayes Algorithm*”. Penelitian ini bertujuan untuk membangun sebuah aplikasi *chatbot* info layanan kantor dengan penerapan *Natural Language Processing* dengan menggunakan *Naïve Bayes Algorithm* [4].
- d. Penelitian yang berjudul “Implementasi Algoritma *Artificial* Dalam Pembuatan *Chatbot* Menggunakan Pendekatan *Natural Language Processing*”. Penelitian ini bertujuan untuk membangun sebuah aplikasi *chatbot* bernama “Peraturanku” yang memuat berbagai aturan perundang-undangan yang ada di Indonesia, sehingga memudahkan masyarakat untuk mengetahui

hukum dan peraturan apa saja yang ada di Indonesia. *Chatbot* dibangun menggunakan salah satu algoritma *Machine Learning* yaitu algoritma (ANN) serta menggunakan modul *TensorFlow* untuk melakukan *training model* [5].

2.2. Landasan Teori

a. Chatbot

Chatbot adalah program komputer yang melakukan interaksi antara manusia dan mesin menggunakan metode audio atau pesan [6]. *Chatbot* bekerja dengan cara pengguna mengajukan pertanyaan atau memulai topik diskusi baru. Berbagai teknologi yang dapat digunakan untuk implementasi *chatbot* yaitu *Machine Learning*, *Artificial Intelligence*, *Deep learning*, dan *Natural Language Processing*. *Natural Language Processing* (NLP) memiliki kemampuan untuk mengerti dan memahami bahasa manusia lalu memberikan respon yang sesuai dengan bahasa yang digunakan pengguna *chatbot*.

b. Natural Language Processing

Bahasa alami adalah bahasa yang biasa digunakan untuk berkomunikasi antar manusia, misalnya bahasa Indonesia, Inggris, Jepang dan sebagainya. *Natural Language Processing* (NLP) atau pengolahan bahasa alami merupakan cabang dari kecerdasan buatan yang berhubungan dengan pemrosesan bahasa alami dan interpretasi komputer [7].

Natural Language Processing (NLP) memiliki kemampuan untuk menginterpretasikan, memproses, memahami dan mengartikan bahasa manusia [8][9]. NLP bertujuan untuk merancang dan membangun aplikasi yang memfasilitasi interaksi manusia dengan mesin dan perangkat lain melalui penggunaan bahasa alami.

c. Neural Network

Neural Network adalah jaringan untuk meniru fungsi otak manusia dan sering disebut dengan jaringan syaraf tiruan. *Neural Network* merupakan bagian dari algoritma *Machine Learning* yang dapat mempelajari pola data secara mandiri, serta mampu memberikan respon berdasarkan input melalui sebuah proses sehingga dapat menghasilkan output yang sesuai [3]. arsitektur *neural network*

menggunakan tiga jenis layer yang saling terhubung yaitu :

1. Input Layer

Input Layer merupakan lapisan yang membawa data masuk kedalam sistem untuk kemudian di proses pada layer selanjutnya.

2. Hidden Layer

Hidden Layer merupakan lapisan yang berada diantara *input layer* dan *output layer* yang terdiri dari sekumpulan neuron penerima data dari *input layer*.

3. Output Layer

Output layer merupakan lapisan yang menghasilkan output akhir dari jaringan *neural network*. Jumlah neuron pada output layer ini harus disesuaikan dengan jumlah output yang diinginkan oleh sistem.

Tahap training pada *Neural Network* terdiri dari 2 tahap yaitu *Forward Pass* dan *Backward Pass*. *Forward pass* atau biasa juga disebut *forward propagation* adalah proses dimana kita membawa data pada *input* melewati tiap *neuron* pada *hidden layer* sampai pada *output layer* yang nanti akan dihitung nilai errornya [10].

3. METODE PENELITIAN

Pada penelitian ini penulis melakukan metode penelitian dengan teknik pengumpulan data dengan melakukan observasi, wawancara, dan studi literatur. Adapun perancangan sistem pada penelitian ini menggunakan SDLC (*Software Development Life Cycle*) dengan model *waterfall*. Tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut:

3.1. Analisis Kebutuhan

Pada tahapan ini penulis mengumpulkan data mengenai hal apa saja yang dibutuhkan dalam membangun *chatbot*, melalui observasi dan wawancara pada pihak terkait, serta mencari studi literatur yang berkaitan dengan topik penelitian ini.

3.2. Perancangan

Pada tahapan ini dilakukan perancangan sistem untuk menentukan kebutuhan-kebutuhan sistem, menerapkan metode *natural language processing* serta mendefinisikan arsitektur sistem secara keseluruhan.

3.3. Implementasi

Pada tahapan ini peneliti menerapkan hasil dari rancangan yang telah dibuat, meliputi pelatihan model *neural network* dan melakukan pengkodean terhadap sistem *chatbot*

menggunakan bahasa pemrograman python dan framework flask.

3.4. Pengujian

Pada tahapan ini penulis melakukan pengujian terhadap *unit-unit* program untuk memastikan apakah sistem sudah berjalan dengan baik sesuai dengan kebutuhan perangkat lunak atau tidak. Pengujian dilakukan menggunakan metode black box testing dan *User Acceptance Test (UAT)*.

