

LAPORAN TUGAS BESAR 3

IF2211 - Strategi Algoritma

Semester II Tahun 2022/2023

**Penerapan String Matching dan Regular Expression dalam
Pembuatan ChatGPT Sederhana**

Dipersiapkan oleh



Ghazi Akmal Fauzan

13521058

Mohammad Farhan Fahrezy

13521106

Muhammad Naufal Nalendra

13521152

Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

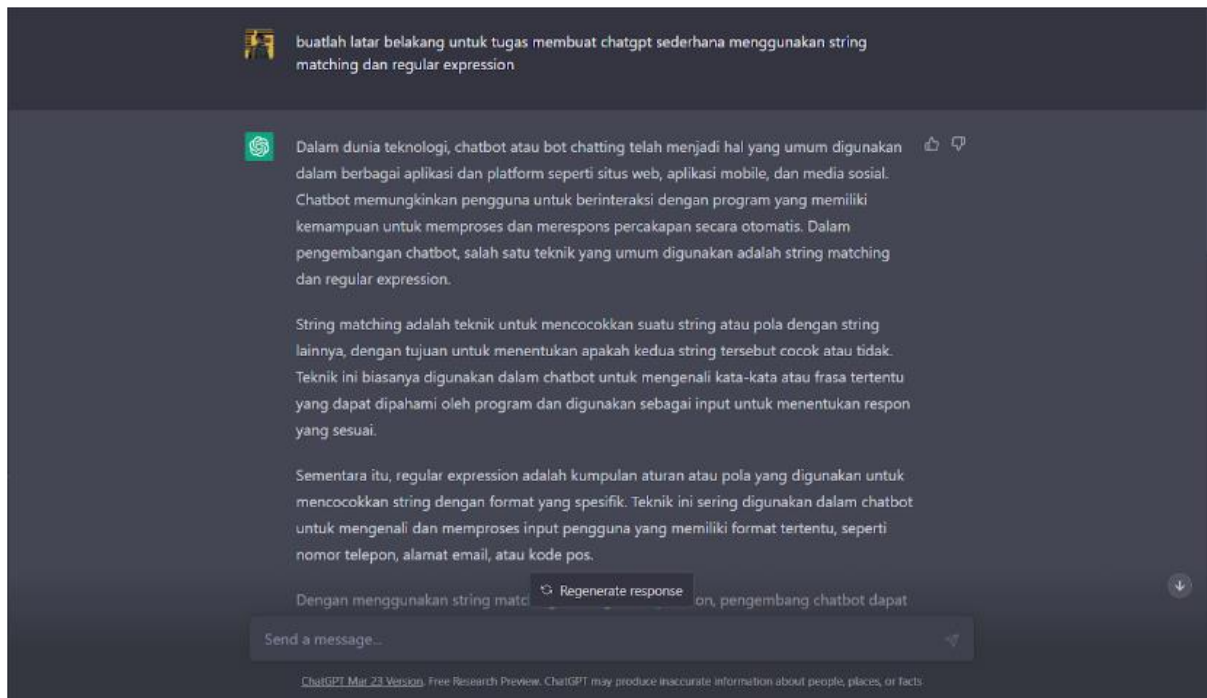
Daftar Isi

1 Deskripsi Tugas	3
1.1. Latar Belakang	3
1.2. Deskripsi Tugas	4
1.3. Fitur-Fitur Aplikasi:	4
2 Landasan Teori.....	9
2.1 Algoritma Knuth–Morris–Pratt (KMP)	9
2.2 Algoritma Boyer-Moore (BM)	10
2.3 Algoritma Levenshtein Distance	10
2.4 Regular Expression (Regex)	12
2.5 Aplikasi Web.....	12
2.5.1 Node.js	13
2.5.2 MySQL	13
2.5.3 Typescript	13
2.5.4 Chakra UI.....	13
3 Analisis Pemecahan Masalah.....	13
3.1 Langkah Pemecahan Masalah Setiap Fitur.....	14
3.1.1 Fitur Tanya Jawab	14
3.1.2 Fitur Kalkulator Sederhana	14
3.1.3 Fitur Mencari Hari dari Sebuah Tanggal	14
3.1.4 Fitur Menambah Pertanyaan dan Jawaban	14
3.1.5 Fitur Menghapus Pertanyaan dan Jawaban	14
3.2 Fitur Fungsional Aplikasi	15
3.2.1 Fitur Tanya Jawab	15
3.2.2 Fitur Kalkulator Sederhana	15
3.2.3 Fitur Mencari Hari dari Sebuah Tanggal	15
3.2.4 Fitur Menambah Pertanyaan dan Jawaban	15
3.2.5 Fitur Menghapus Pertanyaan dan Jawaban	16
3.3 Arsitektur Aplikasi	16
4 Implementasi dan Pengujian.....	16
4.1 Spesifikasi Teknis Program.....	16
4.1.1. Backend.....	16
4.1.2 Frontend	17
4.2 Tata Cara Penggunaan Program	17
4.2.1. Requirements	17
4.2.2. Cara Menjalankan Program	17
4.2.3. Cara Menggunakan Program	18
4.3 Hasil Pengujian	20
4.4 Analisis Hasil Pengujian	27
5 Kesimpulan dan Saran	28
5.1 Kesimpulan.....	28
5.2 Saran.....	28
5.3 Komentar	28

1 Deskripsi Tugas

1.1. 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 1.1. Ilustrasi Chatbot 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.

1.2. 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.

1.3. 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.

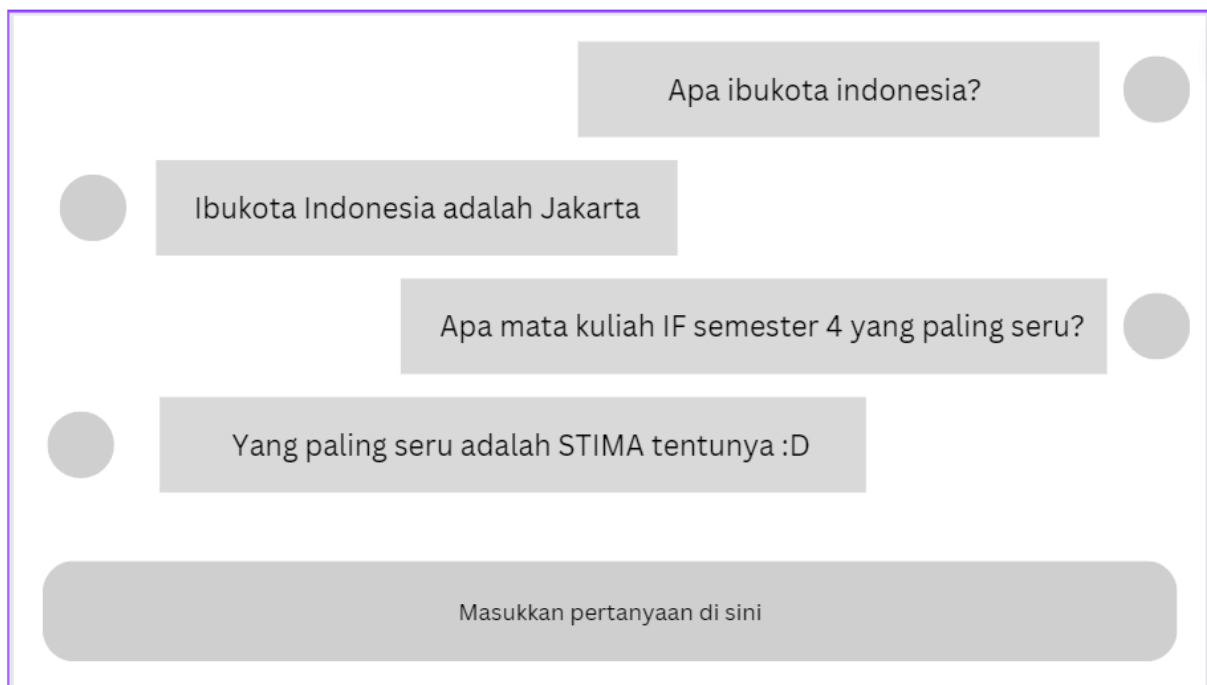
4. Tambah pertanyaan dan jawaban ke database

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.

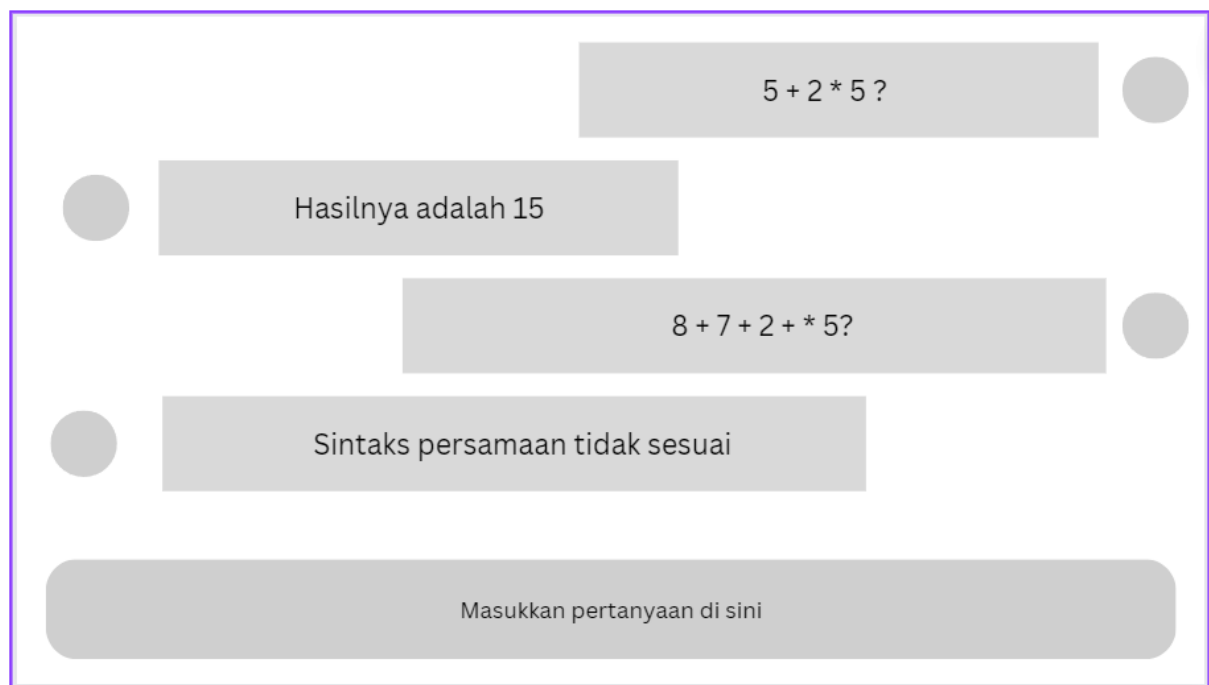
Klasifikasi dilakukan menggunakan **regex** dan terklasifikasi layaknya bahasa sehari - hari. Algoritma string matching KMP dan BM digunakan untuk klasifikasi query teks. Tersedia toggle untuk memilih algoritma KMP atau BM. Semua pemrosesan respons dilakukan pada sisi **backend**. Jika ada pertanyaan yang sesuai dengan fitur, maka tampilkan saja “Pertanyaan tidak dapat diproses”. Berikut adalah beberapa **contoh** ilustrasi sederhana untuk tiap pertanyaannya.



