

TUGAS BESAR 3 IF2211 STRATEGI ALGORITMA
SEMESTER II TAHUN 2022/2023
PENERAPAN STRING MATCHING DAN REGULAR EXPRESSION DALAM
PEMBUATAN CHATGPT SEDERHANA



Disusun Oleh:

ChatGDBLS

13521042 Kevin John Wesley Hutabarat

13521059 Arleen Chrysanthia Gunardi

13521107 Jericho Russel Sebastian

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023

DAFTAR ISI

DAFTAR ISI	1
DAFTAR TABEL	3
DAFTAR GAMBAR	4
BAB I	
DESKRIPSI MASALAH	5
1.1 Latar Belakang	5
1.2 Deskripsi Tugas	5
1.3 Fitur-Fitur Aplikasi	6
BAB II	
LANDASAN TEORI	8
2.1 Algoritma Pattern Matching	8
2.1.1 Algoritma Knuth-Morris-Pratt	8
2.1.2 Algoritma Boyer-Moore	9
2.2 Regular Expression	10
2.3 Penjelasan Mengenai Web yang Dibangun	11
BAB III	
ANALISIS PEMECAHAN MASALAH	13
3.1 Langkah Penyelesaian Masalah Setiap Fitur	13
3.1.1 Fitur Pertanyaan Teks	13
3.1.2 Fitur Kalkulator	13
3.1.3 Fitur Tanggal	14
3.1.4 Fitur Tambah Pertanyaan	14
3.1.5 Fitur Hapus Pertanyaan	14
3.2 Fitur Fungsional dan Arsitektur Aplikasi Web yang Dibangun	14
3.2.1 Fungsionalitas Front-End	14
3.2.2 Fungsionalitas Back-End	15
BAB IV	
IMPLEMENTASI DAN PENGUJIAN	16
4.1 Spesifikasi Teknis Program	16
4.1.1 Struktur Data	16
4.1.2 Fungsi dan Prosedur	17
4.2 Tata Cara Penggunaan Program	18
4.3 Hasil Pengujian	19
4.3.1 Pengujian 1: Menanyakan Pertanyaan pada Basis Data	19
4.3.2 Pengujian 2: Menanyakan Pertanyaan yang Tidak Ada pada Basis Data	20
4.3.3 Pengujian 3: Menambahkan Pertanyaan	20
4.3.4 Pengujian 4: Mengubah Jawaban dari Pertanyaan	21

4.3.5 Pengujian 5: Menghapus Pertanyaan	21
4.3.6 Pengujian 6: Menghapus Pertanyaan yang Tidak Ada pada Basis Data	22
4.3.7 Pengujian 7: Menghitung Ekspresi Matematika Masukan Pengguna yang Valid	22
4.3.8 Pengujian 8: Menghitung Ekspresi Matematika Masukan Pengguna yang Tidak Valid	23
4.3.9 Pengujian 9: Menampilkan Hari pada Tanggal Masukan Pengguna yang Valid	23
4.3.10 Pengujian 10: Menampilkan Hari pada Tanggal Masukan Pengguna yang Tidak Valid	24
4.3.11 Pengujian 11: Menanyakan Pertanyaan dengan Kemiripan di Atas 90%	24
4.3.12 Pengujian 12: Menanyakan Pertanyaan dengan Kemiripan di Bawah 90%	25
4.3.13 Pengujian 13: Fitur Tambahan UwUify	25
4.4 Analisis Hasil Pengujian	26
4.4.1 Pengujian 1: Menanyakan Pertanyaan pada Basis Data	26
4.4.2 Pengujian 2: Menanyakan Pertanyaan yang Tidak Ada pada Basis Data	26
4.4.3 Pengujian 3: Menambahkan Pertanyaan	26
4.4.4 Pengujian 4: Mengubah Jawaban dari Pertanyaan	26
4.4.5 Pengujian 5: Menghapus Pertanyaan	26
4.4.6 Pengujian 6: Menghapus Pertanyaan yang Tidak Ada pada Basis Data	27
4.4.7 Pengujian 7: Menghitung ekspresi matematika masukan pengguna yang valid	27
4.4.8 Pengujian 8: Menghitung ekspresi matematika masukan pengguna yang tidak valid	27
4.4.9 Pengujian 9: Menampilkan hari pada tanggal masukan pengguna yang valid	27
4.4.10 Pengujian 10: Menampilkan hari pada tanggal masukan pengguna yang tidak valid	27
4.4.11 Pengujian 11: Menanyakan Pertanyaan dengan Kemiripan di Atas 90%	27
4.4.12 Pengujian 12: Menanyakan Pertanyaan dengan Kemiripan di Bawah 90%	27
4.4.13 Pengujian 13: Fitur Tambahan UwUify	28
BAB V	
PENUTUP	29
5.1 Simpulan	29
5.2 Saran	29
5.3 Komentar/Refleksi	29
DAFTAR REFERENSI	30
LAMPIRAN	31

DAFTAR TABEL

Tabel 2.2.1 Daftar Notasi Umum Regular Expression	11
Tabel 4.1.1.1 Atribut Tabel Questions	16
Tabel 4.1.1.1 Atribut Tabel Histories	16
Tabel 4.1.1.1 Atribut Tabel Chats	16
Tabel 4.1.2.1 Fungsi dan Prosedur Aplikasi	17

DAFTAR GAMBAR

Gambar 2.1.1.1	Ilustrasi pergeseran algoritma KMP ketika ditemukan mismatch	8
Gambar 2.1.2.1	Proses pattern matching dengan algoritma boyer moore	10
Gambar 4.1.1.1	Struktur Data Aplikasi	16
Gambar 4.3.1.1	Hasil Pengujian Pertanyaan yang Ada pada Basis Data	19
Gambar 4.3.2.1	Hasil Pengujian Pertanyaan yang Tidak Ada pada Basis Data	20
Gambar 4.3.3.1	Hasil Pengujian Fitur Menambahkan Pertanyaan	20
Gambar 4.3.4.1	Hasil Pengujian Fitur Mengubah Jawaban dari Pertanyaan	21
Gambar 4.3.5.1	Hasil Pengujian Fitur Menghapus Pertanyaan	21
Gambar 4.3.6.1	Hasil Pengujian Fitur Menghapus Pertanyaan yang Tidak Ada dalam Basis Data	22
Gambar 4.3.7.1	Hasil Pengujian Fitur Menghitung Ekspresi Matematika yang Valid	22
Gambar 4.3.8.1	Hasil Pengujian Fitur Menghitung Ekspresi Matematika yang Tidak Valid	23
Gambar 4.3.9.1	Hasil Pengujian Fitur Menampilkan Hari pada Tanggal yang Valid	23
Gambar 4.3.10.1	Hasil Pengujian Fitur Menampilkan Hari pada Tanggal yang Tidak Valid	24
Gambar 4.3.11.1	Hasil Pengujian Fitur Pertanyaan ketika Kemiripan di Atas 90%	24
Gambar 4.3.12.1	Hasil Pengujian Fitur Pertanyaan ketika Kemiripan di Bawah 90%	25
Gambar 4.3.13.1	Hasil Pengujian Fitur Tambahan UwUify	25

BAB I

DESKRIPSI MASALAH

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.

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.

BAB II

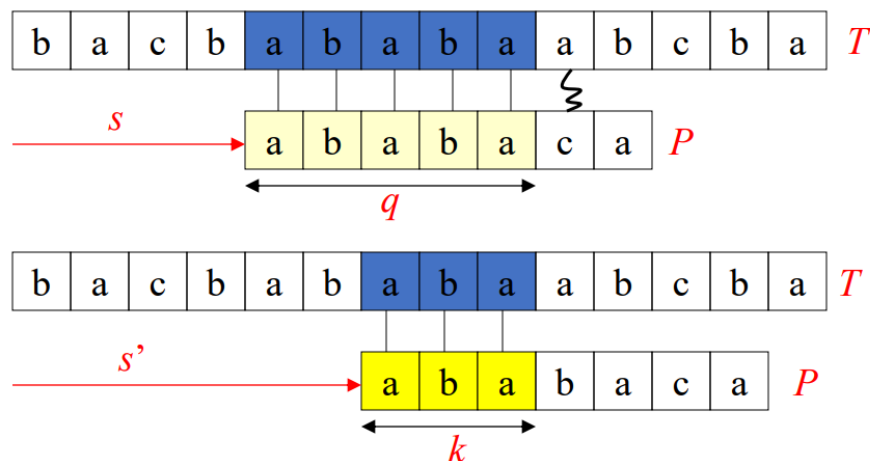
LANDASAN TEORI

2.1 Algoritma Pattern Matching

Algoritma Pattern Matching adalah algoritma yang digunakan untuk menemukan lokasi suatu kata di dalam suatu teks. Algoritma ini memiliki dua elemen, yaitu teks berupa string dengan panjang n karakter, dan pattern berupa string dengan panjang m karakter, dengan m lebih kecil dari n . Algoritma ini dapat diimplementasikan dengan berbagai metode, antara lain:

2.1.1 Algoritma Knuth-Morris-Pratt

Algoritma ini melakukan pencocokan pattern dalam teks dari kiri ke kanan secara karakter per karakter. Algoritma ini menghindari perbandingan yang tidak diperlukan dengan mencari banyak huruf yang tidak perlu diperiksa lagi karena sudah pasti cocok, sehingga algoritma ini lebih baik daripada brute force yang pergeseran pattern terhadap teks hanya satu karakter. Pergeseran pattern terhadap teks pada algoritma KMP ketika menemui *mismatch* adalah banyak karakter pattern dikurangi dengan banyak karakter pada prefix terbesar yang juga merupakan suffix dari pattern yang sudah sesuai. Berikut ini diberikan sebuah contoh pergeseran ketika ditemukan mismatch:



