

# **Laporan Tugas Besar 3 IF2211 Strategi algoritma Penerapan String Matching dan Regular Expression dalam Pembuatan ChatGPT Sederhana**



**Disusun Oleh:**  
**Kelompok 6 “ngasiNamanyaAbisLebaran”**

Fahrian Afdholi	13521031
Muhammad Equilibrie Fajria	13521047
Edia Zaki Naufal Ilman	13521141

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
SEMESTER II TAHUN 2022/2023**

# Daftar Isi

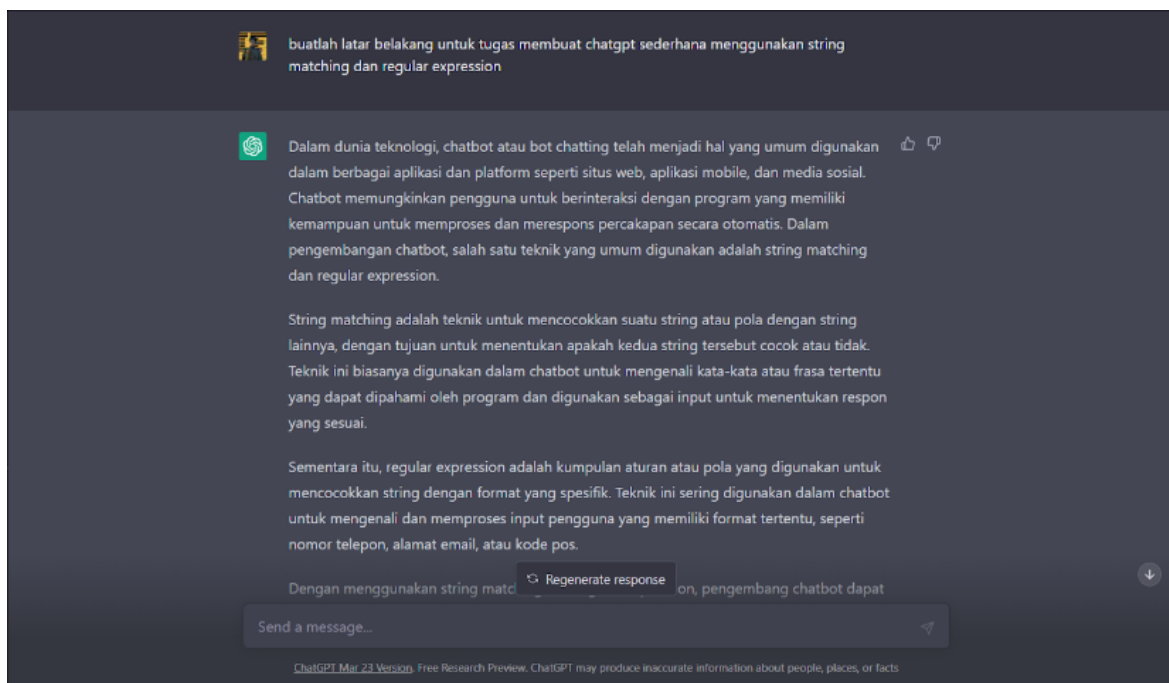
<b>Daftar Isi</b>	<b>1</b>
<b>Bab 1</b>	
<b>Deskripsi Tugas</b>	<b>2</b>
<b>Bab 2</b>	
<b>Landasan Teori</b>	<b>9</b>
2.1. Algoritma KMP	9
2.2. Algoritma BM	9
2.3. Regex	9
2.4. Deskripsi Singkat Program	10
<b>Bab 3</b>	
<b>Analisis Penyelesaian Masalah</b>	<b>11</b>
3.1. Langkah Penyelesaian Masalah Setiap Fitur	11
3.1.1. Fitur Pertanyaan Teks	11
3.1.2. Fitur Kalkulator	11
3.1.3. Fitur Tanggal	12
3.1.4. Tambah Pertanyaan dan Jawaban ke Database	12
3.1.5. Hapus Pertanyaan dari database	12
3.2. Fitur Fungsional dan Arsitektur Aplikasi Web yang Dibangun	12
<b>Bab 4</b>	
<b>Implementasi dan Pengujian</b>	<b>14</b>
4.1. Struktur Data Program	14
4.1. Frontend	14
4.2. Backend	14
4.2.1. Algoritma Fitur	14
4.2.2. Database	16
4.2. Tata Cara Menjalankan Program	17
4.3. Hasil Pengujian	17
4.4. Analisis Hasil Pengujian	21
<b>Bab 5</b>	
<b>Kesimpulan dan Saran</b>	<b>22</b>
5.1. Kesimpulan	22
5.2. Saran	22
<b>Daftar Pustaka</b>	<b>23</b>
<b>Lampiran</b>	<b>24</b>

# Bab 1

## Deskripsi Tugas

### Latar belakang:

Dalam dunia teknologi, chatbot telah menjadi hal yang umum digunakan dalam berbagai aplikasi dan platform seperti situs web, aplikasi mobile, dan media sosial. Chatbot memungkinkan pengguna untuk berinteraksi dengan program yang memiliki kemampuan untuk memproses dan merespons percakapan secara otomatis. Salah satu contoh chatbot yang sedang booming saat ini adalah **ChatGPT**.



*Gambar 1. Ilustrasi Chatbot ChatGPT (funfact latar belakang spek ini dari chatgpt)*

Sumber:

<https://chat.openai.com/chat>

Pembangunan chatbot dapat dilakukan dengan menggunakan berbagai pendekatan dari bidang Question Answering (QA). Pendekatan QA yang paling sederhana adalah menyimpan sejumlah pasangan pertanyaan dan jawaban, menentukan pertanyaan yang paling mirip dengan pertanyaan yang diberikan pengguna, dan memberikan jawabannya kepada pengguna. Untuk mencocokkan input pengguna dengan pertanyaan yang disimpan pada database, kalian bisa menggunakan string matching.

String matching adalah teknik untuk mencocokkan suatu string atau pola dengan string lainnya, dengan tujuan untuk menentukan apakah kedua string tersebut cocok atau tidak. Teknik ini biasanya digunakan dalam chatbot untuk mengenali kata-kata atau frasa tertentu yang dapat dipahami oleh program dan digunakan sebagai input untuk menentukan respon yang sesuai. Sementara itu, regular expression adalah

kumpulan aturan atau pola yang digunakan untuk pencocokan string dengan format yang spesifik. Teknik ini sering digunakan dalam chatbot untuk mengenali dan memproses input pengguna yang memiliki format tertentu, seperti nomor telepon, alamat email, atau kode pos.

### Deskripsi tugas:

Dalam tugas besar ini, anda 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.

### Fitur-Fitur Aplikasi:

ChatGPT sederhana yang anda membuat wajib dapat melakukan beberapa fitur / klasifikasi *query* seperti 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**

