

# **Laporan Tugas Besar 3**

## **IF2211 Strategi Algoritma**

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



#### **Kelompok BKE**

Bintang Hijriawan Tjahja / 13521003

Eunice Sarah Siregar / 13521013

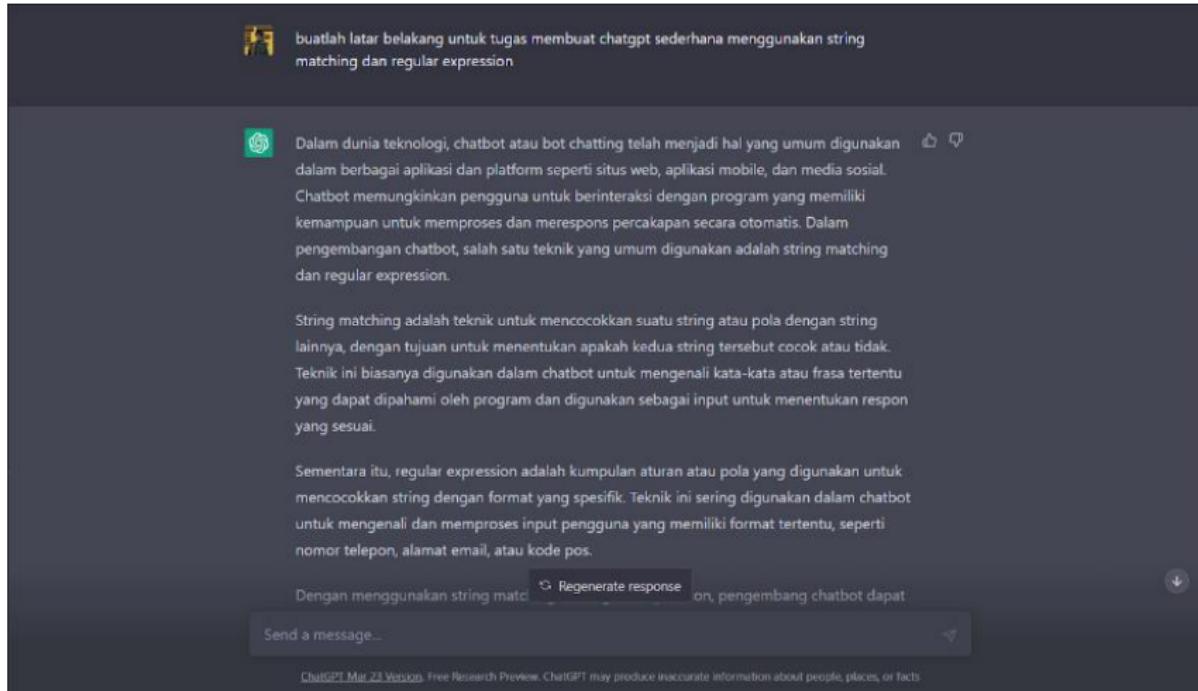
Kenny Benaya Nathan / 13521023

Program Studi Teknik Informatika  
**Sekolah Teknik Elektro dan Informatika**  
**Institut Teknologi Bandung**  
**Tahun 2022/2023**

## DAFTAR ISI

<b>DAFTAR ISI</b>	<b>2</b>
<b>BAB I</b>	<b>3</b>
<b>DESKRIPSI TUGAS</b>	<b>3</b>
<b>BAB II</b>	<b>5</b>
<b>LANDASAN TEORI</b>	<b>5</b>
2.1 Deskripsi Algoritma	5
2.1.1 Deskripsi Algoritma Knuth-Morris-Pratt	5
2.1.2 Deskripsi Algoritma Boyer-Moore	5
2.1.3 Deskripsi Algoritma Regular Expression	6
2.2 Penjelasan Singkat Aplikasi Web	6
<b>BAB III</b>	<b>7</b>
<b>ANALISIS PEMECAHAN MASALAH</b>	<b>7</b>
3.1 Langkah Pemecahan Masalah	7
3.2 Fitur Fungsional dan Arsitektur Aplikasi Website	7
<b>BAB IV</b>	<b>8</b>
<b>IMPLEMENTASI DAN PENGUJIAN</b>	<b>8</b>
4.1 Struktur Data dan Spesifikasi Program	8
4.1.1 Struktur Data	8
4.1.2 Fungsi dan Prosedur	8
4.2 Tata Cara Penggunaan Program	9
4.3 Hasil Pengujian	10
4.4 Analisis Desain Solusi	10
<b>BAB V</b>	<b>11</b>
<b>KESIMPULAN DAN SARAN</b>	<b>11</b>
5.1 Kesimpulan dan Saran	11
5.2 Refleksi dan Tanggapan	11
<b>DAFTAR PUSTAKA</b>	<b>12</b>
<b>LAMPIRAN</b>	<b>13</b>

