

## Perancangan *Chatbot* Pusat Layanan Kantor LAZISMU DIY dengan Pendekatan *Natural Language Processing* dan Algoritma Brute Force

Dewi Soyusiawaty<sup>1</sup>, Almira Haswinitha<sup>2</sup>

<sup>1</sup>Program Studi Informatika, Universitas Ahmad Dahlan, Jl. Ringroad Selatan, Yogyakarta, 55191  
e-mail: <sup>1</sup>dewi.soyusiawaty@tif.uad.ac.id, <sup>2</sup>almirahaswinitha99@gmail.com

Submitted Date: August 14<sup>th</sup>, 2022  
Revised Date: March 16<sup>th</sup>, 2023

Reviewed Date: March 03<sup>rd</sup>, 2023  
Accepted Date: March 23<sup>rd</sup>, 2023

### Abstract

The service center in a company or agency is a place to serve various kinds of questions and customer complaints served by a customer service. However, it is undeniable that the role of customer service has many shortcomings, such as having to answer repeated questions given to different customers and limited hours. With the rapid development of technology, this role is now supported by an interactive automation system in answering questions given to customers. The automation system in question is a virtual assistant in the form of a chatbot. The application in the chatbot design is built by utilizing text mining reasoning, applying patterns between user input and the database using the Brute Force algorithm, and the process of correcting erroneous words using the Peter Norvig algorithm. Text mining can be defined as the process of extracting the required information from textual sources, the Brute Force algorithm performs a pattern from left to right, and the Peter Norvig method for correcting words by deciphering each letter in the wrong word. The chatbot design is useful in increasing the quality and effectiveness of services at LAZISMU DIY. The chatbot system that was built certainly passed the method validation test process. Testing is done by implementing the black-box testing method. The test results obtained are based on 7 test scenarios with a total of 96 questions, which is 93%.

Keywords: customer service; chatbot; text mining; Brute Force; Peter Norvig

### Abstrak

Pusat layanan di sebuah perusahaan atau instansi menjadi wadah dalam melayani berbagai macam pertanyaan serta keluhan pelanggan yang dilayani oleh seorang customer service. Namun, tidak dapat dipungkiri jikalau peran customer service memiliki banyak kekurangan, seperti harus menjawab pertanyaan berulang yang diberikan pelanggan berbeda serta jam pelayanan yang terbatas. Semakin pesat perkembangan teknologi, kini peran tersebut digantikan oleh sistem otomatisasi yang interaktif dalam menjawab pertanyaan yang diberikan pelanggan. Sistem otomatisasi yang dimaksud yaitu virtual assistant berupa chatbot. Penerapan metode dalam rancangan chatbot yang dibangun menggunakan bahasa pemrograman python dengan memanfaatkan penalaran text mining, pencocokan pola antara masukan pengguna dengan database menggunakan algoritma Brute Force, dan proses koreksi kata yang keliru menggunakan algoritma Peter Norvig. Text Mining dapat diartikan sebagai proses ekstraksi informasi yang dibutuhkan dari sumber tekstual, algoritma Brute Force melakukan pencocokan pola dari kiri ke kanan, serta metode Peter Norvig untuk mengoreksi kata dengan cara menguraikan tiap huruf pada kata yang keliru. Rancang bangun chatbot berguna dalam menjadikan kualitas dan efektifitas layanan di LAZISMU DIY meningkat. Sistem chatbot yang dibangun tentu melewati proses uji validasi metode. Pengujian dilakukan dengan mengimplementasikan metode black-box testing. Hasil pengujian yang didapat berdasarkan 7 skenario pengujian dengan total 96 pertanyaan yaitu sebesar 93%

Kata Kunci: pelayanan pelanggan; chatbot; text mining; Brute Force; Peter Norvig

## 1 Pendahuluan

Sebagian besar perusahaan pada masa yang lalu hingga masa kini pasti memiliki *customer service* sebagai layanan perusahaan tersebut. *Customer service* memiliki peranan penting di sebuah perusahaan dalam melayani pertanyaan pelanggan agar menghasilkan pendapatan dan omset. Terkadang masalah yang sering terjadi seperti sering mendapati pertanyaan yang sama dan tidak memungkinkan untuk mendukung layanan 7x24. Belakangan ini telah banyak penggunaan asisten virtual berupa *chatbot* untuk melayani pelanggan di kalangan bisnis. Dengan adanya *chatbot* menjadikan solusi dalam peran customer service karena lebih ekonomis, digunakan kapanpun, dan mengurangi tugas staff dalam melayani pertanyaan pelanggan (Cui et al., 2017). Namun dikarenakan perubahan teknologi yang semakin pesat, *customer service* juga hadir untuk melayani pelanggan di media sosial yang dikenal dengan admin. Biasanya admin yang melayani di media sosial membantu menjawab pertanyaan pelanggan.

Satu di antara banyak perusahaan yang memiliki *customer service* atau admin adalah Lembaga Amil Zakat Infaq Shadaqah Muhammadiyah (LAZISMU) yang memiliki banyak cabang di Indonesia dan salah satunya ialah cabang LAZISMU DIY. LAZISMU DIY merupakan lembaga yang menyediakan wadah untuk masyarakat dalam melakukan zakat atau dana kederewanan lain. Tugas *customer service* atau admin di LAZISMU DIY seperti pada umumnya yaitu melayani pertanyaan masyarakat seputar zakat atau yang berhubungan dengan LAZISMU DIY. Pusat layanan di LAZISMU DIY masih termasuk manual, di mana belum ada sistem atau fitur khusus untuk melayani pelanggan secara otomatis. Apabila ada masyarakat yang ingin bertanya seputar kantor LAZISMU DIY atau ZIS (Zaka Infaq Shadaqah), maka hanya dapat dilakukan dengan menghubungi admin melalui WhatsApp yang berperan sebagai *customer service*. Namun dapat pula melihat daftar pertanyaan yang di *Frequently Asked Questions* (FAQ) yang sudah tersedia di *website*. FAQ berasal dari pertanyaan yang sering diajukan oleh pengguna yang dikemas menjadi satu agar dapat dilihat kembali. Namun kelemahan dari FAQ yaitu pengguna harus mencari dan membandingkan antara pertanyaan yang ingin ditanyakan dengan

pertanyaan yang ada di daftar FAQ. Hal tersebut tentu menjadikan layanan yang ada di LAZISMU DIY kurang efektif karena keterbatasan yang ada, seperti admin yang memegang *WhatsApp* tentu tidak selalu aktif atau memberi tanggapan yang cepat. Sama halnya pada FAQ yang memiliki keterbatasan pertanyaan yang telah disediakan. Pada akhirnya berakibat pada kualitas layanan menjadi menurun.

