

Tugas Besar III IF2211 Strategi Algoritma
Semester II Tahun 2022/2023
**Penerapan String Matching dan Regular Expression dalam
Pembuatan ChatGPT Sederhana**

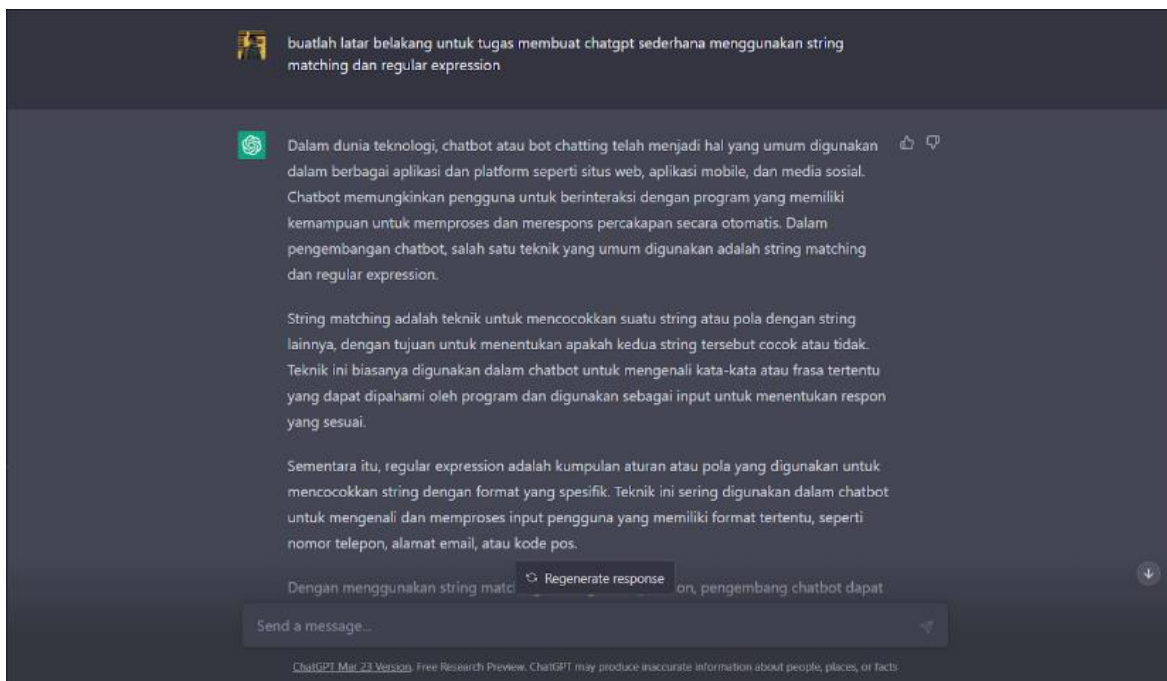
Batas pengumpulan : 5 Mei 2023 pukul 23.59 WIB

Arsip pengumpulan :

- *Source* program yang bisa dijalankan disertai *README.md*
- Laporan (*soft copy*)

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.

