

LAPORAN TUGAS BESAR 3

Penerapan String Matching dan Regular Expression dalam DNA Pattern Matching

Laporan dibuat untuk memenuhi salah satu tugas mata kuliah
IF2211 Strategi Algoritma



Disusun oleh:

Monica Adelia	13520096
Jundan Haris	13520155
Hilda Carissa	13520164

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2022

DAFTAR ISI

DAFTAR ISI	1
Bab I : Deskripsi Tugas	2
Bab II : Landasan Teori	8
Algoritma KMP	8
Algoritma BM	8
Regex	8
Penjelasan Singkat Mengenai search.dna	9
Bab III : Analisis Pemecahan Masalah	10
Langkah Penyelesaian Masalah Setiap Fitur	10
Fitur Fungsional dan Arsitektur Web	10
Bab IV : Implementasi dan Pengujian	12
Spesifikasi Teknis Program	12
Tata Cara Penggunaan Program	14
Hasil Pengujian	16
Analisis Hasil Pengujian	25
Bab V : Kesimpulan, Saran, dan Komentar / Refleksi	26
Kesimpulan	26
Saran	26
Komentar / Refleksi	26
Link Repository dan Video Demo	26
Daftar Pustaka	27

Bab I : Deskripsi Tugas

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.

Fitur-Fitur Aplikasi:

1. Aplikasi dapat menerima input penyakit baru berupa nama penyakit dan sequence DNA-nya (dan dimasukkan ke dalam database).
 - a. Implementasi input sequence DNA dalam bentuk file.
 - b. Dilakukan sanitasi input menggunakan regex untuk memastikan bahwa masukan merupakan sequence DNA yang valid (tidak boleh ada huruf kecil, tidak boleh ada huruf selain AGCT, dan tidak ada spasi).
 - c. Contoh input penyakit:

The diagram shows a rounded rectangular frame containing a form titled "Tambahkan Penyakit". Inside the frame, there are two input fields. The first field is labeled "Nama Penyakit:" and contains the placeholder text "penyakit...". The second field is labeled "Sequence DNA:" and contains the placeholder text "upload file...". Below these fields is a green rectangular button with the word "Submit" in white text.

2. Aplikasi dapat memprediksi seseorang menderita penyakit tertentu berdasarkan sequence DNA-nya.
 - a. Tes DNA dilakukan dengan menerima input nama pengguna, sequence DNA pengguna, dan nama penyakit yang diuji. Asumsi sequence DNA pengguna > sequence DNA penyakit.

- b. Dilakukan sanitasi input menggunakan regex untuk memastikan bahwa masukan merupakan sequence DNA yang valid (tidak boleh ada huruf kecil, tidak boleh ada huruf selain AGCT, tidak ada spasi, dll).
- c. Pencocokan sequence DNA dilakukan dengan menggunakan algoritma string matching.
- d. Hasil dari tes DNA berupa tanggal tes, nama pengguna, nama penyakit yang diuji, dan status hasil tes. Contoh: 1 April 2022 - Mhs IF - HIV - False
- e. Semua komponen hasil tes ini dapat ditampilkan pada halaman web (refer ke poin 3 pada “Fitur-Fitur Aplikasi”) dan disimpan pada sebuah tabel database.
- f. Contoh tampilan web:

Tes DNA

Nama Pengguna:
<pengguna>

Sequence DNA:
upload file...

Prediksi Penyakit:
<penyakit>

Submit

Hasil Tes

<Tanggal> - <pengguna> - <penyakit> - <True/False>

- 3. Aplikasi memiliki halaman yang menampilkan urutan hasil prediksi dengan kolom pencarian di dalamnya. Kolom pencarian bekerja sebagai filter dalam menampilkan hasil.
 - a. Kolom pencarian dapat menerima masukan dengan struktur: , contoh “13 April 2022 HIV”. Format penanggalan dibebaskan, jika bisa menerima >1 format lebih baik.
 - b. Kolom pencarian dapat menerima masukan hanya tanggal ataupun hanya nama penyakit. Fitur ini diimplementasikan menggunakan regex.

c. Contoh ilustrasi:

i. Masukan tanggal dan nama penyakit

13 April 2022 HIV
1. 13 April 2022 - Fulan - HIV - True.
2. 13 April 2022 - Kamal - HIV - False.
3. 13 April 2022 - Entah - HIV - False.
4. 13 April 2022 - Jamal - HIV - True.
5. 13 April 2022 - Yubai - HIV - True.
6. 13 April 2022 - Hika - HIV - False.

ii. Masukan hanya tanggal

13 April 2022
1. 13 April 2022 - Fulan - Diabetes - True.
2. 13 April 2022 - Kamal - Sinusitis - False.
3. 13 April 2022 - Entah - Down Syndrome - False.
4. 13 April 2022 - Jamal - Polio - True.
5. 13 April 2022 - Yubai - TBC - True.
6. 13 April 2022 - Hika - Hepatitis A - False.

iii. Masukan hanya nama penyakit

HIV
1. 13 April 2022 - Fulan - HIV - True.
2. 14 April 2022 - Kamal - HIV - False.
3. 15 April 2022 - Entah - HIV - False.
4. 16 April 2022 - Jamal - HIV - True.
5. 17 April 2022 - Yubai - HIV - True.
6. 18 April 2022 - Hika - HIV - False.

4. (Bonus) Menghitung tingkat kemiripan DNA pengguna dengan DNA penyakit pada tes DNA
- Ketika melakukan tes DNA, terdapat persentase kemiripan DNA dalam hasil tes. Contoh hasil tes: 1 April 2022 - Mhs IF - HIV - 75% - False
 - Perhitungan tingkat kemiripan dapat dilakukan dengan menggunakan Hamming distance, Levenshtein distance, LCS, atau algoritma lainnya (dapat dijelaskan dalam laporan).
 - Tingkat kemiripan DNA dengan nilai lebih dari atau sama dengan 80% dikategorikan sebagai True. Perlu diperhatikan mengimplementasikan atau tidak mengimplementasikan bonus ini tetap dilakukan pengecekan string matching terlebih dahulu.
 - Contoh tampilan:

Tes DNA

Nama Pengguna:

<pengguna>

Sequence DNA:

upload file...

