

LAPORAN TUGAS BESAR 3 IF2211

STRATEGI ALGORITMA

Penerapan String Matching dan Regular Expression dalam DNA Pattern Matching



Disusun oleh :
FRS Part2 - Kelompok 26

Febryola Kurnia Putri (13520140)
Steven Gianmarg H. Siahaan (13520145)
Raden Rifqi Rahman (13520166)

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

Daftar Isi

Daftar Isi	2
BAB I	3
 Deskripsi Tugas	3
 BAB II	8
 Landasan Teori	8
Deskripsi singkat algoritma KMP, BM, dan Regex	8
Penjelasan singkat mengenai aplikasi web yang dibangun	9
 BAB III	11
 Analisis Pemecahan Masalah	11
 Langkah Penyelesaian Masalah Setiap Fitur	11
Fitur fungsional dan arsitektur aplikasi web yang dibangun	13
 BAB IV	15
 Implementasi dan Pengujian	15
Spesifikasi teknis program (struktur data, fungsi, prosedur yang dibangun)	15
Penjelasan tata cara penggunaan program (interface program, fitur-fitur yang disediakan program, dan sebagainya)	21
Hasil pengujian (Screenshot antarmuka dan skenario yang memperlihatkan berbagai kasus yang mencakup seluruh fitur pada aplikasi DNA Sequence Matching)	22
Analisis hasil pengujian	32
 BAB V	35
 Kesimpulan	35
Kesimpulan	35
Saran	35
Refleksi	35
 LAMPIRAN	37
 DAFTAR PUSTAKA	38

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.

Gambar 1. Ilustrasi Input Penyakit

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>

Gambar 2. Ilustrasi Prediksi

- 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: <tanggal_prediksi><spasi><nama_penyakit>, 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:
 1. 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.

Gambar 3. Ilustrasi Interaksi 1

2. 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.

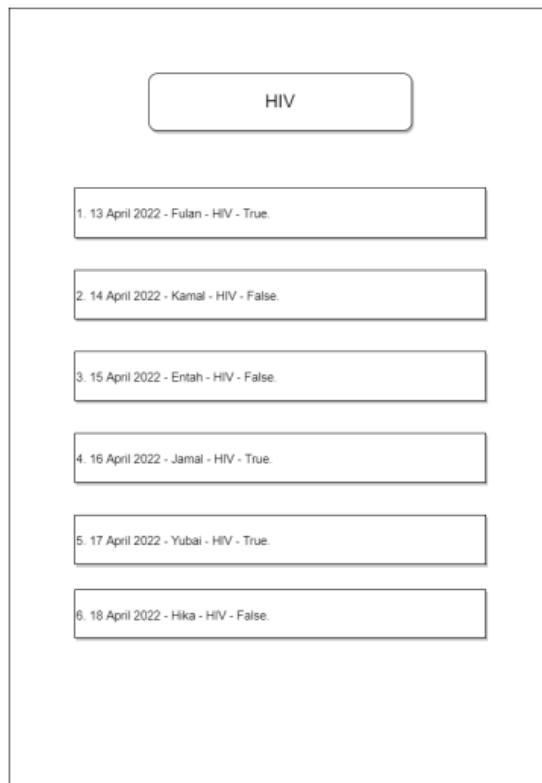
4. 13 April 2022 - Jamal - Polio - True.

5. 13 April 2022 - Yubai - TBC - True.

6. 13 April 2022 - Hika - Hepatitis A - False.

Gambar 4. Ilustrasi Interaksi 2

3. Masukan hanya nama penyakit



Gambar 5. Ilustrasi interaksi 3

4. Menghitung tingkat kemiripan DNA pengguna dengan DNA penyakit pada tes DNA
- a. Ketika melakukan tes DNA, terdapat persentase kemiripan DNA dalam hasil tes. Contoh hasil tes: 1 April 2022 - Mhs IF - HIV - 75% - False
 - b. Perhitungan tingkat kemiripan dapat dilakukan dengan menggunakan Hamming distance, Levenshtein distance, LCS, atau algoritma lainnya (dapat dijelaskan dalam laporan).
 - c. 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.
 - d. Contoh tampilan:

Tes DNA

Nama Pengguna:
`<pengguna>`

Sequence DNA:
`upload file...`

Prediksi Penyakit:
`<penyakit>`

Submit

Hasil Tes

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

Gambar 6. Ilustrasi interaksi 4

BAB II

Landasan Teori

1. Deskripsi singkat algoritma KMP, BM, dan Regex

A. Algoritma KMP

Algoritma KMP atau singkatan dari Knuth-Morris-Pratt merupakan algoritma pencarian pola dalam teks dalam urutan kiri-ke-kanan (seperti pada algoritma brute force) akan tetapi, melakukan penggeseran pola dengan lebih cerdas daripada algoritma brute force. Kompleksitas waktu dari algoritma ini adalah $O(m)$ yang berasal dari fungsi pinggirannya. $O(n)$ untuk pencarian string, serta $O(m+n)$ untuk keseluruhan algoritmanya yang mana sangat cepat jika dibandingkan dengan brute force. KMP memproses pola untuk menemukan kecocokan awalan pola dengan pola itu sendiri. Di mana berlaku hal-hal sebagai berikut:

- j = posisi ketidakcocokan di $P[]$
- k = posisi sebelum ketidakcocokan ($k = j - 1$).
- Fungsi perbatasan $b(k)$ didefinisikan sebagai ukuran yang terbesar
- awalan $P[0..k]$ yang juga merupakan akhiran $P[1..k]$.
- Nama lain: fungsi kegagalan (disingkat: fail)

Hal pertama yang harus dilakukan adalah menentukan border functionnya dengan ketentuan berikut:

- jika ketidakcocokan terjadi di $P[j]$ (yaitu $P[j] \neq T[i]$), lalu

$$\begin{aligned} k &= j-1; \\ j &= b(k); \text{ Dapatkan } j \text{ baru} \end{aligned}$$

Keuntungan dari penggunaan algoritma ini adalah Algoritma tidak perlu bergerak mundur dalam input teks. Ini membuat algoritma yang baik untuk diproses sangat besar file yang dibaca dari perangkat eksternal atau melalui aliran jaringan. Namun, terdapat juga kekurangan dari penggunaan algoritma ini, yaitu tidak bekerja dengan baik seperti pada ukuran peningkatan alfabet.

B. Algoritma Boyer Moore

Tidak seperti algoritme pencarian string yang ditemukan sebelumnya, algoritme Boyer-Moore mulai mencocokkan karakter dari sebelah kanan pattern. Ide di balik algoritme ini adalah bahwa dengan memulai pencocokan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat.

