

Penerapan String Matching dan Regular Expression dalam Pembuatan ChatGPT Sederhana



Oleh:

Muhammad Rifko Favian 13521075

Shelma Salsabila 13521115

Febryan Arota Hia 13521120

Dosen Pengampu: Dr. Ir. Rinaldi Munir, MT.

IF2211 - Strategi Algoritma

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

2023

DAFTAR ISI

DAFTAR ISI	2
BAB I	3
BAB II	5
2.1. Algoritma Program	5
2.1.1 Algoritma Boyer-Moore (BM)	5
2.1.2 Algoritma Knuth-Morris-Pratt (KMP)	5
2.1.3 Algoritma Regex	6
2.1.3 Algoritma Longest Common Subsequence (LCS)	6
2.2. Deskripsi Singkat Mengenai Web	6
BAB III	7
3.1. Langkah-Langkah Pemecahan Masalah	7
3.1.1 Fitur pertanyaan teks (didapat dari database)	7
3.1.2 Fitur kalkulator	7
3.1.3 Fitur tanggal	7
3.1.4 Fitur Tambah Pertanyaan dan Jawaban Pertanyaan ke Database	8
3.2. Fitur Fungsional dan Arsitektur Aplikasi Web	8
3.2.1 Fitur Fungsional	8
3.2.2 Arsitektur Aplikasi Web	
Pada tugas besar ini digunakan framework Next.js, database MongoDB, dan Vercel untuk deployment.	8
BAB IV	9
4.1 Spesifikasi Teknis Program	9
4.2 Penjelasan Tata Cara Penggunaan Program	9
4.3 Test Case Program	10
BAB V	15
5.1 Kesimpulan	15
5.2 Saran	15
5.3 Refleksi	16
BAB VI	17
BAB VII	18

BAB I

DESKRIPSI TUGAS

Dalam tugas besar ini diminta untuk membangun sebuah aplikasi ChatGPT sederhana dengan mengaplikasikan pendekatan QA yang paling sederhana tersebut. Pencarian pertanyaan yang paling mirip dengan pertanyaan yang diberikan pengguna dilakukan dengan algoritma pencocokan string Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM). Regex digunakan untuk menentukan format dari pertanyaan (akan dijelaskan lebih lanjut pada bagian fitur aplikasi). Jika tidak ada satupun pertanyaan pada database yang exact match dengan pertanyaan pengguna melalui algoritma KMP ataupun BM, maka gunakan pertanyaan termirip dengan kesamaan setidaknya 90% Apabila tidak ada pertanyaan yang kemiripannya di atas 90%, maka chatbot akan memberikan maksimum 3 pilihan pertanyaan yang paling mirip untuk dipilih oleh pengguna.

Perhitungan tingkat kemiripan dibebaskan kepada anda asalkan dijelaskan di laporan, namun disarankan menggunakan salah satu dari algoritma Hamming Distance, Levenshtein Distance, ataupun Longest Common Subsequence.

Adapun fitur - fitur yang harus ada dalam aplikasi secara lebih spesifik dapat dijelaskan sebagai berikut :

1. Fitur pertanyaan teks (didapat dari database)

Mencocokkan pertanyaan dari input pengguna ke pertanyaan di database menggunakan algoritma KMP atau BM.

2. Fitur kalkulator

Pengguna memasukkan input query berupa persamaan matematika. Contohnya adalah $2*5$ atau $5+9*(2+4)$. Operasi cukup Tambah, kurang, kali, bagi, pangkat, kurung.

3. Fitur Tanggal

Pengguna memasukkan input berupa tanggal, lalu chatbot akan merespon dengan hari apa di tanggal tersebut. Contohnya adalah 25/08/2023 maka chatbot akan menjawab dengan hari senin.

4. Tambah pertanyaan dan jawaban ke database

Pengguna dapat menambahkan pertanyaan dan jawabannya sendiri ke database dengan query contoh “Tambahkan pertanyaan xxx dengan jawaban yyy”. Menggunakan algoritma

string matching untuk mencari tahu apakah pertanyaan sudah ada. Apabila sudah, maka jawaban akan diperbaharui.

5. Hapus pertanyaan dari database

Pengguna dapat menghapus sebuah pertanyaan dari database dengan query contoh “Hapus pertanyaan xxx”. Menggunakan string algoritma string matching untuk mencari pertanyaan xxx tersebut pada database.

Klasifikasi dilakukan menggunakan regex dan terklasifikasi layaknya bahasa sehari - hari. Algoritma string matching KMP dan BM digunakan untuk klasifikasi query teks. Tersedia toggle untuk memilih algoritma KMP atau BM. Semua pemrosesan respons dilakukan pada sisi backend. Jika ada pertanyaan yang sesuai dengan fitur, maka tampilkan saja “Pertanyaan tidak dapat diproses”. Berikut adalah beberapa contoh ilustrasi sederhana untuk tiap pertanyaannya. (Note: Tidak wajib mengikuti ilustrasi ini, tampilan disamakan dengan chatGPT juga boleh) Adapun hal lain yang harus diperhatikan adalah spesifikasi yang harus dipenuhi sebagai berikut.

1. Aplikasi berbasis website dengan pembagian Frontend dan Backend yang jelas.
2. Implementasi Backend wajib menggunakan Node.js / Golang, sedangkan Frontend dibebaskan tetapi disarankan untuk menggunakan React / Next.js / Vue / Angular. Lihat referensi untuk selengkapnya.
3. Penyimpanan data wajib menggunakan basis data (MySQL / PostgreSQL / MongoDB).
4. Algoritma pencocokan string (KMP dan Boyer-Moore) dan Regex wajib diimplementasikan pada sisi Backend aplikasi.
5. Informasi yang wajib disimpan pada basis data: a. Tabel pasangan pertanyaan dan Jawaban b. Tabel history
6. Skema basis data dibebaskan asalkan mencakup setidaknya kedua informasi di atas.
7. Proses string matching pada tugas ini Tidak case sensitive.
8. Pencocokan yang dilakukan adalah dalam satu kesatuan string pertanyaan utuh (misal “Apa ibukota Filipina?”), bukan kata per kata (“apa”, “ibukota”, “Filipina”).

