

**Laporan Tugas Besar 3 IF2211 Strategi Algoritma
Semester II Tahun 2022/2023**

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



Disusun oleh:

Angger Ilham Amanullah	13521001
Azmi Hasna Zahrani	13521006
Agsha Athalla Nurkareem	13521027

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2023**

DAFTAR ISI

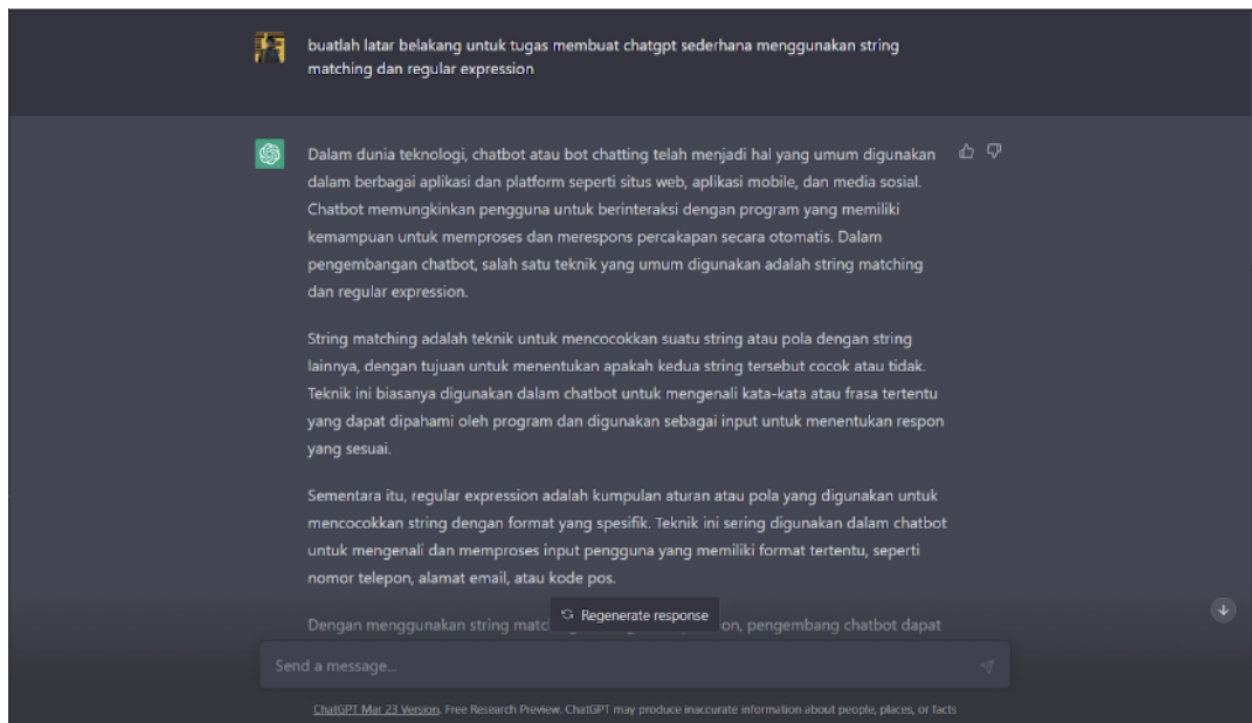
DAFTAR ISI	1
BAB I	2
DESKRIPSI TUGAS	2
BAB II	9
LANDASAN TEORI	9
2.1. Dasar Teori	9
2.1.1. Algoritma Knuth-Morris-Pratt	9
2.1.2. Algoritma Boyer-Moore	9
2.1.3. Regular Expression	10
2.2. Angger Intelligence Web Application	11
BAB III	12
ANALISIS PEMECAHAN MASALAH	12
3.1. Langkah-langkah Pemecahan Masalah	12
3.1.1. Fitur Pertanyaan Teks	12
3.1.2. Fitur Kalkulator	12
3.1.3. Fitur Tanggal	12
3.1.4. Tambah Pertanyaan dan Jawaban ke Database	12
3.1.5. Hapus Pertanyaan Dari Database	13
3.2. Fitur Fungsional dan Arsitektur Aplikasi Web yang dibangun.	13
3.2.1. Fitur Fungsional	13
3.2.2. Arsitektur Web	13
BAB IV	14
IMPLEMENTASI DAN PENGUJIAN	14
4.1. Spesifikasi Teknis Program	14
4.1.1. Struktur Data	14
4.1.2. Fungsi dan Prosedur yang dibangun	14
4.2. Tata Cara Penggunaan Program	15
4.3. Hasil Pengujian	16
4.3.1. Hasil Pengujian Tambah Pertanyaan, Fitur Pertanyaan, dan Hapus Pertanyaan	16
4.3.2. Hasil Pengujian Fitur Kalkulator dan Tanggal	16
4.4. Analisis Hasil Pengujian	17
BAB V	18
KESIMPULAN DAN SARAN	18
5.1. Kesimpulan	18

5.2. Saran	18
5.3. Refleksi	18
5.4. Tanggapan Kelompok	18
DAFTAR PUSTAKA	19
LAMPIRAN	20

BAB I

DESKRIPSI TUGAS

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

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.

Dalam tugas besar ini, anda diminta untuk membangun sebuah aplikasi ChatGPT sederhana dengan mengaplikasikan pendekatan QA yang paling sederhana tersebut. Pencarian pertanyaan yang paling mirip dengan pertanyaan yang diberikan pengguna dilakukan dengan algoritma pencocokan string Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM). Regex digunakan untuk menentukan format dari pertanyaan (akan dijelaskan lebih lanjut pada bagian fitur aplikasi). Jika tidak ada satupun pertanyaan pada database yang exact match dengan pertanyaan pengguna melalui algoritma KMP ataupun BM, maka gunakan pertanyaan termirip dengan kesamaan setidaknya 90%. Apabila tidak ada pertanyaan yang kemiripannya di atas 90%, maka chatbot akan memberikan maksimum 3 pilihan pertanyaan yang paling mirip untuk dipilih oleh pengguna.