Cara kerja algoritma ini adalah :

Misalnya ada sebuah usaha pencocokan yang terjadi pada $\text{teks}[i..i+n-1]$, dan anggap ketidakcocokan pertama terjadi di antara $[i+j]$ dan $\text{pattern}[j]$, dengan $0 < j < n$. Berarti, $\text{teks}[i+j+1..i+n-1] = \text{pattern}[j+1..n-1]$ dan $a = \text{teks}[i+j]$ tidak sama dengan $b = \text{pattern}[j]$. Jika u adalah akhiran dari pattern sebelum b dan v adalah sebuah awalan dari pattern, maka penggeseran-penggeseran yang mungkin adalah:

1. Penggeseran good-suffix yang terdiri dari menyejajarkan potongan teks[i+j+1..i+n-1] = pattern[j+1..n-1] dengan kemunculannya paling kanan di pattern yang didahului oleh karakter yang berbeda dengan pattern[j]. Jika tidak ada potongan seperti itu, maka algoritma akan menyejajarkan akhiran v dari teks[i+j+1..i+n-1] dengan awalan dari pattern yang sama.
2. Penggeseran bad-character yang terdiri dari menyejajarkan teks[i+j] dengan kemunculan paling kanan karakter tersebut di pattern. Bila karakter tersebut tidak ada di pattern, maka pattern akan disejajarkan dengan teks[i+n+1]

C. Regex

Regex adalah singkatan dari *Regular Expression*. Regex merupakan sebuah teks (string) yang mendefinisikan sebuah pola pencarian sehingga dapat membantu kita untuk melakukan *matching* (pencocokan), *locate* (pencarian), dan manipulasi teks. Konsep tentang regex pertamakali muncul di tahun 1951, ketika seorang ilmuwan matematikan bernama Stephen Cole Kleene memformulasikan definisi tentang *bahasa formal*. Kemudian konsep ini diadopsi di beberapa program dan menjadi umum digunakan pada program pemroses teks seperti sed, awk, dan lain-lain. Regex memiliki banyak manfaat di antaranya:

1. untuk validasi data
2. untuk pencarian sebagaimana yang dimanfaatkan dalam pembuatan tugas besar kali ini
3. untuk menemukan dan mengubah pola

Berikut beberapa simbol basic yang digunakan di dalam regex:

- () — untuk grouping;
- | — simbol untuk logika or (atau alternatif);
- [a-z] — untuk mencari karakter dari huruf a sampai z;
- [A-Z] — untuk mencari karakter dari huruf A sampai Z;
- [0-9] — Untuk mencari angka 0 sampai 9;
- \w — untuk mencari kata;
- \d — untuk mencari angka;

2. Penjelasan singkat mengenai aplikasi web yang dibangun

Pada website yang dibangun ini, untuk bagian front endnya menggunakan react js. Dan backend yang menggunakan node js serta mysql pada databasenya. Ekstensi yang digunakan adalah .jsx yang sebenarnya tidak ada bedanya dengan .js. Hanya saja jsx telah dilengkapi kekuatan penuh dari javascript. Pada website ini, terdapat 5 navigation bar yang akan membantu dalam proses perpindahan page, yakni ada Home,Prediction,History,Disease, serta about. Pada setiap pagenya dilengkapi footer yang

seragam yang berisikan nama project yakni “FRS Part 2”, penjelasannya serta menu-menu yang tersedia.

Pada Home hanya berisikan, tampilan yang akan mengarahkan anda ke menu prediction jika anda ingin melakukan cek dan analisis terhadap DNA anda.

Pada menu prediction ini akan meminta anda untuk memasukkan nama, file DNA,disease(memilih penyakit dari database yang ada), serta memilih method yang digunakan(KMP atau Booyer Moore). Untuk memasukkan file, hanya tinggal meng-klik browse file lalu memasukkan file sesuai ketentuan, yakni harus berformat .txt serta hanya berisi huruf A,C,G,T dimana file tanpa spasi &enter serta harus huruf besar. Saat melengkapi data, silahkan liat disebelah kanan untuk memeriksa data yang telah isi(automatis terupdate). Lalu setelah semua data dilengkapi silahkan klik result untuk mendapatkan hasil.

Pada menu history akan menampilkan data hasil pemeriksaan dari database. Pada menu history ini juga akan terdapat input berupa search, pada search tersebut silahkan masukkan inputan sesuai yang diinginkan, pada web ini dapat menghandle beberapa kasus, yakni pencarian melalui nama,pencarian melalui tanggal,pencarian melalui status, pencarian melalui nama penyakit,pencarian melalui nama penyakit dan status serta pencarian melalui tanggal dan nama penyakit.

Pada menu disease, merupakan menu yang disiapkan jika pengguna ingin menambahkan penyakit ke database. Pengguna akan diminta untuk memasukkan file sesuai ketentuan, yakni harus berformat .txt serta hanya berisi huruf A,C,G,T dimana file tanpa spasi &enter serta harus huruf besar. Untuk mencari file tinggal meng-klik browse file lalu melengkapi nama penyakit lalu meng-klik Add untuk menambahkan.

Pada menu about ini akan ditampilkan penjelasan singkat project dan juga penjelasan singkat dari FRS sendiri. Selain itu, juga disertakan *contributor* dalam penggerjaan tugas ini.

BAB III

Analisis Pemecahan Masalah

1. Langkah Penyelesaian Masalah Setiap Fitur

1. Menambahkan penyakit pada database pada relasi jenispenyakit

1. Membuat fungsi reader untuk membaca file yang diinputkan user, apakah file sesuai yaitu hanya dalam format .txt dan isi file harus text dalam format ACGT huruf besar semua dan tanpa spasi
2. Membuat fungsi atau query insert to database pada mysql dalam format js, fungsi ini nantinya akan memasukkan nama penyakit dan rantai DNanya ke dalam relasi jenispenyakit pada database
3. Membaca inputan dari user, berupa nama penyakit dan membaca rantai DNA dari file yang diinputkan
4. Jika inputan rantai DNA sesuai, maka akan lanjut ke langkah berikutnya, akan tetapi jika tidak, maka program akan mengeluarkan message “invalid DNA sequence”
5. Memasukkan nama penyakit dan rentai DNA yang terdapat pada file ke dalam database yaitu pada relasi jenispenyakit dengan menggunakan fungsi yang sudah dibuat pada bagian no 2.

2. Melakukan pencarian hasil prediksi berdasarkan masukan tanggal

1. Membuat fungsi pencarian yaitu getHasilPrediksiByTanggal di mana fungsinya akan menjalankan query berupa `SELECT * FROM hasilprediksi WHERE tanggalPrediksi = date yang diinputkan`
2. Membuat fungsi untuk mendeteksi apakah inputan yang dilakukan sudah benar yaitu tanggal terlebih dahulu dengan format dd namaBulan yy atau dd/mm/yy
3. Membaca inputan tanggal dari user, lalu mengonversikan inputan tanggal agar dapat dibaca program
4. Menjalankan fungsi pada bagian no 1 dengan melakukan pembacaan pada database pada relasi hasilprediksi
5. Jika query dijalankan dan benar (data terdapat pada database), maka akan ditampilkan hasil-hasil yang sesuai pada halaman history
6. Jika masukan tidak sesuai, maka website akan mengeluarkan message “Not Found!”

