

**QUESTION ANSWERING SYSTEM DALAM BENTUK CHATBOT PADA PLATFORM
LINE UNTUK MATA PELAJARAN SEJARAH SMA/MA DENGAN
MENGUNAKAN ALGORITMA LEVENSHTTEIN DISTANCE**

**QUESTION ANSWERING SYSTEM IN THE FORM OF CHATBOT ON THE LINE
PLATFORM FOR HISTORY COURSES OF SMA/MA USING THE LEVENSHTTEIN
DISTANCE ALGORITHM**

Andi Setiawan ¹, Oktariani Nurul Pratiwi ², Riska Yanu Fa'rifah ³.

^{1,2,3} Universitas Telkom, Bandung

¹andisetiawan@student.telkomuniversity.ac.id, ²onurulp@telkomuniversity.ac.id,

³riskayanu@telkomuniversity.ac.id.

Abstrak

Pemanfaatan buku mata pelajaran sejarah sebagai acuan siswa SMA/MA dalam mencari sebuah informasi sudah dianggap tidak efektif dan efisien. Untuk membantu dan memudahkan siswa SMA/MA dalam mencari informasi mengenai mata pelajaran sejarah maka dirancang dan dibangun *question answering sytem* dalam bentuk *chatbot* pada *platform* LINE. Untuk mendapatkan respon yang sesuai dengan masukan pertanyaan dari pengguna chatbot yang dibangun menggunakan algoritma *levenshtein distance*. Inputan menggunakan akan diproses dengan text preprocessing untuk mendapatkan kata kunci. Algoritma *levenshtein distance* digunakan sebagai pencocokan kata kunci berdasarkan inputan pengguna dengan kata kunci yang ada pada *knowledge base* untuk mendapatkan respon yang paling sesuai. Pengujian pada penelitian ini memperoleh hasil akurasi sebesar 100% dengan menggunakan pertanyaan yang berasal dari *knowledge base* dan 72% dengan menggunakan pertanyaan yang ditanyakan langsung oleh siswa SMA/MA. Pada penelitian ini dapat disimpulkan bahwa *chatbot* dapat membantu siswa dalam memberikan jawaban secara benar mengenai mata pelajaran sejarah.

Kata Kunci: *Question Answering System, Chatbot, LINE, Algoritma Levenshtein Distance, Mata Pelajaran Sejarah.*

Abstract

The use of history textbooks as a reference for high school/MA students in searching for information is considered ineffective and inefficient. To assist and facilitate SMA/MA students in finding information about historical subjects, a question answering system was designed and built in the form of a chatbot on the LINE platform. To get a response that is in accordance with the input of questions from chatbot users which was built using the Levenshtein distance algorithm. User input will be processed with text preprocessing to get keywords. The Levenshtein distance algorithm is used to match keywords based on user input with keywords in the knowledge base to get the most appropriate response. The test in this study obtained an accuracy of 100% by using questions from the knowledge base and 72% by using questions asked directly by high school/MA students. In this study, it can be concluded that chatbots can help students in giving correct answers regarding history subjects.

Keyword: *Question Answering System, Chatbot, LINE, Levenshtein Distance Algorithm, History Subject*

1. Pendahuluan

Pemanfaatan buku mata pelajaran sejarah sebagai acuan siswa dalam kegiatan belajar mengajar tentunya memiliki beberapa kekurangan, contohnya untuk mencari informasi yang terdapat di dalam buku memerlukan waktu yang tidak sedikit. Contoh lainnya terkadang buku memiliki ukuran yang cukup tebal dan kurang menarik sehingga membuat siswa menjadi malas membaca. Hal ini membuat siswa menjadi kurang memahami informasi dan mengetahui isi dari buku mata pelajaran sejarah. Permasalahan ini dapat diatasi dengan beberapa cara yaitu dengan siswa bertanya langsung kepada guru. Namun karena keterbatasan waktu dan informasi yang dimiliki, guru tidak dapat menjawab setiap pertanyaan yang diajukan siswa [1]. Cara lainnya adalah dengan mencari informasi secara mandiri melalui mesin pencari (*search engine*). Untuk memperoleh suatu informasi pada *search engine*, sistem

membutuhkan sebuah *query* [2]. Akan tetapi tidak semua informasi dapat dicari dengan praktis dan cepat dikarenakan hasil yang didapatkan dari *search engine* masih berupa kumpulan dokumen dan belum menyediakan hasil yang dibutuhkan secara langsung [3]. Hal ini ditentunya sangat tidak efektif dan efisien.