Prediksi Penyakit:

<penyakit>

Submit

Hasil Tes

<Tanggal> - <pengguna> - <penyakit> - <similarity> - <True/False>

Bab II : Landasan Teori

1. Algoritma KMP

Algoritma KMP (Knuth-Morris-Pratt) adalah algoritma pencarian yang mencocokkan sebuah teks dimulai dari kiri ke kanan, mirip seperti algoritma Brute Force. Perbedaan utamanya adalah, algoritma KMP akan mencocokkan pola dengan lebih ‘pintar’ dibandingkan dengan algoritma brute force. Pada brute force, setiap kali perbandingan yang sedang dilakukan salah, maka akan menggeser sebanyak 1 karakter lalu dibandingkan dari awal lagi. Namun, pada algoritma KMP, perbandingan yang sebelumnya sudah dilakukan “disimpan” sehingga ketika ditemukan yang salah, perbandingan teks dengan pola tidak dilakukan dari awal lagi, melainkan langsung ke posisi dimana string akhir text (yang sudah dicek) sesuai dengan string awal pattern.

2. Algoritma BM

Sama seperti algoritma KMP, algoritma Boyer-Moore (BM) adalah algoritma pencocokan string juga. Algoritma ini berdasarkan dua buah teknik. Pertama teknik *the looking-glass* yaitu pencarian pola di sebuah teks dengan mengecek setiap karakter di pola dari bagian akhir P ke awal. Selanjutnya ada teknik *character jump* dimana jika terdapat ketidakcocokan pada karakter $T[i]$ di teks T, dilakukan perubahan nilai pada i agar seolah menggeser pola yang sedang dicocokkan di teks T.

Terdapat 3 kasus pada pencocokan algoritma BM ini. Pertama jika dalam pola P terdapat x yang indeksnya lebih kecil dari j, maka P akan digeser ke sebelah kanan hingga posisi last occurrence x di P dan T sejajar. Kasus kedua adalah ketika di P terdapat x yang indeksnya lebih besar dari j, maka P di geser ke kanan sebanyak 1 sehingga posisinya berada di $T[i+1]$. Kasus terakhir adalah ketika tidak terdapat x di P maka P digeser hingga $P[0]$ dan $T[i+1]$ sejajar.

3. Regex

Regular Expression adalah sebuah notasi yang biasanya digunakan untuk mendeskripsikan pola dari sebuah kata atau sekumpulan string yang ingin dicari. Regex sendiri memiliki format yang sudah didefinisikan. Setiap karakternya dapat memiliki makna spesial atau literal. Contohnya untuk karakter ‘a’ maka bermakna huruf ‘a’ kecil secara literal. Namun, untuk penggunaan “.” maka artinya adalah semua karakter kecuali karakter newline. Bisa juga seperti penggunaan “?” yang berarti opsional, misalnya untuk kata <<colou?r>> berarti huruf u dalam kata itu bersifat opsional sehingga untuk regex tersebut, kata color dan colour akan diterima. Hal ini akan mempermudah dalam membuat validasi sehingga tidak perlu menambahkan semua pattern yang mungkin melainkan tinggal menggunakan regex saja.

4. Penjelasan Singkat Mengenai search.dna

Search.dna adalah sebuah aplikasi berbasis web yang memiliki beberapa fitur utama. Pertama ada fitur untuk mengetes DNA dari pengguna. Fitur ini akan meminta nama pengguna, sequence DNA pengguna, dan prediksi penyakit serta pilihan penggunaan algoritma. Ketika pengguna menekan submit, aplikasi akan mencocokkan sequence DNA dari prediksi penyakit dengan sequence DNA pengguna. Jika ditemukan sequence DNA penyakit pada DNA pengguna, maka pengguna memiliki penyakit tersebut dan akan dikeluarkan pesan berupa tanggal - nama pengguna - penyakit - status kecocokan.

Kedua, terdapat fitur riwayat prediksi. Dalam fitur ini, pengguna dapat memasukkan query berupa tanggal, nama penyakit atau tanggal dan nama penyakit. Ketika pengguna menekan search, maka akan muncul hasil-hasil prediksi yang sudah dilakukan yang sesuai dengan query yang dimasukkan pengguna.

Terakhir terdapat fitur tambah penyakit. Fitur ini akan meminta input dari pengguna berupa nama penyakit dan sequence DNA-nya. Ketika DNA yang dimasukkan valid, maka nama dan sequence DNA-nya akan dimasukkan ke dalam database dan akan muncul pesan berhasil.

Bab III : Analisis Pemecahan Masalah

1) Langkah Penyelesaian Masalah Setiap Fitur

a) Fitur Tes DNA

- Pertama, dilakukan permintaan input dari user
- Dilakukan sanitasi sequence DNA dari user
- Jika sequence DNA dari user valid, dilakukan pencarian prediksi penyakit pada database
- Jika terdapat penyakit yang sesuai dengan yang di-input user, akan diambil rantai DNA dari penyakit yang bersangkutan dan dilakukan string matching dengan algoritma sesuai permintaan user
- Setelah didapat hasilnya, data yang sudah terolah ini akan dimasukkan ke database dan ditampilkan di layar

b) Fitur Riwayat Prediksi

- User menginput query berupa tanggal, nama penyakit, atau tanggal dan nama penyakit
- Jika query user benar dan terdapat data yang diinginkan di dalam database maka data tersebut akan ditampilkan
- Jika query user salah atau tidak terdapat dalam database maka tidak akan muncul data apapun
-

c) Fitur Tambah Penyakit

- Meminta input user berupa nama penyakit dan file yang berisi sequence DNA dari penyakit terkait
- Sequence DNA akan divalidasi dengan menggunakan Regex
- Apabila sequence DNA terdiri dari huruf A, G, C, T, kapital semua, tanpa spasi, maka dianggap valid dan penyakit akan masuk ke dalam database
- Jika sequence DNA tidak valid, akan muncul pesan bahwa sequence DNA tidak valid