Perhitungan tingkat kemiripan dibebaskan kepada anda asalkan dijelaskan di laporan, namun disarankan menggunakan salah satu dari algoritma Hamming Distance, Levenshtein Distance, ataupun Longest Common Subsequence.

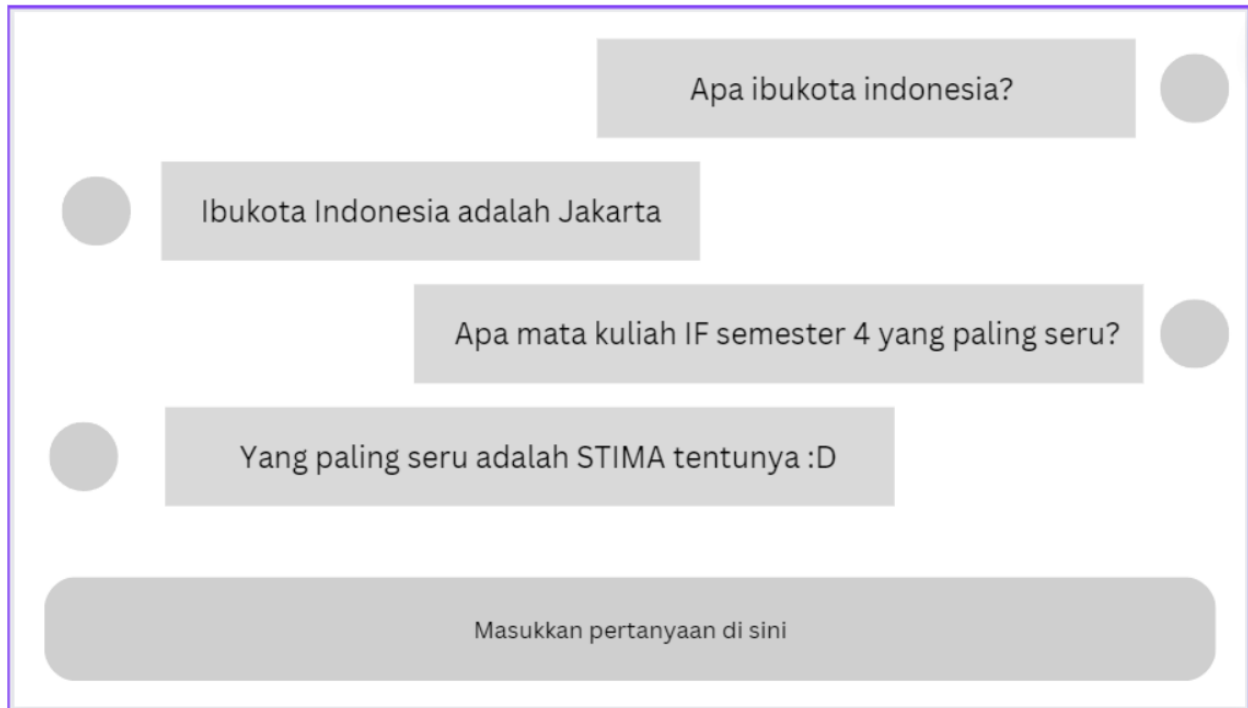
Fitur-Fitur Aplikasi:

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

1. Fitur pertanyaan teks (didapat dari database)
Mencocokkan pertanyaan dari input pengguna ke pertanyaan di database menggunakan algoritma KMP atau BM.
2. Fitur kalkulator
Pengguna memasukkan input query berupa persamaan matematika. Contohnya adalah $2*5$ atau $5+9*(2+4)$. Operasi cukup Tambah, kurang, kali, bagi, pangkat, kurung.
3. Fitur tanggal
Pengguna memasukkan input berupa tanggal, lalu chatbot akan merespon dengan hari apa di tanggal tersebut. Contohnya adalah 25/08/2023 maka chatbot akan menjawab dengan hari senin.
4. Tambah pertanyaan dan jawaban ke database
Pengguna dapat menambahkan pertanyaan dan jawabannya sendiri ke database dengan query contoh “Tambahkan pertanyaan xxx dengan jawaban yyy”. Menggunakan algoritma string matching untuk mencari tahu apakah pertanyaan sudah ada. Apabila sudah, maka jawaban akan diperbaharui.
5. Hapus pertanyaan dari database
Pengguna dapat menghapus sebuah pertanyaan dari database dengan query contoh “Hapus pertanyaan xxx”. Menggunakan string algoritma string matching untuk mencari pertanyaan xxx tersebut pada database.