Berdasarkan pemaparan di atas, maka kendala tersebut dapat di atasi dengan adanya sistem *virtual assistant* berupa *chatbot* untuk membantu menjawab pertanyaan dari pengguna secara otomatis dengan tujuan agar kualitas dan efektifitas layanan di LAZISMU DIY dapat meningkat. *Chatbot* akan secara otomatis menjawab pertanyaan pengguna selama pertanyaan tersebut sesuai dengan *pattern* yang ada di *database*. *Pattern* yang digunakan berasal dari FAQ yang ada di *website* LAZISMU DIY.

*Chatbot* memiliki artian suatu otomatisasi yang bekerja sebagai layanan obrolan untuk memberi informasi melalui pesan dari pengguna (Paliwahet et al., 2017). Berdasarkan pada perubahan zaman yang lebih canggih, *chatbot* berorientasi pada pemberian layanan secara *virtual* yang melayani pengguna tanpa ada batasan waktu (Wijaya & Toba, 2020). Oleh karena itu, keberadaan *chatbot* menjadikan kualitas layanan perusahaan dapat meningkat (Muhammad Sidik, Bambang Gunawan, 2021). Berbagai macam metode dalam sistem *chatbot* yang telah dilakukan pada penelitian yang lalu di antaranya metode *cosine similarity* dengan TF-IDF (Sains et al., 2021), berbasis aturan (*rule-based*) (Ishlakhuddin et al., 2020), dan *string matching* seperti algoritma booyer moore (Salisah et al., 2020).

Penelitian terdahulu pertama terkait rancangan *chatbot* dengan metode *cosine similarity* untuk perhitungan kemiripan dokumen dan menggunakan query dan metode TF-IDF untuk perhitungan bobot masing-masing kata. Tujuan penelitian untuk menyalurkan informasi terkait imuniasi kepada orang tua saat pandemi. Hasil pengujian yang didapat adalah 77% *precision*, 74% *recall*, dan 77% *accuracy* (Sains et al., 2021). Penelitian terdahulu kedua dengan metode *rule-based* yaitu berbasis aturan *IF-THEN* menggunakan *telegram*. Tujuan penelitian untuk menyalurkan informasi terkait Biro Administrasi Akademik (BAAK). Hasil pengujian yang didapat

adalah 98,95% (Ishlakhuddin et al., 2020). Penelitian terdahulu ketiga dengan algoritma booye rmoore menggunakan pencocokan *string* dengan pengecekan kesamaan pola dari kanan ke kiri. Tujuan penelitian untuk menyalurkan informasi seputar wisata Yogyakarta kepada wisatawan dengan hasil pengujian yang sesuai harapan (Salisah et al., 2020).

Penelitian ini akan dibangun *virtual assistant* berupa *chatbot* LAZISMU DIY (LazismuBot) menggunakan pendekatan *Natural Language Processing* (NLP) dan penalaran *text mining*. Sedangkan untuk pencocokan pola antara masukan pengguna dengan *database* menggunakan algoritma *Brute Force* dan pengoreksian kata yang keliru pada pesan pengguna menggunakan algoritma *Peter Norvig*.

## 2 Metodologi Penelitian

Tahap penelitian sistem *chatbot* terdiri atas identifikasi masalah, pengumpulan data, analisis sistem, rancangan sistem, dan uji sistem seperti yang terlihat pada gambar 1.



Gambar 1 Tahap Penelitian

### 2.1 Tahap Identifikasi Masalah

Penelitian ini melihat dari kondisi lapangan yang ada di LAZISMU DIY masih dilakukan manual dalam menjawab pertanyaan masyarakat yang seringkali berulang melalui pesan WhatsApp sehingga dibutuhkan sebuah sistem untuk meningkatkan kualitas dan efektivitas layanan.

### 2.2 Tahap Pengumpulan Data

Pada penelitian ini terdapat dua tahap dalam pengumpulan data yaitu tahap wawancara dan tahap observasi. Tahap wawancara dilakukan dengan berkomunikasi langsung antara peneliti dan pihak LAZISMU DIY untuk mengumpulkan data yang diperlukan dalam penelitian ini. Sedangkan tahap observasi dilakukan dengan mengamati dan mengumpulkan data yang ada di FAQ website LAZISMU DIY yaitu [www.lazismudiy.or.id](http://www.lazismudiy.or.id).

### 2.3 Analisis dan Perancangan Sistem

Suatu penelitian diperlukan tahap analisis dan perancangan sistem agar peneliti dapat mengetahui kebutuhan pada sistem yang akan dibangun. Analisis kebutuhan data bertujuan untuk

dapat mengidentifikasi keterbatasan data yang akan digunakan dalam penelitian. Berikut beberapa data yang dibutuhkan dalam penelitian ini seperti yang terlihat pada tabel 1.