2) Fitur Fungsional dan Arsitektur Web

a) Fitur Fungsional

- i) Melakukan Tes DNA user terhadap prediksi penyakit (yang ada di database)
- ii) Menampilkan riwayat Tes DNA yang sudah dilakukan berdasarkan query dari pengguna
- iii) Menambahkan penyaki dan sequence DNA ke dalam database

b) Arsitektur Web search.dna dibagi menjadi dua bagian, yaitu:

- i) Frontend

Implementasi *frontend* dari web search.dna dibuat dengan menggunakan library ReactJS. Web dibangun sebagai *multi page app* dengan code utama berada pada App.js dan *page-page* lainnya berada pada folder pages. Setiap page, diatur ukuran, warna, letaknya oleh file css. Pages pada web search.dna berjumlah empat, yaitu home page, page untuk fitur riwayat prediksi, page untuk fitur tambah penyakit, dan page untuk test DNA. Pemanggilan algoritma string matching dilakukan pada masing-masing page dengan memanfaatkan *library* axios. Hasil dari string matching akan ditampilkan pada halaman yang sama sesuai fitur web yang sedang digunakan.

ii) Backend

Backend dari web search.dna dibuat dengan Node.js. Mendukung fitur penyimpanan dari web yang kami buat, kami menggunakan *cloud* yang disediakan MongoDB yaitu Atlas. Pada folder backend terdapat folder model yang berisi struktur data untuk masing-masing tabel basisdata. Terdapat juga file algo.js yang berisi fungsi-fungsi yang digunakan untuk validasi input dan algoritma string matching sendiri. Ada juga file app.js yang berisi query-query untuk menghubungkan backend dengan fungsi yang ada di algo.js

Bab IV : Implementasi dan Pengujian

1. Spesifikasi Teknis Program

a. Struktur Data

Struktur data utama yang digunakan untuk pengiriman dan pengambilan data dari ke basis data adalah dengan menggunakan JSON (*Javascript Object Notation*).

Berikut struktur data JSON yang digunakan oleh kelompok kami. Pertama adalah struktur data untuk hasilPrediksi, contohnya seperti ini

```
Let hasilPrediksi = {  
    tanggalPrediksi : "29/04/2022",  
    namaPasien : "Monica",  
    penyakitPrediksi : "encok",  
    statusTerprediksi : false  
};
```

Selain hasilPrediksi, ada juga struktur data jenisPenyakit. Berikut contohnya

```
Let jenisPenyakit = {  
    namaPenyakit = "encok",  
    rantaiDNA = "CTGGACGACTAGGGAAA"  
};
```

b. Fungsi dan Prosedur

No	Fungsi/Prosedur (Parameter)	Keterangan
1.	pisahinQuery(query)	Fungsi ini dipakai untuk memisahkan query dari user untuk pencarian riwayat. Fungsi akan mengambilkan sebuah array yang terdiri dari dua elemen. Elemen pertama berupa tanggal dengan format DD/MM/YYYY jika user memasukkan query tanggal, dan elemen kedua berupa nama penyakit. Jika user hanya memasukkan tanggal maka

		elemen kedua di array kembalian akan berupa string kosong. Sedangkan jika user hanya menuliskan nama penyakit, maka elemen pertama dari array kembalian berisi integer -1.
2.	psiahTanggal(Tanggal)	Fungsi ini digunakan untuk mengubah format tanggal dari query yang dimasukkan oleh user menjadi tanggal berformat DD/MM/YYYY. Fungsi ini menggunakan regex untuk mengenali tanggal, bulan, dan tahun. Kembalian dari fungsi ini adalah string dengan format tanggal seperti yang sudah disebutkan.
3.	isDNAValid(dna)	Fungsi ini digunakan untuk mengecek apakah sebuah rantai DNA valid atau tidak. Rantai DNA akan disebut valid jika terdiri dari huruf A, G, C, T yang semuanya kapital tanpa ada spasi. Jika DNA valid maka akan mengembalikan boolean true, jika tidak valid boolean false.
4.	KMPMatching(dna, disease)	Fungsi ini digunakan untuk mencari apakah dalam dna user terdapat rantai yang cocok dengan rantai dna disease dengan menggunakan algoritma KMP. Jika ditemukan pola dna penyakit yang sama pada dna user, maka akan mengembalikan nilai boolean true, jika tidak akan mengembalikan nilai boolean false.
5.	failTable(disease)	Fungsi untuk menentukan tabel <i>border function</i> dari pola yang sedang dicari. Mengembalikan array berisi nilai dari

		masing-masing <i>border function</i> / <i>fail function</i> .
6.	BMMatching(dna, disease)	Fungsi ini digunakan untuk mencari apakah dalam dna user terdapat rantai yang cocok dengan rantai dna disease dengan menggunakan algoritma BM. Jika ditemukan pola dna penyakit yang sama pada dna user, maka akan mengembalikan nilai boolean true, jika tidak akan mengembalikan nilai boolean false.
7.	lastTable(disease)	Fungsi yang mengembalikan array yang berisi nilai index terakhir munculnya masing-masing huruf pada pola yang diberikan.
8.	tanggalsekarang()	Fungsi untuk mengembalikan tanggal hari ini, fungsi ini akan digunakan pada fitur test DNA. Ketika user melakukan test dna, maka tanggal tes akan secara automatis dimasukkan tanggal hari ini.