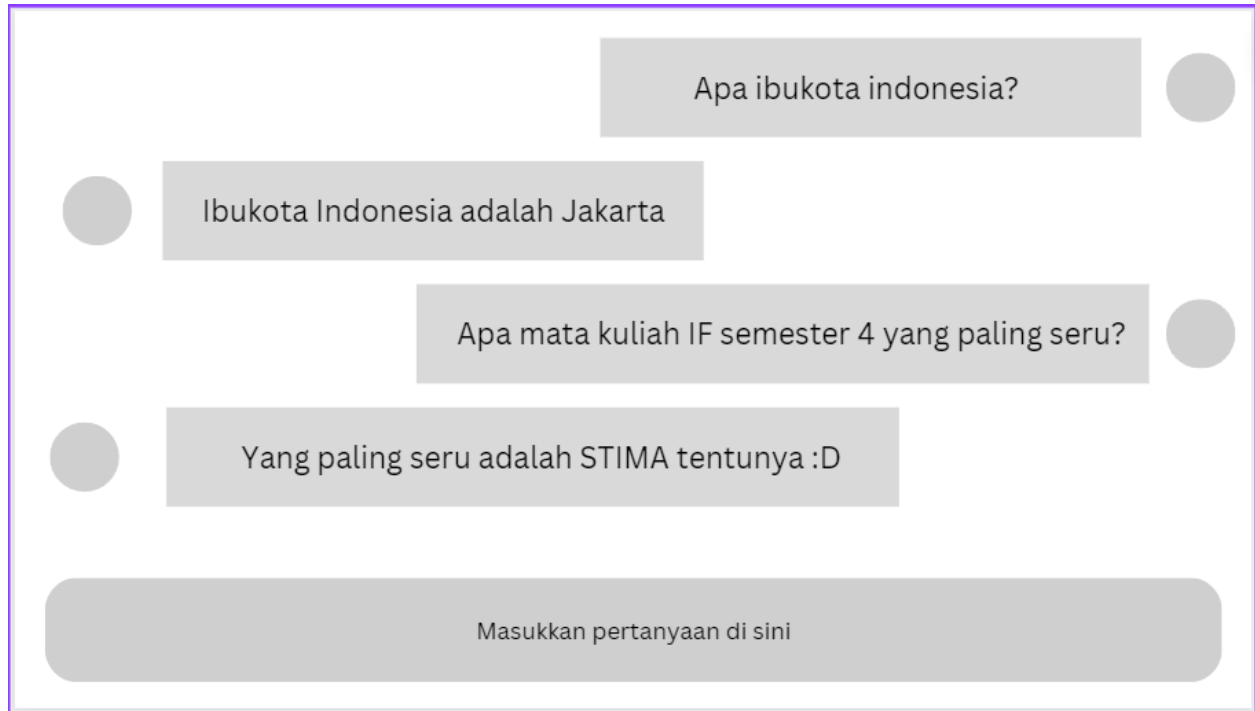
Pengguna dapat menambahkan pertanyaan dan jawabannya sendiri ke database dengan query contoh “Tambahkan pertanyaan xxx dengan jawaban yyy”. Menggunakan algoritma string matching untuk mencari tahu apakah pertanyaan sudah ada. Apabila sudah, maka jawaban akan diperbaharui.

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** pada website bagi pengguna 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.

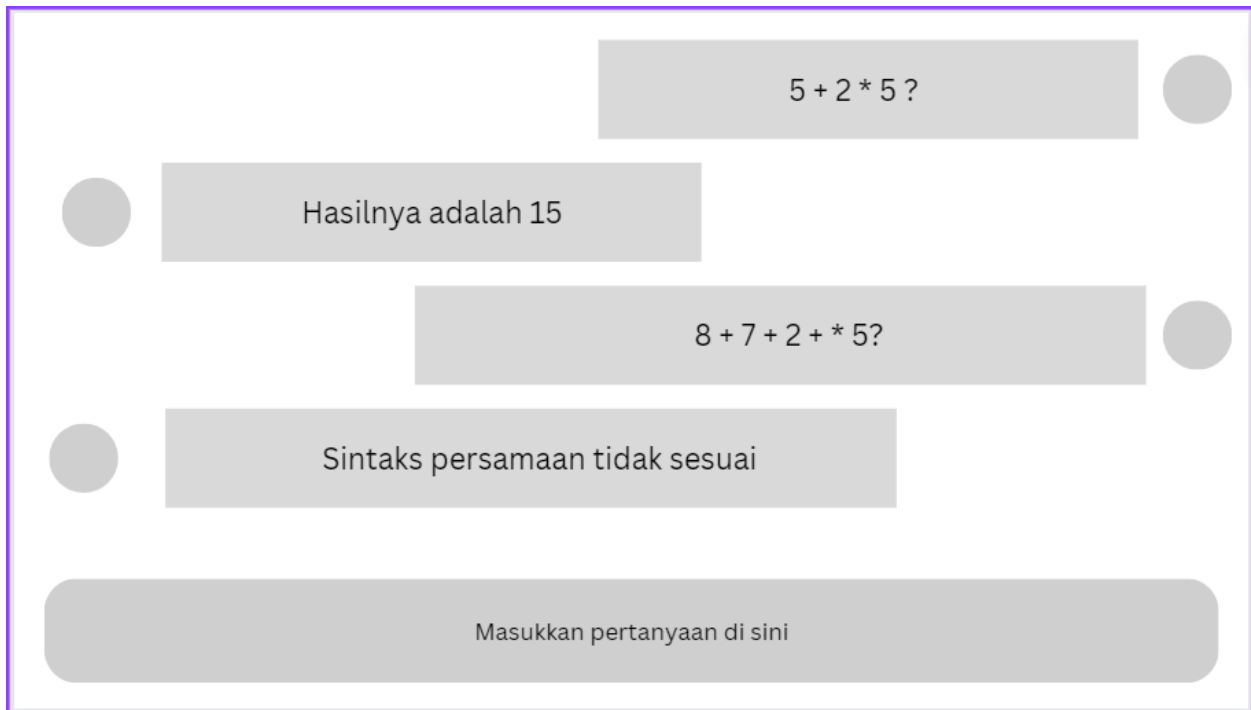
(**Note:** Tidak wajib mengikuti ilustrasi ini, tampilan disamakan dengan chatGPT juga boleh)



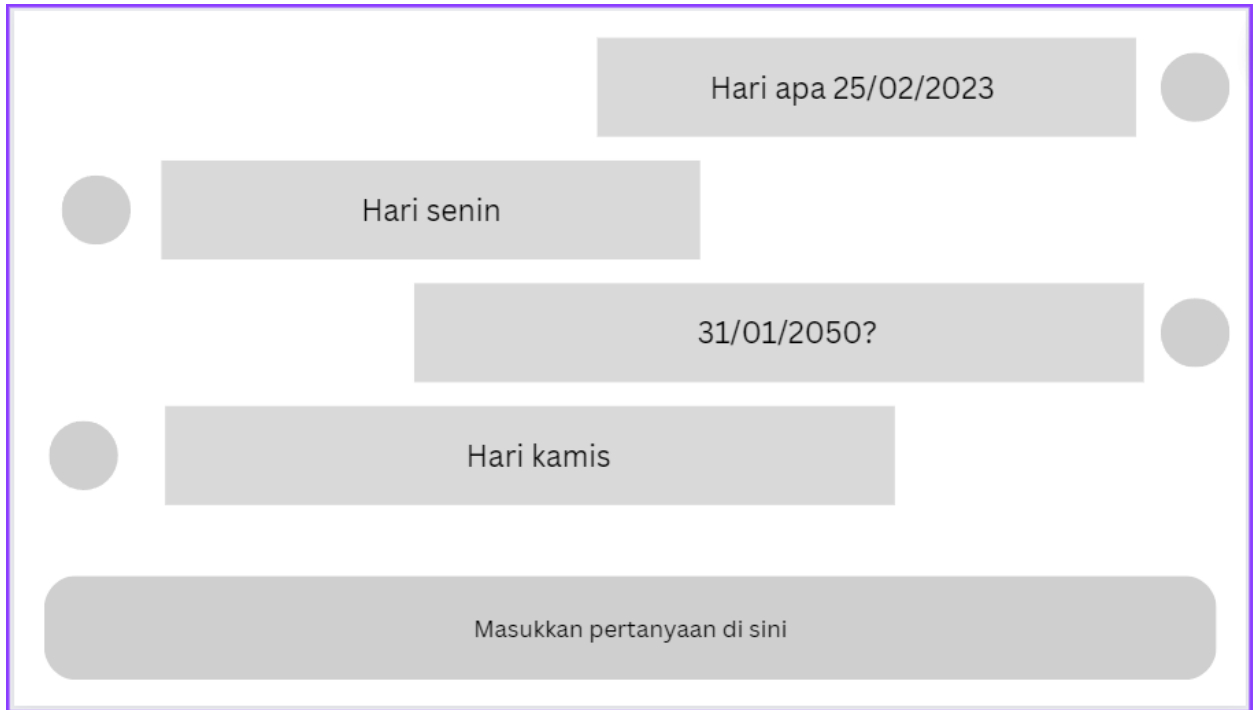
*Gambar 2. Ilustrasi Fitur Pertanyaan teks kasus exact*