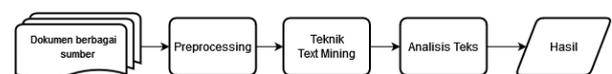
Tabel 1 *Pattern* dan Tanggapan

<i>Pattern</i>	Tanggapan
cara donasi	"Silakan ikuti tata cara berdonasi dibawah ini ya: ..."
metode bayar donasi	"Sobat Lazismu dapat berbagi kebaikan dengan berdonasi menggunakan pilihan metode yang kami sediakan yaitu ..."
cara daftar donasi	"Untuk mendaftarkan program galang dana, dapat mengikuti langkah-langkah berikut: ..."
ketentuan deskripsi	"Berikut ketentuan dalam deskripsi program galang dana: ..."
ketentuan periode	"Untuk ketentuan periode ditentukan langsung oleh campaigner, sesuai dengan kebutuhan penggalangan dana."
cara cair dana	"Dana bisa dicairkan kapan saja sesuai kebutuhan program yang sedang berjalan. Pencairan dana program dapat dilakukan dengan langkah-langkah berikut:..."
biaya galang dana	"Tidak ada biaya yang harus dikeluarkan oleh user pada saat penggalangan dana"

Adapun beberapa metode yang berkaitan dengan pembangunan sistem pada penelitian ini seperti:

#### a. *Text Mining*

*Text Mining* dapat diartikan sebagai proses ekstraksi informasi dari kumpulan dokumen. Di sisi lain, *text mining* berguna dalam bahasa alami untuk sentralisasi teks (Feldman & Sanger, 2006). Proses dalam *text mining* seperti yang ditunjukkan Gambar 2 berikut.



Gambar 2 Proses *Text Mining*

Berdasarkan pada gambar 1, proses *text mining* diawali dengan kumpulan dokumen yang terstruktur maupun tidak terstruktur. Kemudian dilakukan pengolahan data dengan *pre-processing*. Proses ini berguna dalam meminimalisir kata-kata tidak penting untuk dapat diproses lebih lanjut seperti tidak memiliki signifikansi makna, serta

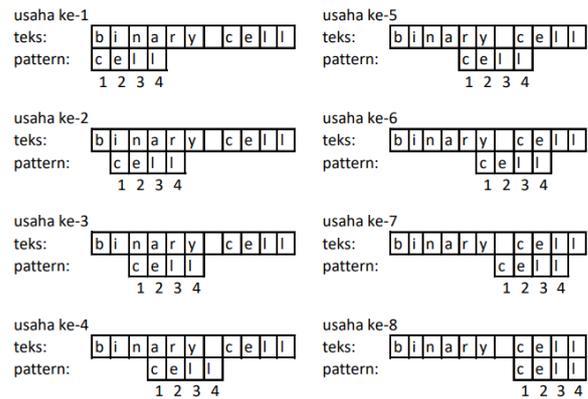
untuk mengatasi redundansi dan pengulangan (Ayu et al., 2022). Lingkup kerja *preprocessing* di antaranya *case folding*, *tokenizing*, *stemming*, dan *filtering*. *Case folding* yaitu perubahan menjadi *lowercase* dan penghilangan karakter. *Tokenizing* yaitu menguraikan kalimat menjadi kata-kata. *Stemming* yaitu transformasi kata yang ada menjadi kata dasar. *Filtering* yaitu penyaringan kata, dilakukan dengan dua cara yaitu *stopword* dan *wordlist*. *Stopword* yakni proses peniadaan kata yang tidak bermakna, sedangkan *wordlist* yakni proses penyimpanan kata yang bermakna (Novendri et al., 2020). Sedangkan tahap analisis teks menyesuaikan teknik *text mining* yang dibutuhkan, beberapa di antaranya *Information Extraction (IE)*, *Categorization*, *Clustering*, *Visualization*, dan *Summarization*, *Topic Tracking*, *Concept Linkage*, dan *Question Answering* (VijayGaikwad et al., 2014)(Vom Brocke et al., 2016).

b. Peter Norvig

*Peter Norvig* merupakan salah satu metode dalam *spelling correction* yang dibuat oleh *Director of Research Google*. *Peter Norvig* bekerja dengan beberapa proses seperti *split* yaitu menguraikan tiap huruf pada kata yang keliru, *deletion* yaitu penghilangan satu huruf, *transposition* yaitu pergantian huruf yang berdampingan, *an alteration* yaitu mengganti, dan *an insertion* yaitu mengimbuahkan satu huruf (Mutammimah et al., 2017). Namun kekurangan dari metode ini adalah tidak dapat bekerja maksimal apabila terdapat kekeliruan pada dua huruf atau lebih, serta perlu adanya korpus yang lebih lengkap (Salinka Simanjuntak et al., 2018).

c. Algoritma Brute Force

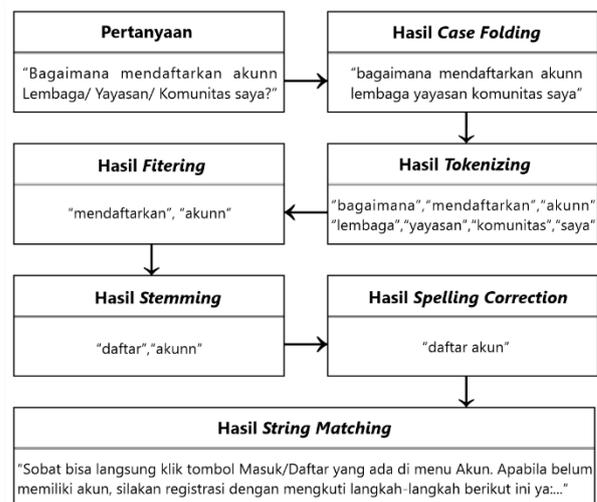
Algoritma *Brute Force* yakni penyelesaian masalah secara sederhana dengan *straight forward* yang didasarkan pada *problem statement* (Gunawan et al., 2018). Proses kerja algoritma *Brute Force* seperti yang terlihat pada gambar 2.



