



## UMMIBOT SEBAGAI MEDIA LAYANAN INFORMASI PENERIMAAN MAHASISWA BARU UNIVERSITAS MUHAMMADIYAH SUKABUMI

M. Rizky Suherlan<sup>1</sup>, Asriyanik<sup>2</sup>, Agung Pambudi<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, Universitas Muhammadiyah Sukabumi  
Sukabumi, Jawa Barat, Indonesia 43113

[rizkysuherlan07@gmail.com](mailto:rizkysuherlan07@gmail.com), [asriyanik263@ummi.ac.id](mailto:asriyanik263@ummi.ac.id), [agungpambd@ummi.ac.id](mailto:agungpambd@ummi.ac.id)

### Abstract

*The continuous advancement in information and communication technology has overcome geographical limitations and enhanced the speed of information and knowledge transfer. Information and communication technology has been applied in various aspects, including improving services, speeding up communication processes, facilitating access to information, and effectively sharing information. One significant technological innovation is the chatbot, an artificial intelligence technology that enables interactive communication. Implementing a chatbot at the Admissions Office of Muhammadiyah University Sukabumi (UMMI) aims to assist prospective students in obtaining information regarding new student admissions. The main challenge addressed is the limitation of office hours, which often hinders prospective applicants from accessing information quickly. This research employs the Natural Language Processing (NLP) method in chatbot development, intending to enable the chatbot to understand natural language and search for relevant information. The research objective is to create an efficient chatbot to respond to user inquiries outside of office hours. The stages of chatbot development using the NLP method encompass data collection, text processing, feature engineering, modeling, inference, deployment, and evaluation. The research results indicate that the chatbot, named UMMIBOT, successfully achieved a functional accuracy rate of 83.67%, addressing the issue of office hour limitations and providing responses that align with user input with high accuracy.*

**Keywords:** Chatbot, Machine Learning, Natural Language Processing, New Student Admissions, UMMIBOT

### Abstrak

Perkembangan teknologi informasi dan komunikasi yang terus maju telah mengatasi batasan jarak dan meningkatkan kecepatan transfer informasi dan pengetahuan. Teknologi informasi dan komunikasi telah diterapkan dalam berbagai aspek, termasuk meningkatkan pelayanan, mempercepat proses komunikasi, memudahkan akses informasi, dan efektif dalam berbagi informasi. Salah satu inovasi teknologi yang penting adalah chatbot, sebuah teknologi kecerdasan buatan yang memungkinkan komunikasi interaktif. Penerapan *chatbot* di PMB Universitas Muhammadiyah Sukabumi bertujuan untuk membantu calon mahasiswa dalam mendapatkan layanan informasi seputar pendaftaran mahasiswa baru. Kendala utama yang diatasi adalah keterbatasan jam kerja yang seringkali menghambat calon pendaftar dalam mendapatkan informasi dengan cepat. Penelitian ini menggunakan metode NLP (*Natural Language Processing*) dalam pengembangan *chatbot*, yang bertujuan untuk memungkinkan *chatbot* memahami bahasa alami dan mencari informasi yang diperlukan. Tujuan penelitian ini adalah menciptakan chatbot yang efektif dalam memberikan respon terhadap pertanyaan pengguna di luar jam kerja. Tahapan pengembangan *chatbot* dengan metode NLP meliputi *data collection*, *text processing*, *feature engineering*, *modeling*, *inference*, *deployment*, dan *evaluation*. Hasil penelitian menunjukkan bahwa *chatbot* yang dinamakan UMMIBOT berhasil mencapai tingkat akurasi fungsionalitas sebesar 83,67%, mengatasi masalah keterbatasan jam kerja, dan memberikan respon yang sesuai dengan input pengguna dengan tingkat akurasi yang tinggi.

**Kata kunci:** Chatbot, Pembelajaran Mesin, Pemrosesan Bahasa Alami, Penerimaan Mahasiswa Baru, UMMIBOT

### 1. PENDAHULUAN

Perkembangan pesat dalam Teknologi Informasi dan Komunikasi (TIK) telah mengatasi batasan geografis dan meningkatkan transfer informasi serta pengetahuan secara dramatis. Fenomena ini tercermin dalam adopsi luas oleh

masyarakat terhadap perangkat teknologi tinggi. Pemanfaatan TIK telah meresap ke berbagai sektor, termasuk dalam upaya meningkatkan pelayanan, mempercepat aliran informasi, memudahkan akses terhadap informasi, dan efektif dalam berbagi pengetahuan. Salah

satu tonggak penting dalam evolusi teknologi adalah *chatbot*, sebuah aplikasi kecerdasan buatan yang memungkinkan komunikasi interaktif. Terdapat beberapa pengertian tentang *chatbot*, yaitu teknologi kecerdasan buatan yang memungkinkan komunikasi interaktif [1]. *Chatbot*, yang juga dikenal sebagai *Chatterbot*, adalah suatu program yang memiliki kemampuan untuk memproses *input* yang diberikan oleh pengguna dan menghasilkan respon yang dikirim kembali ke pengguna [2]. *Chatbot* merupakan aplikasi yang berfungsi sebagai wadah percakapan (*chatting*) antara mesin dan manusia [3]. Dengan menerapkan NLP (*Natural language Processing*), *chatbot* memiliki kemampuan untuk memahami informasi dan secara cerdas mengumpulkan data spesifik hanya dengan menggunakan bahasa alami. NLP, yang merupakan bagian dari kecerdasan buatan, merupakan ilmu yang memfasilitasi interaksi antara mesin dan manusia melalui penggunaan bahasa alami. Dengan kata lain, NLP memungkinkan seseorang berinteraksi dengan mesin seolah-olah sedang berkomunikasi dengan manusia [4]. *Natural Language Processing* (NLP) adalah disiplin ilmu dalam bidang kecerdasan buatan yang memfasilitasi interaksi antara mesin dan manusia melalui bahasa alami [5].

