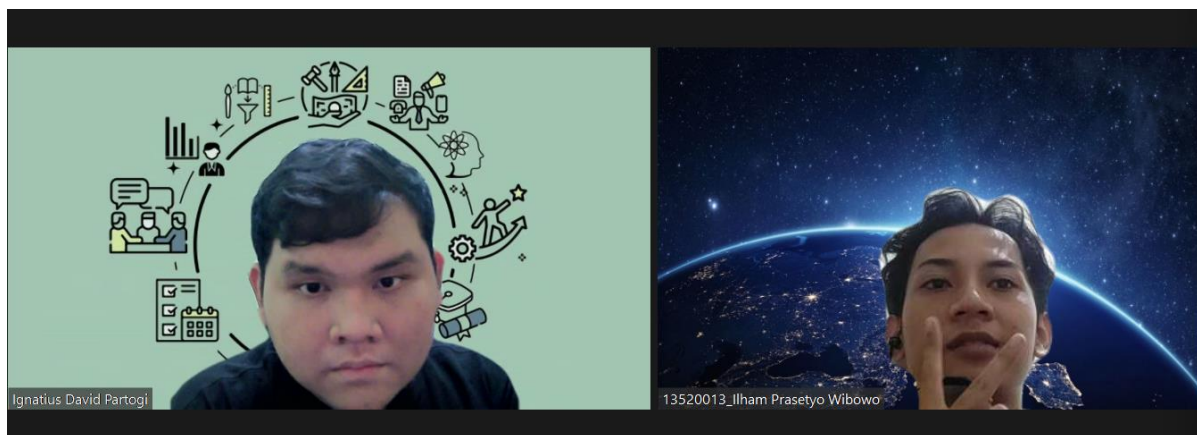


**LAPORAN TUGAS BESAR STRATEGI ALGORITMA**  
**“Penerapan String Matching dan Regular Expression dalam DNA Pattern Matching”**

**IF2211 STRATEGI ALGORITMA**



Kelompok : Mix Team 1

Disusun oleh :

Ignatius David Partogi (13518014)

Ilham Prasetyo Wibowo (13520013)

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**SETKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2021**

## **BAB I**

### **DESKRIPSI TUGAS**

Manusia umumnya memiliki 46 kromosom di dalam setiap selnya. Kromosom-kromosom tersebut tersusun dari DNA (deoxyribonucleic acid) atau asam deoksiribonukleat. DNA tersusun atas dua zat basa purin, yaitu Adenin (A) dan Guanin (G), serta dua zat basa pirimidin, yaitu sitosin (C) dan timin (T). Masing-masing purin akan berikatan dengan satu pirimidin. DNA merupakan materi genetik yang menentukan sifat dan karakteristik seseorang, seperti warna kulit, mata, rambut, dan bentuk wajah. Ketika seseorang memiliki kelainan genetik atau DNA, misalnya karena penyakit keturunan atau karena faktor lainnya, ia bisa mengalami penyakit tertentu. Oleh karena itu, tes DNA penting untuk dilakukan untuk mengetahui struktur genetik di dalam tubuh seseorang serta mendeteksi kelainan genetik. Ada berbagai jenis tes DNA yang dapat dilakukan, seperti uji pra implantasi, uji pra kelahiran, uji pembawa atau carrier testing, uji forensik, dan DNA sequence analysis.

Salah satu jenis tes DNA yang sangat berkaitan dengan dunia bioinformatika adalah DNA sequence analysis. DNA sequence analysis adalah sebuah cara yang dapat digunakan untuk memprediksi berbagai macam penyakit yang tersimpan pada database berdasarkan urutan sekuens DNA-nya. Sebuah sekuens DNA adalah suatu representasi string of nucleotides yang disimpan pada suatu rantai DNA, sebagai contoh: ATTCGTAAGTAAAGTTA. Teknik pattern matching memegang peranan penting untuk dapat menganalisis sekuens DNA yang sangat panjang dalam waktu singkat. Oleh karena itu, mahasiswa Teknik Informatika berniat untuk membuat suatu aplikasi web berupa DNA Sequence Matching yang menerapkan algoritma String Matching dan Regular Expression untuk membantu penyedia jasa kesehatan dalam memprediksi penyakit pasien. Hasil prediksi juga dapat ditampilkan dalam tabel dan dilengkapi dengan kolom pencarian untuk membantu admin dalam melakukan filtering dan pencarian.