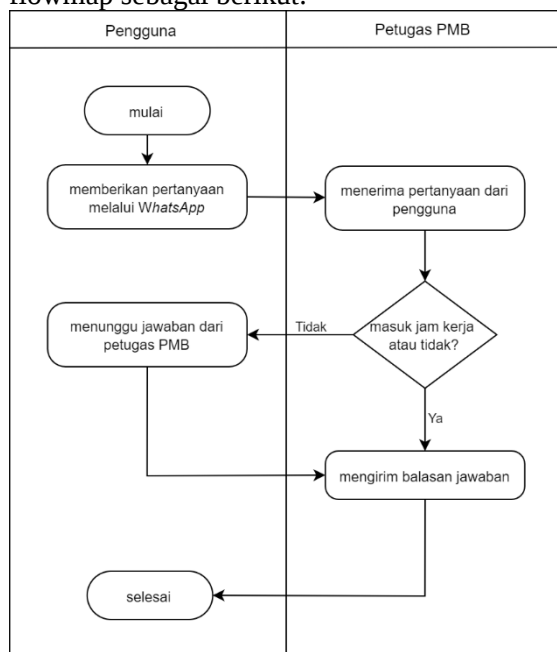
4. HASIL DAN PEMBAHASAN

4.1. Analisis Kebutuhan

4.1.1. Analisis Sistem

a. Analisis Sistem Yang Sedang Berjalan

Berdasarkan hasil observasi dan wawancara yang telah dilakukan, alur sistem yang sedang berjalan saat ini dapat digambarkan melalui flowmap sebagai berikut.

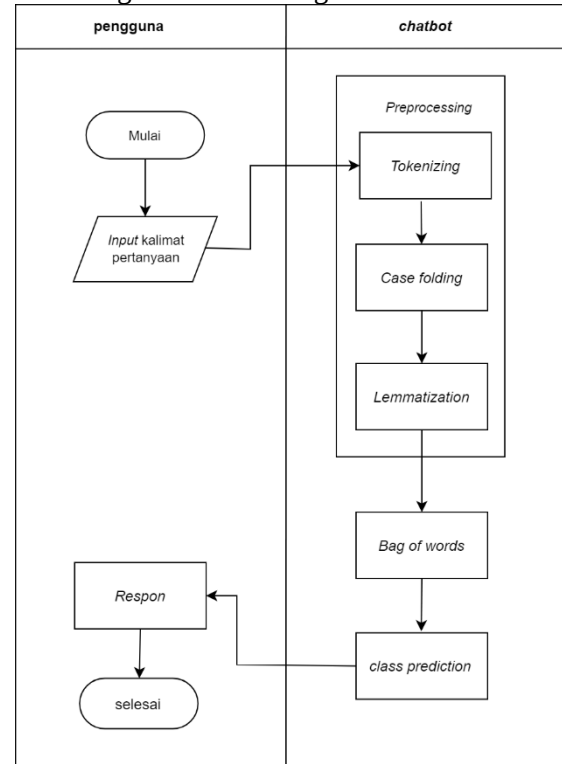


Gambar 1. Flowmap sistem yang sedang berjalan

Pada gambar diatas menunjukkan bahwa pengguna dapat mengajukan pertanyaan melalui whatsapp, kemudian petugas akan langsung memberikan balasan jawaban jika masuk kedalam jam kerja, jika tidak masuk kedalam jam kerja, maka calon mahasiswa harus menunggu jawaban dari petugas. Dari alur sistem yang sedang berjalan saat ini dapat disimpulkan bahwa keterbatasan jam kerja dalam layanan menggunakan *WhatsApp* mengakibatkan respon yang lambat terhadap pertanyaan calon mahasiswa.

b. Analisis Sistem Yang Akan Dikembangkan

Berikut merupakan desain sistem yang akan dirancang untuk membangun *chatbot*.



Gambar 2. Flowmap sistem yang akan dibangun

Pada tahap pertama, terdapat *input* yang dimasukkan oleh pengguna berupa pertanyaan, kemudian kalimat pertanyaan tersebut akan memasuki tahap *preprocessing* dan *bag of words*, kemudian program akan memprediksi kelas mana yang sesuai dengan kalimat pertanyaan berdasarkan model yang telah dilatih sebelumnya. Hasilnya program akan memberikan respon sesuai dengan kalimat yang ditanyakan oleh pengguna.

4.1.2. Analisis Kebutuhan Sistem

a. Kebutuhan Perangkat Keras

Perangkat keras atau hardware yang penulis gunakan untuk membangun aplikasi *chatbot* ini antara lain:

Tabel 1. Kebutuhan perangkat keras

No	Perangkat Keras	Keterangan
1.	Laptop	HP Laptop 14s-fq0xxx
2.	Processor	AMD Ryzen 3 3250U with Radeon Graphics

3.	RAM (<i>Random Acces Memory</i>)	8192 MB/ 8 GB
4.	Penyimpanan data berupa SSD (<i>Solid State Drive</i>)	512 GB

b. Kebutuhan Perangkat Lunak

Adapun kebutuhan perangkat lunak (software requirement) yang digunakan dalam membangun aplikasi *chatbot* ini antara lain:

Tabel 2. Kebutuhan perangkat lunak

No	Perangkat Lunak	Keterangan
1.	Sistem Operasi	Windows 11 Home Single Language 64-bit
2.	Text Editor	Microsoft Visual Studio Code v1.91.1
3.	Penjelajah Web	Microsoft Edge v127.0.2651.74
4.	Environment Manager	Anaconda v4.12.0
4.	Framework	Flask v3.0.3

c. Kebutuhan Pengguna

Aplikasi *chatbot* ini memiliki dua pengguna, yaitu *admin* dan *user*. Pada tabel 3 dibawah akan dijelaskan mengenai kebutuhan setiap pengguna:

Tabel 3. Kebutuhan pengguna

No	Kebutuhan	Pengguna
1.	Melakukan login	Admin
2.	Melihat halaman dashboard admin	Admin
3.	Melihat halaman chatbot	Admin dan User
4.	Menambah, mengubah dan menghapus data chatbot	Admin
5.	Memasukkan pertanyaan pada chatbot	User
6.	Menerima jawaban dari chatbot	User

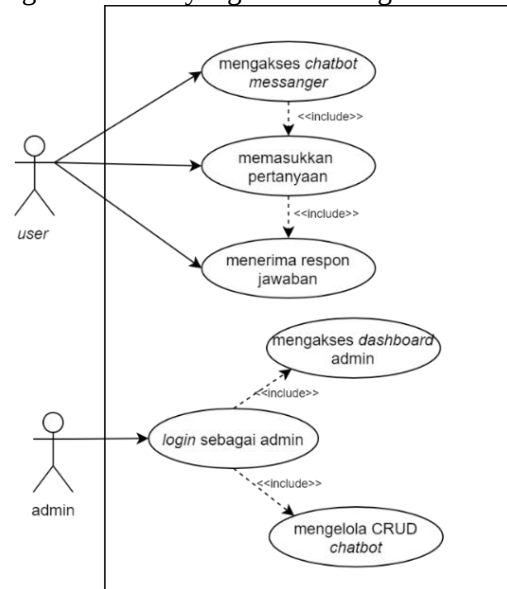
4.2. Perancangan

4.2.1. Perancangan Sistem

a. Use Case Diagram

Use case diagram adalah diagram yang menggambarkan interaksi antara pengguna

(*user*) sebuah sistem dengan suatu sistem tersendiri melalui skenario atau cerita yang menjelaskan bagaimana sistem tersebut digunakan [11]. Berikut merupakan *use case diagram* yang akan dibangun.

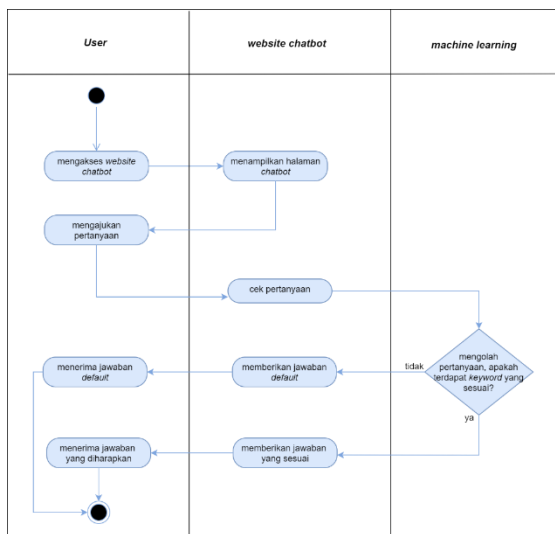


Gambar 3 Use case diagram

Pada *use case diagram* diatas terdapat dua aktor, yaitu *user* dan *admin*. *user* dapat mengakses *chatbot messenger* dan mengajukan kalimat pertanyaan seputar penerimaan mahasiswa baru. *Chatbot* akan memproses pertanyaan yang telah dimasukkan oleh *user*, kemudian *user* dapat menerima respon berupa jawaban yang sesuai dengan pertanyaan yang diajukan. Sedangkan *admin* dapat melakukan login sebagai *admin*, kemudian mengakses halaman dashboard dan mengelola CRUD (*create*, *update*, dan *delete*) pada data *chatbot*.

b. Activity Diagram

Activity diagram menggambarkan sejumlah aliran aktivitas di dalam sistem yang sedang direncanakan [11].

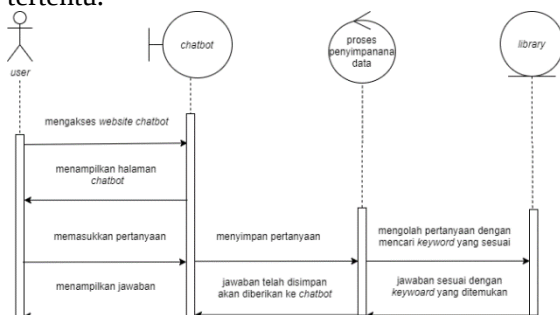


Gambar 4. Activity diagram

Pada *activity diagram* diatas menunjukkan, pengguna dapat mengakses *website chatbot*, kemudian *chatbot* akan menampilkan halaman *chatbot*. Selanjutnya *user* atau calon mahasiswa dapat mengajukan pertanyaan kepada *chatbot*, yang kemudian pertanyaan akan dicek terlebih dahulu oleh *Machine Learning* untuk mencari *keywords* yang sesuai dan hasil jawaban yang ditemukan akan diarahkan kembali ke *website chatbot* untuk ditampilkan kepada pengguna.

c. Sequence Diagram

Sequence diagram merupakan gambaran interaksi antara objek-objek didalam dan disekitar sistem yang diatur berdasarkan waktu tertentu.



Gambar 5. Sequence diagram

Keterangan:

- Langkah pertama, *user* mengakses *website chatbot*.
- Langkah kedua, *chatbot* akan menampilkan halaman *chatbot*.
- Langkah ketiga, *user* memasukkan pertanyaan yang ingin ditanyakan.