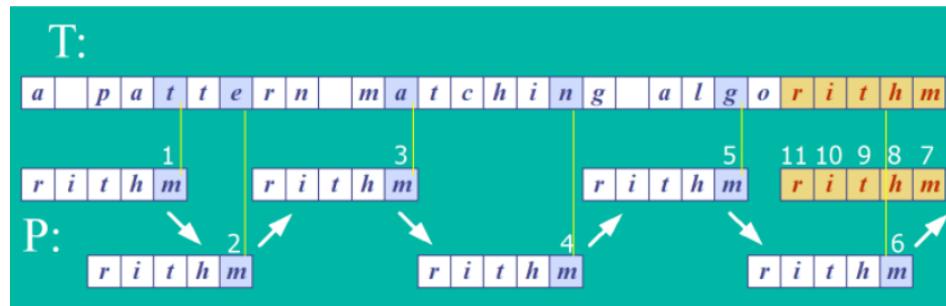
BAB II

DASAR TEORI

2.1. Algoritma Program

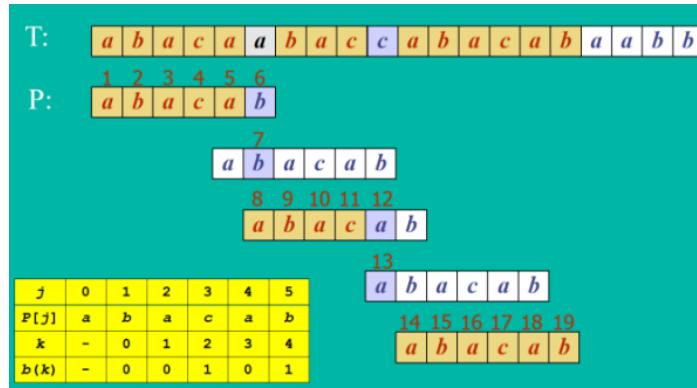
2.1.1 Algoritma Boyer-Moore (BM)

Algoritma Boyer-Moore merupakan salah satu algoritma pencarian string yang dipublikasikan pada tahun 1977 oleh Robert S. Boyer dan J. Strother Moore. Terdapat dua teknik yang diimplementasikan pada algoritma ini. Teknik pertama adalah looking-glass technique, dimana pencarian P pada T dilakukan dengan penelusuran mundur yang dimulai dari akhir teks. Selain itu, diimplementasikan juga teknik character-jump, yang akan berlaku apabila ditemukan pola yang tidak sesuai dengan teks ($P[j] \neq T[i]$). Algoritma ini dikenal sebagai algoritma yang paling efisien dan paling cocok ketika digunakan pada dokumen dengan variasi alfabet yang besar. Adapun algoritma ini secara lebih visual dapat dilihat pada gambar berikut.



2.1.2 Algoritma Knuth-Morris-Pratt (KMP)

Algoritma Knuth Morris Pratt adalah salah satu algoritma yang kerap kali digunakan untuk mencari sebuah string. Algoritma ini dikembangkan pada tahun 1966 oleh Donald E. Knuth, James H. Morris, dan Vaughan R. Pratt. Algoritma ini bekerja dengan cara mencari pola pada teks dengan urutan kiri ke kanan seperti bruteforce, namun dengan menerapkan pergeseran pola yang lebih efisien. Algoritma ini bekerja dengan cara membuat fungsi pinggiran KMP yang didefinisikan sebagai ukuran prefix terbesar dari pola $P[0 .. k]$ dan juga suffix dari $P[1 .. k]$ dan kerap disebut failure function. Fungsi ini digunakan untuk menggeser posisi suatu pola agar tidak dilakukan pergeseran satu per satu seperti pada algoritma brute-force. Algoritma ini sangat cocok untuk digunakan pada dokumen-dokumen yang besar dengan variasi alfabet yang kecil. Adapun secara visual algoritma ini dapat digambarkan sebagai berikut.



2.1.3 Algoritma Regex

Algoritma Regular Expression atau yang biasa dikenal regex merupakan teknik pencarian pola dalam string dengan menggunakan pola tertentu. Pola tersebut dapat berupa urutan string ataupun serangkaian karakter yang mendefinisikan sebuah pola pencarian. Pencocokan dengan teknik regex dilakukan dengan menyediakan string literal yang dicocokkan langsung dengan karakter yang sama atau metakarakter yang merupakan karakter khusus. Metakarakter dalam regex adalah $<(\{\{\wedge=\$\!]\})?*+\>$. Ketika menggunakan metakarakter, maka perlu digunakan backslash ‘\’.

2.1.3 Algoritma Longest Common Subsequence (LCS)

Algoritma LCS (Longest Common Subsequence) adalah sebuah algoritma yang digunakan untuk mencari subsequence terpanjang yang sama dari dua atau lebih urutan (sequence) yang berbeda. Algoritma LCS sering digunakan dalam bidang komputasi, khususnya dalam pengembangan software, dalam pemrosesan teks (text processing), dan juga dalam analisis DNA. Algoritma LCS bekerja dengan mencari urutan (sequence) terpanjang yang sama antara dua buah teks atau string (string matching), tanpa harus mencari kesamaan di posisi yang sama. Algoritma ini sangat penting karena dapat membantu menentukan hubungan antara dua buah teks atau sequence, dan juga dapat digunakan untuk membandingkan teks dalam proses pengolahan bahasa alami.