Dalam tugas besar ini, anda diminta untuk membangun sebuah aplikasi DNA Pattern Matching. Dengan memanfaatkan algoritma String Matching dan Regular Expression yang telah anda pelajari di kelas IF2211 Strategi Algoritma, anda diharapkan dapat membangun sebuah aplikasi interaktif untuk mendeteksi apakah seorang pasien mempunyai penyakit genetik tertentu. Hasil prediksi tersebut dapat disimpan pada basis data untuk kemudian dapat ditampilkan berdasarkan query pencarian.

## **BAB II**

### **LANDASAN TEORI**

#### **1. Algoritma String Matching**

Algoritma string matching/pattern matching adalah algoritma yang digunakan untuk mencocokkan suatu pola dengan teks. Atau bisa juga mengecek kemunculan sebuah kata pada teks. Algoritma pencocokan string dapat diimplementasikan dengan strategi brute-force. Strategi brute-force mengecek string tiap huruf, jika huruf itu tidak sama, dilakukan pergeseran satu kali ke kanan. Terdapat algoritma yang lebih efisien dalam pencocokan string, yaitu algoritma Knuth-Morris-Pratt (KMP) dan algoritma Boyer-Moore (BM).

##### **a. Algoritma Knuth-Morris-Pratt**

Ide dari pencocokan string algoritma kmp mirip dengan brute-force. Namun, dilakukan modifikasi pergeseran untuk mempercepat pencarian. Setiap algoritma mendeteksi ketidakcocokan, algoritma menggeser beberapa huruf kekanan dengan informasi yang dimiliki (ketidakcocokan). KMP menggeser sebanyak prefix terbesar  $P[0..j-1]$  yang juga suffix dari  $P[1..j-1]$

Algoritma memiliki fungsi pinggiran untuk preproses pola yang akan dicari. Fungsi pinggiran mencari kesamaan prefix dengan pola itu sendiri. KMP memiliki kelebihan tidak pernah untuk melakukan runut balik.

##### **b. Algoritma Boyer-Moore**

Algoritma Boyer-Moore memiliki dua teknik, yaitu teknik looking-glass, dan teknik character-jump. Looking-glass mencari pola di teks dari belakang. Teknik character-jump adalah ketika terdapat ketidakcocokan pada  $T[i] = x$ , huruf pada  $P[j]$  tidak sama dengan  $T[i]$ . Kelebihan anggota Boyer-Moore adalah algoritma ini efisien untuk teks yang panjang.

#### **2. Pembuatan Aplikasi Website**

Banyak cara untuk membangun sebuah aplikasi web. Namun pada tugas kali ini penulis menggunakan framework frontend React.js, backend Node.js, dengan menggunakan basisdata MySQL.

##### **1. Node.js**

Node.js adalah sebuah runtime environment yang bersifat open-source. Node.js menggunakan bahasa JavaScript. Node.js menyediakan banyak pustaka.

Keunggulan Node.js

##### **a. Proses Cepat**

Node.js berjalan pada V8 JavaScript Engine, sehingga sangat cepat untuk mengeksekusi kode.

##### **b. Pustaka Lengkap**

Pustaka Node.js sangat lengkap yang akan memudahkan pengembang dalam mengembangkan aplikasi website.

##### **c. Open-source**

Siapa saja bisa melihat source-code dari pustaka Node.js

Arsitektur Node.js terdiri dari Request, server, event queue, thread pool, dan event loop. Server dari Node.js adalah platform yang menerima request dari user dan

memprosesnya. Event queue menyimpan request yang masuk, dan memprosesnya satu per satu.

## 2. React.js

React.js adalah suatu library JavaScript yang digunakan untuk membangun web front-end (client side). Keunggulan React.js adalah komponen-komponen dari React.js sudah tertulis dan pengembang hanya perlu menggunakannya. Komponen bisa digunakan berkali kali membuat kode reusable.

Fitur-fitur React.js :

- a. Deklaratif
- b. Berbasis komponen
- c. Kode reusable

## 3. MySQL

MySQL adalah salah satu jenis basis data yang bersifat open-source. MySQL digunakan untuk menyimpan data yang digunakan dalam fungsionalitas perangkat lunak. MySQL menggunakan perintah SQL (Structured Query Language).

