

Tugas Besar IF2211 Strategi Algoritma

Penerapan String Matching dan Regular Expression  
dalam DNA Pattern Matching

Semester II Tahun 2021/2022



Disusun oleh:

Kelompok 16 - GuysNamanyaMauApa?

Ziyad Dhia Rafi 13520064

Vincent Christian Siregar 13520136

Willy Wilsen 13520160

Institut Teknologi Bandung

Sekolah Teknik Elektro dan Informatika

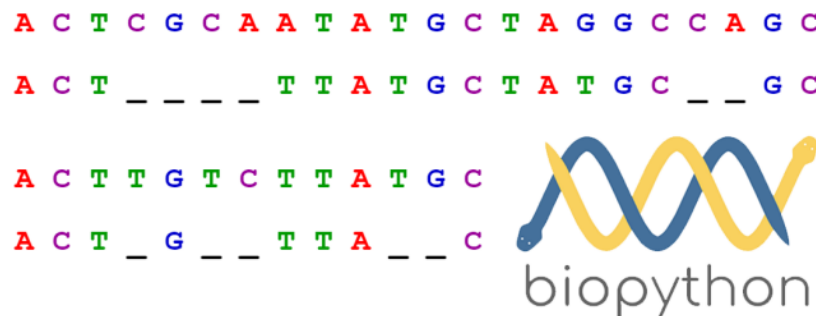
<b>Bab I</b>	<b>3</b>
Latar Belakang	3
Deskripsi	4
Fitur Aplikasi	4
<b>Bab II</b>	<b>10</b>
KMP	10
Booyer-Moore	10
Regex	11
LCS (Longest Common Substring)	12
Website	12
<b>Bab III</b>	<b>13</b>
Langkah-langkah pemecahan masalah	13
Menambahkan penyakit baru	13
Melakukan Tes DNA	13
Untai DNA sampel di tes dengan menggunakan algoritma KMP dan Booyer-Moore dengan unta DNA penyakit. Jika fungsi KMP dan Booyer-Moore menghasilkan true, maka hasil tes dimasukkan ke dalam basis data dengan persentase 100% dan hasil ditampilkan ke layar. Jika false, lanjut ke tahap selanjutnya.	13
Fitur Fungsional dan Arsitektur Program	14
Fitur Fungsional	14
Menambahkan penyakit baru	14
Melakukan Tes DNA	14
Arsitektur Program	14
Frontend	14
Backend	14
<b>Bab IV</b>	<b>15</b>
Spesifikasi Teknis Program	15
Struktur data	15
Penjelasan tata cara penggunaan program	15
Add disease (tambah penyakit)	15
Disease prediction (tes dna)	16
Detail prediction (searching history)	16
Hasil Pengujian	16
Tampilan awal	16
Halaman input penyakit	17
Analisis Hasil Pengujian	24
<b>Bab V</b>	<b>26</b>

# Bab I

## Deskripsi Tugas

### A. Latar Belakang

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.



Gambar 1. Ilustrasi Sekuens DNA

Sumber:

<https://towardsdatascience.com/pairwise-sequence-alignment-using-biopython-d1a9d0ba861f>