- Langkah keempat, *chatbot* akan menyimpan pertanyaan tersebut kedalam proses penyimpanan data.
- Langkah kelima, *library machine learning* akan mengolah pertanyaan tersebut untuk menemukan *keyword* yang sesuai.
- Langkah keenam, *library machine learning* akan menemukan jawaban sesuai dengan *keywords* yang terdapat pada pertanyaan yang ditanyakan oleh pengguna.
- Langkah ketujuh, jawaban akan dibawa ke proses menyimpan data yang kemudian akan diberikan ke *chatbot*.
- Langkah kedelapan, *chatbot* menampilkan jawaban yang sesuai atau jawaban *default* jika tidak menemukan *keywords* yang sesuai.
- Terakhir, *user* (pengguna) menerima jawaban dari *chatbot*.

4.2.2. Perancangan Antarmuka

Berikut merupakan perancangan antarmuka aplikasi *chatbot* yang akan dibangun.

a. Tampilan Halaman Beranda



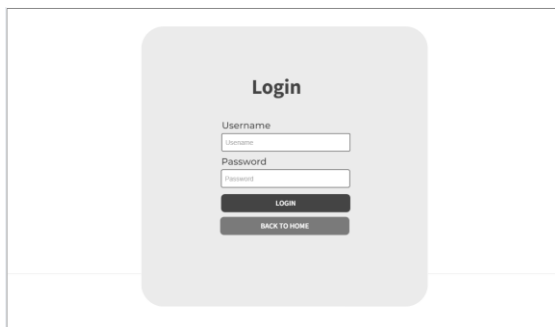
Gambar 6. Tampilan halaman beranda

b. Tampilan Pesan Chatbot

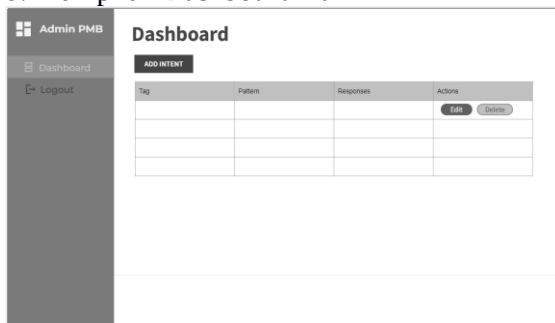


Gambar 7. Tampilan pesan chatbot

c. Tampilan Login Admin



Gambar 8. Tampilan login admin

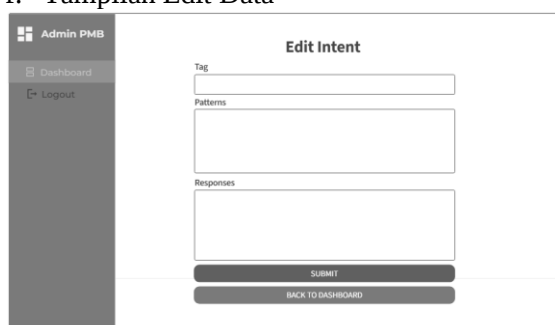
d. Tampilan *Dashboard Admin*Gambar 9. Tampilan *dashboard* admin

e. Tampilan Tambah Data



Gambar 10. Tampilan tambah data

f. Tampilan Edit Data



Gambar 11. Tampilan edit data

4.2.3. Metode Yang Digunakan

Pada penelitian ini penulis menggunakan metode Natural Language Processing (NLP) dalam pengembangan *chatbot*. Tahapan NLP

yang penulis gunakan pada penelitian ini meliputi pembuatan dataset, text preprocessing, bag of words, dan model *neural network*.

a. Dataset

Dataset dibuat secara manual dalam bentuk file .JSON. File ini berisi pertanyaan-pertanyaan yang diperlukan agar sistem dapat membaca setiap pola pertanyaan yang timbul beserta target jawabannya. Pemisahan data ini dibagi menjadi 3 kolom yaitu:

- Tag (label), merupakan sebuah kategori yang menjadi acuan sistem dalam menentukan respons.
- Patterns (pola), merupakan suatu rangkaian huruf yang diharapkan cocok dengan satu atau lebih dari input pengguna.
- Responses (output), yaitu berupa hasil jawaban yang akan keluar berdasarkan index tag dan patterns yang ditentukan oleh sistem.

b. Text Preprocessing

Text preprocessing merupakan salah satu tahapan dalam metode *Natural Language Processing* yang bertujuan untuk membersihkan unsur – unsur yang tidak diperlukan dalam kalimat pertanyaan agar lebih mudah diolah. Tahapan pada *text preprocessing* meliputi; *tokenizing*, *case folding*, dan *lemmatization*.

1. Tokenizing

Tokenizing merupakan proses memecah teks menjadi unit-unit kata atau token [12]. Alur tokenizing dapat dilihat pada gambar 6 dibawah ini.



Gambar 12. Alur tokenizing

Keterangan:

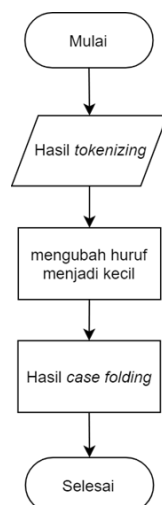
- Mulai
- Pengguna memasukkan kalimat yang ingin ditanyakan pada program.
- Kalimat akan diubah menjadi kata menggunakan fungsi *word.tokenize* yang terdapat pada *library* NLTK.
- *Output* dari proses ini adalah mengubah kalimat menjadi kata.
- Selesai.

Tabel 4 Contoh hasil tokenizing

Sebelum tokenizing	"Apa saja berkas pendaftaran PMB Politeknik TEDC Bandung"
Sesudah tokenizing	"Apa", "saja", "berkas", "pendaftaran", "PMB", "Politeknik", "TEDC", "Bandung"

2. Case Folding

Case folding merupakan proses dimana semua huruf diubah menjadi huruf kecil (*lowercase*). Berikut merupakan alur case folding.



