

## **APLIKASI *CHATBOT* SEBAGAI LAYANAN *LIVE CHAT* UNTUK PENERIMAAN MAHASISWA BARU MENGGUNAKAN METODE *WORD STEMMING* DENGAN *REGULAR EXPRESSION PATTERN MATCHING***

**Muhamad Rizki<sup>1)</sup>, Ahmad Fitriansyah<sup>2)</sup>, Moh. Narji<sup>3)</sup>**

<sup>1,3</sup>Prodi Teknik Informatika, Fakultas Komputer, Universitas MH Thamrin

<sup>2</sup>Prodi Teknik Informatika, Fakultas Teknologi, ITB Swadharma Jakarta

Correspondence author: A. Fitriansyah, hafaskom@gmail.com, Jakarta, Indonesia

### **Abstract**

Artificial intelligence is a technology trend that raises rapidly. Many applications build with artificial intelligence, for example, a chatbot that can answer chat messages automatically so that chat services can operate 24 hours a day. Currently, the live chat service for new student admissions at LP3I is provided by operators who have working hours so they cannot operate 24 hours a day. The system development method was applied using the prototyping method with data collection techniques through interviews and observations of the running system. The results of data collection obtained the application system requirements desired by LP3I. The design of this chatbot application using a Question and Answering System-based Natural Language Processing (NLP) approach, the chatbot can learn to understand the context of sentences based on the existing knowledge base and provide answers in natural language by applying basic text processing algorithms such as word stemming, cosine similarity calculations to measure sentence similarity, and regular expression matching algorithms. Application testing carries out using the black box testing method. This chatbot application is web-based as the primary user interface, and Android and web-based applications as the admin interface.

**Keywords** : artificial intelligence, chatbot, natural language processing, word stemming, regular expression matching

### **Abstrak**

Kecerdasan buatan merupakan tren teknologi yang saat ini berkembang sangat pesat. Banyak aplikasi yang dikembangkan dengan kecerdasan buatan, salah satunya adalah chatbot yang dapat menjawab pesan chat secara otomatis, sehingga layanan chat dapat beroperasi selama 24 jam penuh. Saat ini layanan live chat penerimaan mahasiswa baru di LP3I masih dilayani oleh operator yang memiliki jam kerja tertentu sehingga tidak bisa beroperasi selama 24 jam. Metode pengembangan sistem yang diterapkan adalah metode prototyping dengan teknik pengumpulan data melalui wawancara dan pengamatan pada sistem yang berjalan. Hasil pengumpulan data diperoleh kebutuhan sistem aplikasi yang diinginkan oleh pihak LP3I. Perancangan aplikasi chatbot ini dikembangkan dengan pendekatan Natural Language Processing (NLP) berbasis Question and Answering System, chatbot dapat belajar memahami konteks kalimat berdasarkan knowledge base

yang ada dan memberikan jawaban dengan bahasa yang natural, dengan menerapkan algoritma pemrosesan teks dasar seperti word stemming, perhitungan cosine similarity untuk mengukur kemiripan kalimat, dan algoritma pencocokan regular expression matching. Pengujian aplikasi dilakukan dengan metode blackbox testing. Aplikasi chatbot ini berbasis web sebagai antarmuka utama user, dan aplikasi berbasis android dan web sebagai antarmuka admin.

**Kata Kunci** : kecerdasan buatan, layanan chat, *natural language processing*, *word stemming*, *regular expression matching*

## A. PENDAHULUAN

Sebelum berkembangnya era internet yang mempengaruhi pasar bisnis, perusahaan dan para pelaku bisnis biasanya menggunakan jasa *advertisement* yang kemudian diarahkan untuk menghubungi nomor telepon yang dicantumkan apabila calon pelanggan tertarik dan ingin mempelajari/membeli produk mereka. Saat ini semuanya sudah berubah, siapa pun yang tertarik dan ingin mempelajari sebuah produk/layanan dari penyedia jasa/produk dan memutuskan untuk membelinya atau tidak, biasanya mereka akan berkunjung terlebih dahulu ke *website* penyedia jasa/produk tersebut untuk melakukan pengecekan dan informasi lebih dalam mengenai produk/layanan yang mereka inginkan.

Pada era industri 4.0 seperti sekarang, konsumen menginginkan kemudahan dan kecepatan. Konsumen menginginkan kendala yang dihadapi terkait produk atau layanan segera ditangani dan diselesaikan. Agar dapat beradaptasi dengan hal itu, hampir semua bisnis saat ini berbasis *online* dan digital. Hampir semua perusahaan kecil hingga besar memiliki setidaknya satu *website* resmi agar mempermudah orang-orang mengenal perusahaan mereka. Pada setiap *website*, terutama *website* yang menjual sesuatu produk atau layanan, tidak jarang ditemui sebuah layanan *live chatting*. *Live chatting* merupakan sebuah aplikasi layanan yang memungkinkan pengunjung untuk berinteraksi dan berkomunikasi secara langsung dengan bagian penjualan dari

perusahaan. Namun dengan *live chatting* yang hanya mengandalkan satu atau beberapa operator secara langsung akan menghadapi kendala ketika terjadi lonjakan pengunjung dengan ribuan pertanyaan yang harus dijawab secepat mungkin, karena kecepatan dalam membalas pesan tersebut akan sangat berpengaruh terhadap penilaian *customer* pada kualitas layanan perusahaan.

*Live chat* dalam penerimaan mahasiswa baru memiliki peran yang sangat penting, karena dengan adanya *live chatting* calon mahasiswa dapat bertanya mengenai kampus dan dapat memahami lebih baik tentang jurusan dan aturan kampus yang dia minati. Namun sistem *live chat* untuk beberapa kampus yang ada saat ini masih terbatas, dan hanya *online* di jam kerja saja, sehingga bagi calon mahasiswa yang melakukan riset mengenai kampus tujuannya di hari libur atau ketika malam hari menjadi tidak terlayani dengan baik. Calon mahasiswa yang mengirim *chat* di kotak *live chatting* harus menunggu beberapa lama untuk mendapatkan balasan. Ada beberapa kampus yang memberikan nomor kontak *chat* WhatsApp untuk menyalahi hal tersebut, namun tetap saja hal ini masih kurang efektif karena harus dibalas manual oleh bagian Penerimaan Mahasiswa Baru (PMB).