2.2. Deskripsi Singkat Mengenai Web

Aplikasi web merupakan sebuah aplikasi yang dapat diakses dengan menggunakan web melalui suatu jaringan seperti Internet atau intranet. Aplikasi web memberikan kemudahan bagi para penggunaanya karena dapat diakses hanya dengan menggunakan internet dan web browser, tanpa perlu memasang software tambahan pada perangkat. Selain itu, setup dan juga pemeliharaan aplikasi web juga cenderung mudah dan efisien. Aplikasi web ini biasanya digunakan untuk online ticketing, online payment, online

shop, platform sosial media, dan masih banyak lagi. Keunggulan dari aplikasi web ini selain mudah diakses dan dipelihara, juga lebih murah, lebih hemat penyimpanan, dan juga dapat digunakan pada multi-platform.

BAB III

Analisis Pemecahan Masalah

3.1. Langkah-Langkah Pemecahan Masalah

3.1.1 Fitur pertanyaan teks (didapat dari database)

Fitur ini mengharuskan program dapat mengeluarkan jawaban dari database ketika dituliskan sebuah query tertentu. Hal ini layaknya mesin pencarian seperti biasanya. Fitur ini menggunakan algoritma KMP dan BM. Dimana kedua algoritma ini disimpan di dalam file stringMatching.js dalam folder algorithm. Adapun jika, ketika di analisis dengan KMP ataupun BM tidak dideteksi sama maka akan digunakan algoritma LCS untuk menghitung kemiripannya, dan jika kemiripan melebihi 90% yang telah ditentukan maka akan mengirimkan tiga jawaban terdekat. Adapun secara lebih spesifik dijelaskan sebagai berikut,

1. Diinputkan query pertanyaan
2. Program akan mengecek dari query yang di database apakah pertanyaan itu ada dengan menggunakan KMP atau BM
3. Jika tidak ditemukan KMP atau BM akan return -1
4. Jika itu terjadi maka akan dihitung kemiripan dengan LCS. Adapun algoritma LCS ini juga terdapat pada file dan folder dengan BM dan KMP
5. Jika ketika dihitung dengan LCS nilainya lebih dari 90% maka, akan dikeluarkan tiga pertanyaan terdekat yang ada di database
6. Jika tidak akan ditampilkan pesan error

3.1.2 Fitur kalkulator

Fitur ini mengharuskan program dapat menghitung hasil dari inputan pengguna. Ini akan diidentifikasi dengan regex. Fitur ini diatur oleh fungsi calculator pada bagian file getAnswer.js pada folder algorithm. Fungsi ini memanfaatkan regex dalam penerapannya. Ketika sesuai dengan format yang ada maka program akan mengeluarkan hasil dari operasi karakter-karakter yang diinputkan. Jika tidak, maka akan ditampilkan pesan error.

3.1.3 Fitur tanggal

Sama halnya seperti fitur kalkulator fungsi ini memanfaatkan regex dalam implementasinya. Ketika menerima input dari pengguna, input itu diparse seperti format yang ada dalam regex yaitu dalam format day, month, dan year. Hasil parse disimpan dalam variabel day, month, dan year. Kemudian dari informasi itu dibuatlah sebuah objek date yang baru. Kemudian diperiksa apakah hasil parse sesuai jika tidak outputnya akan pesan error. Jika sesuai maka day akan diubah ke dalam bentuk string dalam bahasa indonesia yaitu string - string yang ada dalam array

dayOfWeekString. Sehingga output dari ini adalah hari dari tanggal yang diinputkan.

3.1.4 Fitur Tambah Pertanyaan dan Jawaban Pertanyaan ke Database

Fitur ini diatur oleh fungsi addQuestiontoDatabase dimana fungsi ini memiliki 4 parameter yaitu option, data, questionToAdd, dan answer. Parameter option digunakan untuk mengidentifikasi identifikasi pertanyaan apakah sudah ada di database apakah menggunakan algoritma bm atau kmp. Karena algoritma ini berjalan pertama dengan mencari terlebih dahulu apakah pertanyaan itu ada pada database atau tidak. Jika tidak maka akan dipanggil fungsi createQna lalu akan ditambahkan ke database pertanyaan itu. Namun, jika pertanyaan itu ada di database maka akan dioutputkan pesan error.

3.1.5 Fitur Hapus Pertanyaan dari Database

Fitur ini oleh diatur oleh fungsi deleteQuestiontoDatabase dimana fungsi ini memiliki 3 parameter yaitu option, data, dan questionToDelete. Option adalah algoritma pencocokan string yang ingin digunakan bisa menggunakan KMP ataupun BM. Kemudian, akan dilakukan looping dengan mencari satu persatu pertanyaan yang ada pada array data. Jika pertanyaan diidentifikasi ada maka akan memanggil fungsi deleteQna dan pertanyaan akan dihapus dari database. Jika tidak, maka akan ditampilkan pesan error

3.2. Fitur Fungsional dan Arsitektur Aplikasi Web

3.2.1 Fitur Fungsional

Pada tugas besar ini terdapat beberapa fitur fungsional, yaitu fitur utama *chat bot*, fitur opsi untuk memilih algoritma yang akan digunakan, dan fitur history *chat* yang dapat ditambah dan juga dihapus.

3.2.2 Arsitektur Aplikasi Web

Pada tugas besar ini digunakan *framework* Next.js, database MongoDB, dan Vercel untuk *deployment*.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Spesifikasi Teknis Program

A. Struktur Data

Data disimpan dengan menggunakan mongodb. Terdapat tiga tabel yang digunakan yaitu Qna, Room, dan Chat. Qna digunakan untuk menampung pertanyaan - pertanyaan yang nantinya akan diproses dan dihitung kemiripannya dengan algoritma-algoritma string matching. Room digunakan untuk menyimpan *history* suatu *chat*, dan table Chat digunakan untuk menyimpan *history* pertanyaan user dan jawaban yang diberikan *bot*.