3. Melakukan pencarian hasil prediksi berdasarkan masukan nama penyakit

1. Membuat fungsi pencarian yaitu getHasilPrediksiByPenyakit di mana fungsinya akan menjalankan query berupa `SELECT * FROM hasilprediksi WHERE penyakitPrediksi = nama penyakit yang diinputkan`
2. Membaca inputan nama penyakit dari user

3. Menjalankan fungsi pada bagian no 1 dengan melakukan pembacaan pada database pada relasi hasilprediksi
 4. Jika query dijalankan dan benar (data terdapat pada database), maka akan ditampilkan hasil-hasil yang sesuai pada halaman history
 5. Jika masukan tidak sesuai, maka website akan mengeluarkan message “Not Found!”
- 4. Melakukan pencarian hasil prediksi berdasarkan masukan tanggal dan nama penyakit**
 1. Membuat fungsi pencarian yaitu getHasilPrediksiByTanggalAndPenyakit di mana fungsinya akan menjalankan query berupa `SELECT * FROM hasilprediksi WHERE tanggalPrediksi = date yang diinputkan AND penyakitPrediksi = nama penyakit yang diinputkan`
 2. Membuat fungsi untuk mendeteksi apakah inputan yang dilakukan sudah benar yaitu tanggal terlebih dahulu dengan format dd namaBulan yy atau dd/mm/yy dan nama penyakit dengan diikuti tanda spasi
 3. Membaca inputan tanggal dan nama penyakit dari user
 4. Menjalankan fungsi pada bagian no 1 dengan melakukan pembacaan pada database pada relasi hasilprediksi
 5. Jika query dijalankan dan benar (data terdapat pada database), maka akan ditampilkan hasil-hasil yang sesuai pada halaman history
 6. Jika masukan tidak sesuai, maka website akan mengeluarkan message “Not Found!”
- 5. Melakukan Tes DNA (main fitur)**
 1. Membuat fungsi reader untuk membaca file yang diinputkan user, apakah file sesuai yaitu hanya dalam format .txt dan isi file harus text dalam format ACGT huruf besar semua dan tanpa spasi
 2. Membuat fungsi untuk membaca seluruh kolom penyakitPrediksi pada bagian relasi hasilprediksi lalu menampilkannya ke pilihan disease pada inputan user
 3. Membuat fungsi untuk membuat pilihan metode pada inputan metode user yang terdiri atas KMP dan Boyer Moore
 4. Membuat fungsi DNA Pattern matching dalam algoritma KMP dan algoritma boyer moore sebagaimana sesuai dengan spesifikasi tugas
 5. Membaca semua inputan user, yaitu nama pasien, file .txt yang berisikan rantai DNA dalam format ACGT, penyakit yang dipilih, serta metode yang dipilih
 6. Jika masukan rantai DNA formatnya sesuai (dalam ACGT) maka akan dilanjutkan pada langkah selanjutnya. Jika masukan tidak sesuai, maka website akan mengeluarkan message “Not Found!”
 7. Selanjutnya DNA yang dibaca akan dimasukkan ke fungsi pada bagian no 3 untuk menentukan hasil pattern matchingnya dan tingkat similarity
 8. Menampilkan hasil pada bagian kanan website
 9. Memasukkan semua data inputan user pada database pada relasi hasilprediksi

2. Fitur fungsional dan arsitektur aplikasi web yang dibangun

a. Backend

a. bagian proses

Backend pada bagian ini akan berisikan penjelasan mengenai penggunaan algoritma KMP, algoritma Boyer Moore, dan regex dalam penentuan DNA pattern matching ini. Di mana pada bagian ini akan dilakukan peninjauan apakah inputan rantai DNA pengguna sesuai dengan yang terdapat pada database dan menampilkan tingkat similaritynya di mana jika >80% akan dinyatakan true dan selain itu akan dinyatakan false.

b. bagian database

Backend bagian database dibuat dengan menggunakan mysql hosting server, di mana pada bagian ini dibuat database dengan relasi hasilprediksi dan jenispenyakit. Di mana relasi hasilprediksi memiliki atribut-atribut berupa (id dengan domain int dan sebagai primary key, namaPasien dengan domain string(varchar), tanggalPrediksi dengan domain date, penyakitPrediksi dengan domain string(varchar), statusPrediksi dengan default nilai yaitu true dan false, createdAt dengan domain date, dan updatedAt dengan domain date). Selanjutnya untuk relasi jenispenyakit memiliki atribut sebagai berikut, yaitu (id dengan domain int dan sebagai primary key, namaPenyakit dengan domain string(varchar), rantaiDNA dengan domain text, createdAt dengan domain date, dan updatedAt dengan domain date).

b. Frontend

Pada bagian front end digunakan react js dalam membangun

a. App

Folder ini berisikan routes yang akan digunakan sebagai path.

b. images

Folder ini berisikan gambar-gambar yang digunakan selama proses pembangunan website.

c. page

i. Prediction.jsx

File ini berisikan halaman prediction yang akan melakukan pemeriksaan kecocokan dengan algoritma backend yang telah dibuat sebelumnya. Halaman ini akan di connect ke database.

ii. Disease.jsx

File ini berisikan halaman disease yang akan melakukan penambahan penyakit.

iii. Home.jsx

File ini berisikan halaman home yang akan menampilkan button check,yang jika di click akan mengarahkan ke menu prediksi.

iv. History.jsx

File ini berisikan halaman history yang akan menampilkan riwayat pemeriksaan.

v. About.jsx

File ini berisikan halaman about yang berisikan penjelasan project dan juga penjelasan contributors.

- d. App.jsx

BAB IV

Implementasi dan Pengujian

1. Spesifikasi teknis program (struktur data, fungsi, prosedur yang dibangun)

A. Struktur Data

a. Data

Atribut	Fungsi
n = text.length	untuk menyimpan panjang dari text
m = pattern.length	untuk menyimpan panjang dari pattern
border = calcKmpBorder(pattern)	untuk menghitung fungsi pinggiran dari pattern
border = Array(string.length)	untuk menyimpan panjang dari string dalam array
lastCharIndices = lastIndices(pattern);	untuk menyimpan last indices character dari pattern

b. Method

Method	Fungsi
function match(text, pattern)	untuk melakukan pencocokan string secara otomatis baik dengan algoritma KMP ataupun Boyer Moore
function kmpMatch(text, pattern)	untuk melakukan pencocokan string dengan algoritma KMP
function calcKmpBorder(string)	untuk menghitung fungsi pinggiran pada algoritma KMP
function bmMatch(text, pattern)	untuk melakukan pencocokan string dengan algoritma Boyer Moore

function lastIndices(string)	untuk menghitung index terakhir karakter dari string
function levenshteinDistance(string1, string2)	untuk menghitung levenshtein distance dari dua buah string yang diberikan

B. Fungsi dan Prosedur yang Dibangun

1. Fungsi untuk pattern matching dengan algoritma KMP

```
function kmpMatch(text, pattern) {
    let n = text.length;
    let m = pattern.length;
    let border = calcKmpBorder(pattern);
    let i = 0;
    let j = 0;
    while (i < n) {
        if (text[i] == pattern[j]) {
            if (j == m - 1) {
                return i - m + 1;
            }
            i++;
            j++;
        } else if (j > 0) {
            j = border[j - 1];
        } else {
            i++;
        }
    }
    return -1;
}
```

2. Fungsi untuk pattern matching dengan algoritma Boyer Moore

```
function bmMatch(text, pattern) {  
    let lastCharIndices = lastIndices(pattern);  
    let n = text.length;  
    let m = pattern.length;  
    let i = m - 1;  
    let j = m - 1;  
    if (i > n - 1) {  
        return -1;  
    }  
    do {  
        if (text[i] == pattern[j]) {  
            if (j == 0) {  
                return i;  
            } else {  
                i--;  
                j--;  
            }  
        } else {  
            let lastIndex = lastCharIndices[text[i]];  
            lastIndex = lastIndex == undefined ? -1 : lastIndex;  
            i = i + m - Math.min(j, 1 + lastIndex);  
            j = m - 1;  
        }  
    } while (i <= n - 1);  
    return -1;  
}
```