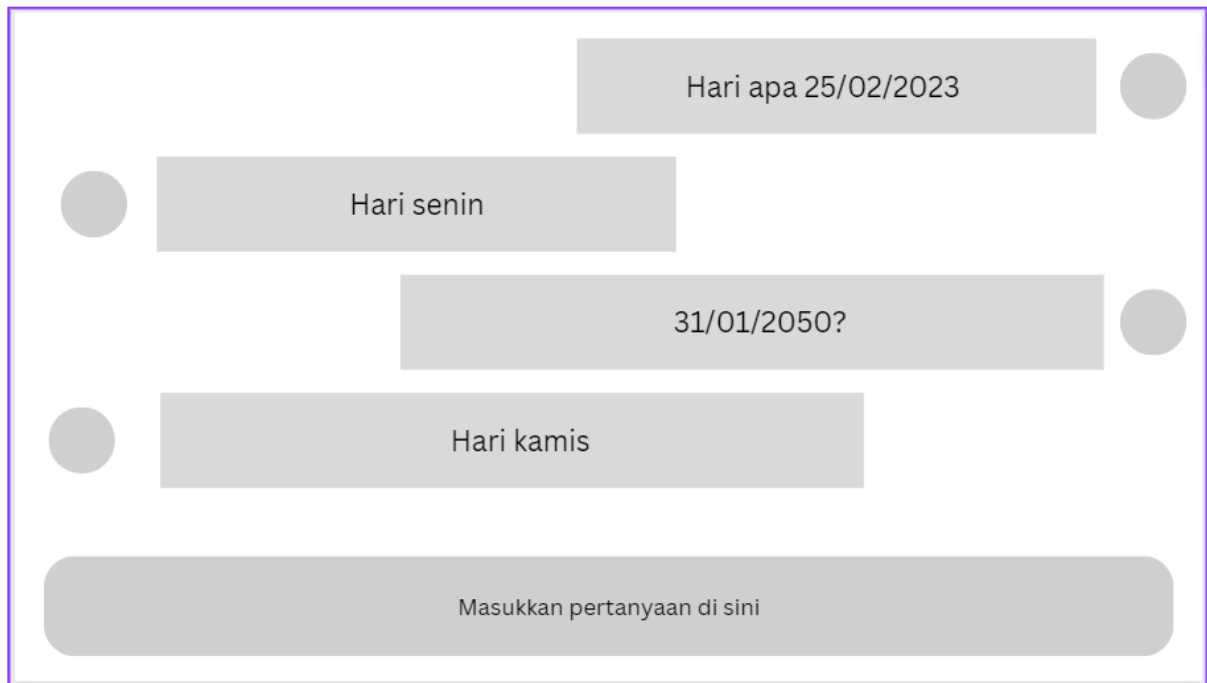
Gambar 1.2. Ilustrasi Fitur Pertanyaan teks kasus exact



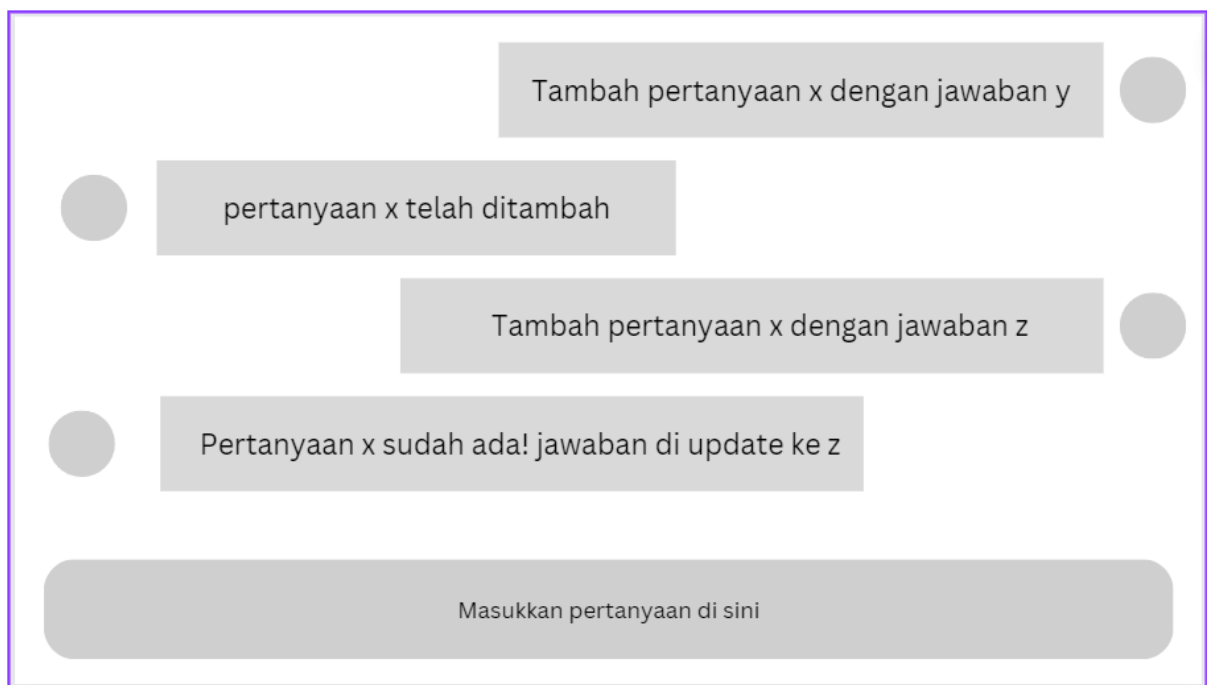
Gambar 1.3. Ilustrasi Fitur Pertanyaan teks kasus tidak exact