Penelitian terdahulu telah mengeksplorasi penggunaan *chatbot* dalam berbagai konteks. Sebagai contoh, dalam sebuah penelitian yang dilakukan oleh Agung Siswanto Bayu Aji dengan judul "Membangun *Chatbot* Layanan *Helpdesk* Perpajakan KPP Pratama Jakarta Setiabudi Satu", *chatbot* ini telah diintegrasikan ke dalam aplikasi WhatsApp dan menggunakan algoritma *exact match* sebagai metode responnya. Algoritma *exact match* beroperasi dengan cara langsung mencocokkan input pengguna dengan pola yang telah ditentukan sebelumnya. Hasil dari penelitian ini menunjukkan bahwa *chatbot* mampu memberikan respon dengan cepat dan akurat ketika pengguna memasukkan pesan yang cocok dengan pola yang telah ada. Namun, jika pengguna memasukkan pesan yang tidak sesuai dengan pola yang ada, *chatbot* mungkin tidak memberikan respons atau merespon dengan pola yang sama berulang kali untuk mengarahkan pengguna agar memasukkan pesan sesuai dengan pola yang diminta [6]. Penelitian lain oleh Bestralaga Rusmarasy mencoba menerapkan *chatbot* dengan judul "Pengembangan *Chatbot* pada CoMa Untuk Memberikan Motivasi Kepada Pengguna Menggunakan AIML", *chatbot* dikembangkan menggunakan AIML (*Artificial Intelligence Markup Language*). AIML bekerja dengan prinsip *template matching*, di mana *chatbot* akan mencocokkan *input* dari pengguna dengan pola yang telah ditentukan sebelumnya. Penelitian ini menggunakan metode pengujian *blackbox* dan *whitebox*, yang menghasilkan tingkat akurasi sebesar 100% dan tingkat kegunaan (*usability*) sebesar 88,8%. Namun, terdapat kekurangan dalam pengembangan *chatbot* menggunakan AIML, yaitu jika terdapat *input* yang tidak sesuai dengan pola yang telah ditentukan, *chatbot* tidak dapat memberikan respon apapun

[2]. Dalam penelitian yang dilakukan oleh Eva Mursidah dengan judul "Implementasi *Chatbot* Layanan Informasi Pendaftaran Mahasiswa Baru Program Pascasarjana Departemen Teknik Informatika", *chatbot* digunakan sebagai media pendukung untuk memberikan layanan informasi pendaftaran program Pascasarjana Departemen Teknik Informatika di ITS. *Chatbot* ini dikembangkan dengan menggunakan metode *natural language processing* (NLP) dan alat bantu dialogflow yang diintegrasikan ke dalam platform WhatsApp. *Chatbot* yang dihasilkan mampu menerima *input* dari pengguna dengan berbagai cara, baik melalui penggunaan angka maupun mengetikkan kata kunci dari topik yang telah tersedia. Selain itu, *chatbot* juga dapat memberikan jawaban *default* jika *input* dari pengguna tidak sesuai dengan template pertanyaan yang telah ditentukan sebelumnya. Hasil pengujian menunjukkan tingkat akurasi sebesar 98,82% dari 85 kalimat pertanyaan terkait pendaftaran program Pascasarjana di ITS [7].

Di Universitas Muhammadiyah Sukabumi, *chatbot* dapat menjadi solusi untuk memberikan layanan informasi terkait penerimaan mahasiswa baru. Dimana terdapat kendala dalam hal keterbatasan waktu dalam memberikan informasi kepada calon mahasiswa baru di luar jam kerja. *Chatbot* yang dilengkapi dengan NLP dapat memberikan respons yang cepat dan akurat terhadap pertanyaan-pertanyaan ini. Dengan mengintegrasikan pengalaman dari penelitian sebelumnya, tujuan penelitian ini adalah untuk mengembangkan *chatbot* yang mampu mengatasi masalah ini dengan memberikan layanan informasi yang lebih baik dan responsif kepada calon mahasiswa baru di Universitas Muhammadiyah Sukabumi. Hasil penelitian ini diharapkan akan memberikan manfaat signifikan dalam memberikan informasi yang tepat waktu dan akurat kepada calon mahasiswa baru di Universitas Muhammadiyah Sukabumi. *Chatbot* ini akan menjadi alat yang efektif dalam memfasilitasi penerimaan mahasiswa baru dengan memberikan respon yang cepat dan relevan terhadap pertanyaan-pertanyaan mereka. Dengan demikian, penelitian ini akan membantu meningkatkan pengalaman calon mahasiswa baru dalam proses penerimaan, mengatasi kendala waktu yang seringkali menjadi hambatan dalam memperoleh informasi yang mereka butuhkan.

## 2. METODE PENELITIAN

### 2.1 Tahapan Pengumpulan Data

Untuk mengumpulkan data dalam penelitian ini, digunakan beberapa teknik yang telah dipilih, antara lain:

#### 2.1.1 Studi Pustaka

Dalam penelitian ini, digunakan metode studi pustaka yang melibatkan pencarian, pengumpulan, dan analisis literatur serta penelitian yang telah dilakukan sebelumnya yang berasal dari jurnal, buku dan internet. Metode studi pustaka ini bertujuan untuk memperoleh pemahaman yang

menyeluruh mengenai teori, konsep, dan temuan yang berkaitan dengan topik penelitian.

### 2.1.2 Wawancara

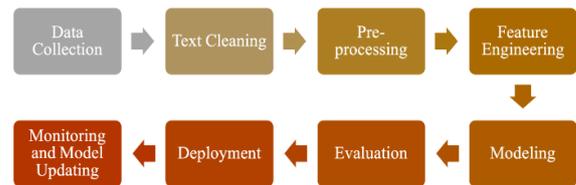
Pada tahap ini, dilakukan interaksi langsung dengan admin Hotline HPPMB UMMI yang melibatkan wawancara mendalam serta diskusi terkait permasalahan yang dihadapi serta solusi yang ditawarkan. Wawancara tersebut dilakukan untuk mendapatkan pemahaman yang lebih mendalam mengenai isu-isu yang relevan dengan pendaftaran mahasiswa baru di Universitas Muhammadiyah Sukabumi. Dalam proses wawancara, dilakukan pengumpulan data berupa pandangan, perspektif, dan pengalaman dari admin Hotline HPPMB UMMI. Informasi yang diperoleh dari wawancara ini menjadi sumber data yang sangat berharga dalam penelitian ini, karena dapat memberikan wawasan yang lebih komprehensif mengenai berbagai aspek yang terkait dengan proses pendaftaran mahasiswa baru. Selain itu, wawancara juga memberikan kesempatan untuk menggali informasi yang mungkin sulit ditemukan melalui sumber lain, sehingga dapat memberikan kontribusi yang signifikan dalam menemukan solusi yang efektif dan relevan dalam meningkatkan pelayanan informasi terkait pendaftaran mahasiswa baru di Universitas Muhammadiyah Sukabumi.