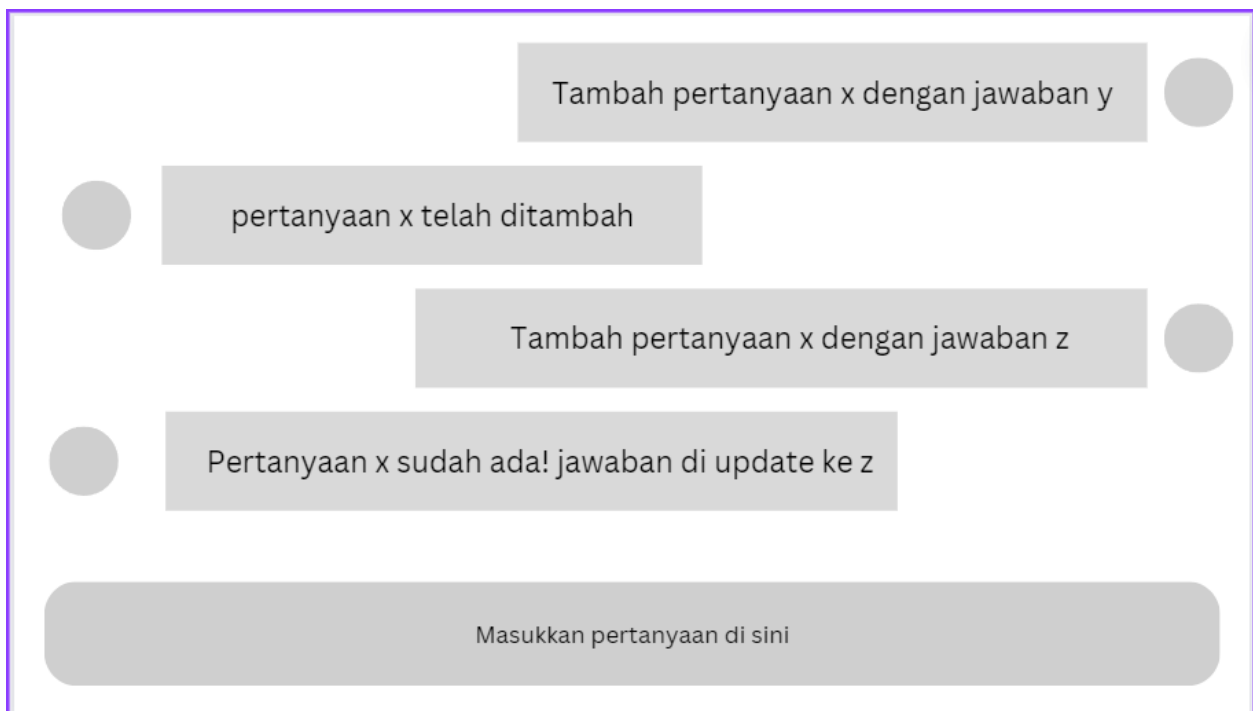
Gambar 3. Ilustrasi Fitur Pertanyaan teks kasus tidak exact



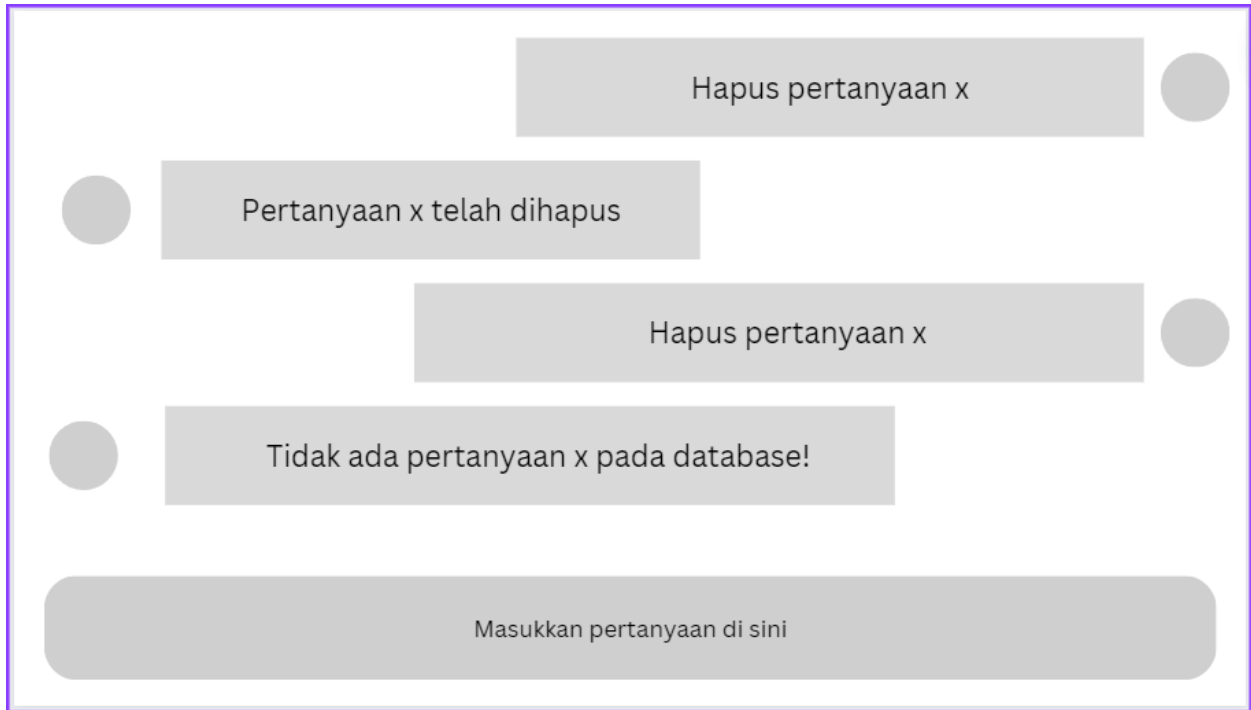
Gambar 4. Ilustrasi Fitur Kalkulator



*Gambar 5. Ilustrasi Fitur Tanggal*

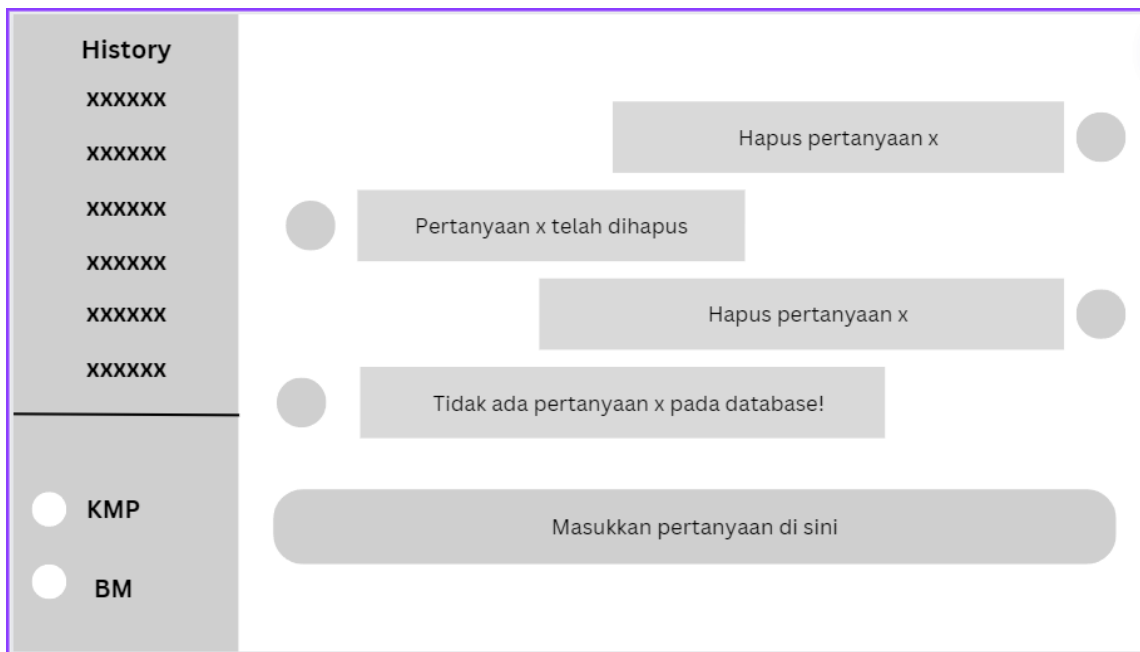


*Gambar 6. Ilustrasi Fitur Tambah Pertanyaan*



Gambar 7. Ilustrasi Fitur Hapus Pertanyaan

Layaknya **ChatGPT**, di sebelah kiri disediakan **history** dari hasil pertanyaan anda. Cukup tampilkan 5-10 pertanyaan terbaru di toolbar kiri. Perhatikan bahwa sistem history disini disamakan dengan chatGPT, sehingga satu history yang diklik menyimpan **seluruh pertanyaan pada sesi itu**. Apabila history diclick, maka akan merestore seluruh pertanyaan dan jawaban di halaman utama. Contoh ilustrasi keseluruhan:



Gambar 8. Ilustrasi Keseluruhan



### Spesifikasi Program:

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 2

### Landasan Teori

#### 2.1. Algoritma KMP

Algoritma KMP (Knuth-Morris-Pratt) merupakan algoritma pencarian string yang dikembangkan oleh Donald Ervin Knuth yang mencari pola (*pattern*) pada sebuah text dengan urutan dari kiri ke kanan. Algoritma KMP melakukan pencarian string dengan cara mencocokkan string dan menggeser jika tidak sesuai seperti pada algoritma brute force. Namun, pada algoritma KMP pergeseran dilakukan dengan lebih efisien. Pergeseran pada algoritma KMP dilakukan berdasarkan fungsi pinggiran ( $b(k)$ ). Fungsi pinggiran merupakan fungsi yang menghitung ukuran prefix dari  $P[0..k]$  yang juga merupakan suffix dari  $P[1..k]$ .

#### 2.2. Algoritma BM

Algoritma BM (Boyer-Moore) merupakan algoritma pencarian string yang menggunakan dua teknik, yaitu:

1. The looking-glass technique  
Mencari P di T dengan bergerak mundur melalui P, mulai dari bagian akhir
2. The character-jump technique  
Ketika terjadi ketidakcocokan (*mismatch*) pada  $T[i] \neq x$ , karakter pada pola  $P[j]$  tidak sama dengan  $T[i]$

Dengan menggunakan kedua teknik tersebut, algoritma BM mencari string dan melakukan pergeseran berdasarkan tiga kasus, yakni:

1. Kasus 1  
Jika pada P terdapat x, maka cobalah untuk menggeser P ke kanan untuk menyesuaikan posisi kemunculan terakhir x pada P dengan posisi x pada  $T[i]$ .
2. Kasus 2  
Jika pada P terdapat x namun untuk menyesuaikan kemunculan terakhir x pada P dengan posisi x pada  $T[i]$ , maka P digeser ke kanan satu karakter.
3. Kasus 3  
Jika kasus 1 dan 2 tidak berlaku, maka P digeser ke kanan sampai posisi  $P[0]$  selaras dengan  $T[i+1]$ .

#### 2.3. Regex

Regex (regular expression) adalah notasi standar yang digunakan untuk mendeskripsikan suatu pola (*pattern*) berupa urutan karakter atau string. Pola tersebut dapat digunakan untuk mencocokkan, mencari, dan mengelola text. Regex sangat berguna pada pencocokan string karena efisien. Regex sendiri sudah menjadi standar internasional pada tools dan bahasa pemrograman.

## **2.4. Deskripsi Singkat Program**

Untuk membuat aplikasi ChatGPT sederhana, diperlukan suatu algoritma untuk menjawab query pertanyaan dari pengguna. Oleh karena itu, pada program ini digunakan regex, algoritma KMP, dan algoritma BM. Regex pada program ini digunakan untuk mengklasifikasikan pertanyaan ke dalam klasifikasi query tertentu. Pada program ini terdapat lima klasifikasi query/fitur. Ketika pengguna mengirimkan suatu query, maka query tersebut pertama-tama akan diklasifikasikan ke dalam query yang mana. Setelah di klasifikasikan, barulah query tersebut diproses oleh program sesuai dengan fiturnya dan jawaban hasil proses tersebut dikirimkan ke pengguna. Pemanfaatan algoritma KMP dan BM terdapat pada fitur pertama, fitur pernyataan teks.

## **Bab 3**

### **Analisis Penyelesaian Masalah**

#### **3.1. Langkah Penyelesaian Masalah Setiap Fitur**

##### **3.1.1. Fitur Pertanyaan Teks**

Pada fitur ini terdapat dua cara dalam memperoleh jawaban, yaitu menggunakan algoritma KMP dan algoritma BM. Cara yang digunakan bergantung pada pilihan pengguna. Kedua cara tersebut sama-sama membutuhkan semua data dari database. Dalam proses pencocokkan query pertanyaan, pertama-tama Program akan meminta semua tuple pertanyaan pada database. Setelah itu, dilakukan pencocokkan dengan menggunakan algoritma sesuai pilihan pengguna.

Untuk algoritma KMP, pertama-tama dicari fungsi pinggiran query dari pertanyaan. Lalu pencocokan mulai dilakukan dari index 0 query pertanyaan dan pertanyaan dari database. Jika karakter pada indeks tersebut sama maka index pencarian akan bergeser. Jika terjadi perbedaan karakter, maka index pencarian akan bergeser sesuai fungsi pinggiran. Jika tidak ada perbedaan karakter sampai index query pertanyaan paling akhir, maka jawaban telah ditemukan dan akan dikembalikan. Proses pencocokkan dilakukan untuk semua pertanyaan yang ada pada database.

Untuk algoritma BM, pertama-tama pencocokkan dilakukan pada index terakhir query pertanyaan dan index ke-m pada pertanyaan dari database, dimana m merupakan panjang query pertanyaan kurang satu. Jika karakter yang dicocokkan sama, maka pencarian akan berlanjut mundur. Jika ditemukan perbedaan karakter, maka akan dilakukan proses pergeseran ke kanan sesuai kasus pada landasan teori. Pencocokkan lalu dilakukan kembali mulai dari index terakhir query pertanyaan. Pencocokkan akan berakhir jika semua karakter pada query pertanyaan sama atau index pencarian pada pertanyaan database sudah sampai index terakhir.

##### **3.1.2. Fitur Kalkulator**

Fitur ini menggunakan konsep rekursif. Query pertanyaan pertama-tama diperiksa. Jika query pertanyaan merupakan sebuah angka, maka query tersebut akan langsung dikembalikan. Jika query pertanyaan tersebut merupakan operasi antara dua bilangan, maka akan dihitung terlebih dahulu hasil dari operasi kedua bilangan tersebut. Setelah dihitung, hasilnya akan langsung dikembalikan. Lalu

jika query pertanyaan bukan merupakan dua kasus di atas, maka akan dicari prioritas operasi berdasarkan operatornya lalu dicari hasil dari operasi tersebut menggunakan fungsi yang sama. Pada bagian inilah rekursif terjadi. Fungsi akan terus memanggil diri sendiri sampai query pertanyaan hanya tersisa satu angka yang merupakan jawaban dari query pertanyaan.

### **3.1.3. Fitur Tanggal**

Pada fitur tanggal, pertama-tama query pertanyaan diubah ke format waktu yang sesuai pada golang. Setelah diubah, query pertanyaan dicari harinya (weekday). Lalu hari tersebut dikembalikan.

### **3.1.4. Tambah Pertanyaan dan Jawaban ke Database**

Pada fitur ini, query pertanyaan diambil bagian pertanyaan dan jawabannya dengan menggunakan regex. Setelah itu, dilakukan pengecekan terlebih dahulu menggunakan algoritma string matching. Apabila ada pertanyaan pada database yang sama dengan pertanyaan yang akan ditambahkan, maka jawaban diupdate dengan jawaban terbaru. Jika tidak ada pertanyaan yang sama pada database, maka pertanyaan dan jawaban di insert ke database.