## BAB I DESKRIPSI TUGAS



Gambar 1. Ilustrasi Chatbot ChatGPT

Dalam tugas besar ini, 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:

- Fitur pertanyaan teks (didapat dari database) Mencocokkan pertanyaan dari input pengguna ke pertanyaan di database menggunakan algoritma KMP atau BM.
- 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 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 Deskripsi Algoritma

##### 2.1.1 Deskripsi Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt atau bisa disingkat KMP adalah algoritma yang berfungsi untuk pencocokan string (string/ pattern matching). Berbeda dengan algoritma brute force, algoritma ini melakukan pemeliharaan informasi untuk menghitung pergeseran pattern yang diperlukan. Dalam algoritma KMP, terdapat prefix dan suffix, seperti contoh berikut:

Index	2	3	4	5	6	7
T	a	b	a	a	b	x

Index	0	1	2	3	4	5
P	a	b	a	a	b	a

Terjadi ketidak cocokan pada x ( $j = 5$ ). Lalu, dalam algoritma KMP tidak diperlukan pencocokan lagi dari awal, hanya saja menggeser T sehingga prefix 'ab' ada di sufix 'ab' sehingga  $j = 2$ . Dapat dihitung bahwa jumlah pergeseran yang dilakukan sebesar 3 diperoleh dari panjang(abbab) - panjang (ab).

Index		0	1	2
		a	b	a

Untuk mempercepat proses pencarian, dilakukan preprocessing untuk membentuk tabel berdasarkan aturan prefix  $T[0..j-1]$  = sufix  $T[0..j-1]$  yang dinamakan border function.

##### 2.1.2 Deskripsi Algoritma Boyer-Moore

Boyer-Moore merupakan algoritma string matching berdasarkan dua teknik, yaitu:

1. The looking-glass technique

Memeriksa kecocokan pattern P dengan teks T yang dimulai dari indeks terakhir pada pattern P, sedangkan pada teks T tetap dimulai dari awal.

2. The character-jump technique

Teknik ini dilakukan ketika menemukan ketidakcocokan. Kasus ini terjadi karena terjadi ketidakcocokan pada  $T[i]$  dan  $P[j]$ , sedangkan karakter pada  $T[i]$  adalah x. Terdapat beberapa macam kasus yang terjadi, seperti ketika terdapat x di P dengan indeks yang lebih besar dari j sehingga melakukan pergeseran pattern P satu karakter ke kanan. Selain itu, terjadi juga ketika keadaan sebaliknya dengan x di pattern P dengan indeks yang lebih kecil dari j sehingga dilakukan pergeseran ke

kanan juga. Terakhir, kasus ini terjadi ketidakcocokan pada  $T[i]$  dan  $P[j]$  dan karakter pada  $T[i]$  adalah x.

#### 2.1.3 Deskripsi Algoritma Regular Expression

Regular expression atau regex merupakan pola untuk mendeskripsikan sejumlah teks tertentu. Biasanya regex digunakan untuk memanipulasi, mengganti, mencari, atau memproses suatu string. Dalam tugas besar ini, regex digunakan untuk memvalidasi masukkan dari pengguna sebelum diproses oleh algoritma KMP atau BM.

## 2.2 Penjelasan Singkat Aplikasi Web

Untuk menyelesaikan tugas besar ini, dibagi menjadi dua bagian yaitu frontend dan backend. Pada frontend bertugas untuk tampilan kepada pengguna. Bagian frontend menggunakan bahasa pemrograman Typescript dengan menggunakan framework Tailwind CSS. Selain itu, terdapat bagian backend yang bertanggung jawab untuk memproses aktivitas pengguna dari bagian frontend. Bagian backend ditulis menggunakan bahasa pemrograman Node.js. Untuk penyimpanan database, disimpan menggunakan MySQL pada Prisma.

## **BAB III**

### **ANALISIS PEMECAHAN MASALAH**

#### **3.1 Langkah Pemecahan Masalah**

Dalam penerapan string matching dan regular expression, terdapat fitur-fitur seperti sebagai berikut.