Untuk memudahkan siswa dalam mencari informasi pada buku mata pelajaran sejarah maka di butuhkan sebuah sistem yaitu *Question Answering System*. *Question Answering System* adalah sebuah sistem untuk mencari sebuah informasi dengan cara memasukan pertanyaan dalam bahasa natural, yaitu bahasa yang digunakan dalam percakapan sehari-hari dan memperoleh jawaban dengan cepat serta ringkas, atau bahkan terkadang disertai dengan kalimat yang cukup untuk mendukung kebenaran dari jawaban tersebut [3]. *Question Answering System* dibangun dalam bentuk *chatbot* pada *platform* LINE. Dari 90 juta pengguna LINE yang ada di Indonesia, 80 persen di antaranya adalah pengguna anak muda [4]. Banyaknya pengguna LINE yang didominasi oleh anak muda merupakan alasan dipilihnya LINE sebagai *platform* dalam membangun *Question Answering Sytem* ini dimana nantinya nantinya sistem yang dibangun akan ditujukan kepada anak muda yang berada di bangku SMA/MA.

Algoritma yang digunakan dalam membangun *Question Answering System* dalam bentuk *chatbot* adalah algoritma *levenshtein distance*. Algoritma *levenshtein distance* adalah salah satu algoritma yang digunakan untuk mengukur kesamaan antara dua *string*[5]. Algoritma ini dipilih karena algoritma *levenshtein distance* cocok untuk mengatasi kecocokan *keyword base* (kata kunci) pada *chatbot* dengan *inputan* dari pengguna [6].

2. Landasan Teori

2.1 Question and Answering System

Question and Answering System (QAS) adalah sebuah sistem yang mengizinkan pengguna untuk memasukan pertanyaan dalam bahasa natural, yaitu bahasa yang digunakan dalam percakapan sehari-hari dan memperoleh jawaban dengan cepat serta ringkas, atau bahkan terkadang disertai dengan kalimat yang cukup untuk mendukung kebenaran dari jawaban tersebut [3]. *Question and Answering System* merupakan salah satu solusi terhadap banyaknya permintaan dari pengguna untuk mendapatkan informasi dengan cepat dan akurat [7].

2.2 Chatbot

Chatbot adalah sebuah layanan yang memungkinkan seseorang dapat berinteraksi melalui antarmuka obrolan (*chatting*) dengan menggunakan suara atau teks dan bisa juga mendapat respon melalui suara atau teks [8]. Terdapat 2 jenis *chatbot* yaitu [8]:

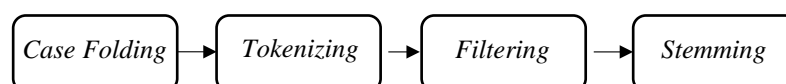
1. *Chatbot* yang dapat beroperasi berdasarkan sekumpulan aturan yang sudah dibuat sebelumnya. *Chatbot* dapat merespon perintah yang spesifik. Jika ada seseorang melakukan perintah yang tidak ada dalam aturan yang disediakan, maka *chatbot* tidak tau maksud dari perintah orang tersebut.
2. *Chatbot* yang menggunakan *machine learning* dan *artificial intelligence* untuk menyediakan respon yang baik.

2.3 LINE

LINE adalah sebuah *platform* komunikasi yang tidak hanya menawarkan layanan *chatting*, *voice call* dan *video call* namun merupakan *smart portal* dengan beragam solusi terkait konten, hiburan dan bisnis [9]. Dari 220 juta pengguna di seluruh dunia, terdapat 90 juta pengguna di Indonesia, dimana angka ini lebih besar dari jumlah pengguna di negaranya sendiri [10]. Dari 90 juta pengguna LINE yang ada di indonesia, 80 persen di antaranya adalah anak muda [4].

2.4 Preprocessing

Preprocessing adalah proses pengubahan bentuk data yang tidak terstruktur menjadi data yang terstruktur sesuai kebutuhan. Terdapat beberapa tahapan yang dilakukan dalam proses *preprocessing* yaitu *case folding*, *tokenizing*, *filtering*, *stemming*, *tagging* dan *analyzing* [11]. Pada penelitian ini tahap *preprocessing* yang dilakukan yaitu *Case Folding*, *Tokenizing*, *Filtering* dan *Stemming*.