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

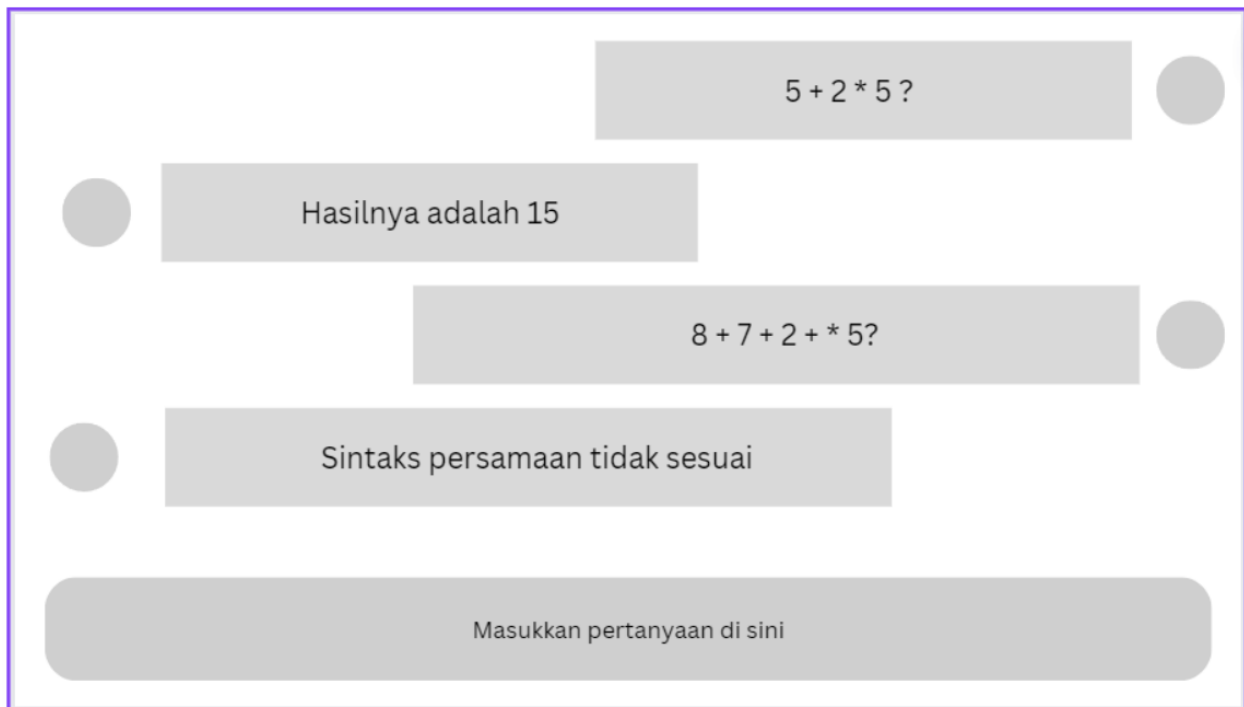
(Note: Tidak wajib mengikuti ilustrasi ini, tampilan disamakan dengan chatGPT juga boleh)



Gambar 2. Ilustrasi Fitur Pertanyaan teks kasus exact



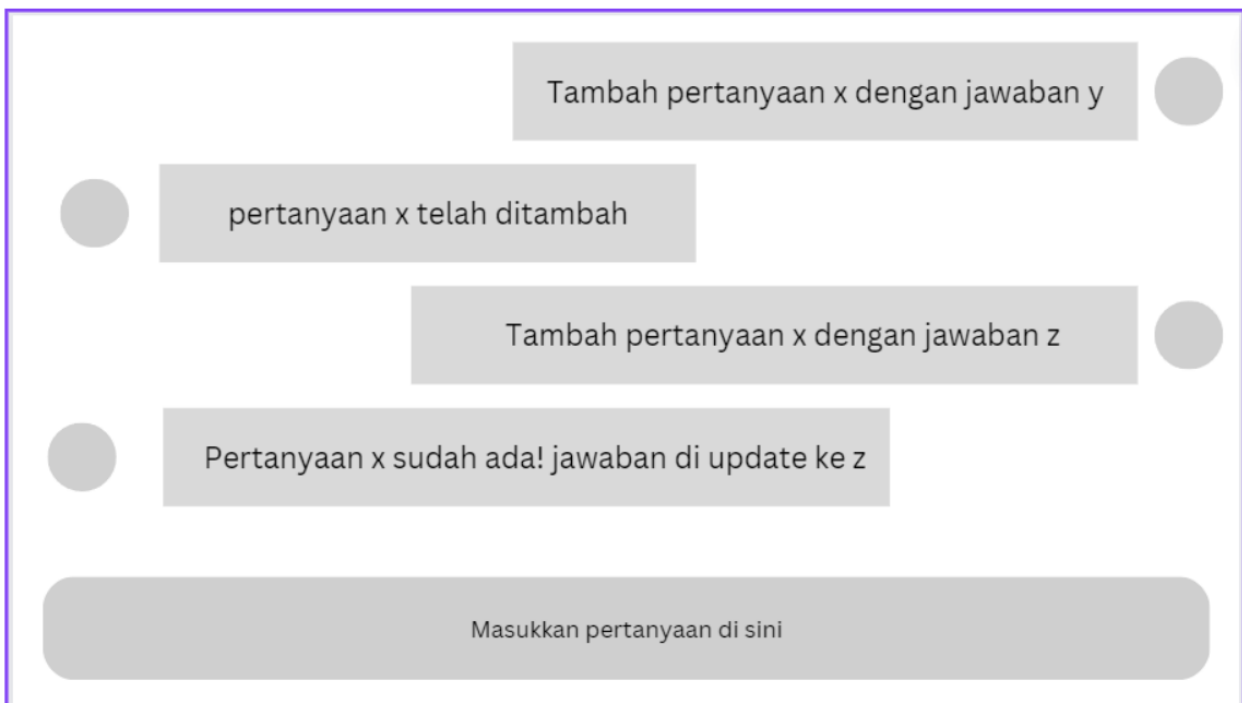
Gambar 3. Ilustrasi Fitur Pertanyaan teks kasus tidak exact