Gambar 13. Alur case folding

Keterangan:

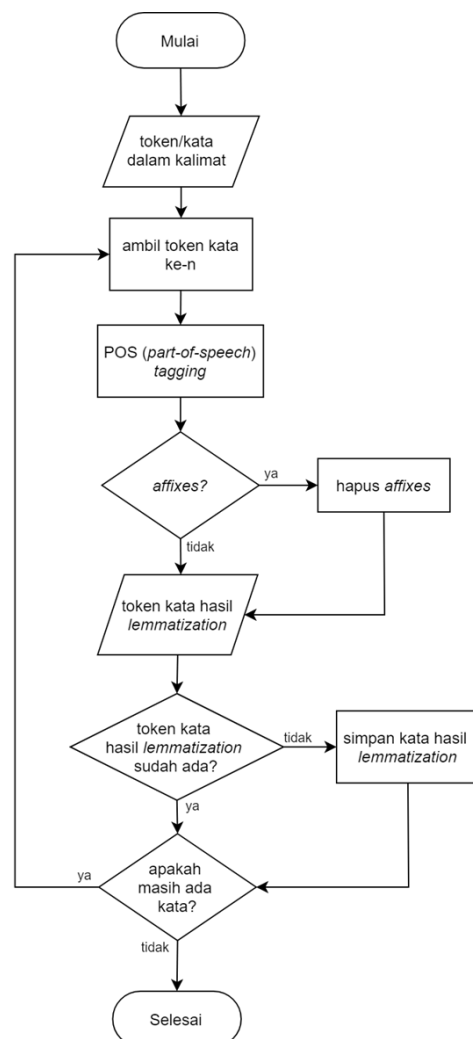
- Mulai
- *Output* pada proses *tokenizing* dijadikan *input* pada proses *case folding*
- Program akan menjalankan fungsi *.lower()* untuk mengubah karakter menjadi huruf kecil
- Hasil dari proses ini adalah mengubah semua huruf besar menjadi kecil
- Selesai.

Tabel 5. Contoh hasil case folding

Sebelum case folding	"Apa", "saja", "berkas", "pendaftaran", "PMB", "Politeknik", "TEDC", "Bandung"
Sesudah case folding	"apa", "saja", "berkas", "pendaftaran", "pmb", "politeknik", "tedc", "bandung"

3. Lemmatization

Lemmatization atau lematisasi adalah proses mengubah kata menjadi bentuk dasarnya. lematisasi mempertimbangkan konteks dan mengubah kata menjadi bentuk dasar yang sesuai secara gramatikal. Pada penelitian ini proses lematisasi menggunakan *library* NLTK dengan fungsi *WordNetLemmatizer*.



Gambar 14. Alur lemmatization

Keterangan:

- Mulai
- *Output* pada proses *tokenizing* dan *case folding* dijadikan *input* pada proses *lemmatization*
- Program akan mengambil kata demi kata pada kalimat masukkan
- Program akan menentukan kelas kata (*POS tagging*) pada token (misal : kata benda, kata sifat, kata kerja, dll) sesuai dengan konteks gramatikal pada kalimat tersebut
- Program akan menghapus imbuhan pada token kata yang mengandung imbuhan
- Program akan menyimpan token/kata hasil *lemmatization*, jika menemukan token/kata yang sama program hanya akan menyimpan 1 kata saja
- Program akan kembali pada proses *POS tagging* sampai semua kata terambil
- Hasil dari proses ini adalah mengembalikan kata ke bentuk dasar atau “lemma”-nya
- Selesai.

Tabel 6. Contoh hasil lemmatization

Sebelum lemmatization	“apa”, “saja”, “berkas”, “pendaftaran”, “pmb”, “politeknik”, “tedc”, “bandung”
Sesudah lemmatization	“apa”, “saja”, “berkas”, “daftar”, “pmb”, “politeknik”, “tedc”, “bandung”

c. Bag Of Words (BoW)

Bag of Words bertujuan untuk mengubah data teks menjadi vektor yang dapat dipahami oleh komputer. Proses ini dimulai dengan menyimpan data training kedalam sistem, yang kemudian akan diberi nilai 1 jika kata tersebut terdapat pada kantung kata dan nilai 0 jika tidak ada dalam kantung kata [3]. Berikut merupakan alur *bag of words* pada penelitian ini.



Gambar 15. Alur bag of words

Keterangan:

- Mulai.
- *Output* pada proses *lemmatization* digunakan untuk proses *bag of words* sebagai *input*.
- selanjutnya setiap kata akan diberikan nilai 1 jika muncul pada kantung kata dan 0 jika tidak muncul. Kata tersebut akan diubah menjadi matriks berbentuk *array*. Data akan dibagi menjadi dua yaitu x_{train} dan y_{train} .
- Selesai

Tabel 7. Contoh hasil bag of words

Kata	Bag of Words vector				
	apa	saja	berkas	daftar	pmb
apa	1	0	0	0	0
saja	0	1	0	0	0
berkas	0	0	1	0	0
daftar	0	0	0	1	0
pmb	0	0	0	0	1

d. Model Neural Network

Pada penelitian ini, pembuatan model menggunakan algoritma *neural network*. Untuk membuat model *neural network*, dibutuhkan jumlah layer dan activation function. Berikut merupakan arsitektur *neural network* yang digunakan untuk membangun *chatbot*.