Gambar 2.1.1.1 Ilustrasi pergeseran algoritma KMP ketika ditemukan mismatch

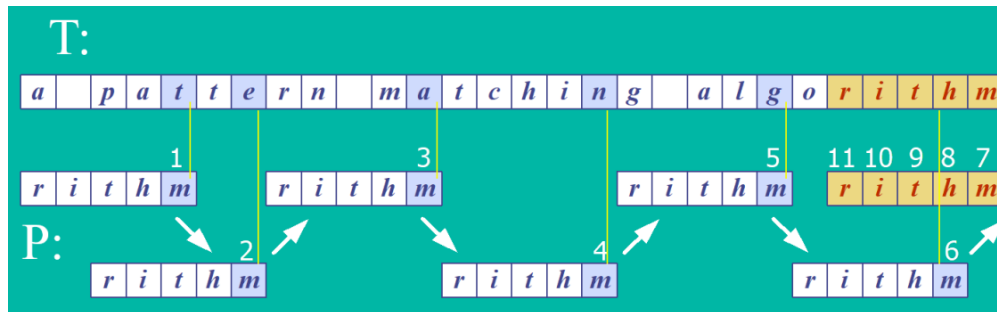
Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>

Pada ilustrasi di atas, pattern mengalami mismatch ketika sampai pada karakter c, sehingga pattern yang sudah sesuai adalah 'ababa'. Pada 'ababa', prefix terpanjang yang juga merupakan suffixnya adalah 'aba'. Pergeseran pemeriksaan dilakukan sebanyak panjang karakter pattern yang sudah diperiksa dikurangi dengan panjang 'aba', yaitu $5-3 = 2$. Untuk melanjutkan pemeriksaan, pattern akan digeser terhadap teks sebanyak dua karakter ke kanan, dan pemeriksaan akan dimulai dari posisi karakter pada pattern yang mengalami mismatch. Proses ini akan berulang sampai ditemukan subset dari teks yang sama persis dengan pattern yang diberikan.

Kompleksitas waktu untuk algoritma ini adalah $O(m+n)$, dengan kompleksitas waktu untuk menghitung fungsi pinggiran adalah $O(m)$, dan kompleksitas waktu untuk pencarian string adalah $O(n)$. Algoritma ini akan sangat menguntungkan jika dipakai untuk mencari pattern dalam teks dengan variasi karakter yang sedikit, sehingga jika ditemui mismatch dapat bergeser dengan signifikan. Sebaliknya, jika variasi karakter terlalu banyak, algoritma ini tidak begitu menguntungkan.

2.1.2 Algoritma Boyer-Moore

Algoritma Boyer-Moore adalah algoritma pencocokan string yang dilakukan dengan cara memeriksa karakter per karakter antara pattern dan teks secara terbalik (*backward*). Hal pertama yang dilakukan algoritma ini adalah membuat karakter pertama pada teks dan pattern menjadi sejajar, kemudian memeriksa teks dari indeks terakhir pattern. Jika terjadi mismatch, pergeseran pattern akan meninjau huruf mismatch tersebut. Jika karakter tersebut dimiliki oleh pattern, maka pergeseran akan dilakukan sedemikian rupa sehingga karakter tersebut sejajar dengan kemunculan karakter terakhir pada pattern. Kemudian proses pencarian dilanjutkan dari indeks terakhir pattern kembali. Berikut ini diberikan sebuah contoh pergeseran ketika ditemukan mismatch:



Gambar 2.1.2.1 Proses pattern matching dengan algoritma boyer moore

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>

Pada ilustrasi di atas, pattern akan disejajarkan terlebih dahulu dengan teks. Pada pemeriksaan pertama, terjadi mismatch antara karakter 't' dan 'm'. Algoritma akan mencari kemunculan 't' dalam pattern, kemudian huruf 't' pada pattern dan teks disejajarkan untuk melanjutkan proses pencarian. Selanjutnya pada pemeriksaan kedua, terjadi mismatch antara karakter 'e' dan 'm'. Karena 'e' tidak muncul dalam pattern, maka pattern akan digeser ke indeks berikutnya dari karakter yang mismatch. Proses pencarian akan berlanjut seperti itu sampai ditemukan match antara pattern dengan teks.

Kompleksitas waktu dari algoritma Boyer-Moore adalah $O(mn+A)$, dengan n adalah panjang karakter dari teks, m adalah panjang karakter dari pattern, dan A adalah banyak variasi karakter pada teks. Algoritma ini akan berjalan dengan cepat apabila variasi karakter dari teks yang diperiksa cukup banyak seperti bahasa inggris, sementara algoritma ini akan berjalan dengan lambat apabila variasi karakter dari teks yang diperiksa sedikit, seperti bahasa binary.

2.2 Regular Expression

Regular Expression adalah serangkaian karakter yang mendefinisikan sebuah pola pencarian. Biasanya, pola tersebut digunakan oleh algoritma pencarian string untuk operasi "find" atau "find and replace". Dengan regular expression, dapat dicari suatu kata dalam teks yang memenuhi pola tersebut. Regular Expression ini sangat membantu untuk menemukan kata yang bermaksud sama namun dengan penulisan yang sedikit berbeda, misalnya memiliki perbedaan huruf kapital, tanda baca, perbedaan satu karakter antar kata dan pattern, dll.

Notasi umum untuk regular expression adalah sebagai berikut:

Tabel 2.2.1 Daftar Notasi Umum Regular Expression

.	Karakter apapun kecuali newline
\.	Karakter titik (dapat diterapkan untuk *,\(\,\),dan lain-lain
^	Awalan dari string
\$	Akhir dari string
\d,\w,\s	Digit; karakter kata[A-Za-z0-9]; whitespace
\D,\W,\S	Apapun selain digit; karakter kata[A-Za-z0-9]; whitespace
[abc]	Satu karakter di dalam kurung siku (a, b, atau c)
[^abc]	Satu karakter lain selain karakter di dalam kurung siku (a, b, atau c)
aa bb	aa atau bb
?	Nol atau satu dari karakter sebelumnya
*	Nol atau lebih dari karakter sebelumnya
+	Satu atau lebih dari karakter sebelumnya
{n}	Karakter sebelumnya sebanyak n
{n,}	Karakter sebelumnya sebanyak n atau lebih
{m,n}	Karakter sebelumnya sebanyak m sampai n
??,*?,+?,{n}>,etc.	Sama seperti di atas, tetapi sesedikit mungkin
{expr}	Capturing expr for use with \1, etc.
(?:expr)	Non-capturing group
(?=expr)	Diikuti oleh expr.
(?!expr)	Tidak diikuti oleh expr.

2.3 Penjelasan Mengenai Web yang Dibangun

Aplikasi berbasis web ini dibuat menggunakan bahasa pemrograman NodeJS untuk mengimplementasikan backend nya. Untuk frontend menggunakan React. Data yang dimuat di dalam web dikelola dengan DBMS MySQL. Aplikasi ini dapat berinteraksi dengan user dengan menjawab pertanyaan-pertanyaan user seperti hari pada tanggal tertentu, menjawab persoalan matematika sederhana, juga pertanyaan-pertanyaan lain yang apabila pertanyaan tersebut

tersimpan dalam database, aplikasi akan menjawab pertanyaan tersebut. Pengguna juga dapat menambahkan ataupun menghapus pertanyaan pada database.

BAB III

ANALISIS PEMECAHAN MASALAH

3.1 Langkah Penyelesaian Masalah Setiap Fitur

3.1.1 Fitur Pertanyaan Teks

1. User memasukkan *query* dan algoritma nya
2. Jika *query* cocok dengan *regular expression* pertanyaan teks, maka *query* akan dikelola dengan fitur pertanyaan teks
3. Program mencari jawaban dari database dengan mencocokkan *query* dengan pertanyaan dalam *database* dengan algoritma KMP atau BM sesuai dengan masukan pengguna
4. Jika tidak ditemukan pertanyaan yang *exact match*, program akan mencari kemiripan *query* dengan pertanyaan-pertanyaan dalam database dengan metode *levenshtein distance*
5. Jika ada pertanyaan yang kemiripannya lebih dari 90%, maka program akan menampilkan jawaban dari pertanyaan tersebut
6. Jika tidak ada, maka program akan menampilkan saran pertanyaan berdasarkan tiga pertanyaan yang paling mirip dengan *query*.

3.1.2 Fitur Kalkulator

1. User memasukkan *query* dan algoritma
2. Jika *query* cocok dengan *regular expression* kalkulator, maka *query* akan dikelola dengan fitur kalkulator
3. Fitur kalkulator dimulai dengan inisialisasi *stack* untuk operator dan *value*.
4. Fungsi membaca karakter satu per satu dari kiri. Jika token berupa bilangan, akan dipush ke *stack value*, sedangkan jika token berupa operator, akan di-*push* ke *stack operator*
5. Setiap kali operator akan di-*push*, tinjau elemen paling atas dari *stack operator*. Apabila urutan prioritas dari operator di *stack* lebih besar atau sama dengan daripada operator yang akan di-*push*, hitung terlebih dahulu ekspresi sebelumnya dengan *pop* dua *value* dari *value stack* dan 1 operator dari *operator stack* sampai urutan prioritas dari operator teratas *stack operator* lebih kecil daripada yang akan di-*push*, atau sampai *stack operator* kosong.
6. Ketika fungsi menerima token “)”, fungsi akan menghitung setiap ekspresi dari *stack value* dan operator sampai ketemu token “(” pada *stack operator*. Jika tidak ada, maka ekspresi tidak valid.
7. Saat semua ekspresi dari *query* sudah di-*push* ke *stack* dan semua ekspresi sudah diolah, tetapi *stack operator* tidak kosong, maka ekspresi tidak valid.