Untuk itu, diperlukan sebuah solusi yang dapat mengatasi hal tersebut, salah satunya adalah dengan layanan *chatbot live chatting*. *Chatbot* merupakan salah satu dari pengembangan aplikasi dicabang keilmuan kecerdasan buatan (*Artificial Intelligence*), *chatbot* dapat melakukan interaksi dengan pelanggan lebih cepat, dapat melakukan

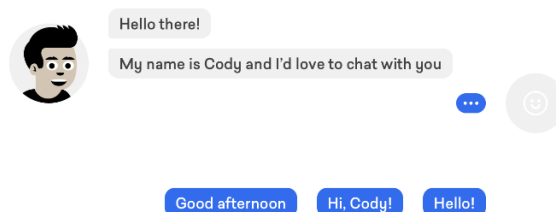
tugasnya 24 jam penuh tanpa istirahat, dan lebih memudahkan dalam hal bisnis.

Beberapa *chatbot live chatting* saat ini memiliki bahasa komunikasi yang formal dan kurang *natural*, dan memiliki banyak keterbatasan mengenai pengolahan *input* percakapan sehingga harus menjelaskan pertanyaan mereka dengan ejaan yang dibenarkan dalam bahasa Indonesia yang baku. Ada beberapa penelitian yang telah dilakukan untuk mencoba mengatasi hal ini, di antaranya dengan pendekatan *Natural Language Processing* menggunakan teknik dasar *tokenize*, *stopword*, *stemming*, *correction*, *analyzing* (Ma'ruf, 2019), *Chatbot* berbasis *machine learning* dengan metode *supervised learning* (Murhadi, 2019), *Chatbot Artificial Intelligence Markup Language* (Amalia & Wibowo, 2019), *Chatbot* dengan algoritma *bigram* dan metode *forward chaining* (Benedictus et al., 2017), *Chatbot* menggunakan *Regular Expression Pattern Matching* pada *knowledge base* (Domarco & Iswari, 2017). Dari beberapa metode penelitian yang telah dilakukan tersebut, masing-masing metode memiliki kelebihan dan kekurangannya sendiri, namun masih belum bisa menjawab persoalan sebelumnya yaitu kurang *naturalnya* bahasa balasan yang diberikan oleh *chatbot*, atau terbatasnya pemahaman *bot* untuk bisa memahami bahasa *input* yang kurang baku atau kalimat yang tidak sesuai dengan kaidah penulisan bahasa Indonesia menurut Ejaan Yang Dibenarkan (EYD).

Rupesh Singh menerangkan bahwa *chatbot* adalah sebuah program komputer yang dirancang untuk mensimulasikan percakapan antar manusia (Amalia & Wibowo, 2019). *Chatbot* dibekali dengan kecerdasan buatan dan pemrosesan bahasa alami yang membuat *chatbot* menjadi program komputer yang cerdas dan dapat menjawab pertanyaan yang diberikan oleh manusia. Proses *chatbot* dimulai dengan masukan dari pengguna menggunakan bahasa alami dan sistem akan menjawab dengan

respon yang masuk akal atau bisa dikatakan cerdas untuk bahasa yang sebenarnya. *Chatbot* diberikan kecerdasan buatan yang memiliki kemampuan untuk menirukan percakapan manusia yang sedang *chatting* atau mengobrol dengan suara bersama orang biasa. *Live support chat* merupakan sebuah layanan bantuan berupa obrolan antara pelanggan dengan sistem secara *real-time*, *live support chat* dapat memberikan jawaban mengenai pertanyaan, keluhan, dan kekhawatiran pelanggan yang dmasukan ke dalam sistem *chat* (Elmorshidy et al., 2015).

*Natural Language Processing* (NLP) bertujuan untuk mengembangkan program komputer yang dapat memproses bahasa alami yang dapat dipahami oleh manusia (Ma'ruf, 2019). NLP merujuk ke penggunaan sistem komputer untuk memproses bahasa alami, contohnya melakukan ekstraksi kalimat tertentu dari sebuah teks (Srinivasa-Desikan, 2018). *Conversational Interface* merupakan sebuah cara komunikasi antara manusia dan sistem komputer dengan bantuan yang berupa interface text, gambar, dan sebagainya untuk mempermudah sistem memahami maksud kalimat yang disampaikan (Janarthanam, 2017).



Gambar 1. Contoh *Conversational Interfaces* pada chat (CI)

*Formalizer* merupakan formalisasi atau perbaikan dari kata maupun kalimat yang memiliki *typo* ataupun bahasa tidak baku untuk diganti menjadi kata dengan kemiripan yang tertinggi. Jika kata dengan kemiripan yang cukup tidak ditemukan maka kata tersebut dianggap sebagai kata asing dan tidak mengalami formalisasi. Kata-kata yang dianggap formal dan *typo* dimasukkan ke

dalam sebuah file yang selanjutnya dijadikan sebagai database kata (Musthafa et al., 2018).

*Stemming* dan *lemmatization* adalah salah satu cara yang digunakan untuk meningkatkan performa temu balik informasi (*information retrieval*) dengan cara mentransformasi kata-kata dalam sebuah dokumen teks ke bentuk kata dasarnya. Proses *stemming* pada teks Berbahasa Indonesia lebih rumit/kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan *root word* (kata dasar) dari sebuah kata (Susandi & Sholahudin, 2016). Algoritma Nazief Adriani sebagai algoritma *stemming* untuk teks Berbahasa Indonesia yang memiliki kemampuan persentase keakuratan (presisi) lebih baik dari algoritma lainnya. Algoritma Nazief Adriani ini membutuhkan sebuah daftar kata dasar, sehingga bisa menggunakan bantuan *database* atau *array* pada sebuah program, di mana daftar kata dasar Bahasa Indonesia disimpan di dalam *database* (Susandi & Sholahudin, 2016).

