

Penerapan String Matching dan Regular Expression dalam Pembuatan ChatGPT Sederhana

Diajukan sebagai pemenuhan tugas besar III.



Oleh:

1. 13521077 - Husnia Munzayana
2. 13521111 - Tabitha Permalla
3. 13521130 - Althaaf Khasyi Atisomya

Dosen Pengampu :

Dr. Ir. Rinaldi Munir, M.T

Dr. Nur Ulfa Mauldievi, S.T, M.Sc

Ir. Rila Mandala, M.Eng, Ph.D

IF2211 - Strategi Algoritma

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

2022

DAFTAR ISI

DAFTAR ISI	2
BAB 1	
DESKRIPSI TUGAS	3
BAB 2	
LANDASAN TEORI	4
2.1. Algoritma KMP (Knuth-Morris-Pratt)	4
2.2. Algoritma BM (Booyer-Moore)	4
2.3. Regular Expression	5
2.4. Web app	6
BAB 3	
ANALISIS PEMECAHAN MASALAH	7
3.1. Langkah Penyelesaian Masalah	7
3.1.1. Penentuan Fitur	7
3.1.2. Date	7
3.1.3. Calculator	8
3.1.4. Tambah/Hapus Pertanyaan	8
3.1.5. ChatBot	8
3.2. Fitur Fungsional dan Arsitektur	9
3.2.1. Fitur Fungsional	9
3.2.2. Arsitektur Program	9
BAB 4	
IMPLEMENTASI DAN PENGUJIAN	10
4.1. Spesifikasi Teknis Program	10
4.2. Tata Cara Penggunaan Program	10
4.2.1. Tata cara menjalankan backend	10
4.2.2. Tata cara menjalankan frontend	10
4.3. Hasil Pengujian	11
4.3.1. Fitur Date	11
4.3.2. Fitur Kalkulator	12
4.3.3. Fitur Tambah/Hapus Pertanyaan	13
4.3.3.1. Tambah Pertanyaan	13
4.3.3.2. Ubah Jawaban Pertanyaan	13
4.3.3.3. Hapus Pertanyaan	14
4.3.4. Fitur ChatBot	15
4.3.4.1. Dengan Algoritma KMP	15
4.3.4.2. Dengan Algoritma BM	16
4.4. Analisis Hasil Pengujian	17
BAB 5	
KESIMPULAN DAN SARAN	19
5.1. Kesimpulan	19
5.2. Saran	19
5.3. Komentar/Refleksi	19
DAFTAR PUSTAKA	20
LAMPIRAN	21

BAB 1

DESKRIPSI TUGAS

Dalam tugas besar ini, program yang dibangun adalah sebuah aplikasi *ChatGPT* sederhana dengan mengaplikasikan pendekatan QA yang paling sederhana. 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.

ChatGPT sederhana ini 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 diperbarui.

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 fitur sesuai masukan pengguna 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.

BAB 2

LANDASAN TEORI

2.1. Algoritma KMP (Knuth-Morris-Pratt)

Algoritma KMP adalah salah satu algoritma yang digunakan dalam proses pencocokan string. Algoritma ini dinamai berdasarkan nama ketiga penemunya: Donald Knuth, Vaughan Pratt, dan James Morris.

Algoritma KMP mencari pola di dalam teks dalam urutan kiri ke kanan, seperti pada algoritma *brute force*. Akan tetapi, algoritma KMP melakukan *shift* pada pola secara lebih intelijen dan efisien dibandingkan algoritma *brute force*. Algoritma KMP bekerja dengan mencocokkan pola yang ingin dicari dengan teks secara efisien tanpa mengulangi pencocokan yang tidak perlu. Hal ini dilakukan dengan memanfaatkan informasi yang diperoleh dari pencocokan sebelumnya untuk menentukan posisi selanjutnya dalam pencocokan.

Algoritma KMP menggunakan *degenerating property* (pola yang memiliki sub-pola yang sama yang muncul lebih dari sekali di dalam pola tersebut) dari pola tersebut dan meningkatkan kompleksitas terburuknya menjadi $O(n)$. Ide dasar dari algoritma KMP adalah sebagai berikut: setiap kali ditemukan *mismatch* (setelah sejumlah *match*), kita sudah mengetahui sebagian karakter di dalam teks selanjutnya. Kita mengambil keuntungan dari informasi ini untuk menghindari *matching* karakter yang kita ketahui pasti akan sesuai.

2.2. Algoritma BM (Booyer-Moore)

Algoritma BM (Booyer-Moore) adalah algoritma salah satu alternatif algoritma yang digunakan untuk pencocokan string. Algoritma ini ditemukan oleh Bob Boyer dan J. Strother Moore pada tahun 1994. Algoritma ini digunakan untuk melakukan pencarian *exact matching* pada string.