Tabel 8. Arsitektur model *neural network*

Layer	Jumlah node	Activation function
Input layer	128	ReLU
Hidden layer	64	ReLU
Output layer	Hasil dari y_{train}	Softmax

Model dibuat menggunakan 3 lapisan (layer) yang terdiri dari input layer, hidden layer, dan output layer. Pada penelitian ini menggunakan jenis klasifikasi feedforward dan backpropagation untuk koreksi nilai output pada proses training.

4.3. Implementasi

Pada tahap ini merupakan implementasi dari perancangan sistem dan metode natural language processing yang telah dibuat.

4.3.1. Training Model

Berikut merupakan implementasi model *neural network* kedalam bentuk code yang dapat dilihat pada gambar dibawah.

```
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))

# compile model
sgd = SGD(learning_rate=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])

#training and save model
hist = model.fit(np.array(train_x), np.array(train_y), epochs=450, batch_size=5, verbose=1)
model.save('model.h5', hist)
print("model created")
```

Gambar 16. Potongan kode training model

Setelah pembuatan model selesai, untuk melihat tingkat akurasi kita harus melakukan eksekusi terhadap *code* tersebut. Proses pelatihan pada *chatbot* dilakukan sebanyak 450 *epoch* dan mendapatkan hasil dengan *final loss* 0.3685 dan *accuracy* sebesar 0.9077. Hasil pelatihan model dapat dilihat pada gambar 17 dibawah ini.

```
...
Epoch 449/450
53/53 0s 3ms/step - accuracy: 0.8973 - loss: 0.3385
Epoch 450/450
53/53 0s 3ms/step - accuracy: 0.9077 - loss: 0.3685
```

Gambar 17. Hasil training model

Setelah pelatihan, model yang telah dilatih disimpan dalam *file* model.h5 yang nantinya akan digunakan untuk memprediksi *intent* atau kelas yang relevan pada aplikasi *chatbot* yang akan dibangun.

4.3.2. Prediksi Model

Selanjutnya adalah membuat prediksi model yang berisikan jawaban berdasarkan pertanyaan dari pengguna. Proses mendapatkan hasil jawaban dari *chatbot* yaitu dengan mencari nilai terbesar dari setiap kata yang masuk dan dicocokkan dengan tag pada intent. Berikut merupakan potongan kode program respon *chatbot*.

```
def predict_class(sentence, model):
    p = bow(sentence, words, show_details=False)
    res = model.predict(np.array([p]))[0]
    ERROR_THRESHOLD = 0.25
    results = [(i,r) for i,r in enumerate(res) if r>ERROR_THRESHOLD]
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append({"intent": classes[r[0]], "probability": str(r[1])})
    return return_list

def getResponse(ints, intents_json):
    tag = ints[0]['intent']
    list_of_intents = intents_json['intents']
    for i in list_of_intents:
        if(i['tag']== tag):
            result = random.choice(i['responses'])
            break
    return result
```

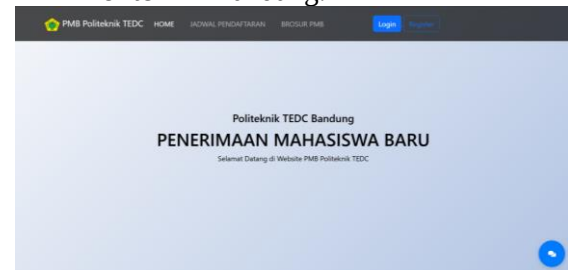
Gambar 18. Potongan kode prediksi model

4.3.3. Implementasi Antarmuka

Tahap ini bertujuan untuk menampilkan seluruh tampilan sistem yang telah dibuat sehingga mendapatkan hasil yang maksimal.

a. Halaman Beranda

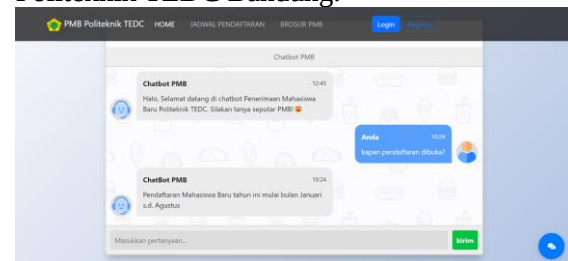
Halaman beranda menampilkan informasi mengenai *website* PMB Politeknik TEDC Bandung. Berikut contoh halaman pada *website* PMB Politeknik Bandung.



Gambar 19. Halaman beranda

b. Chatbot Messenger

Chatbot Messenger pada *website* PMB Politeknik TEDC Bandung digunakan sebagai sarana untuk menjembatani pengguna dengan *bot* agar dapat berinteraksi dengan mudah. Tugas dari *chatbot* adalah untuk menjawab pertanyaan dari pengguna seputar PMB Politeknik TEDC Bandung.



Gambar 20. Chatbot messenger

c. Login Admin

Pada halaman beranda *website* PMB Politeknik TEDC Bandung terdapat menu *login* yang dapat diakses oleh . dapat langsung masuk menggunakan *username* dan *password* yang

telah tersedia. Berikut merupakan tampilan *login*.

Gambar 21. Login admin

d. Dashboard Admin

Pada halaman *dashboard admin* menampilkan daftar data yang digunakan pada *chatbot* dan memiliki aksi untuk menambah, memperbaharui, dan menghapus data.

Gambar 22. Dashboard admin

e. Tambah Data

Pada halaman *tambah data*, dapat memasukkan *tag*, *patterns*, dan *responses* untuk menambahkan data pada *chatbot*

Gambar 23. Tambah data

f. Edit Data

Pada halaman *edit data*, mempunyai akses untuk mengubah data sesuai dengan kebutuhan. Terdapat 3 form *input* yaitu *tag*, *pattern*, dan *responses* dan menampilkan data sesuai dengan data yang dipilih.