3. Fungsi untuk menampilkan pencarian berdasarkan tanggal

```
const getHasilPrediksiByTanggal = async (date) => {
  return new Promise((resolve, reject) => {
    pool.getConnection((err, connection) => {
      connection.connect(() => {
        connection.query(
          `SELECT * FROM hasilprediksi WHERE tanggalPrediksi = '${date}'`,
          (err, result) => {
            resolve(result); } } );
    });
  });
}
```

4. Fungsi untuk menampilkan pencarian berdasarkan nama penyakit

```
const getHasilPrediksiByPenyakit = async (disease) => {
  return new Promise((resolve, reject) => {
    pool.getConnection((err, connection) => {
      connection.connect(() => {
        connection.query(
          `SELECT * FROM hasilprediksi WHERE penyakitPrediksi = '${disease}'`,
          (err, result) => {
            resolve(result); } } );
    });
  });
}
```

5. Fungsi untuk menampilkan pencarian berdasarkan tanggal dan nama penyakit

```
const getHasilPrediksiByTanggalAndPenyakit = async (date, disease) => {
  return new Promise((resolve, reject) => {
    pool.getConnection((err, connection) => {
```

```

connection.connect(() => {
  connection.query(
    `SELECT * FROM hasilprediksi WHERE tanggalPrediksi = '${date}' 
    AND penyakitPrediksi = '${disease}'`,
    (err, result) => {
      resolve(result);});});;

```

6. Fungsi untuk menampilkan semua pada database

```

const getHasilPrediksi = async () => {
  return new Promise((resolve, reject) => {
    pool.getConnection((err, connection) => {
      connection.connect(() => {
        connection.query("SELECT * FROM hasilprediksi;", (err, result) => {
          resolve(result);});});});;

```

7. Fungsi untuk menambahkan hasil prediksi pada database

```

const createHasilPrediksi = async (date, name, disease, status) => {
  return new Promise((resolve, reject) => {
    pool.getConnection((err, connection) => {
      connection.connect(() => {
        connection.query(
          `INSERT INTO hasilprediksi (tanggalPrediksi, namaPasien,
          penyakitPrediksi, statusPrediksi, createdAt)
          VALUES ('${date}', '${name}', '${disease}', '${status}', '${currentDate()}')`,
          (err, result) => {`
```

```
    resolve(result);} } } } };
```

8. Fungsi untuk mendapatkan respons dari pencarian (history)

```
function respondWithHistory(response, records) {  
  const responseData = [];  
  for (const record of records) {  
    const date = new Date(Date.parse(record.tanggalPrediksi));  
    responseData.push({  
      name: record.namaPasien,  
      Date: dateToString(date),  
      Disease: record.penyakitPrediksi,  
      result: record.statusPrediksi,  
    });  
  }  
  response.status(200).send(responseData).end();}
```

9. Fungsi untuk melakukan update pada hasil prediksi

```
const updateHasilPrediksi = async (date, name, disease, status) => {  
  return new Promise((resolve, reject) => {  
    pool.getConnection((err, connection) => {  
      connection.connect(() => {  
        connection.query(  
          `UPDATE hasilprediksi  
          SET  
          VALUES ('${date}', '${name}', '${disease}', '${status}', '${currentDate()}')  
        (err, result) => {  
          if (err) {  
            reject(`Error: ${err.message}`);  
          } else {  
            resolve(result);  
          }  
        };  
      });  
    });  
  });  
};
```

```
    resolve(result);}}}}};
```

10. Fungsi untuk mendapatkan current date

```
function currentDate() {  
  const date = new Date(Date.now());  
  return `${date.getFullYear()}-${date.getMonth() + 1}-${date.getDate()}`;}
```

11. Fungsi untuk mendapatkan metode algoritma yang diinginkan baik algoritma KMP maupun algoritma Booyer Moore

```
switch (method) {  
  case "auto":  
    foundAt = match(dnaSequence, diseaseSequence);  
    break;  
  case "kmp":  
    foundAt = kmpMatch(dnaSequence, diseaseSequence);  
    break;  
  case "bm":  
    foundAt = bmMatch(dnaSequence, diseaseSequence);  
    break;
```

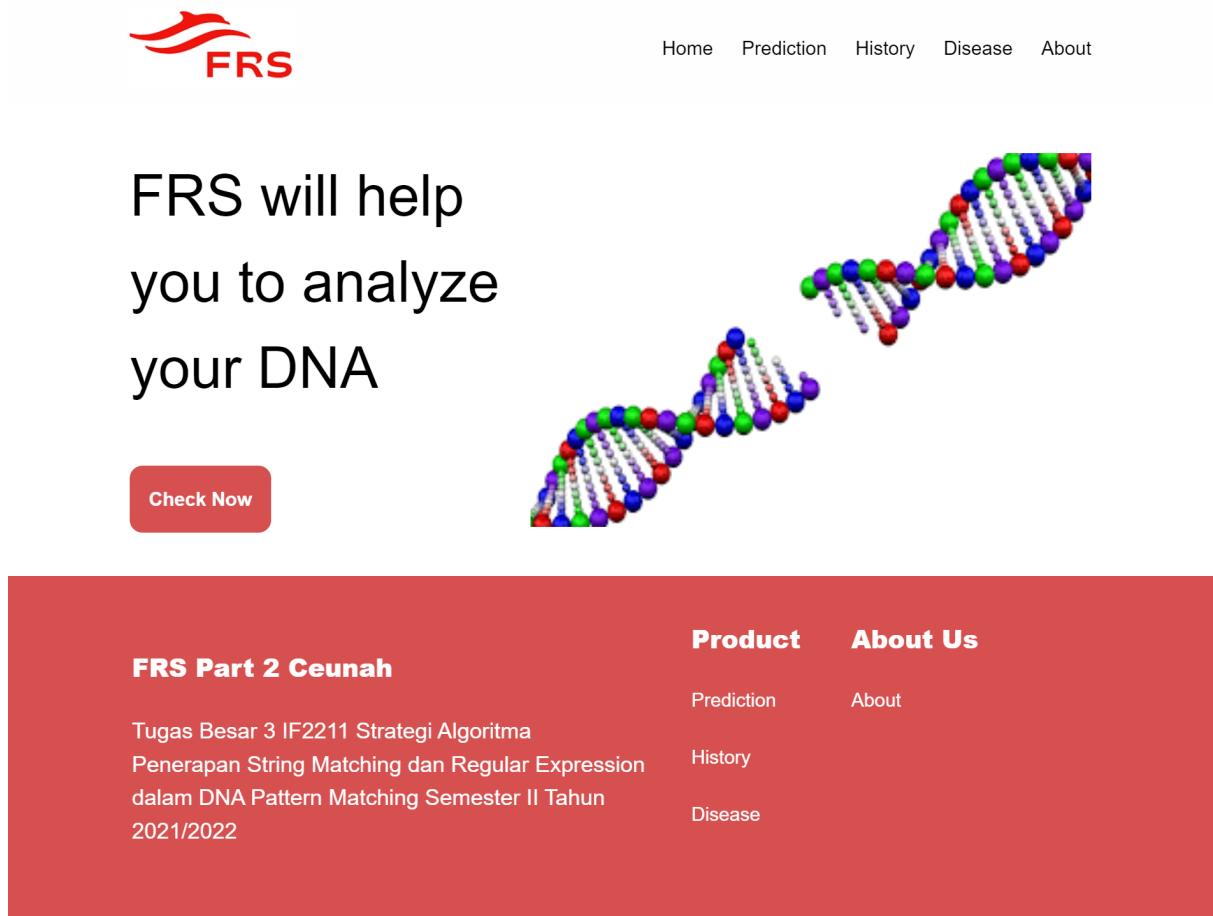
2. Penjelasan tata cara penggunaan program (interface program, fitur-fitur yang disediakan program, dan sebagainya)