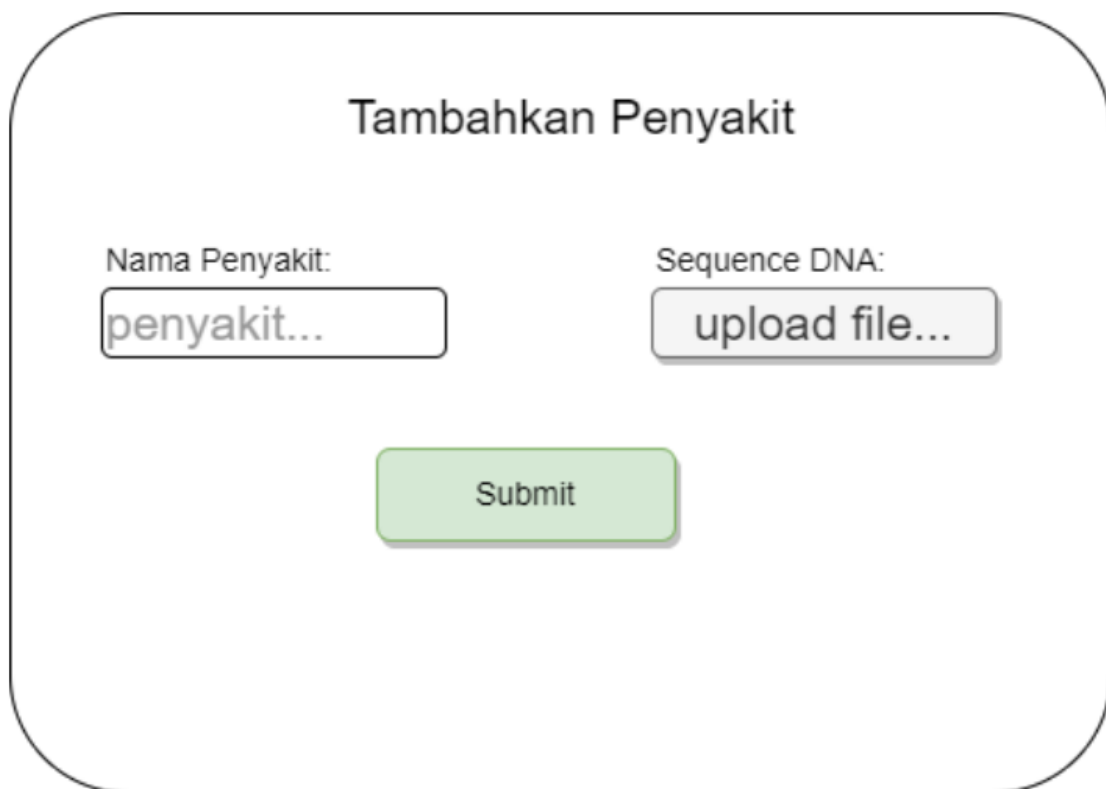
Salah satu jenis tes DNA yang sangat berkaitan dengan dunia bioinformatika adalah DNA sequence analysis. DNA sequence analysis adalah sebuah cara yang dapat digunakan untuk memprediksi berbagai macam penyakit yang tersimpan pada database berdasarkan urutan sekuens DNA-nya. Sebuah sekuens DNA adalah suatu representasi string of nucleotides yang disimpan pada suatu rantai DNA, sebagai contoh: ATTCGTAAGTAAGTTA. 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.

## B. Deskripsi

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.

## C. 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 image shows a web form titled "Tambahkan Penyakit" (Add Disease). It has two input fields: "Nama Penyakit:" (Disease Name) with a text input containing "penyakit..." and "Sequence DNA:" with a file upload button labeled "upload file...". Below these fields is a green "Submit" button.

*Gambar 2. 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:

Sequence DNA:

Prediksi Penyakit:

---

**Hasil Tes**

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

*Gambar 3. 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: , 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.

***Gambar 4. Ilustrasi Interaksi 1***

ii. Masukkan 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 5. Ilustrasi Interaksi 2***

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.

***Gambar 6. Ilustrasi Interaksi 3***

4. (Bonus) 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>

## Bab II

### Landasan Teori

#### A. KMP

Algoritma The Knuth-Morris-Pratt (KMP) adalah sebuah algoritma pattern matching yang mencari pattern dalam text dalam urutan kiri ke kanan seperti algoritma Brute Force. Akan tetapi, algoritma ini melakukan *pattern shift* lebih baik dibanding algoritma brute force.

Kompleksitas Waktu KMP

- Menghitung fungsi pinggiran :  $O(m)$ ,
- Pencarian string :  $O(n)$
- Kompleksitas waktu algoritma KMP adalah  $O(m+n)$ . (sangat cepat dibanding brute force)

Keuntungan KMP

- Algoritma tidak perlu mundur dari text input.
- Algoritma baik untuk memproses file yang sangat besar.

Kerugian KMP

- KMP tidak baik untuk variasi character yang banyak
- Variasi character yang banyak menyebabkan peluang pattern sesuai semakin sedikit

#### B. Boyer-Moore

Algoritma Boyer-Moore berdasarkan dua jenis teknik, yaitu *looking-glass technique* dan *character-jump technique*.