Gambar 3. Proses kerja algoritma *Brute Force*

Pada umumnya proses kerja algoritma *Brute Force* seperti yang terlihat pada gambar 2 di atas dilakukan dengan pencocokan satu demi satu karakter pada *pattern* dan teks, yang dimulai dari kiri lalu bergeser satu karakter *pattern* ke kanan. Pencocokan akan berhenti dilakukan apabila tidak terdapat kecocokan antarkarakter, menemukan kesamaan antarkarakter, atau pencocokan telah berada di akhir teks (Rismayani et al., 2021).

Studi kasus yang terjadi apabila sistem *chatbot* menerima masukan pengguna berupa pertanyaan seputar DIY seperti yang terlihat pada gambar 4 berikut.



Gambar 4 Studi Kasus LazismuBot

Berdasarkan studi kasus yang terlihat pada gambar 4, maka hal pertama yang dilakukan oleh *chatbot* adalah proses *case folding* yaitu mengubah kalimat menjadi *lower case* dan menghilangkan karakter seperti tanda tanya, koma, dan garis miring. Hasil dari *case folding* tersebut akan masuk

ke proses *tokenizing* yaitu mengubah *string* menjadi *array*. Kemudian hasil dari *tokenizing* akan masuk ke proses *filtering* menggunakan cara *stopword*. Proses *filtering* yang dilakukan yaitu menghilangkan kata yang tidak sesuai atau tidak penting. Setelah dari mendapatkan hasil dari proses *filtering*, sistem akan lanjut pada proses *stemming* yaitu mengubah kata yang didapat dari hasil *filtering* menjadi kata dasar. Selanjutnya dari proses *stemming* dilanjutkan dengan proses *spelling correction* dengan *Peter Norvig*. Pada proses ini, sistem akan mengoreksi kekeliruan pada kata yang didapat dari hasil *stemming*. Jika terdapat kesalahan dalam penulisan, maka akan menghasilkan kata yang benar. Jika tidak terdapat kesalahan dalam penulisan, maka hasil dari *spelling correction* akan sama dengan hasil *stemming*. Sesudah dari proses *spelling correction*, maka hasil tersebut dicocokkan dengan *pattern* yang ada di *database* menggunakan algoritma *Brute Force*. Apabila terdapat kesamaan *pattern*, maka sistem akan memberikan tanggapan sesuai dengan *pattern* tersebut. Namun apabila tidak terdapat kesamaan *pattern*, maka sistem akan memberikan tanggapan bahwa masukan tidak diketahui.

## 2.4 Tahap Pengujian

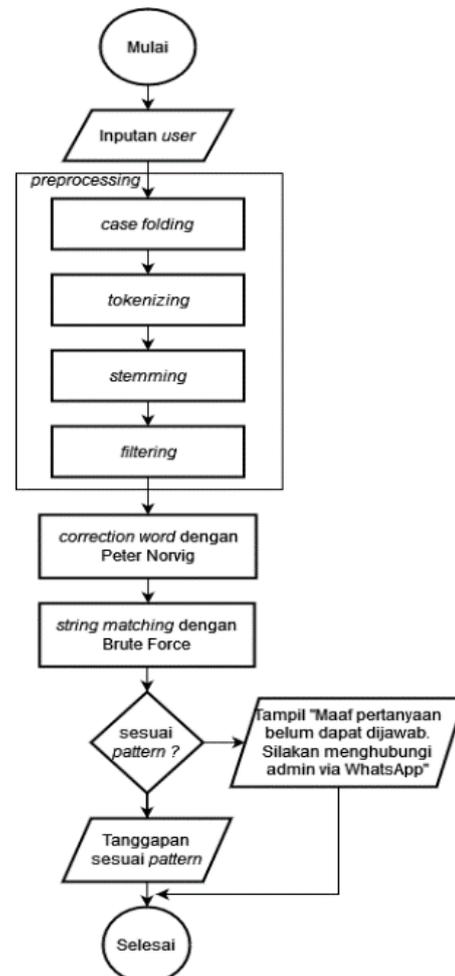
Keadaan yang harus dipenuhi dalam tahap pengujian pada *black box testing* dengan mengulas apa yang menjadi keperluan dan hal-hal yang harus ada pada pengujian (Muhammad Sidik, Bambang Gunawan, 2021). *Black box testing* merupakan proses uji pada fungsi suatu sistem dengan harapan agar sesuai dengan tujuan awal dari fungsi tersebut. Oleh karena itu, diperlukan sebuah skenario pengujian untuk menguraikan topik yang akan diujikan (Ishlakhuddin et al., 2020). Penelitian ini memiliki tujuh macam skenario pengujian. Tiap skenario pengujian memiliki tag, di mana tiap tag berisi empat pertanyaan dan satu tanggapan. Hasil dari uji validitas metode dapat dihitung dengan persamaan berikut:

$$\text{Uji Validitas} = \frac{\sum \text{jawaban tepat}}{\sum \text{pertanyaan}}$$

## 3 Hasil dan Pembahasan

Penelitian ini merancang sebuah sistem otomatis untuk melayani pertanyaan pengguna dengan tujuan memajukan kualitas dan efektifitas layanan di LAZISMU DIY. Berikut alur proses

kerja sistem yang terjadi pada penelitian ini seperti yang terlihat pada gambar 4.