Program dapat dijalankan dengan 2 cara, yaitu dengan menggunakan CLI pada terminal berupa ‘npm run dev’ atau langsung mengakses website yang telah dideploy pada link berikut : <https://hmif.link/FRSPart2.com> . Pada awal pengaksesannya, pengguna akan diarahkan pada halaman home page yang tampilannya dapat dilihat pada bagian 3 di bawah. Adapun fitur-fitur yang terdapat pada program ini di antaranya adalah, Menambahkan penyakit pada database "jenis penyakit", Melakukan

pencarian hasil prediksi berdasarkan tanggal, Melakukan pencarian hasil prediksi berdasarkan penyakit prediksi, Melakukan pencarian hasil prediksi berdasarkan tanggal dan penyakit prediksi, Melakukan tes DNA, Menampilkan result statusnya (true or false), dan Menampilkan persen kemiripan penyakit.

3. Hasil pengujian (screenshot antarmuka dan skenario yang memperlihatkan berbagai kasus yang mencakup seluruh fitur pada aplikasi DNA Sequence Matching)

A. Tampilan Awal Semua AntarMuka



Gambar 6. Tampilan home page website

Test DNA!

Name

File DNA

*Format File .txt
*File hanya berisikan A, C, G, dan/atau T tanpa enter spasi, dan harus huruf besar

Browse File No File Selected

Disease

Select Disease

Select Method

Result

Result

Name Date Test Disease Method Result Similarity

Empty

Empty

Empty

Empty

Empty

Empty

Gambar 7. Tampilan halaman prediksi

History

Search History Disease

Search

Name	Date Test	Disease	Result
Tepen	2022-04-22T00:00:00.000Z	HIV	true
Bono	2022-04-13T00:00:00.000Z	Radang Pernapasan	true
Jono	2022-04-27T00:00:00.000Z	Gatel Paru Paru	false
Raden	2022-04-27T00:00:00.000Z	Borok	true
Yoll	2022-04-27T00:00:00.000Z	Albinisme	true

Gambar 8. Tampilan Halaman History



Home Prediction History Disease About

Add Disease

Input Name

File DNA

*Format File .txt
*File hanya berisi huruf A, C, G, dan/atau T tanpa enter,spasi, dan harus huruf besar

Browse File

No File Selected

Add

Nama Penyakit/Kelainan yang Sudah Terdaftar

Gatel Paru Paru

Borok

Budug

Koreng

Rakitis

Disentri

Flu Babi

Gambar 9. Tampilan Halaman Add Penyakit Baru



Home Prediction History Disease About

About

DNA Pattern Matching adalah cara mencocokkan DNA user dengan DNA penyakit untuk memprediksi berbagai macam penyakit yang tersimpan pada basis data berdasarkan urutan sekuens DNA-nya. Sebuah sekuens DNA adalah proses atau teknik penentuan urutan basa nukleotida pada suatu molekul DNA serta merupakan suatu representasi *string of nucleotides* yang disimpan pada suatu rantai DNA, sebagai contoh: ACGTAGTTCGTAACTAGTAAAGTTA.





Gambar 10. Tampilan menu About

B. Case Tambah Penyakit Baru dengan inputan file benar (hanya berisikan ACGT format)

Home Prediction History Disease About

Add Disease

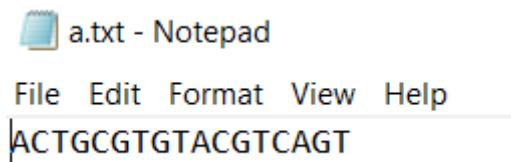
Input Name

File DNA

*Format File .txt
*File hanya berisi huruf A, C, G, dan/atau T tanpa enter,spasi, dan harus huruf besar

a.txt

Gambar 11. Inputan penyakit baru dengan file benar



Gambar 12. Isi File pengujian

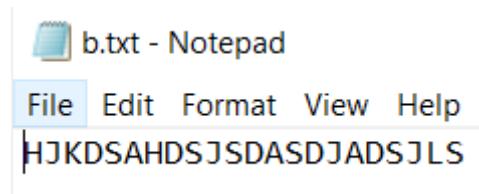
Anjaymon
Kutil
Kutil2
LmaoWth
Sengklek
Kangen

Gambar 13. Penambahan Data pada database

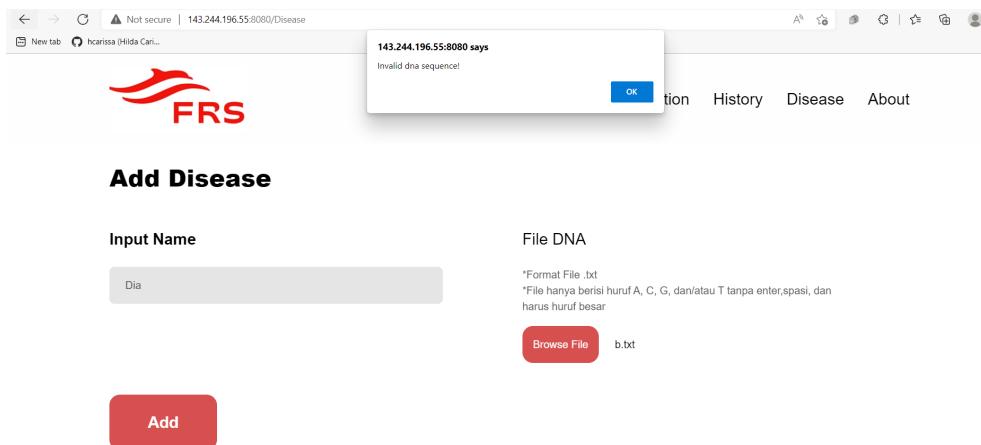
C. Case Tambah Penyakit Baru dengan inputan file salah (berisikan rantai DNA selain ACGT format)

The screenshot shows a web-based application for adding diseases. At the top right, there's a navigation bar with Home, Prediction, History, Disease, and About. Below it, a red header says 'Add Disease'. On the left, there's an input field labeled 'Input Name' containing 'Dia'. On the right, there's a file input field labeled 'File DNA' with a placeholder 'b.txt'. Below the file input, there's a note: '*Format File .txt *File hanya berisi huruf A, C, G, dan/atau T tanpa enter,spasi, dan harus huruf besar'. A red 'Browse File' button is next to the file input. At the bottom left is a red 'Add' button.