3.1.3 Fitur Tanggal

1. User memasukkan *query* dan algoritma nya
2. Jika *query* cocok dengan regular expression tanggal, maka *query* akan dikelola dengan fitur tanggal
3. Jika *query* tanggal tidak dalam format dd/mm/yyyy, maka *syntax* tidak valid
4. *Query* dikelola dengan Zeller's Congruence untuk mendapatkan hari pada tanggal tersebut.
5. Jika *query* valid, program akan menampilkan hari pada tanggal tersebut

3.1.4 Fitur Tambah Pertanyaan

1. User memasukkan *query* dan algoritma nya
2. Jika *query* cocok dengan *regular expression* tambah pertanyaan (diawali dengan "tambah pertanyaan"), *query* akan dikelola dengan fitur tambah pertanyaan
3. Jika pertanyaan yang dimasukkan pengguna sudah ada dalam *database*, jawaban dari pertanyaan tersebut akan diperbarui dengan jawaban dari masukan pengguna dengan *query* "UPDATE questions SET answer_pattern = <jawaban_pengguna> WHERE question_pattern = <pertanyaan_pengguna>"
4. Jika pertanyaan tidak ada dalam *database*, pertanyaan dan jawaban akan ditambahkan ke dalam *database* dengan *query* "INSERT INTO questions VALUES (<pertanyaan_pengguna>,<jawaban_pengguna>)"

3.1.5 Fitur Hapus Pertanyaan

1. User memasukkan *query* dan algoritma nya
2. Jika *query* cocok dengan regular expression tambah pertanyaan, *query* akan dikelola dengan fitur tambah pertanyaan
3. Jika pertanyaan tidak ditemukan dalam *database*, pertanyaan gagal dihapus dan program akan mengembalikan pesan bahwa pertanyaan tidak dapat dihapus
4. Jika tidak, pertanyaan akan dihapus dari *database* dengan *query* "DELETE FROM questions WHERE question_pattern = <pertanyaan>"

3.2 Fitur Fungsional dan Arsitektur Aplikasi Web yang Dibangun

3.2.1 Fungsionalitas Front-End

Berikut merupakan fungsionalitas *front-end* aplikasi web:

1. Fitur chat
Fitur ini dapat digunakan untuk menanyakan pertanyaan kepada aplikasi. Masukan pertanyaan yang diterima adalah pertanyaan biasa, ekspresi matematika, tanggal, menambahkan pertanyaan, mengubah jawaban pertanyaan, serta menghapus pertanyaan.

2. Fitur history
Fitur ini dapat digunakan untuk melihat semua riwayat pertanyaan dan jawaban yang pernah ditanyakan. Setiap pertanyaan yang ditanyakan akan ditambahkan pada history.
3. Fitur new chat
Fitur ini dapat digunakan untuk menambahkan percakapan baru.

3.2.2 Fungsionalitas Back-End

Berikut merupakan fungsionalitas *front-end* aplikasi web:

1. Fitur Kalkulator
Fitur ini dapat digunakan untuk menghitung ekspresi matematika masukan pengguna
2. Fitur Tanggal
Fitur ini dapat digunakan untuk menampilkan hari yang dimiliki oleh tanggal masukan pengguna
3. Fitur menjawab pertanyaan
Fitur ini dapat digunakan untuk menjawab pertanyaan dari pengguna yang terdapat dalam database
4. Fitur menambahkan pertanyaan
Fitur ini dapat digunakan untuk menambahkan pertanyaan dan jawabannya yang sebelumnya belum ada dalam database
5. Fitur menghapus pertanyaan
Fitur ini dapat digunakan untuk menghapus pertanyaan dari database

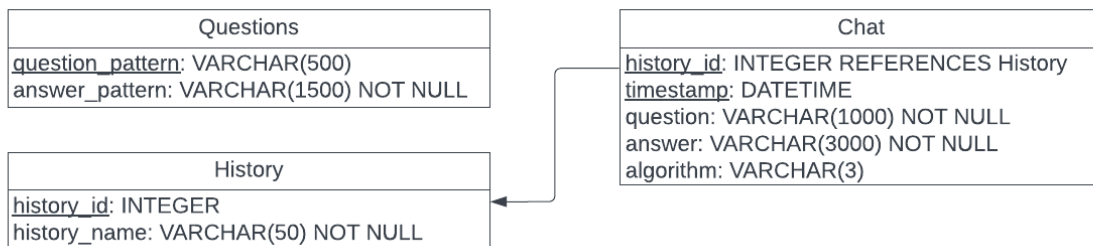
BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Spesifikasi Teknis Program

4.1.1 Struktur Data

Struktur data pada program ini dapat direpresentasikan dengan diagram berikut.



Gambar 4.1.1.1 Struktur Data Aplikasi

1. Tabel questions

Tabel 4.1.1.1 Atribut Tabel Questions

Nama atribut	Keterangan
question_pattern	Menyimpan pertanyaan
answer_pattern	Menyimpan jawaban

2. Tabel histories

Tabel 4.1.1.1 Atribut Tabel Histories

Nama atribut	Keterangan
history_id	Menyimpan id dari history
history_name	Menyimpan nama dari history

3. Tabel chats

Tabel 4.1.1.1 Atribut Tabel Chats

Nama atribut	Keterangan
history_id	Menyimpan id history tempat chat berada

question	Menyimpan query dari pengguna
answer	Menyimpan respon dari program
timestamp	Menyimpan waktu saat query diproses
algorithm	Menyimpan algoritma yang digunakan untuk menampilkan respon

4.1.2 Fungsi dan Prosedur

Tabel 4.1.2.1 Fungsi dan Prosedur Aplikasi

Nama fungsi/prosedur	Keterangan
bmMatch(String text, String pattern) → int	Mencari kemunculan pattern dalam text dengan algoritma Boyer-Moore. Jika ditemukan, akan mengembalikan indeks pertama tempat pattern tersebut ditemukan. Jika tidak ditemukan akan mengembalikan -1
kmp(String text, String pattern) → int	Mencari kemunculan pattern dalam text dengan algoritma Knutt-Morris-Pratt. Jika ditemukan, akan mengembalikan indeks pertama tempat pattern tersebut ditemukan. Jika tidak ditemukan akan mengembalikan -1
levenshtein(String string1,String string2) → float	Mengembalikan sebuah integer berupa banyaknya penambahan, penghapusan, atau penghapusan yang dilakukan agar kedua string sama
getSimilarityPercentage(String string1,String string2) → float	Mengembalikan persentase kemiripan antara string1 dan string2.
findTheDay(String date) → string	Mengembalikan hari pada tanggal masukan
evaluate(String str) → double	Mengembalikan hasil perhitungan dari ekspresi masukan
acceptUserQuery(String query, UserQueryConfig config)	Mengolah masukan pengguna, menentukan fitur yang akan dijalankan selanjutnya
addHistory(String name)	Menambahkan history ke dalam database

addChat(int id,String question, String answer, String algorithm)	Menambahkan chat ke dalam database
addQuestion(String question, String answer)	Menambahkan pertanyaan ke dalam database
removeQuestion(String question)	Menghapus pertanyaan dari database
getQuestions()	Mengembalikan semua pertanyaan dalam database
getAllHistory()	Mengembalikan semua history chat
getHistoryName(id)	Mengembalikan history_name yang memiliki history_id id
getChatinHistory(id)	Mengembalikan semua chat yang memiliki history_id id
getData(query, values = [])	Mendapatkan data sesuai dengan query
loadSQL()	Mengambil data dari file .sql
dumpSQL()	Men- <i>dump</i> data ke file .sql