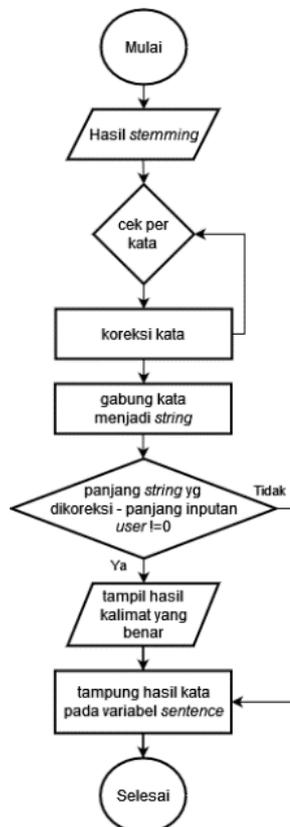


Gambar 5 Alur Utama Sistem Chabot

Gambar 5 merupakan alur utama perancangan sistem *chatbot* yang melibatkan *preprocessing* yaitu *case folding*, *stemming*, *filtering*, *tokenizing*. Lalu dilanjutkan algoritma *Peter Norvig* kemudian algoritma *Brute Force*. *Preprocessing* bertujuan untuk membersihkan kalimat dari karakter yang tidak penting dan menghasilkan kata yang dapat diproses setelah itu, dilanjutkan dengan *spelling correction* dengan algoritma *Peter Norvig* untuk mengoreksi kata yang memiliki salah dalam penulisan, lalu mencocokkan hasil yang didapat dengan *pattern* yang ada di *database* menggunakan algoritma *Brute Force*. Jika masukan *user* sama dengan *pattern*, maka sistem menampilkan tanggapan sesuai *pattern*. Jika sebaliknya, maka sistem menampilkan bahwa pertanyaan belum dapat

dijawab dan diarahkan untuk menghubungi admin melalui *WhatsApp*.

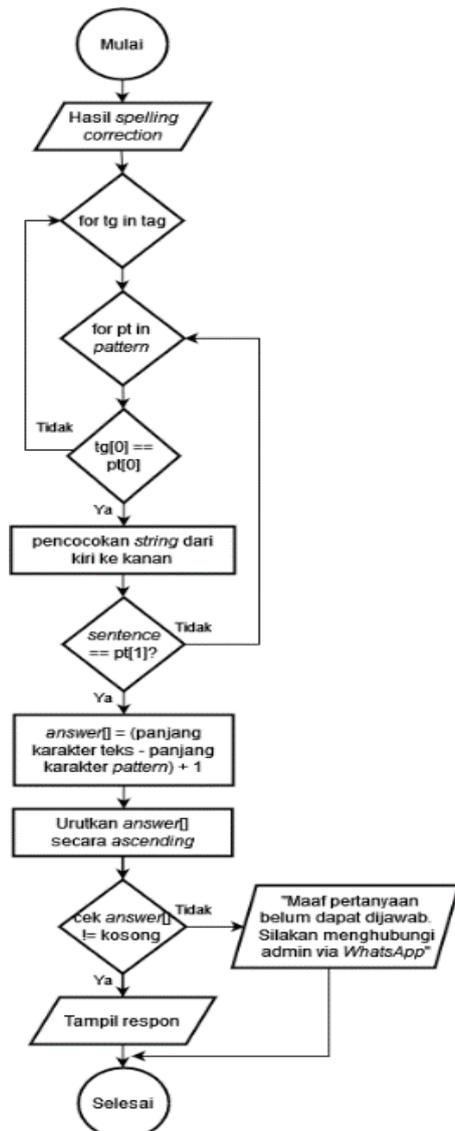
Alur proses kerja sistem dalam mengoreksi kata yang terdapat dalam pertanyaan yang diajukan pengguna seperti yang terlihat pada gambar 6 berikut.



Gambar 6 Alur Koreksi Kata

Gambar 6 merupakan proses yang terjadi setelah dilakukan *preprocessing*. Proses tersebut yaitu pengecekan kata. Hasil dari *stemming* akan dilakukan pengecekan per kata. Apabila terdapat kekeliruan pada kata, maka kata tersebut akan dikoreksi menggunakan algoritma *Peter Norvig*. Pengecekan akan terus dilakukan sampai semua kata telah dikoreksi. Setelah itu menggabungkan kata tersebut menjadi *string*. Sistem akan menampilkan kalimat yang benar apabila selisih dari panjang *string* yang telah dikoreksi dengan panjang karakter pada masukan *user* tidak sama dengan nol alias panjang karakter sama, maka kalimat tersebut langsung ditampung dalam variabel *sentence*. Namun jika memiliki selisih, maka sistem akan menampilkan hasil kalimat yang tepat.

Alur proses kerja sistem dalam pencocokan *string* seperti yang terlihat pada gambar 7 berikut.

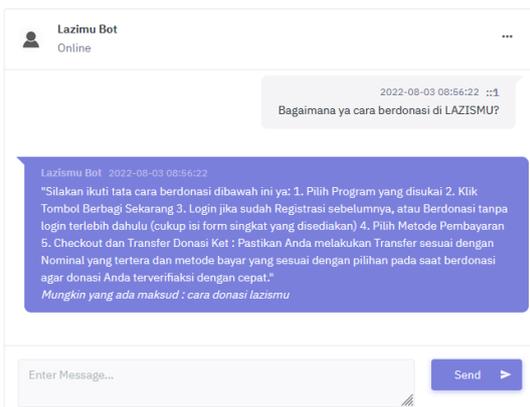


Gambar 7 Pencarian Respon Atas Pertanyaan Pengguna

Hasil dari *spelling correction* akan diproses pada pencocokan *string*. Proses tersebut dimulai dari pencarian nilai dari variabel *tg* pada data *tag*. Data *tag* berisi *id* dan *tag* dalam tabel *tag* yang ada di *database*. Lalu pencarian nilai dari variabel *pt* pada data *pattern*. Data *pattern* berisi *id\_tag* dan *pattern* dalam tabel *pattern* yang ada di *database*. *Id* dalam tabel *tag* merupakan *primary key*, sedangkan *id\_tag* dalam tabel *pattern* merupakan *foreign key*. Jika terdapat kesamaan pada nilai di variabel *tg[0]* alias *id* dan nilai di variabel *pt[0]* alias *id\_tag*, maka akan dilakukan pencocokan *string* dari kiri ke kanan menggunakan *Brute*