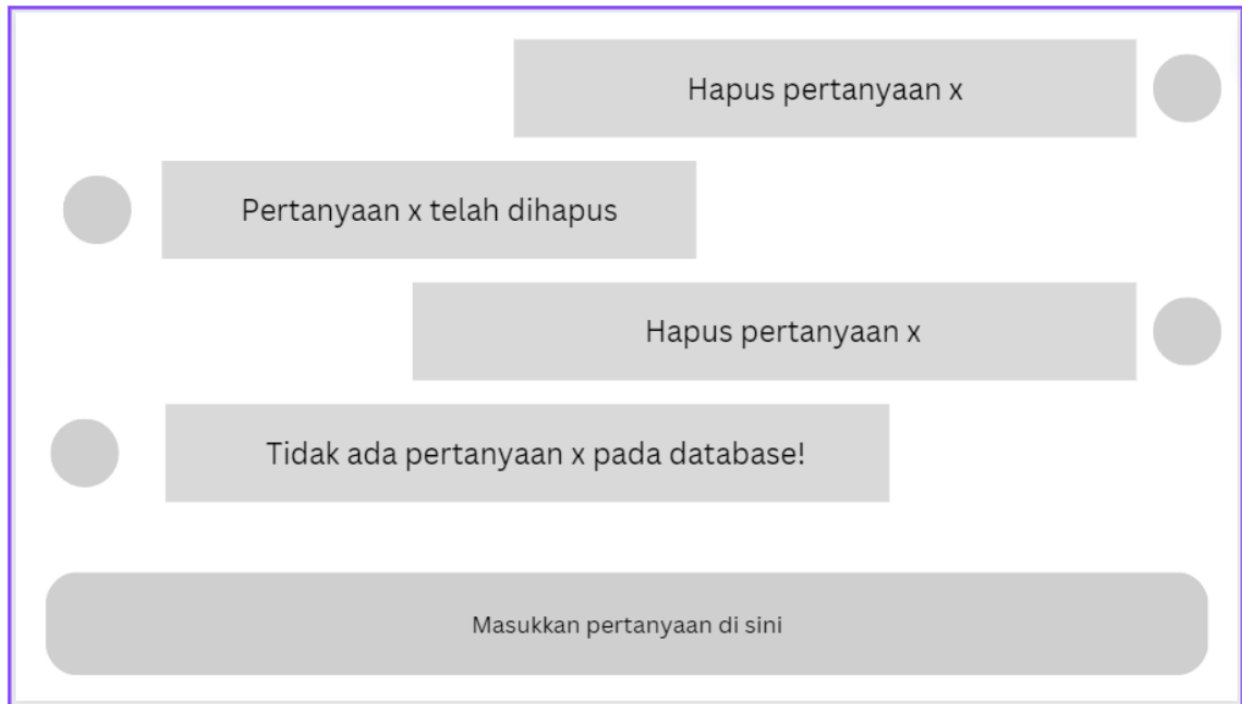
Gambar 4. Ilustrasi Fitur Kalkulator



Gambar 5. Ilustrasi Fitur Tanggal

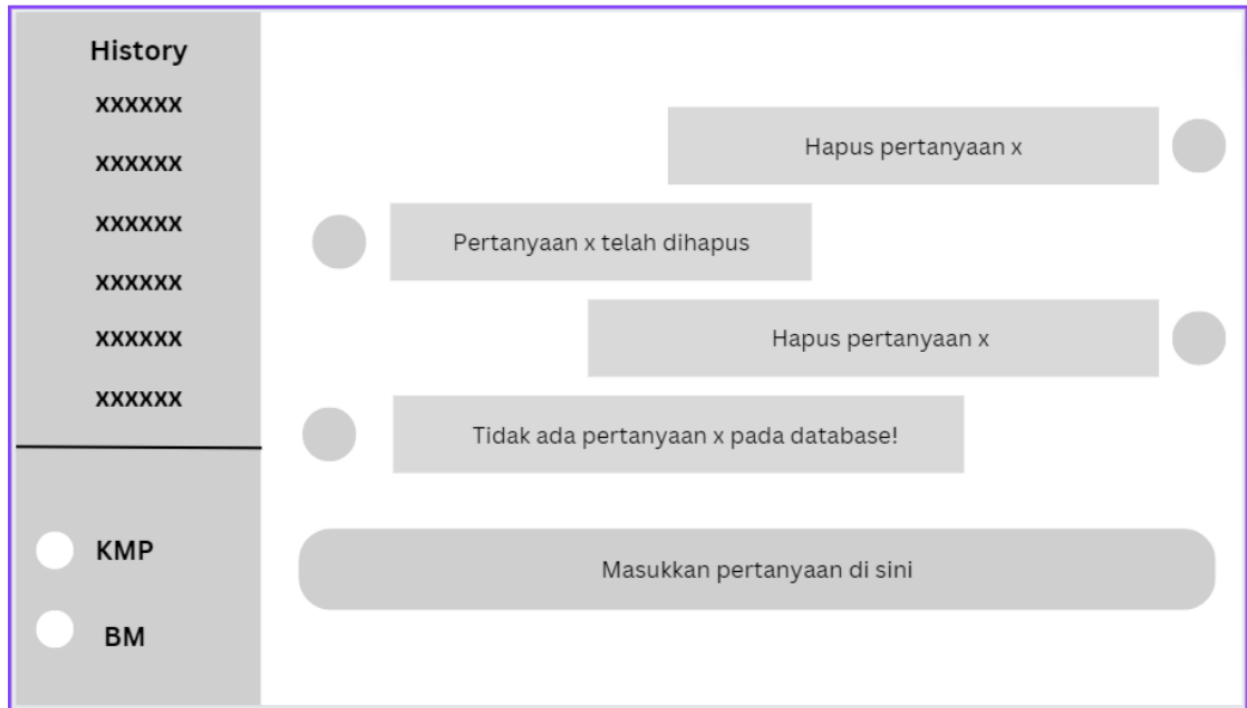


Gambar 6. Ilustrasi Fitur Tambah Pertanyaan



Gambar 7. Ilustrasi Fitur Hapus Pertanyaan

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



Gambar 8. Ilustrasi Keseluruhan

BAB II

LANDASAN TEORI

2.1. Dasar Teori

2.1.1. Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt merupakan salah satu algoritma pencocokan string yang dikembangkan oleh Donald Ervin Knuth, James H. Morris, serta Vaughan R. Pratt secara terpisah. Algoritma KMP melakukan pencocokan string dengan melihat pola dalam string dengan left-to-right order mirip dengan brute force, tetapi lebih ‘pintar’. Algoritma ini akan mencocokkan karakter per karakter dari kiri ke kanan hingga salah satu dari kondisi tersebut di bawah ini terpenuhi:

- a. Karakter di pattern dan di teks yang dibandingkan tidak cocok (*mismatch*).
- b. Semua karakter di pattern cocok.

Algoritma Knuth-Morris-Pratt menggunakan fungsi pembatas (*border function*) yang digunakan untuk memproses pola untuk menemukan pola awalan yang cocok dengan pola itu sendiri.

j = posisi *mismatch* pada $P[]$

k = posisi sebelum *mismatch* ($k=j-1$)

Border function $b(k)$ yang juga sering disebut sebagai *failure function* didefinisikan sebagai ukuran terbesar awalan dari $P[0\dots k]$ yang juga merupakan akhiran dari $P[1..k]$.