Algoritma Boyer-Moore didasarkan pada dua Teknik, yaitu *looking-glass technique* dan *character-jump technique*. The *looking-glass technique* dilakukan dengan mencocokkan suatu karakter pada pattern dengan karakter pada text, dan jika kedua karakter sama, maka akan dilakukan pergeseran mundur pada kedua string hingga string pada text selesai dicari atau telah ditemukan substring yang cocok. Berbeda dengan *the looking-glass technique*, pada *the character-jump technique* dilakukan ketika ditemukan karakter yang berbeda pada kedua string. Pergeseran dengan *character-jump technique* terbagi menjadi 3 kemungkinan:

Misalkan karakter ke- i pada teks atau $T[i]$ adalah x dan $P[j] \neq T[i]$, maka

Kasus 1 : Jika pattern P mengandung karakter x , maka akan dilakukan pergeseran ke kanan untuk menyajarkan kemunculan terakhir x pada pattern P dengan $T[i]$

Kasus 2 : Jika pattern P mengandung karakter x , tetapi karakter x tersebut berada setelah $P[j]$, maka dilakukan pergeseran sebanyak satu karakter ke kanan sehingga $P[j]$ sejajar dengan $T[i+1]$

Kasus 3: Jika kasus 1 dan kasus 2 tidak dapat diterapkan, maka lakukan pergeseran sehingga $P[0]$ atau karakter pertama pattern sejajar dengan $T[i+1]$.

Untuk membantu pergeseran pada kasus 1, terdapat *pre-processing* pada algoritma Boyer-Moore, yaitu menentukan *Last Occurrence Function*. *Last Occurrence Function* adalah fungsi untuk menentukan letak kemunculan terakhir setiap karakter yang terdapat pada pattern. Dalam kasus terburuknya, kompleksitas waktu algoritma Boyer-Moore adalah $O(n m + A)$ dimana A adalah kompleksitas ukuran text. Algoritma ini lebih efektif untuk digunakan saat text yang kita gunakan cukup besar. Namun, algoritma ini secara signifikan lebih cepat dibandingkan dengan algoritma Brute Force.

2.3. Regular Expression

Selain dengan dua algoritma di atas, proses pencocokan teks atau *string matching* dapat dilakukan dengan *regular expression*. *Regular Expression* (Regex) merupakan notasi yang mendefinisikan pola sebuah kata/pola tertentu pada text. Konsep ini diperkenalkan oleh Stephen Cole Kleene pada tahun 1951. Pada masa itu, Stephen Cole Kleene sedang memformulasikan definisi tentang bahasa formal.

Notasi umum *regular expression* antara lain:

Options Quick Reference	
.	Any character except newline.
\.	A period (and so on for *, \(), \), etc.)
^	The start of the string.
\$	The end of the string.
\d,\w,\s	A digit, word character [A-Za-z0-9_], or whitespace.
\D,\W,\S	Anything except a digit, word character, or whitespace.
[abc]	Character a, b, or c.
[a-z]	a through z.
[^abc]	Any character except a, b, or c.
aa bb	Either aa or bb.
?	Zero or one of the preceding element.
*	Zero or more of the preceding element.
+	One or more of the preceding element.
{n}	Exactly n of the preceding element.
{n,}	n or more of the preceding element.
{m, n}	Between m and n of the preceding element.
??,*?,+?,{n}, etc.	Same as above, but as few as possible.
(expr)	Capture expr for use with \1, etc.
(?:expr)	Non-capturing group.
(?=expr)	Followed by expr.
(?!expr)	Not followed by expr.

[Near-complete reference](#)

2.4. Web app

Web-Application atau WebApp adalah perangkat lunak yang diakses melalui web browser, seperti Google Chrome, Mozilla Firefox, Microsoft Edge, maupun Safari. Pada Aplikasi ini, program disimpan di server jarak jauh (remote) dan diakses melalui jaringan internet. Untuk mengaksesnya, client tidak perlu mengunduh atau menginstall aplikasi tertentu.

Terdapat 3 komponen utama yang diperlukan dalam pengembangan WebApp antara lain web server, application server, dan database. Web server digunakan untuk mengelola permintaan yang datang dari client atau pengguna untuk kemudian permintaan tersebut dieksekusi oleh application server, sedangkan database digunakan untuk menyimpan informasi yang diperlukan.

<<foto app kita berjalan di browser>>

BAB 3

ANALISIS PEMECAHAN MASALAH

3.1. Langkah Penyelesaian Masalah

3.1.1. Penentuan Fitur

Penentuan fitur pada ChatBot dilakukan dengan identifikasi menggunakan regular expression. Pengecekan dilakukan dengan urutan sebagai berikut.

Pertama-tama, masukan pengguna diperiksa apakah memenuhi regex tanggal. Jika memenuhi, maka ChatBot akan melaksanakan fitur Date. Jika tidak, pengecekan dilanjutkan untuk regex kalkulator. Selanjutnya dilakukan pemeriksaan untuk penambahan atau penghapusan pertanyaan dari database. Apabila masukan pengguna tidak memenuhi regex untuk tanggal, kalkulator, tambah pertanyaan, atau hapus pertanyaan, maka ChatBot akan memasuki fitur utamanya, yaitu menjawab pertanyaan.