Gambar 1. Tahapan proses *preprocessing*

- a. *Case Folding*
Case Folding adalah sebuah proses mengubah semua huruf menjadi huruf kecil dan menghilangkan karakter yang terdapat dalam kalimat seperti tanda '?', '!', '.', ',', ' ' [5].
- b. *Tokenizing*

Tokenizing adalah proses pemotongan string masukan berdasarkan kata-kata yang menyusunnya. Setiap kata yang teridentifikasi atau terpisahkan dengan kata lain oleh karakter spasi, sehingga proses *tokenizing* mengandalkan karakter spasi pada kalimat untuk melakukan pemisahan kata [12].

- c. *Filtering* adalah proses mengambil kata-kata penting dari hasil proses *tokenizing*. Bisa dengan menggunakan algoritma *stop list* (membuang kata yang kurang penting) atau pun *word list* (menyimpan kata yang penting). Jika dalam hasil *tokenizing* terdapat kata yang ada pada *stopword* maka kata tersebut akan dibuang sehingga kata-kata yang tersisa hanya kata yang dianggap penting [12].
- d. *Stemming*
Stemming adalah proses transformasi suatu kata ke bentuk dasarnya (*root word*) [11].

2.4 Levenshtein Distance

Algoritma *edit distance* atau yang biasa dikenal dengan algoritma *levenshtein distance* adalah salah satu algoritma yang digunakan untuk mengukur kesamaan antara dua *string* [5]. Selisih perbedaan antar *string* dapat diperoleh dengan cara memeriksa kesesuaian antara string sumber dengan string target. Nilai selisih perbedaan ini dapat disebut *edit distance* atau jarak *levenshtein* [5]. Langkah-langkah dari algoritma *levenshtein distance* sebagai berikut [13] :

Misalkan $S = \text{String Awal}$, dan $T = \text{String Target}$

Langkah 1 : Inisialisasi

- 1) Hitung Panjang S dan T , misalkan m dan n
- 2) Buat matriks berukuran $0..m$ baris dan $0..n$ kolom
- 3) Inisialisasikan baris pertama dengan $0..n$
- 4) Inisialisasi kolom pertama dengan $0..m$

Langkah 2 : Proses

- 1) Periksa $S[i]$ untuk $1 < i < n$
- 2) Periksa $T[j]$ untuk $1 < j < m$
- 3) Jika $S[i] = T[j]$, maka entrinya adalah nilai yang terletak pada diagonal atas sebelah kiri, yaitu $d[i, j] = d[i - 1, j - 1]$
- 4) Jika $S[i] \neq T[j]$, maka entrinya adalah $d[i, j]$ minimum dari :
 Nilai yang terletak tepat di atasnya ditambah satu, yaitu :
 $d[i - 1, j] + 1$
 Nilai yang terletak tepat dikirinya ditambah satu, yaitu :
 $d[i, j - 1] + 1$

Nilai yang terletak tepat pada diagonal atas sebelah kirinya ditambah satu yaitu:

$$d[i - 1, j - 1] + 1$$

Langkah 3 : Hasil entri matriks pada baris ke- i dan kolom ke- j , yaitu $d[i, j]$

Langkah 2 diulang hingga entri $d[m, n]$ ditemukan.

		M	O	N	D	A	Y
	0	1	2	3	4	5	6
S	1	1	1	3	4	5	6
U	2	2	2	3	4	5	6
N	3	3	3	2	3	4	5
D	4	4	4	3	2	3	4
A	5	5	5	4	3	2	3
Y	6	6	6	5	4	3	2

Gambar 2. Representasi jarak *Levenshtein Distance*

Gambar 2 adalah representasi contoh perhitungan antara dua kata yaitu *string* awal (S) = "SUNDAY" dan *string* target (T) = "MONDAY" dengan pendekatan pemrograman dinamis. Pada gambar 2. 1 menunjukkan jarak antara kedua kata yaitu 2 sebagaimana nilai terakhir pada sel kanan bawah.