2. Tata Cara Penggunaan Program

- User dapat mengakses program dalam repository github yang ada di bagian akhir laporan ini
- Clone repository github, dan ikuti cara menjalankan program sesuai yang ada di Readme
- Setelah program berhasil dijalankan, user akan masuk ke home page yang berisi penjelasan mengenai search.dna dan 3 fitur yang ada
- User memilih / menekan salah satu fitur
 - Jika user menekan fitur tambah penyakit, user dapat meyiapkan file txt berupa sequence DNA dari penyakit yang akan dimasukkan ke database
 - Ketik nama penyakit yang akan dimasukkan
 - Tekan choose file dan pilih file .txt yang sudah disiapkan sebelumnya
 - Tekan tombol submit

- Jika file .txt berisi sequence DNA yang valid, akan ada pesan “Data berhasil ditambahkan！”, jika tidak valid, akan muncul pesan “rantai DNA tidak valid.”
- Jika user menekan fitur Tes DNA, user dapat menyiapkan file .txt berupa sequence DNA dari user, dan prediksi penyakit yang diperkirakan user
 - Ketik nama user pada kolom Nama Pengguna
 - Masukkan file sequence DNA yang sudah disiapkan
 - Ketik prediksi penyakit
 - Pilih algoritma pencocokkan string
 - Tekan tombol submit
 - Jika rantai DNA dalam file yang dimasukkan tidak valid, akan muncul pesan kesalahan
 - Jika prediksi penyakit tidak ada dalam database penyakit maka akan muncul pesan kesalahan
 - Jika rantai DNA valid dan prediksi penyakit ada di dalam database, akan muncul hasil test dengan format

{Nama pengguna} - {Tanggal hari ini} - {Prediksi Penyakit} - {Hasil test}
- Jika user menekan fitur riwayat prediksi
 - Ketik query pencarian. Query dapat berupa tanggal / nama penyakit / tanggal dan nama penyakit.
 - Untuk format tanggal dapat berupa :
 - DD/MM/YYYY, contoh : 23/04/2022
 - DD-MM-YYYY, contoh : 23-04-2022
 - DD MM YYYY, contoh : 23 04 2022
 - DD/Month/YYYY, contoh : 23/April/2022
 - DD-Month-YYYY, contoh : 23-April-2022
 - DD Month YYYY, contoh : 23 April 2022
 - DD/Mon/YYYY, contoh : 23/Apr/2022
 - DD-Mon-YYYY, contoh : 23-Apr-2022
 - DD Mon YYYY, contoh : 23 Apr 2022
 - Jika query yang dimasukkan benar dan terdapat data yang sesuai di database, maka akan muncul data prediksi penyakit dengan format

{tanggal DD/MM/YYYY} - {Nama Pengguna} - {Nama Penyakit} - {Status penyakit}
 - Jika tidak, maka tidak akan muncul apapun

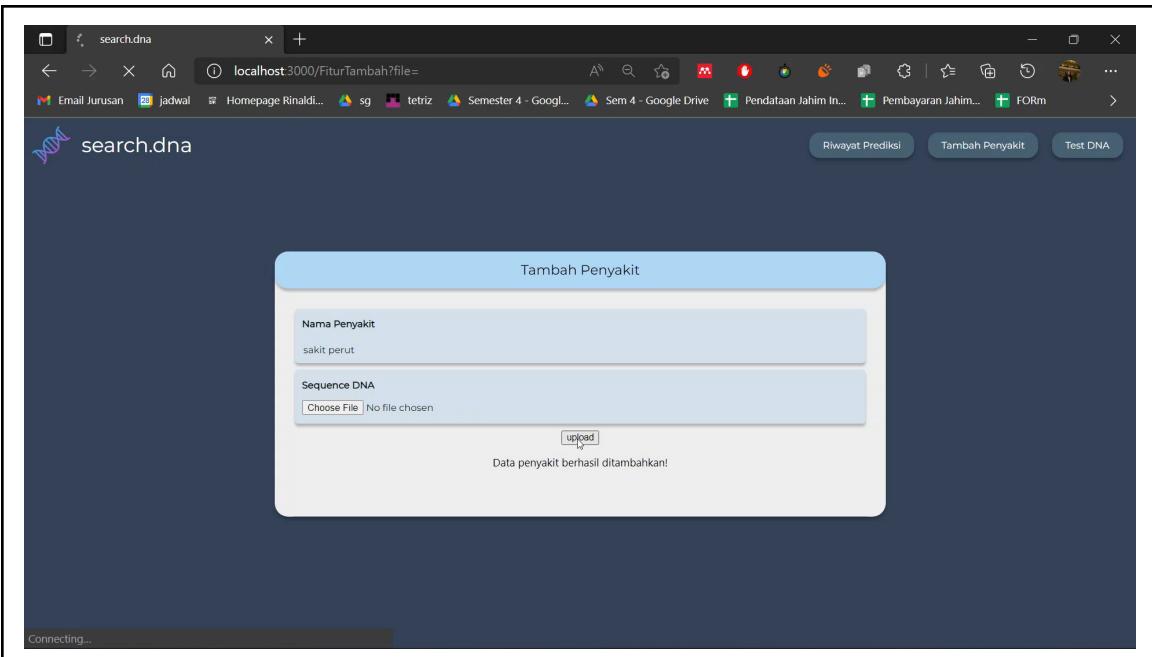
3. Hasil Pengujian

Home Page

The screenshot shows the search.dna website on a Microsoft Edge browser. The title bar says "search.dna" and "localhost:3000". The main content area has a dark blue background. On the left, there's a box titled "search.dna" containing a brief description of the website's purpose. On the right, there's a section titled "FITUR KAMI" with three buttons: "Riwayat Prediksi" (with a sub-instruction "Caril hasil test berdasarkan tanggal dan nama penyakit!"), "Tambah Penyakit" (with a sub-instruction "Tambahkan sequence DNA untuk penyakit baru!"), and "Test DNA" (with a sub-instruction "Cek apakah kamu memiliki penyakit tertentu berdasarkan sequence DNAmu!").