### **3.1.5. Hapus Pertanyaan dari database**

Sama seperti fitur keempat, tambah pertanyaan dan jawaban ke database, pada fitur ini dilakukan pemisahan terlebih dahulu dengan mengambil bagian pertanyaan dari query. Setelah itu, dilakukan pengecekan juga menggunakan algoritma string matching. Jika pertanyaan ditemukan, maka pertanyaan tersebut akan dihapus dari database. Jika tidak ditemukan, maka akan ditampilkan pesan tidak ditemukan.

## **3.2. Fitur Fungsional dan Arsitektur Aplikasi Web yang Dibangun**

Aplikasi web ini terdiri dua bagian utama yaitu bagian frontend dan backend. Bagian frontend merupakan bagian yang menyediakan interface dan berinteraksi langsung dengan pengguna. Pada bagian frontend terdapat panel chat box yang menampilkan pertanyaan dari user dan jawaban dari aplikasi. Pada panel chat box terdapat text field yang dapat diisi pertanyaan/perintah dari pengguna. Ada juga panel riwayat yang menampilkan riwayat percakapan sebelumnya. Untuk menambahkan riwayat disediakan tombol tambah sesi chat baru. Lalu untuk memilih algoritma yang akan digunakan untuk menjawab pertanyaan dari pengguna, ada panel yang menyediakan opsi antara KMP dan BM.

Pada bagian backend, terdapat lima fitur (penjelasan terdapat pada bagian 3.1) yang masing-masing diproses dengan fungsi yang berbeda pada file yang berbeda. Pada backend juga terdapat database untuk menyimpan pasangan pertanyaan-jawaban, percakapan, serta riwayat percakapan. Singkatnya, bagian backend dari aplikasi berfungsi dalam penyimpanan, pengolahan data dan pemrosesan jawaban dari suatu query pertanyaan. Sedangkan bagian frontend berfokus pada tampilan dan interaksinya dengan pengguna.

## Bab 4

### Implementasi dan Pengujian

#### 4.1. Struktur Data Program

##### 4.1. Frontend

Nama Komponen	Penjelasan
radio_button	Untuk memilih algoritma pencarian KMP atau BM
history	Untuk memilih sesi chat
tombol_tambah	Untuk menambah sesi chat baru
tombol_clear	Untuk menghapus sesi chat
chat_box	Untuk mengirim pertanyaan ke aplikasi dan menampilkan pertanyaan dan jawaban

##### 4.2. Backend

###### 4.2.1. Algoritma Fitur

Nama File	Nama fungsi/prosedur	Penjelasan
HammingDistance.go	HammingDistance	Untuk Mengecek kemiripan teks
KMP.go	sufPref	Mengembalikan fungsi pinggiran
	KMP	Mengembalikan indeks awal ditemukannya kesamaan (-1 jika tidak ada) dan persentase kemiripan dengan algoritma KMP
BM.go	buildLast	Mengembalikan list pergeseran

	BM	Mengembalikan indeks awal ditemukannya kesamaan (-1 jika tidak ada) dan persentase kemiripan dengan algoritma BM
Calculator.go	Calculate	Mengembalikan hasil operasi perhitungan dari query pertanyaan
	getQuery	Mengembalikan operasi perhitungan yang merupakan prioritas paling tinggi dalam suatu operasi perhitungan
	isQueryAtom	Menghasilkan true bila operasi perhitungan merupakan operasi paling dasar (terdiri dari dua angka dan satu operator misal 1+1)
	calculateTwoNumber	Mengembalikan hasil operasi dari dua angka (operasi masukan harus merupakan operasi antara dua angka)
GetDate.go	SearchDate	Mengembalikan hari dari suatu tanggal
ManipQuestAns.go	InsertQuestionAnswer	Menambah pasangan pertanyaan-jawaban ke database. Jika pertanyaan sudah ada maka akan memanggil fungsi UpdateAnswer untuk mengupdate jawaban
	UpdateAnswer	Mengupdate jawaban pada database
Classification.go	CheckEraseQues	Mengecek dan menghapus suatu pertanyaan apabila



		perintah dari user merupakan perintah hapus pertanyaan dan pertanyaan ditemukan
--	--	---

#### 4.2.2. Database

```

+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_history_chat | varchar(255)  | YES  | MUL | NULL    |      |
| id_chat         | varchar(191)  | NO   | PRI | NULL    |      |
| from            | longtext      | YES  |     | NULL    |      |
| chat            | longtext      | YES  |     | NULL    |      |
| type            | longtext      | YES  |     | NULL    |      |
| time            | longtext      | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+

```

Tabel chats

```

+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id             | varchar(191)  | NO   | PRI | NULL    |      |
| question       | longtext      | YES  |     | NULL    |      |
| answer         | longtext      | YES  |     | NULL    |      |
| time_create    | longtext      | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+

```

Tabel quest\_ans

```

+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id             | varchar(191)  | NO   | PRI | NULL    |      |
| last_chat      | longtext      | YES  |     | NULL    |      |
| update_at      | longtext      | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+

```

Tabel chat\_histories

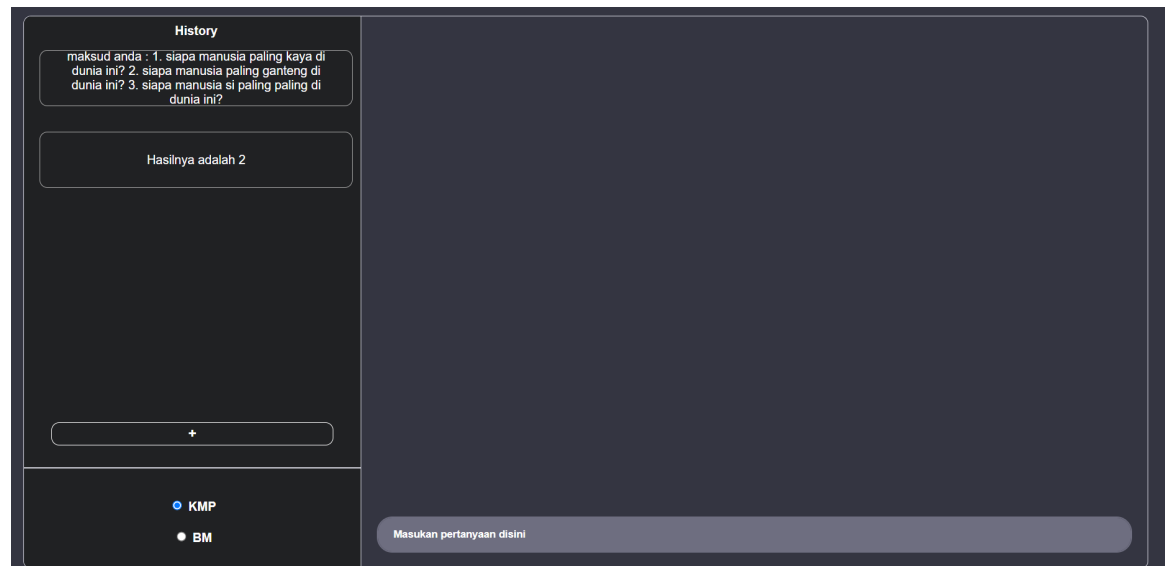
## 4.2. Tata Cara Menjalankan Program

Sebelum menjalankan program, terdapat requirement program yang perlu diinstal terlebih dahulu, yaitu NPM. Untuk mengunduh NPM dapat mengunduh Node js terlebih dahulu dari link berikut <https://nodejs.org/en>. NPM akan otomatis terpasang bersama Node js.

Setelah menginstall NPM, Lakukan langkah-langkah di bawah ini:

1. Buka terminal dan cd folder client atau buka terminal pada folder client yang terdapat pada folder frontend (src\frontend\client)
2. Ketik npm install (jika sudah, abaikan langkah ini)
3. Ketik npm run start

Saat pertama aplikasi berjalan, pengguna akan diminta untuk memilih/membuat sesi percakapan terlebih dahulu.