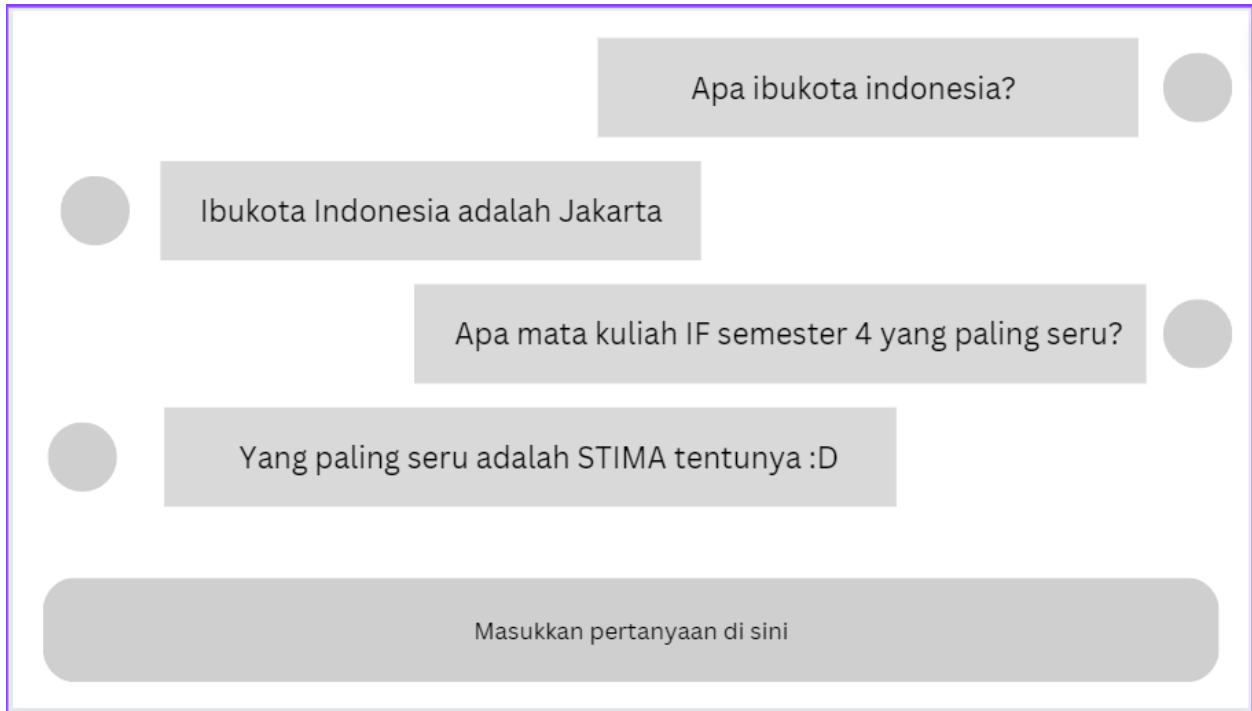
4. Tambah pertanyaan dan jawaban ke database

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.

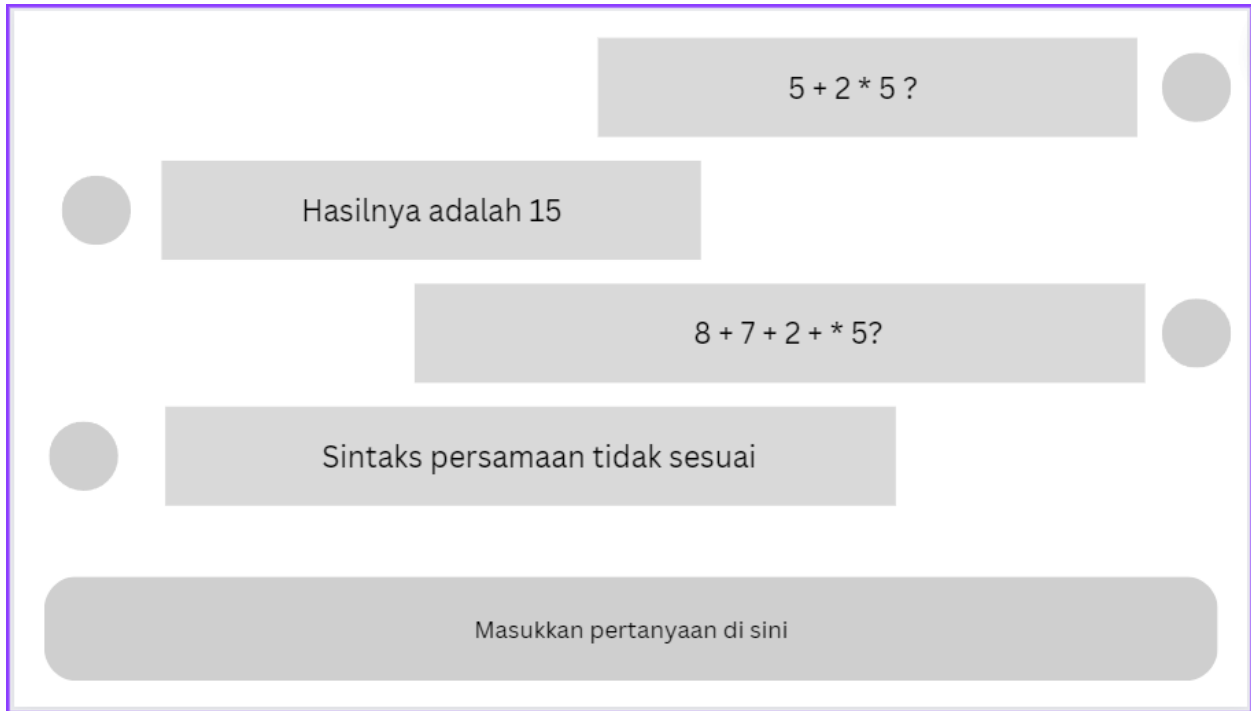
Klasifikasi dilakukan menggunakan **regex** dan terklasifikasi layaknya bahasa sehari - hari. Algoritma string matching KMP dan BM digunakan untuk klasifikasi query teks. Tersedia toggle untuk memilih algoritma KMP atau BM. Semua pemrosesan respons dilakukan pada sisi **backend**. Jika ada pertanyaan 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)**