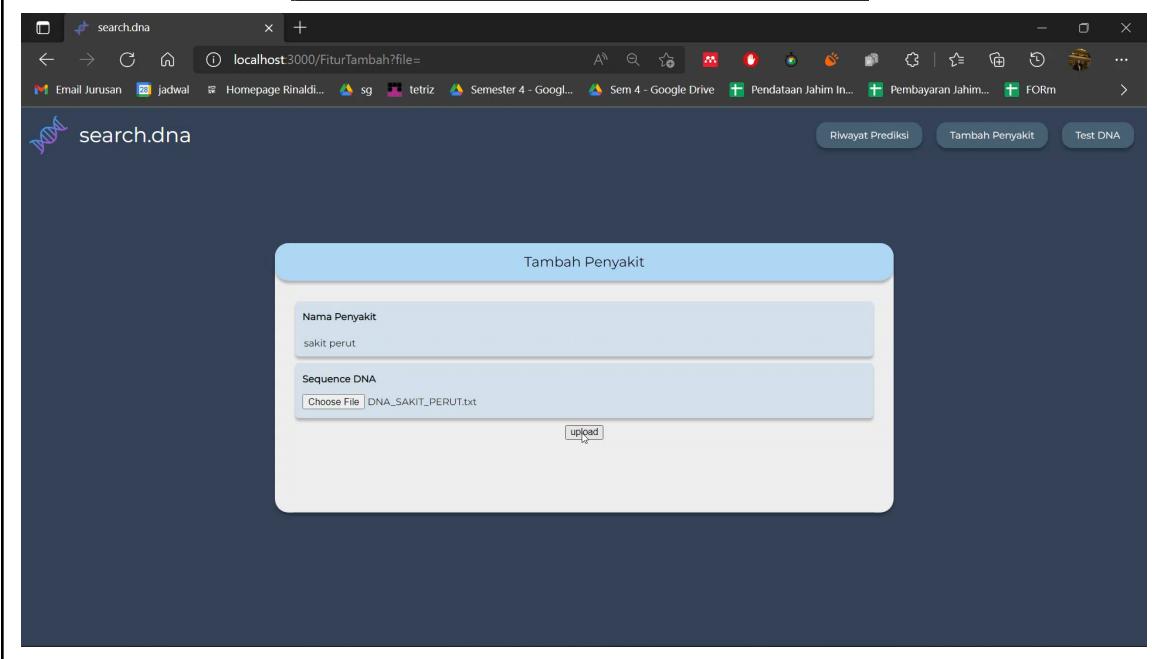
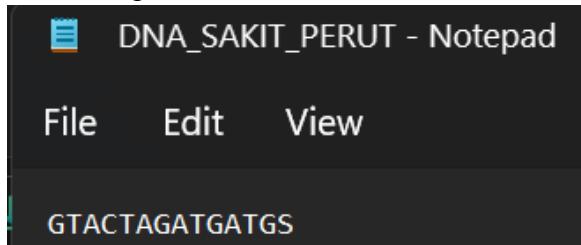
Fitur Tambah Penyakit - Rantai DNA Valid

This screenshot shows the "Tambah Penyakit" feature of the search.dna website. At the top, there's a small window showing a Notepad file named "DNA_SAKIT_PERUT_TRUE - Notepad" with the sequence "GTACTAGATGATGC" in it. Below this, the main browser window shows the "Tambah Penyakit" form. The form has two input fields: "Nama Penyakit" (containing "sakit perut") and "Sequence DNA" (containing "Choose File DNA_SAKIT_PERUT_TRUE.txt"). There is also an "upload" button below the sequence field.



Fitur Tambah Penyakit - Rantai DNA Tidak Valid

Terdapat huruf S di akhir antai DNA



localhost:3000/FiturTambah?file=

search.dna

Tambah Penyakit

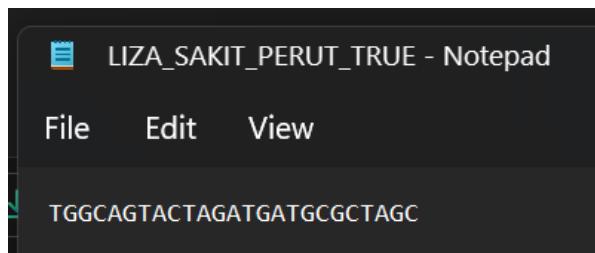
Nama Penyakit
sakit perut

Sequence DNA
Choose File | No file chosen

upload

Rantai DNA tidak valid

Fitur Tes DNA - Rantai DNA Valid, Prediksi Penyakit ada di database, Terdapat Sequence DNA yang sesuai pada DNA pengguna



localhost:3000/FiturPrediksi

search.dna

Test DNA

Nama Pengguna
lisa

Sequence DNA
Choose File | LIZA_SAKIT_PERUT_TRUE.txt

Prediksi Penyakit
sakit perut

Algoritma Pencocokan String
Knuth-Morris-Pratt

Submit

Test DNA

Nama Pengguna
lisa

Sequence DNA
Choose File | No file chosen

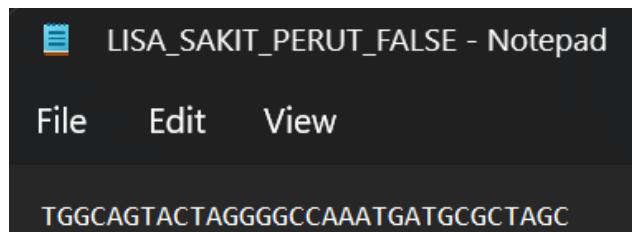
Prediksi Penyakit
sakit perut

Algoritma Pencocokan String
Knuth-Morris-Pratt

Submit

lisa - 29/04/2022 - sakit perut - true

Fitur Tes DNA - Rantai DNA Valid, Prediksi Penyakit ada di database, tidak terdapat Sequence DNA yang sesuai pada DNA pengguna