Gambar 24. Edit data

4.4. Pengujian

Tahap pengujian merupakan tahap yang bertujuan untuk mengetahui hasil proses dari sistem yang telah dibangun dan untuk mengetahui kesalahan yang ada pada sistem.

4.4.1. Blackbox Testing

Pada pengujian sistem ini menggunakan metode *blackbox* dimana pengujian berfokus pada fungsionalitas sistem dan menguji kesesuaian *input* dan *output* pada sistem. Berikut merupakan hasil pengujian serta evaluasi dengan metode *blackbox*.

a. Halaman beranda

Tabel 9. Skenario halaman beranda

Skenario Pengujian	Yang Diharapkan	Hasil
Halaman beranda	Saat membuka aplikasi, menampilkan halaman beranda	Sesuai

b. Halaman Chatbot Messenger

Tabel 10. Uji chatbot messenger

Skenario Pengujian	Yang Diharapkan	Hasil
Flying Button Chat	Menampilkan <i>pop up chatbot mesanger</i> saat di klik .	Sesuai
Chatbot messenger	Memberikan <i>auto respon</i> dari masukan pertanyaan	Sesuai

c. Halaman Login Admin

Tabel 11. Skenario halaman login admin

Skenario Pengujian	Yang Diharapkan	Hasil
Halaman Login	Menampilkan halaman <i>login</i>	Sesuai
Halaman Login jika gagal	Menampilkan halaman <i>login</i>	Sesuai

d. Halaman *Dashboard Admin*Tabel 12. Skenario *dashboard admin*

Skenario Pengujian	Yang Diharapkan	Hasil
Halaman <i>Dashboard</i>	Saat <i>login</i> berhasil, menampilkan halaman <i>dashboard</i>	Sesuai
Tombol biru (<i>create data</i>)	Saat tombol di klik mengalihkan ke halaman <i>create data</i>	Sesuai
Tombol kuning (<i>edit data</i>)	Saat tombol di klik mengalihkan ke halaman <i>edit data</i>	Sesuai
Tombol merah (<i>delete data</i>)	Menghapus data yang dipilih	Sesuai

e. Halaman *Create Data*Tabel 13. Skenario halaman *create data*

Skenario Pengujian	Yang Diharapkan	Hasil
Halaman <i>Create Data</i>	Menampilkan halaman <i>create data</i>	Sesuai
Tombol submit	Menyimpan penambahan data	Sesuai
Tombol kembali	Kembali ke halaman <i>dashboard</i>	Sesuai

f. Halaman *Edit Data*Tabel 14. Skenario halaman *edit data*

Skenario Pengujian	Yang Diharapkan	Hasil
Halaman <i>edit data</i>	Menampilkan halaman <i>edit data</i>	Sesuai
Tombol submit	Menyimpan perubahan data	Sesuai
Tombol kembali	Kembali ke halaman <i>dashboard</i>	Sesuai

g. *Logout Admin*Tabel 15. Uji *logout admin*

Skenario Pengujian	Yang Diharapkan	Hasil
Menu <i>Logout</i>	Menutup sesi masuk dan mengalihkan ke halaman <i>login</i>	Sesuai

4.4.2. *User Acceptance Test*

Pengujian *User Acceptance Test* (UAT) bertujuan untuk menguji sistem yang telah dibangun sesuai dengan kebutuhan yang diharapkan. Pengujian ini dilakukan oleh 10 orang responden termasuk diantaranya adalah bagian *staff* PMB Politeknik TEDC Bandung, dengan teknis pengujian responden menguji secara langsung aplikasi *chatbot* penerimaan mahasiswa baru dan mengisi kuisioner yang telah dibuat oleh penulis.

Penilaian aspek uji dikelompokkan menjadi 3 (tiga) kelompok utama yaitu desain, fitur dan kepuasan pengguna. Berdasarkan pengujian dari responden diperoleh hasil rekapitulasi skor dan hasil perhitungan pada pengujian UAT sebagai berikut.

Penilaian		Desain			Fitur			Kepuasan			Jumlah (P1-P9)	Total Nilai
Huruf	Angka	P1	P2	P3	P4	P5	P6	P7	P8	P9		
SB	5	8	6	6	4	4	5	5	5	6	49	245
B	4	2	4	4	5	6	3	3	5	4	36	144
C	3				1		2	2			5	15
K	2											
SK	1											
Total Pengujian		140			130			134			90	404

Gambar 25. Hasil perhitungan pengujian UAT

Keterangan :

Jumlah responden = 10 orang
 Skor tertinggi = $5 \times (9 \times 10) = 450$
 Skor terendah = $1 \times (9 \times 10) = 90$

Tabel 16. Range persentase kepuasan pengguna

Kategori	Range
Sangat Kurang	0% - 20%
Kurang	21% - 40%
Cukup	41% - 60%
Baik	61% - 80%
Sangat Baik	81% - 100%

Persentase hasil pengujian = $(404 / 450) \times 100\% = 89,78\%$

Berdasarkan perhitungan yang diperoleh, dapat disimpulkan bahwa hasil pengujian *User Acceptance Test* (UAT) termasuk kedalam kategori sangat baik dengan persentase sebesar 89,78% dari 10 responden.

5. KESIMPULAN

Berdasarkan hasil pembahasan dan pengujian pada Aplikasi *Chatbot* Penerimaan Mahasiswa Baru di Politeknik TEDC Bandung dapat disimpulkan sebagai berikut.