*Regular Expression Pattern Matching* menggunakan sekumpulan *regular expression* yang disusun menjadi sebuah *pattern*. Bila kalimat masukan cocok dengan salah satu *pattern* yang ada, maka sistem akan melakukan proses sesuai perintah yang ditentukan pada *pattern* tersebut (Domarco & Iswari, 2017).

Penerimaan mahasiswa baru merupakan kegiatan rutin yang dilakukan oleh seluruh perguruan Tinggi di Indonesia setiap tahunnya. Kegiatan ini dilakukan sebagai titik awal proses pencarian input mahasiswa baru yang berkualitas (Kurniawan et al., 2015). Proses ini mulai dari promosi, rekrutmen, seleksi sampai dinyatakan lulus untuk menjadi mahasiswa baru. Pelaksanaan promosi untuk mendapatkan mahasiswa baru salah satunya melalui *website* kampus yang menyediakan fasilitas *live chat*.

Dari latar belakang tersebutlah, pada penelitian ini penulis menggunakan metode *word stemming* dengan *pattern matching regular expression matching* terhadap

*knowledge base* yang akan dikembangkan dengan basis *Natural Language Processing* (NLP) dan *Conversational Interfaces* (CI) untuk *live chatting* dalam penerimaan mahasiswa baru. Studi kasus yang diambil dalam penelitian ini adalah *live chat* penerimaan mahasiswa baru pada kampus LP3I.

## B. METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini metode penelitian dan pengembangan yaitu metode yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut (Sugiyono, 2017). Untuk pengembangan produknya menggunakan metode *Prototyping*. Sedangkan untuk pengumpulan data dilakukan dengan cara mengekstrak data berupa *history chat* dari sistem berjalan, *chat* WhatsApp dan *chat e-mail* yang akan dijadikan sebagai *knowledge base* sistem sebagai pola chat pada aplikasi *chatbot* antara pengguna dan aplikasi, pengumpulan data juga didapatkan dengan cara meneliti data dan observasi pada website resmi penerimaan mahasiswa baru kampus LP3I dan wawancara dengan bagian penerimaan mahasiswa baru untuk mendapatkan informasi umum mengenai proses sistem berjalan saat ini.

## C. HASIL DAN PEMBAHASAN

Sistem berjalan saat ini di Kantor Pusat LP3I terkait layanan *live chat support* dimulai dari calon mahasiswa baru atau pengguna *website* LP3I memulai sesi percakapan dengan membuka *window chat* Tawk.to pada *website*, lalu calon mahasiswa baru atau pengguna mengisi nama, alamat email, dan pertanyaan atau keluhan seputar proses penerimaan mahasiswa baru. Apabila admin yaitu staf marketing dan PR sedang *online*, maka user akan menunggu hingga admin tersedia untuk membalas, lalu sesi percakapan akan berlangsung, dan admin

dapat membalasnya secara langsung. Namun ketika admin *offline*, percakapan akan masuk ke *dashboard* admin, dan memberikan notifikasi ke email Admin bahwa ada sesi percakapan baru, kemudian Admin membalas chat tersebut yang selanjutnya akan diterima oleh user dalam bentuk balasan email. Ketika user memiliki pertanyaan selanjutnya, maka user dapat membalas email tersebut untuk melanjutkan sesi percakapan.

Pada sistem *live chat support* yang berjalan saat ini, ditemukan kendala yaitu sistem *live chat support* mengharuskan seorang admin untuk tetap online untuk mencapai layanan bantuan yang baik, dan *slow-response* dalam menanggapi pertanyaan-pertanyaan dan keluhan yang disampaikan oleh calon mahasiswa baru, padahal beberapa pertanyaan memiliki jawaban yang hampir sama, dan mereka harus menunggu jawaban yang dibalas oleh admin. Sebagai solusi pemecahan masalah maka dibangun *Chatbot* untuk digunakan sebagai pengganti sistem *live chat support* yang digunakan saat ini, dengan pilihan *auto-reply* dengan bahasa yang natural.

Metode pengembangan perangkat lunak *prototyping* yang digunakan berorientasi pada sistem requirement, yang telah diklasifikasikan menjadi Elisitasi tahap I hingga final draft Elisitasi. Proses elisitasi setiap tahap dapat dilihat pada tabel 1 sampai tabel 4.

Pada Elisitasi tahap I ini diperoleh dari hasil pengumpulan data yang dilakukan dengan cara observasi dan wawancara mengenai kebutuhan sistem.

Tabel 1. Hasil Elisitasi Tahap I

<b>Functional: Analisis Kebutuhan</b>	
No	Keterangan
1	Menampilkan menu <i>login</i>
2	Menampilkan menu <i>dashboard admin</i>
3	Menampilkan data <i>history chat</i>
4	Menampilkan tampilan <i>live chat</i>

5	Menampilkan tampilan <i>input</i> nama dan <i>email</i> ketika memulai sesi <i>chat</i>
6	Terdapat opsi untuk Admin dapat membalas <i>chat</i> manual
7	Dapat menjawab pertanyaan umum mengenai penerimaan mahasiswa baru
8	Dapat melihat <i>user</i> yang sedang <i>online</i>

**Non Functional : Saya ingin sistem dapat:**

1	Mudah digunakan oleh <i>user</i>
2	Menampilkan <i>login</i> dan <i>logout</i>
3	Tampilan menarik
4	Menampilkan data dan dapat mengelola <i>knowledge base</i>
5	Menampilkan data pertanyaan tidak terjawab

Elisitasi tahap II mengklasifikasikan kembali Elisitasi tahap I dengan metode MDI (*Mandatory, Desireable, Inessential*). *Requirement* yang diberi opsi *Inessential* (I) harus dieliminasi.