4.2 Tata Cara Penggunaan Program

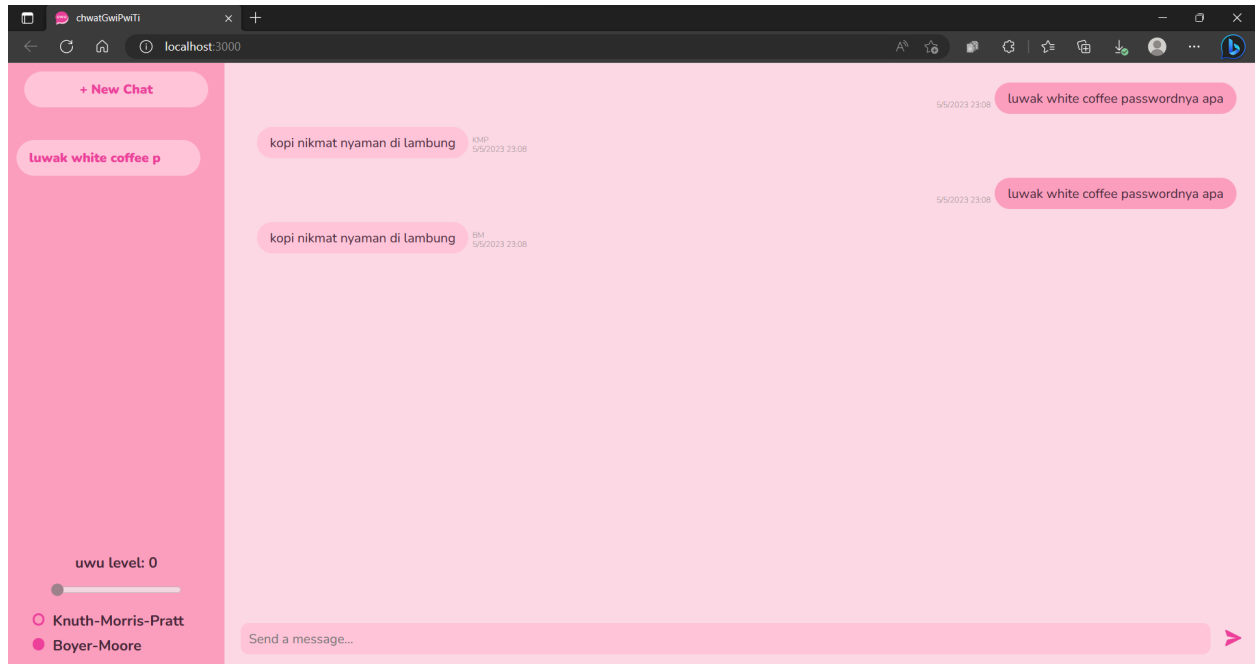
Berikut ini adalah tata cara penggunaan program secara umum:

- Link ke repository dapat diakses di bagian lampiran pada laporan ini
- Clone repository tersebut, kemudian jalankan aplikasi sesuai dengan langkah-langkah yang tertera di readme
- Pengguna dapat memasukkan *query* yang berupa pertanyaan random, ekspresi matematika, ataupun pertanyaan hari pada tanggal tertentu.
- Pengguna diminta untuk memilih algoritma *string matching*.
- Kemudian tekan tombol send
 - Jika query tersebut tergolong ke dalam pertanyaan random, program akan menampilkan jawaban dari pertanyaan tersebut dari database
 - Jika query tersebut tergolong ekspresi matematika, program akan menampilkan hasil evaluasi dari ekspresi matematika tersebut
 - Jika query tersebut tergolong pertanyaan hari pada tanggal tertentu, program akan menampilkan hari pada tanggal tersebut
 - Jika query tersebut tergolong menambahkan pertanyaan ke dalam database, program akan menambahkan pertanyaan dan jawaban ke dalam database
 - Jika query tersebut tergolong menghapus pertanyaan ke dalam database, program akan menghapus pertanyaan dari database

- Jika pengguna ingin menambahkan laman chat baru, dapat menekan tombol “+ new chat” di sebelah kiri atas

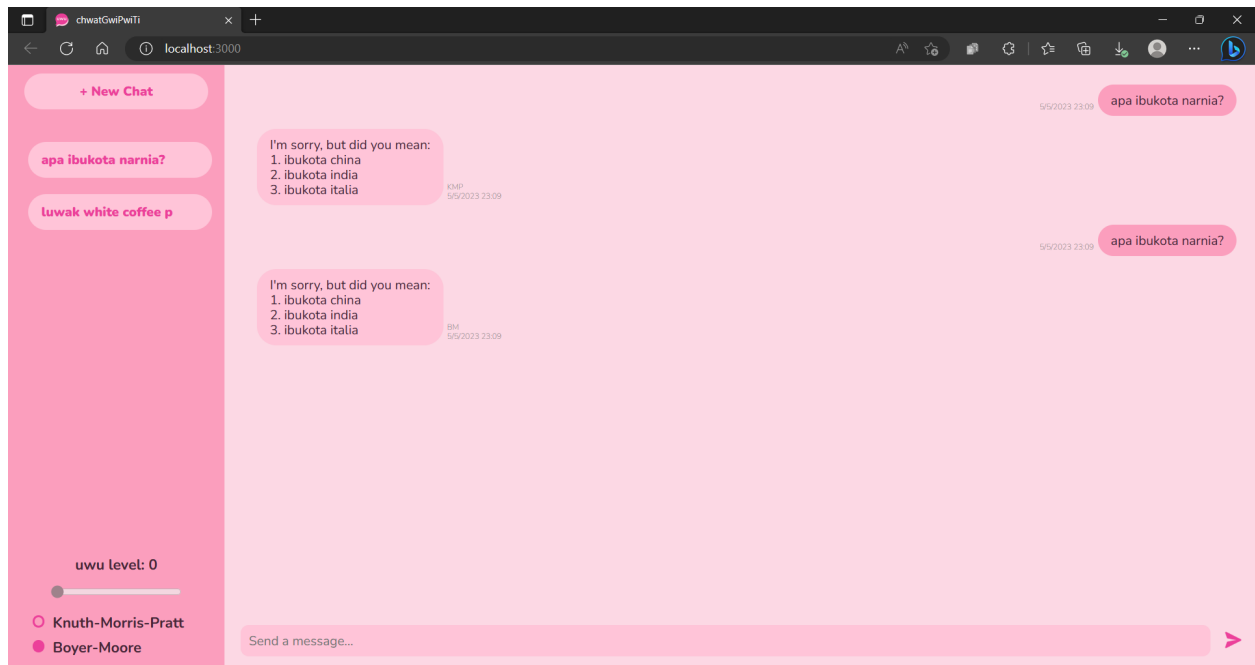
4.3 Hasil Pengujian

4.3.1 Pengujian 1: Menanyakan Pertanyaan pada Basis Data



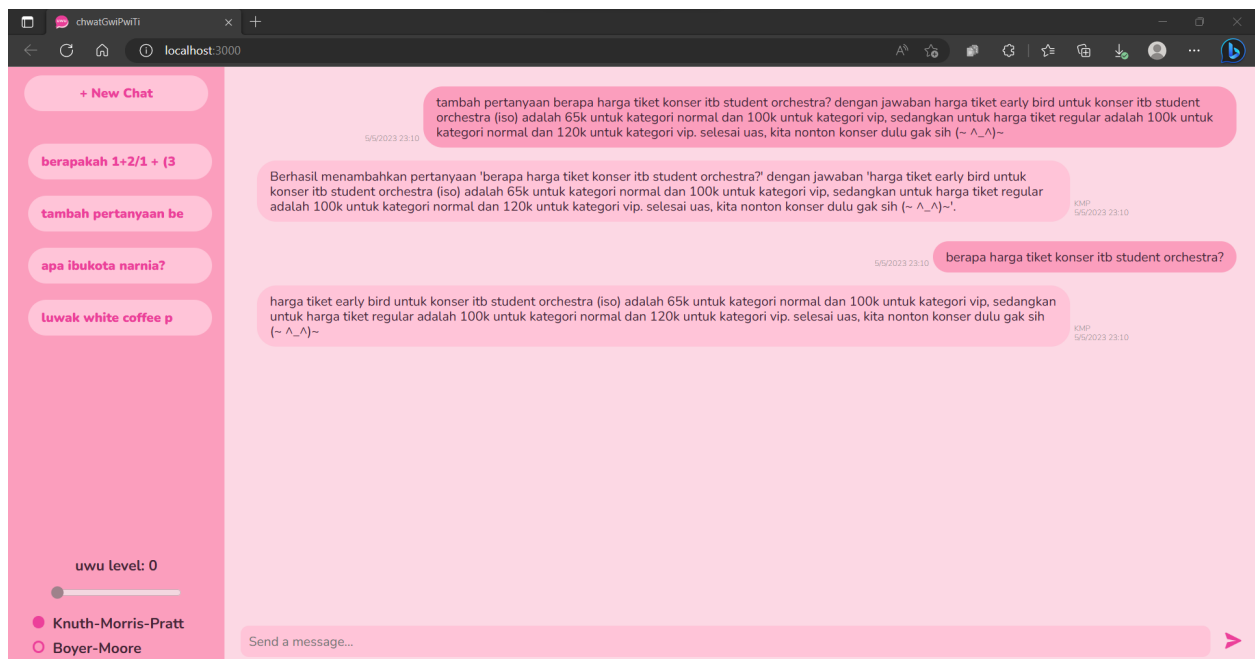
Gambar 4.3.1.1 Hasil Pengujian Pertanyaan yang Ada pada Basis Data