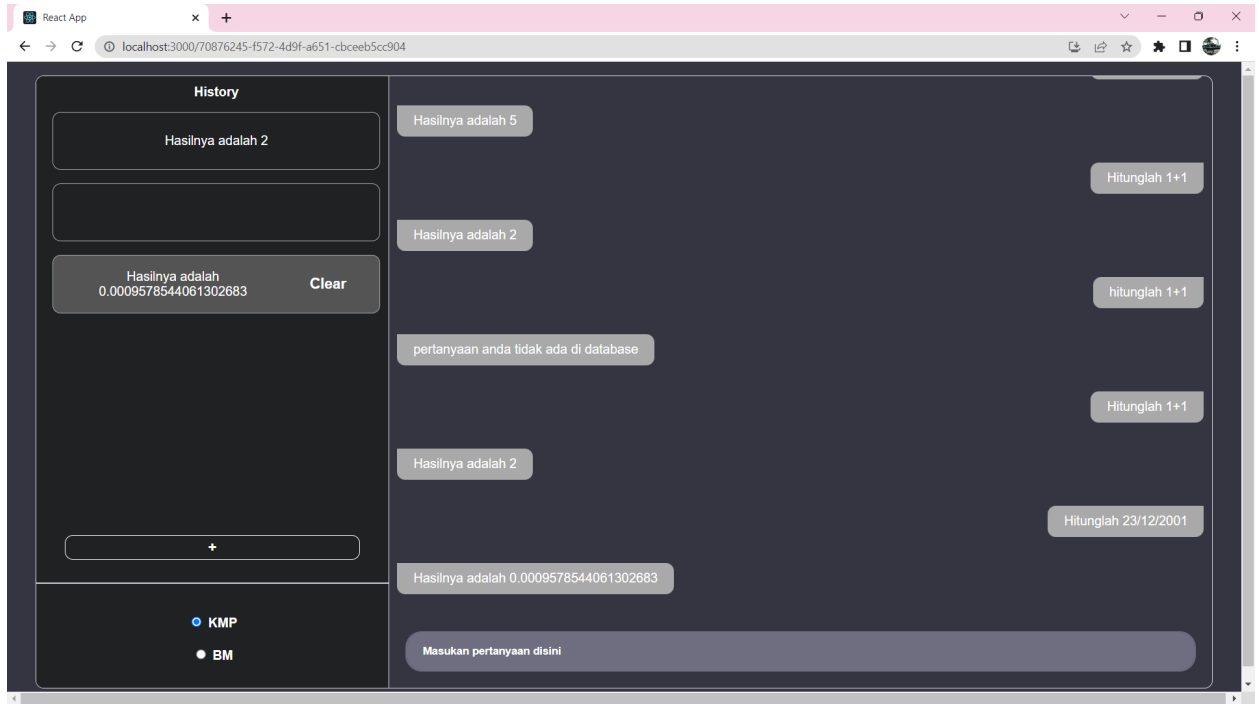


Setelah itu, pengguna dapat mengetikkan pertanyaan/perintah pada chat box dan mengirimnya dengan mengklik enter. Pengguna juga dapat merubah opsi algoritma yang digunakan dalam mencari jawaban. Pertanyaan/perintah yang dapat dijawab oleh program hanyalah pertanyaan yang berkaitan dengan:

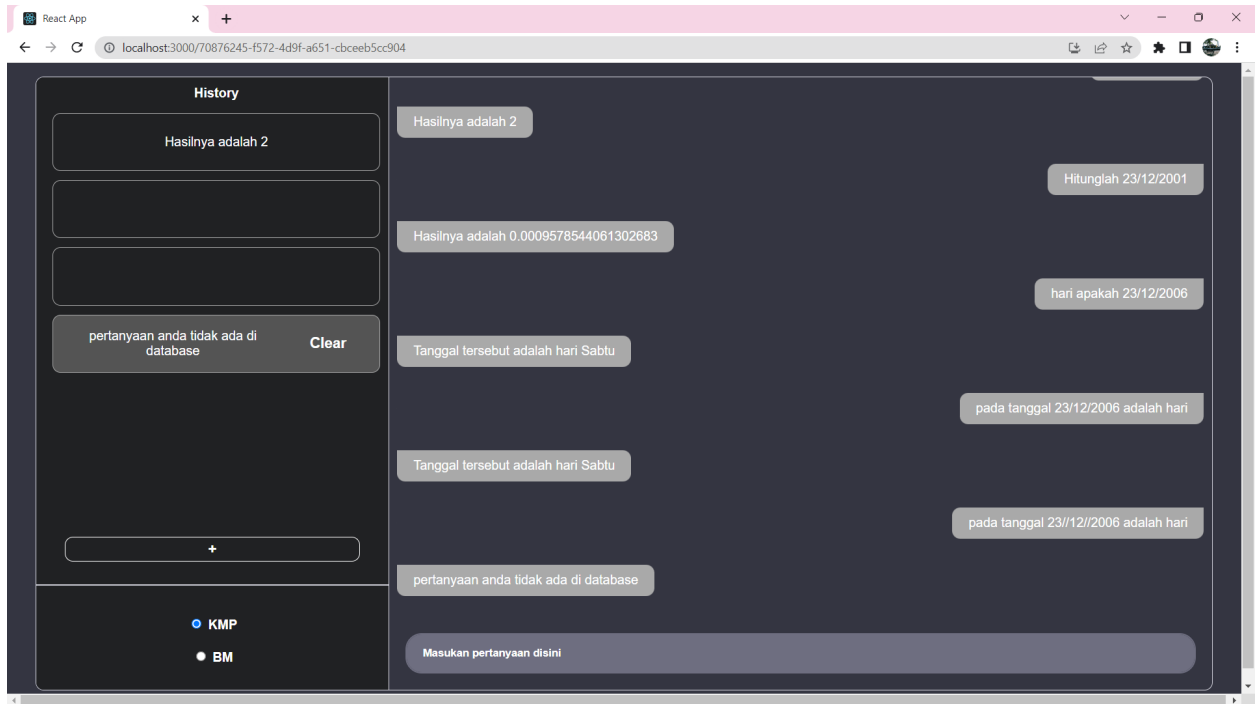
1. Pertanyaan yang ada di database
2. Operasi perhitungan
3. Menentukan hari dari suatu tanggal
4. Menambah pertanyaan dan jawaban ke database
5. Menghapus pertanyaan dari database

## 4.3. Hasil Pengujian

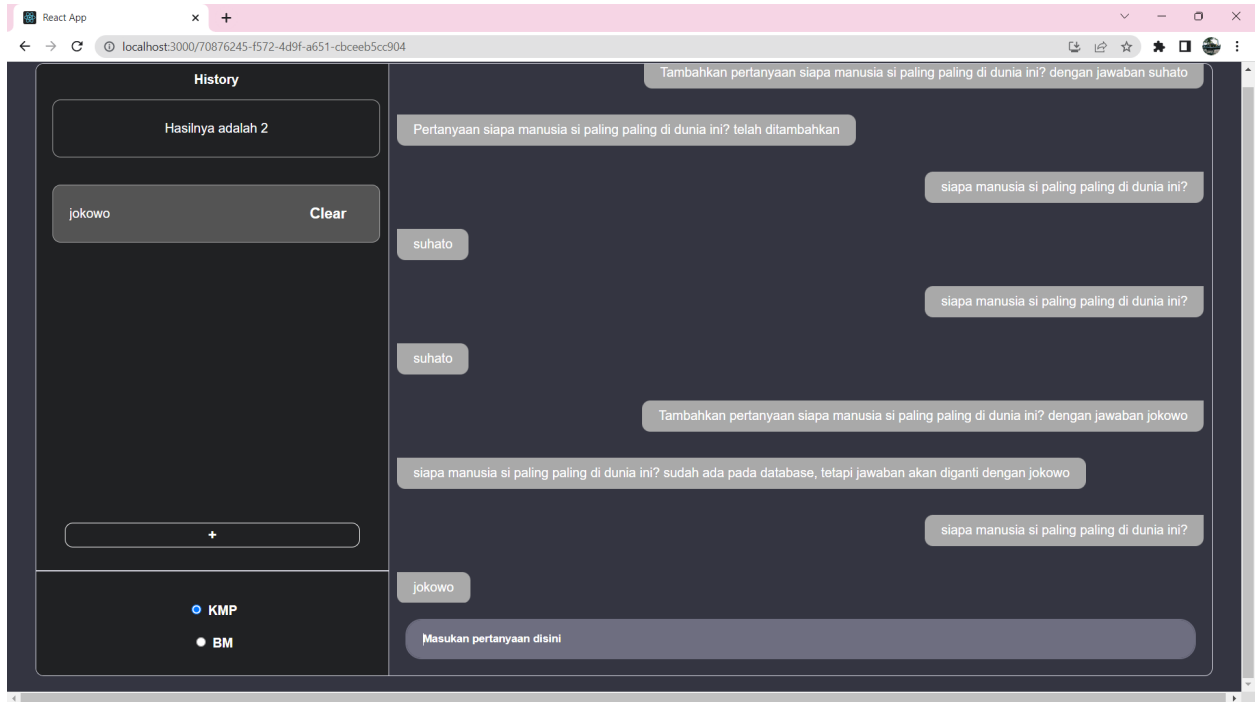
### Test Case 1: Merespons operasi perhitungan