### 2.1.3 Observasi

Observasi dilakukan dengan tujuan untuk memperoleh pemahaman tentang situasi dan kondisi yang terjadi pada proses Penerimaan Mahasiswa Baru di Universitas Muhammadiyah Sukabumi. Pada tahap ini, observasi dilakukan untuk mengamati langsung dan memperoleh informasi mengenai berbagai aspek yang terkait dengan proses Penerimaan Mahasiswa Baru. Observasi dilakukan dengan tujuan untuk mendapatkan pemahaman tentang situasi aktual yang terjadi di lingkungan Universitas Muhammadiyah Sukabumi saat pelaksanaan penerimaan mahasiswa baru. Dalam observasi ini, berbagai aspek yang diamati meliputi proses pendaftaran, mekanisme seleksi, komunikasi antara pihak universitas dengan calon mahasiswa, serta keterlibatan berbagai pihak terkait dalam proses penerimaan. Observasi dilakukan dengan cermat dan sistematis untuk mendapatkan data dan informasi yang akurat mengenai situasi yang ada.

## 2.2 Tahapan Penelitian

Dalam pembuatan UMMIBOT metode pengembangan yang digunakan yaitu *Natural Language Processing Pipeline*. *Natural Language Processing Pipeline* merupakan serangkaian langkah berurutan yang digunakan untuk memproses dan menganalisis teks dalam konteks bahasa alami. *Pipeline* NLP terdiri dari beberapa tahapan utama, termasuk pra-pemrosesan, pemrosesan teks, dan analisis teks [8]. Berikut merupakan tahapan dari *NLP Pipeline* sebagaimana ditunjukkan pada Gambar 1 di bawah ini.



Gambar 1. Alur *Natural Language Processing Pipeline*

### 2.2.1 Data Collection

*Data Collection* merupakan proses yang melibatkan pengumpulan data teks yang akan digunakan dalam analisis NLP. Sumber data yang digunakan dapat bervariasi, seperti korpus teks, data pengguna, atau sumber data lain yang relevan dengan tujuan aplikasi NLP.

### 2.2.2 Text Processing

Untuk melakukan pemrosesan teks pada data yang telah diformat sebagai JSON, data tersebut akan dimuat dan dimasukkan ke dalam sebuah daftar yang disesuaikan berdasarkan *tag*. Pada tahap ini, langkah-langkah berikut dilakukan dalam memproses daftar *patterns* yang berisi pertanyaan:

- 1) Menghapus tanda baca dan mengubah huruf menjadi huruf kecil (*remove punctuation and case folding*), mengubah semua huruf kapital menjadi huruf kecil dan menghilangkan tanda baca yang tidak diperlukan [9].
- 2) Normalisasi kata (*word normalization*), kata-kata yang tidak baku atau tidak standar akan diubah menjadi bentuk yang standar menggunakan teknik normalisasi kata. Hal ini bertujuan untuk mencapai konsistensi dalam representasi kata-kata dalam teks [10].
- 3) Penghapusan *stopword* (*stopword removal*), kata-kata yang memiliki tingkat informasi yang rendah atau sering muncul dalam teks, seperti kata hubung atau kata penghubung, akan dihapus menggunakan *library* NLTK *stopword*. Tujuannya adalah untuk fokus pada kata-kata yang lebih relevan dalam analisis [9].
- 4) *Stemming*, kata-kata yang memiliki afiks atau imbuhan akan diubah menjadi bentuk dasarnya melalui proses *stemming* menggunakan *library* *sastrawi*. Dengan melakukan *stemming*, variasi bentuk kata dengan akar yang sama dapat dikurangi, sehingga memudahkan pemrosesan dan pemahaman teks [10].
- 5) Tokenisasi, data yang telah dibersihkan akan dipecah menjadi *token*, yaitu unit-unit kecil seperti kata atau frasa, menggunakan *tokenizer*. Setiap *token* akan diberi indeks dan kata yang sesuai dalam pembentukan kamus (*vocabulary*) [9].
- 6) *Sequence*, tahapan dimana teks diubah menjadi representasi numerik yang dapat dipahami oleh model *machine learning*. Setiap *token* dalam urutan tersebut mewakili kata atau unit penting dalam teks, dan urutan *token* tersebut membentuk informasi berurutan yang dapat dimanfaatkan oleh model [9].

7) *Padding*, sebelum data masuk ke lapisan *embedding*, dilakukan *padding* untuk memastikan panjang teks dalam urutan yang seragam. Hal ini penting agar data dapat diproses secara efisien dan konsisten oleh model jaringan saraf [9].

### 2.2.3 Feature Engineering

Persiapan data untuk *machine learning* seringkali melibatkan tugas utama yang disebut *Feature Engineering* [11]. Proses ini merupakan langkah dalam mengembangkan fitur-fitur yang relevan dari fitur-fitur yang ada, dengan tujuan untuk meningkatkan kinerja prediksi atau klasifikasi. *Feature engineering* merupakan proses ekstraksi fitur dari data mentah dan mengubahnya menjadi format yang sesuai dengan model *machine learning* [12].

### 2.2.4 Modeling

Pada tahap ini, dilakukan pelatihan model mesin yang telah di-vektorisasi menggunakan data latihan. Model akan mempelajari pola-pola yang ada dalam data latihan dan menyesuaikan parameter internalnya untuk meningkatkan performa. Dalam pengembangan *chatbot*, model yang digunakan adalah *Bidirectional Long Short-Term Memory*. Metode dalam *deep learning* yang dikenal sebagai *Bidirectional Long Short-Term Memory* (Bi-LSTM) terdiri dari dua *layer* LSTM yang beroperasi secara bertentangan. *Layer* pertama, yang disebut *forward layer*, memproses kata-kata dari awal hingga akhir. Sementara itu, *layer* kedua, yang disebut *backward layer*, memproses kata-kata dari akhir hingga awal. Dengan demikian, Bi-LSTM mampu memperoleh informasi kontekstual dari kedua arah, yang berguna dalam pemrosesan bahasa alami dan tugas-tugas lain dalam analisis teks[13].