Terdapat tiga kasus yang dapat terjadi:

- Case 1  
Jika P mengandung x, coba geser P ke kanan untuk menyelaraskan kemunculan terakhir x di P dengan  $T[i]$ .
- Case 2  
Jika P berisi x di suatu tempat, tetapi bergeser ke kanan ke yang terakhir kemunculannya tidak mungkin, lalu geser P ke kanan sebanyak 1 karakter ke  $T[i+1]$ .
- Case 3  
Jika kasus 1 dan 2 tidak berlaku, maka geser P untuk meratakan  $P[0]$  dengan  $T[i+1]$ .

Analisis algoritma boyer-moore:

- Boyer-Moore worst case memiliki kompleksitas waktu  $O(nm + A)$
- Boyer-Moore baik untuk variasi karakter banyak (contoh: bahasa inggris) namun buruk untuk variasi sedikit (contoh: binary).
- Boyer-Moore jauh lebih cepat dari brute force untuk searching bahasa inggris.

### C. Regex

Regex adalah singkatan dari regular expression. Regex adalah kumpulan karakter yang menjelaskan pola suatu text. Biasanya, regex digunakan untuk melakukan pencarian pada string atau untuk validasi input.

Beberapa pola regex yang umum digunakan:

A. Bracket ([ ])

#### **brackets [ ] : disjunction**

RE	Match	Example Patterns
/ [wW] oodchuck /	Woodchuck or woodchuck	" <u>W</u> oodchuck"
/ [abc] /	'a', 'b', or 'c'	"In uomini, in sold <u>a</u> ti"
/ [1234567890] /	any digit	"plenty of <u>7</u> to 5"

#### **Brackets [ ] ditambah garis sambung: range**

RE	Match	Example Patterns Matched
/ [A-Z] /	an uppercase letter	"we should call it ' <u>D</u> renched Blossoms'"
/ [a-z] /	a lowercase letter	" <u>m</u> y beans were impatient to be hoed!"
/ [0-9] /	a single digit	"Chapter <u>1</u> : Down the Rabbit Hole"

B. Caret (^)

RE	Match (single characters)	Example Patterns Matched
[^A-Z]	not an uppercase letter	"O <u>y</u> fn pripetchik"
[^Ss]	neither 'S' nor 's'	" <u>I</u> have no exquisite reason for't"
[^\.]	not a period	" <u>o</u> ur resident Djinn"
[e^]	either 'e' or '^'	"look up <u>^</u> now"
a^b	the pattern 'a^b'	"look up <u>a^b</u> now"

C. Tanda tanya (?)

RE	Match	Example Patterns Matched
woodchucks?	woodchuck or woodchucks	<u>“woodchuck”</u>
colou?r	color or colour	<u>“colour”</u>

D. Titik(.)

RE	Match	Example Patterns
/beg.n/	any character between <i>beg</i> and <i>n</i>	<u>begin</u> , <u>beg’n</u> , <u>begun</u>

8

**D. LCS (Longest Common Substring)**

LCS adalah salah satu algoritma pattern matching yang dapat mencari substring terpanjang yang merupakan dari dua buah string. Contohnya pada string “aabbccdd” dan string “xxbccdy” memiliki kesamaan substring terpanjang “bccd”. LCS dapat digunakan untuk mencari persentase kecocokan substring terpanjang pada kedua buah string. Semakin panjang common substring, maka semakin cocok.

**E. Website**

Situs web adalah sekumpulan halaman web yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi (source wikipedia).

Website Tes DNA yang dibuat pada proyek ini merupakan website yang digunakan untuk melakukan tes sampel dna pengguna untuk mendeteksi penyakit bawaan yang diidap oleh pengguna. Website juga dapat menerima input untaian DNA penyakit.

## **Bab III**

### **Analisis Pemecahan Masalah**

#### **A. Langkah-langkah pemecahan masalah**

##### **a. Menambahkan penyakit baru**

Langkah-langkah:

- untai DNA dicek sesuai dengan aturan (Hanya terdiri dari character 'A', 'G', 'T' dan 'C') dengan menggunakan regex.
- Mencari apakah rantai DNA dan nama penyakit sudah terdapat pada basis data atau belum. Jika sudah terdapat pada basis data, berikan pesan penyakit atau untai DNA sudah terdaftar. Jika belum terdaftar, lanjut ke langkah selanjutnya.
- Nama penyakit dan rantai DNA di masukan ke dalam basis data.