B. Fungsi

Ada beberapa fungsi dalam pencarian jawaban yang digunakan yakni,

- kmpSearch(text,pattern) : Algoritma string matching KMP
- bmSearch(text,pattern) : Algoritma string matching BM
- computeLCS(str1, str2) : Algoritma kemiripan LCS
- getAnswer(question, option) : Algoritma utama yang memproses semua program
- addQuestontoDatabase(option, data, questionToAdd, answer) : Algoritma untuk menambah pertanyaan pada database
- createQna : Algoritma untuk membuat pertanyaan baru pada database
- deleteQna : Algoritma untuk menghapus pertanyaan yang ada pada database
- getQna : Algoritma untuk mendapatkan semua pertanyaan dan jawaban di database
- deleteQuestionDatabase(option, data, questionToDelete) : Algoritma untuk menghapus suatu pertanyaan yang diinputkan pengguna pada database
- calculator(question) : Algoritma untuk menghitung persamaan matematika sederhana ketika pengguna memberikan pertanyaan dalam bentuk operasi matematika
- date(question) : Algoritma yang meg-cover kasus ketika input pengguna dalam bentuk tanggal sehingga me-return hari dari tanggal itu
- isYearKabisat(year) : Algoritma untuk menentukan apakah year yang diinput merupakan tahun kabisat atau bukan
- countDay(day, month, year) : Implementasi dari Algoritma Zeller's Congruence

C. Prosedur yang dibangun

Awalnya pertanyaan akan di test menggunakan regex untuk menentukan pertanyaan tersebut masuk ke fitur mana. Prioritas klasifikasi fitur yang kami buat yaitu Tanggal → Kalkulator → Tambah pertanyaan → Hapus pertanyaan → String Matching Teks.

Kemudian pertanyaan akan diproses di fitur yang masuk pada salah satu klasifikasi tersebut. Apabila pertanyaan tidak ada yang cocok dengan regex, maka ia akan otomatis

masuk ke fitur string matching. Apabila pertanyaan tidak match dengan satupun yang ada di database, maka ia akan dicek dengan algoritma kemiripan LCS. Namun, apabila pertanyaan tidak ada yang mirip dengan yang ada di database, maka ia akan mengeluarkan “Pertanyaan tidak dapat diproses”.

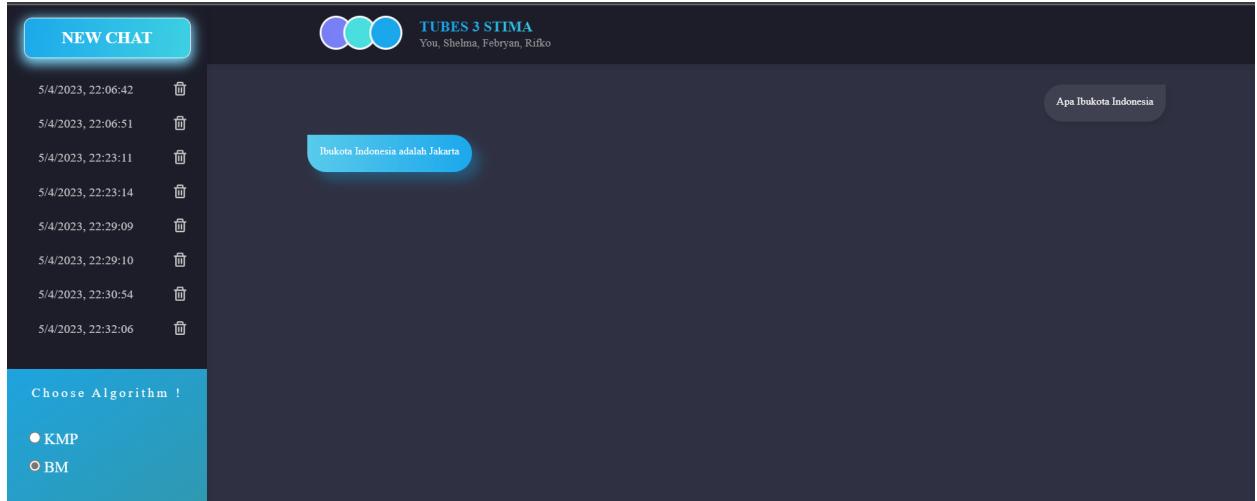
4.2 Penjelasan Tata Cara Penggunaan Program

Instalasi dan Persiapan Program

1. Clone repository dengan link github sebagai berikut.
https://github.com/fbryanarota/Tubes3_13521075.git
2. Buka folder hasil clone
3. Kemudian, *install requirements* dengan cara mengetik “npm install” kemudian pastikan halaman yang dibuka adalah https://localhost:3000
4. Setelah itu program dapat digunakan dengan mengetikkan pertanyaan sesuai fitur-fitur yang telah dijelaskan pada bagian spesifikasi.
5. Selain meng-clone repository ini, program juga dapat dijalankan pada link berikut
<https://tanya-tanya.vercel.app/>