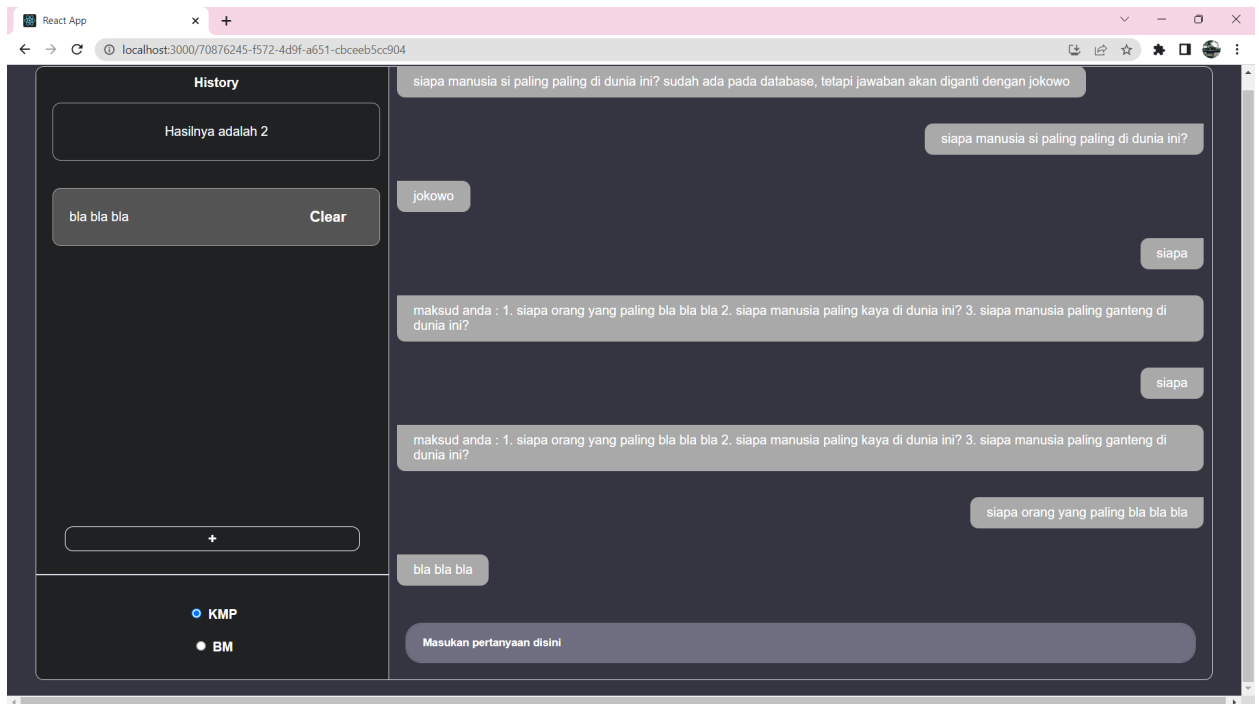
## Test Case 2: Menentukan hari dari suatu tanggal



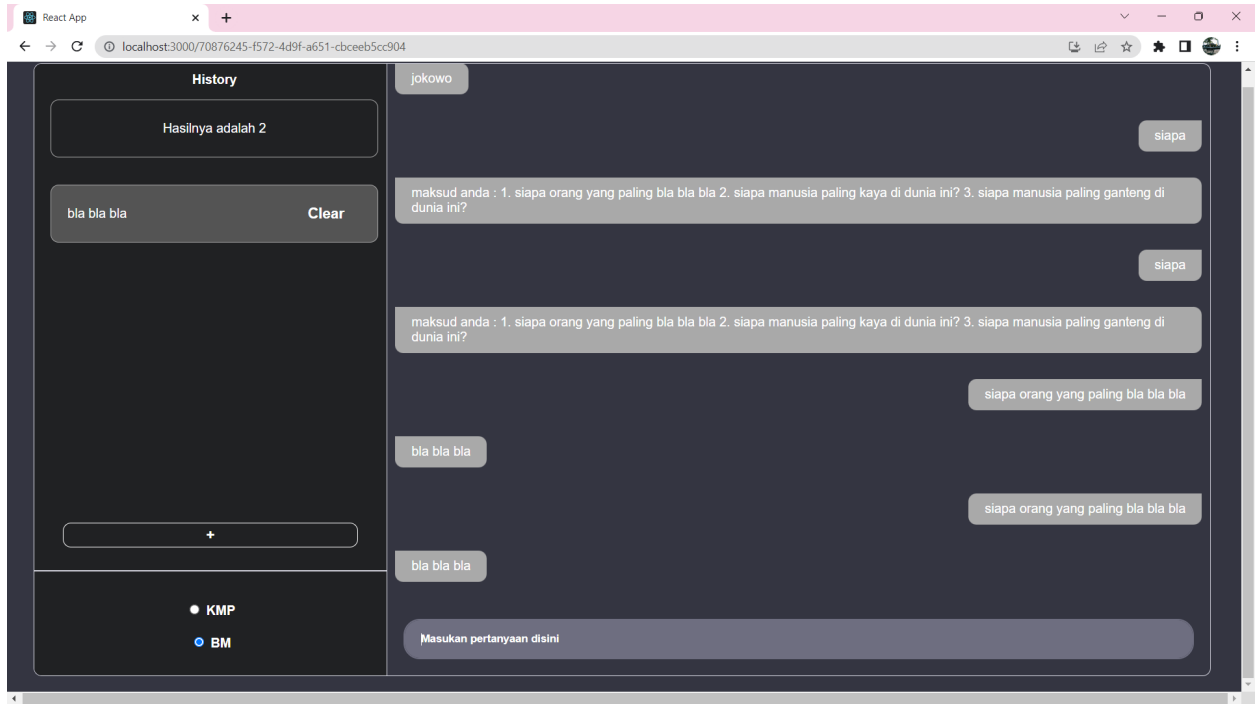
## Test Case 3: Menambahkan pertanyaan dan mengajukan pertanyaan yang ada di database