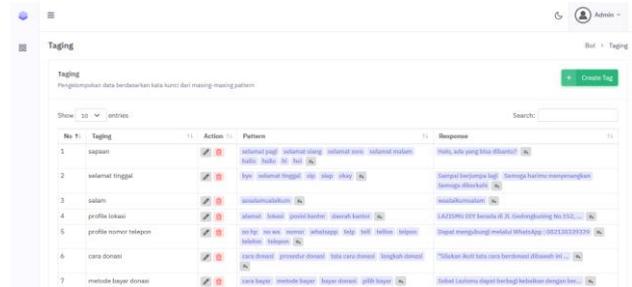
*Force*. Namun jika tidak terdapat kesamaan, maka akan kembali pada proses pencarian *id* dan *tag* dalam tabel *tag*. Pencocokan *string* dilakukan antara variabel *sentence* dengan *pattern* dalam tabel *pattern*. Jika pencocokan *string* berhasil, maka hasil tersebut akan dilakukan perhitungan yaitu panjang karakter teks dikurangi panjang karakter *pattern*, lalu dikurangi satu. Hasil perhitungan tersebut ditampilkan dalam variabel *answer* yang berupa *array*. Jika tidak terdapat kecocokan *string*, maka akan kembali pada proses pencarian nilai dari variabel *pt* sampai terdapat kesamaan nilai di variabel *tg[0]* dan *pt[0]*. Setelah dilakukan perhitungan, hasil variabel *answer* diurutkan secara *ascending* yaitu urutan nilai dari terkecil hingga terbesar. Kemudian mengecek variabel *answer[]* apakah kosong atau tidak. Jika kosong, maka sistem akan mengeluarkan output bahwa pertanyaan tidak diketahui. Jika *answer[]* tidak kosong, maka sistem akan mengeluarkan output dari nilai *answer[]* yang terbesar sebagai tanggapan atas pertanyaan yang diberikan oleh pengguna. Proses ini berguna saat pertanyaan yang diajukan pengguna mengandung dua *pattern*.

*Chabot* merupakan sebuah wadah yang berguna dalam mengatasi berbagai pertanyaan terkait lingkup tertentu yang diberikan pengguna. Pada penelitian ini, *chatbot* memiliki lingkup seputar berdonasi di LAZISMU DIY. Berikut seperti yang terlihat pada gambar 8 terkait halaman *chatbot*.



Gambar 8 Tampilan *Chabot*

Gambar 8 merupakan tampilan *chatbot* ketika pengguna menanyakan seputar cara berdonasi di LAZISMU DIY. Untuk tampilan pada halaman admin seperti terlihat pada gambar 9 berikut.



Gambar 9 Tampilan Admin

Tampilan pada sisi admin berisi *tag* (topik) dan *pattern* yang langsung dapat mengubah, menambah, dan menghapus *pattern* sesuai kebutuhan seperti pada gambar 9.

Sistem yang sudah dirancang selanjutnya dilakukan uji sistem dengan *black box testing* menggunakan skenario pengujian sistem berguna untuk mengukur tingkat kesesuaian antara hasil yang didapat dengan capaian sistem. Berikut skenario pengujian pada rancangan sistem *chatbot* seperti terlihat pada tabel 2.

Tabel 2 Skenario Pengujian

Deskripsi	Keterangan	Topik
Skenario Pengujian 1	Sapaan dan perpisahan	Salam Sapaan Perpisahan
Skenario Pengujian 2	Pertanyaan seputar LAZISMU DIY	Profil lokasi Profil nomor telepon Profil jam kerja Cara masuk dan registrasi Verifikasi akun
Skenario Pengujian 3	Pertanyaan seputar cara berdonasi	Cara donasi Donasi tanpa nama
Skenario Pengujian 4	Pertanyaan seputar pembayaran donasi	Cara bayar donasi Bayar donasi dengan transfer Bayar donasi dengan <i>virtual account</i> Bayar donasi dengan <i>gopay</i> Kode unik saat transfer Tidak ada notifikasi setelah transfer Ganti metode bayar Rekening saat transfer

Deskripsi	Keterangan	Topik
Skenario Pengujian 5	Pertanyaan seputar membuka program donasi	Daftar program donasi Ketentuan deskripsi Ubah deskripsi
Skenario Pengujian 6	Pertanyaan seputar periode donasi	Ketentuan periode Perpanjang periode
Skenario Pengujian 7	Pertanyaan seputar dana program	Cara mencairkan dana Masalah pencairan dana Potongan biaya program Biaya galang dana

Pengujian dilakukan pada 96 pertanyaan dengan jumlah 24 topik dan tiap tag memiliki 4 pertanyaan. Jumlah pertanyaan dengan tanggapan sesuai hasil yang diharapkan yaitu 89 pertanyaan dan jumlah pertanyaan tidak sesuai hasil yang diharapkan yaitu 7 pertanyaan. Perhitungan uji sistem terlihat pada perhitungan berikut.

$$Uji Sistem = \frac{100}{108} \times 100\% = 93\%$$

Hasil yang didapat pada perhitungan uji sistem *chatbot* dengan *black box testing* yaitu sebesar 93%. Setelah dilakukan uji sistem, terdapat beberapa faktor seperti yang terlihat pada tabel 3 berikut.