##### **b. Melakukan Tes DNA**

Langkah-Langkah:

- Nama Pengguna divalidasi tidak kosong. Jika nama kosong, program menampilkan pesan nama tidak boleh kosong.
- Nama Penyakit dicari pada basis data. Jika penyakit tidak ditemukan, program menampilkan pesan nama penyakit tidak terdaftar.
- untai DNA sampel dicek sesuai dengan aturan (Hanya terdiri dari character 'A', 'G', 'T' dan 'C') dengan menggunakan regex.
- **Untai DNA sampel di tes dengan menggunakan algoritma KMP dan Booyer-Moore dengan untai DNA penyakit. Jika fungsi KMP dan Booyer-Moore menghasilkan true, maka hasil tes dimasukkan ke dalam basis data dengan persentase 100% dan hasil ditampilkan ke layar. Jika false, lanjut ke tahap selanjutnya.**
- Untai DNA sampel di tes menggunakan LCS (Longest Common Substring) untuk mencari persentase kemiripan untai DNA penyakit. Jika hasil diatas 80% maka pengguna divonis mengidap penyakit tersebut dan hasil ditampilkan ke layar. Jika hasil dibawah 80% maka pengguna divonis tidak mengidap penyakit tersebut dan hasil ditampilkan ke layar.

##### **c. Searching History**

Langkah-Langkah:

- Input pengguna diparsing dengan melakukan split berdasarkan karakter spasi (" ").
- Jika input terdiri dari beberapa kata, maka kata pertama di tes dengan regex tanggal. Jika kata pertama merupakan tanggal, data hasil tes yang dilakukan pada tanggal tersebut diambil dari basis data. Data yang memiliki nama penyakit yang sama dengan input kata kedua ditampilkan pada layar.

- Jika input hanya terdiri satu kata, kata di tes dengan menggunakan regex tanggal.
- Jika input merupakan tanggal, data hasil tes yang dilakukan pada tanggal tersebut diambil dari basis data dan ditampilkan pada basis data
- Jika input bukan merupakan tanggal, seluruh data hasil tes diambil dari basis data.
- Setiap nama penyakit dari data yang diambil dan input di tes dengan menggunakan regex searching (nama penyakit ditampilkan ke layar jika mengandung kata yang terdapat pada input).

## **B. Fitur Fungsional dan Arsitektur Program**

### **a. Fitur Fungsional**

- Menambahkan penyakit baru
- Melakukan Tes DNA
- Searching history berdasarkan tanggal
- Searching history berdasarkan nama penyakit
- Searching history berdasarkan tanggal dan nama penyakit

### **b. Arsitektur Program**

Arsitektur program dibagi menjadi frontend dan backend.

#### **i. Frontend**

Frontend adalah istilah yang digunakan untuk komponen website yang berada pada "client-side". Frontend mengatur bagaimana client berinteraksi dengan website. Hal yang diatur oleh front end adalah tampilan visual pada layar dan interaksi client dengan objek pada layar. Framework yang digunakan pada proyek ini adalah ReactJS dengan bahasa pemrograman *Javascript*.

#### **ii. Backend**

Backend adalah istilah yang digunakan untuk komponen website yang berada pada "server-side". Backend mengatur bagaimana data diolah dan disimpan. Pada program ini, backend berguna untuk memproses pattern matching pada DNA dan menyimpan data hasil pemrosesan pattern matching. Bahasa yang digunakan pada proyek ini adalah Golang dengan basis data menggunakan mysql.