Setelah proses pelatihan selesai, model siap digunakan untuk melakukan klasifikasi dan prediksi pada data baru. *Activation function* yang digunakan dalam model ini adalah *softmax*, yang menghasilkan probabilitas untuk setiap kategori atau *label*.

### 2.2.5 Inference

Tahap *Inference* melibatkan replikasi *chatbot* dari tahap menerima *input* hingga memberikan respon. Pada tahap ini, *chatbot* akan menganalisis *input* yang diterima dan menggunakan model atau algoritma yang telah dilatih sebelumnya untuk menghasilkan respon yang sesuai. Proses ini melibatkan pemrosesan *input*, pemrosesan bahasa alami, dan penerapan aturan atau logika yang telah ditentukan. Setelah *input* diproses dan informasi yang relevan diambil, *chatbot* akan menggunakan pengetahuan dan pemahaman yang dimilikinya untuk memberikan respon yang tepat. Respon yang dihasilkan dapat berupa teks, suara, atau visual, tergantung pada implementasi *chatbot* yang digunakan.

### 2.2.6 Deployment

Pada langkah ini, *chatbot* akan diintegrasikan ke dalam sebuah *website* dengan menggunakan kerangka kerja *Flask*. Proses ini melibatkan pengembangan dan konfigurasi *chatbot* agar dapat berinteraksi dengan pengguna melalui *website* yang telah dibuat menggunakan *Flask*. Kerangka kerja *Flask* memungkinkan pengembang untuk membuat titik akhir (*endpoint*) dan menghubungkan fungsi-fungsi *chatbot* dengan permintaan pengguna yang diterima melalui *website* tersebut [14].

### 2.2.7 Evaluation

Pada tahap ini, dilakukan evaluasi dan pengujian terhadap model untuk mengevaluasi kualitas dan performanya. Evaluasi dilakukan dengan menggunakan metode *blackbox testing* yang fokus pada pengujian fungsionalitas *chatbot*. *Blackbox testing* adalah suatu pendekatan dalam pengujian perangkat lunak yang berfokus pada pengujian fungsi-fungsi atau fitur-fitur aplikasi tanpa mempertimbangkan struktur *internal* atau cara kerja aplikasi tersebut. [15]. Evaluasi ini dilakukan oleh admin *hotline* UPT HPPMB melalui *website* untuk menguji kesesuaian *input* dan respons *chatbot*. *Input* yang diberikan berupa pertanyaan yang sering ditanyakan kepada admin *hotline*, termasuk proses pendaftaran, pembayaran, perpindahan jurusan, informasi mengenai *browsure*, dan beasiswa. Melalui evaluasi tersebut, dapat diidentifikasi kelebihan dan kekurangan *chatbot* dalam menjalankan tugasnya. Evaluasi secara menyeluruh memberikan wawasan yang berharga untuk meningkatkan kualitas dan performa *chatbot*.

## 3. HASIL DAN PEMBAHASAN

Penelitian ini menggambarkan pengembangan UMMIBOT, sebuah *chatbot* yang dibuat untuk membantu calon mahasiswa baru di Universitas Muhammadiyah Sukabumi dalam mendapatkan informasi seputar penerimaan mahasiswa baru. *Chatbot* ini dikembangkan dengan menggunakan metode Natural Language Processing (NLP) Pipeline, yang memungkinkan *chatbot* untuk memahami bahasa alami pengguna dan memberikan respons yang sesuai.

### 3.1. Data Collection

*Dataset* yang diperoleh melalui wawancara dengan admin *Hotline* UPT HPPMB UMMI serta pengumpulan data secara langsung melalui *website* PMB UMMI. Adapun data yang dikumpulkan yaitu informasi seputar Universitas Muhammadiyah Sukabumi dan riwayat pertanyaan dari calon mahasiswa pada admin *hotline* UPT HPPMB UMMI. Dalam proses pengumpulan data, diperoleh 101 kategori *dataset* dan 425 pertanyaan yang akan digunakan sebagai *dataset* dalam pembuatan UMMIBOT. Setelah memperoleh data, dilakukan analisis dan pemahaman terhadap data tersebut. Data yang telah diperoleh akan disusun dalam format JSON yang mencakup elemen-elemen seperti *intent*, *tag*, *patterns*, dan *response*. Elemen *tag* akan berfungsi

sebagai *label* atau kategori, *patterns* berisi daftar pertanyaan yang terkait dengan *tag* tersebut, dan *response* merupakan jawaban yang akan diberikan. Struktur JSON yang digunakan dapat ditemukan pada Gambar 2 seperti yang ditunjukkan di bawah ini.

```

1- {}
2- "intents": [
3-   {
4-     "tag": "salam",
5-     "patterns": [
6-       "Hai",
7-       "Hi",
8-       "Halo",
9-       "Apa Kabar",
10-      "Selamat Pagi!",
11-      "Selamat Siang!",
12-      "Selamat Malam!",
13-      "Salam",
14-      "Assalamualaikum wr.wb",
15-      "Hallo"
16-    ],
17-    "responses": [
18-      "Halo!",
19-      "Hai",
20-      "Halo, ada yang bisa saya bantu?",
21-      "Halo selamat datang",
22-      "Hai Kawan"
23-    ]
24-  },

```

Gambar 2. Implementasi Struktur JSON

### 3.2. Text Processing

*Dataset* yang terdapat dalam daftar *patterns* akan melalui proses pembersihan agar lebih terstruktur. Berikut merupakan tahapan dari *text processing*:

#### 3.2.1 Remove Punctuation and Case Folding

Pada tahap ini tanda baca akan dihilangkan dan mengganti semua huruf kapital menjadi huruf kecil. Untuk implementasi *remove punctuation and case folding* dapat dilihat pada Gambar 3 berikut ini.