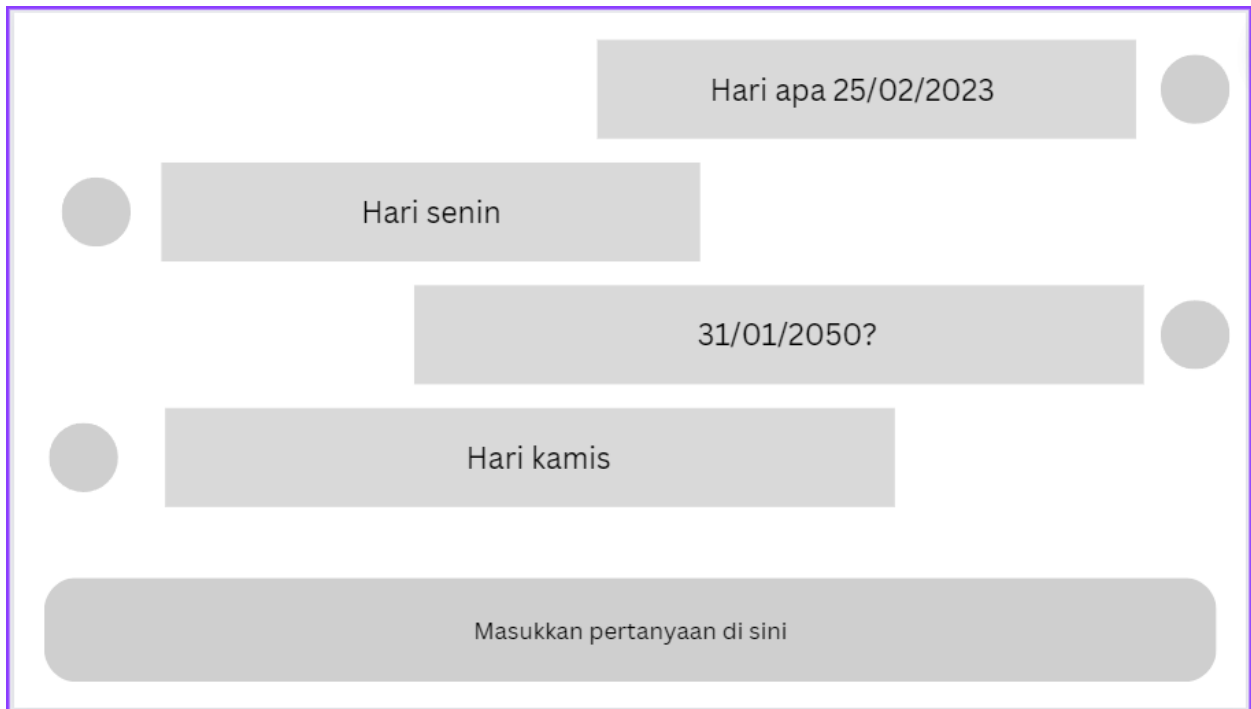
Gambar 2. Ilustrasi Fitur Pertanyaan teks kasus exact