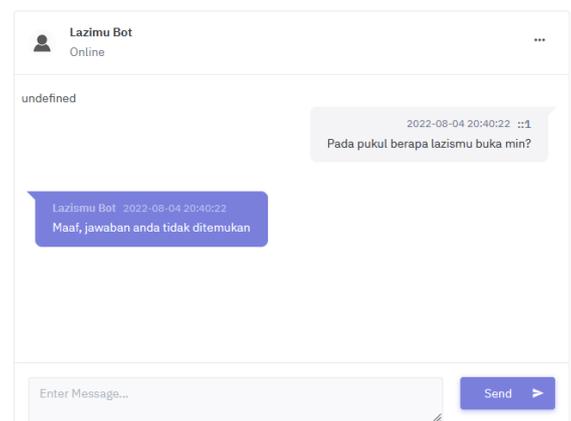
Tabel 3 Contoh Pertanyaan yang Tidak Sesuai

No	Contoh Pertanyaan yang Tidak Sesuai	
1	Tag/Topik:	profil jam kerja
	Pattern:	jam
	Pertanyaan:	Pada pukul berapa lazismu buka min?
	Output yang Diharapkan:	Jam Layanan kantor Lazismu: Senin – Jumat (09.00 – 16.00)
	Output yang Diharapkan:	Masukan tidak diketahui
2	Tag/Topik:	biaya galang dana
	Pattern:	biaya, harga
	Pertanyaan:	Berapa biaya untuk membuat program galang dana?
	Output yang Diharapkan:	Tidak ada biaya yang harus dikeluarkan oleh user pada saat penggalangan dana

	Output yang Diharapkan:	Untuk mendaftarkan program galang dana, dapat mengikuti langkah-langkah berikut: ...
3	Tag/Topik:	donasi tanpa nama
	Pattern:	tanpa nama, tidak cantum nama, tidak tampil nama, tanpa tampil nama, nama tidak lihat, tanpa cantum nama
	Pertanyaan:	Gimana cara donasi tanpa mecatumkan nama?
	Output yang Diharapkan:	Ketika donatur melakukan Donasi, pada kolom nama donatur silakan centang donasi sebagai anonim
	Output yang Diharapkan:	Silakan ikuti tata cara berdonasi dibawah ini ya:...

Tabel 3 menunjukkan beberapa contoh pertanyaan pengguna yang menghasilkan *output* tidak sesuai harapan. Faktor yang menyebabkan hal tersebut terjadi ialah:

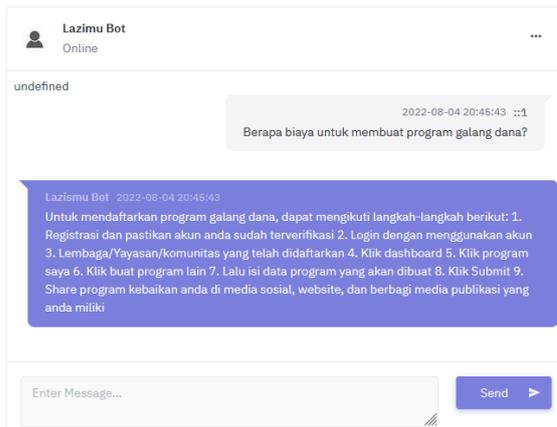
- Pertanyaan kesatu mengandung makna ganda sehingga sulit untuk dijadikan *pattern* seperti yang terlihat pada gambar 9 berikut.



Gambar 10 Pengujian Tag Profil Jam Kerja

Berdasarkan gambar 11, apabila pertanyaan tersebut dipecah menjadi kata, seperti kata 'pukul' atau 'buka', maka sulit untuk memasukkan kata tersebut menjadi *pattern* dikarenakan memiliki makna yang berbeda apabila kata dipecah, seperti kata "pukul" dapat bermakna kata kerja atau menunjukkan waktu.

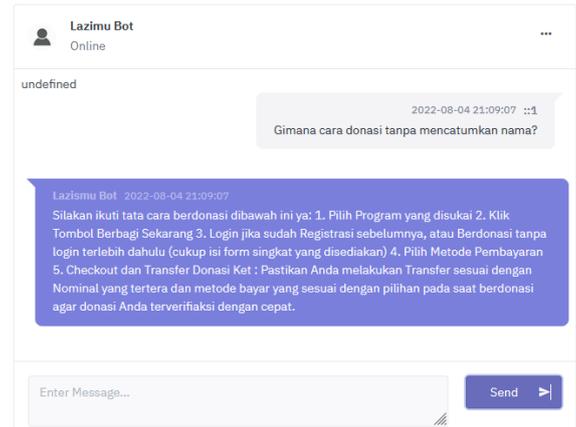
- Pertanyaan kedua mengandung dua *pattern* yang berbeda *tag* seperti yang terlihat pada gambar 11 berikut.



Gambar 12 Pengujian *Tag* Biaya Galang Dana

Berdasarkan gambar 11, pertanyaan tersebut mengandung dua *pattern* yaitu “biaya” dan “buat program”. Namun sistem menghasilkan tanggapan dari *pattern* “buat program”, padahal *output* yang diharapkan ialah tanggapan dari *pattern* “biaya”. Dalam hal ini, sistem membandingkan antara dua *pattern* tersebut yang memiliki karakter paling panjang maka tanggapan tersebut yang akan dipilih untuk menjadi *output*. Panjang karakter pada kata “biaya” sebanyak 5 karakter, sedangkan pada kata “buat program” sebanyak 11 karakter. Dengan demikian, sistem *chatbot* memilih tanggapan dari *pattern* “buat program” untuk menjadi *output chatbot* dikarenakan karakter pada kata tersebut lebih panjang daripada *pattern* “biaya”. Hal tersebut dikarenakan sistem mengecek *pattern* yang memiliki kecocokan lebih banyak apabila terdapat dua *pattern*.

- c. Pertanyaan ketiga mengandung kesalahan ejaan yang tidak terdeteksi oleh *correction word* seperti yang terlihat pada gambar 12 berikut.



Gambar 13 Pengujian *Tag* Donasi Tanpa Nama

Berdasarkan gambar 12, penulisan yang tidak terdeteksi seperti pada pertanyaan keempat yaitu kata “mencatumkan”. Apabila kata tersebut dapat terdeteksi oleh *correction word*, maka tanggapan yang diberikan *chatbot* akan sesuai dengan *output* yang diharapkan karena pertanyaan tersebut mengandung kata yang sesuai dengan *pattern* yaitu “tanpa cantumkan”. Hal tersebut dikarenakan kelemahan pada metode *Peter Norvig* yaitu tidak dapat bekerja maksimal apabila terdapat kekeliruan pada dua huruf atau lebih, serta perlu adanya korpus yang lebih lengkap.