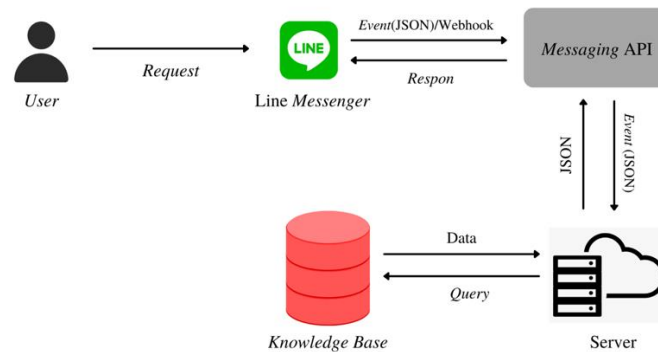
2.5 Mata Pelajaran Sejarah

Sebagai sebuah mata pelajaran di sekolah, sejarah tidak lagi terpisah dari nilai-nilai dan peneladanan dari tokoh-tokoh sebuah bangsa dan negara yang diharapkan akan diteruskan oleh para generasi berikutnya[14]. Terdapat dua tujuan penting dari pendidikan sejarah, yang pertama yaitu sebagai media untuk memperkenalkan nilai-nilai bangsa yang terus bertahan, berubah dan menjadi milik bangsa masa kini dan tujuan yang kedua adalah sebagai wahana pendidikan untuk mengembangkan disiplin ilmu sejarah [14].

3. Hasil dan Pembahasan

3.1 Perancangan Sistem

3.1.1 Arsitektur Aplikasi



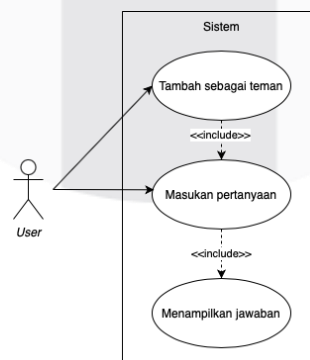
Gambar 3. Arsitektur Aplikasi

Beberapa komponen penting yang ada pada Gambar 3 adalah sebagai berikut :

1. LINE Messenger sebagai *Chat Channel Service* yang berfungsi sebagai *user* berinteraksi dengan *bot* untuk mencari informasi mengenai mata pelajaran sejarah.
2. DB Browser for SQLite sebagai *knowledge base* pada sistem *chatbot* yang berfungsi untuk menyimpan data berisikan jawaban mengenai mata pelajaran sejarah beserta kata kunci.
3. Server menggunakan *Cloud Platform Service Heroku* yang merupakan *Platform As a Service (PaaS)* yang berfungsi sebagai *web service* untuk mengelola informasi dan *knowledge base*.

3.1.2 Use Case

Use Case Diagram merupakan sebuah diagram yang dapat menjelaskan interaksi yang terjadi antar *user* dengan sistem. *Use Case Diagram* dari sistem yang akan dibangun seperti pada Gambar 4.

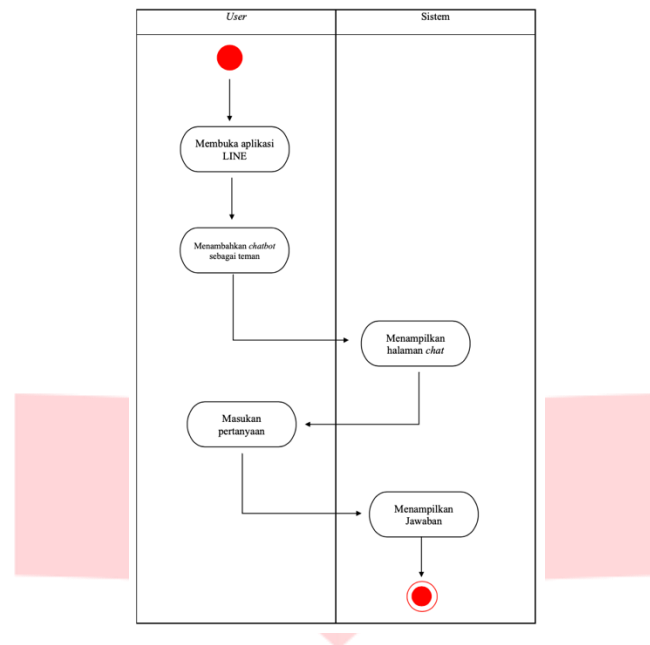


Gambar 4. Use case diagram chatbot

Pada Gambar 4 dapat dilihat bahwa *user* yang akan melakukan percakapan dengan *chatbot*. Pertama, *user* harus menambahkan Line *chatbot* sebagai teman agar dapat melakukan interaksi dengan chatbot. Kemudian *user* memasukan pertanyaan untuk mencari informasi. *Chatbot* akan menampilkan jawaban yang tepat kepada *user* sesuai dengan masukan pertanyaan dari *user*.

3.1.3 Activity Diagram

Activity Diagram merupakan diagram yang menggambarkan aktivitas dalam sebuah sistem yang sedang berjalan. *Activity diagram* dari sistem yang akan dibangun dapat dilihat pada Gambar 5.