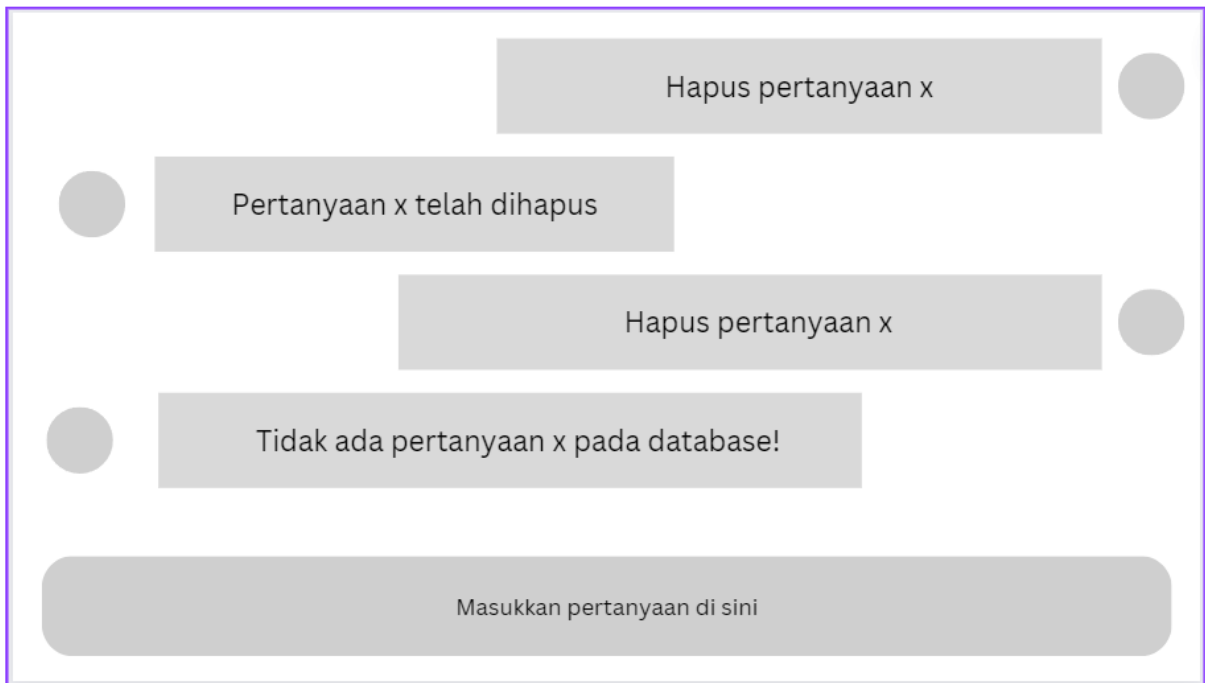
Gambar 1.4. Ilustrasi Fitur Kalkulator



Gambar 1.5. Ilustrasi Fitur Tanggal

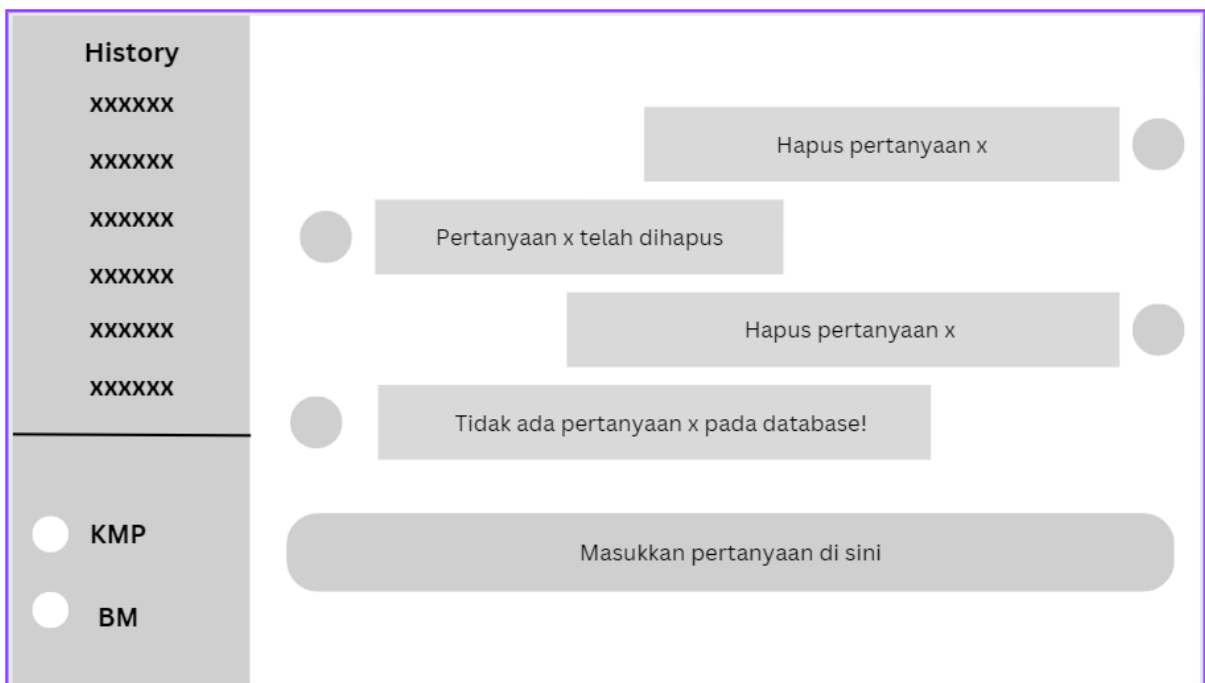


Gambar 1.6. Ilustrasi Fitur Tambah Pertanyaan



Gambar 1.7. 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 1.8. Ilustrasi Keseluruhan

2 Landasan Teori

Pada chatGPT sederhana yang dirancang, terdapat fitur tanya jawab dimana pengguna dapat memberi aplikasi pertanyaan dan aplikasi akan menjawab pertanyaan pengguna sesuai informasi pada basis data aplikasi. Beberapa algoritma seperti Knuth-Morris-Pratt dan Boyer-Moore digunakan untuk mencari string yang sama sedangkan algoritma levenshtein distance digunakan untuk mencari jarak perbedaan antar dua buah string

Selain itu, fitur-fitur lain yang ada pada aplikasi membutuhkan cara agar dapat mengantisipasi berbagai variasi user input untuk pertanyaan yang sama. Untuk mengatasi hal tersebut, digunakanlah regex atau regular expression sehingga fitur dapat digunakan dengan baik

2.1 Algoritma Knuth–Morris–Pratt (KMP)

Algoritma Knuth-Morris-Pratt (KMP) melakukan pencocokan *string* dari kiri ke kanan (seperti algoritma *brute force*). Tetapi, algoritma ini lekaukan pergeseran lebih cerdas dibandingkan dengan algoritma *brute force*. Hal tersebut dilakukan untuk mengurangi perbandingan yang sia- sia.

Misalkan *pattern* disimpan di dalam variabel P. Langkah pertama yang dilakukan algoritma ini adalah dengan membuat larik yang berisikan nilai *border function* $b(k)$ untuk setiap kemungkinan posisi ketidakcocokan yang terjadi di P[].

Dengan :

j = ketidakcocokan yang terjadi di P[]

k = posisi sebelum ketidakcocokan ($k = j - 1$)

Border function $b(k)$ didefinisikan sebagai *prefix* terpanjang dari $P[0..k]$ yang juga merupakan *suffix* dari $P[1..k]$. Berikut ini adalah contoh dari *border function* pada pattern “abaaba”.

j	0	1	2	3	4	5
P[j]	a	b	a	a	b	a

k	0	1	2	3	4
b(k)	0	0	1	1	2

Dengan menggunakan *border function* tersebut, algoritma ini memodifikasi pergeseran pada algoritma *brute force*. Misalnya jika ketidakcocokan karakter terjadi pada indeks ke-j, maka ubah nilai j menjadi $b(k)$ dengan $k = j - 1$.

2.2 Algoritma Boyer-Moore (BM)

Misalkan *pattern* P akan dicocokkan dengan teks T. Algoritma Boyer-Moore melakukan pencocokan string menggunakan dua teknik, yaitu :

1. Teknik *looking-glass*

Mencari P di T dengan bergerak ke belakang melewati P, dimulai dari huruf terakhir P

2. Teknik *character-jump*

Dilakukan ketika karakter yang dicocokkan tidak sama. Misalkan huruf yang tidak sama pada T adalah x. Terdapat 3 kasus

2.1. Kasus 1 :

Jika P mengandung x, cobalah geser P ke kanan untuk menyesuaikan posisi x di P dengan T[i]

2.2. Kasus 2 :

Jika P mengandung x, namun penggeseran ke kemunculan terakhir tidak mungkin, maka geser P sebesar 1 karakter ke T[i+1]

2.3. Kasus 3 :

Jika kasus 1 dan kasus 2 tidak memenuhi, geser P sehingga menyesuaikan P[0] dengan T[i+1]

Pada kasus 1 dan 2, algoritma BM membutuhkan informasi mengenai kemunculan terakhir sebuah karakter pada T dalam pola P. Oleh karena itu, sebelum menjalankan pengecekan string, mula-mula algoritma ini melakukan pembuatan larik yang menyimpan kemunculan terakhir elemen pada P dengan menggunakan fungsi *Last Occurrence* L(x).

Misalkan P sebagai berikut

a	b	a	c	a	b
0	1	2	3	4	5

Dengan A = {a, b, c, d}

Maka larik yang berisi hasil dari fungsi *Last Occurrence* sebagai berikut

x	a	b	c	d	b
L(x)	1	2	3	4	5

Larik tersebut digunakan untuk menentukan lokasi pergeseran.

2.3 Algoritma Levenshtein Distance

Algoritma Levenshtein Distance, atau sering disebut dengan Edit Distance merupakan algoritma untuk mencari jumlah perbedaan antara dua buah string. Algoritma ini ditemukan pada tahun 1965 oleh seorang ilmuwan Rusia bernama Vladimir Levenshtein.

Algoritma Levenshtein Distance pada dasarnya akan menghitung jumlah minimum dari upaya transformasi suatu string menjadi string lain. Transformasi ini meliputi penggantian, penghapusan, dan penyisipan. Kegunaan dari levenshtein distance adalah untuk mengoptimalkan pencarian string karena apabila dilakukan pencarian setiap kombinasi operasi-operasi string tersebut maka akan membutuhkan sumber daya yang besar dan tidak efektif.

$$D(s, t) = \min D(s-1, t) + 1 \quad (\text{penghapusan}) \quad (1)$$

$$D(s, t) = \min D(s, t-1) + 1 \quad (\text{penyisipan}) \quad (2)$$

$$D(s, t) = \min D(s-1, t-1) + 1, s_j \neq t_i \quad (\text{penggantian}) \quad (3)$$

$$D(s, t) = \min D(s-1, t-1), s_j = t_i \quad (\text{tidak ada perubahan}) \quad (4)$$

Keterangan:

s = String sumber	$s(j)$ = Karakter <i>string</i> sumber ke- j
t = String target	$t(i)$ = Karakter <i>string</i> target ke- i
D = jarak edit <i>Levenshtein Distance</i>	

Gambar 2.1. levenshtein distance

Adapun langkah-langkah algoritma Levenshtein distance dalam mendapatkan nilai edit distance adalah sebagai berikut:

Misalkan S = String sumber, dan T = String target

Langkah 1: Inisialisasi

- a) Hitung panjang S dan T, misalkan m dan n
- b) Buat matriks berukuran 0...m baris dan 0...n kolom
- c) Inisialisasi baris pertama dengan 0...n
- d) Inisialisasi kolom pertama dengan 0...m

Langkah 2: Proses

- a) Periksa S[i] untuk $1 < i < n$
- b) Periksa T[j] untuk $1 < j < m$

c) Jika $S[i] = T[j]$, maka entrinya adalah nilai yang terletak pada tepat didiagonal atas sebelah kiri, yaitu $d[i,j] = d[i-1,j-1]$

d) Jika $S[i] \neq T[j]$, maka entrinya adalah $d[i,j]$ minimum dari:

- Nilai yang terletak tepat diatasnya, ditambah satu, yaitu $d[i,j-1]+1$
- Nilai yang terletak tepat dikirinya, ditambah satu, yaitu $d[i-1,j]+1$
- terletak pada tepat didiagonal 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.

2.4 Regular Expression (Regex)

Regular expression adalah sekumpulan notasi dan karakter yang digunakan untuk mendeskripsikan suatu pola pada pencarian berbasis karakter. Regex juga merupakan pola (atau filter) yang menjelaskan kumpulan string yang sesuai dengan pola tersebut. Dengan kata lain, regex menerima sekelompok string tertentu dan menolak sisanya.

Regex terdiri dari urutan karakter dan metakarakter (seperti `.`, `\d`, `\D`, `\s`, `\S`, `\w`, `\W`), dan operator (seperti `+`, `*`, `?`, `|`, `^`). Mereka dibangun dengan menggabungkan banyak sub-ekspresi yang lebih kecil.