Sebelum	Sesudah
Informasi tentang perpustakaan ummi?	informasi tentang perpustakaan ummi
apakah di ummi menerima beasiswa KIP?	apakah di ummi menerima beasiswa kip
Bagaimana cara melakukan pengunduran diri dari... PTI	bagaimana cara melakukan pengunduran diri dari... pti
Agribisnis ummi FE	agribisnis ummi fe
logo ummi	logo ummi
Bagaimana sistem kehadiran di ummi?	bagaimana sistem kehadiran di ummi
apa itu ILC?	apa itu ilc
jurusan yang tersedia?	jurusan yang tersedia

Gambar 3. Perbandingan Teks Sebelum dan Sesudah Proses *Remove Punctuation and Case Folding*

#### 3.2.2 Word Normalization

Pada tahap ini, kata-kata yang tidak standar akan diubah menjadi kata-kata standar atau baku. Contohnya, kata "abis" akan diubah menjadi "habis", dan kata "info" akan diubah menjadi "informasi". Implementasi dari *word normalization* bisa dilihat pada Gambar 4 berikut ini.

```

key_norm = pd.read_csv('key_norm.csv')

def text_normalize(text):
    text = ' '.join([key_norm[key_norm['singkat'] == word]['hasil'].values[0]
                    if (key_norm['singkat'] == word).any()
                    else word for word in text.split()])
    text = str.lower(text)
    return text

print(text_normalize('info'))

```

Gambar 4. Contoh Implementasi *Word Normalization*

#### 3.2.3 Stopword

Pada tahap ini kata yang tidak penting biasanya berupa kata hubung akan dihilangkan. Misalnya kata "ada", "dan", "di", dan lainnya. Implementasi *stopword* bisa dilihat pada Gambar 5 berikut ini.

	Sebelum	Sesudah
225	saya ingin bertanya saya ingin ganti pilihan j...	ganti pilihan jurusan
227	Saya dari kelas reguler ingin pindah ke non re...	kelas reguler pindah non reguler
106	biaya kuliah non reguler	biaya kuliah non reguler
183	visi misi upt humas dan pmb	visi misi upt humas pmb
328	tahapan pembayaran menggunakan bank Dana	tahapan pembayaran bank dana
324	cara pembayaran menggunakan bank Muamalat?	pembayaran bank muamalat
274	cara membuat SKL	skl
377	Administrasi Bisnis	administrasi bisnis
73	program pasca sarjana	program pasca sarjana
62	RPL	rpl

Gambar 5. Perbandingan Teks Sebelum dan Sesudah Proses *Stopword*

#### 3.2.4 Stemming

Di tahap ini kata berimbuhan akan diubah dalam bentuk dasarnya, misalnya kata "pindahan" menjadi "pindah". Implementasi *stemming* dapat dilihat pada Gambar 6 berikut ini.

	Sebelum	Sesudah
328	tahapan pembayaran menggunakan bank Dana	tahap bayar bank dana
331	Fakultas Keguruan dan Ilmu Pendidikan ummi	fakultas guru ilmu didik ummi
277	cara membuat surat keterangan pengganti ijazah	surat terang ganti ijazah
146	fasilitas yang ada di ummi	fasilitas ummi
209	Bagaimana cara pembayaran perkuliahan mengguna...	bayar kuliah metode transfer bank ummi
156	Apa saja tugas dari Lembaga al islam dan kemuh...	tugas lembaga al islam kemuhammadiyah
355	Fakultas Ekonomi	fakultas ekonomi
405	Fakultas Hukum ummi	fakultas hukum ummi
235	pindah dari ummi	pindah ummi
416	arti logo ummi	arti logo ummi

Gambar 6. Perbandingan Teks Sebelum dan Sesudah Proses *Stemming*

#### 3.2.5 Tokenizing and Fit on Text

Di tahap ini kalimat pada *dataset* akan dipecah menjadi *token*. Dan dijadikan kamus yang berisi kosa kata dalam *dataset*. Implementasi *tokenizing and fit on text* dapat dilihat pada Gambar 7 berikut ini.

```
{1: 'ummi',
 2: 'bayar',
 3: 'daftar',
 4: 'program',
 5: 'bank',
 6: 'fakultas',
 7: 'kuliah',
 8: 'pindah',
 9: 'beasiswa',
10: 'kampus',
11: 'studi',
12: 'tahap',
```

Gambar 7. DaftarToken yang Dihasilkan Setelah Proses *Tokenizing and Fit on Text*

### 3.2.6 Sequence

Token yang dipecah akan diubah menjadi *integer* dimana nilai *integer* akan bergantung pada urutan kata dalam kamus yang telah dibuat. Berikut contoh implementasi *sequence* sebagaimana ditunjukkan pada Gambar 8 berikut ini.

```
print("Sequences = ", train[4])
print("Artinya = ", ['daftar[61]', 'daftar[194]'])
```

Sequences = [61, 194]  
Artinya = [ selamat , pagi ]

Gambar 8. Pembentukan *Sequence* pada *Dataset*

### 3.2.7 Padding

Di tahap ini akan dilakukan proses penyamarataan panjang dari *sequence*. Berikut implementasi dari *padding* sebagaimana ditunjukkan pada Gambar 9 berikut ini.

```
print(x_train)
```

```
[[190  0  0 ...  0  0  0]
 [191  0  0 ...  0  0  0]
 [192  0  0 ...  0  0  0]
 ...
 [309  1  0 ...  0  0  0]
 [189  1  0 ...  0  0  0]
 [189 21  0 ...  0  0  0]]
```

Gambar 9. Contoh Implementasi *Padding* pada *Dataset*

## 3.3. Feature Engineering

Setelah data melalui tahapan *text processing*, selanjutnya data akan diubah menjadi vektor numerik berdimensi rendah dimana vektor-vektor tersebut memiliki hubungan semantik antar kata.

## 3.4. Modeling

Data yang sudah divektorisasi akan dimasukan dalam model untuk ditrain. Model yang digunakan dalam penelitian ini yaitu *Bidirectional Long Short-Term Memory*. Berikut merupakan struktur model yang digunakan sebagaimana ditunjukkan pada Gambar 10 berikut ini.

```
model.summary()
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 7, 10)	3100
bidirectional (Bidirectional)	(None, 20)	1680
flatten (Flatten)	(None, 20)	0
dense (Dense)	(None, 101)	2121

Total params: 6,901  
Trainable params: 6,901  
Non-trainable params: 0

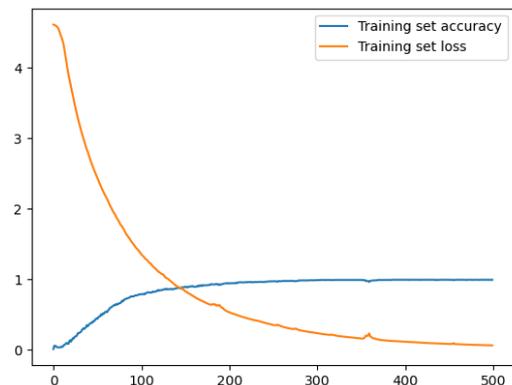
Gambar 10. Struktur Model yang Digunakan

Berikut merupakan hasil *training model* dari 500 iterasi dengan jumlah *learning rate* sebesar 0,01 yang ditunjukkan pada Gambar 11 berikut ini.