Tabel 2. Hasil Elisitasi Tahap II

<b>Functional: Analisis Kebutuhan</b>			
No	Keterangan	M	D I
1	Menampilkan menu <i>login</i>	√	
2	Menampilkan menu <i>dashboard admin</i>		√
3	Menampilkan data <i>history chat</i>	√	
4	Menampilkan tampilan <i>live chat</i>	√	
5	Menampilkan tampilan <i>input</i> nama dan <i>email</i> ketika memulai sesi <i>chat</i>		√
6	Terdapat opsi untuk Admin dapat membalas <i>chat</i> manual		√
7	Dapat menjawab pertanyaan umum mengenai penerimaan mahasiswa baru	√	
8	Dapat melihat <i>user</i> yang sedang <i>online</i>		√
<b>Non Functional: Saya ingin sistem dapat:</b>			
1	Mudah digunakan oleh <i>user</i>	√	
2	Menampilkan <i>login</i> dan <i>logout</i>	√	
3	Tampilan menarik		√
4	Menampilkan data dan dapat mengelola <i>knowledge base</i>	√	
5	Menampilkan data pertanyaan tidak terjawab		√

Elisitasi tahap III mengklasifikasikan kembali Elisitasi tahap II dengan metode TOE (*Technical, Operational, Economical*): *Technical* yaitu tingkat kesulitan dalam pengembangan sistem yang akan dibuat, *Operational* merupakan tingkat kesulitan dalam penggunaan sistem oleh user dan *Economical* yaitu tingkat biaya yang dibutuhkan dalam pengembangan sistem, dengan opsi HML (*High, Middle, Low*). Requirement dengan opsi *High* (H) harus dieliminasi

Tabel 3. Hasil Elisitasi Tahap III

<b>Functional: Analisis Kebutuhan</b>										
No	Keterangan	T			O			E		
		H	M	L	H	M	L	H	M	L
1	Menampilkan menu <i>login</i>		√				√			√
2	Menampilkan menu <i>dashboard admin</i>		√				√			√
3	Menampilkan data <i>history chat</i>		√				√			√
4	Menampilkan tampilan <i>live chat</i>		√				√			√
5	Menampilkan tampilan <i>input</i> nama dan <i>email</i> ketika memulai sesi <i>chat</i>		√				√			√
6	Terdapat opsi untuk Admin dapat membalas <i>chat</i> manual		√				√			√
7	Dapat menjawab pertanyaan umum mengenai penerimaan mahasiswa baru		√				√			√
8	Dapat melihat <i>user</i> yang sedang <i>online</i>		√				√			√
<b>Non Functional: Saya ingin sistem dapat</b>										
1	Mudah digunakan oleh <i>user</i>		√				√			√

2	Menampilkan <i>login</i> dan <i>logout</i>	√		√						√
3	Tampilan menarik	√					√			√
4	Menampilkan data dan dapat mengelola <i>knowledge base</i>	√					√			√
5	Menampilkan data pertanyaan tidak terjawab	√					√			√

Tahap terakhir dalam elisitasi yang mana merupakan sebuah daftar requirement yang telah di klasifikasikan dan menghilangkan requirement yang harus dieliminasi.

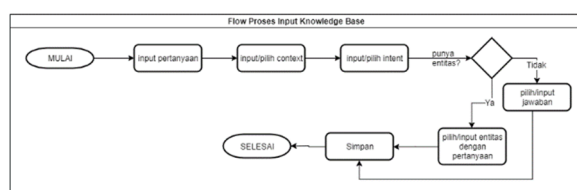
Tabel 4. Hasil Elisitasi Tahap Final

<b>Functional: Analisis Kebutuhan</b>	
No	Keterangan
1	Menampilkan menu <i>login</i>
2	Menampilkan menu <i>dashboard admin</i>
3	Menampilkan data <i>history chat</i>
4	Menampilkan tampilan <i>live chat</i>
5	Menampilkan tampilan <i>input</i> nama dan <i>email</i> ketika memulai sesi <i>chat</i>
6	Terdapat opsi untuk Admin dapat membalas <i>chat</i> manual
7	Dapat menjawab pertanyaan umum mengenai penerimaan mahasiswa baru
8	Dapat melihat <i>user</i> yang sedang <i>online</i>
<b>Non Functional: Saya ingin sistem dapat:</b>	
1	Mudah digunakan oleh <i>user</i>
2	Menampilkan <i>login</i> dan <i>logout</i>
3	Tampilan menarik
4	Menampilkan data dan dapat mengelola <i>knowledge base</i>
5	Menampilkan data pertanyaan tidak terjawab

Dalam perancangan aplikasi chatbot ini, secara garis besar terdapat dua proses, yakni proses '*teaching chatbot*' yaitu mengajari *chatbot* untuk memberikan respons seperti apa ketika ditanya apa, lalu proses yang kedua yaitu proses utama *chatbot* bekerja. Proses '*teaching chatbot*' dilakukan pada *interface* web oleh Admin, proses dimulai

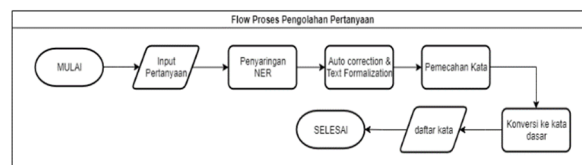
dengan Admin memasukkan sebuah pertanyaan, lalu memasukkan respons yang seharusnya diberikan, lalu memasukkan juga kata kunci yang menandai pertanyaan/jawaban itu mengenai hal apa. Ketiga data masukkan ini diperlukan untuk proses *pattern matching* ketika mencari jawaban oleh *chatbot*. Data-data pertanyaan dan respons jawaban dapat dikumpulkan dari *history chat* dari sistem yang berjalan, yaitu *history chat* WhatsApp dan *history chat* sistem Tawk.to.

Pertanyaan dan jawaban kemudian diproses kembali dengan beberapa algoritma dasar yaitu *tokenize word* untuk memecah kalimat menjadi kata-kata, diproses kembali dengan metode *stemming* dan *lemmatization* untuk mendapatkan bentuk dasar dari kata-kata tersebut, setelah itu hasilnya disimpan ke *database* untuk digunakan sebagai *pattern matching*.