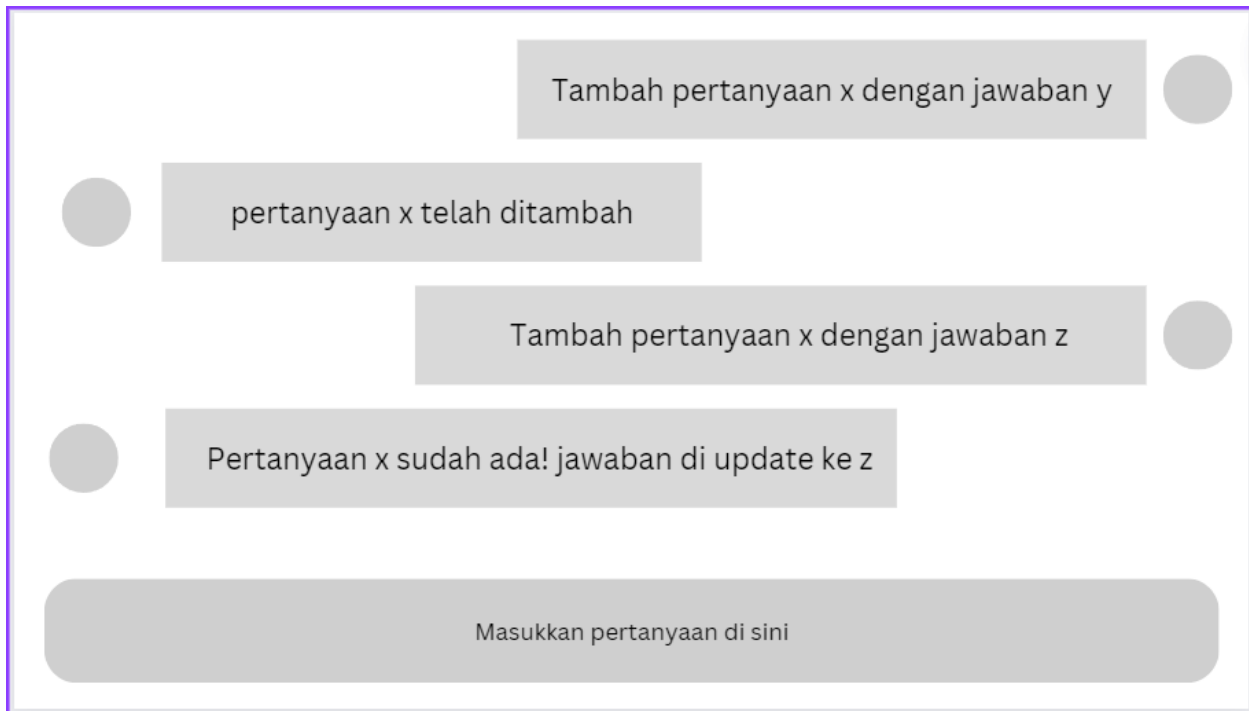
Gambar 3. Ilustrasi Fitur Pertanyaan teks kasus tidak exact