Kelebihan MySQL :

- a. Open-source
- b. Multi-user
- c. Bisa digunakan dalam berbagai bahasa pemrograman
- d. Tipe data variatif
- e. Struktur tabel yang fleksibel
- f. Sistem keamanan yang terjamin
- g. Memiliki performa yang baik.

### **BAB III**

#### **ANALISIS PEMECAHAN MASALAH**

##### **1. Pembuatan Basis Data**

Fitur yang pertama dalam spesifikasi adalah menambah penyakit baru dan sequence DNAny. Sebelum implementasi pada website, diperlukan sarana untuk menyimpan dan akses data. Sehingga dibuat basis data menggunakan MySQL. Dari kebutuhan website, informasi yang wajib disimpan adalah jenis penyakit dan hasil prediksi. Sehingga akan dibuat basis data dengan dua tabel yaitu tabel yang menyimpan jenis penyakit, dengan atribut nama penyakit dan rantai DNA. Tabel yang kedua adalah hasil prediksi dengan atribut tanggal prediksi, nama pasien, penyakit, dan status terprediksi.

##### **2. Fitur menerima input penyakit baru dan rantai DNA.**

Input nama penyakit akan diimplementasikan di sisi front-end. Sedangkan input rantai DNA perlu dibuat file handler untuk membaca isi dari file. Sanitasi input akan dilakukan di sisi backend, dengan membuat fungsi regex. Jika input dari pengguna adalah valid, maka data akan dimasukkan dalam basis data. Perlu dibuat fungsi untuk mengunggah ke basis data.

##### **3. Fitur memprediksi seseorang menderita penyakit tertentu berdasarkan rantai DNA-nya.**

Untuk bagian front-end, hampir sama seperti fitur sebelumnya yaitu memerlukan fungsi untuk membaca isi file, dan mengirim data masukan pengguna ke server. Input juga akan disanitasi menggunakan regex, untuk memastikan tidak ada masukan yang salah.

Diperlukan sebuah fungsi untuk mengecek kecocokan rantai DNA penyakit dan orang yang dites. Sehingga akan diimplementasikan algoritma pattern matching KMP dan BM. Implementasi algoritma pattern matching akan dilakukan di sisi server.

Akan dilakukan pencocokan antara rantai DNA pengguna dengan seluruh rantai DNA yang ada pada basis data. Jika ditemukan kecocokan dan penyakit sama dengan prediksi pengguna, maka status prediksi akan menjadi true, sebaliknya akan menjadi false. Setelah itu, data tes akan di masukkan ke dalam basis data.

##### **4. Pencarian riwayat tes.**

Akan dibuat fungsi untuk mengecek masukan dari pengguna. Kemudian akan dilakukan pengecekan apakah masukan hanya tanggal saja, hanya penyakit saja, atau dua-duanya. Kemudian, sesuai dengan luaran dari fungsi, akan dilakukan pencarian pada basis data. Hasil dari pencarian tersebut akan dikirim ke sisi front-end, dan ditampilkan dengan sebuah tabel.

##### **5. Pembuatan web**

Pembuatan website akan memanfaatkan Node.js sebagai backend (server-side), React.js sebagai front-end (client-side), dan MySQL sebagai basis data untuk menyimpan data penyakit dan hasil tes. Implementasi server juga menggunakan Express.js untuk mempermudah pembuatan website.

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **A. Spesifikasi Teknis Program**

Program akan dibagi menjadi dua bagian yaitu backend dan frontend. Backend dan frontend dipisahkan dalam dua folder berbeda.

##### **1. Backend**

Terdapat beberapa folder dalam backend, yaitu config, dan script.

###### **a. Config**

Folder config menyimpan kode untuk konfigurasi koneksi antara program dengan basis data. Koneksi program dengan basis data menggunakan modul mysql.

###### **b. Script**

Dalam script disimpan algoritma pencocokan string KMP, BM, serta algoritma regex.

File utama dalam backend adalah index.js yang terdiri dari beberapa fungsi.

###### **a. /fetch/sequence**

Digunakan untuk mendapatkan seluruh data penyakit dari basis data

###### **b. /fetch/test**

Untuk mendapatkan seluruh data hasil tes dari basis data.

###### **c. /search**

Digunakan untuk mencari data di database. Pencarian bergantung dari masukan user, jika masukan hanya berupa tanggal, akan dicari data tes yang memiliki tanggal yang sama, sama halnya dengan tanggal dan penyakit.