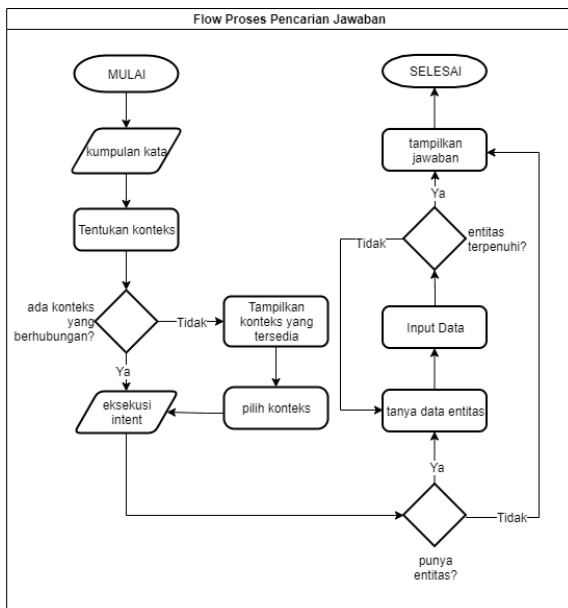
Gambar 2. Flowchart input *knowledge base*

Pada *knowledge base chatbot*, dibagi menjadi dua bagian besar, yaitu *knowledge base* yang dapat dimengerti oleh *chatbot* (*Answered Questions*), dan *knowledge base* yang tidak dapat dimengerti oleh *chatbot* (*Unanswered Questions*). *Answered Questions* berisi kumpulan pertanyaan dan jawaban yang sebelumnya sudah diajarkan pada *chatbot*, sedangkan *Unanswered Questions* merupakan kumpulan pertanyaan yang belum bisa terjawab oleh *chatbot*, yang ke depannya dapat digunakan oleh Admin untuk dijawab, dan diajarkan kepada *chatbot* bagaimana seharusnya merespons, selanjutnya hal ini akan menjadi *knowledge base Answered Questions*. Bagian input *knowledge base* ini merupakan sebuah aplikasi berbasis web yang dibangun dengan *library React.JS*.



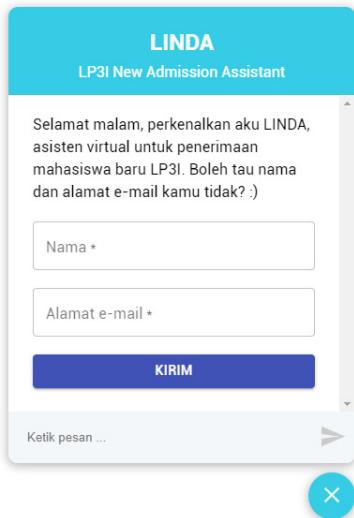
Gambar 3. Flowchart pemrosesan kalimat

Proses berikutnya adalah proses *chatbot* memahami konteks kalimat dan memberikan respons yang tepat. Pada proses ini, *chatbot* hanya akan menggunakan *pattern matching* berbasis *Cosine Similarity* berdasarkan bentuk kata dasar dan kata kunci kalimat. Ketika *chatbot* menerima sebuah pertanyaan, *chatbot* akan melakukan *word tokenizing* dan *word stemming* pada pertanyaan tersebut, kemudian *chatbot* melakukan pencocokan dari hasil *tokenize* dan *stemming* menggunakan algoritma *Cosine Similarity* terhadap *knowledge base* yang telah disimpan, apabila *chatbot* tidak menemukan hasil yang lebih dari atau sama dengan 70% kesamaannya, maka *chatbot* akan melakukan pencocokan berdasarkan kata kunci pertanyaan untuk menampilkan *Conversational Interface* yang memberikan beberapa opsi yang dapat dipilih user sebagai alternatif pencarian jawaban. Tetapi jika dalam pencocokan kata kunci juga tidak menemukan apa-apa, maka *chatbot* akan memberikan respons bahwa *chatbot* tidak mengenali bentuk pertanyaan yang diberikan dan meminta mengulang pertanyaan kembali atau menghubungi Admin langsung, pada proses ini *chatbot* akan menyimpan pertanyaan tersebut sebagai *knowledge base* pertanyaan yang belum bisa terjawab. Proses ini dilakukan pada aplikasi *server* berbasis Node.JS dengan penggunaan *library socket.io* untuk *service chatnya*, dibuat dengan bahasa pemrograman JavaScript.



Gambar 4. Flowchart *chatbot response*

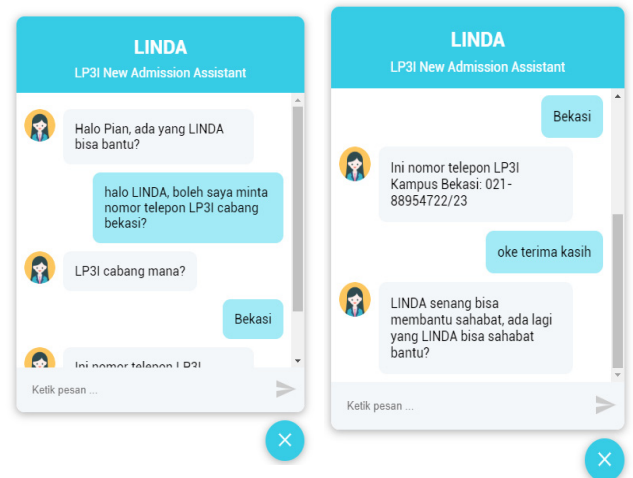
Tampilan awal aplikasi pada *platform web* berupa *widget chat* dapat dilihat pada gambar 5, di mana publik dapat melakukan interaksi dengan *chatbot*.



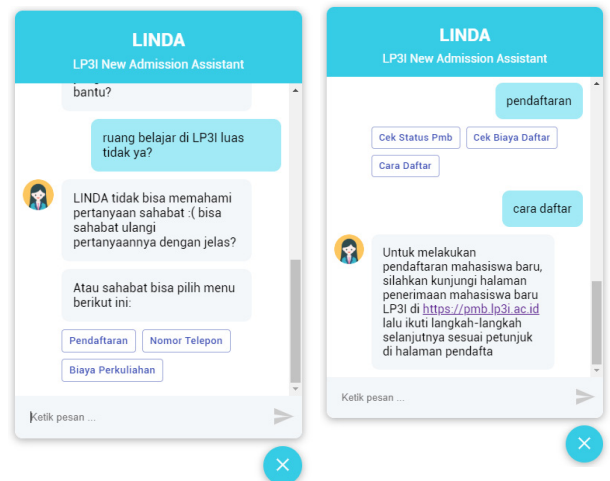
Gambar 5. Tampilan awal aplikasi *chat*

Tampilan sambutan dan interaksi antara pengguna dengan *chatbot* dapat dilihat pada gambar 6, sementara ketika *chatbot* tidak memahami maksud pertanyaan, maka *chatbot* akan memberikan opsi berupa menu bantuan, atau dapat memilih untuk melakukan *chat* dengan Admin

langsung seperti yang terlihat pada gambar 7.