Berikut adalah contoh regex :

1. Nomor Telepon

```
/^(^+62\s?|^0)(\d{3,4}-?)?{2}\d{3,4}$/
```

Contoh string yang diterima ekspresi regex tersebut adalah '+62 812 3456 7890' dan '0812-3456-7890'

2. Alamat Rumah

```
/^[a-zA-Z0-9\s.,-]+$ /g
```

Contoh string yang diterima ekspresi regex tersebut adalah 'Jalan Mawar No. 15' dan 'Jalan Ahmad Yani'

2.5 Aplikasi Web

Aplikasi Web yang kami bangun menggunakan beberapa teknologi yang berguna untuk implementasi algoritma pada *backend*, informasi yang disimpan pada database, serta visualisasi aplikasi web pada bagian *frontend*. Teknologi tersebut adalah sebagai berikut:

2.5.1 Node.js

NodeJS adalah lingkungan runtime sumber terbuka dan lintas platform yang dibangun di atas mesin JavaScript V8 Chrome untuk menjalankan kode JavaScript di luar browser. NodeJS bukanlah sebuah kerangka kerja (framework), dan bukan pula bahasa pemrograman. NodeJS menyediakan lingkungan runtime lintas platform yang didukung oleh I/O yang non-blocking (asynchronous) dan berbasis event untuk membangun aplikasi server-side yang sangat scalable menggunakan bahasa JavaScript.

2.5.2 MySQL

Structured Query Language (SQL) adalah bahasa komputer yang digunakan untuk menyimpan, memanipulasi, dan mengambil data yang tersimpan dalam sistem manajemen database relasional (RDBMS). SQL dikembangkan di IBM oleh Donald Chamberlin, Donald C. Messerli, dan Raymond F. Boyce pada tahun 1970-an.

MySQL adalah Sistem Manajemen Database Relasional open-source yang menyimpan data dalam format terstruktur menggunakan baris (rows) dan kolom (columns). Bahasa MySQL mudah digunakan dibandingkan dengan bahasa pemrograman lain seperti C, C++, Java, dll. Dengan mempelajari beberapa perintah dasar, kita dapat bekerja, membuat, dan berinteraksi dengan Database.

2.5.3 Typescript

TypeScript adalah bahasa pemrograman berorientasi objek Open Source yang dikembangkan dan dipelihara oleh Microsoft Corporation. TypeScript adalah bahasa yang strongly typed dan versi pertamanya diperkenalkan pada tahun 2012. TypeScript adalah Strict Super Set dari JavaScript, yang berarti bahwa semua yang diimplementasikan di JavaScript dapat diimplementasikan menggunakan TypeScript bersamaan dengan pilihan menambahkan fitur-fitur yang ditingkatkan (setiap Kode JavaScript yang ada merupakan Kode TypeScript yang valid). Karena kode TypeScript dikonversi menjadi kode JavaScript, maka ini membuatnya lebih mudah diintegrasikan ke dalam proyek JavaScript. TypeScript dirancang terutama untuk proyek skala besar.

2.5.4 Chakra UI

Chakra UI adalah sebuah library komponen modern untuk React. Ini dilengkapi dengan serangkaian komponen React yang dapat digunakan kembali dan dapat digabungkan untuk membuat aplikasi front-end. Keunggulannya terletak pada ke-sederhanaannya, ke-modularannya, dan ke-aksesibilitasnya. Chakra UI dapat digunakan untuk membuat aplikasi React yang mudah diakses serta mempercepat proses pengembangan. Chakra UI menggunakan Emotion dan Styled System. Style System adalah arsitektur yang sangat baik untuk membangun sebuah pustaka komponen UI.

3 Analisis Pemecahan Masalah

3.1 Langkah Pemecahan Masalah Setiap Fitur

3.1.1 Fitur Tanya Jawab

Program akan menerima input user lalu menggunakan algoritma KMP atau BM untuk mencari data pada database yang sama persis, jika tidak ada maka program akan mencari jarak antara input dengan semua string pada database dan diurutkan menurun. Jika ada string dengan kemiripan lebih dari atau sama dengan 90% maka akan ditampilkan. Jika masih belum ada akan ditampilkan 3 pilihan pertanyaan yang paling mirip

3.1.2 Fitur Kalkulator Sederhana

Program akan menerima input user untuk menggunakan kalkulator. Selanjutnya program akan memproses input menjadi sebuah stack berisi operator dan stack berisi operand. Program akan menghitung dua operand dan operator dari tiap stack hingga stack habis. Lalu hasil perhitungan akan ditampilkan ke layar

3.1.3 Fitur Mencari Hari dari Sebuah Tanggal

Program akan menerima input yang berformat tanggal dengan pemisah antara elemen tanggal bisa berupa tanda hubung, garis miring, maupun spasi. Selanjutnya program akan memeriksa apakah tanggal yang dimasukkan valid. Jika iya program akan mengembalikan hari dari tanggal yang dicari

3.1.4 Fitur Menambah Pertanyaan dan Jawaban

Program akan memeriksa jika input user meminta untuk menambah pertanyaan dengan mengantisipasi pola kalimat "Tambahkan pertanyaan ... dengan jawaban ..." menggunakan regex. Jika iya, program akan memanggil fungsi addQuestion untuk mengatasi permintaan user memasukkan pertanyaan ke database.

3.1.5 Fitur Menghapus Pertanyaan dan Jawaban

Program akan memeriksa jika input user meminta untuk menghapus pertanyaan dengan mengantisipasi pola kalimat "Hapus pertanyaan ..." menggunakan regex. Jika iya maka program akan memanggil fungsi deleteQuestion untuk menghapus pertanyaan beserta jawabannya dengan cara mengambil bagian dari string setelah pattern.

3.2 Fitur Fungsional Aplikasi

3.2.1 Fitur Tanya Jawab

Aplikasi dapat menyocokkan input pertanyaan pengguna dengan pasangan pertanyaan dan jawaban pada database menggunakan algoritma Knuth-Morris-Pratt serta algoritma Booyer-Moore. Kedua algoritma diatas akan mencari string dengan kesamaan yang tepat sama dengan input user. Jika tidak ada pertanyaan yang sama, akan digunakan algoritma levenshtein distance untuk mencari pertanyaan dengan kesamaan minimal 90 %. Apabila tidak ada pertanyaan dengan kesamaan diatas 90 % maka aplikasi akan menunjukkan tiga pilihan pertanyaan yang paling mirip dengan input pengguna. Pengguna dapat mengetikkan kembali pertanyaan yang hendak ditanyakan

3.2.2 Fitur Kalkulator Sederhana

Aplikasi dapat melakukan operasi matematika sederhana yang menggunakan operator tambah, kurang, kali, bagi, pangkat, serta tanda kurung. Aplikasi akan menerima input user lalu memvalidasi operasi yang diberikan menggunakan regex. Operasi akan diterima jika tidak ada operator yang tidak dikenali, sementara kasus lain seperti pembagian dengan nol misalnya akan menghasilkan 'NaN' atau *not a number*.

3.2.3 Fitur Mencari Hari dari Sebuah Tanggal

Aplikasi dapat menerima input berupa tanggal lalu akan merespon dengan hari yang sesuai dengan tanggal tersebut. Aplikasi akan pertama memvalidasi input yang diberikan oleh user menggunakan regex. Format tanggal yang diterima adalah dua digit hari, dua digit bulan atau nama bulan, serta empat digit tahun. Tanda pemisah antara hari, bulan, dan tahun dapat berupa tanda hubung, garis miring, maupun spasi. Program juga akan memeriksa jika tanggal yang diberikan tidak memiliki jumlah hari yang melebihi bulan yang diberikan. Setelah pencarian selesai, program akan menunjukkan kepada pengguna hari dari sebuah tanggal.

3.2.4 Fitur Menambah Pertanyaan dan Jawaban

Aplikasi memberi pengguna kesempatan untuk menambahkan pertanyaan dan jawabannya sendiri ke database. Aplikasi akan memeriksa jika input user meminta untuk menambah pertanyaan dengan mengantisipasi pola kalimat "Tambahkan pertanyaan ... dengan jawaban ..." menggunakan regex. Selanjutnya aplikasi akan menggunakan algoritma string matching untuk mencari tahu apakah pertanyaan sudah ada. Jika iya, maka jawaban dari pertanyaan tersebut akan diperbaharui. Sementara jika tidak, maka pertanyaan beserta jawaban akan ditambahkan ke database.

3.2.5 Fitur Menghapus Pertanyaan dan Jawaban

Aplikasi memberi pengguna kesempatan untuk menghapus pertanyaan dan jawabannya dari database. Aplikasi akan memeriksa jika input user meminta untuk menghapus pertanyaan dengan mengantisipasi pola kalimat "Hapus pertanyaan ...". menggunakan regex. Selanjutnya aplikasi akan menggunakan algoritma string matching untuk mencari tahu apakah pertanyaan sudah ada. Jika iya, maka pertanyaan beserta jawaban akan dihapus. Sementara jika tidak, maka program akan memberitahu user kalau tidak ada pertanyaan yang sesuai.

3.3 Arsitektur Aplikasi

Backend aplikasi ini menggunakan bahasa Node js dengan framework Express dengan pendekatan yang digunakan adalah REST-ful API. Frontend aplikasi ini menggunakan bahasa pemrograman Typescript dengan framework React. Sementara itu, untuk membantu *styling* dan penggunaan komponen *reusable*, digunakan library bootstrap dan react-bootstrap.

4 Implementasi dan Pengujian

4.1 Spesifikasi Teknis Program

4.1.1. Backend

Spesifikasi teknis program untuk backend terdiri dari modul-modul berikut:

1. Database.js

Modul ini bertugas mengatur seluruh koneksi aplikasi ke database yang ada. Fungsi – fungsi pada modul dapat menambahkan pertanyaan dan jawaban, mendapatkan history chat, dan sebagainya.

2. Date.js

Modul ini bertugas mengatur fitur mencari hari dari tanggal yang diberikan oleh user. Modul berisi fungsi `getDayOfDate` yang menerima parameter `dateStr` berupa string. Fungsi akan memproses input dan mencari jawaban yang dikehendaki user.