3.1.2. Date

Pencarian hari pada tanggal tertentu diselesaikan dengan menghitung banyaknya selisih hari antara tanggal yang dicari dengan tanggal 01/01/0000. Menurut kalender Gregorian, 1 Januari 0001 jatuh pada hari Sabtu. Dengan melakukan perhitungan secara manual, didapatkan Tanggal 1 Januari 0000 jatuh pada hari minggu. Hasil perhitungan selisih hari tersebut kemudian akan dilakukan modulo 7 dan dari hasil modulo tersebut akan ditentukan hari yang sesuai dengan aturan sebagai berikut:

- Jika selisih mod 7 = 0, maka pada tanggal tersebut merupakan hari Minggu.
- Jika Selisih mod 7 = 1, maka pada tanggal tersebut merupakan hari Senin.
- Jika Selisih mod 7 = 2, maka pada tanggal tersebut merupakan hari Selasa.
- Jika Selisih mod 7 = 3, maka pada tanggal tersebut merupakan hari Rabu.
- Jika Selisih mod 7 = 4, maka pada tanggal tersebut merupakan hari Kamis.
- Jika Selisih mod 7 = 5, maka pada tanggal tersebut merupakan hari Jumat.
- Jika Selisih mod 7 = 6, maka pada tanggal tersebut merupakan hari Sabtu.

3.1.3. Calculator

Validasi dan perhitungan ekspresi dilakukan dengan membaca setiap karakter pada ekspresi dan mengkategorikannya menjadi operan dan operator. Dalam hal ini, ekspresi matematika yang diterima merupakan operasi dalam notasi postfix. Setiap angka akan dimasukkan kedalam array operands, sedangkan setiap operator, yaitu +, -, *, / akan dimasukkan kedalam array operators.

Jika ditemukan tanda kurung buka "(" maka setiap pembacaan operan dan operator akan dimasukkan kedalam array yang sesuai hingga ditemukan tanda kurung tutup ")". Ketika tanda kurung tutup ditemukan, maka akan dilakukan evaluasi terhadap ekspresi yang terdapat pada tanda kurung atau hingga tanda kurung buka "(" ditemukan pada array.

Proses evaluasi dilakukan dengan mencari operator dengan *precedence* yang lebih tinggi untuk terlebih dahulu dikalkulasi. Hasil kalkulasi tersebut kemudian dimasukkan kembali pada array operands. Proses dilakukan secara berulang hingga semua operator sudah dieksekusi.

3.1.4. Tambah/Hapus Pertanyaan

Fitur penambahan pertanyaan dilakukan dengan pemanggilan *query* ke *database*. Pertama-tama dilakukan pengecekan apakah pertanyaan sudah ada di *database*. Apabila pertanyaan belum ada, maka pertanyaan ditambahkan ke *database*. Jika pertanyaan sudah ada, maka dilakukan *update* terhadap jawaban dari pertanyaan tersebut.

Fitur hapus pertanyaan juga dilakukan dengan pemanggilan *query* ke *database*. Jika pertanyaan berhasil dihapus, maka akan memberikan pesan berhasil dihapus. Jika pertanyaan tidak ada pada *database*, maka tidak terjadi perubahan pada *database* dan memberikan pesan bahwa pertanyaan tidak ada di database.

3.1.5. ChatBot

ChatBot adalah program komputer yang didesain untuk mensimulasikan percakapan dengan pengguna. Pada program ini, user dapat memasukkan string pertanyaan untuk kemudian pertanyaan tersebut dijawab oleh komputer. Pada implementasinya, database sudah menyimpan berbagai pasangan pertanyaan-jawaban, sehingga ketika user memasukkan pertanyaan tertentu, akan dicari pertanyaan yang sesuai di database untuk kemudian ditampilkan ke layar. Proses pencarian pertanyaan di database yang sesuai dengan pertanyaan user dapat memanfaatkan algoritma *string matching*. Pada program ini, proses pencarian pertanyaan di database diimplementasikan dengan algoritma KMP(Knuth-Morris-Pratt) atau algoritma BM (Booyer-Moore). Kedua algoritma ini dapat melakukan pencocokan karakter pada string secara *exact matching*.

Apabila tidak ditemukan string pada database yang *exact match* dengan input pertanyaan oleh user, akan dilakukan perhitungan kesamaan setiap string pertanyaan di database dengan string pertanyaan input user. Perhitungan rasio kesamaan kedua string ini diimplementasikan dengan algoritma Longest Common Subsequence (LCS). Proses traversal perhitungan nilai kesamaan antara dua string akan berhenti jika ditemukan string pada database yang memiliki rasio kesamaan minimum 90% atau 0.9. Namun, jika hingga akhir traversal masih belum ditemukan pertanyaan di database yang memiliki kesamaan tidak kurang dari 90%, maka akan ditampilkan 3 pertanyaan dengan nilai kesamaan tertinggi, atau dengan kata lain akan diberikan pilihan 3 pertanyaan paling mirip dengan pertanyaan yang dimasukkan user.