###### **d. /penyakit/add**

Digunakan untuk menambah data penyakit.

###### **e. /hasil/add**

Untuk menambah data hasil tes pada basis data. Dalam fungsi ini diimplementasikan fungsi pencocokan string. Pencocokan string dicari di seluruh rekaman pada basis data. Jika ditemukan, kecocokan string dan prediksi sama, maka akan dimasukkan pada basis data dengan prediksi bernilai true, sebaliknya bernilai false.

###### **f. /penyakit/delete**

Menghapus rekaman dari basis data.

##### **2. Frontend**

Dalam folder frontend, terdapat folder Pages yang berisi halaman dari website.

Halaman website terdiri dari tiga halaman yaitu :

###### **a. DiseasePage**

DiseasePage menyimpan halaman penambahan data penyakit. Dalam disease page terdapat form yang menerima nama penyakit dan file rantai DNA. Halaman dihubungkan dengan server menggunakan fungsi fetch() dengan url “/penyakit/add”. Fungsi mengirim json file yang berisi nama penyakit dan sequence DNA.

###### **b. DNAPage**

Hampir sama seperti DiseasePage, DNAPage terdapat form yang menerima nama pengguna, file rantai DNA, dan prediksi penyakit. Ketiga masukan tersebut akan dikirim ke server dengan url “/hasil/add”. Pengecekan masukan

dan pencocokan string terdapat pada server. Hasil akan ditampilkan setelah pencarian dilakukan.

c. HistoryPage

Dalam halaman HistoryPage terdapat form untuk menerima kunci pencarian. Kunci memiliki format Tanggal Penyakit. Dengan format tanggal adalah YYYY-MM-DD. Pencarian bisa hanya menerima tanggal, atau hanya menerima penyakit saja. Hasil akan diperlihatkan sebagai tabel pada halaman.

Website juga memiliki navbar di bagian atas. Bar navigasi terdapat 3 link untuk menuju ke halaman DNAPage, HistoryPage, atau DiseasePage. Nilai dari form yang ada pada ketiga halaman dipindahkan kedalam sebuah variabel menggunakan fungsi useState() dari modul react.

3. Basis Data

Implementasi basis data menggunakan MySQL dan terdapat dua tabel, yaitu hasil, dan sequence. Basis data di-host di db4free.net

- Tabel hasil

```
MariaDB [penyakit]> describe hasil;
```

Field	Type	Null	Key	Default	Extra
ID_Hasil	int(11)	NO	PRI	NULL	auto_increment
Tanggal	date	YES		NULL	
Nama	varchar(255)	YES		NULL	
Nama_Penyakit	varchar(255)	YES	MUL	NULL	
Prediksi	varchar(20)	YES		NULL	

5 rows in set (0.105 sec)

- Tabel sequence

```
MariaDB [penyakit]> describe sequence;
```

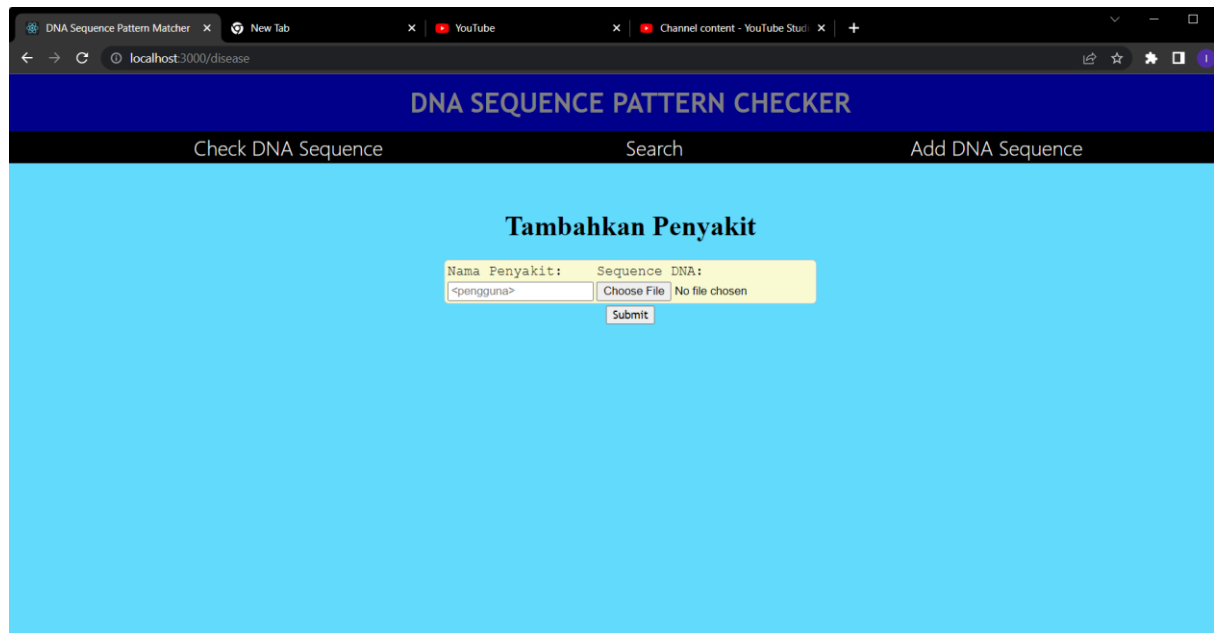
Field	Type	Null	Key	Default	Extra
ID_Penyakit	int(11)	NO	PRI	NULL	auto_increment
Nama_Penyakit	varchar(255)	NO		NULL	
DNASequences	text	YES		NULL	