Test DNA

Nama Pengguna
lisa

Sequence DNA
Choose File | LISA_SAKIT_PERUT_FALSE.txt

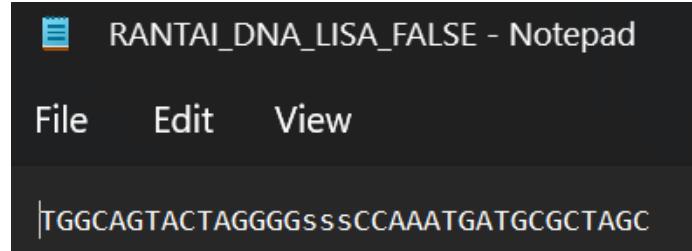
Prediksi Penyakit
sakit perut

Algoritma Pencocokan String
Boyer-Moore

Submit

Fitur Tes DNA - Rantai DNA Tidak Valid

Terdapat “sss” di tengah rantai DNA yang membuat rantai tidak valid



A screenshot of a web browser window titled "search.dna" at "localhost:3000/FiturPrediksi". The main content area is a "Test DNA" form. It contains four input fields: "Nama Pengguna" (User Name) with the value "lisa", "Sequence DNA" (DNA Sequence) with a placeholder "Choose File | No file chosen", "Prediksi Penyakit" (Disease Prediction) with a placeholder "nyeuri beuteund", and "Algoritma Pencocokan String" (String Matching Algorithm) with a dropdown menu set to "Boyer-Moore". Below the form is a message: "Rantai DNA tidak valid" (DNA chain is invalid). At the top right of the form are three buttons: "Riwayat Prediksi" (Prediction History), "Tambah Penyakit" (Add Disease), and "Test DNA".

Fitur Tes DNA - Nama Penyakit Tidak ada di database

A screenshot of a web browser window titled "search.dna" at "localhost:3000/FiturPrediksi". The main content area is a "Test DNA" form. It contains four input fields: "Nama Pengguna" (User Name) with the value "lisa", "Sequence DNA" (DNA Sequence) with a placeholder "Choose File | LIZA_SAKIT_PERUT_TRUE.txt", "Prediksi Penyakit" (Disease Prediction) with a placeholder "nyeuri beuteund", and "Algoritma Pencocokan String" (String Matching Algorithm) with a dropdown menu set to "Knuth-Morris-Pratt". Below the form is a message: "Submit". At the top right of the form are three buttons: "Riwayat Prediksi" (Prediction History), "Tambah Penyakit" (Add Disease), and "Test DNA".

localhost:3000/FiturPrediksi

search.dna

Test DNA

Nama Pengguna

Sequence DNA
Choose File | No file chosen

Prediksi Penyakit

Algoritma Pencocokan String
Knuth-Morris-Pratt ▾

Submit

Prediksi penyakit tidak ada di database

Fitur Riwayat Prediksi - Query Penyakit

localhost:3000/FiturRiwayat

search.dna

Riwayat Penyakit

Masukkan Query
sakit perut

Search

29/04/2022 - lisa - sakit perut - true
29/04/2022 - lisa - sakit perut - false

Fitur Riwayat Prediksi - Query Tanggal

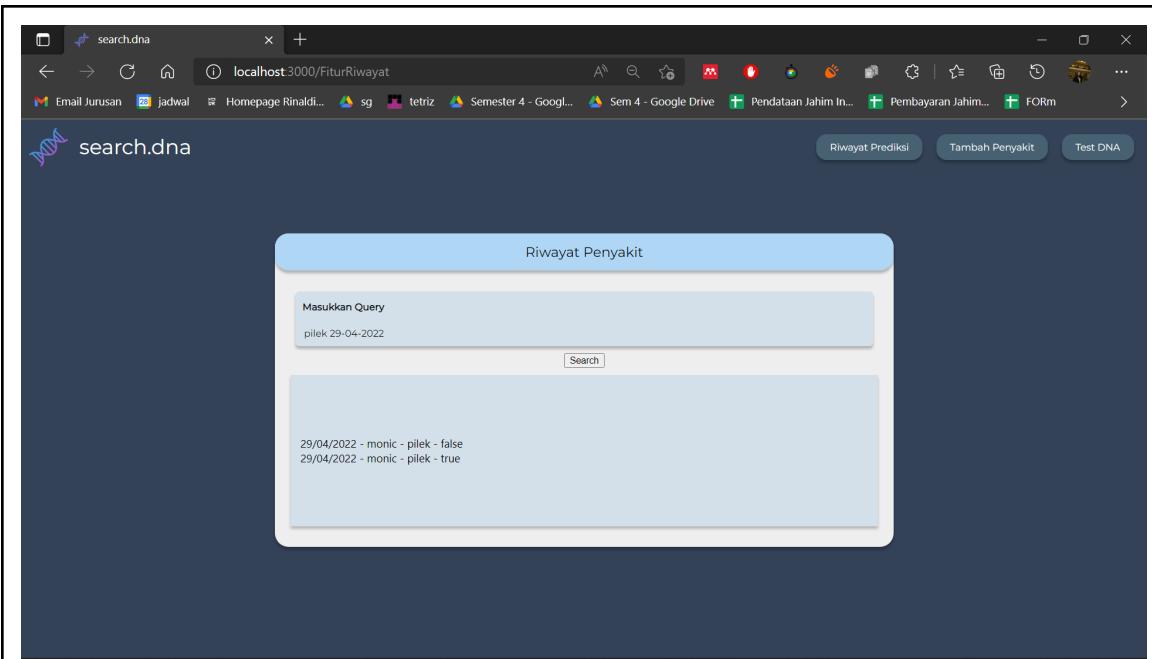
The screenshot shows a web browser window titled "search.dna" with the URL "localhost:3000/FiturRiwayat". The main content area is titled "Riwayat Penyakit" and contains a "Masukkan Query" input field with the value "29 Apr 2022". Below the input field is a "Search" button. The main content area displays a list of entries from April 29, 2022, each consisting of a date, a name, and a status. The entries are:

- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - false
- 29/04/2022 - fasdfa - Kena mental - false
- 29/04/2022 - jundann - Kena mental - false
- 29/04/2022 - sss - Kena mental - false
- 29/04/2022 - monic gila - Kena mental - false
- 29/04/2022 - fade - Kena mental - false
- 29/04/2022 - monic - Maag - false
- 29/04/2022 - monic - bau banget - false
- 29/04/2022 - monic - encok - false
- 29/04/2022 - monic - kaget - true
- 29/04/2022 - monic - kaget - false
- 29/04/2022 - monic - kaget - false
- 29/04/2022 - monic - encok - true
- 29/04/2022 - monic - encok - false
- 29/04/2022 - monic - plek - true
- 29/04/2022 - monic - plek - true
- 29/04/2022 - monic - plek - true
- 29/04/2022 - lisa - sakit perut - false
- 29/04/2022 - Monic - Batuk - true
- 29/04/2022 - Santo - Batuk - false
- 29/04/2022 - siti - Batuk - false

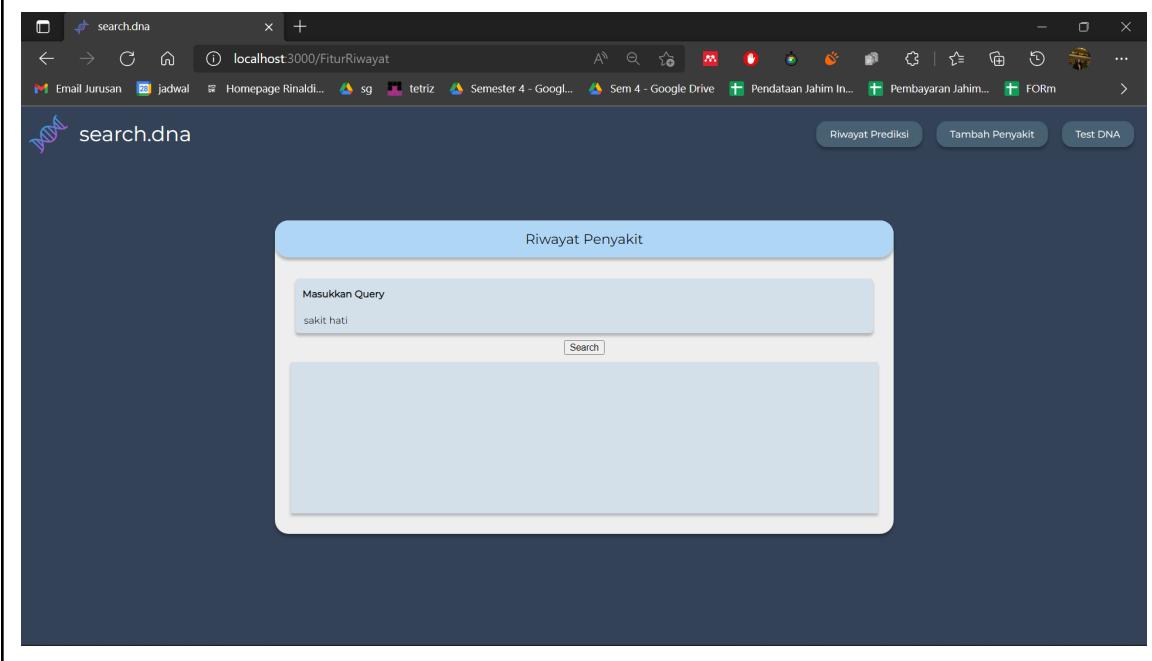
Fitur Riwayat Prediksi - Query Tanggal Penyakit dan Penyakit Tanggal

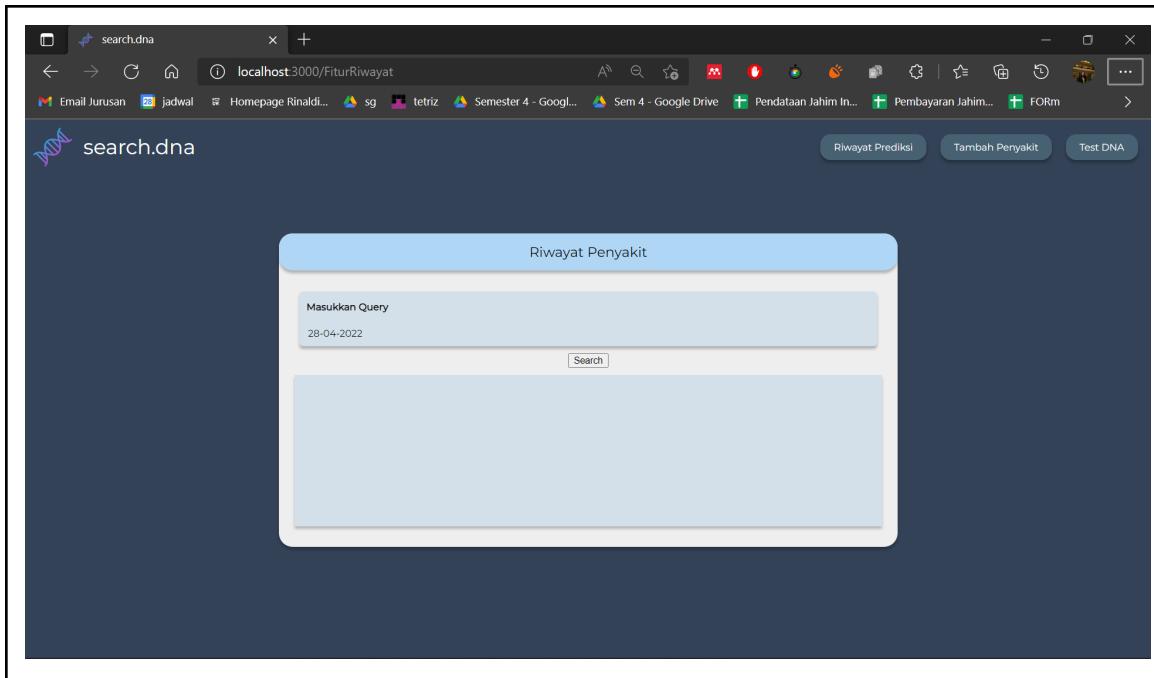
The screenshot shows a web browser window titled "search.dna" with the URL "localhost:3000/FiturRiwayat". The main content area is titled "Riwayat Penyakit" and contains a "Masukkan Query" input field with the value "29 Apr 2022 Kena". Below the input field is a "Search" button. The main content area displays a list of entries from April 29, 2022, each consisting of a date, a name, and a status. The entries are:

- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - true
- 29/04/2022 - hilda - Kena mental - false
- 29/04/2022 - fasdfa - Kena mental - false
- 29/04/2022 - jundann - Kena mental - false
- 29/04/2022 - sss - Kena mental - false
- 29/04/2022 - monic gila - Kena mental - false
- 29/04/2022 - fade - Kena mental - false



Fitur Riwayat Prediksi - Query Penyakit dan Tanggal Tidak Ada di Database





4. Analisis Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan, aplikasi web search.dna secara garis besar sudah bekerja sesuai dengan fitur-fiturnya. Pada fitur tambah DNA, jika kita memasukkan penyakit dengan rantai DNA yang benar, maka akan muncul pesan bahwa data sudah ditambahkan. Sedangkan jika rantai DNA yang dimasukkan salah, maka muncul juga pesan kesalahan bahwa rantai DNA tidak valid. Pada fitur test DNA-pun hal ini sudah bekerja dengan baik. Dengan terdapat pesan jika rantai DNA tidak valid atau penyakit yang user ketik tidak terdapat di database, namun juga menampilkan hasil test yang sesuai jika rantai DNA serta nama penyakit yang diinput sudah valid. Terakhir pada fitur Riwayat Prediksi, aplikasi akan menampilkan data yang sesuai untuk query yang benar. Query tanggal dapat memiliki berbagai format, dan query untuk bersamaan dengan penyakit juga dapat diinput dengan format tanggal penyakit ataupun penyakit tanggal. Akan tetapi, satu kekurangan dari fitur ini adalah jika format query yang diinput salah atau tidak terdapat data, tidak muncul pesan dan layar terlihat seperti tidak ada kejadian apapun.

Untuk string matching dalam program ini, user juga dapat memilih untuk menggunakan algoritma yang mana. Terdapat menu drop down untuk memudahkan user dalam memilih antara Knuth Morris Pratt Algorithm atau Boyer Moore Algorithm. Hasil yang ditampilkan juga sesuai, apabila DNA yang user input memiliki substring DNA penyakit yang juga user input, maka akan mengeluarkan hasil true, sedangkan jika DNA user tidak memiliki substring itu, akan dikembalikan hasil false.

Bab V : Kesimpulan, Saran, dan Komentar / Refleksi

1. Kesimpulan

Berdasarkan hasil pembuatan web aplikasi DNA Pattern Matching, dapat diambil keputusan bahwa algoritma Knuth–Morris–Pratt dan algoritma Boyer Moore dapat menyelesaikan permasalahan pattern matching dengan baik. Algoritma *string matching* maupun *regular expression* berhasil diimplementasikan dan digunakan dalam tes DNA. Kode (*backend maupun frontend*) dapat terhubung dengan baik sehingga web search.dna dapat tercipta dengan baik yang memenuhi spesifikasi tugas yang diberikan.

2. Saran

Untuk kedepannya, mungkin harus lebih sering mengulik mengenai web development, karena jika sebelumnya belum pernah sama sekali memegang web development akan sedikit kesusahan untuk menghubungkan dari backend ke frontend dan penggunaan database-nya. Selain itu, dapat juga diberi fitur-fitur lain agar web tidak ‘kosong’ seperti laman About Us, atau mungkin laman untuk menampilkan video demo. Untuk fitur-fitur yang ada sendiri, mungkin dapat ditambahkan pesan kesalahan saat query yang dimasukkan salah atau tidak terdapat data pada fitur riwayat prediksi.

3. Komentar / Refleksi

Materi yang diberikan dari kuliah Strategi Algoritma pada topik string matching kali ini jauh lebih efektif dibandingkan string matching dengan brute force seperti yang dipelajari di awal semester. Pemberian materi Regex juga membuat penggeraan fitur yang diberikan lebih mudah. Dengan penggeraan pada tubes ini, penulis juga jadi lebih mengerti materi regex dibandingkan saat mengenal regex pertama kali pada kuliah di semester 3 lalu.

Tugas besar ketiga ini cukup menantang, karena dari kami bertiga sendiri belum ada yang pernah secara penuh memegang web development sehingga kami semua sama-sama belajar dalam penggeraannya. Kerap kali kami merasa frustasi saat hasil yang diinginkan tidak tercapai juga, namun berkat google dan teman-teman yang mau menjawab saat diberi pertanyaan, akhirnya tugas ini selesai juga. Kami jadi banyak mengeksplor penggunaan MongoDB, MongoDB Atlas, dan React sehingga tubes ini juga bisa dijadikan protofolio untuk kedepannya.

Link Repository dan Video Demo

[hcarissa/Tubes3_13520096 \(github.com\)](https://github.com/hcarissa/Tubes3_13520096)

<https://hmif.link/VideoDemoTubes3StimaKel29>

Daftar Pustaka

Pencocokan String (String/Pattern Matching). Munir, R. 2022.

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

Diakses pada 26 April 2022, pukul 20.30 WIB.

String Matching dengan Regular Expression. Munir, R. 2022.

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regular-Expression-2019.pdf> Diakses pada 28 April 2022, pukul 19.30 WIB.