4.3.2 Pengujian 2: Menanyakan Pertanyaan yang Tidak Ada pada Basis Data



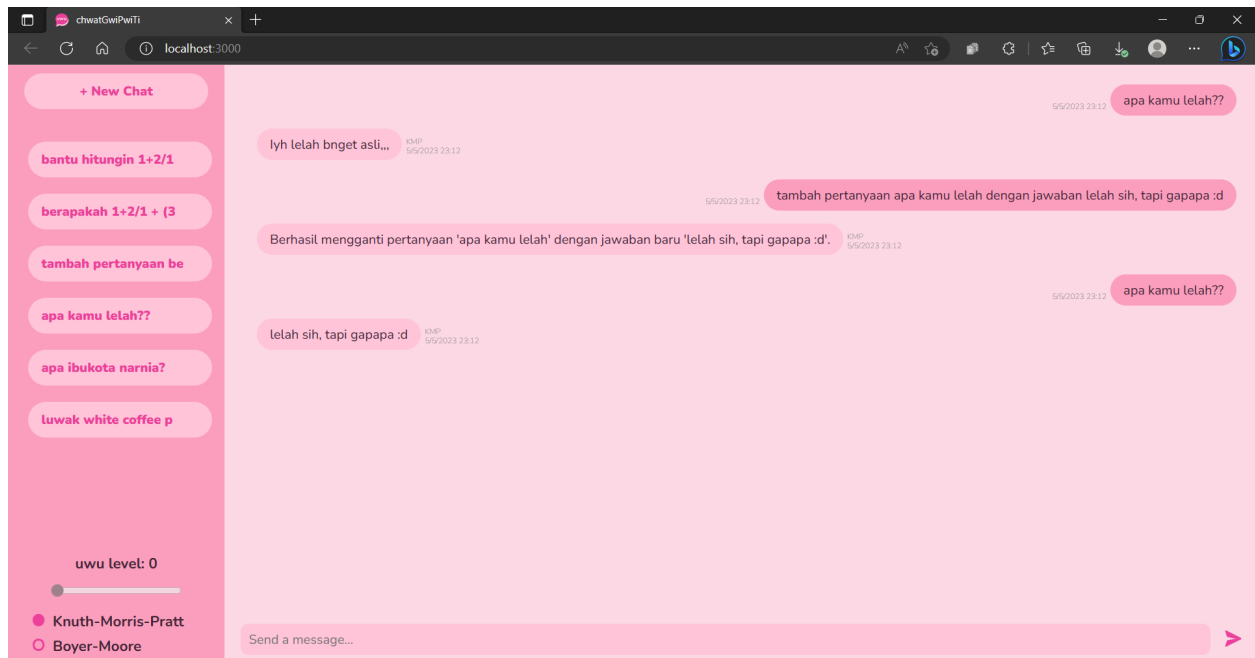
Gambar 4.3.2.1 Hasil Pengujian Pertanyaan yang Tidak Ada pada Basis Data

4.3.3 Pengujian 3: Menambahkan Pertanyaan



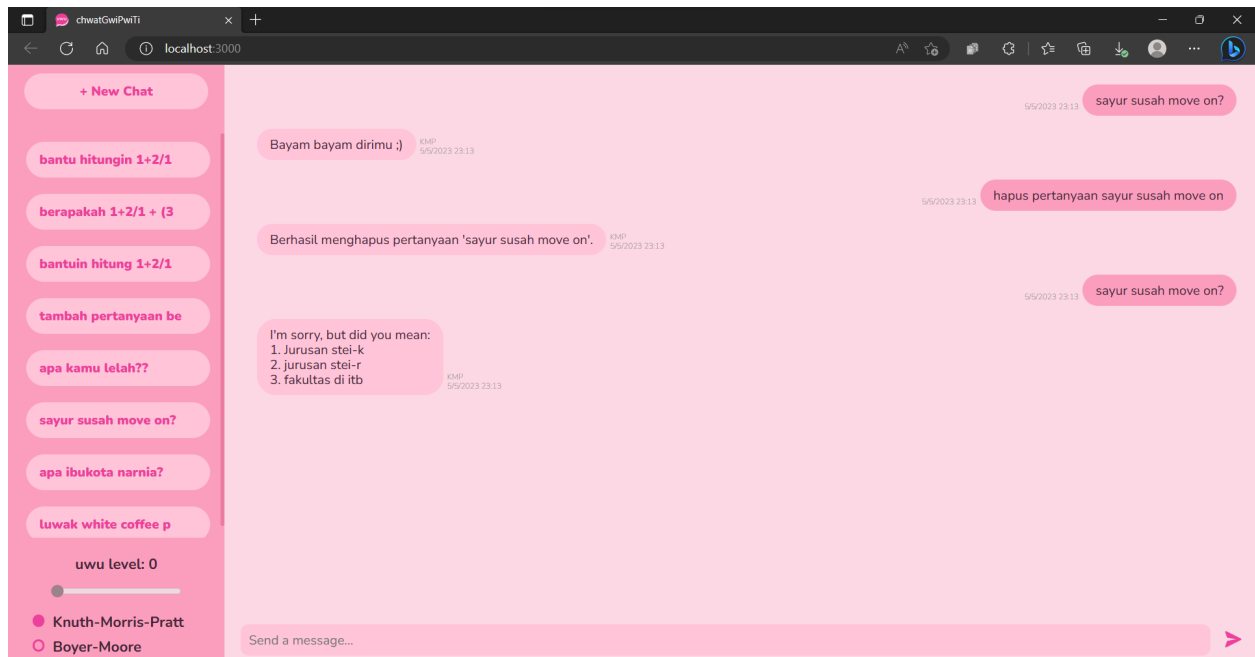
Gambar 4.3.3.1 Hasil Pengujian Fitur Menambahkan Pertanyaan

4.3.4 Pengujian 4: Mengubah Jawaban dari Pertanyaan



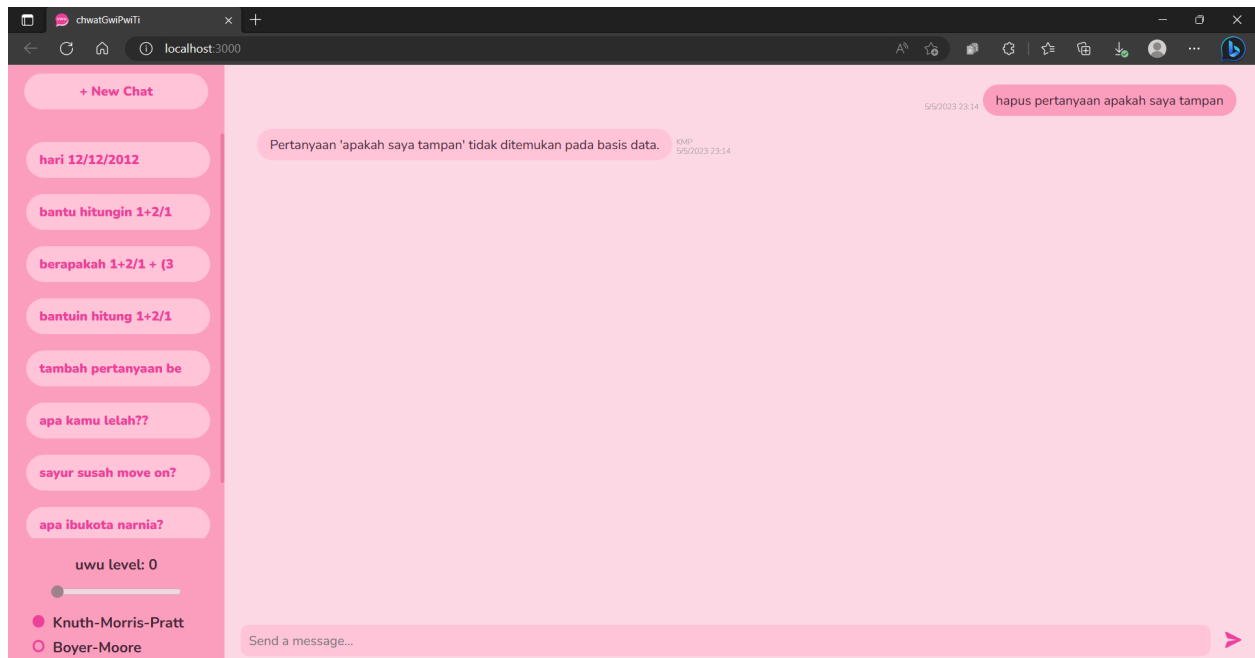
Gambar 4.3.4.1 Hasil Pengujian Fitur Mengubah Jawaban dari Pertanyaan

4.3.5 Pengujian 5: Menghapus Pertanyaan



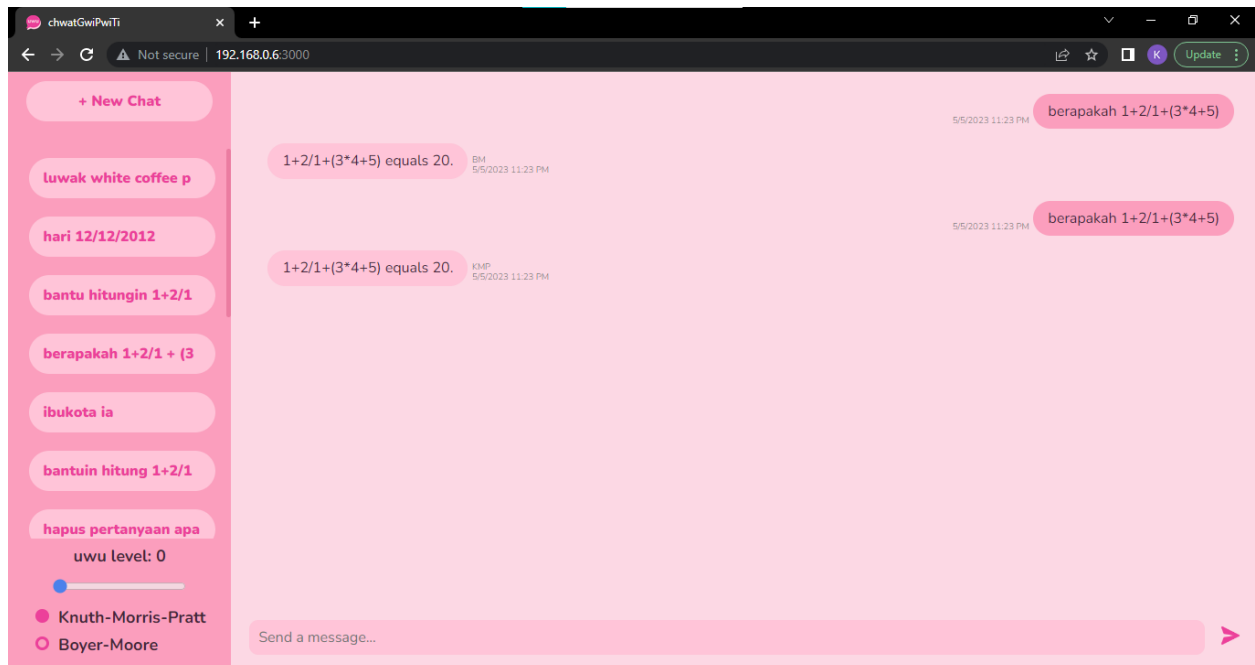
Gambar 4.3.5.1 Hasil Pengujian Fitur Menghapus Pertanyaan