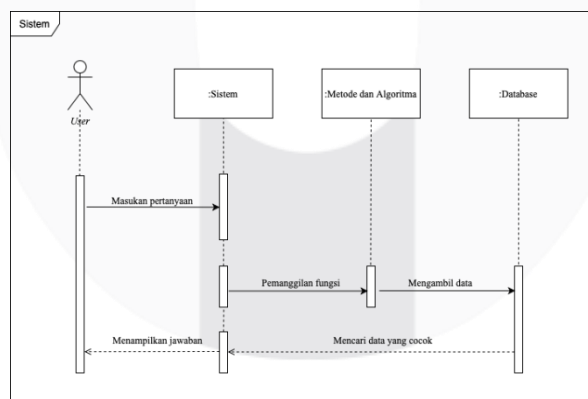


Gambar 5. *Activity diagram chatbot*

Pada Gambar 5 dapat dilihat bahwa *user* terlebih dahulu membuka aplikasi LINE. Selanjutnya *user* menambahkan *chatbot* sebagai teman. Kemudian sistem akan menampilkan halaman *chat* untuk melakukan proses tanya jawab pada *chatbot*. Pada tampilan *chat* di aplikasi LINE, *user* dapat memasukan pertanyaan yang kemudian akan ditampilkan jawaban oleh sistem berdasarkan masukan pertanyaan dari *user*.

3.1.4 Sequence Diagram

Sequence Diagram adalah suatu diagram yang menggambarkan interaksi antar objek pada sistem dalam sebuah rangkaian waktu. *Sequence diagram* dari sistem yang akan dibangun dapat ditunjukkan pada Gambar 6.

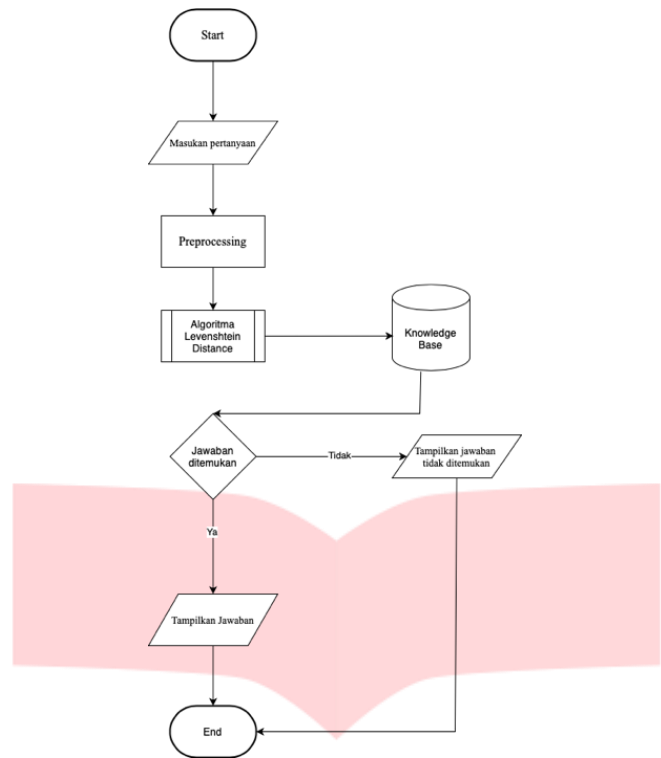


Gambar 6. *Sequence diagram chatbot*

Pada Gambar 6 langkah pertama *user* memasukan pertanyaan, langkah kedua sistem memanggil fungsi metode dan algoritma, langkah ketiga metode dan algoritma mengambil data pada database untuk melakukan pencocokan, langkah keempat *database* mencari data yang cocok untuk dikirim kesistem, langkah kelima sistem menampilkan jawaban sesuai dengan masukan pertanyaan kepada *user*.

3.1.5 Flowchart Sistem

Flowchart adalah bagan dengan simbol tertentu yang menggambarkan suatu proses dalam suatu sistem. Berikut merupakan *flowchart* pada *chatbot*. Yang dapat dilihat pada Gambar 7.