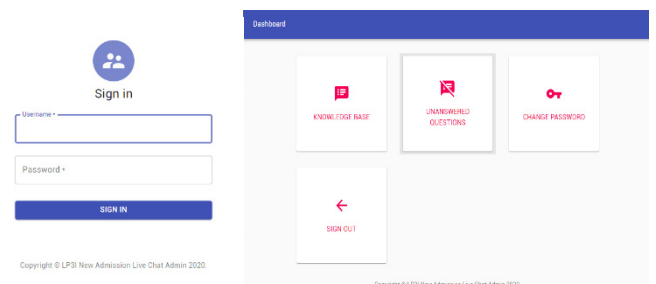


Gambar 6. Tampilan interaksi *user* dengan aplikasi



Gambar 7. *Conversational Interface* pada *chatbot*

Tampilan halaman *login* untuk *dashboard Admin* di *web* dan tampilan awal / menu pada *dashboard*.

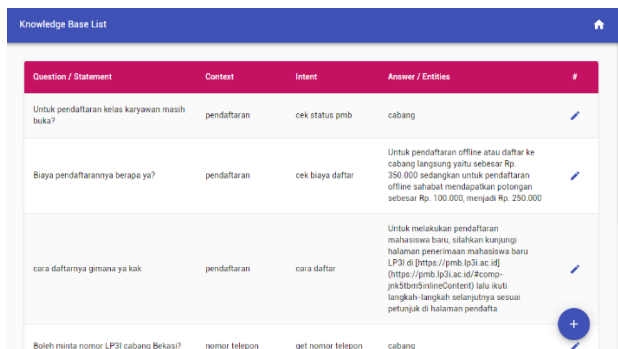


Gambar 8. Tampilan *login* dan halaman *dashboard Admin*

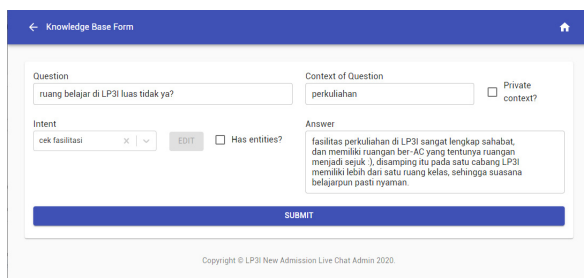


Aplikasi *Chatbot* Sebagai Layanan *Live Chat* Untuk Penerimaan Mahasiswa Baru Menggunakan Metode *Word Stemming* Dengan *Regular Expression Pattern Matching*  
 Muhamad Rizki, Ahmad Fitriansyah, Moh. Narji

Tampilan *dashboard* kelola *knowledge base chatbot*, Admin dapat menambah atau mengubah *knowledge base* yang sudah ada

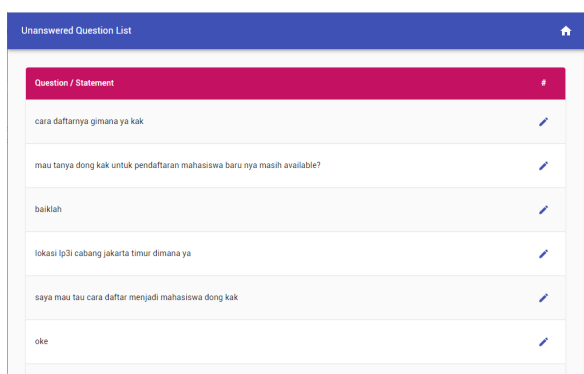


Gambar 9. Tampilan menu *list knowledge base chatbot*



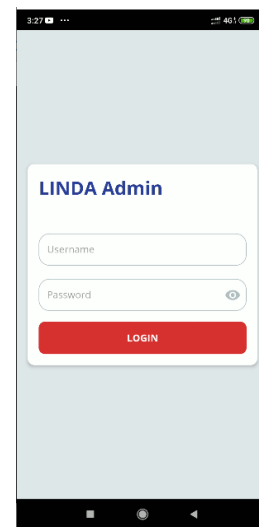
Gambar 10. Tampilan *form edit/tambah knowledge base*

Tampilan *dashboard* daftar pertanyaan yang tidak terjawab oleh *chatbot*, pada tampilan ini Admin dapat membuat *knowledge base* baru berdasarkan pertanyaan yang tidak terjawab.



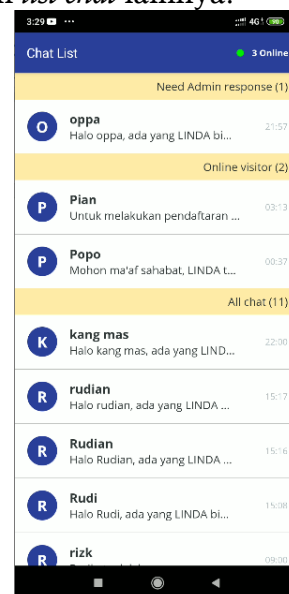
Gambar 11. Tampilan *list pertanyaan tidak terjawab*

Tampilan *login* pada *platform mobile apps* yang ditujukan untuk Admin.