4.3.6 Pengujian 6: Menghapus Pertanyaan yang Tidak Ada pada Basis Data



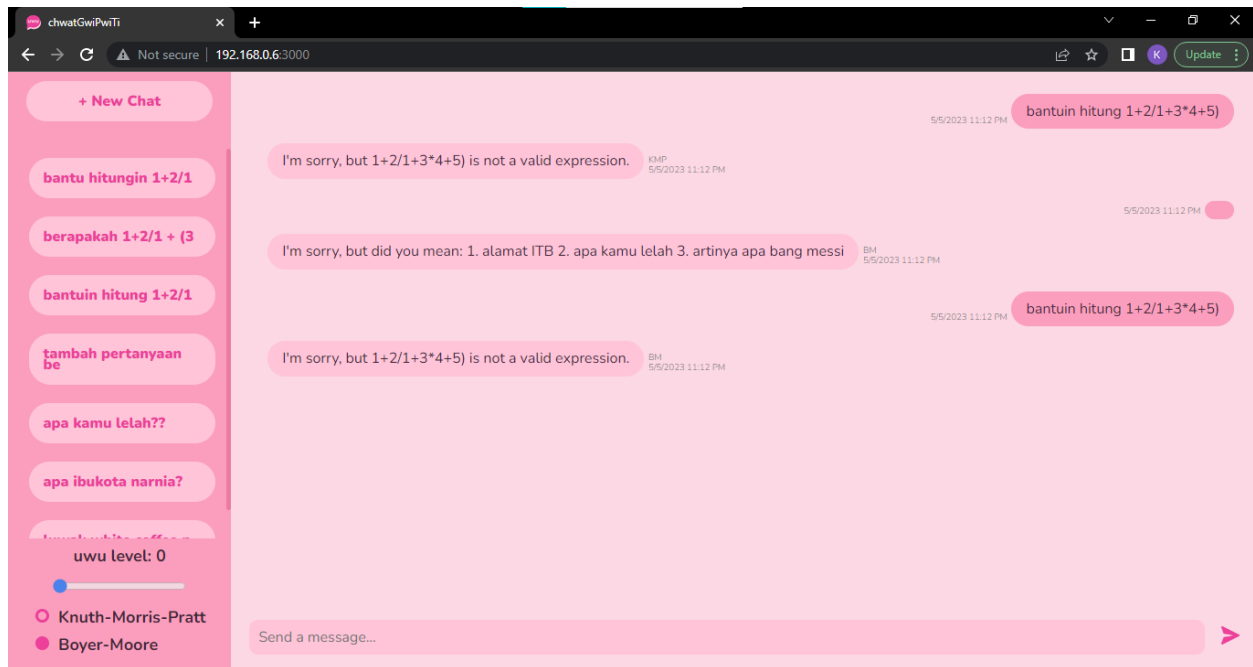
Gambar 4.3.6.1 Hasil Pengujian Fitur Menghapus Pertanyaan yang Tidak Ada dalam Basis Data

4.3.7 Pengujian 7: Menghitung Ekspresi Matematika Masukan Pengguna yang Valid



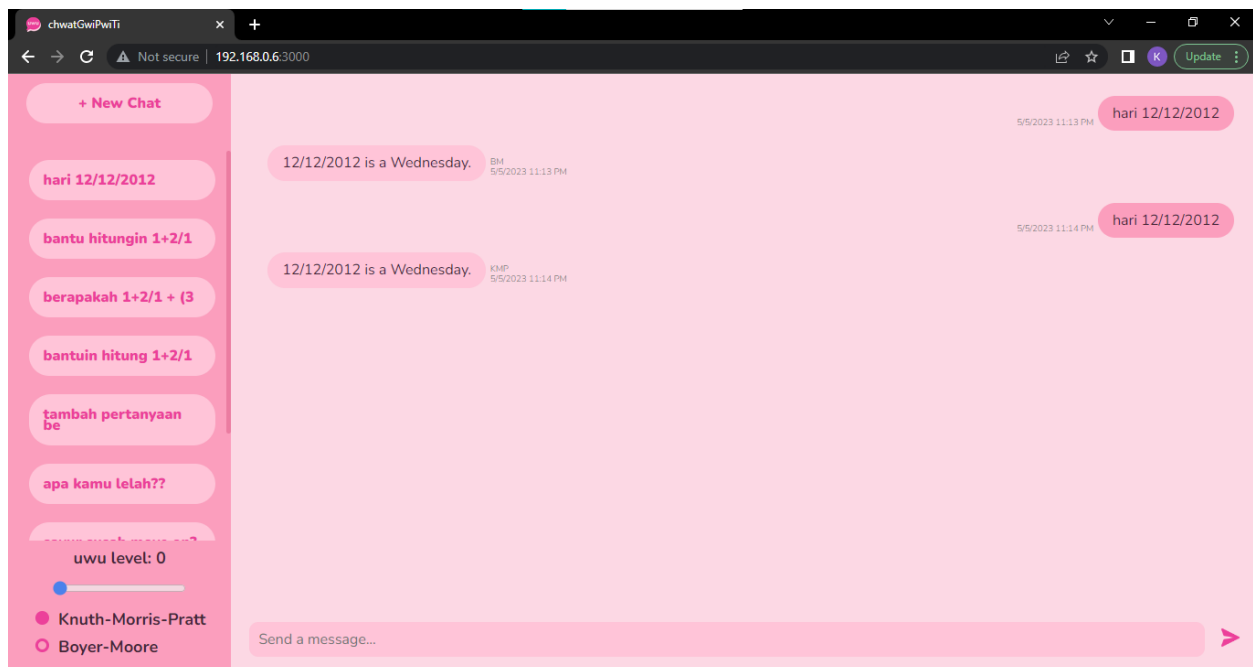
Gambar 4.3.7.1 Hasil Pengujian Fitur Menghitung Ekspresi Matematika yang Valid

4.3.8 Pengujian 8: Menghitung Ekspresi Matematika Masukan Pengguna yang Tidak Valid



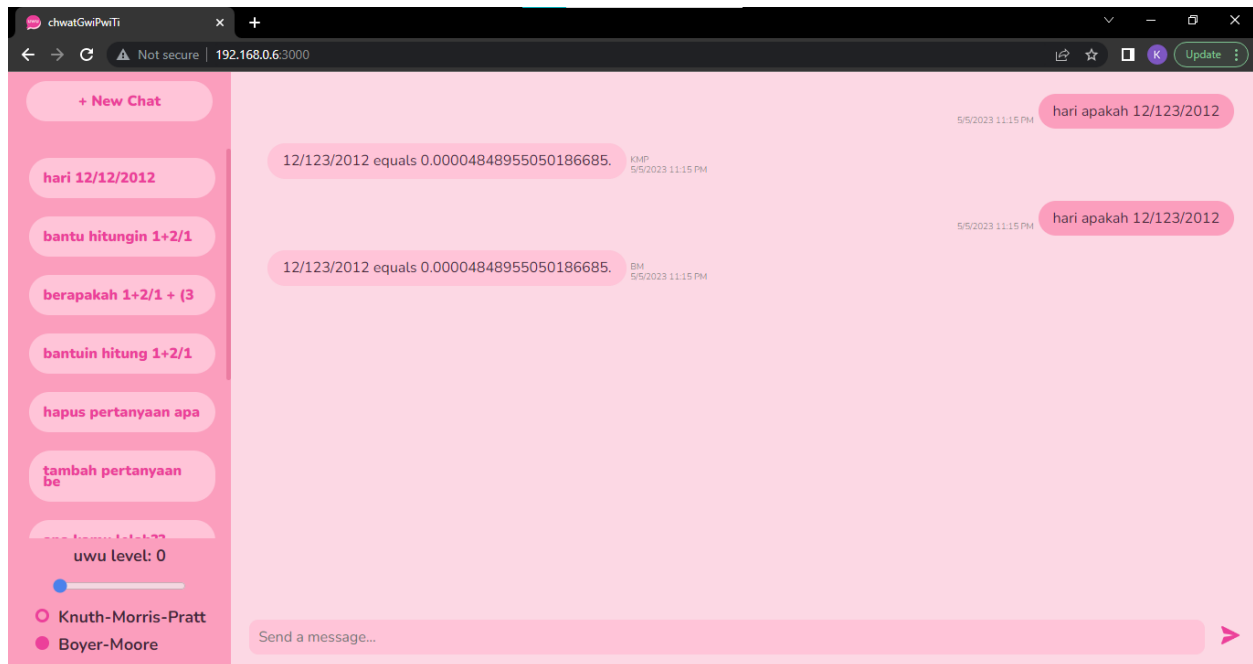
Gambar 4.3.8.1 Hasil Pengujian Fitur Menghitung Ekspresi Matematika yang Tidak Valid