```
...
Epoch 499/500
14/14 [=====] - 0s 14ms/step - loss: 0.0624 - accuracy: 0.9906
Epoch 500/500
14/14 [=====] - 0s 18ms/step - loss: 0.0619 - accuracy: 0.9930
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Gambar 11. Hasil *Training Model*

Grafik dari hasil *training model* dapat dilihat pada Gambar 12 berikut ini.



Gambar 12. Grafik Hasil *Train*

## 3.5. Inference

Pada tahap ini akan dilakukan simulasi percakapan dengan model yang telah dibuat. Model akan menganalisis *input* dan memberikan respon. Implementasi *inference* bisa dilihat pada Gambar 13 berikut ini.

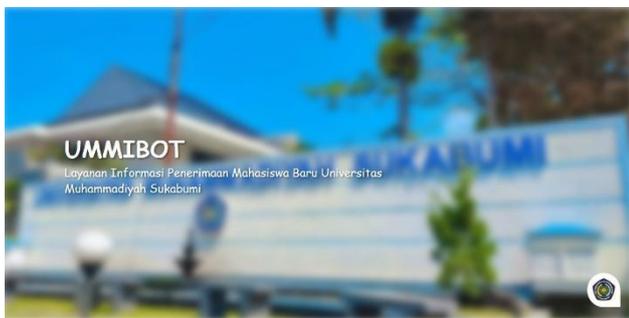
```

input : makasih
text_p ['terimakasih']
p1 [[121]]
p2 [121]
p3 [[121 0 0 0 0 0 0]]
1/1 [=====] - 0s 53ms/step
1/1 [=====] - 0s 49ms/step
out 88
bye
answer : Kalau ada masalah, hubungi aku lagi ya
    
```

Gambar 13. Inference

### 3.6. Deployment

Model yang telah dibuat akan diintegrasikan dalam *website* menggunakan *flask*. Adapun tampilan *website* dapat dilihat pada Gambar 14 yang menunjukkan tampilan awal *website* sebelum *bubble* di klik dan Gambar 15 yang menunjukkan tampilan *chatbot* setelah *bubble* di klik.



Gambar 14. Tampilan Awal Chatbot



Gambar 15. Tampilan Chatbox pada Chatbot

### 3.7. Evaluation

Pengujian menggunakan *blackbox testing* yang berfokus pada fungsionalitas dan kualitas respon yang dihasilkan. Berikut merupakan hasil pengujian dari *chatbot* sebagaimana ditunjukkan pada Tabel 1 berikut ini.

Tabel 1. Uji Fungsionalitas Chatbot

Komponen yang Diuji	Skenartion Pengujian	Keterangan
Input tidak ada dalam dataset	Menguji percakapan dimana <i>input user</i> tidak ada dalam <i>dataset</i> .	Sesuai
Input ada dalam dataset namun susunan acak	Menguji percakapan dimana <i>input user</i> berada dalam <i>dataset</i> tetapi dengan susunan kata yang acak	Sesuai

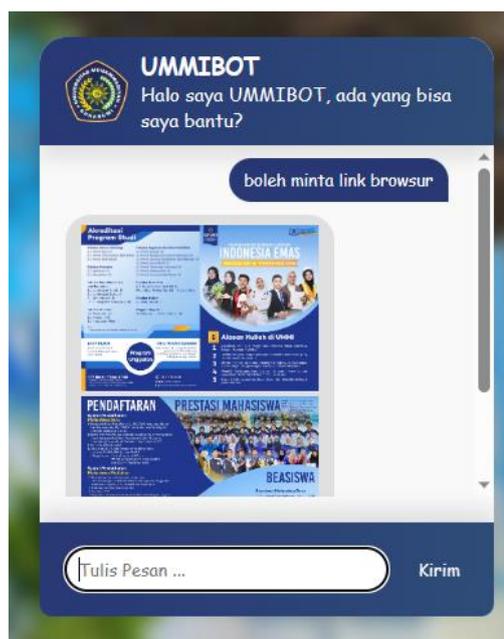
Komponen yang Diuji	Skenartion Pengujian	Keterangan
Input dengan dataset	Menguji percakapan dimana <i>input</i> sesuai dengan <i>dataset</i>	Sesuai
Input dengan makna ganda	Menguji percakapan dimana <i>input</i> mengandung dua makna yang berbeda	Tidak Sesuai

Pada Gambar 16 dapat dilihat bahwa jika *input* tidak tersedia atau terdeteksi dalam kamus maka respon yang diberikan adalah “maaf saya tidak mengerti”.



Gambar 16. Pengujian Input Tidak Ada dalam Dataset

Gambar 17 menunjukkan bahwa jika *user* memberikan *input* sesuai dengan *patterns* pada *dataset* maka respon yang diberikan akan sebagai berikut.



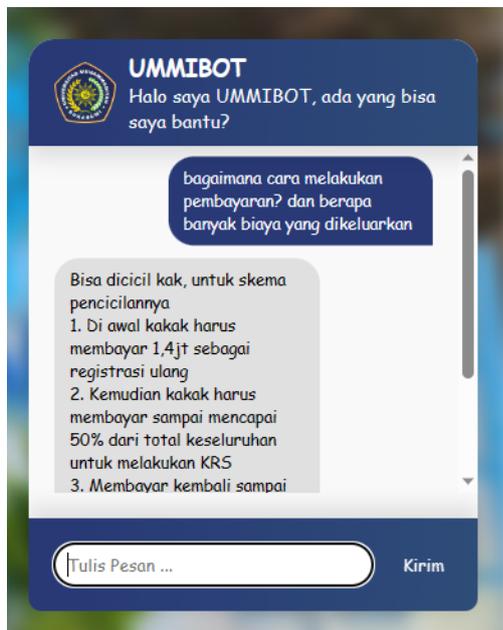
Gambar 17. Pengujian Input Tersedia dalam Dataset

Jika *user* memberikan *input* dengan susunan acak maka respon yang diberikan akan seperti pada Gambar 18 berikut ini.