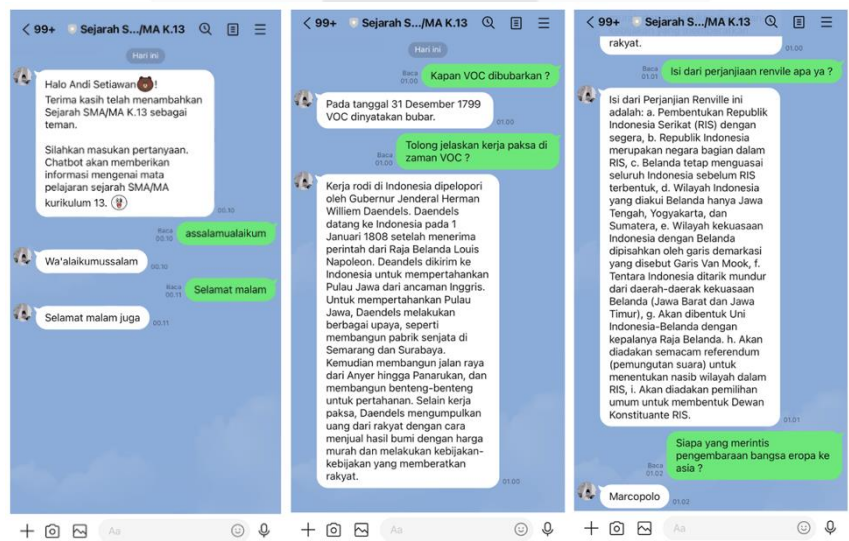


Gambar 7. Flowchart chatbot

Gambar 7 Menggambarkan alur sistem secara umum pada chatbot, dimana pertama kali user memasukan pertanyaan untuk berinteraksi dengan chatbot, Kemudian masukan pertanyaan dilakukan proses preprocessing, selanjutnya hasil preprocessing berupa keyword dilakukan pencocokan dengan knowledge base menggunakan algoritma levenshtein distance. Kemudian akan dicari jarak dengan nilai terkecil. Jika ditemukan kecocokan maka sistem akan menampilkan jawaban dan jika tidak cocok maka chatbot akan menampilkan jawaban tidak ditemukan. Dari data hasil preprocessing diolah untuk mendapatkan nilai distance di setiap row yang ada pada knowledge base dan kemudian di cari nilai mana yang paling kecil kemudian akan ditampilkan sebagai respon jawaban. String awal (S) adalah masukan pertanyaan dari pengguna yang telah diolah melalui tahap preprocessing dan string target (T) adalah data pada kolom keyword yang ada pada tabel di database.

3.2 Implementasi Sistem

Berdasarkan analisa dan perancangan yang telah dilakukan hasil aplikasi yang dibangun dapat terlihat pada Gambar 8.



Gambar 8. Interaksi user dan chatbot pada aplikasi LINE

Pada Gambar 8 menunjukkan interaksi antara sistem dengan *chatbot* dimulai dengan *greeting message* yang ditampilkan *chatbot* ketika pertama kali *user* menambahkan *chatbot* sebagai teman. Kemudian *user* melakukan percakapan dengan *chatbot* dengan memasukkan kata sapaan dan *chatbot* membalas kata sapaan tersebut. Interaksi antara *user* dan *chatbot* dilanjutkan dengan tanya jawab mengenai mata pelajaran sejarah. *Chatbot* akan menampilkan respon jawaban terdekat sesuai dengan masukan pertanyaan *user*.

3.3 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan sistem yang dibangun dapat berjalan dengan baik sesuai dengan analisa dan perancangan sistem. Pengujian sistem ini terletak pada proses pencarian jawaban atas inputan pertanyaan dari pengguna sesuai dengan batasan masalah. Pengujian ini dilakukan dengan membuat skenario *test*. Skenario yang dilakukan terbagi menjadi 2 yaitu bertanya kepada *chatbot* dengan menggunakan pertanyaan yang berasal dari *knowledge base* dan bertanya menggunakan pertanyaan sembarang. Berikut ini merupakan pengujian bagaimana sistem berjalan pada *chatbot* di aplikasi LINE.

a. Bertanya menggunakan pertanyaan yang berasal dari knowledge base

Dari keseluruhan percobaan dengan menggunakan pertanyaan yang berasal dari *knowledge base* sistem menjawab dengan benar benar 100%. Untuk mengukur tingkat akurasi sistem maka dilakukanlah perhitungan berdasarkan persamaan yang ada pada metode evaluasi sebagai berikut :

Jumlah total pertanyaan : 15

Jumlah pertanyaan yang dijawab benar : 15

$$Accuracy = \frac{15}{15} \times 100\%$$

$$Accuracy = 100\%$$

Hasil pengujian menunjukkan bahwa tingkat akurasi yang dimiliki *chatbot* ketika menjawab pertanyaan yang berasal dari *knowledge base* adalah 100%.

b. Bertanya menggunakan pertanyaan yang ditanyakan langsung oleh siswa SMA/MA

Pengujian dilakukan dengan menggunakan pertanyaan yang ditanyakan langsung oleh beberapa siswa SMA/MA. Pertanyaan yang ditanyakan siswa SMA/MA berdasarkan materi yang ada pada buku mata pelajaran sejarah yang digunakan.