4.3.9 Pengujian 9: Menampilkan Hari pada Tanggal Masukan Pengguna yang Valid



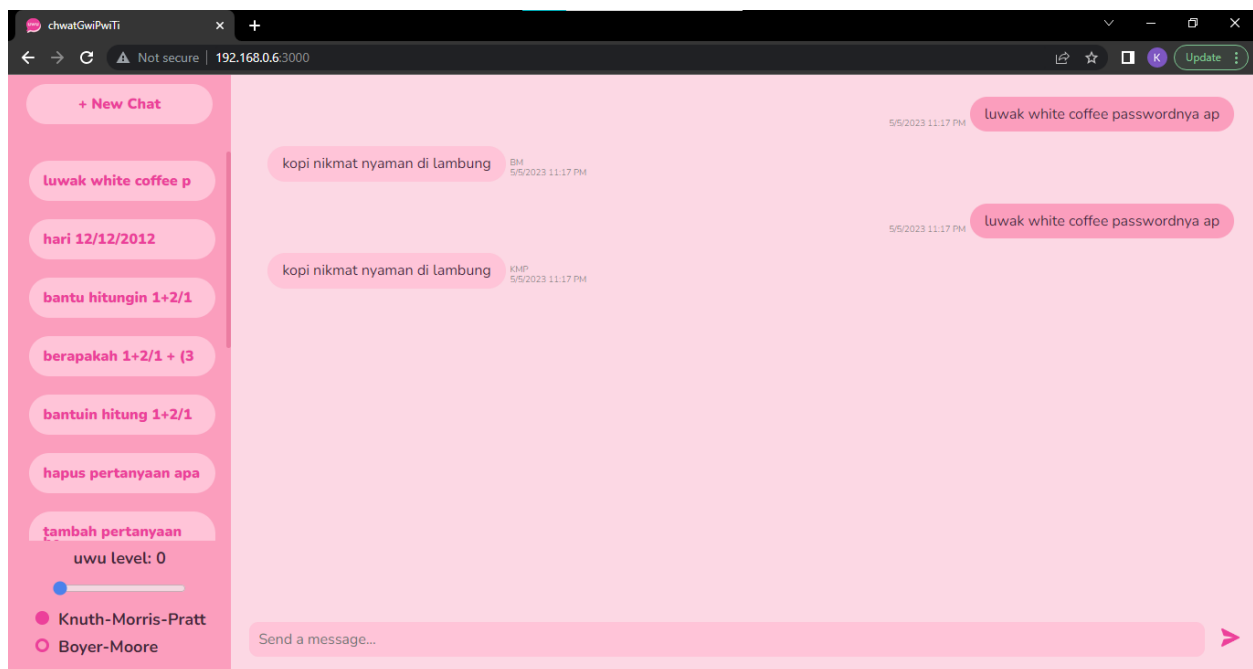
Gambar 4.3.9.1 Hasil Pengujian Fitur Menampilkan Hari pada Tanggal yang Valid

4.3.10 Pengujian 10: Menampilkan Hari pada Tanggal Masukan Pengguna yang Tidak Valid



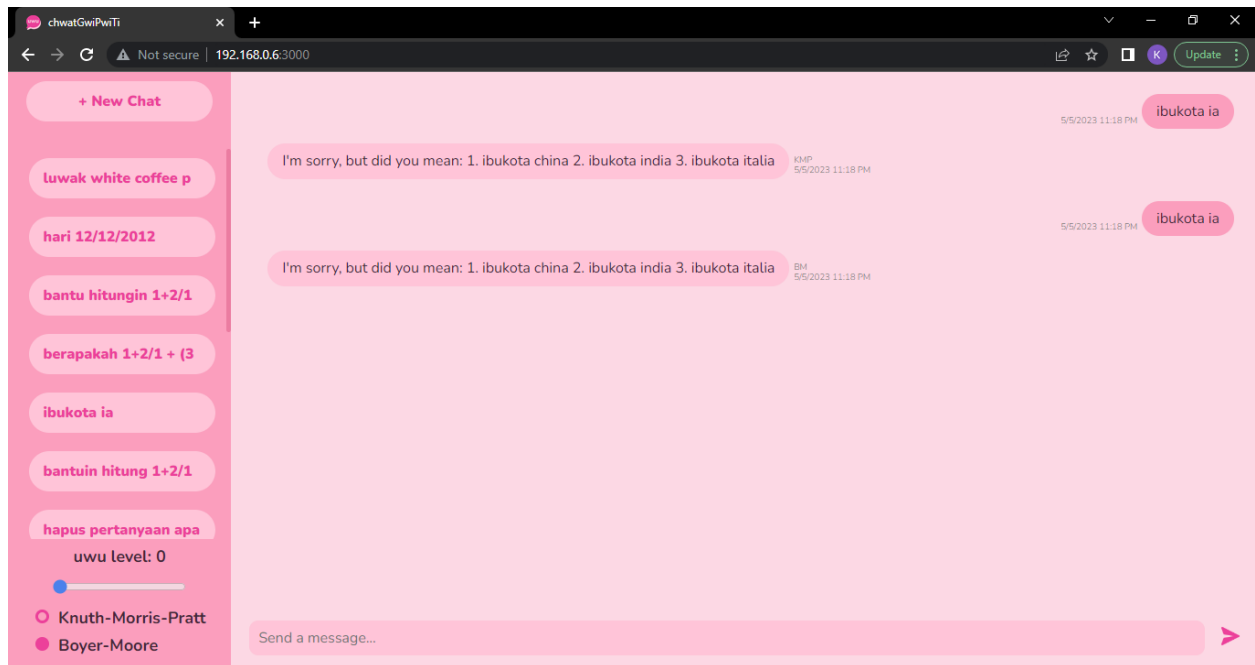
Gambar 4.3.10.1 Hasil Pengujian Fitur Menampilkan Hari pada Tanggal yang Tidak Valid

4.3.11 Pengujian 11: Menanyakan Pertanyaan dengan Kemiripan di Atas 90%



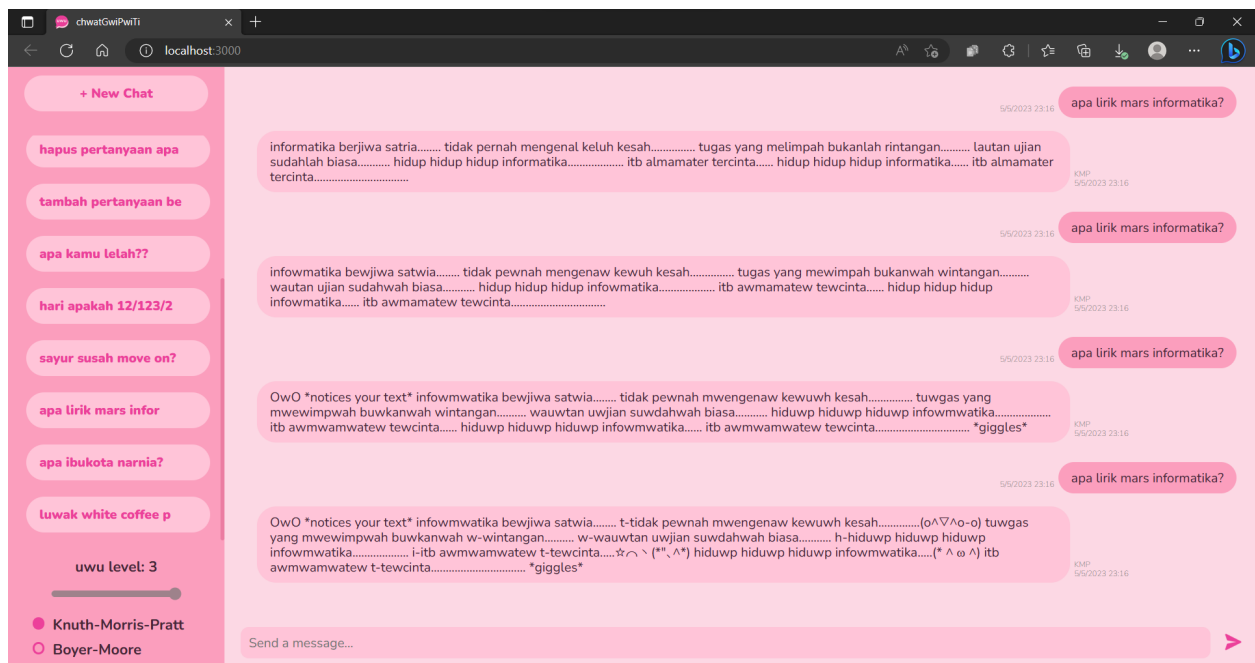
Gambar 4.3.11.1 Hasil Pengujian Fitur Pertanyaan ketika Kemiripan di Atas 90%

4.3.12 Pengujian 12: Menanyakan Pertanyaan dengan Kemiripan di Bawah 90%



Gambar 4.3.12.1 Hasil Pengujian Fitur Pertanyaan ketika Kemiripan di Bawah 90%

4.3.13 Pengujian 13: Fitur Tambahan UwUify



Gambar 4.3.13.1 Hasil Pengujian Fitur Tambahan UwUify

4.4 Analisis Hasil Pengujian

4.4.1 Pengujian 1: Menanyakan Pertanyaan pada Basis Data

Berdasarkan hasil pengujian pada bagian 4.3.1, pertanyaan yang ditanyakan adalah “Luwak White Coffee passwordnya apa”. Dengan algoritma Knuth-Morris-Pratt (KMP), jawaban yang ditampilkan oleh program adalah “kopi nikmat nyaman di lambung”. Pertanyaan yang sama ditanyakan pula dengan algoritma Boyer-Moore (BM). Jawaban yang diberikan oleh program tetap sama, yaitu “kopi nikmat nyaman di lambung”. Pada kasus ini, pertanyaan yang ditanyakan sama persis dengan pertanyaan yang disimpan pada basis data (persentase kemiripan 100%).