Kompleksitas waktu algoritma Knuth-Morris-Pratt dihitung dari kompleksitas menghitung fungsi pinggiran $O(m)$ serta kompleksitas pencarian string $O(n)$. Sehingga, kompleksitas waktu algoritma Knuth-Morris-Pratt adalah $O(m+n)$, jauh lebih cepat daripada brute force. Selain itu, algoritma ini juga memiliki kelebihan yaitu algoritmanya tidak perlu bergerak secara *backwards* pada input teks sehingga baik untuk memproses file yang sangat besar. Namun, algoritma ini juga memiliki kekurangan yaitu apabila ukuran alphabet meningkat, algoritma tidak berjalan begitu sebaik sebelumnya.

2.1.2. Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan algoritma yang dikembangkan oleh Bob Boyer dan J. Strother Moore. Algoritma ini menjadi standar untuk berbagai literatur pencarian string. Boyer-Moore *pattern matching* didasarkan pada dua teknik, yaitu teknik *looking-glass* dan teknik *character-jump*. Teknik *looking-glass* bekerja dengan mencari P (*pattern*) di dalam T (teks) dengan berjalan dari belakang ke depan (*backward*) melalui P . Sedangkan, teknik *character-jump* bekerja apabila terdapat *mismatch* pada $T[i] \neq x$, maka karakter pada $P[j]$ tidak sama dengan karakter pada $T[i]$. Terdapat 3 kasus yang mungkin terjadi.

- i. Kasus I

Ketika terjadi mismatch pada $T[i]$ dan $P[j]$, $T[i] \neq x$ dan terdapat x di P dengan indeks lebih kecil daripada j , maka seolah-olah menggeser P ke

kanan agar posisi x di $T[i]$ sejajar dengan posisi kemunculan terakhir x di P .

ii. Kasus II

Ketika terjadi mismatch pada $T[i]$ dan $P[j]$, $T[i] = x$ dan terdapat x di P pada posisi indeks yang lebih besar daripada j , maka seolah-olah menggeser P ke satu karakter ke kanan agar indeks terakhir P sejajar dengan posisi akhir $T[i+1]$.

iii. Kasus III

Ketika terjadi mismatch pada $T[i]$ dan $P[j]$, $T[i] = x$ dan tidak ditemukan karakter x pada P , maka seolah-olah menggeser P agar posisi indeks pertama P ($P[0]$) sejajar dengan indeks $i+1$.

Pada kasus i dan ii diperlukan informasi mengenai kemunculan karakter x pada P untuk menentukan banyak pergeseran yang perlu dilakukan. Informasi ini dapat didapat dengan mengaplikasikan *Last Occurrence Function*.

Kompleksitas waktu algoritma pada kasus *worst case* adalah $O(nm + A)$. Namun, apabila alphabet (A) memiliki ukuran yang besar, Boyer-Moore akan bekerja dengan cepat dan apabila A memiliki ukuran yang kecil, algoritma ini akan bekerja dengan lambat. Algoritma Boyer-Moore memiliki kompleksitas waktu yang lebih cepat dibanding Brute Force untuk pencarian *English text*.

2.1.3. Regular Expression

Regular Expression (Regex) merupakan notasi yang digunakan untuk mendeskripsikan pola dari kata yang ingin dicari. Regex diapit oleh tanda slash ('/') dalam pola awk yang sesuai dengan input record yang teksnya dimiliki oleh set dari string. Regex ditulis dalam bahasa formal dan banyak dimanfaatkan untuk proses validasi string. Terdapat beberapa konsep yang perlu dipahami dalam Regex, yaitu sebagai berikut.

- Karakter: Regular expression terdiri dari karakter-karakter yang membentuk pola. Karakter tersebut bisa berupa huruf, angka, simbol, atau karakter khusus seperti $\backslash d$ (digit), $\backslash w$ (word character), dan lain-lain.
- Operator: Regular expression juga menggunakan operator-operator seperti $*$ (zero or more), $+$ (one or more), $?$ (zero or one), $|$ (atau), dan lain-lain. Operator-operator ini digunakan untuk menggabungkan karakter-karakter atau kelompok-kelompok karakter.
- Kelompok: Dalam regular expression, karakter-karakter dapat dikelompokkan menggunakan tanda kurung (). Kelompok-kelompok ini dapat digunakan untuk memanipulasi teks secara bersamaan.
- Karakter escape: Beberapa karakter dalam regular expression memiliki arti khusus dan harus di-escape dengan menggunakan karakter backslash. Contohnya adalah . (titik), * (bintang), dan lain-lain.

- e. Metakarakter: Metakarakter adalah karakter-karakter khusus dalam regular expression yang digunakan untuk mencari pola-pola tertentu. Beberapa metakarakter yang sering digunakan antara lain ^ (awal baris), \$ (akhir baris), \b (awal atau akhir kata), dan lain-lain.

2.2. Angger Intelligence Web Application

Angger Intelligence merupakan sebuah aplikasi web chat bot yang dapat menerima pertanyaan serta menampilkan jawaban dari pertanyaan yang diinput. Aplikasi ini dibangun dengan backend menggunakan bahasa Golang, frontend menggunakan react.js, serta penyimpanan basis data menggunakan mysql. Aplikasi Angger Intelligence bekerja dengan pencocokan string menggunakan algoritma Knuth-Morris-Pratt, Boyer-Moore, serta Regular Expression. Selain menerima dan menjawab pertanyaan, aplikasi web ini memiliki fitur kalkulator, tanggal, menambah pertanyaan serta jawaban ke dalam database, serta menghapus pertanyaan dari database.