## Bab IV

### Implementasi dan pengujian

#### A. Spesifikasi Teknis Program

##### a. Struktur data

Struktur data yang digunakan pada backend adalah sebagai berikut (dalam bahasa go)

- HasilPrediksi

```
type HasilPrediksi struct {
    TanggalPrediksi string
    NamaPasien        string
    PenyakitPrediksi string
    SampelDNA          string
    TingkatKemiripan int
    Status             int
}
```

- Penyakit

```
type Penyakit struct {
    NamaPenyakit string
    DNA           string
}
```

Struktur basis data pada mysql adalah sebagai berikut:

- Tabel hasil prediksi

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation
1	TanggalPred...	DATE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		
2	NamaPasien	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci
3	PenyakitPred...	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci
4	SampelDNA	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci
5	TingkatKemiri...	INT	11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No default		
6	Status	INT	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	No default		

- Tabel jenis penyakit

#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation
1	NamaPenyakit	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci
2	DNA	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		utf8_unicode_ci

##### b. Fungsi dan Prosedur

No.	Fungsi/Prosedur	Penjelasan
1.	Border(string pattern) -> []int	Border function untuk membantu algoritma KMP.
2.	KMP(string pattern, string text) -> bool	Menghasilkan true jika string pattern merupakan substring

		dari string text dengan algoritma KMP
3.	BooyerMoore(string s1,string s2) -> boolean	Menghasilkan true jika string s2 merupakan substring dari string s1 dengan algoritma Booyer-Moore
4.	LCS (string str1, string str2) -> int	Menghasilkan persentase kemiripan dna
5.	Regex(string s1) -> bool	Menghasilkan true jika DNA valid (terdiri dari "A","G","C","T") dengan rumus regex (^[AGTC]*\$)
6.	RegexSearch(string str1, string str2) -> bool	Menghasilkan true jika str1 terdapat pada str2 dengan rumus regex (.* + str1 + `.*`)
7.	RegexTanggal(string str1) -> bool	Menghasilkan true jika str1 merupakan tanggal dengan rumus regex (^[0-9][0-9][0-9][0-9]-[0-9][0-9]?-[0-9][0-9]?\$)

## B. Penjelasan tata cara penggunaan program

Untuk menggunakan website, akses link [tesdnanamanyamauapa.netlify.app](https://tesdnanamanyamauapa.netlify.app).

Pilih salah satu dari 3 fitur yang disediakan pada website:

### a. Add disease (tambah penyakit)

- Masukkan nama penyakit pada kolom nama penyakit
- Masukkan file berisi untaian DNA penyakit (dalam file .txt)
- Tekan tombol tambahkan penyakit

### b. Disease prediction (tes dna)

- Masukkan nama pengguna pada kolom nama pengguna
- Masukkan file berisi untaian DNA sampel (dalam file .txt)
- Masukkan nama penyakit pada kolom nama penyakit
- Tekan tombol submit

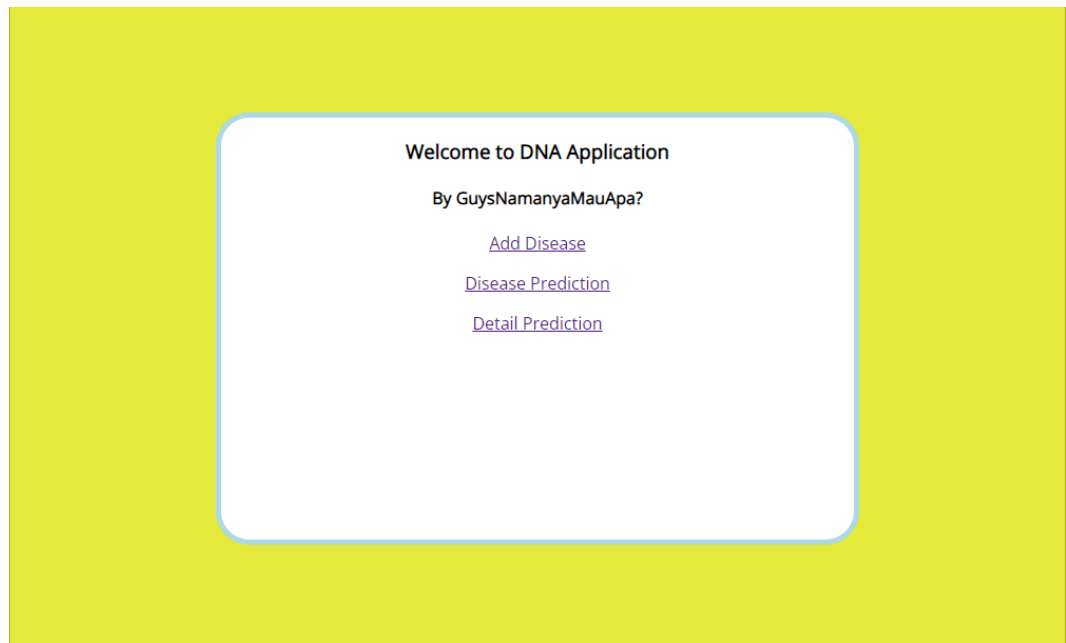
### c. Detail prediction (searching history)