3 rows in set (0.031 sec)

B. Tata cara penggunaan program.

Terdapat 3 halaman :

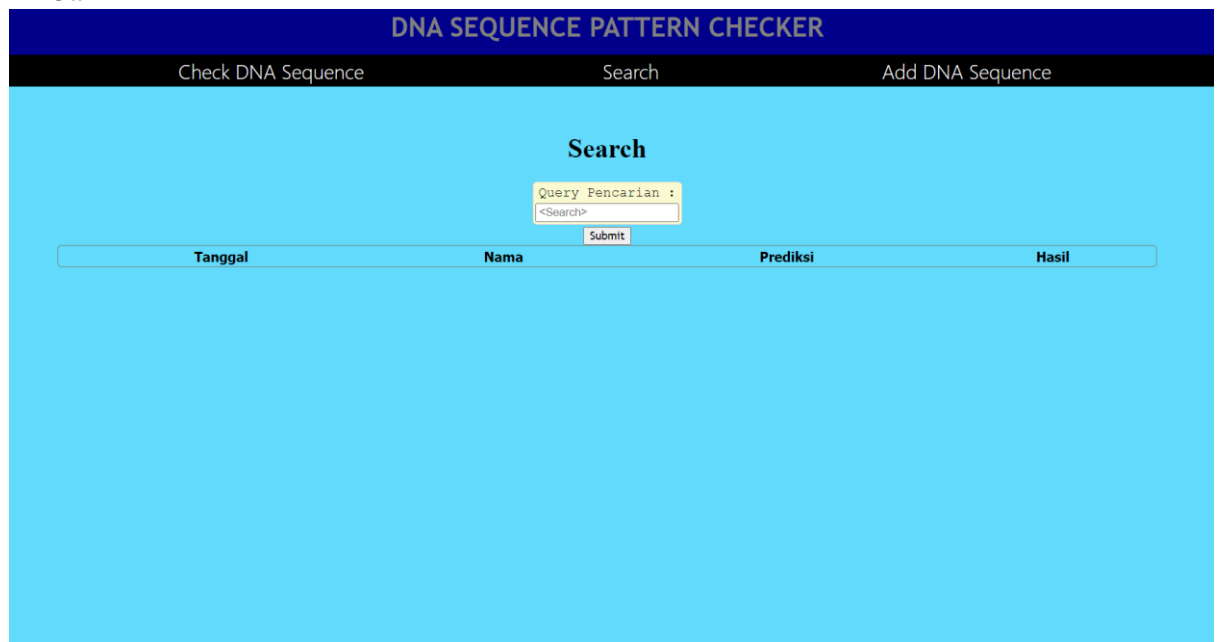
- Tambah data penyakit



Penggunaan dengan memasukkan nama penyakit, dan file sequence dnanya.

Kemudian submit

➤ Cari



Penggunaan dengan memasukkan key pencarian

➤ Tes DNA



Penggunaan dengan memasukkan nama pengguna, rantai dna dan prediksi penyakit. Kemudian klik tombol submit.