3. Calculator.js

Modul ini bertugas mengatur fitur kalkulator dan melakukan operasi aritmatika sederhana. Modul memiliki fungsi utama `calculate` yang akan menerima ekspresi aritmatika dari user dan menghitung menggunakan konsep *stack expression evaluation*. Fungsi tersebut juga akan memeriksa jika terdapat operator atau operand yang tidak valid dan memberitahu user bahwa terdapat kesalahan pada input

4. stringMatching.js

Modul ini bertugas mengatur fitur pencocokan string dan tanya jawab menggunakan regex, algoritma KMP, algoritma BM, serta algoritma levenshtein distance. Modul memiliki

beberapa fungsi untuk implementasi algoritma KMP serta BM, begitu pula algoritma utama untuk string matching. Terdapat fungsi findMatchingString yang memiliki parameter input user, database, serta algoritma yang dipilih. Fungsi akan menjawab pertanyaan user berdasarkan informasi pada basis data.

5. Main.js

Modul ini merupakan modul yang bertugas menggabungkan modul – modul semua fitur yang ada serta menerima input dan mengembalikan output. Modul memiliki fungsi determineFeature yang akan mengklasifikasi input user menggunakan regex lalu menentukan fitur mana yang dibutuhkan untuk menjawab permintaan user.

4.1.2 Frontend

Adapun spesifikasi teknis program untuk modul – modul pada backend adalah sebagai berikut:

1. App.tsx

Modul ini merupakan modul yang bertugas untuk mengatur algoritma kode frontend. Segala hal yang terkait input dan output frontend diatur oleh modul ini.

4.2 Tata Cara Penggunaan Program

4.2.1. Requirements

Untuk menjalankan program, Anda harus menginstall bahasa Node js, MySQL, serta beberapa dependencies yang ada dengan cara mengetikkan pada terminal :

```
npm install
```

4.2.2. Cara Menjalankan Program

Setelah anda memasang semua *requirements* yang dibutuhkan, anda dapat membuka terminal dan mengetikkan :

```
npm run dev
```

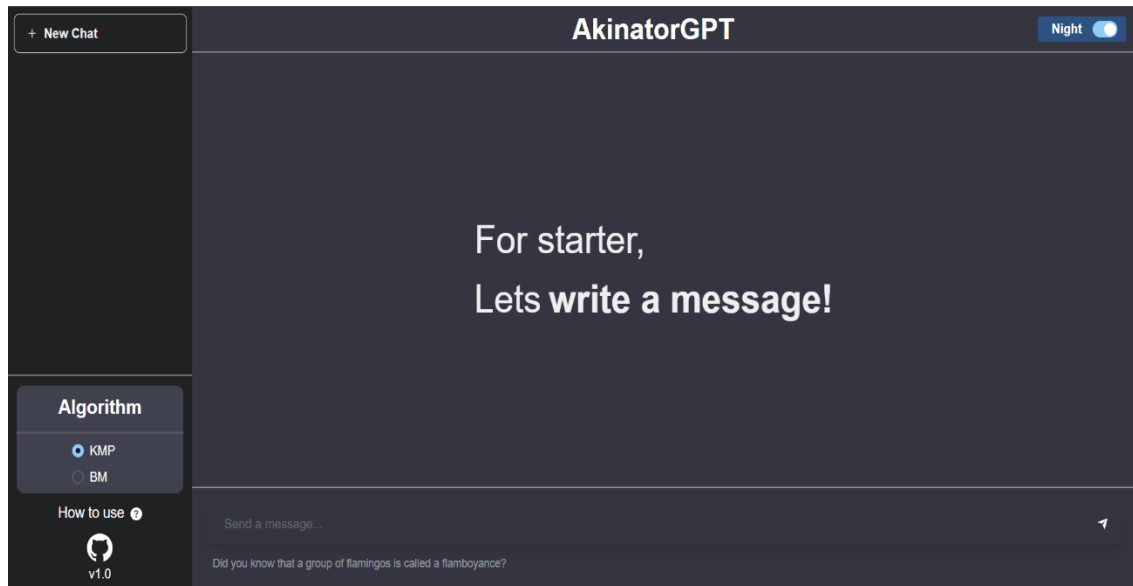
Setelah mengetikkan command diatas pada terminal, selanjutnya akan muncul tampilan sebagai berikut :

```
> akinator@0.0.0 dev
> vite

VITE v4.3.1 ready in 540 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h to show help
```

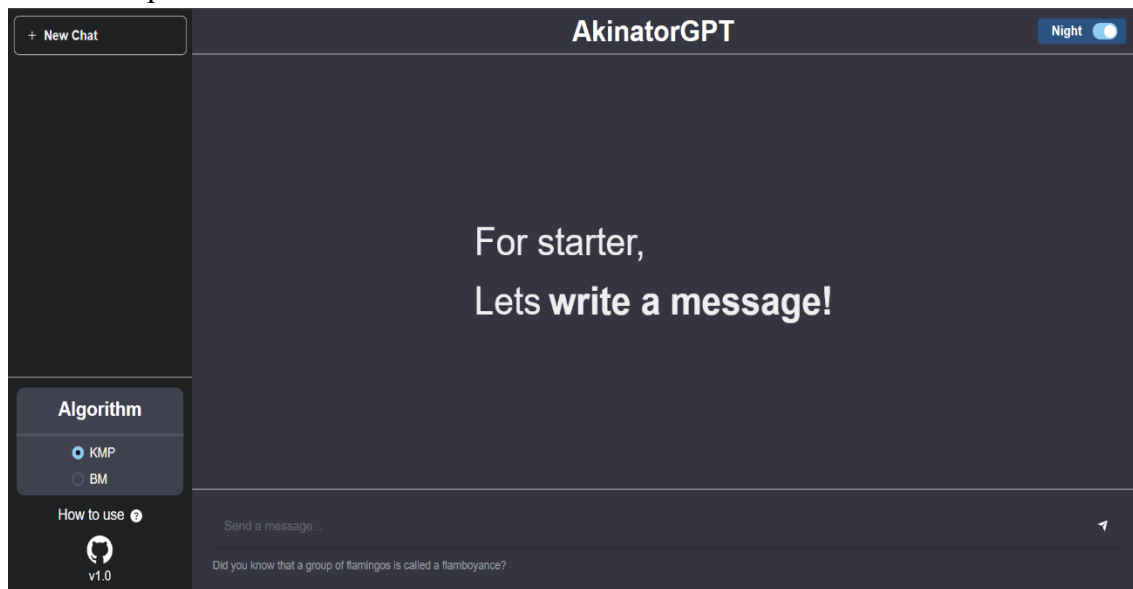
Setelah sedikit waktu berlalu Anda dapat menekan link yang ada dan dapat melihat tampilan aplikasi web serta menggunakan fitur – fitur yang ada.



4.2.3. Cara Menggunakan Program

Berikut adalah beberapa langkah yang dapat Anda gunakan sebaai acuan mengenai cara menggunakan program AkinatorGPT :

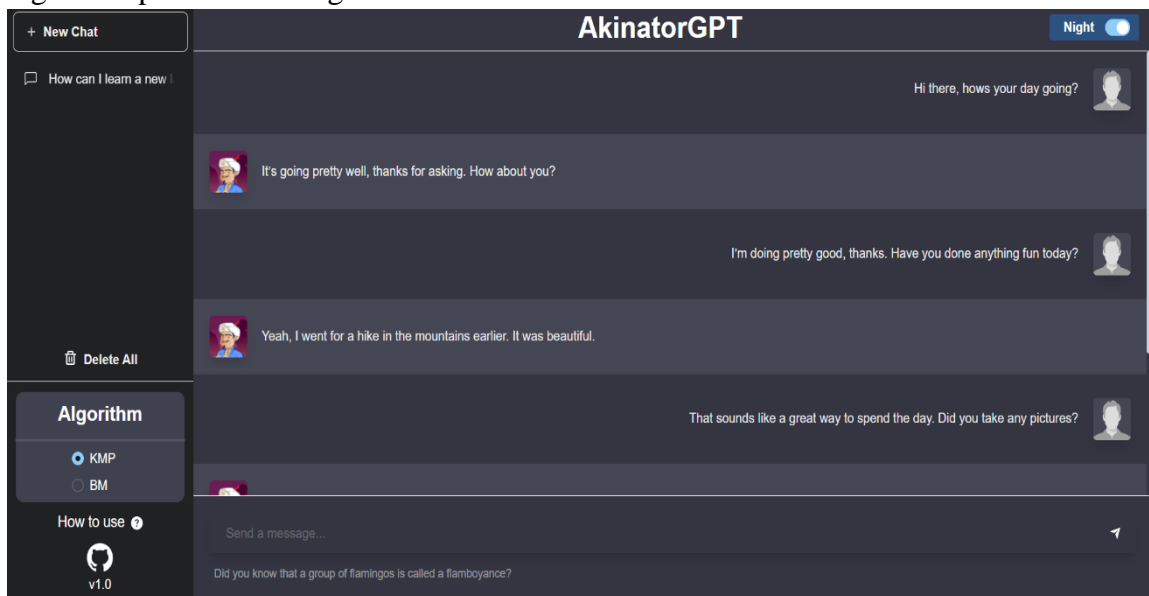
2. Buka link aplikasi web



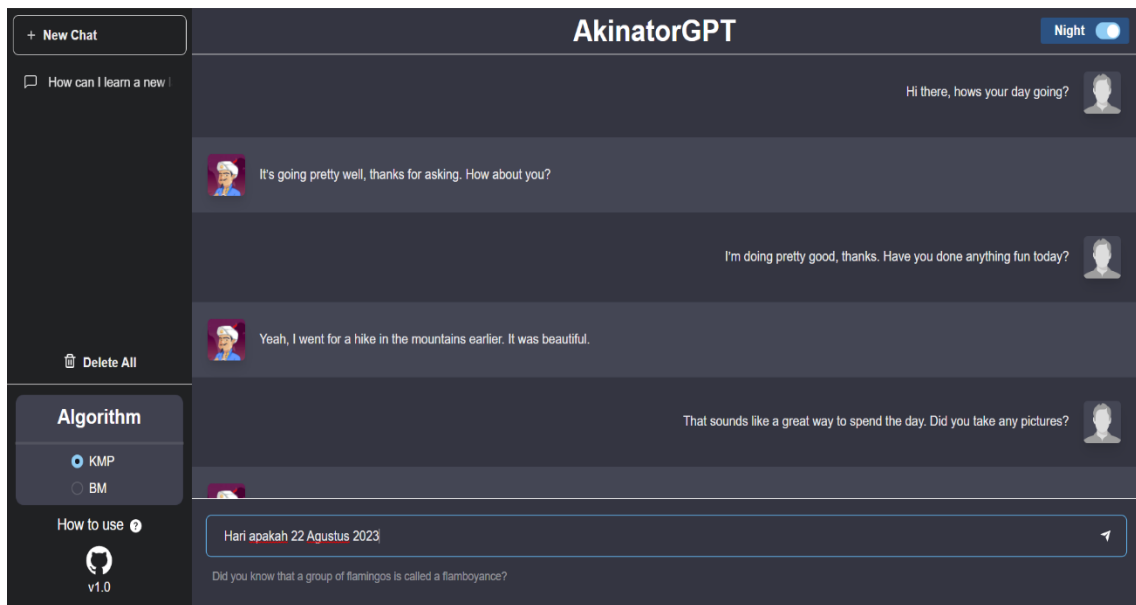
3. Klik tombol new chat pada bagian kiri atas