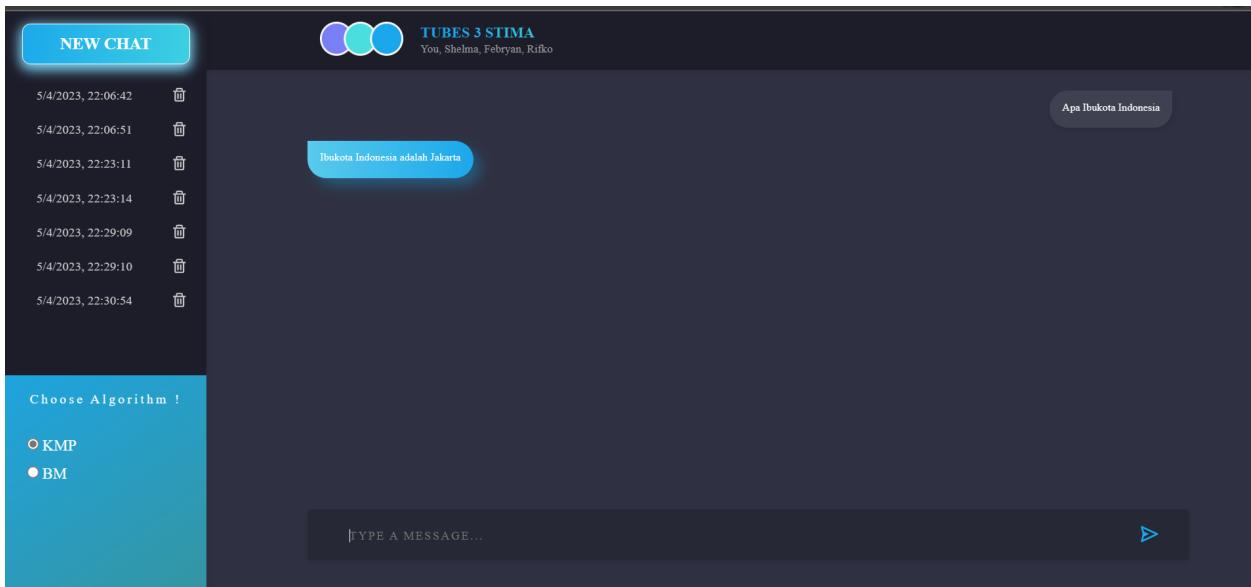
4.3 Test Case Program

1. TestCase Fitur Pernyataan teks (Didapat dari database) dengan BM



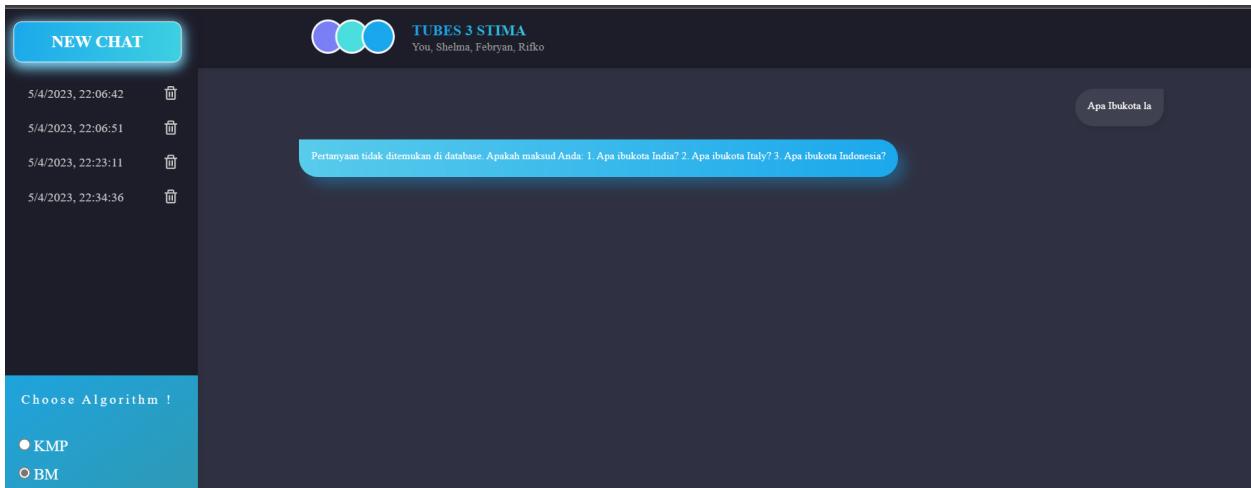
Dari test case ini kita bisa melihat bahwa program dapat mendeteksi suatu pertanyaan ada di database lalu mengeluarkan pasangan jawaban dari pertanyaan itu. Adapun kemiripan pertanyaan input user dengan database dianalisis dengan string matching BM.

2. TestCase Fitur Pernyataan teks(Didapat dari database) dengan KMP



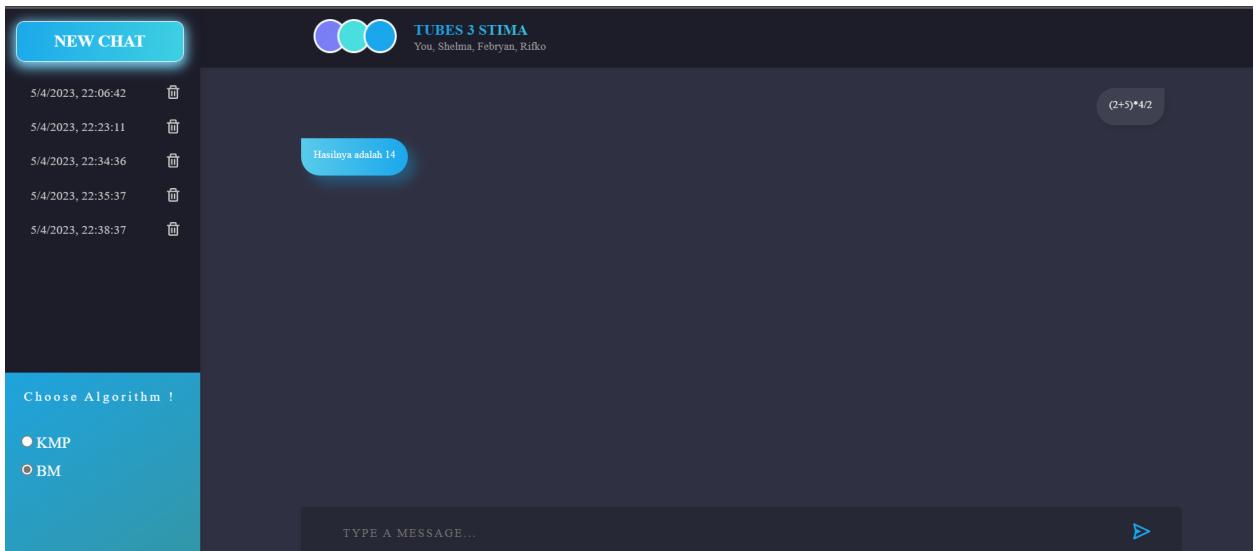
Dari test case ini kita bisa melihat bahwa program dapat mendeteksi suatu pertanyaan ada di database lalu mengeluarkan pasangan jawaban dari pertanyaan itu. Adapun kemiripan pertanyaan input user dengan database dianalisis dengan string matching KMP.