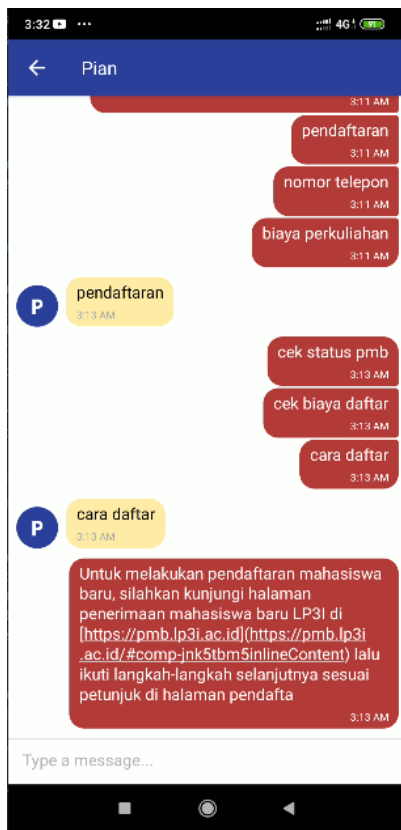
Gambar 12. Tampilan *login* aplikasi *mobile Admin*

Tampilan daftar *history chat*, yang diurutkan berdasarkan tanggal *chat* terakhir, dengan urutan prioritas, *waiting list* yang ingin *chat* dengan Admin, lalu *list online*, dan *list chat* lainnya.



Gambar 13. Tampilan *list history chat*

Tampilan *chat* pada aplikasi *mobile*, pada tampilan ini Admin dapat memberikan respons manual apabila *room chat* berada pada *waiting list*, jika tidak maka Admin hanya dapat melihat *history chat*.



Gambar 14. Tampilan chat pada mobile apps

### Pengujian Aplikasi

Metode pengujian pada aplikasi ini dilakukan dengan metode *BlackBox Testing*, dengan skenario sebagai berikut:

Tabel 5. *Scenario case* interaksi user dengan chatbot

Scenario case	Expected result	Testing result
User memasukkan nama dan email, lalu meng-klik tombol kirim	Chatbot berhasil terinisialisasi dan memberikan kalimat sambutan kepada user	Diterima
User memasukkan nama dan email dengan format email yang salah (kalimat acak)	Chatbot menolak input user, dan mengharuskan user memasukkan alamat email dengan format yang valid	Diterima
User memberikan pertanyaan kepada chatbot seputar penerimaan	Chatbot dapat memberikan tanggapan yang	Diterima

mahasiswa baru dan kampus LP3I  
 - cara daftar  
 - biaya pendaftaran  
 - info nomor telepon  
 - mengetahui program studi dan jurusan  
 - mengetahui lokasi cabang kampus

Pengujian agar User dapat memilih menu ketika tidak mendapatkan jawaban yang tepat dari chatbot

Tabel 6. *Scenario case* interaksi user dengan menu chatbot

Scenario case	Expected result	Testing result
User bertanya mengenai hal-hal yang tidak bersangkutan dengan kampus LP3I ataupun penerimaan mahasiswa baru.	Chatbot memberikan user interface berupa menu yang membantu user untuk berinteraksi dengan chatbot	Diterima
User memilih menu untuk melakukan sesi chat dengan Admin	Chatbot menampung user tersebut ke dalam waiting list yang harus direspons Admin.	Diterima

Pengujian user Admin dapat login ke dashboard web

Tabel 7. *Scenario case* interaksi Admin pada halaman login

Scenario case	Expected result	Testing result
Admin memasukkan username dan password yang benar	Admin masuk ke halaman dashboard web	Diterima
Admin memasukkan username dan password yang salah	System menolak aksi tersebut sehingga Admin tidak dapat masuk ke dashboard	Diterima

Pengujian user Admin dapat melihat dan mengelola list knowledge base chatbot

Tabel 8. *Scenario case* Admin mengelola *knowledge base*

<i>Scenario case</i>	<i>Expected result</i>	<i>Testing result</i>
Admin memilih menu <i>knowledge base</i>	Web Admin menampilkan daftar <i>knowledge base chatbot</i>	Diterima
Admin menambah dan mengubah <i>knowledge base</i> yang ada	Semua data yang di <i>input</i> Admin pada <i>knowledge base</i> tersimpan di <i>database</i>	Diterima

Pengujian user Admin dapat melihat *list* pertanyaan yang tidak terjawab

Tabel 9. *Scenario case* Admin melihat daftar pertanyaan tidak terjawab

<i>Scenario case</i>	<i>Expected result</i>	<i>Testing result</i>
Admin memilih menu <i>Unanswered Question</i>	Web Admin akan menampilkan daftar pertanyaan yang tidak terjawab.	Diterima
Admin memilih sebuah pertanyaan untuk dijadikan <i>knowledge base</i> baru	Admin dapat membuat <i>knowledge base</i> baru berdasarkan pertanyaan yang dipilih	Diterima

Pengujian Admin dapat login ke *dashboard mobile*

Tabel 10. *Scenario case* interaksi Admin dengan *dashboard mobile application*

<i>Scenario case</i>	<i>Expected result</i>	<i>Testing result</i>
Admin memasukkan <i>username</i> dan <i>password</i> yang benar	Aplikasi meneruskan aksi ke halaman <i>dashboard mobile</i>	Diterima
Admin memasukkan <i>username</i> dan <i>password</i> yang salah	Aplikasi menolak permintaan Admin dan tidak meneruskan ke <i>dashboard</i>	Diterima

Pengujian Admin dapat melihat *history chat*

Tabel 11. *Scenario case* Admin melihat *history chat*

<i>Scenario case</i>	<i>Expected result</i>	<i>Testing result</i>
Admin ingin melihat <i>waiting list chat user</i>	Aplikasi menampilkan <i>waiting list chat user</i>	Diterima
Admin ingin melihat <i>user</i> yang sedang <i>online</i>	Aplikasi menampilkan <i>list user yang online</i>	Diterima
Admin ingin melihat <i>history chat</i> sebelumnya	Aplikasi menampilkan <i>list history chat</i>	Diterima

Pengujian Admin dapat membalas manual *chat* yang ada pada *waiting list*

Tabel 12. *Scenario case* Admin memberikan respons manual kepada pertanyaan *user*

<i>Scenario case</i>	<i>Expected result</i>	<i>Testing result</i>
Admin memilih sebuah <i>room chat</i> pada <i>waiting list</i> , dan membalas pesan	Pesan dari Admin akan diteruskan pada aplikasi <i>user</i> di web	Diterima
Admin melakukan aksi tutup sesi <i>chat</i> untuk mengakhiri <i>chat</i> yang sedang aktif	Admin dapat menutup <i>room chat</i> pada <i>waiting list</i> apabila sesi ingin di akhiri	Diterima

Kesimpulan pengujian dari 18 *scenario test case* yang telah dilakukan pada tabel 5 sampai dengan tabel 12, semua hasil *scenario test case* secara keseluruhan sesuai harapan, aplikasi dapat memberikan hasil *output* dan validasi yang sesuai 100%. *Chatbot* dapat berkembang menjadi lebih baik apabila *chatbot* terus menerus dilatih dengan melakukan penambahan *knowledge base*. Namun *chatbot* akan menemui kesulitan dalam memproses kalimat dan tidak bisa menjawab, jika kalimat pertanyaan memiliki banyak kesalahan dalam penulisan. Sehingga *chatbot* hanya akan mengembalikan tanggapan berupa menu.