BAB III

ANALISIS PEMECAHAN MASALAH

3.1. Langkah-langkah Pemecahan Masalah

3.1.1. Fitur Pertanyaan Teks

Pada fitur tanya jawab, kelompok membuat *source* program *front-end* menggunakan react serta *back-end* menggunakan golang. Dalam program front-end terdapat file App.js serta App.css sebagai stylenya. Sedangkan, dalam *back-end* terdapat file database berupa setup.go serta source yaitu BM.go, controller.go, KMP.go, serta Levenshtein.go. Fitur tanya jawab dapat membaca input pertanyaan yang kemudian akan dilakukan pencocokan string antara input dengan database yang ada menggunakan regex serta salah satu algoritma KMP atau BM yang dapat dipilih pada antarmuka. Setelah melakukan pencocokan string input dengan database, akan ditampilkan jawaban yang sesuai dengan pertanyaan.

3.1.2. Fitur Kalkulator

Pada fitur kalkulator, kelompok membuat *source* program pada folder *back-end* yaitu dalam file bernama calculator.go. Dengan fitur ini, pengguna dapat menanyakan operasi matematika berupa tambah, kurang, kali, bagi, dan pangkat. Fitur ini juga menggunakan regex untuk melakukan pencocokan input string dan database.

3.1.3. Fitur Tanggal

Pada fitur tanggal, kelompok membuat *source* program dalam folder *back-end* yaitu pada file bernama date.go. Dengan fitur ini, pengguna dapat memasukkan tanggal untuk mendapatkan nama hari pada tanggal tersebut. Fitur ini menggunakan regex untuk memastikan input string berupa tanggal, kemudian string tersebut akan dipisahkan antara tanggal, bulan, serta tahun. Setelah itu, tanggal, bulan, dan tahun dimasukkan ke dalam rumus Zeller's Congruence untuk mendapatkan indeks hari dari 0-6 (Minggu-Sabtu).

3.1.4. Tambah Pertanyaan dan Jawaban ke Database

Pada fitur ini, kelompok membuat *source* program pada folder *back-end* yaitu pada file bernama manageDB.go. Dengan fitur ini, pengguna dapat memasukkan pertanyaan serta jawaban yang diinginkan dari pertanyaan tersebut ke dalam database. Pada fitur ini juga menggunakan regex sebagai pencocokan stringnya, dengan pattern "Tambahkan pertanyaan ... dengan jawaban ...". Selain menambahkan pertanyaan dan jawaban, dalam fitur ini juga dapat melakukan update jawaban dari pertanyaan yang telah ada.

3.1.5. Hapus Pertanyaan Dari Database

Pada fitur ini, kelompok membuat *source* program pada folder *back-end* yaitu pada file bernama *manageDB.go*. Dengan fitur ini, pengguna dapat menghapus pertanyaan yang telah ada dalam database. Pada fitur ini menggunakan regex sebagai pencocokan stringnya, dengan pattern “Hapus pertanyaan ...”.

3.2. Fitur Fungsional dan Arsitektur Aplikasi Web yang dibangun.

3.2.1. Fitur Fungsional

- a. Melakukan pencocokan string yaitu untuk mengimplementasikan chatbot (menerima pertanyaan dan menjawab pertanyaan).
- b. Melakukan perhitungan matematika menggunakan fitur kalkulator yaitu untuk menghitung tambah, kurang, kali, bagi, dan pangkat.
- c. Melakukan perhitungan Zeller’s Congruence untuk menghitung tanggal dan mendapatkan indeks hari dari tanggal tersebut.
- d. Melakukan penambahan pertanyaan dan jawaban ke dalam database yang dimiliki.
- e. Melakukan penghapusan pertanyaan dari database yang dimiliki.

3.2.2. Arsitektur Web

- a. Front End

Implementasi *front-end* dari web Angger Intelligence dibuat dengan menggunakan library ReactJS serta desain web yang dibuat pada aplikasi figma. Web dibangun dengan code utama pada *App.js* yang hanya memiliki satu halaman utama untuk menerima chat pertanyaan dan menampilkan hasil dari pertanyaan. Pemanggilan algoritma string matching dilakukan dengan memanfaatkan *library* *axios*. History chat dapat disimpan dan pengguna dapat melakukan open new chat dengan menekan tombol “New Chat”.

- b. Back End

Implementasi *back-end* dari web Angger Intelligence dibuat dengan menggunakan bahasa Golang serta database menggunakan MySQL. Pada folder *back-end* terdapat beberapa file yang berisi setiap fitur yang terdapat pada web. File tersebut yaitu *BM.go*, *calculator.go*, *controller.go*, *date.go*, *KMP.go*, *Levenshtein.go*, *manageDB.go*, serta *setup.go*.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Spesifikasi Teknis Program

4.1.1. Struktur Data

Struktur data utama yang digunakan untuk pengiriman dan pengambilan data dari basis data dengan menggunakan MySQL.

4.1.2. Fungsi dan Prosedur yang dibangun

No	Fungsi/Prosedur (Parameter)	Keterangan
1.	min(a int, b int)	Fungsi ini digunakan untuk membandingkan dua buah integer dan mengembalikan integer yang paling kecil.
2.	buildLast(pattern string) []int	Fungsi ini digunakan untuk
3.	bmMatch(text string, pattern string) int	Fungsi ini digunakan untuk string matching dengan menggunakan algoritma BM.
4.	getVal(calculator string) string	Fungsi ini digunakan untuk mendapatkan value dan memvalidasi input operasi matematika.
5.	getDay(dateString string) string	Fungsi ini digunakan untuk memvalidasi input berupa tanggal, bulan, dan tahun serta menghitungnya untuk mendapatkan hari dari tanggal tersebut.
6.	kmpMatch(text string, pattern string) int	Fungsi ini digunakan untuk string matching dengan menggunakan algoritma KMP.
7.	computeBorder(pattern string) []int	Fungsi ini digunakan untuk

		menghitung border function.
8.	Minimum (a int, b int, c int) int	Fungsi ini digunakan untuk menentukan integer terkecil.
9.	Levenshtein(text string, pattern string) int	Fungsi ini digunakan untuk membandingkan antara string input dengan database.
10.	distance(text string, pattern string) float32	Fungsi ini digunakan untuk menghitung presentase kemiripan antara string input dengan database.
11.	AddPertanyaan(str string, db *sql.DB) (string, bool)	Fungsi ini digunakan untuk menambahkan pertanyaan serta jawaban dari pertanyaan tersebut ke dalam database dan mengupdate jawaban dari pertanyaan tersebut.