- Masukkan input (dapat berupa tanggal (tahun-bulan-tanggal), nama penyakit, atau tanggal + nama penyakit)
- Tekan tombol process.

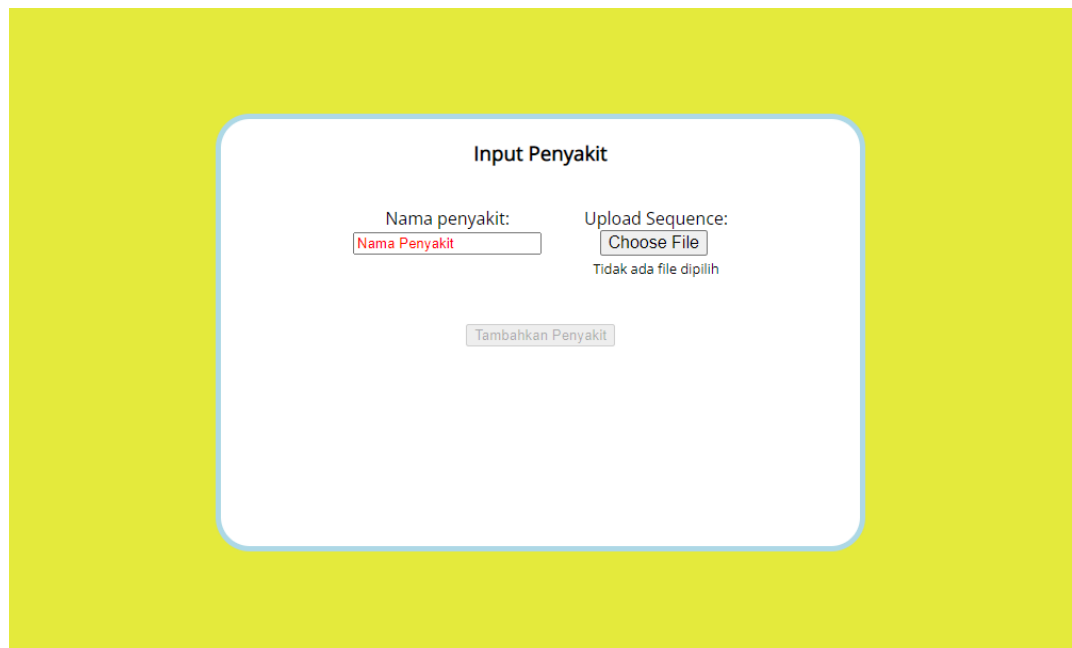


## C. Hasil Pengujian

### a. Tampilan awal



### b. Halaman input penyakit



### c. Halaman input penyakit dengan ekstensi file tidak valid

### Input Penyakit

Nama penyakit:

Corona

Upload Sequence:

Choose File

Ekstensi file tidak valid

Tambahkan Penyakit

- d. Halaman input penyakit dengan untai DNA tidak valid

The screenshot shows a web form titled "Input Penyakit" on a yellow background. The form has two input fields: "Nama penyakit:" with the value "Corona" and "Upload Sequence:" with a "Choose File" button. Below the "Choose File" button, the text "test2.txt" is displayed. A "Tambahkan Penyakit" button is centered below the inputs. At the bottom of the form, the message "Untai DNA tidak valid" is displayed.

- e. Halaman input penyakit dengan untai DNA sudah terdapat pada basis data

The screenshot shows the same "Input Penyakit" form. The "Nama penyakit:" field contains "Corona" and the "Upload Sequence:" field has a "Choose File" button with "Test.txt" listed below it. The "Tambahkan Penyakit" button is present. At the bottom of the form, the message "untai DNA peyakit sudah ada dalam database" is displayed.