### C. Hasil Pengujian

#### ❖ Tambah data penyakit

#### ❖ Search tanggal

DNA SEQUENCE PATTERN CHECKER

Check DNA Sequence
Search
Add DNA Sequence

### Search

Query Pencarian :

2022-04-29

Submit

Tanggal	Nama	Prediksi	Hasil
2022-04-29	John	HIV	False
2022-04-29	Smith	Cancer	False
2022-04-29	Wong	Cancer	False
2022-04-29	Roshi	Heart Attack	False
2022-04-29	Shen	Heart Attack	False
2022-04-29	Shen	Heart Attack	True
2022-04-29	Roshi	Heart Attack	True
2022-04-29	Roshi	HIV	False

❖ Search penyakit

DNA SEQUENCE PATTERN CHECKER

Check DNA Sequence
Search
Add DNA Sequence

### Search

Query Pencarian :

Heart Attack

Submit

Tanggal	Nama	Prediksi	Hasil
2022-04-29	Roshi	Heart Attack	False
2022-04-29	Shen	Heart Attack	False
2022-04-29	Shen	Heart Attack	True
2022-04-29	Roshi	Heart Attack	True

❖ Search tanggal + penyakit

DNA SEQUENCE PATTERN CHECKER

Check DNA Sequence
Search
Add DNA Sequence

### Search

Query Pencarian :

2022-04-29 Cancer

Submit

Tanggal	Nama	Prediksi	Hasil
2022-04-29	Smith	Cancer	False
2022-04-29	Wong	Cancer	False

❖ Tambah tes jika benar

The screenshot shows the 'DNA SEQUENCE PATTERN CHECKER' web application. The header has three navigation links: 'Check DNA Sequence', 'Search', and 'Add DNA Sequence'. The main content area is titled 'Tes DNA' and contains a form with three input fields: 'Nama Pengguna:' (containing 'Person'), 'Sequence DNA:' (containing 'person3.txt'), and 'Prediksi Penyakit:' (containing 'Heart Attack'). A 'Submit' button is located below the form. The result displayed is 'Result : 2022-4-29 - Person - Heart Attack - True'.

❖ **Tambah hasil tes jika salah**

The screenshot shows the 'DNA SEQUENCE PATTERN CHECKER' web application. The header has three navigation links: 'Check DNA Sequence', 'Search', and 'Add DNA Sequence'. The main content area is titled 'Tes DNA' and contains a form with three input fields: 'Nama Pengguna:' (containing 'Person'), 'Sequence DNA:' (containing 'person3.txt'), and 'Prediksi Penyakit:' (containing 'Cancer'). A 'Submit' button is located below the form. The result displayed is 'Result : 2022-4-29 - Person - Cancer - False'.

**D. Analisis Hasil Implementasi**

Hasil pengujian cukup memuaskan untuk setiap fitur. Namun, belum bisa memilih algoritma yang akan digunakan dari website untuk pencocokan string. Untuk memilih algoritma pencocokan harus mengubah kode terlebih dahulu (mengubah pemanggilan fungsi saja, kedua algoritma sudah diimplementasi dan bisa dijalankan). Selain itu, user interface masih perlu pengembangan lagi. Juga belum bisa mengukur tingkat kemiripan dari DNA. Untuk fungsi fungsi seperti navigation bar semuanya berjalan dengan lancar. Hasil yang dikeluarkan oleh tiap fitur juga sesuai ekspektasi. Fungsi search juga sudah bisa menerima tiga bentuk pencarian.

## **BAB V**

### **KESIMPULAN DAN SARAN**

Algoritma pencocokan string bisa digunakan dalam berbagai hal. Dalam kasus ini, algoritma pencocokan string digunakan untuk mencocokkan rantai DNA. Terdapat dua algoritma pencocokan string yang lebih efisien daripada menggunakan algoritma brute-force yaitu algoritma Knuth-Morris-Pratt dan algoritma Boyer-Moore. Adapun program masih banyak kekurangan, seperti masih belum mengimplementasikan persentase kecocokan. Website juga belum di-deploy, masih berjalan di lokal.

## DAFTAR PUSTAKA

<https://nodejs.org/en/docs/>

<https://reactjs.org/docs/>

<https://dev.mysql.com/doc/>

<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>

## LAMPIRAN

Tautan menuju source code :

<https://github.com/ilhamwibowo/TUBES-STIMA-3>

Video demonstrasi program:

<https://youtu.be/GlqdS35s6TU>

Pembagian tugas :

Backend, database, frontend functionality : 13520013

Frontend styling : 13518014