- Aplikasi dapat menerima input berupa teks pertanyaan dari database. Input tersebut akan diproses menggunakan string matching dengan algoritma KMP atau algoritma BM.
- Aplikasi dapat menghitung aritmatika sederhana (kalkulator) seperti tambah, kurang, kali, bagi, pangkat, dan kurung. Untuk mengetahui operasi yang diminta, dilakukan menggunakan regular expression (regex).
- Aplikasi dapat mengeluarkan hari ketika diberikan tanggal yang valid.
- Pengguna dapat menambah pertanyaan dan jawaban ke database
- Pengguna dapat menghapus pertanyaan dari database

#### **3.2 Fitur Fungsional dan Arsitektur Aplikasi Website**

##### 1. Fitur fungsional frontend

Dalam frontend menggunakan bahasa pemrograman Typescript dengan framework Tailwind CSS. Berikut adalah bentuk frontend website

- New Chat  
Membuka percakapan baru dengan bot untuk memulai bertanya dan pecarian
- History  
Dapat melihat pertanyaan apa saja yang pernah ditanyakan sebelumnya dalam page chat berbeda
- Select Algorithm  
Pengguna dapat memilih algoritma apa yang akan digunakan untuk melakukan pencocokan string

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Struktur Data dan Spesifikasi Program

##### 4.1.1 Struktur Data

```
src
  └── app
      ├── api
      ├── assets
      │   └── fonts
      │       └── DaysOne-Regular.ttf
      ├── icon
      │   ├── bubble-bot.svg
      │   ├── bubble-history.svg
      │   └── bubble-user.svg
      ├── contexts
      │   ├── Chat.tsx
      │   ├── HistoryList.tsx
      │   └── Sidebar.tsx
      ├── styles
      │   └── globals.css
      ├── favicon.ico
      ├── layout.tsx
      └── page.tsx
  └── backend
      ├── bm.go
      ├── calculator.go
      ├── date.go
      ├── kmp.go
      ├── levenshteinDistance.go
      └── parser.go
```

##### 4.1.2 Fungsi dan Prosedur

- func `bmMatch(text string, pattern string) → int`, fungsi untuk pencocokan string dengan algoritma Boyer-Moore dengan menerima masukan pengguna dan database dengan keluaran boolean (0 jika tidak cocok dan 1 jika cocok)
- func `buildLast(pattern string) → map[byte]int`, fungsi untuk menentukan border function dari pattern yang sedang dilakukan pencarian dengan mengembalikan array berisi nilai dari masing-masing border function
- func `parseExpression(expr string) → (result float64, err error)`, fungsi yang menerima ekspresi matematika dalam bentuk string, dan mengembalikan hasil evaluasi dari ekspresi tersebut.
- func `getPrecedence(operator string) → int`, mengembalikan integer urutan operator yang harus dikerjakan duluan. \* dan / mengembalikan 2, + dan - mengembalikan 1

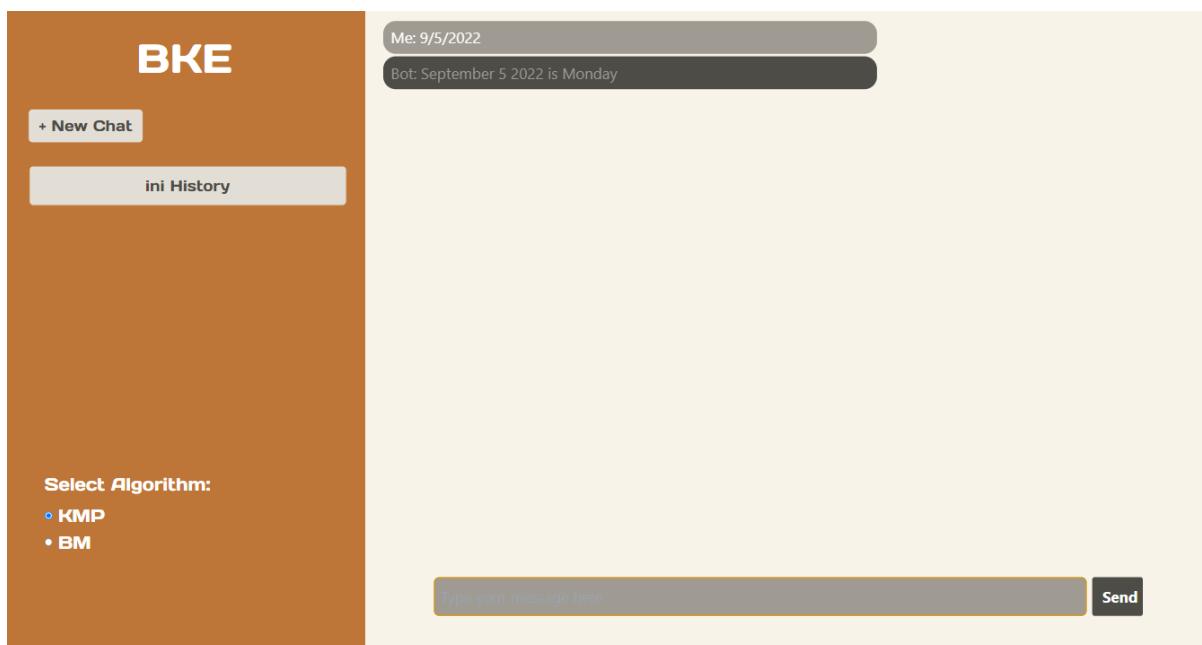
- func evaluateExpression(a float64, b float64, operator string) → float64, mengembalikan hasil evaluasi a operator b.
- func GetDayOfWeek(year int, month int, day int)--> map[string]interface{}, fungsi untuk mencari hari dari tanggal yang dimasukkan pengguna
- func kmpMatch(text string, pattern string) → int, fungsi untuk mencocokan string menggunakan algoritma Knuth-Morris-Pratt dengan menerima masukan pengguna dan database dengan keluaran boolean (-1 jika tidak cocok dan index jika cocok)
- func computeLPSArray(pattern string) → []int, fungsi untuk mengembalikan array yang berisi index terakhir dari kemunculan masing-masing huruf pada pattern
- func levenshteinDistance(str1, str2 string) → int, mengembalikan banyak perubahan yang harus dilakukan agar str1 bisa menjadi str2.
- func parser(str string) → string, menerima parameter pesan dari user, dan memanggil fungsi yang cocok untuk mendapatkan jawaban dari pesan user tersebut.
- function parseDate(str) → boolean, mengembalikan true jika parameter string bisa diparsing menjadi tanggal
- function isMathExpression(str) → boolean, mengembalikan true jika parameter string merupakan ekspresi matematika.
- func min(x, y int) → int, fungsi untuk menentukan nilai minimum