#### 4 Kesimpulan

Beberapa hal yang dapat disimpulkan dari penelitian sistem *chatbot* ini yaitu:

- a. Peneliti telah merancang dan menerapkan sistem tanya jawab secara *virtual* berupa *chatbot* dengan mengimplementasikan pendekatan *Natural Language Processing* dan penalaran *text mining*. Selain itu menggunakan algoritma *Brute Force* dalam pencocokan pola dengan *database* dan *spelling correction* menggunakan algoritma *Peter Norvig*.
- b. Sistem dapat mengeluarkan jawaban yang sesuai apabila pertanyaan memiliki kesamaan dengan *pattern* yang ada di *database*
- c. Hasil pengujian dengan *black box testing* telah didapat sebesar 92%

Adapun beberapa masukan untuk pengembangan sistem *chatbot* selanjutnya yaitu diharapkan sistem dapat dirancang lebih dinamis

pada fungsi *tag* agar dapat melakukan klasifikasi secara otomatis pada tiap pertanyaan. pengguna Selain itu gunakan metode *spelling correction* yang lebih baik untuk meminimalkan kesalahan dalam mendeteksi ejaan apabila kata tersebut memiliki kesamaan dengan dua atau lebih *pattern* yang ada di *database*.

## Referensi

- Ayu, M. A., Irawan, E., & Mantoro, T. (2022). Text mining approaches for analyzing an Indonesian tafseer and translation of the Holy Quran. *Indonesian Journal of Electrical Engineering and Computer Science*, 25(3), 1469–1480. <https://doi.org/10.11591/ijeecs.v25.i3.pp1469-1480>
- Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., & Zhou, M. (2017). Superagent: A customer service chatbot for E-commerce websites. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of System Demonstrations*, 97–102. <https://doi.org/10.18653/v1/P17-4017>
- Feldman, R., & Sanger, J. (2006). The Text Mining Handbook. In *The Text Mining Handbook*. <https://doi.org/10.1017/cbo9780511546914>
- Gunawan, C. R., Ihsan, A., & Munawir, M. (2018). Optimasi Penyelesaian Permainan Rubik's Cube Menggunakan Algoritma IDA\* dan Brute Force. *Jurnal Infomedia*, 3(1), 37–42. <https://doi.org/10.30811/jim.v3i1.627>
- Ishlakhuddin, F., Basir, A., & Nurlaela. (2020). Rancang Bangun Sistem Tanya-jawab Berbasis Aturan STMIK Muhammadiyah Paguyangan Brebes dengan Menggunakan Telegram Chatbot. *Jurnal Informatika*, 5(3), 100–105.
- Muhammad Sidik, Bambang Gunawan, D. A. (2021). Pembuatan Aplikasi Chatbot Kolektor Dengan Metode Extreme Programming Dan Strategi Forward Chaining . Manufacturing of Chatbot for Collector Application With. *Jurnal Internasional*, 8(2), 293–302. <https://doi.org/10.25126/jtiik.202184298>
- Mutammimah, Sujaini, H., & Nyoto, R. D. (2017). Peter Norvig dan Spelling Checker BK-Trees pada Kata Berbahasa Indonesia. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, 5(1), 1–5.
- Novendri, R., Callista, A. S., Pratama, D. N., & Puspita, C. E. (2020). Sentiment Analysis of YouTube Movie Trailer Comments Using Naïve Bayes. *Bulletin of Computer Science and Electrical Engineering*, 1(1), 26–32. <https://doi.org/10.25008/bcsee.v1i1.5>
- Paliwahet, I. N. S., Sukarsa, I. M., Gede, I. K., & Putra, D. (2017). Pencarian Informasi Wisata Daerah Bali menggunakan Teknologi Chatbot. 8(3), 144–153.
- Rismayani, R., Sambo Layuk, N., Wahyuni, S., Wali, H., & Marselina, N. K. (2021). Pencarian Kata Pada Aplikasi Kamus Istilah Komputer dan Informatika Menggunakan Algoritma Brute Force Berbasis Android. *Komputika : Jurnal Sistem Komputer*, 10(1), 43–52. <https://doi.org/10.34010/komputika.v10i1.3644>
- Sains, S., Fitri, N. A., & Nurvembrianty, I. (2021). Midwife Virtual Menggunakan Aplikasi Pelayanan Chatbot Polita Sebagai Media Untuk Informasi Imunisasi. *Sains Dan Teknologi Informasi (SATIN)*, 7(1), 12–21. <https://doi.org/10.33372/stn.v7i1.678>
- Salinka Simanjuntak, M., Sujaini, H., & Safriadi, N. (2018). *Spelling Corrector Bahasa Indonesia dengan Kombinasi Metode Peter Norvig dan N-Gram*. 4(1), 17–23.
- Salisah, T., Sari, B. P., Yulianto, Y., & Hartanto, A. D. (2020). Implementasi Algoritma Boyer-Moore Pada Chatbot Wisata Yogyakarta. *Technomedia Journal*, 5(1), 54–66. <https://doi.org/10.33050/tmj.v5i1.1189>
- VijayGaikwad, S., Chaugule, A., & Patil, P. (2014). Text Mining Methods and Techniques. *International Journal of Computer Applications*, 85(17), 42–45. <https://doi.org/10.5120/14937-3507>
- Vom Brocke, J., Mueller, O., & Debortoli, S. (2016). Class Notes : Power of Text-mining in BPM. *BPTrends*, 2015, 1–14.
- Wijaya, B., & Toba, H. (2020). Implementasi Sistem Tanya Jawab Berbasis Skenario untuk Mendukung Proses Akademik dengan IBM Watson Assistant. 6(2), 154–166.