3. TestCase Fitur Pernyataan teks(Didapat dari database) yang menunjukan pemanggilan LCS



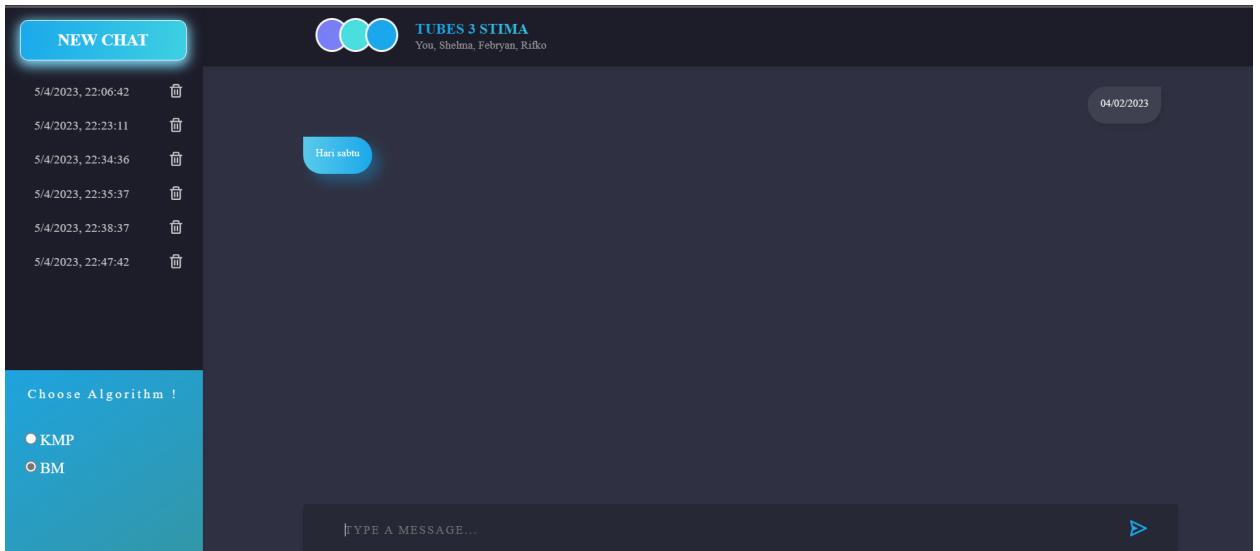
Dari test case ini kita bisa melihat bahwa program dapat mendeteksi suatu pertanyaan tidak ada di database tapi saat dihitung dengan LCS nilai kemiripannya tinggi dengan beberapa pertanyaan di database sehingga, chat bot mengeluarkan tiga pertanyaan termirip

4. TestCase Fitur Kalkulator



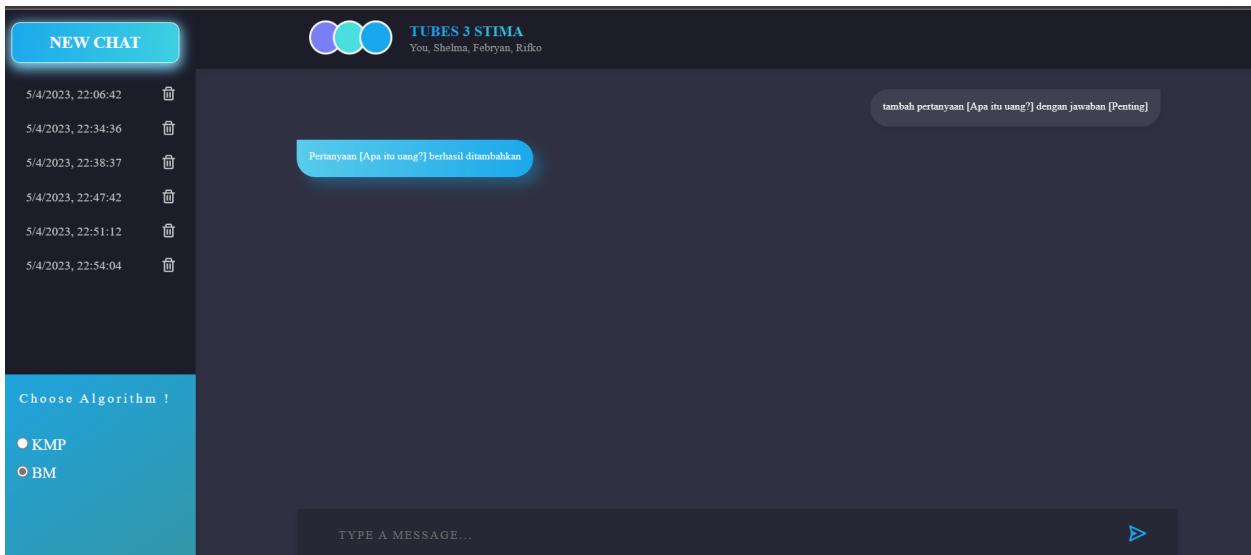
Dari test case ini kita bisa melihat bahwa program dapat mendeteksi suatu pertanyaan merupakan operasi matematika dan program dapat mengeluarkan hasil operasinya