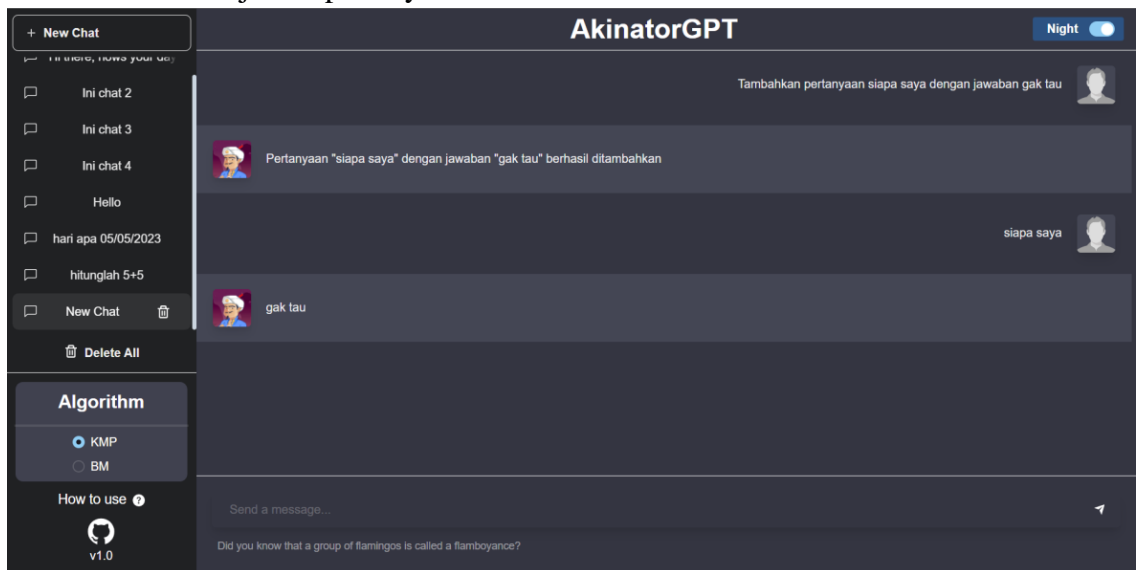
4. Akan muncul chat window baru dimana anda bisa memasukkan input dan memilih algoritma pencarian string



5. Anda dapat mencoba menanyakan sesuatu kepada chatbot



6. Chatbot akan menjawab pertanyaan Anda

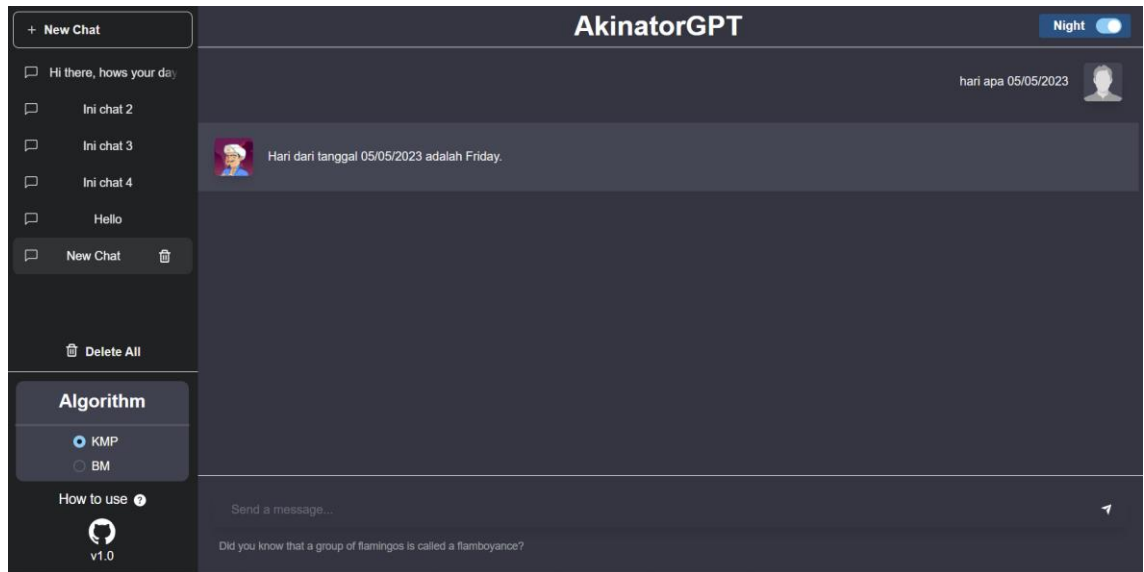


4.3 Hasil Pengujian

1. Tes Fitur Tanggal

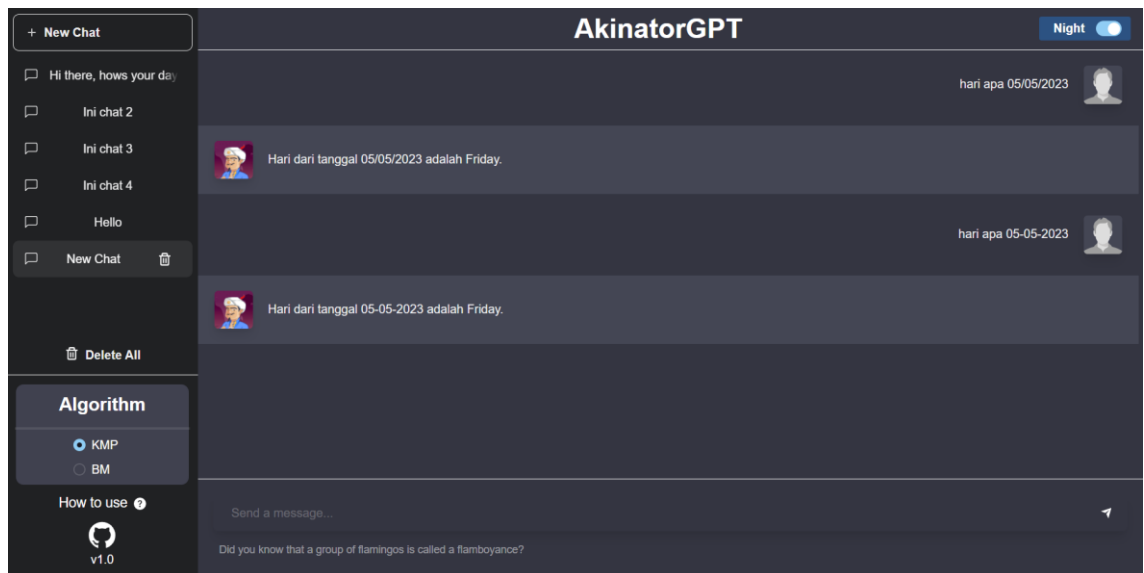
1.1. Hari apa 05/05/2023

Pada contoh dibawah fitur tanggal dapat membaca input user dengan jenis pemisah antar elemen tanggal berupa tanda miring



1.2. Hari apa 05-05-2023

Pada contoh dibawah fitur tanggal dapat membaca input user dengan jenis pemisah antar elemen tanggal berupa tanda hubung



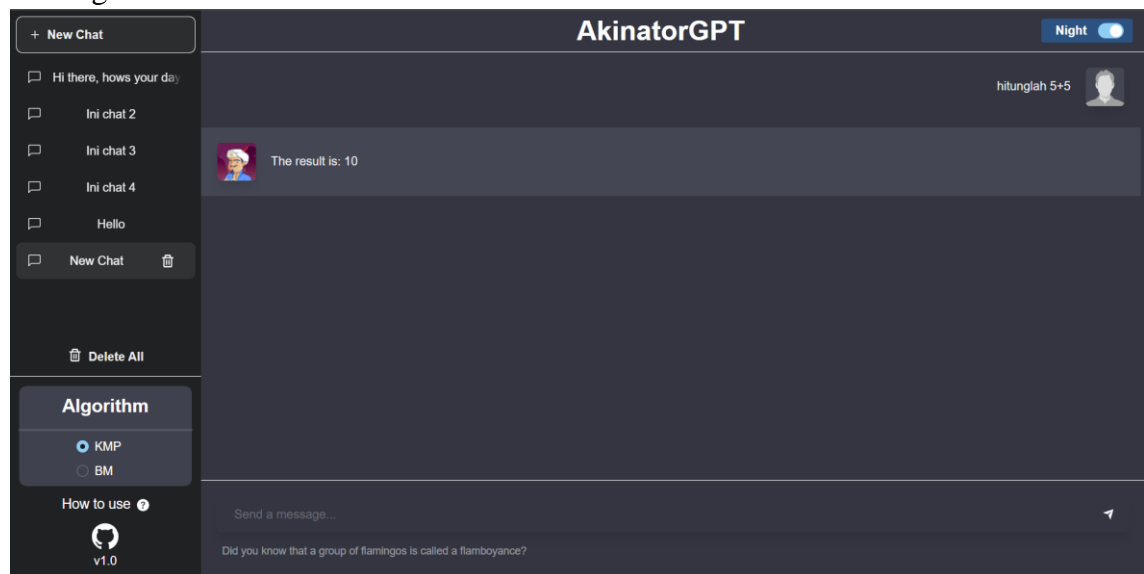
1.3. Hari apa 06 May 2023

Pada contoh dibawah fitur tanggal dapat membaca input user dengan jenis pemisah antar elemen tanggal berupa spasi dengan format bulan menggunakan nama bulan