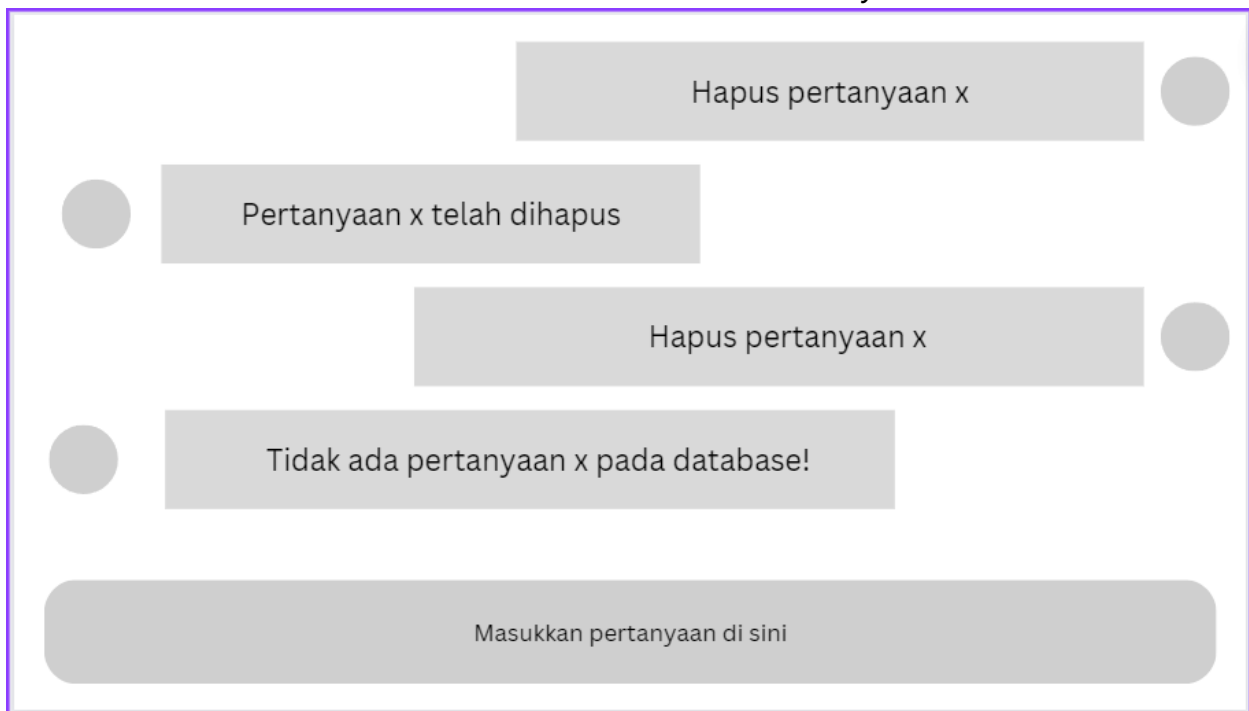
Gambar 4. Ilustrasi Fitur Kalkulator



Gambar 5. Ilustrasi Fitur Tanggal



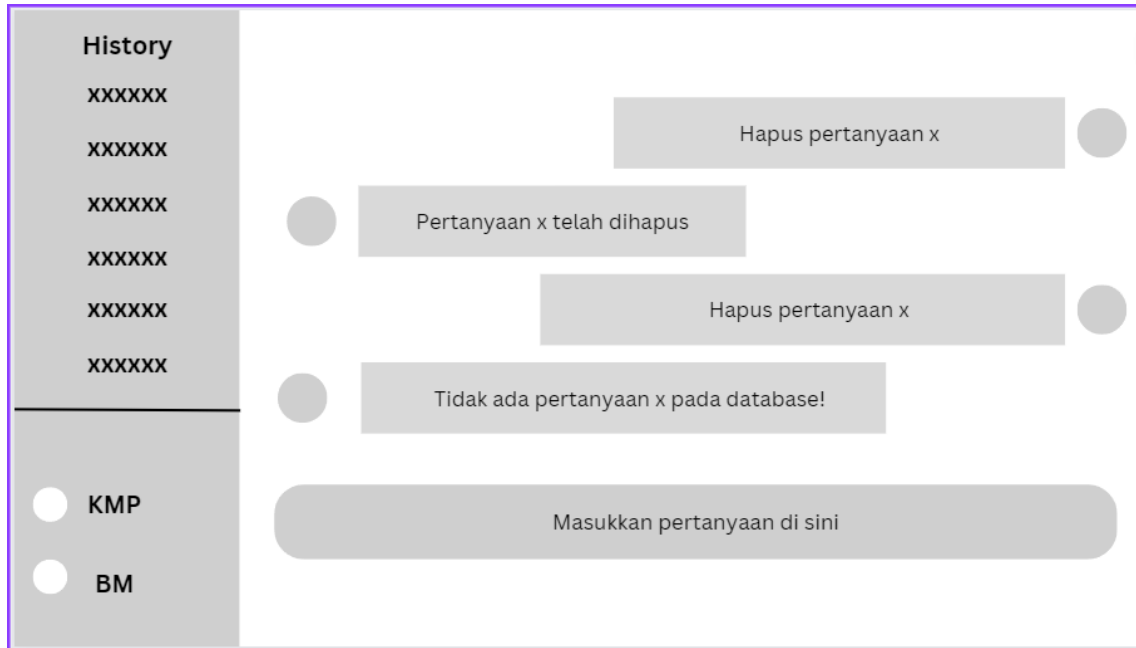
Gambar 6. Ilustrasi Fitur Tambah Pertanyaan



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").

Lain-lain:

1. Anda dapat menambahkan fitur-fitur lain yang menunjang program yang anda buat (unsur kreativitas).

2. Tugas dikerjakan berkelompok, minimal 2 orang dan maksimal 3 orang, boleh lintas kelas namun **tidak boleh sekelompok** dengan **orang yang sama dengan tubes ataupun tucil stima sebelumnya**.
3. Semua kelompok harap mengisi data kelompok mereka pada link <https://bit.ly/KelompokTubes3Stima>
4. Batas akhir pengisian anggota kelompok adalah **16 April, Pukul 22:11..** Mahasiswa yang belum mendapatkan kelompok setelah tanggal ini akan diacak kelompoknya.
5. Anda harus membuat aplikasi dan program ini sendiri kecuali library regex, tetapi belajar dari contoh-contoh program serupa yang sudah ada tidak dilarang (tidak boleh melakukan plagiasi source code dari program orang lain). Program harus dibuat sendiri, tidak boleh sama dengan teman.
6. Program harus modular dan mengandung komentar yang jelas.
7. Dilarang menggunakan kode program yang diunduh dari Internet. Mahasiswa harus membuat program sendiri, tetapi belajar dari program yang sudah ada tidak dilarang.
8. Batas akhir pengumpulan tugas adalah **5 Mei, Pukul 23:59..** Keterlambatan dalam mengumpulkan akan diberi penalti pengurangan skor yang cukup signifikan.