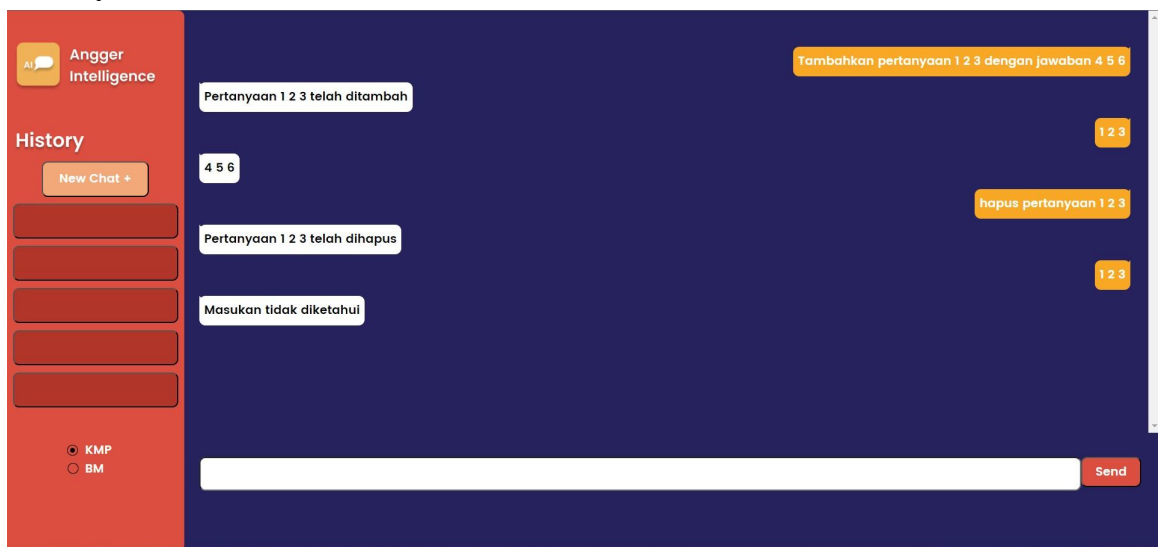
4.2. Tata Cara Penggunaan Program

- Pengguna dapat mengakses program dalam repository github yang tercantum dalam lampiran laporan ini.
- Clone repository github dan ikuti cara menjalankan program sesuai yang ada pada readme.
- Program akan berjalan dengan menampilkan antarmuka web.
- Pengguna dapat menggunakan fitur-fitur yang ada dalam program.
 - a. Fitur pertanyaan teks dapat dijalankan dengan mengetikkan pertanyaan pada kolom teks lalu menekan enter atau tombol send. Kemudian jawaban akan ditampilkan sesuai dengan string matching antara pertanyaan dengan database yang dimiliki.
 - b. Fitur kalkulator dapat dijalankan dengan mengetikkan pertanyaan matematika pada kolom teks lalu menekan enter atau tombol send. Kemudian chatbot akan menampilkan jawaban dengan melakukan perhitungan yang ada pada file calculator.go.

- c. Fitur tanggal dapat dijalankan dengan mengetikkan tanggal dengan format tanggal/bulan/tahun lalu program akan menghitung tanggal tersebut dan mencari indeks hari pada tanggal tersebut.
- d. Tambah pertanyaan dan jawaban ke database dapat dijalankan dengan mengetik format “Tambah pertanyaan ... dengan jawaban ...” lalu menekan enter atau menekan tombol send. Lalu akan ditampilkan hasil bahwa pertanyaan tersebut telah ditambahkan ke dalam database.
- e. Hapus pertanyaan dari database dapat dijalankan dengan mengetik format