Gambar 14. Inputan penyakit baru dengan file yang salah



Gambar 15. Isi file pengujian



Gambar 16. Output yang dikeluarkan program saat inputan salah

D. Case Pencarian Hasil Prediksi Berdasarkan Tanggal (Ada pada Database)

Name	Date Test	Disease	Result
Tepen	22 April 2022	HIV	true
Bono	13 April 2022	Radang Pernapasan	true
Jono	27 April 2022	Gatel Paru Paru	false
Raden	27 April 2022	Borok	true
Yoll	27 April 2022	Albinisme	true
Bebe	27 April 2022	Alergi	false
Yaya	7 April 2015	Anemia	true

Gambar 17. Database hasil prediksi pada saat ini



Gambar 18. Pencarian hasil berdasarkan tanggal dengan format tanggal nama bulan tahun



Home Prediction History Disease About

History

13/04/2022

Search

Name	Date Test	Disease	Result
Bono	13 April 2022	Radang Pernapasan	true

Gambar 19. Pencarian hasil berdasarkan tanggal dengan format tanggal/bulan/tahun

E. Case Pencarian Hasil Prediksi Berdasarkan Nama Penyakit (Ada pada Database)



Home Prediction History Disease About

History

HIV

Search

Name	Date Test	Disease	Result
Tepen	22 April 2022	HIV	true

Gambar 20. Pencarian hasil berdasarkan nama penyakit

F. Case Pencarian Hasil Prediksi Berdasarkan Tanggal dan Nama Penyakit (Ada Pada Database)



Home Prediction History Disease About

History

27 April 2022 Gatel Paru Paru

Search

Name	Date Test	Disease	Result
Jono	27 April 2022	Gatel Paru Paru	false

Gambar 21. Pencarian hasil berdasarkan tanggal dan nama penyakit

G. Case Pencarian Hasil Prediksi Berdasarkan Tanggal (Tidak Ada Pada Database)



Home Prediction History Disease About

History

22 April 2023

Search

Name

Date Test

Disease

Result

Not Found!

Gambar 22. Pencarian hasil berdasarkan tanggal (not found)

H. Case Pencarian Hasil Prediksi Berdasarkan Nama Penyakit (Tidak Ada Pada Database)



Home Prediction History Disease About

History

Lordosis

Search

Name

Date Test

Disease

Result

Not Found!

Gambar 23. Pencarian hasil berdasarkan nama penyakit (not found)

I. Case Pencarian Hasil Prediksi Berdasarkan Tanggal dan Nama Penyakit (Tidak Ada pada Database)



Home Prediction History Disease About

History

30 Januari 2001 HIV

Search

Name

Date Test

Disease

Result

Not Found!

Gambar 24. Pencarian hasil berdasarkan tanggal dan nama penyakit (not found)

J. Case Pencarian Prediction dengan file yang benar (dalam format ACGT) dengan method KMP

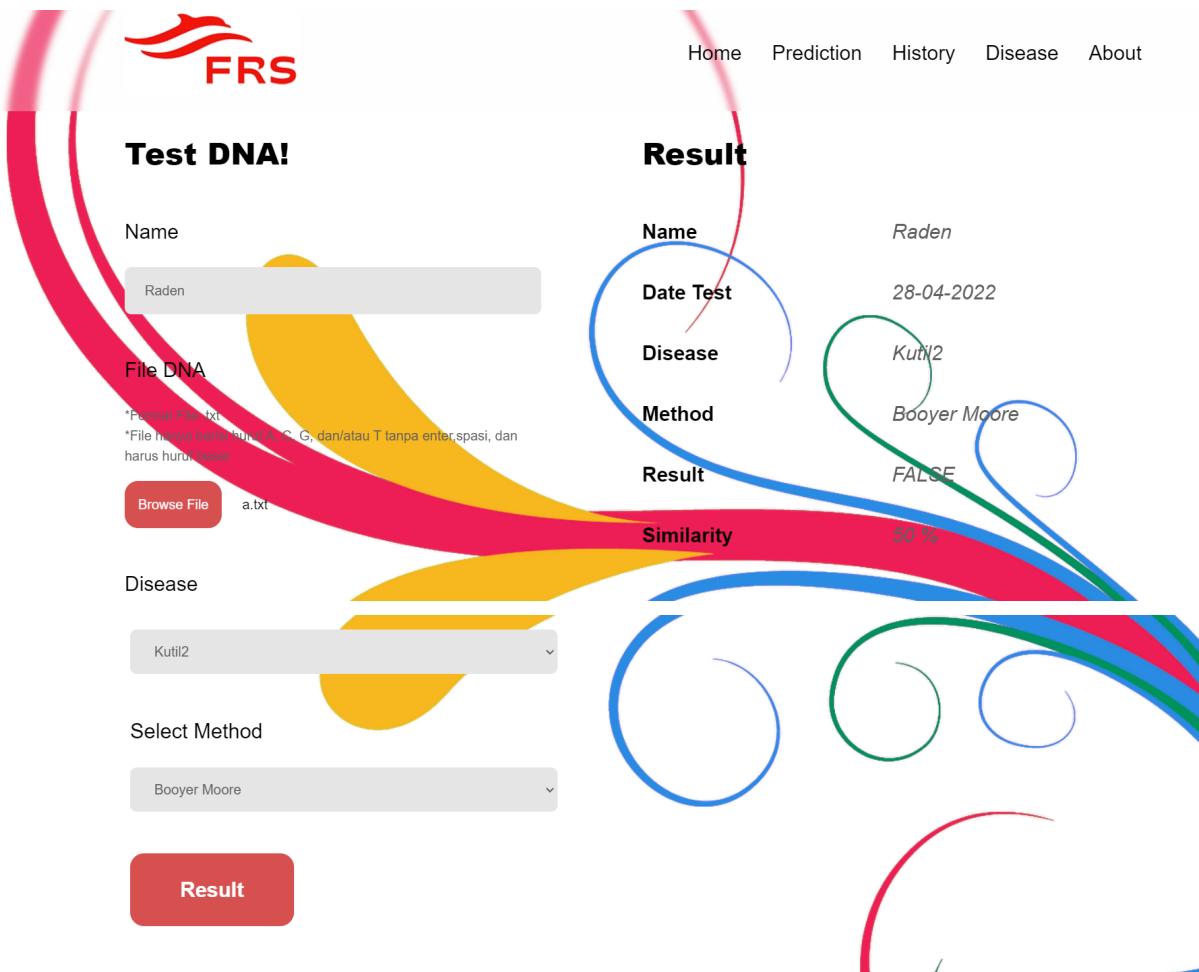
The screenshot shows a web application interface for a DNA test. On the left, there's a form titled "Test DNA!" with fields for "Name" (Yola), "File DNA" (a.txt), "Disease" (Anjaymon), and "Select Method" (KMP). A red "Result" button is at the bottom. On the right, under the heading "Result", is a summary for "Yola" from "28-04-2022". It includes "Disease" (Anjaymon), "Method" (KMP), "Result" (FALSE), and "Similarity" (50%). Below this summary are three circular icons: a blue one, a green one, and a white one.