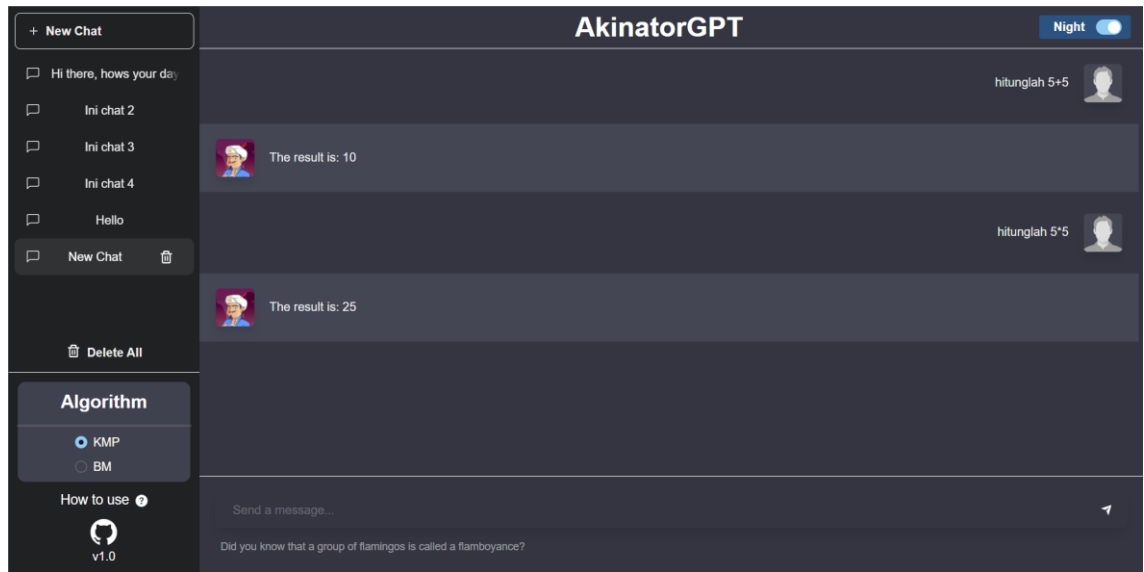


2. Tes Fitur Kalkulator

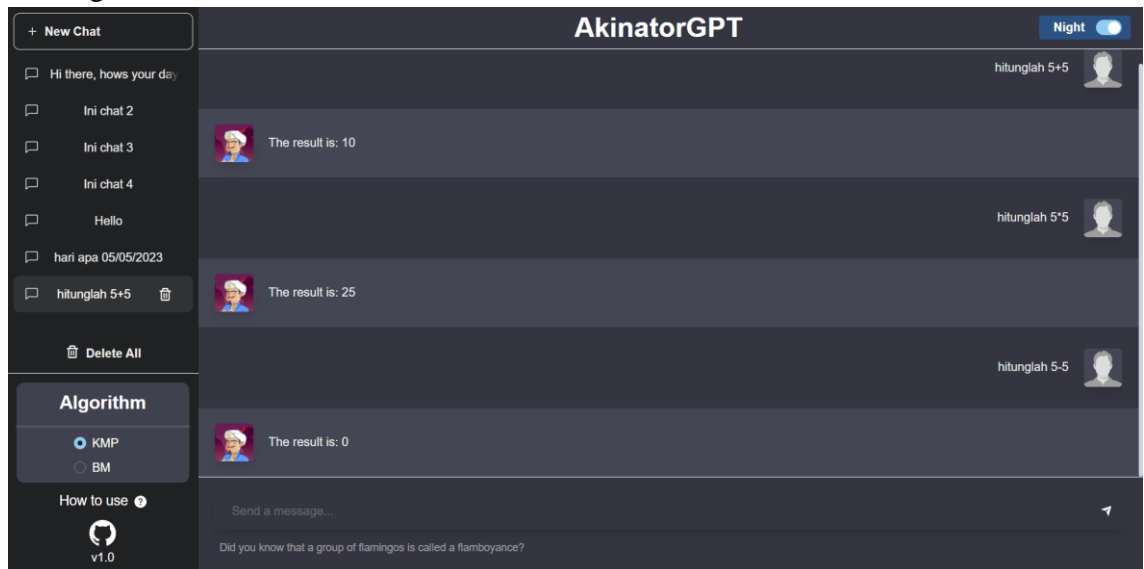
2.1. Hitunglah 5+5



2.2. Hitunglah 5*5



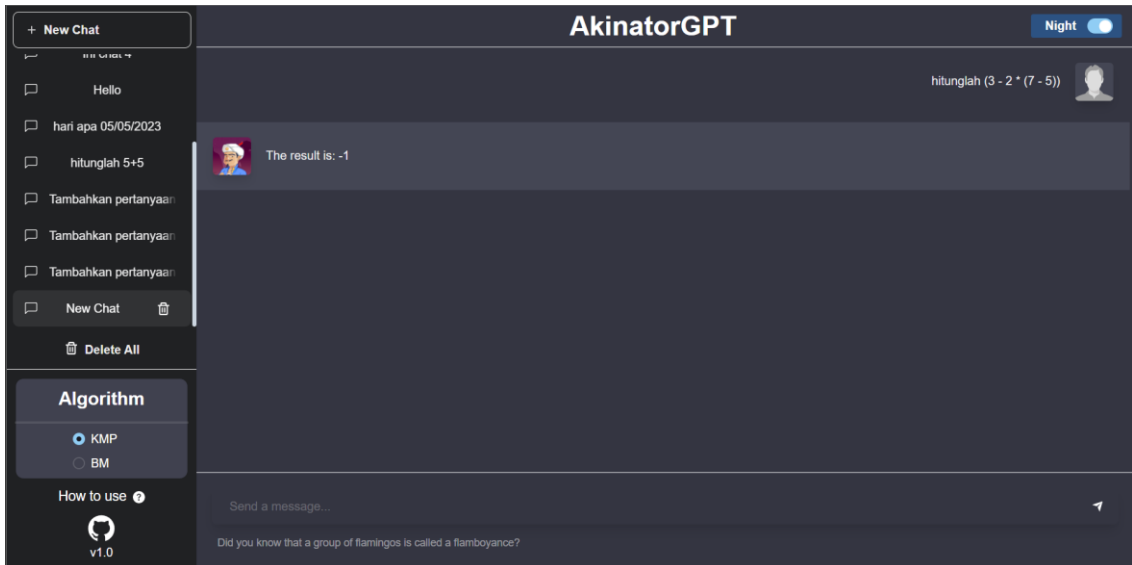
2.3. Hitunglah 5-5



2.4. Hitunglah 5/5



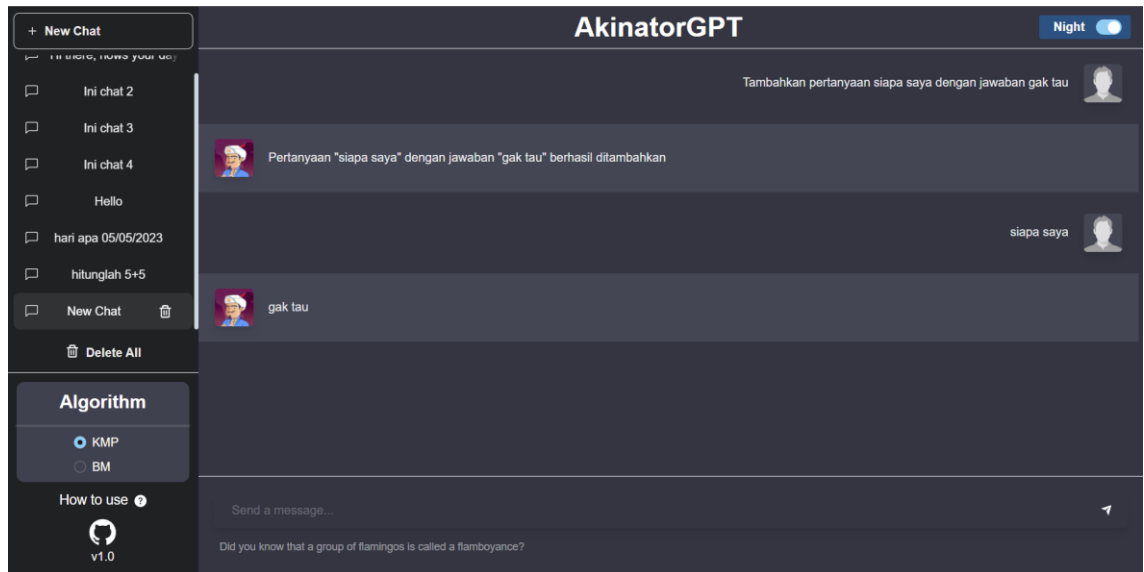
2.5 hitunglah $(3 - 2 * (7 - 5))$



3. Tes Fitur Menambah Pertanyaan dan Tanya Jawab

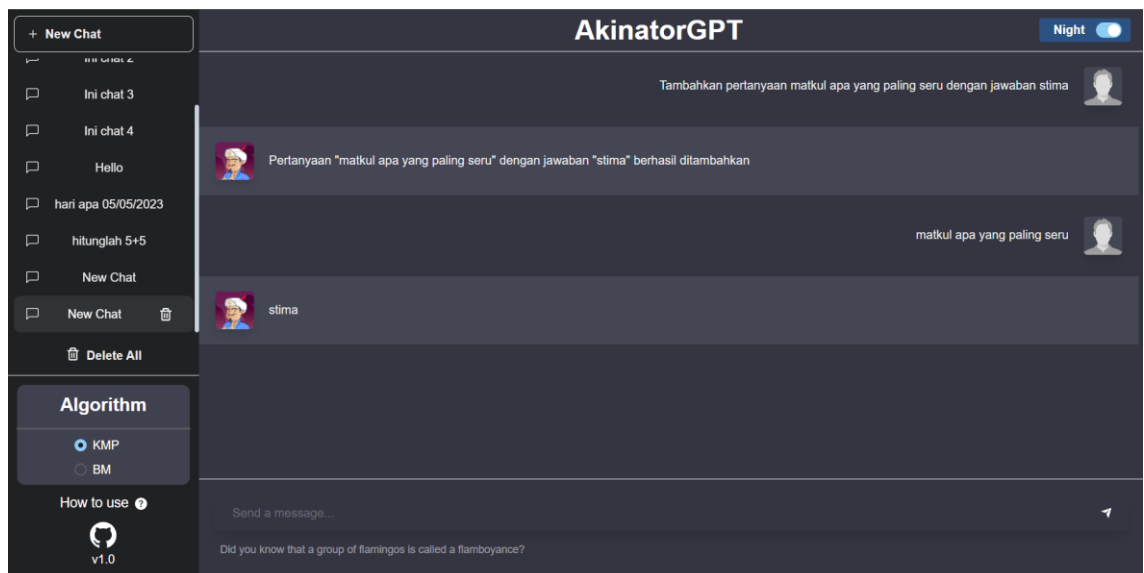
3.1. Siapa saya – gak tau

Pada contoh dibawah aplikasi dapat menambahkan pertanyaan dan jawaban dengan benar, lalu dapat menjawab pertanyaan yang baru diberikan dengan benar. Contoh pertanyaan adalah "Siapa saya" yang memiliki jawaban "gak tau"