## 4.2 Tata Cara Penggunaan Program

Program ‘ChatBKE’ dapat dijalankan hanya pada server lokal. program ini akan berjalan pada localhost dengan port address 3000. Beberapa teknologi yang digunakan yang harus diinstall terlebih dahulu adalah node.js.

Program dapat digunakan dengan melakukan *clone* pada *repository* dengan tautan yang terlampir pada lampiran. Setelah melakukan *clone* ke dalam lokal, pengguna dapat mengetik ‘yarn’ untuk menginstall *dependencies* yang digunakan untuk projek ini pada terminal dengan *current working directory* pada root repository. Setelah menginstall dependencies, pengguna dapat mengetik ‘yarn dev’ untuk menjalankan program ini secara lokal );.

### 4.3 Hasil Pengujian



### 4.4 Analisis Desain Solusi

Berdasarkan hasil aplikasi web, dapat dianalisis sebagai berikut.

- Aplikasi dapat menerima pertanyaan berdasarkan masukan pengguna dan mengecek kebenarannya dengan algoritma KMP dan BM.
- Aplikasi dapat menghitung formula sebagai kalkulator
- Aplikasi dapat memberi tahu hari dari tanggal yang diberikan oleh pengguna

IF2211 Strategi Algoritma  
Penerapan String Matching dan Regular Expression  
dalam Pembuatan ChatGPT Sederhana  
Tahun 2022/2023

## **BAB V** **KESIMPULAN DAN SARAN**

### **5.1 Kesimpulan dan Saran**

Setelah menyelesaikan tugas besar ini, dapat disimpulkan bahwa untuk pencocokan string dapat dilakukan dengan berbagai cara seperti algoritma Knuth-Morris-Pratt dan algoritma Boyer-Moore. Lalu, setelah dicocokan dapat menghitung persentase kemiripan dengan algoritma Levenshtein. Pada tugas besar ini juga menambah pengetahuan mengenai frontend dan backend dengan bahasa pemrograman Next.js dan Go.

### **5.2 Refleksi dan Tanggapan**

Dapat ditarik refleksi berdasarkan tugas besar ini untuk tidak mengerjakan tugas mendekati deadline sehingga tidak terburu-buru. Selain itu dapat direfleksikan untuk tidak lalib dalam memilih bahasa pemrograman.

IF2211 Strategi Algoritma  
Penerapan String Matching dan Regular Expression  
dalam Pembuatan ChatGPT Sederhana  
Tahun 2022/2023

## **DAFTAR PUSTAKA**

- <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>  
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>

IF2211 Strategi Algoritma  
Penerapan String Matching dan Regular Expression  
dalam Pembuatan ChatGPT Sederhana  
Tahun 2022/2023

## LAMPIRAN

Link Repository : [https://github.com/kennypanjaitan/Tubes3\\_13521003](https://github.com/kennypanjaitan/Tubes3_13521003)