Berdasarkan hasil pengujian maka diperoleh data jawaban rekomendasi yang berasal dari pertanyaan yang ditanyakan oleh siswa SMA/MA dan dibandingkan dengan data yang ada pada *knowledge base*. Untuk mengukur tingkat akurasi sistem maka dilakukanlah perhitungan berdasarkan persamaan yang ada pada metode evaluasi sebagai berikut :

Jumlah total pertanyaan : 25

Jumlah pertanyaan yang dijawab benar : 18

$$Accuracy = \frac{18}{25} \times 100\%$$

$$Accuracy = 72\%$$

Hasil pengujian menunjukkan bahwa tingkat akurasi yang dimiliki *chatbot* ketika menjawab pertanyaan yang ditanyakan langsung oleh siswa adalah 72%.

Berdasarkan hasil pengujian yang dilakukan membuktikan bahwa implementasi *question answering system* dalam bentuk *chatbot* dengan menggunakan algoritma *levenshtein distance* dapat membantu dan memudahkan siswa SMA/MA dalam mencari informasi mengenai mata pelajaran sejarah. Tetapi, perbandingan hasil persentase sebesar 100% dengan menggunakan pertanyaan yang berasal dari *knowledge base* dan 72% dengan menggunakan pertanyaan yang ditanyakan langsung oleh siswa SMA/MA juga membuktikan perlunya penyempurnaan pada *chatbot* ini. Kelengkapan *keyword* dan jawaban yang ada pada *knowledge base* sangat berpengaruh pada keberhasilan *chatbot* ini.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Berdasarkan penelitian yang dilakukan maka dapat disimpulkan

1. Pada penelitian ini penulis berhasil merancang dan membangun *question answering system* dalam bentuk *chatbot* dengan menggunakan algoritma *levenshtein distance*. Perancangan sistem menggunakan *Unified Modelelling Language* (UML) yang terdiri dari *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram*. *Chatbot* yang dibangun menggunakan media sosial LINE sebagai *Chat Channel Service* yang berfungsi sebagai tempat *user* berinteraksi dengan *bot*. *Chatbot* yang dibangun dapat digunakan untuk membantu dan memudahkan siswa SMA/MA dalam mencari sumber informasi mengenai buku mata pelajaran sejarah.
2. Dalam memberikan jawaban secara benar, *chatbot* menggunakan *text preprocessing* dan algoritma *levenshtein distance*. *Text preprocessing* berguna untuk mengubah input pertanyaan dari user menjadi kata kunci dan algoritma *levenshtein distance* berguna sebagai pencocokan kata kunci yang berasal dari user dengan kata kunci yang ada di *knowledge base*. Dari hasil pengujian yang dilakukan *chatbot* mampu memberikan jawaban secara benar berdasarkan kata kunci yang ada.
3. *Question Answering System* dalam bentuk *chatbot* dapat diimplementasikan menggunakan algoritma *levenshtein distance*. Dari pengimplementasian *Question Answering System* tersebut didapatkan tingkat akurasi sebesar 100% dengan menggunakan pertanyaan yang berasal dari *knowledge base* dan tingkat akurasi sebesar 72% dengan menggunakan pertanyaan yang ditanyakan langsung oleh siswa SMA/MA. Performansi dari sistem dapat ditingkat dengan pengembangan dan penambahan keyword dan jawaban yang ada pada *knowledge base*.

4.2 Saran

Adapun saran yang dapat diberikan untuk pengembangan penelitian ini adalah sebagai berikut :

1. Kemampuan *chatbot* dalam menjawab setiap pertanyaan dengan benar ditentukan oleh data yang ada pada *knowledge base* sehingga untuk penelitian selanjutnya diharapkan untuk memperbanyak data *keyword* dan jawaban yang ada pada *knowledge base*.
2. Untuk pengembangan selanjutnya sistem diharapkan mampu memproses kata-kata tidak baku, gaul dan singkatan. Selanjutnya pada *text preprocessing* pada tahap *stopword removal* diharapkan mampu memproses lebih banyak lagi kata-kata yang dianggap tidak penting dalam Bahasa Indonesia sehingga penulis tidak perlu melakukan kustomisasi kata-kata yang ada pada *stoplist* secara manual.
3. Untuk pengembangan selanjutnya diharapkan sistem tidak hanya dapat menjawab pertanyaan dalam bentuk *text* saja namun bisa dalam bentuk gambar, video atau pesan suara.
4. Supaya performa *chatbot* dapat berkerja dengan maksimal maka sistem ini dapat dikembangkan dengan melakukan penambahan metode dan algoritma pencocokan *string* yang lain.