- f. Halaman input penyakit dengan nama penyakit sudah terdapat pada basis data

Input Penyakit

Nama penyakit:

HIV

Upload Sequence:

Choose File

Test.txt

Tambahkan Penyakit

Nama penyakit sudah ada dalam database

**g. Halaman input penyakit berhasil**

Input Penyakit

Nama penyakit:

Corona

Upload Sequence:

Choose File

Test.txt

Tambahkan Penyakit

Penyakit berhasil ditambahkan ke dalam database

#### h. Halaman tes DNA

### Tes DNA

Nama Pengguna:

Sequence DNA:  

Choose File No file chosen

Prediksi Penyakit:

Submit

### Hasil

#### i. Halaman tes DNA dengan nama kosong

### Tes DNA

Nama Pengguna:

Sequence DNA:  

Choose File Test.txt

Prediksi Penyakit:

Submit

### Hasil

Nama Tidak Boleh Kosong

j. Halaman tes DNA dengan DNA tidak valid

### Tes DNA

Nama Pengguna:

Sequence DNA:  

Choose File

 test2.txt

Prediksi Penyakit:

Submit

### Hasil

DNA Tidak Valid!

k. Halaman tes DNA dengan validasi dibawah 80%

### Tes DNA

Nama Pengguna:

Sequence DNA:  

Choose File

 Mantap.txt

Prediksi Penyakit:

Submit

### Hasil

2022/4/28 - Budi - 15% - Corona - False

**l. Halaman tes DNA dengan validasi diatas 80%**



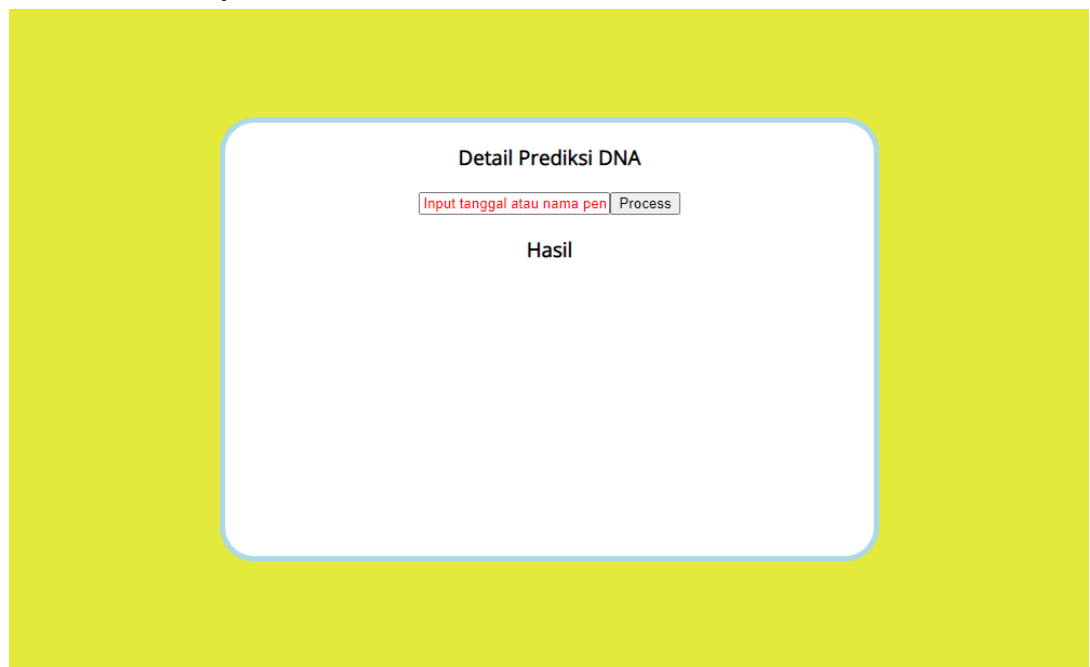
**Tes DNA**

Nama Pengguna:  Sequence DNA:  Test.txt Prediksi Penyakit:

**Hasil**

2022/4/28 - Ani - 100% - Corona - True

**m. Halaman detail prediksi**



**Detail Prediksi DNA**

**Hasil**

**n. Halaman detail prediksi input tanggal**

Detail Prediksi DNA

2022-04-27

Process

Hasil

1. 2022-04-27 - Udin - Pilek - 100% - True

2. 2022-04-27 - Udin - Pilek - 38% - False

3. 2022-04-27 - Vincent - HIV - 25% - False

o. Halaman detail prediksi input nama penyakit full

Detail Prediksi DNA

HIV

Process

Hasil

1. 2022-04-27 - Vincent - HIV - 25% - False



**p. Halaman detail prediksi nama penyakit sebagian**

The screenshot shows a web interface titled "Detail Prediksi DNA". It features a search bar with the text "pil" and a "Process" button. Below the search bar, the word "Hasil" is displayed. There are two rounded rectangular boxes containing prediction results:

- 1. 2022-04-27 - Udin - Pilek - 100% - True
- 2. 2022-04-27 - Udin - Pilek - 38% - False

**q. Halaman detail prediksi tanggal plus nama penyakit**

The screenshot shows a web interface titled "Detail Prediksi DNA". It features a search bar with the text "2022-04-28 Corona" and a "Process" button. Below the search bar, the word "Hasil" is displayed. There are two rounded rectangular boxes containing prediction results:

- 1. 2022-04-28 - Ani - Corona - 100% - True
- 2. 2022-04-28 - Budi - Corona - 15% - False

**D. Analisis Hasil Pengujian**

Dari pengujian diatas, dapat dilihat setiap fitur berjalan dengan yang diharapkan dan dapat menyelesaikan permasalahan. Fitur tambahkan penyakit dapat menambahkan penyakit dengan validasi untai DNA dan dipastikan tidak ada nama penyakit dan untai DNA yang terduplikasi.

Fitur tes DNA dapat melakukan tes DNA dengan validasi input untai DNA. Tes DNA juga dapat memberikan persentase kemiripan dengan menggunakan algoritma Longest Common Substring (LCS) yang di determinasi dari panjang LCS/panjang dna penyakit.

Fitur melihat detail prediksi dapat menerima input tanggal, nama penyakit, dan gabungan tanggal penyakit. Input dikenali oleh regex dengan baik dan dapat membedakan input tanggal ataupun input nama penyakit.

## Bab V

### Kesimpulan dan Saran

#### Kesimpulan

Kesimpulan yang dapat diambil dari implementasi berbagai algoritma pattern matching dan regex adalah algoritma-algoritma tersebut memiliki kelebihan masing-masing. Dengan memahami persoalan secara mendalam terlebih dahulu, kita bisa menentukan algoritma yang paling baik digunakan. KMP berjalan lebih efektif saat variasi karakternya sedikit, sedangkan Boyer Moore berjalan lebih efektif saat variasi karakternya banyak. Sementara itu, regular expression atau Regex berguna untuk pencarian, apabila hasil yang ingin dicari bukanlah hasil eksak, melainkan dapat berupa berbagai variasi. Terdapat juga algoritma seperti LCS yang dapat memberikan persentase kecocokan pattern dengan parameter tertentu.

#### Saran

Berdasarkan hasil kerja kelompok kami, terdapat berbagai saran yang dapat digunakan untuk membantu pengimplementasian algoritma pattern matching dan regex, maupun dalam pembuatan aplikasi berbasis web.

1. Diperlukan pemahaman yang baik terkait konsep Cross Origin Resource Sharing (CORS) sehingga mempermudah pembuatan aplikasi berbasis web
2. Lebih banyak membaca dokumentasi terkait bahasa Go termasuk library yang digunakan, karena merupakan bahasa baru sehingga dapat membantu proses pengerjaan lebih baik.

Video demo program dapat diakses pada link berikut:

<https://youtu.be/DDSMaUr35ko>

Source code program dapat diakses pada link berikut:

[https://github.com/ziyaddr/Tubes3\\_13520064](https://github.com/ziyaddr/Tubes3_13520064)

Link Deploy:

<http://tesdnanamanyamauapa.netlify.app> (Frontend)

<https://tesdna.herokuapp.com/> (Backend)