3.2. Fitur Fungsional dan Arsitektur

3.2.1. Fitur Fungsional

Beberapa fitur fungsional yang diimplementasikan dalam program GPTkw antara lain:

- a. Fitur untuk menjawab pertanyaan teks
- b. Fitur untuk melakukan kalkulasi ekspresi
- c. Fitur untuk menentukan hari dari tanggal tertentu
- d. Fitur untuk menambahkan pertanyaan ke database
- e. Fitur untuk menghapus pertanyaan pada database
- f. Fitur history untuk menyimpan jejak penggunaan program
- g. Fitur untuk memilih penggunaan algoritma KMP atau BM

3.2.2. Arsitektur Program

Web-Application GPTkw adalah program yang dapat berjalan pada web browser. Program ini dikembangkan dengan bahasa pemrograman Node.js pada sisi back-end serta Next.js pada sisi front-end. Terdapat beberapa paket-paket atau modul-modul tambahan yang perlu diinstall sebelum menjalankan program, seperti mysql yang diinstall melalui `npm install mysql` dan priorityqueue yang dapat diinstall melalui `npm install fastpriorityqueue`.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1. Spesifikasi Teknis Program

Berikut adalah *prerequisite* yang harus dipenuhi sebelum menjalankan program.

- Java (minimal Java 11), dapat diunduh melalui <https://www.oracle.com/java/technologies/downloads/#java>
- NodeJS, dapat diunduh melalui <https://nodejs.org/en/download/>
- Paket/Modul tambahan :
 - Mysql, dapat diinstall dengan mengetikkan `npm install mysql` pada terminal
 - PriorityQueue, dapat diinstall dengan mengetikkan `npm install fastpriorityqueue` pada terminal
 - Next, dapat diinstall dengan mengetikkan `npm install next` pada terminal

4.2. Tata Cara Penggunaan Program

4.2.1. Tata cara menjalankan backend

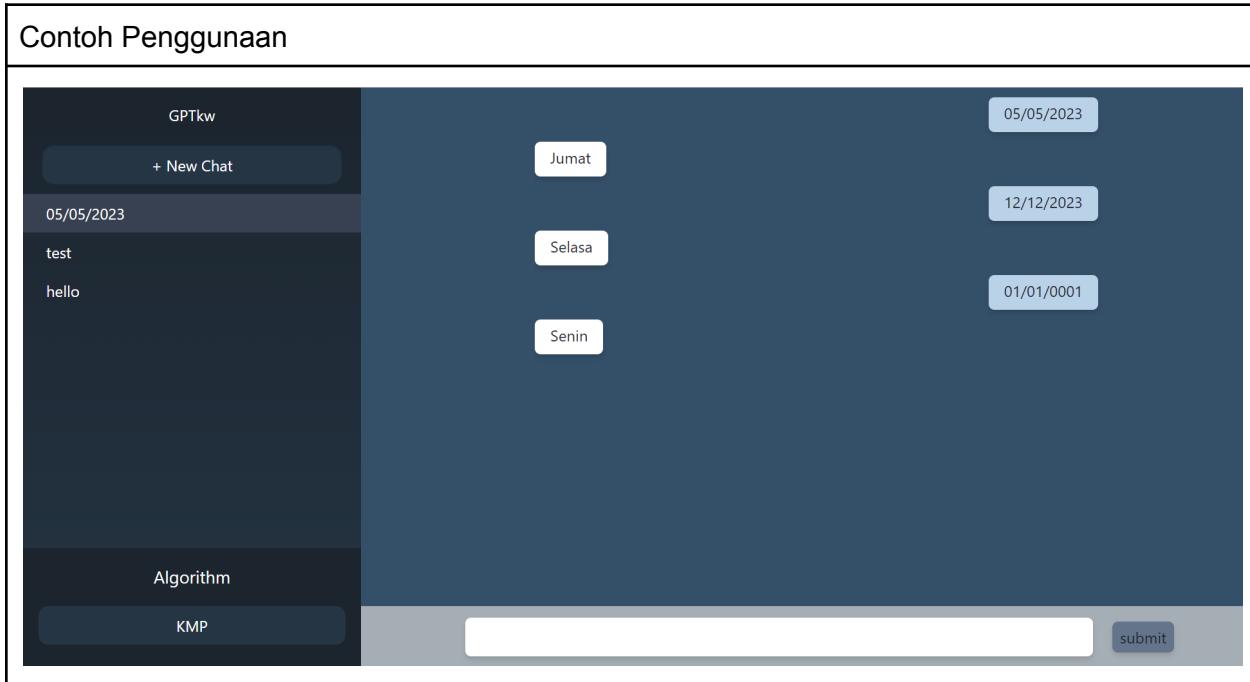
- a. Buka terminal baru pada repository
- b. Pindahkan directory terminal ke folder backend atau ketikkan `cd backend`
- c. Pastikan port :8000 sedang tidak terpakai. Lakukan pengecekan dengan mengetikkan `netstat -ano | findstr :8000`. Anda dapat melakukan terminate program pada port tersebut dengan mengetikkan `taskkill /pid <Process_ID> /F`
- d. Ketikkan perintah `npm run dev` pada terminal untuk menjalankan program backend

4.2.2. Tata cara menjalankan frontend

- a. Buka terminal baru pada repository
- b. Pindahkan directory terminal ke folder backend atau ketikkan `cd frontend`
- c. Ketikkan perintah `npm run dev` pada terminal untuk menjalankan program frontend
- d. Buka program dengan meng-klik URL yang muncul pada terminal frontend