Referensi

- [1] Ferbiansyah, R. R. R. (2015). *Question Answering System Berbahasa Indonesia Menggunakan Metode Pattern Based Approach (Studi Kasus Bahan Ajar Mata Kuliah Sistem Operasi Prodi SI Sistem Informasi Universitas Airlangga)* [Universitas Airlangga]. <http://repository.unair.ac.id/id/eprint/27924>
- [2] Gunawan, & Lovina, G. (2006). Question Answering System Dan Penerapannya Pada Alkitab. *Jurnal Informatika*, 7(1), 1–9. <https://doi.org/10.9744/informatika.7.1.pp.1-9>
- [3] Adhi, B. W., Al-Faraby, S., & Adiwijaya. (2019). *Analisis Metode Rule-Based pada Question and Answering System untuk Hukum-Hukum Agama Islam*. 1–8.
- [4] Fikrie, Muhammad. 2018. "Pengguna LINE di Indonesia Capai 90 Juta, Didominasi Anak Muda" <https://kumparantech.com/kumparantech/penggunalinea-di-indonesia-capai-90-juta-didominasi-anak-muda/full>. Diakses pada 23 Desember 2020.
- [5] Sari, D. W. (2018). *Implementasi Natural Language Processing pada Chatbot Peribahasa Indonesia*. Universitas Sumatera Utara
- [6] Firdausillah, F., & Arieansyah. (2019). Implementasi Algoritma Levenshtein Distance Sebagai Chatbot Agen Pariwisata Berbasis Aplikasi LINE. *Seminar Nasional APTIKOM (SEMNASTIK)*, 1, 377–385.
- [7] Yunmar, R. A., & Wisesa, I. W. W. (2020). Pengembangan Mobile-Based Question Answering

- System Mobile-Based Question Answering System Development With Ontology Based Knowledge. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 7(4), 693–700. <https://doi.org/10.25126/jtiik.202072255>
- [8] Aan Choesni Herlingga, IGL. Putra Eka Prisma, Dedy Rahman Prehanto, D. A. D. (2020). Algoritma stemming nazief & adriani dengan metode cosine similarity untuk chatbot telegram terintegrasi dengan e-layanan. *Jurnal of Informatics and Computer Science*, 2(1), 19–26.
- [9] Wijaya, M., Junaedy, & Arfandy, H. (2017). Perancangan Chatbot Untuk Informasi Penerimaan Mahasiswa Baru Pada Stmik Kharisma Makassar. *Jurnal Ilmu Komputer*, 1, 1–11.
- [10] Chandra, Y. I., & Kosdiana. (2019). Rancang Bangun Aplikasi Chat Bot Line Menggunakan Pendekatan Agile Process Dengan Model Extreme Programming Berbasis Web (Studi Kasus Di STMIK JAKARTA STI&K). *Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK)*, 3(1), 149–160. <http://ejournal.jak-stik.ac.id/index.php/sentik/article/view/2696/2152>
- [11] Radiant Victor Imbar, Adelia, Mewati Ayub, A. R. (2014). Implementasi Cosine Similarity dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks. *Jurnal Informatika*, 10, 31–42
- [12] Maitri, A. L., & Sutopo, J. (2019). Rancangan Bangun Chatbot Sebagai Pusat Informasi Lembaga Kursus Dan Pelatihan Menggunakan Pendekatan Natural Language Processing. *Eprints.Uty.Ac.Id*, 1–9. <http://eprints.uty.ac.id/>
- [13] Ibrahim, M., Agung, A. I., & Sucipto, H. (2020). *Perancangan Sistem Cerdas Chatbot Admin Toko Online Dengan Algoritma Levenshtein Distance*. *Vcc*, 43–50.
- [14] Ulhaq, Z. (2017). Pembelajaran Sejarah Berbasis Kurikulum 2013 di SMA Kotamadya Jakarta Timur. *Jurnal Pendidikan Sejarah*, 6(2), 49–60. <https://doi.org/10.21009/jps.062.06>