4.4.2 Pengujian 2: Menanyakan Pertanyaan yang Tidak Ada pada Basis Data

Berdasarkan hasil pengujian pada bagian 4.3.2, pertanyaan yang ditanyakan adalah “Apa ibukota Narnia?”. Pertanyaan tersebut tidak ada pada basis data, sehingga program menampilkan rekomendasi pertanyaan yang mirip dengan pertanyaan yang ditanyakan, yaitu “ibukota china”, “ibukota india”, dan “ibukota italia”.

4.4.3 Pengujian 3: Menambahkan Pertanyaan

Berdasarkan hasil pengujian pada bagian 4.3.3, pengguna menanyakan “Tambah pertanyaan Berapa harga tiket konser ITB Student Orchestra? dengan jawaban Harga tiket early bird untuk konser ITB Student Orchestra (ISO) adalah 65k untuk kategori normal dan 100k untuk kategori VIP, sedangkan untuk harga tiket regular adalah 100k untuk kategori normal dan 120k untuk kategori VIP. Selesai UAS, kita nonton konser dulu gak sih (~ ^ _ ^)~”.

Pertanyaan berhasil ditambahkan pada basis data dan dapat ditanyakan kembali.

4.4.4 Pengujian 4: Mengubah Jawaban dari Pertanyaan

Berdasarkan hasil pengujian pada bagian 4.3.4, pengguna menambahkan “Tambah pertanyaan apa kamu lelah dengan jawaban lelah sih, tapi gapapa :D”.

Jawaban dari pertanyaan “apa kamu lelah” berhasil diperbaharui. Program akan mengembalikan jawaban baru ketika pengguna menanyakan pertanyaan tersebut.

4.4.5 Pengujian 5: Menghapus Pertanyaan

Berdasarkan hasil pengujian pada bagian 4.3.5, pengguna menambahkan “Hapus pertanyaan sayur susah move on”. Dengan demikian, pertanyaan tersebut berhasil dihapus dari basis data. Ketika ditanyakan, maka program akan merekomendasikan pertanyaan serupa dengan “sayur susah move on”.

4.4.6 Pengujian 6: Menghapus Pertanyaan yang Tidak Ada pada Basis Data

Berdasarkan hasil pengujian pada bagian 4.4.6, pertanyaan masukan pengguna adalah “Apakah saya tampan”. Pertanyaan tersebut tidak dapat dihapus karena pertanyaan tersebut memang tidak terdapat di dalam database.

4.4.7 Pengujian 7: Menghitung ekspresi matematika masukan pengguna yang valid

Berdasarkan hasil pengujian pada bagian 4.4.7, ekspresi matematika masukan pengguna adalah “ $1+2/1+(3*4+5)$ ”. Ekspresi matematika tersebut valid dan program menampilkan hasil yang sesuai, yaitu 20.

4.4.8 Pengujian 8: Menghitung ekspresi matematika masukan pengguna yang tidak valid

Berdasarkan hasil pengujian pada bagian 4.4.8, ekspresi matematika masukan pengguna adalah “ $1+2/1+3*4+5$ ”. Ekspresi matematika tidak valid karena tidak ada tanda kurung buka sebelum kurung tutup, sehingga ekspresi tidak dapat dihitung.

4.4.9 Pengujian 9: Menampilkan hari pada tanggal masukan pengguna yang valid

Berdasarkan hasil pengujian pada bagian 4.4.9, *query* tanggal masukan pengguna adalah 12/12/2012. Hari pada tanggal tersebut pada kalender adalah hari Rabu, dan program menampilkan hari Rabu. Artinya program sudah menampilkan jawaban yang benar.

4.4.10 Pengujian 10: Menampilkan hari pada tanggal masukan pengguna yang tidak valid

Berdasarkan hasil pengujian pada bagian 4.4.10, *query* tanggal masukan pengguna adalah 12/123/2012. Masukan tersebut tidak valid karena pada format bulan terdapat tiga digit, sedangkan format yang diterima oleh program adalah dd/mm/yyyy. Karena tidak valid, *query* diproses sebagai ekspresi matematika.

4.4.11 Pengujian 11: Menanyakan Pertanyaan dengan Kemiripan di Atas 90%

Berdasarkan hasil pengujian pada bagian 4.4.11, pertanyaan masukan pengguna adalah “Luwak white coffee passwordnya ap”. Program menampilkan jawaban berupa “kopi nikmat nyaman di lambung”, yang merupakan jawaban dari pertanyaan “Luwak white coffee passwordnya apa”. Dapat dilihat bahwa walaupun pertanyaan tidak exact match dengan database, jika kemiripan lebih dari 90%, program tetap dapat memberikan jawaban yang diinginkan oleh pengguna.

4.4.12 Pengujian 12: Menanyakan Pertanyaan dengan Kemiripan di Bawah 90%

Berdasarkan hasil pengujian pada bagian 4.4.12, pertanyaan masukan pengguna adalah “Ibukota ia”. Tidak ada pertanyaan tersebut dalam database, dan kemiripan pertanyaan tersebut dengan pertanyaan dalam database tidak ada yang lebih dari 90%. Program menampilkan saran

pertanyaan berupa tiga pertanyaan pada database yang persentase kemiripannya paling besar. Setiap pertanyaan pada database dihitung levenshtein distance nya dengan pertanyaan masukan pengguna dan diurutkan, dan tiga pertanyaan yang ditampilkan merupakan tiga pertanyaan teratas.

4.4.13 Pengujian 13: Fitur Tambahan UwUify

Fitur ini merupakan fitur tambahan, sebagai bentuk kreativitas. Fitur ini mengubah teks biasa menjadi teks lucu. Terdapat beberapa level UwU, mulai dari level 0 (teks normal) hingga level 3 (teks UwU). Fitur ini memanfaatkan *regular expression* untuk mengubah beberapa huruf dan/atau suku kata menjadi huruf atau suku kata lainnya.

BAB V

PENUTUP

5.1 Simpulan

Berdasarkan hasil dari Tugas Besar ini, yaitu aplikasi berbasis web *chwatGwiPwiti*, dapat disimpulkan bahwa algoritma Knuth-Morris-Pratt dan algoritma Boyer Moore dapat menyelesaikan permasalahan pattern matching dengan baik. Fitur yang akan diakses oleh pengguna dapat ditentukan oleh string matching dengan regular expression. Jika ada kesalahan pengetikan dari pengguna, program tetap dapat menampilkan hasil yang diinginkan pengguna asalkan kemiripannya di atas 90% dengan algoritma Levenshtein.

5.2 Saran

Sebaiknya lebih sering mencari tahu tentang web development agar tidak kesusahan dalam mengembangkan web. Jika tidak pernah mengelola web sebelumnya, akan mengalami kesusahan untuk menghubungkan frontend, backend, dan database. Aplikasi ini dapat dikembangkan dengan pemberian fitur berupa *me-rename* dan menghapus chat history dan login.

5.3 Komentar/Refleksi

1. Kevin John Wesley (13521042)

Dengan tugas besar ini, saya jadi mengerti tentang implementasi pattern matching ke dalam permasalahan di dunia nyata. Menurut saya, algoritma ini sangat banyak membantu untuk banyak permasalahan di dunia luar. Ditambah dengan spesifikasi tugas besar ini yang mengharuskan mahasiswa mengembangkan aplikasi web, membuat saya banyak belajar hal tentang *web development*.

2. Arleen Chrysanthia Gunardi (13521059)

Melalui tugas besar ini, saya belajar banyak hal baru, salah satunya tentang *web development*. Hal ini merupakan hal yang sangat baru bagi saya, sehingga tugas besar ini sangat menantang. Namun, proses pengerjaan tugas ini cukup asik tetapi melelahkan. Dari sini, saya juga belajar tentang *string matching* dan *pattern matching* yang telah diajarkan di kelas.

3. Jericho Russel Sebastian (13521107)

Tugas besar ini pengalaman pertama saya terjun ke dalam dunia *webapp development*. Sebelumnya, saya belum pernah memiliki ketertarikan yang kuat terhadap pengembangan aplikasi berbasis *web*. Namun, melalui proses dan perjuangan menyelesaikan tugas ini dalam kurun waktu yang cukup terbilang singkat, terlepas dari kepenatan luar biasa, nyatanya tugas ini adalah kesempatan emas, yang lewatnya saya dapat mempelajari banyak hal, serta memperkuat pemahaman mengenai materi yang sudah dipelajari.

DAFTAR REFERENSI

- Munir, Rinaldi. 2023. Pencocokan string (String matching/pattern matching). Homepage Rinaldi Munir Sekolah Teknik Elektro dan Informatika (STEI) ITB.
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>
- Munir, Rinaldi. 2023. Pencocokan string dengan Regular Expression (Regex). Homepage Rinaldi Munir Sekolah Teknik Elektro dan Informatika (STEI) ITB.
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>

LAMPIRAN

Tautan repository tugas besar: https://github.com/JerichoFletcher/Tubes3_13521042