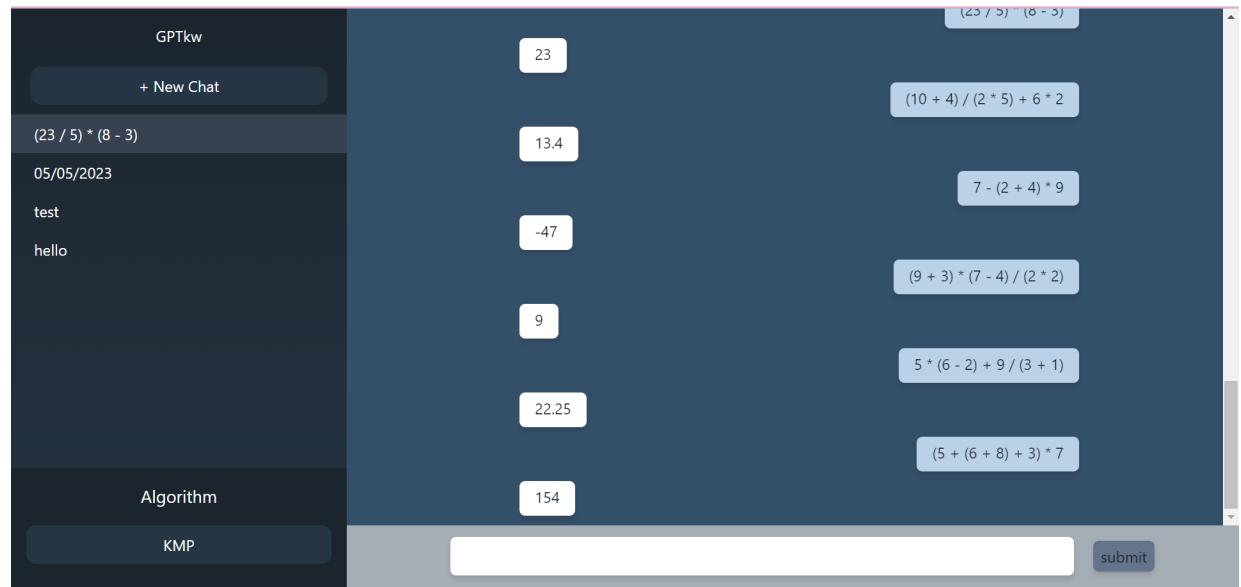
4.3. Hasil Pengujian

4.3.1. Fitur Date

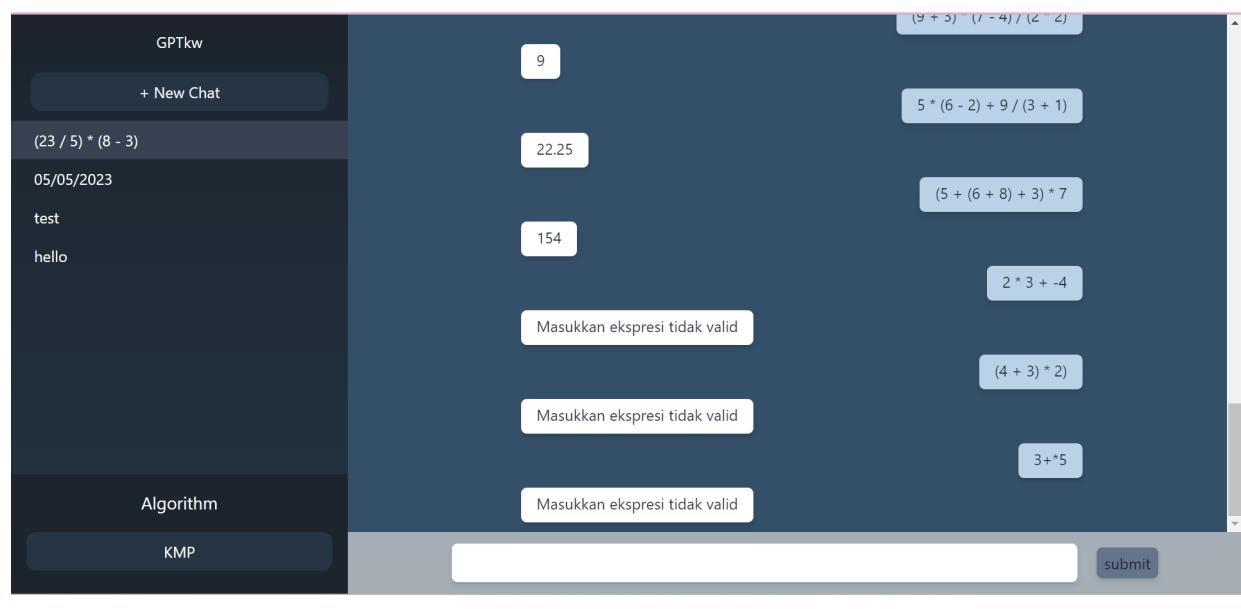


4.3.2. Fitur Kalkulator

Contoh Penggunaan

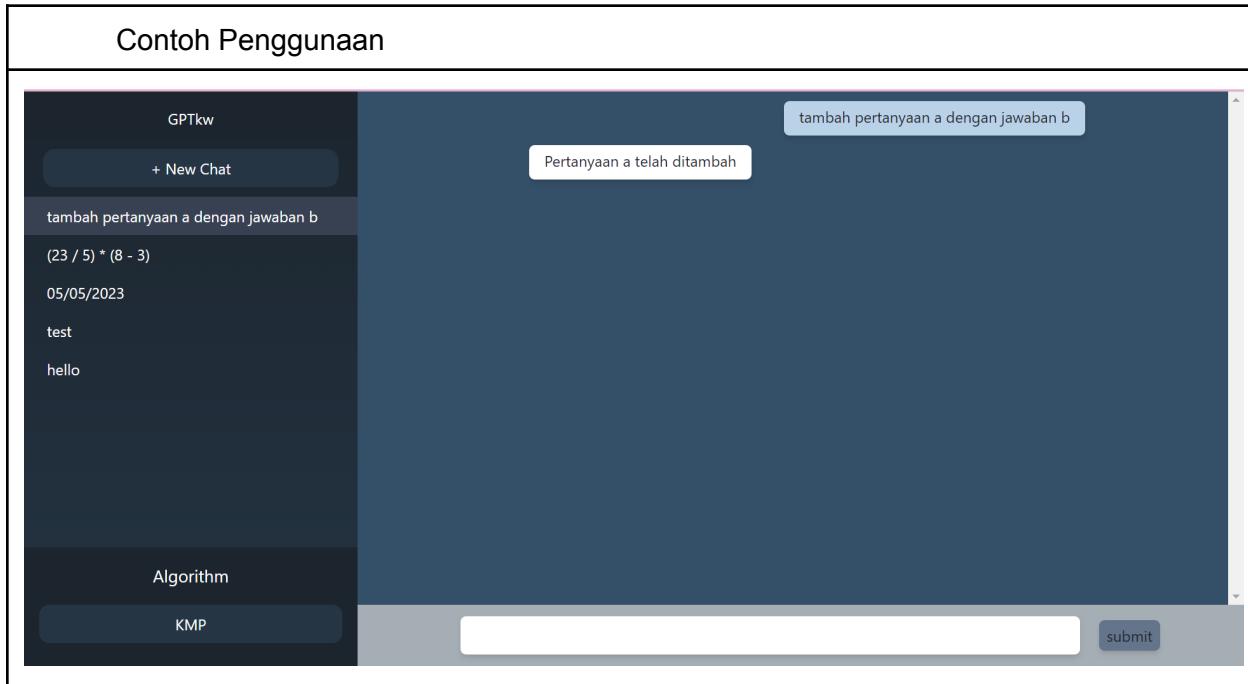


Contoh Validasi

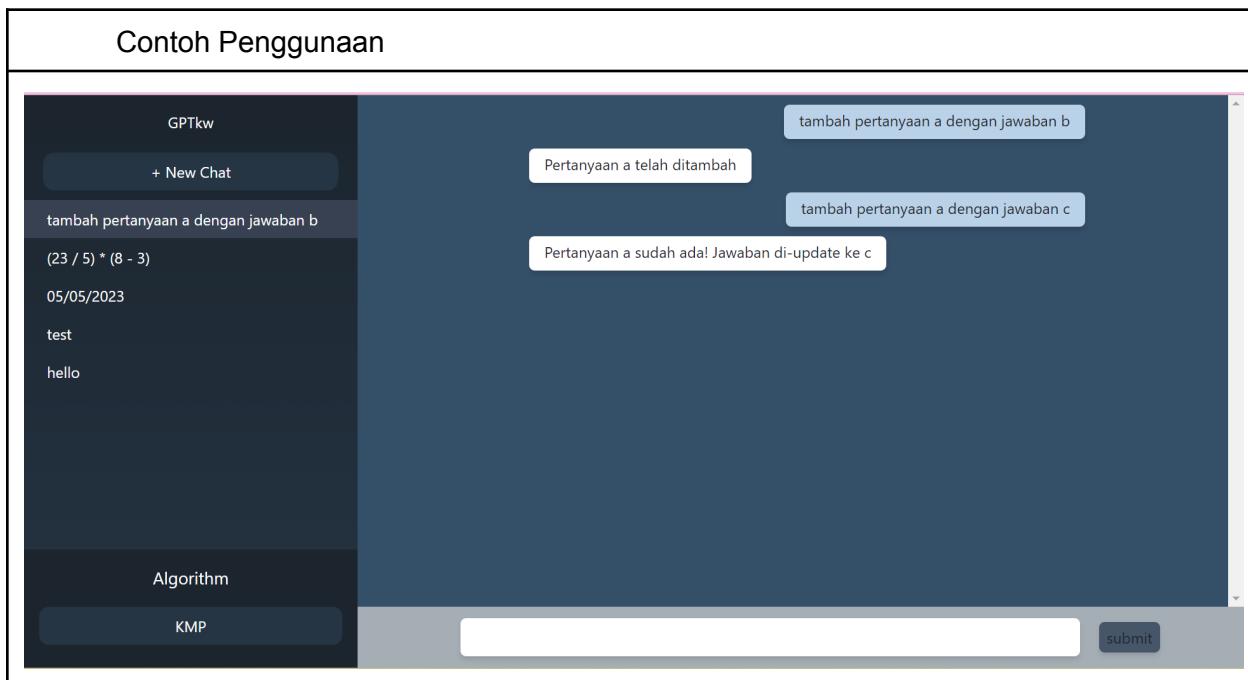


4.3.3. Fitur Tambah/Hapus Pertanyaan

4.3.3.1. Tambah Pertanyaan



4.3.3.2. Ubah Jawaban Pertanyaan



4.3.3.3. Hapus Pertanyaan

Contoh Penggunaan

The screenshot shows a chat application with a sidebar on the left containing user information and a main chat area on the right.

Left Sidebar:

- GPTkw
- + New Chat
- tambah pertanyaan a dengan jawaban b
- (23 / 5) * (8 - 3)
- 05/05/2023
- test
- hello
- Algorithm
- KMP

Main Chat Area:

- tambah pertanyaan a dengan jawaban b
- Pertanyaan a telah ditambah
- tambah pertanyaan a dengan jawaban c
- Pertanyaan a sudah ada! Jawaban di-update ke c
- hapus pertanyaan a
- Pertanyaan a telah dihapus

Bottom Bar:

- submit

Contoh penggunaan jika pertanyaan tidak ada di database

The screenshot shows a similar chat application setup.

Left Sidebar:

- GPTkw
- + New Chat