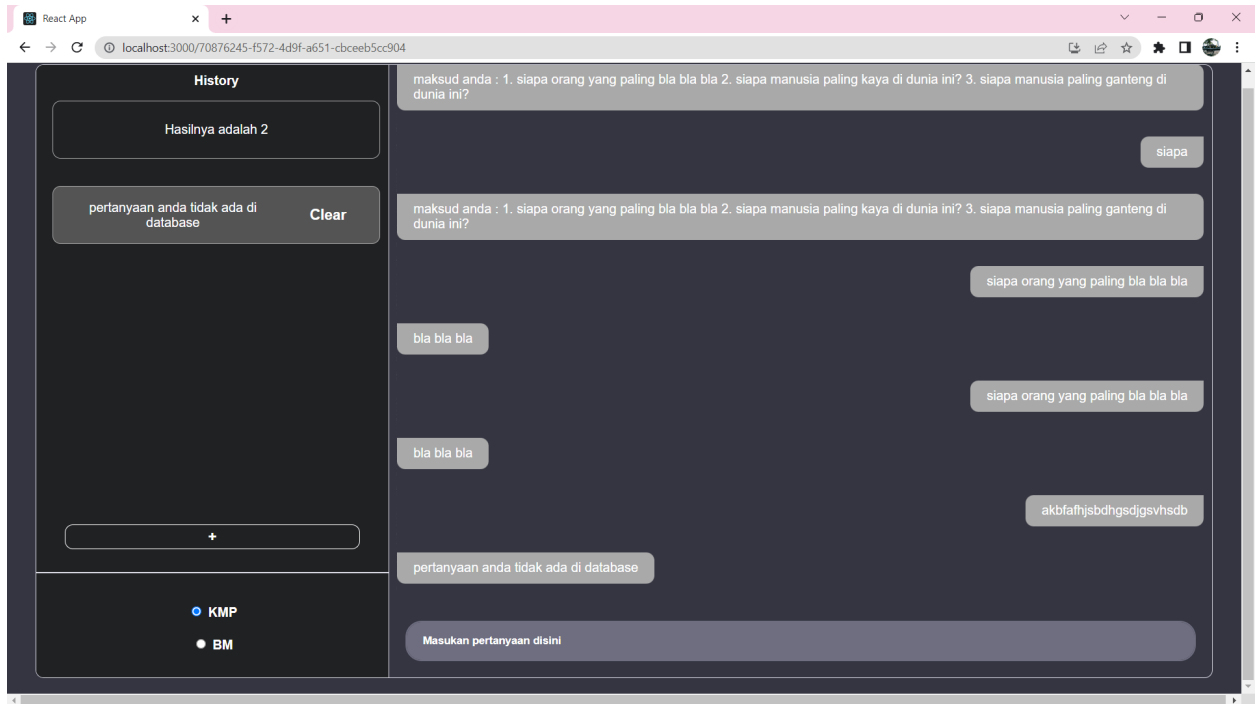
#### Test Case 4: Pertanyaan tidak lengkap, muncul prompt (KMP)



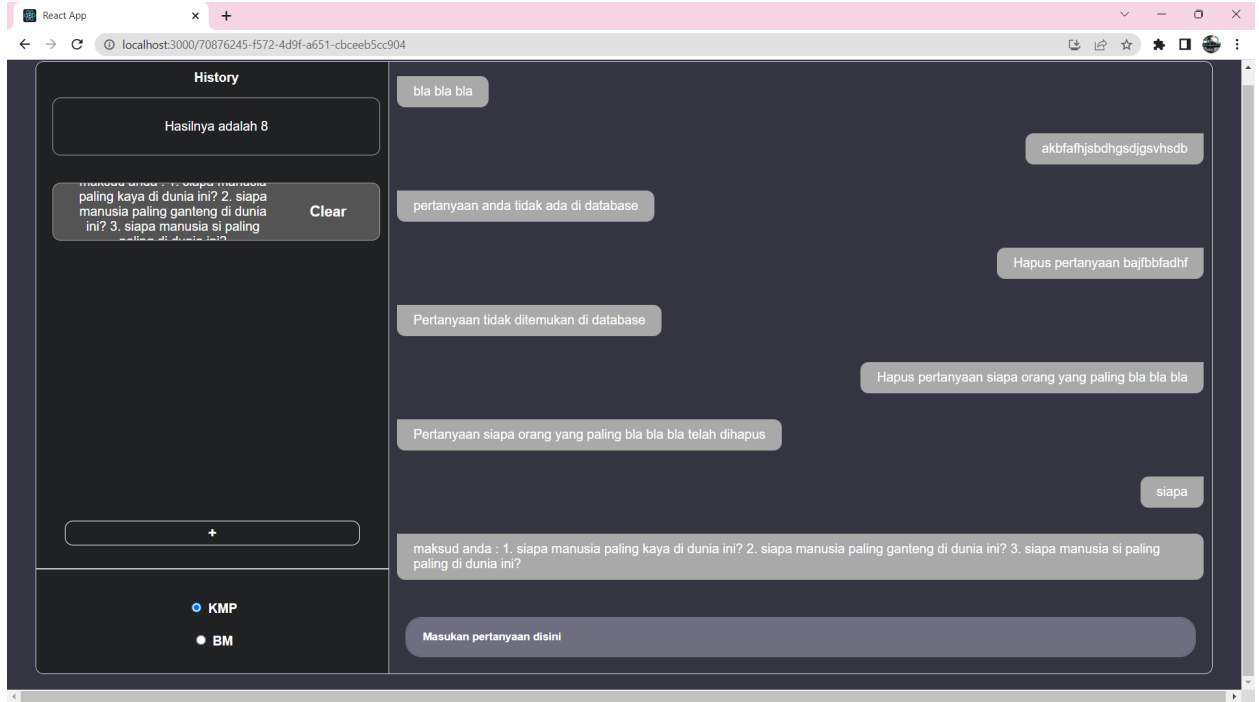
#### Test Case 5: Pertanyaan tidak lengkap, muncul prompt (BM)



## Test Case 6: Pertanyaan huruf random



## Test Case 7: Hapus pertanyaan



#### 4.4. Analisis Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, aplikasi ini dapat merespons pertanyaan/perintah sesuai dengan kelima fitur yang telah ada. Aplikasi mampu mengklasifikasikan pertanyaan/perintah ke fitur yang tepat berdasarkan keyword pertanyaan/perintah. Keyword pertanyaan/perintah harus sesuai agar pertanyaan bisa diklasifikasikan sesuai dengan fiturnya. Jika pertanyaan/perintah tidak tercover oleh fitur, maka akan muncul pesan “pertanyaan anda tidak ada di database”. Saat menggunakan KMP maupun BM, kecepatan pencarian dirasa kurang lebih sama. Hal ini dikarenakan database jumlah datanya masih kecil sehingga kebutuhan akan resource tidak terlalu besar.

## **Bab 5**

### **Kesimpulan dan Saran**

#### **5.1. Kesimpulan**

Berdasarkan Tugas Besar 3 IF2211 Strategi Algoritma Semester 2 2022/2023 berjudul Penerapan String Matching dan Regular Expression dalam Pembuatan ChatGPT Sederhana, kami mendapati bahwa untuk merancang dan mengimplementasi aplikasi ChatGPT Sederhana dapat dilakukan dengan menggunakan regex, algoritma KMP, dan algoritma BM. Regex dimanfaatkan untuk mengklasifikasikan suatu pola string, pada kasus aplikasi ini klasifikasi tersebut digunakan untuk menentukan fitur yang dijalankan dan mengekstrak bagian tertentu dari query pertanyaan. Sedangkan algoritma KMP dan algoritma BM digunakan untuk mencari jawaban yang sesuai pada database. Algoritma KMP dapat berjalan lebih efisien jika ukuran alfabet dari query pertanyaan dan database relatif kecil, sedangkan bila ukuran alfabetnya besar maka algoritma BM akan lebih efisien.

#### **5.2. Saran**

Adapun saran-saran yang dapat kami berikan terkait Tugas Besar 3 IF2211 Strategi Algoritma Semester 2 2022/2023:

1. Beberapa kode yang ditulis masih belum *clean code*. Dari segi maintenance dapat menyebabkan kebingungan. Oleh karena itu, kode masih dapat lebih dirapikan dan diberi komentar agar lebih jelas.
2. Beberapa implementasi program masih dapat dioptimasi agar lebih menghemat resource dan lebih cepat.
3. Dokumentasi terkait pemrograman masih dapat dilengkapi lagi.

## Daftar Pustaka

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>  
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>  
<https://docs.google.com/document/d/1ls6h1A6m-Zhzw6e5eriwMNUAG0D1iwL-eVmVMS2XQoc/edit>



## Lampiran

Link GitHub: [https://github.com/fchrgrib/Tubes3\\_ngasiNamanyaAbisLebaran.git](https://github.com/fchrgrib/Tubes3_ngasiNamanyaAbisLebaran.git)

Link Video: <https://youtu.be/bii94uECJKo>

Link Deploy: <http://ngasi-namanya-abis-lebaran.up.railway.app/>