Gambar 18. Pengujian *Chatbot* setelah Susunan *Input* Diacak

Gambar 19 dibawah ini menunjukkan interaksi dimana jika *input user* mengandung dua makna yang berbeda maka respon yang diberikan akan menjadi *miss* klasifikasi.



Gambar 19. *Input User* Mengandung Dua Makna yang Berbeda

Untuk memperoleh akurasi, *chatbot* akan diuji dengan memberikan pertanyaan dari kategori-kategori yang sering ditanyakan oleh calon mahasiswa kepada admin *Hotline UPT HPPMB UMMI*. Hasil pengujian dari setiap kategori pertanyaan yang ditunjukkan pada Tabel 2 berikut ini.

Tabel 2. Uji Akurasi Berdasarkan Kesesuaian Pertanyaan

No	Pertanyaan	Keterangan
1.	Bagaimana proses pendaftaran untuk kuliah di universitas ini?	Sesuai
2.	Apa persyaratan yang harus dipenuhi untuk mendaftar sebagai mahasiswa baru?	Sesuai
3.	Bagaimana proses pendaftaran kuliah di ummi?	Sesuai
4.	Bagaimana cara mendapatkan informasi lebih lanjut tentang proses pendaftaran kuliah di ummi?	Sesuai
5.	Apakah ada batas waktu pendaftaran untuk kuliah di ummi?	Sesuai
6.	Bagaimana cara melakukan pembayaran perkuliahan di UMMI?	Sesuai
7.	Apa saja metode pembayaran yang tersedia untuk pembayaran perkuliahan di UMMI?	Sesuai
8.	Apakah bisa melakukan pembayaran melalui bank BSI? Bagaimana caranya?	Sesuai
9.	Bagaimana cara pembayaran perkuliahan menggunakan metode transfer bank di UMMI?	Sesuai
10.	Apakah di UMMI menyediakan pembayaran angsuran untuk perkuliahan?	Sesuai
11.	Apa saja persyaratan untuk melakukan perpindahan prodi di UMMI?	Sesuai
12.	Bagaimana tahapan untuk melakukan perpindahan program studi?	Sesuai
13.	Selain pindah program studi apakah juga bisa pindah fakultas?	Sesuai
14.	Apakah ada batasan waktu atau semester tertentu untuk melakukan perpindahan prodi di UMMI?	Tidak Sesuai
15.	Syarat apa saja yang diperlukan agar bisa pindah kampus?	Sesuai
16.	Bagaimana cara mengajukan beasiswa di UMMI?	Sesuai
17.	Apa saja jenis beasiswa yang tersedia di UMMI?	Sesuai
18.	Bagaimana cara mengajukan Beasiswa KIP di UMMI dan apa persyaratannya?	Sesuai
19.	Apakah ada beasiswa khusus untuk mahasiswa yang berasal dari latar belakang ekonomi kurang mampu, seperti Beasiswa KIP (Kartu Indonesia Pintar)?	Tidak Sesuai
20.	Kalo dapet beasiswa dapet potongannya berapa?	Sesuai
21.	Kak kalau misalnya udah ditahap menunggu ujian masih bisa ganti prodi gak?	Tidak Sesuai
22.	Saya dapat nilai ujian terbesar di sekolah, apakah bisa mengajukan beasiswa?	Sesuai
23.	Bagaimana cara melakukan pembayaran lewat BCA	Sesuai
24.	Bagaimana cara melakukan pembayaran?	Sesuai
25.	Bagaimana cara melakukan pendaftaran?	Sesuai
26.	Penyetaraan jurusannya apa saja?	Tidak Sesuai
27.	Bagaimana cara melakukan transfer bank BNI?	Sesuai
28.	Kan di ummi ada beasiswa non akademik kriterianya apa saja?	Sesuai
29.	Kak pembayaran melalui bank BCA gimna caranya?	Sesuai
30.	Kalo beasiswa dapet potongan berapa?	Sesuai
31.	Untuk pendaftaran online gimana caranya ya kak?	Sesuai

No	Pertanyaan	Keterangan
32.	Apakah masih ada beasiswa untuk alumni muhammadiyah?	Sesuai
33.	Di ummi menerima beasiswa KIP gak? kalo menerima bagaimana langkah-langkahnya?	Sesuai
34.	Tahapan pendaftaran beasiswa KIP bagaimana?	Sesuai
35.	Kalo pembayaran lewat dana gimana caranya?	Tidak Sesuai
36.	Biaya pendaftaran untuk mahasiswa baru berapa ya kak?	Tidak Sesuai
37.	Biaya pendaftaran secara keseluruhan berapa?	Tidak Sesuai
38.	Saya ingin ganti pilihan jurusan saya, bisa ganti gak?	Sesuai
39.	Boleh saya tau biaya kuliah non reguler kak?	Sesuai
40.	Aku mau tanya seputar pendaftaram, gimana caranya?	Sesuai
41.	Bagaimana tahapan untuk perpindahan program studi?	Sesuai
42.	Mau tanya boleh? kalo uang semester 1 bisa dicicil atau harus langsung semua?	Sesuai
43.	Kalo kelas non reguler itu bagaimana?	Sesuai
44.	Kalo biaya pendidikan itu bayarnya bisa bertahap gak?	Sesuai
45.	Apakah sekarang tersedia beasiswa peringkat kelas?	Sesuai
46.	Boleh minta info rincian biaya persemetermya?	Sesuai
47.	Izin bertanya mengenai informasi penggunaan kip di ummi?	Tidak Sesuai
48.	Kalo di ummi bisa menggunakan KIP kuliah?	Sesuai
49.	Untuk biaya perkuliahan itu bagaimana?	Sesuai

Berdasarkan tabel di atas terdapat satu respon yang tidak sesuai. Hal ini dikarenakan ketika *input* diproses dan diprediksi terdapat 2 atau lebih tag yang memiliki data yang sama seperti *input*. Kata “batasan waktu” terdapat pada tag “jadwal pendaftaran” dan “perpindahan prodi” berada pada tag “perpindahan prodi dan fakultas” sehingga respon menjadi *miss predict*. Dengan demikian, akurasi dari *chatbot* dapat dihitung dengan menggunakan rumus:

$$\frac{\text{Jawaban Benar}}{\text{Total Pertanyaan}} \times 100\%$$

Pada tabel 2 di atas dapat dilihat pengujian respon *chatbot* terhadap inputan dari pengguna. Hasil yang didapatkan yaitu terdapat 8 respon yang tidak sesuai dari 49 pertanyaan. Sehingga dapat dihitung akurasi dari sistem yaitu sebesar 83,67 %.