- tambah pertanyaan a dengan jawaban b
- (23 / 5) * (8 - 3)
- 05/05/2023
- test
- hello
- Algorithm
- KMP

Main Chat Area:

- tambah pertanyaan a dengan jawaban b
- Pertanyaan a telah ditambah
- tambah pertanyaan a dengan jawaban c
- Pertanyaan a sudah ada! Jawaban di-update ke c
- hapus pertanyaan a
- hapus pertanyaan a
- Tidak ada pertanyaan a pada database!

Bottom Bar:

- submit

4.3.4. Fitur ChatBot

4.3.4.1. Dengan Algoritma KMP

Contoh Penggunaan

The screenshot shows a dark-themed chat interface. On the left, the sidebar displays the bot's name "GPTkw", a "+ New Chat" button, and a list of recent conversations: "ibukota Indonesia", "tambah pertanyaan a dengan jawaban b", "(23 / 5) * (8 - 3)", "05/05/2023", "test", and "hello". Below these are sections for "Algorithm" and "KMP". The main area shows a conversation:
User: ibukota Indonesia
Bot: ibukota Indonesia adalah Jakarta
User: ibukota perancis
Bot: Ibukota Perancis adalah Paris

Contoh Menampilkan 3 pertanyaan termirip

The screenshot shows a similar dark-themed chat interface. The sidebar includes "GPTkw", "+ New Chat", and the same list of recent conversations. The "Algorithm" and "KMP" sections are also present. The main area shows a conversation:
User: ibukota Indonesia
Bot: ibukota Indonesia adalah Jakarta
User: ibukota perancis
Bot: Ibukota Perancis adalah Paris
User: halo
Bot: Hmm, saya kurang mengerti. Mungkin yang kamu maksud: 1. hello 2. Algoritma BM 3. Siapa penemu telepon? (ketikkan kembali pertanyaan yang Anda maksud)
User: hello
Bot: Halo! Selamat datang di GPTkw!

4.3.4.2. Dengan Algoritma BM

Contoh Penggunaan

Ibukota Indonesia
ibukota Indonesia adalah Jakarta
ibukota Perancis
ibukota Perancis adalah Paris
halo
hello

Contoh Menampilkan 3 pertanyaan termirip

Ibukota Indonesia
ibukota Indonesia adalah Jakarta
ibukota Perancis
ibukota Perancis adalah Paris
Halo! Saya kurang mengerti. Mungkin yang kamu maksud: 1. hello 2. Algoritma BM 3. Siapa penemu telepon? (ketikkan kembali pertanyaan yang Anda maksud)
halo
Hello! Selamat datang di GPTkw!

4.4. Analisis Hasil Pengujian

Analisis desain solusi yang ditampilkan di bawah ini merupakan hasil analisis dari setiap pengujian yang telah ditampilkan pada sub bab sebelumnya.