5. TestCase Fitur Tanggal



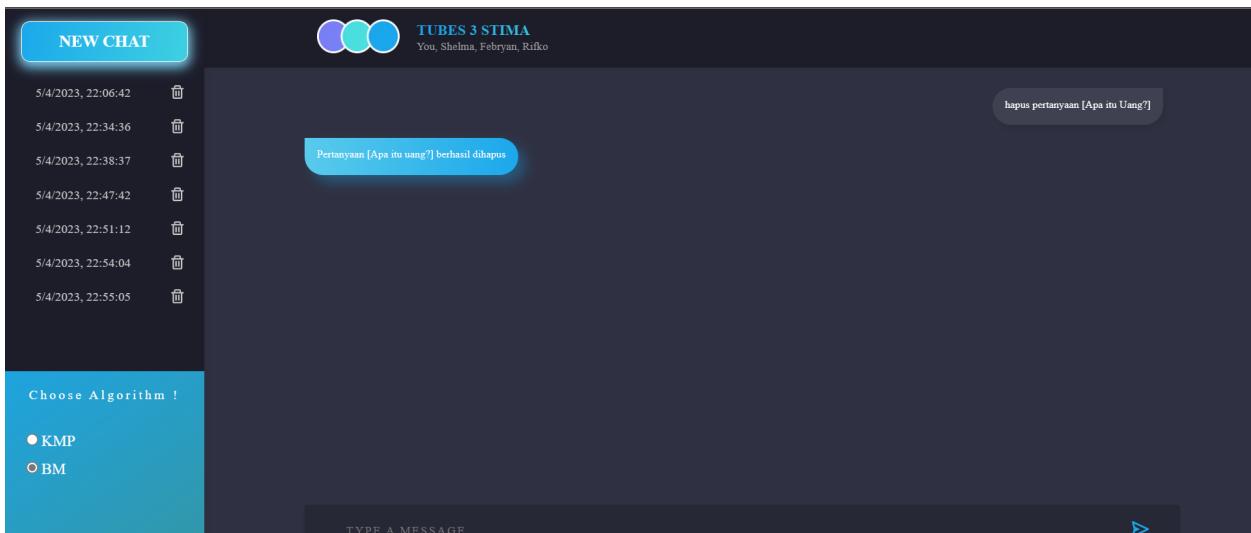
Dari test case ini kita bisa melihat bahwa program dapat mendeteksi suatu pertanyaan merupakan tanggal dan program dapat mengeluarkan hari dari tanggal yang diinputkan

6. TestCase Fitur Menambah Pertanyaan Pada Database



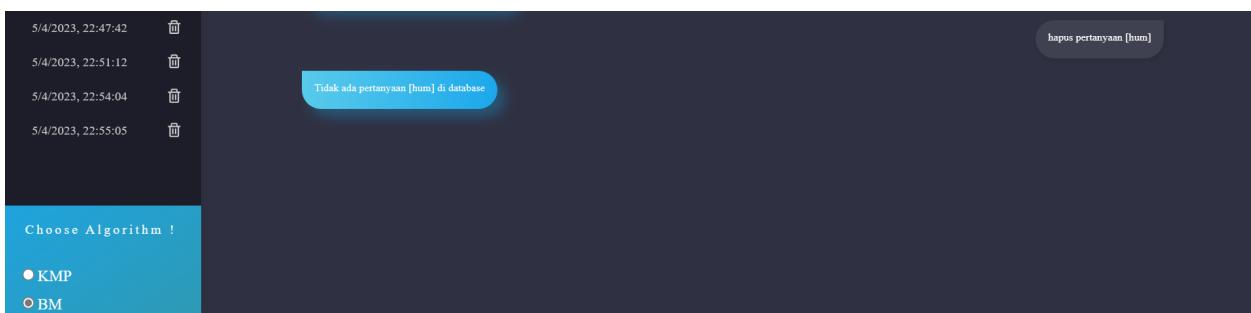
Dari test case ini dapat dilihat program dapat menambah pertanyaan ketika pertanyaan itu tidak ada di database

7. TestCase Fitur Menghapus Pertanyaan Pada Database



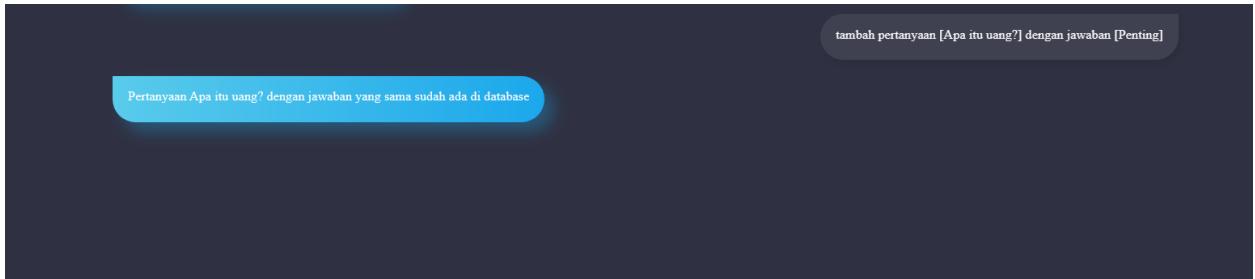
Dari test case ini dapat dilihat juga bahwa program dapat menghapus pertanyaan yang ada pada database