#### 4. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan pada pembuatan *chatbot* sebagai media untuk memberikan layanan informasi terkait pendaftaran mahasiswa baru di Universitas Muhammadiyah Sukabumi dengan menggunakan *natural language processing* ditemukan bahwa *chatbot* yang telah dibuat dan diberi nama UMMIBOT berhasil mencapai tingkat akurasi

fungsionalitas sebesar 83,67%. Hal ini menunjukkan bahwa UMMIBOT mampu memberikan respons yang sesuai dengan input yang diberikan oleh pengguna dengan tingkat keakuratan yang tinggi. Namun, selama pengujian terdapat beberapa kelemahan yang perlu diperhatikan dalam UMMIBOT. Salah satu kelemahan yang teridentifikasi adalah ketidakmampuan untuk mengenali kalimat-kalimat dengan makna ganda. Misalnya, ketika pengguna memberikan input yang mengandung kata-kata dengan makna ganda, UMMIBOT tidak mampu membedakan konteks yang dimaksud oleh pengguna, akibatnya respons yang diberikan tidak sesuai dengan harapan pengguna. Secara keseluruhan, pengembangan UMMIBOT adalah langkah yang positif dalam memanfaatkan teknologi untuk meningkatkan pelayanan informasi di bidang penerimaan mahasiswa baru. Dengan perbaikan yang tepat, UMMIBOT memiliki potensi untuk menjadi alat yang sangat berguna dalam membantu calon mahasiswa baru dalam memahami proses pendaftaran dan mendapatkan informasi yang dibutuhkan dengan cepat dan akurat.

#### DAFTAR PUSTAKA

- [1] Z. N. Adesfiana, I. Astuti, and E. Enawaty, “Pengembangan Chatbot Berbasis Web Menggunakan Model Addie,” *J. Khatulistiwa Inform.*, vol. 10, no. 2, pp. 147–152, 2022, [Online]. Available: <https://ejournal.bsi.ac.id/ejurnal/index.php/khatulistiwa/article/view/14050>
- [2] B. Rusmarasy, B. Priyambadha, and F. Pradana, “Pengembangan Chat Bot pada CoMa untuk Memberikan Motivasi Kepada Pengguna Menggunakan AIML,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 5, pp. 4484–4490, 2019.
- [3] A. A. Chandra, V. Nathaniel, and F. R. Satura, “Pengembangan Chatbot Informasi Mahasiswa Berbasis Telegram dengan Metode Natural Language Processing,” vol. 3, no. 1, pp. 20–27, 2022.
- [4] H. E. Rosyadi, F. Amrullah, R. D. Marcus, and R. R. Affandi, “Rancang Bangun Chatbot Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP ( Natural Language Processing ),” *BRILIANT J. Ris. dan Konseptual*, vol. 5, no. 1, pp. 619–626, 2020.
- [5] S. Wijanarko, “Analisis Kesesuaian Komentar Mahasiswa Pada Sistem Akademi Online Angket Penilaian Dosen Menggunakan Supervised Model,” *J. Tek. Inform.*, vol. 8, no. 2, pp. 74–84, 2022, doi: 10.51998/jti.v8i2.503.
- [6] A. S. B. Aji, “Membangun Chatbot Layanan Helpdesk Perpajakan Kpp Pratama Jakarta Setiabudi Satu,” *Sebatik*, vol. 26, no. 1, pp. 194–201, 2022, doi: 10.46984/sebatik.v26i1.1916.
- [7] E. Mursidah, L. Ambarwati, A. Karima, K. Kunci, : Chatbot, and M. Baru, “Implementasi Chatbot Layanan Informasi Pendaftaran Mahasiswa Baru Program Pascasarjana Departemen Teknik Informatika Its Implementation of Chatbot Information Services for New Student Registration

- Postgraduate Program Its Information Engineering Departmenten,” *J. Ilm. NERO*, vol. 7, no. 1, p. 2022, 2022.
- [8] A. Chen, “NLP Pipeline,” *Python Notes For Linguistics*, 2020. <https://alvinntnu.github.io/python-notes/nlp/nlp-pipeline.html#a-general-nlp-pipeline> (accessed Jul. 03, 2023).
- [9] P. B. Wintoro, H. Hermawan, M. A. Muda, and Y. Mulyani, “Implementasi Long Short-Term Memory pada Chatbot Informasi Akademik Teknik Informatika Unila,” *Expert J. Manaj. Sist. Inf. dan Teknol.*, vol. 12, no. 1, p. 68, 2022, doi: 10.36448/expert.v12i1.2593.
- [10] A. R. W. Rapsanjani and E. Junianto, “Implementasi Probabilistic Neural Network Dan Word Embedding Untuk Analisis Sentimen Vaksin Sinovac,” *J. Responsif Ris. Sains dan Inform.*, vol. 3, no. 2, pp. 233–242, 2021, doi: 10.51977/jti.v3i2.588.
- [12] C. M. Sitorus, A. Rizal, and M. Jajuli, “Prediksi Risiko Perjalanan Transportasi Online Dari Data Telematik Menggunakan Algoritma Support Vector Machine,” *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 254–265, 2020, doi: 10.28932/jutisi.v6i2.2672.
- [13] E. Subowo, F. Adi Artanto, I. Putri, and W. Umaedi, “BLTSM untuk analisis sentimen berbasis aspek pada aplikasi belanja online dengan cicilan,” *J. Fasilkom*, vol. 12, no. 2, pp. 132–140, 2022.
- [14] R. Somya, “Perancangan Aplikasi Chatting Berbasis Web di PT. Pura Barutama Kudus menggunakan Socket.IO dan Framework Foundation,” *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 4, no. 1, pp. 8–15, 2018, doi: 10.23917/khif.v4i1.5979.
- [15] A. P. Putra, F. Andriyanto, K. Karisman, T. D. M. Harti, and W. P. Sari, “Pengujian Aplikasi Point of Sale Menggunakan Blackbox Testing,” *J. Bina Komput.*, vol. 2, no. 1, pp. 74–78, 2020, doi: 10.33557/binakomputer.v2i1.757.