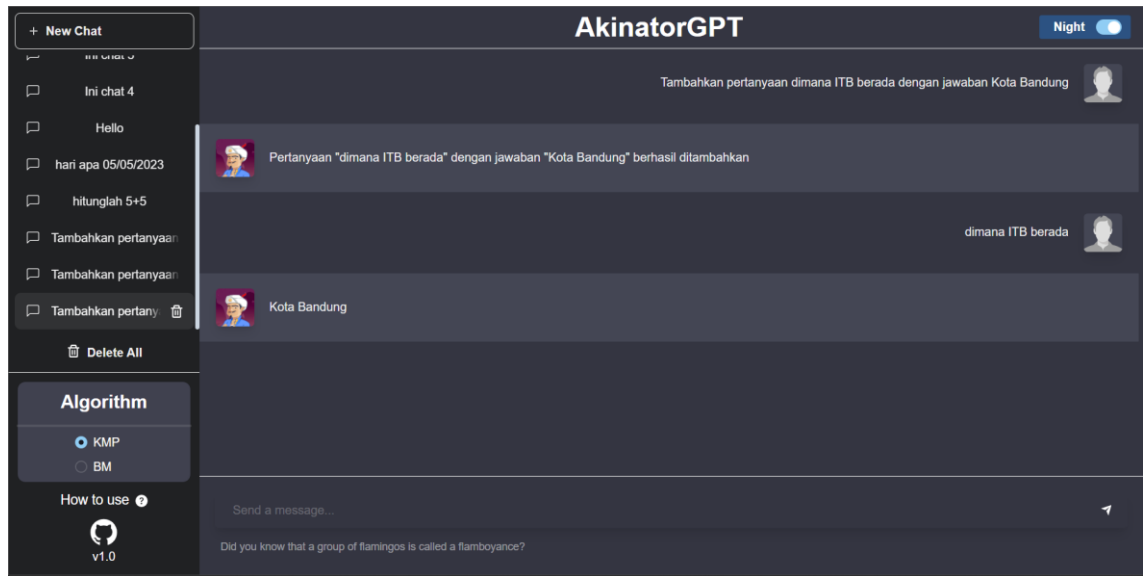
3.2. Matkul apa yang paling seru – stima

Pada contoh dibawah aplikasi dapat menambahkan pertanyaan dan jawaban dengan benar, lalu dapat menjawab pertanyaan yang baru diberikan dengan benar. Contoh pertanyaan adalah "Matkul apa yang paling seru" yang memiliki jawaban "stima"



3.3. Dimana ITB berada – Kota Bandung

Pada contoh dibawah aplikasi dapat menambahkan pertanyaan dan jawaban dengan benar, lalu dapat menjawab pertanyaan yang baru diberikan dengan benar. Contoh pertanyaan adalah "Dimana ITB Berada" yang memiliki jawaban "Kota Bandung"



4. Tes Fitur Menghapus Pertanyaan

4.1. Hapus pertanyaan siapa saya

Pada contoh dibawah aplikasi dapat menghapus pertanyaan yang ada sehingga tidak dapat dicari jawabannya pada database. Contoh pertanyaan adalah “Siapa saya”



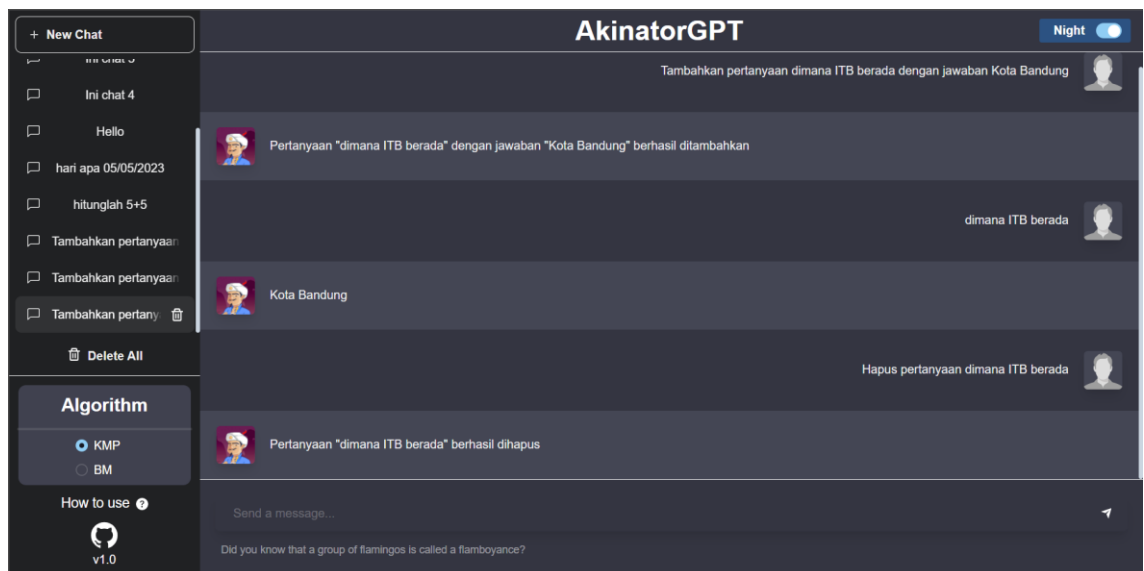
4.2. Hapus pertanyaan matkul apa yang paling seru

Pada contoh dibawah aplikasi dapat menghapus pertanyaan yang ada sehingga tidak dapat dicari jawabannya pada database. Contoh pertanyaan adalah “Matkul apa yang paling seru”



4.3. Hapus pertanyaan dimana ITB berada

Pada contoh dibawah aplikasi dapat menghapus pertanyaan yang ada sehingga tidak dapat dicari jawabannya pada database. Contoh pertanyaan adalah “Dimana ITB berada”



4.4 Analisis Hasil Pengujian

1. Tes Fitur Tanggal

Sesuai dengan hasil pengujian, fitur berjalan dengan baik. Fitur dapat menampilkan tanggal dengan benar.

2. Tes Fitur Kalkulator

Sesuai dengan hasil pengujian, fitur kalkulator dapat berjalan dengan baik dan mendapatkan nilai dari operasi aritmatika

3. Tes Fitur Menambah Pertanyaan
Sesuai dengan hasil pengujian, fitur menambah pertanyaan dapat berjalan dengan baik dan menambahkan pertanyaan ke database
4. Tes Fitur Menghapus Pertanyaan
Sesuai dengan hasil pengujian, fitur menghapus pertanyaan dapat berjalan dengan baik dan dapat menghapus pertanyaan dari database.
5. Tes Fitur Tanya Jawab
Sesuai dengan hasil pengujian, fitur tanya jawab dapat mencari string eksak menggunakan algoritma KMP maupun BM serta memberikan hasil paling mirip jika tidak ada yang sama

5 Kesimpulan dan Saran

5.1 Kesimpulan

Kami berhasil membuat sebuah aplikasi chatGPT berbasis web menggunakan Node js, MySQL, dan React. Aplikasi ini menggunakan algoritma Knuth-Morris-Pratt dan Booyer-Moore untuk melakukan pencocokan dua string serta regex untuk melakukan klasifikasi string. Aplikasi chatGPT 'Akinator' yang kami buat memiliki lima fitur utama yaitu fitur tanya jawab, kalkulator, mencari hari, serta menambah dan mengurangi pertanyaan dari database.

Aplikasi akan menerima input user lalu mengklasifikasi fitur yang user inginkan menggunakan regex, lalu memanggil fungsi – fungsi yang ada untuk mengembalikan output yang diharapkan

5.2 Saran

Aplikasi berbasis web yang kami buat sudah berjalan dengan baik, namun masih bisa ditingkatkan lagi di bagian user interface agar lebih menarik serta efisiensi algoritma agar aplikasi dapat berjalan lebih cepat

5.3 Komentar

Tugas besar ini sangat mengasah kemampuan kami dalam pembuatan aplikasi web beserta pengetahuan kami mengenai beberapa bahasa dan framework yang ada. Terdapat beberapa kekurangan yang ada dalam aplikasi kami tapi seluruh fitur yang ada dapat berjalan dengan baik

Referensi :

- <https://www.geeksforgeeks.org/mysql-introduction/>
- <https://www.geeksforgeeks.org/nodejs/>
- <https://www.knowledgehut.com/blog/web-development/chakra-ui-in-react>
- <https://www.geeksforgeeks.org/introduction-to-typescript/>
- [https://www3.ntu.edu.sg/home/ehchua/programming/howto/Regexe.html#:~:text=A%20Regular%20Expression%20\(or%20Regex,strings%20and%20rejects%20the%20rest.](https://www3.ntu.edu.sg/home/ehchua/programming/howto/Regexe.html#:~:text=A%20Regular%20Expression%20(or%20Regex,strings%20and%20rejects%20the%20rest.)
- <https://www.trivusi.web.id/2022/04/apa-itu-algoritma-levenshtein-distance.html>
- <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>
-

Link Penting :

Github :

[Frontend](#)

[Backend](#)

Demo : bit.ly/VideoAkinatorGegeGeming