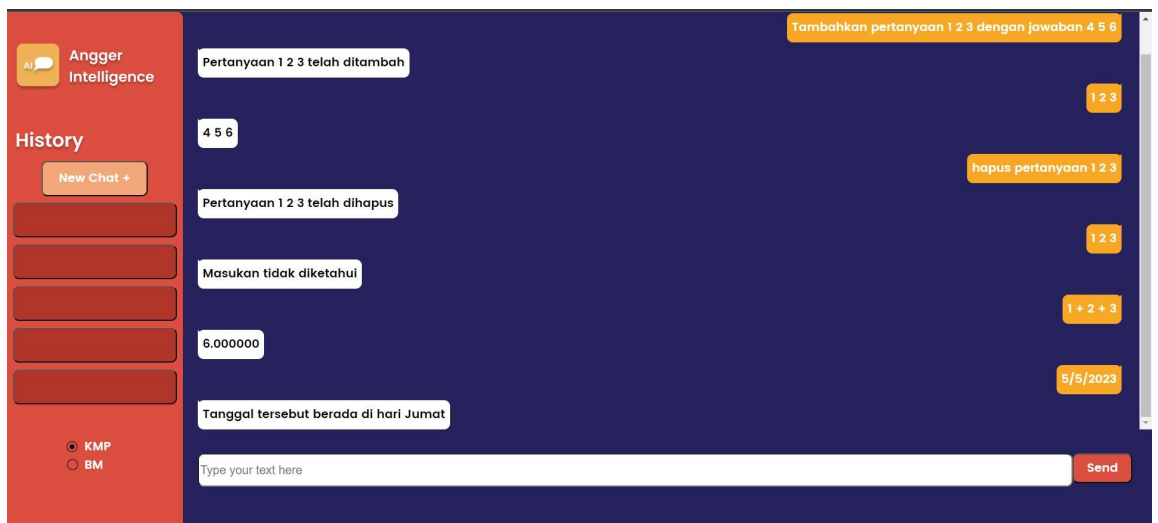
4.3. Hasil Pengujian

4.3.1. Hasil Pengujian Tambah Pertanyaan, Fitur Pertanyaan, dan Hapus Pertanyaan



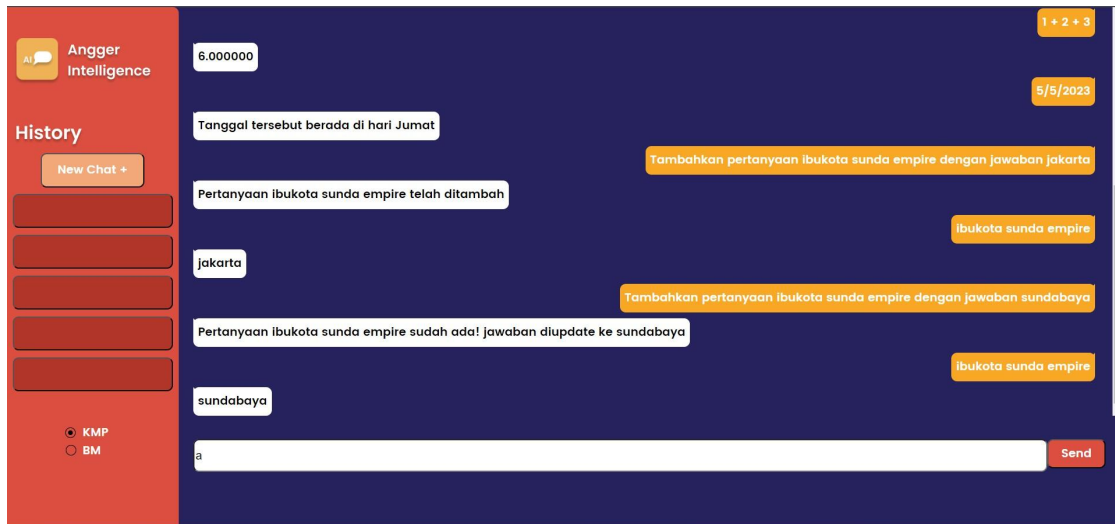
Gambar 4.3.1. Hasil Pengujian 1

4.3.2. Hasil Pengujian Fitur Kalkulator dan Tanggal



Gambar 4.3.2. Hasil Pengujian 2

4.3.3. Hasil Pengujian Update Jawaban



Gambar 4.3.3. Hasil Pengujian 3

4.4. Analisis Hasil Pengujian

Berdasar hasil pengujian yang telah dilakukan, aplikasi web Angger Intelligence secara garis besar sudah bekerja sesuai dengan fitur-fiturnya. Pada fitur pertanyaan, chatbot telah dapat menampilkan jawaban yang sesuai dengan database yang ada dengan mengimplementasi regex. Pada fitur kalkulator, chatbot telah dapat menampilkan jawaban dengan tepat dalam bentuk tipe data float. Pada fitur tanggal, chatbot telah dapat menampilkan jawaban hari dengan benar. Begitu pula dengan fitur tambah pertanyaan, hapus pertanyaan, dan update pertanyaan, semua telah dapat diimplementasikan dengan baik dalam web serta database.

Untuk string matching dalam program ini, pengguna dapat memilih untuk menggunakan algoritma KMP atau BM dengan menekan pilihan radio bot yang ada pada *side bar*.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Algoritma Knuth-Morris-Pratt dan Boyer-Moore merupakan algoritma pencocokan string yang dapat digunakan untuk membuat aplikasi chatbot yang dapat menerima input string serta menampilkan jawabannya. Knuth-Morris-Pratt mencocokkan pola string dari kanan ke kiri. Sedangkan, Boyer-Moore mencocokkan string dengan teknik *looking-glass* dan *character-jump*. Selain kedua algoritma tersebut, Regular Expression juga dapat digunakan untuk mengimplementasikan string matching dalam aplikasi chatbot. Pada tugas besar 2 IF2211 Strategi Algoritma ini, kelompok telah mengimplementasikan algoritma tersebut dalam program *Angger Intelligence*.

5.2. Saran

Berikut adalah saran yang kami berikan untuk pengerjaan tugas besar ini:

1. Eksplorasi yang mendalam terhadap bahasa golang serta library ReactJS dalam *web development*.
2. Memerhatikan kebutuhan instalasi ReactJS dalam pengerjaan program.
3. Tidak mengerjakan tugas dekat dengan deadline.

5.3. Refleksi

Melalui tugas besar ini, kelompok belajar memahami algoritma Knuth-Morris-Pratt, Boyer-Moore, dan Regular Expression serta implementasinya dalam permasalahan string matching. Kelompok belajar untuk bekerja sama dan bekerja keras memikirkan algoritma-algoritma tersebut serta desain UI yang ada dalam aplikasi web yang dibuat. Selain itu, melalui batasan yang diberikan pada tugas ini, kelompok belajar untuk mengikuti arahan yang diberikan sesuai spesifikasi.

5.4. Tanggapan Kelompok

Tanggapan kelompok terhadap tugas besar 3 IF2211 Strategi Algoritma adalah sebagai berikut.

NIM	Tanggapan
13521001	Speknya banyak yang miss.
13521006	Di awal semangat ngerjain makin deket deadline makin banyak tubes makin susah manage waktu.
13521027	Tubesnya seru tapi deadlinenya ga seru.

DAFTAR PUSTAKA

1. R. Munir, Bahan Kuliah IF2211 Strategi Algoritma: Pencocokan String. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf> [diakses pada 4 Mei 2023].
2. M. Ilham dan A.H. Mirza, Journal of Software Engineering Ampera: Penerapan Algoritma Knuth Morris Pratt Dalam Fitur Pencarian Pengarsipan Dokumen Pada SMA Plus Negeri 17 Palembang. Vol. 1, No. 2. Palembang: Bina Darma University, 2020.

LAMPIRAN

1. Tautan Repository
https://github.com/ilhamanullah/Tubes3_AnggerIntelligence
2. Tautan Video
<https://bit.ly/AnggerIntelligenceDemoVideo>
3. Foto kelompok yang bagus