Gambar 25. Pencarian hasil prediksi dengan metode KMP

<input type="checkbox"/>				14	2022-04-27	tepene	Disentri	false	2022-04-27	NULL
<input type="checkbox"/>				15	2022-04-28	AA	LmaoWth	false	2022-04-28	NULL
<input type="checkbox"/>				16	2022-04-28	Yola	Anjaymon	false	2022-04-28	NULL

Gambar 26. Penambahan database dengan data terbaru

K. Case Pencarian Prediction dengan file yang benar (dalam format ACGT) dengan method Boyer Moore

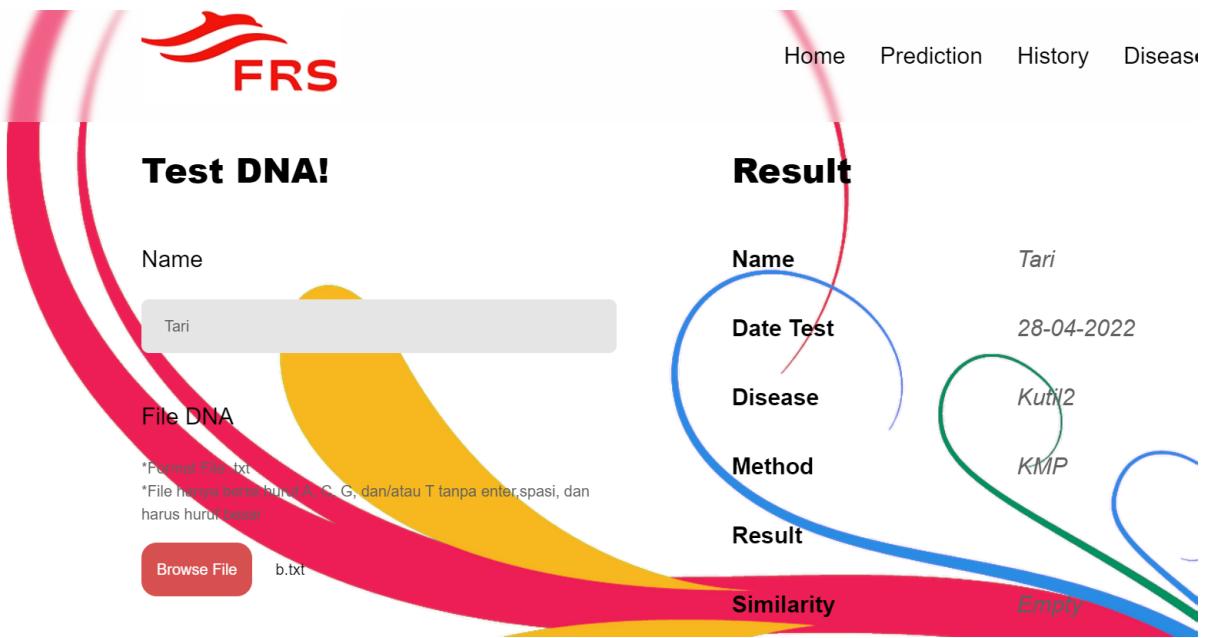


Gambar 27. Pencarian hasil prediksi dengan algoritma booyer moore

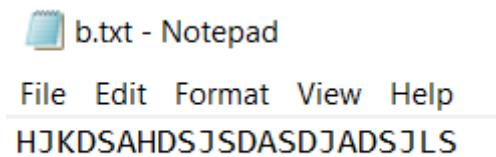
<input type="checkbox"/>				14	2022-04-27	tepene	Disentri	false	2022-04-27	NULL
<input type="checkbox"/>				15	2022-04-28	AA	LmaoWth	false	2022-04-28	NULL
<input type="checkbox"/>				16	2022-04-28	Yola	Anjaymon	false	2022-04-28	NULL
<input type="checkbox"/>				17	2022-04-28	Raden	Kutil2	false	2022-04-28	NULL

Gambar 28. Penambahan database dengan data terbaru

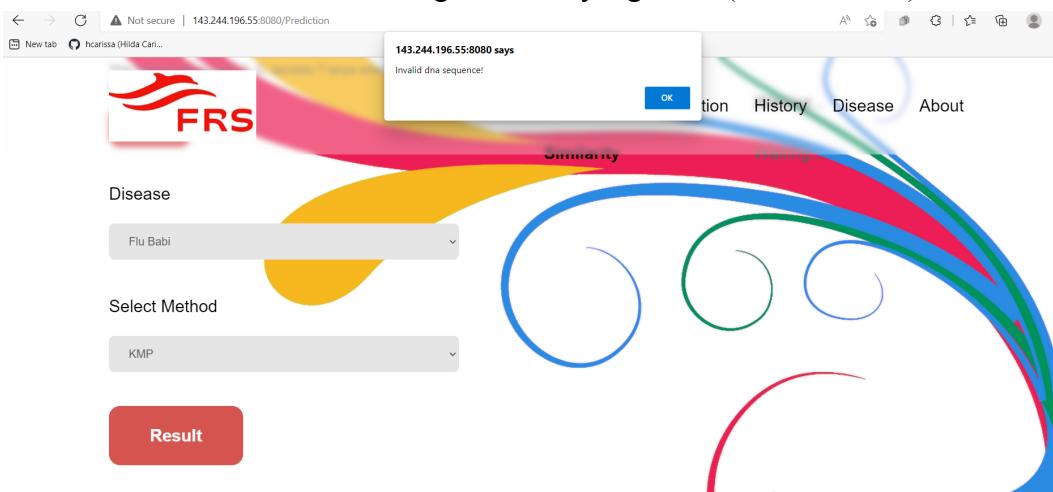
L. Case Pencarian Prediction dengan file yang salah (tidak dalam format ACGT)



Gambar 29. Pencarian hasil prediksi dengan file yang salah



Gambar 30. Isi file dengan format yang salah (bukan AGCT)



Gambar 31. Pencarian hasil prediksi dengan file yang salah

4. Analisis hasil pengujian

Program Website FRS ini, memiliki beberapa fitur, di antaranya:

1. Menambahkan penyakit pada database pada relasi jenispenyakit

Fitur ini digunakan di saat pengguna ingin menginputkan tambahan penyakit baru pada database. dengan cara memasukkan 2 inputan, yaitu berupa nama penyakit yang tidak memiliki constraint apapun asalkan dalam bentuk string dan

sebuah file dalam format .txt (hanya file .txt yang diterima) yang berisikan rantai DNA dengan format ACGT. Berdasarkan pengujian yang telah dilakukan pada bagian nomor 3, dilakukan inputan nama penyakit “Kangen” dan file bernama “a.txt” yang berisikan rantai DNA dalam format ACGT sehingga diterima dan terdaftar pada database dan dapat langsung dilihat penambahannya pada bagian database yang telah di-*display* pada *page website* yang sama. Lalu, pada pengujian berikutnya dilakukan penginputan file dengan nama “b.txt” yang berisikan rantai DNA tidak dalam format ACGT sehingga melanggar constraint dan program akan mengeluarkan pesan “invalid DNA sequence”

2. Melakukan pencarian hasil prediksi berdasarkan masukan tanggal

Fitur ini digunakan pada page history di mana pengguna dapat melakukan pencarian pada database yang ada berdasarkan inputan tanggalnya. Format inputan tanggal yang diterima dalam fitur ini ada 2 yaitu berdasarkan slicing dan nama bulan, misal 13/04/2022 atau 13 April 2022. Pada pengujian yang dilakukan pada nomor 3, dilakukan inputan tanggal 13 April 2022 dan 13/04/2022 yang menampilkan satu hasil pencarian teratas yang terdapat pada database. Dan data yang ditampilkan benar dan sesuai. Lalu, untuk pengujian berikutnya dilakukan pengujian jika inputan tanggal tidak ada pada database, sebagaimana pada pengujian bagian no 3, dilakukan inputan tanggal “22 April 2023” di mana tanggal ini tidak terdapat pada database, maka pada program input yang dikeluarkan adalah message “Not Found”.