- Aplikasi *Chatbot* Penerimaan Mahasiswa Baru di Politeknik TEDC Bandung mampu menjawab pertanyaan-pertanyaan pengguna

- terkait penerimaan mahasiswa baru di Politeknik TEDC Bandung. Ini menunjukkan bahwa *chatbot* dapat berfungsi sebagai alat yang efektif untuk memberikan informasi dan membantu calon mahasiswa dengan respon yang cepat dan relevan.
- b. Aplikasi *Chatbot* Penerimaan Mahasiswa Baru di Politeknik TEDC Bandung dibangun dengan mengimplementasikan metode *Natural Language Processing* (NLP). Pembuatan model pada *chatbot* menggunakan algoritma , model dilatih dengan nilai *epoch* sebanyak 450 *epoch* dan mendapat hasil akurasi sebesar 90.77% dan *error* sebesar 36.85%.
 - c. Hasil pengujian UAT pada Aplikasi *Chatbot* Penerimaan Mahasiswa Baru di Politeknik TEDC Bandung terbukti dapat diterima dengan baik oleh pengguna, dengan nilai persentase mencapai 89,78% dan pengujian *blackbox* yang dilakukan oleh penulis pada sistem ini dapat berjalan sesuai yang diharapkan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada dosen pembimbing yang telah memberikan arahan, Politeknik TEDC Bandung sebagai tempat penelitian, serta semua pihak yang terlibat dalam pembuatan penelitian ini.

DAFTAR PUSTAKA

- [1] T. A. Zuraiyah, D. K. Utami, and D. Herlambang, "Implementasi Chatbot Pada Pendaftaran Mahasiswa Baru Menggunakan Recurrent *Neural Network*," *J. Ilm. Teknol. dan Rekayasa*, vol. 24, no. 2, pp. 91–101, 2019, doi: 10.35760/tr.2019.v24i2.2388.
- [2] D. Heryati, Zulkifli, and R. M. Fajri, "Aplikasi Chatbot Untuk Penerimaan Mahasiswa Baru Universitas Indo Global Mandiri Menggunakan Deep Learning," 2023. doi: 10.36982/jinig.v1i1.3073.
- [3] A. Hikmah, F. Azmi, and R. A. Nugrahaeni, "Implementasi Natural Language Processing Pada Chatbot Untuk Layanan Akademik," *e-Proceeding of Engineering*, vol. 10, no. 1, pp. 371–382, Feb. 2023.
- [4] K. D. Ningtyas, R. Kurniawan, and Armansyah, "Penerapan Natural Language Processing Pada Aplikasi Chatbot Info Layanan Kantor Menggunakan Naïve Baiyes Algorithm," *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD*, vol. 6, no 1, pp. 266–273, 2023, doi: <https://doi.org/10.53513/jsk.v6i1.7413>.
- [5] N. A. Purwitasari and M. Soleh, "Implementasi Algoritma Artificial *Neural Network* Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Parocessing," *J. IPTEK*, vol. 6, no. 1, pp. 14–21, 2022, doi: 10.31543/jii.v6i1.192.
- [6] I. S. Nugroho, A. Voutama, "Implementasi Chat Bot Untuk Pelayanan Pelanggan Yang Terintegrasi Web Toko Komputer," *JATI (jurnal Mhs. Tek. Inform.,* vol. 8, no. 3, pp. 3132–3136, 2024, doi: <https://doi.org/10.36040/jati.v8i3.9630>.
- [7] M. Singla, "Advance in Computer Science", *AkiNik Publications*, vol.7, 2020. doi: <https://doi.org/10.22271/ed.book.784>.
- [8] I. Amirulloh *et al.*, "Rancang Bangun Chatbot Whatsapp Menggunakan Node Js dan Model Natural Languange Processing Untuk Layanan PPDB SMK YPC Tasikmalaya," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 1, pp. 541–550, 2024, doi: <http://dx.doi.org/10.23960/jitet.v12i1.3846>.
- [9] Y. Aprilinda, T. Martavia, F. N. Afandi, and U. Rizal, "Chatbot Menggunakan Natural Language Processing untuk Pembelajaran Bahasa Inggris Berbasis Android," *Jurnal Manajemen Sistem Informasi dan Teknologi.*, vol. 8, no. 200, pp. 8–12, 2022. doi: <http://jurnal.ubl.ac.id/index.php/expert/article/view/2629>
- [10] M. F. Fadli, G. A. Buntoro, and F. Masykur, "Penerapan Algoritma *Neural Network* Pada Chatbot Pmb Universitas Muhammadiyah Ponorogo Berbasis Web," *J. Sist. dan Teknol. Inf. Komun.*, vol. 6, no. 1, pp. 13–22. doi: <https://doi.org/10.32524/jusitik.v6i1.786>.
- [11] T. B. Kurniawan and Syarifuddin, "Perancangan Sistem Aplikasi Pemesanan Makanan Dan Minuman Pada Cafeteria No Caffé Di Tanjung Balai Karimun Menggunakan Bahasa Pemograman PHP Dan Mysql," *Jurnal TIKAR*, vol. 1, no. 2, pp. 192–206, 2020.
- [12] A. N. Oktavia, M. Iqbal, R. W. Saputra, and M. Ichsan, "Implementasi Metode Natural Language Processing Dalam Studi Analisis Semantik Dan Emosi Buzzer Pada Tweet Di Aplikasi X," *BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia*, vol. 2, no. 1, pp. 154–159, 2024, doi: <https://jurnalmahasiswa.com/index.php/biikma/article/view/1093>.