Bagian Pengujian	Analisis
Date	Saat pengguna memberi masukan dengan format dd/mm/yyyy maka akan mengembalikan hari dari tanggal tersebut.
Kalkulator	Saat pengguna memberi masukan berupa kombinasi angka, operator (+-*/), dan/atau tanda kurung, maka akan memasuki fitur kalkulator.
	Pada contoh penggunaan, bot terlihat mengembalikan nilai hasil perhitungan.
	Pada contoh validasi, bot terlihat mengembalikan pesan "Masukan ekspresi tidak valid" saat persamaan tidak sesuai sintaks.
Tambah/hapus pertanyaan	Saat diberi perintah 'tambah pertanyaan a dengan jawaban b', dan pertanyaan a belum ada di database, maka pertanyaan a dan jawaban b ditambahkan ke database.
	Saat diberi perintah 'tambah pertanyaan a dengan jawaban c', dan pertanyaan a sudah ada di database, maka jawaban dari pertanyaan a yang disimpan di database diubah menjadi c.
	Saat diberi perintah 'hapus pertanyaan a', dan pertanyaan a ada di database, maka pertanyaan a dihapus dari database.
	Saat diberi perintah 'hapus pertanyaan a', dan pertanyaan a tidak ada di database, maka diberikan pesan 'Tidak ada pertanyaan a pada database!' dan tidak ada perubahan pada database.
ChatBot - KMP	Apabila masukan pengguna tidak memenuhi fitur-fitur lainnya, maka akan memasuki fitur chat bot. Program melakukan string matching antara masukan pengguna dengan pertanyaan yang terdapat di database dengan metode KMP. Apabila ditemukan pertanyaan dengan kemiripan di atas 90%, maka bot akan mengembalikan jawaban dari pertanyaan yang paling mirip tersebut.
	Apabila tidak ditemukan pertanyaan dengan kemiripan di atas 90%, maka bot akan mengembalikan 3 pertanyaan dari database yang paling mirip dengan masukan pengguna.

ChatBot - BM	<p>Apabila masukan pengguna tidak memenuhi fitur-fitur lainnya, maka akan memasuki fitur chat bot.</p> <p>Program melakukan string matching antara masukan pengguna dengan pertanyaan yang terdapat di database dengan metode BM. Apabila ditemukan pertanyaan dengan kemiripan di atas 90%, maka bot akan mengembalikan jawaban dari pertanyaan yang paling mirip tersebut.</p>
	<p>Apabila tidak ditemukan pertanyaan dengan kemiripan di atas 90%, maka bot akan mengembalikan 3 pertanyaan dari database yang paling mirip dengan masukan pengguna.</p>

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pembuatan program chatbot GPTkw ini, dapat disimpulkan bahwa Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM) dapat digunakan sebagai algoritma pencocokan teks atau string exact matching. Pencarian string pada database yang match dengan string masukkan user dengan algoritma KMP dan BM dapat menghasilkan string yang *match*.

5.2. Saran

Untuk pengembangan aplikasi serupa kedepannya dapat dilakukan eksplorasi lebih lagi. Eksplorasi dapat dilakukan dengan menerapkan algoritma-algoritma pencocokan string selain KMP dan Boyer-Moore. Selain itu, mungkin dapat dilakukan perhitungan kompleksitas waktu dan ruangnya untuk memperoleh algoritma yang paling efisien dan sesuai.

5.3. Komentar/Refleksi

Sebagai refleksi, kami ingin mengevaluasi diri agar kedepannya dapat membuat timeline penggerjaan yang lebih jelas dan rapi. Kami juga harus disiplin dalam mengerjakan tugas masing-masing pada tenggat waktu yang telah disepakati. Hal ini untuk menghindari penggerjaan yang terlalu dekat dengan tanggal pengumpulan sehingga pada saat penggabungan pekerjaan masing-masing muncul banyak permasalahan.

DAFTAR PUSTAKA

- Hanafiah, Novita. (2018, November 26). Regular Expression. <https://soc.s.binus.ac.id/>. Diakses pada 25 April 2023, dari <https://soc.s.binus.ac.id/2018/11/26/regular-expression/>
- Khodra, Masayu Leylia. (2019). String Matching dengan Regular Expression. <https://informatika.stei.itb.ac.id/>. Diakses pada 15 April 2023 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>
- Program Studi Teknik Informatika STEI-ITB. (2021). Bahan Kuliah IF2211 Strategi Algoritma Pencocokan String (String/Pattern Matching). <https://informatika.stei.itb.ac.id/>. Diakses pada 15 April 2023 dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>

LAMPIRAN

Link repository GitHub :

https://github.com/munzayanahusn/Tubes3_13521077

Link video YouTube:

<https://youtu.be/vFgA3VEupwo>

TABEL CHECKLIST SPEK

Poin	Ya	Tidak
1. Fitur menjawab pertanyaan teks dengan algoritma KMP	✓	
2. Fitur menjawab pertanyaan teks dengan algoritma BM	✓	
3. Fitur melakukan kalkulasi terhadap masukkan ekspresi	✓	
4. Fitur menentukan hari dari tanggal tertentu	✓	
5. Fitur menambahkan pertanyaan pada database	✓	
6. Fitur menghapus pertanyaan pada database	✓	
7. Fitur menyimpan jejak percakapan atau history	✓	
8. Bonus deploy		✓
9. Bonus video	✓	

PEMBAGIAN TUGAS

NIM	Nama	Tugas
13521077	Husnia Munzayana	Kalkulator, Penentuan hari, String matching (meliputi algoritma KMP, BM, LCS), Laporan, Video
13521111	Tabitha Permalla	Database, Regular expression, Laporan, Video
13521130	Althaaf Khasyi Atisomya	Membangun front-end, History, Laporan, Video