9. Semua pertanyaan menyangkut tugas ini dapat dikomunikasikan lewat QnA yang bisa diakses pada bit.ly/TugasStimaQnA
10. **Bonus (maksimal 10 poin):**
 - a. Mendeploy aplikasi web yang telah dibangun (hosting provider dibebaskan). Deployment website harus dipertahankan sampai demo tugas besar.
 - b. Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke Youtube. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Pada waktu demo aplikasi di depan asisten, mahasiswa mengakses video Youtube tersebut dan memutarinya di depan asisten. Beberapa contoh video tubes tahun-tahun sebelumnya dapat dilihat di YouTube dengan menggunakan kata kunci “Tubes Stima”, “Tugas besar stima”, “strategi algoritma”, dll.
11. Demo akan dilakukan, tunggu informasi lanjut setelah waktu pengerjaan tugas berakhir.
12. Setiap anggota kelompok harus memahami seluruh program, termasuk bagian yang bukan bagian mereka.
13. Program disimpan dalam folder **Tubes3_NIM** (jika menggunakan *repository* dapat mengubah nama *repository*-nya) dengan NIM merupakan NIM anggota terkecil. Berikut merupakan struktur dari isi folder tersebut.
 - a. Folder **src** berisi **source code**.
 - b. Folder **doc** berisi **laporan tugas besar** dengan format **nama_kelompok.pdf**
 - c. README selengkap mungkin (dapat menjelaskan cara menjalankan *backend* dan *frontend* di lokal). Referensi README dapat diakses pada <https://github.com/ritaly/README-cheatsheet> atau referensi lain yang serupa.