8. TestCase Fitur Menghapus Pertanyaan Pada Database Ketika Pertanyaan Tidak Ada di Database



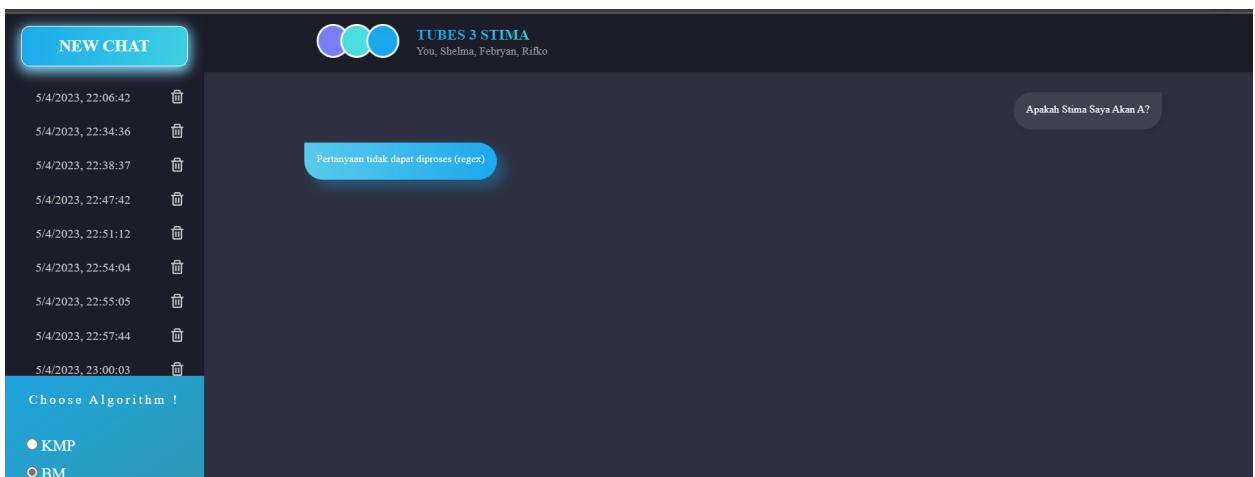
Dari test case ini dapat dilihat ketika pertanyaan tidak ada di database lalu kita meminta menghapusnya maka akan disampaikan pesan error

9. TestCase Fitur Menambah Pertanyaan Ketika Pertanyaan Ada Pada Database



Dari test case ini dapat dilihat ketika kita menambah pertanyaan namun pertanyaan itu telah ditambah sebelumnya(ada di database) maka akan dituliskan pesan error

10. TestCase Mencari Jawaban Tapi Pertanyaan Tidak ada di Database



Dari test case ketika pertanyaan tidak ada di database maka akan ditampilkan pesan error juga

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari Tugas Besar III IF2211 Strategi Algoritma Semester II 2022/2023 berjudul Penerapan String Matching dan Regular Expression dalam Pembuatan ChatGPT Sederhana, menunjukan bahwa implementasi hasil pembelajaran algoritma KMP dan Boyer-Moore telah berhasil diimplementasikan dengan menggunakan string matching antara question dan input pengguna kemudian mengeluarkan output yang ada di database. Ataupun jika menggunakan fitur tanggal atau kalkulator digunakan regex.

Dengan demikian, kelompok kami dapat menyimpulkan bahwa dalam proses pembuatan chatGPT algoritma yang digunakan tidak jauh dari string matching.

5.2 Saran

Tugas besar IF2211 Strategi Algoritma Semester 2 Tahun 2021/2022 menjadi salah satu proses pembelajaran bagi kelompok dalam menerapkan ilmu-ilmu yang dipelajari pada perkuliahan ataupun dengan melakukan eksplorasi materi secara mandiri. Berikut ini adalah saran dari kelompok untuk pihak-pihak yang ingin melakukan atau mengerjakan hal serupa.

- Dibutuhkan brainstorming yang cukup dalam mengerjakan tugas ini. Dengan melakukan brainstorming, strategi yang digunakan akan menjadi lebih efisien.
- Melakukan riset yang mendalam terkait web-app development dengan cara membaca dokumentasi dan banyak berlatih. Dengan menerapkan hal ini, pekerjaan akan selesai dengan lebih cepat.
- Lakukan analisis setiap ada pergantian masif terhadap algoritma. Hal ini dapat membantu para programmer untuk mencari letak kesalahan dan hal-hal lainnya yang sekiranya masih bisa diimplementasikan.
- Dalam mengerjakan suatu tugas secara berkelompok, penting untuk memiliki strategi serta distribusi tugas yang baik dan efisien. Cara penulisan kode dan kemampuan menulis komentar menjadi hal yang sangat penting dalam mengerjakan kode pemrograman secara berkelompok. Dengan adanya komentar pada kode, anggota lain pada kelompok dapat memahami cara kerja suatu kode dengan lebih cepat. Kemampuan tersebut juga didukung dengan adanya version control system (VCS) yang baik untuk digunakan oleh programmer dalam membuat sebuah kode pemrograman secara bersamaan. Kelompok sangat menyarankan penggunaan ‘Github’ untuk digunakan sebagai version control system (VCS) dalam pengerjaan tugas besar, maupun pada pembuatan program yang lainnya.

5.3 Refleksi

Dengan mengerjakan tugas besar kali ini, kelompok menjadi lebih paham terkait penggunaan algoritma KMP dan juga Bayer-Moore. Selain itu, kelompok juga menjadi lebih familiar dengan teknik-teknik pembuatan website.

BAB VI

LAMPIRAN

Github : https://github.com/Mifkiyan/Tubes3_13521075

Link Deploy : <https://tanya-tanya.vercel.app/>

Link Video : <https://www.youtube.com/watch?v=TTVhHxNiZ1o>

BAB VII

Daftar Pustaka

Tim Mata Kuliah IF2211 Strategi Algoritma. (2022). Spesifikasi Tugas Besar,
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/Tubes3-Stima-2023.pdf>,
diakses 20 April 2023.

Munir, Rinaldi. (2021). Pencocokan String (String/Pattern Matching).
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>,
diakses 25 April 2023.

Munir, Rinaldi. (2021). String Matching dengan Regular Expressions.
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>, diakses 25 April 2022.