3. Melakukan pencarian hasil prediksi berdasarkan masukan nama penyakit

Fitur ini digunakan pada page history di mana pengguna dapat melakukan pencarian pada database yang ada berdasarkan inputan nama penyakitnya. Format inputan nama yang diterima adalah dalam format string. Pada pengujian yang dilakukan pada nomor 3, dilakukan inputan nama penyakit berupa HIV yang menampilkan satu hasil pencarian teratas yang terdapat pada database. Dan data yang ditampilkan benar dan sesuai. Lalu, untuk pengujian berikutnya dilakukan pengujian jika inputan nama penyakit tidak ada pada database, sebagaimana pada pengujian bagian no 3, dilakukan inputan nama penyakit “Lordosis” di mana penyakit ini tidak terdapat pada database, maka pada program input yang dikeluarkan adalah message “Not Found”. Masukan nama penyakit tidak memerhatikan huruf besar atau kecilnya, oleh karena itu untuk misal inputan penyakit ‘HIV’ jika pengguna menginputkan ‘Hiv’ atau ‘Hlv’ sekalipun. maka program masih dapat mendeteksinya sebagai penyakit ‘HIV’

4. Melakukan pencarian hasil prediksi berdasarkan masukan tanggal dan nama penyakit

Fitur ini digunakan pada page history di mana pengguna dapat melakukan pencarian pada database yang ada berdasarkan inputan tanggal dan nama penyakitnya. Format inputan nama yang diterima adalah dalam format string. sedangkan format inputan tanggal yang diterima dalam fitur ini ada 2 yaitu

berdasarkan slicing dan nama bulan, misal 13/04/2022 atau 13 April 2022. Pada pengujian yang dilakukan pada nomor 3, dilakukan inputan tanggal berupa ‘27 April 2022’ dan nama penyakit berupa ‘Gatel Paru Paru’ keduanya dipisahkan dengan menggunakan 1 spasi yang menampilkan satu hasil pencarian teratas yang terdapat pada database. Dan data yang ditampilkan benar dan sesuai. Lalu, untuk pengujian berikutnya dilakukan pengujian jika inputan tanggal yang tidak ada pada database, sebagaimana pada pengujian bagian no 3, dilakukan inputan nama tanggalnya berupa 30 Januari 2001 di mana tidak terdapat pada database, maka pada program input yang dikeluarkan adalah message “Not Found!”. Karena fitur ini memerlukan 2 masukan dan keduanya harus benar jadi baik tanggal maupun nama penyakit yang diinputkan harus merupakan pasangan yang tepat yang terdapat pada database. Jika salah satunya saja salah atau tidak terdapat pada database, maka website akan menampilkan message “Not Found!”

5. Melakukan Tes DNA (main fitur)

Fitur ini digunakan untuk menampilkan hasil prediksi DNA Pattern Matching dengan inputan berupa nama pasien, nama penyakit, file .txt yang berisikan rantai DNA pasien, serta pilihan method yang ingin digunakan. Inputan ini memiliki constraint di antaranya, nama pasien berupa string (varchar), inputan nama penyakit dilakukan dengan pilihan yang sudah disediakan (dimana pilihan ini merupakan daftar penyakit yang terdapat pada database), inputan rantai DNA dari file yang eksensinya .txt yang berisikan text dengan format hanya ACGT, lalu pilihan metode yaitu KMP atau Boyer Moore. Setiap melakukan inputan, hasil inputan akan ter-*display* di bagian kanan website sebagai hasil prediksi, lalu jika menekan tombol submit, maka akan ditampilkan result, yaitu status prediksi dari kecocokan DNA pasien dan DNA penyakit, serta similarity atau tingkat kemiripan antara keduanya. Result dinyatakan true jika hasil kemiripan atau similaritynya $\geq 80\%$, selain itu hasil atau status prediksinya adalah false. Lalu, jika inputan file yang diberikan tidak sesuai, sebagaimana dalam pengujian pada bagian no 3, maka website akan mengeluarkan message “invalid DNA sequence” karena format rantai DNA yang diinputkan tidak dalam format ACGT.

BAB V

Kesimpulan

1. Kesimpulan

Dari tugas besar 3 IF2211 Strategi Algoritma semester 2 2021/2022 berjudul “Penerapan String Matching dan Regular Expression dalam DNA Pattern Matching”, kami berhasil membuat sebuah *website* dengan pemisahan yang jelas antara *backend* dan *frontend*-nya dan sesuai dengan spesifikasi. Adapun materi mata kuliah strategi algoritma yang diterapkan dalam tugas besar kali ini adalah algoritma boyer moore, algoritma KMP, dan juga regex (*regular expression*). Semua materi tersebut berhasil kami terapkan dalam pembuatan *website* ini. Program berhasil dijalankan tanpa bug dan sesuai dengan spesifikasi tugas besar 3 IF2211 yang telah diberikan.

2. Saran

Saran-saran yang dapat kami berikan untuk tugas besar 3 IF2211 Strategi Algoritma semester 2 2021/2022 adalah:

1. Melakukan eksplorasi lebih lanjut mengenai pembuatan *website* terutama dalam penanganan backend dan juga frontend yang digunakan
2. Memperjelas pemberian spesifikasi dan batasan-batasan setiap program pada *file* tugas besar untuk mencegah adanya multitafsir dan kesalahpahaman pada proses pembuatan program.
3. Melakukan eksplorasi lebih lanjut dalam penggunaan algoritma Boyer Moore dan algoritma KMP
4. Melakukan eksplorasi lebih lanjut dalam regex (*regular expression*) untuk berbagai kasus yang memungkinkan

3. Refleksi

Belajar mengimplementasikan algoritma Boyer Moore, algoritma KMP, dan juga regex dalam suatu program merupakan hal yang menarik karena dapat menambah

insight-insight baru. Setelah mengerjakan tugas ini, timbul apresiasi dari kami untuk semua yang telah terlibat dalam pembagian materi ini ataupun tugas besar ini karena melalui tugas ini kami mempelajari banyak hal, salah satunya dalam pembuatan website. Selain itu, dalam mengerjakan tugas ini, kami menyadari banyak kegunaan dan manfaat dari algoritma Boyer Moore, algoritma KMP, serta regex dalam kehidupan sehari-hari yang dapat diterapkan. Hal yang paling penting adalah dalam penggerjaan tugas ini, kami merasa dapat memperbaiki kinerja kami dalam berkelompok mulai dari pembagian tugas ataupun *timeline* untuk mengejar *deadline*. Dengan *timeline* dan pembagian tugas yang jelas ini kami dapat berkontribusi bersama dengan baik dan mengumpulkan tugas dengan baik dan tepat waktu.

LAMPIRAN

Link Video : <https://youtu.be/NXyHl-QA9W4>

Link Repository Github: https://github.com/febryola/Tubes3_13520140

Link Deploy : <http://frs.lightbubble.net/>

DAFTAR PUSTAKA

Slide Kuliah IF2211 Strategi Algoritma Semester 2 Tahun 2021/2022

“Framework backend in node js (express)”, <https://expressjs.com/en/starter/installing.html>

“Framework frontend react”, <https://reactjs.org/docs/getting-started.html#react-for-beginners>