Isi laporan:

- **Cover:** Cover laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo “gajah” ganesha.
- **Bab 1:** Deskripsi tugas (dapat menyalin spesifikasi tugas ini).

- **Bab 2:** Landasan Teori.
 - Deskripsi singkat algoritma KMP, BM, dan Regex
 - Penjelasan singkat mengenai aplikasi web yang dibangun
- **Bab 3:** Analisis Pemecahan Masalah.
 - Langkah penyelesaian masalah setiap fitur
 - Fitur fungsional dan arsitektur aplikasi web yang dibangun
- **Bab 4:** Implementasi dan pengujian.
 - Spesifikasi teknis program (struktur data, fungsi, prosedur yang dibangun)
 - Penjelasan tata cara penggunaan program (interface program, fitur-fitur yang disediakan program, dan sebagainya)
 - Hasil pengujian (*screenshot* antarmuka dan skenario yang memperlihatkan berbagai kasus yang mencakup seluruh fitur pada aplikasi anda)
 - Analisis hasil pengujian
- **Bab 5:** Kesimpulan, saran, dan komentar/refleksi tentang tugas besar 3 ini.
- Daftar Pustaka.

Keterangan laporan:

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

Penilaian:

1. **Bagian 1** : Laporan (25%)
 - a. Langkah penyelesaian masalah setiap fitur (10%)
 - b. Hasil pengujian dan analisis algoritma (10%)
 - c. Komponen-komponen lain dalam laporan (5%)
2. **Bagian 2** : Implementasi Program (75%)
 - a. Kebenaran program (30%)
 - b. Pemahaman terhadap cara kerja program (25%)
 - c. Interface, Features, dan Unsur Kreativitas (20%)
3. **Bagian 3** : Bonus (10%)
 - a. Melakukan *deployment* ke dalam website
 - b. Membuat video demonstrasi program

Referensi Belajar:

- [Dasar-dasar Go](#)
- [Framework Backend Go Echo](#)
- [Framework Backend Go Fiber](#)
- [Framework Backend Go Gin](#)
- [Framework Backend Node.js Express](#)
- [Framework Backend Node.js Fastify](#)
- [Framework Backend Node.js Hapi](#)
- [Framework Frontend Vue.js](#)
- [Framework Frontend Angular](#)
- [Framework Frontend React](#)
- [Framework Frontend Next.js](#)
- [Connect to MySQL from Node.js](#)
- [Connect to MySQL from Node.js using Sequelize](#)

- [Connect to SQL from Go](#)
- [API Call using Axios](#)
- [Hamming Distance Explanation](#)
- [Levenshtein Distance Explanation](#)
- [LCS Explanation](#)

--- Selamat Mengerjakan! ---

“Mendedikasikan tubes ketiga demi life saviour semesterku: GPT <3”
– Saul –

“X: Tubes 3 stima bakal gampang kan karena cuman *string matching*?
chat-Se-Ti-Ma: Yakin mas? *Use case*-nya apa mas?”
– Steven –

“Gampang kan cuma nyocok nyocokin kata”
– Dimas –

“Wajib dibales ya jangan cuma diread”
– Willy –

“Selamat meninggal ^ _ ^”
– Bryan –

“Jangan dikerjain ChatGPT ya :D”
– Ziyad –

“Yu bisa yu, dikit lagii”
–Vionie –

“Menarik ngga sih tubes ini :v”
–Angel –