#### D. PENUTUP

Aplikasi dirancang berdasarkan aplikasi yang berjalan saat ini di LP3I, yang masih menggunakan layanan *chat messenger* seperti Facebook dan WhatsApp. Sehingga *knowledge base* aplikasi terbatas hanya terpaku berdasarkan *history chat* yang sudah ada pada layanan *chat messenger* tersebut.

Layanan *live chat* penerimaan mahasiswa baru saat ini masih dilakukan secara manual, sehingga waktu respons terhadap pertanyaan calon mahasiswa pun menjadi sangat lama, memerlukan paling lama satu hari kerja untuk mendapat balasan. Hal ini masih kurang efektif pada layanan saat ini.

Rancangan aplikasi *chatbot* sebagai layanan *live chat* ini dapat mengatasi permasalahan pada layanan yang sedang berjalan, sehingga layanan dalam *live chat support* pada penerimaan mahasiswa baru pun dapat lebih baik dari sebelumnya, dimana penanya dapat menerima jawaban dengan instan, dan tidak terikat waktu kerja. Dengan *chatbot* calon mahasiswa dapat bertanya kapan pun dan mendapatkan jawaban pada saat itu juga. Hal ini dapat meningkatkan kualitas layanan, selain itu dari segi efektivitas dan produktivitas staf marketing juga lebih baik dari sistem yang berjalan sebelumnya, karena staf marketing tidak lagi harus selalu *standby* dengan aplikasi chat.

Untuk pengembangan lebih lanjut dari hasil penelitian ini, maka ada beberapa hal yang dapat dilakukan untuk lebih meningkatkan akurasi dan kecerdasan *chatbot* dalam berinteraksi dengan user, di antaranya dengan menerapkan *Part Of Speech (POS) Tagging* pada kalimat, dan menerapkan metode pengembangan *chatbot* dengan pendekatan *deep-learning Recurrent Neural Network (RNN)*, sehingga proses training *chatbot* bisa lebih akurat dan mendekati harapan terutama dalam bagian pemrosesan *Natural Language Processing (NLP)*.

#### E. DAFTAR PUSTAKA

- Amalia, E. L., & Wibowo, D. W. (2019). Rancang Bangun Chatbot Untuk Meningkatkan Performa Bisnis. *Jurnal Ilmiah Teknologi Informasi Asia*, 13(2), 137–142.
- Benedictus, R. R., Wowor, H. F., & Sambul, A. (2017). Rancang Bangun Chatbot Helpdesk untuk Sistem Informasi Terpadu Universitas Sam Ratulangi. *Jurnal Teknik Informatika*, 11(1).
- Domarco, D., & Iswari, N. M. S. (2017). Rancang Bangun Aplikasi Chatbot Sebagai Media Pencarian Informasi Anime Menggunakan Regular Expression Pattern Matching. *Ultimatics: Jurnal Teknik Informatika*, 9(1), 19–24.
- Elmorshidy, A., Mostafa, M. M., El-Moughrab, I., & Al-Mezen4, H. (2015). Factors Influencing Live Customer Support Chat Services: An Empirical Investigation in Kuwait. *Journal of Theoretical and Applied Electronic Commerce Research*, 10(3).
- Janarthanam, S. (2017). *Hands-On Chatbots and Conversational UI Development, Build chatbots and voice user interfaces with Chatfuel, Dialogflow, Microsoft Bot Framework, Twilio, and Alexa Skills*. Birmingham : Packt Publishing.
- Kurniawan, E., Mustafidah, H., & Shofiyani, A. (2015). Metode Topsis Untuk Menentukan Penerimaan Mahasiswa Baru Pendidikan Dokter di Universitas Muhammadiyah Purwokerto. *JUITA: Jurnal Informatika*, 3(4), 201–206.
- Ma'ruf, F. (2019). *Skripsi : Implementasi Chatter Bot sebagai Media Informasi Sekolah dengan Pendekatan Natural Language Processing Berbasis Web (Studi Kasus: Madrasah Aliyah Negeri Yogyakarta 2)*. University of Technology Yogyakarta.
- Murhadi. (2019). Rancang Bangun Aplikasi

Chatbot Sebagai Bentuk Pelayanan Prima Untuk Penerimaan Mahasiswa Baru. *Intek: Jurnal Informatika Dan Teknologi Informasi*, 2(1), 10–16.

Musthafa, M. F., Buliali, J. L., & Hariadi, V. (2018). Pemodelan Multilabel Tweet Media Sosial Mahasiswa Untuk Klasifikasi Keluhan. *Jurnal Teknik ITS*, 7(1), A247–A252.

Srinivasa-Desikan, B. (2018). *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Birmingham : Packt Publishing.

Sugiyono. (2017). *Metode Penelitian Bisnis: Pendekatan Kuantitatif, Kualitatif, Kombinasi dan R&D*. Bandung : Alfabeta.

Susandi, D., & Sholahudin, U. (2016). Pemanfaatan Vector Space Model pada Penerapan Algoritma Nazief Adriani, KNN dan Fungsi Similarity Cosine untuk Pembobotan IDF dan WIDF pada Prototipe Sistem Klasifikasi Teks Bahasa Indonesia. *ProTekInfo(Pengembangan Riset Dan Observasi Teknik Informatika)*, 3(1), 22–29. <https://doi.org/10.30656/protekinfo.v3i0>.  
54