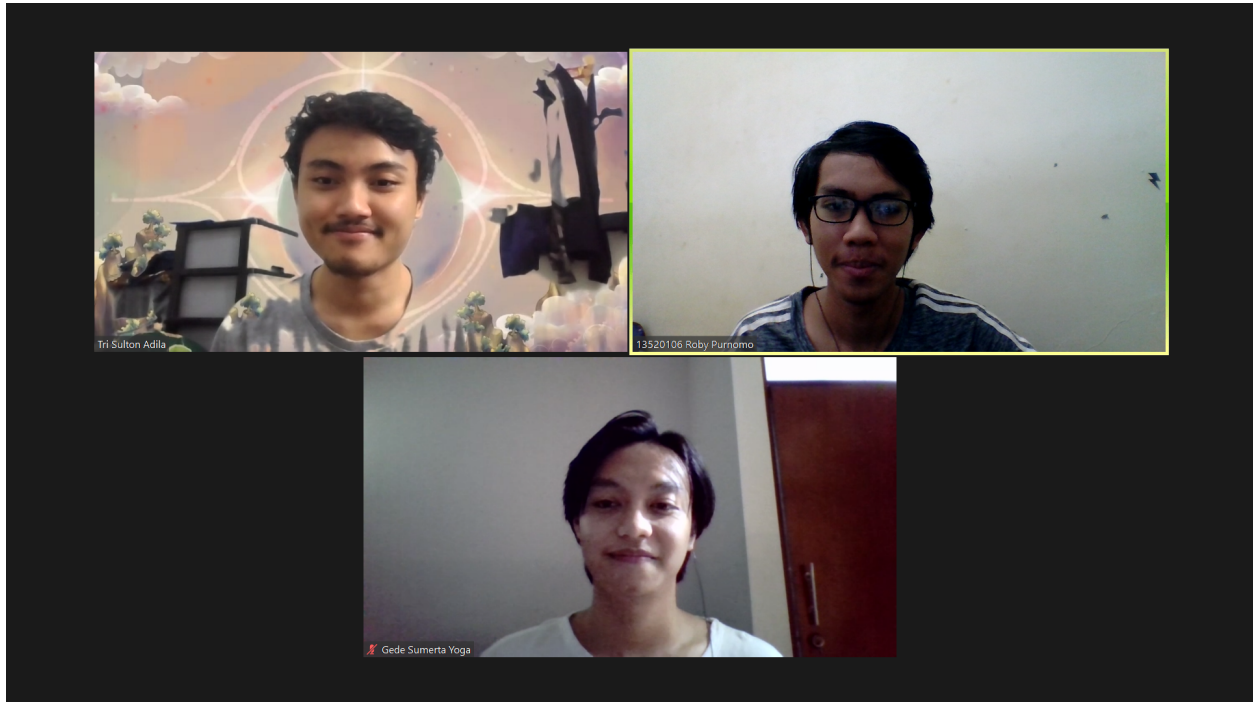


# **Laporan Tugas Besar 3**

## **IF2211 Strategi Algoritma**

### **Penerapan String Matching dan Regular Expression dalam DNA Pattern Matching**



Disusun oleh:  
Kelompok Nyari Penyakit

Gede Sumerta Yoga	(13520021)
Tri Sulton Adila	(13520033)
Roby Purnomo	(13520106)

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**2022**

## **BAB 1**

### **DESKRIPSI TUGAS**

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

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

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

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:

**Tambahkan Penyakit**

Nama Penyakit:

Sequence DNA:

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>**

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.
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:

Sequence DNA:

Prediksi Penyakit:

---

**Hasil Tes**

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

#### Spesifikasi Program:

1. Aplikasi berbasis website dengan pembagian Frontend dan Backend yang jelas.
2. Implementasi Backend wajib menggunakan Node.js / Golang, sedangkan Frontend disarankan untuk menggunakan React / Next.js / Vue / Angular. Lihat referensi untuk selengkapnya.
3. Penyimpanan data wajib menggunakan basis data (MySQL / PostgreSQL / MongoDB).
4. Algoritma pencocokan string (KMP dan Boyer-Moore) wajib diimplementasikan pada sisi Backend aplikasi.
5. Informasi yang wajib disimpan pada basis data:
  - a. Jenis Penyakit: - Nama penyakit - Rantai DNA penyusun.
  - b. Hasil Prediksi: - Tanggal prediksi - Nama pasien - Penyakit prediksi - Status terprediksi.
6. Jika mengerjakan bonus tingkat kemiripan DNA, simpan hasil tingkat kemiripan tersebut pada basis data.

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1. Deskripsi Singkat Algoritma BM, KMP, dan Regex**

##### **2.1.1. Algoritma Boyer-Moore**

Algoritma Boyer-Moore ini merupakan salah satu algoritma pencocokan atau pencarian string yang terkenal saat ini. Algoritma ini ditemukan oleh Bob Boyer dan J. Strother Moore. Algoritma ini berdasar dari dua teknik, yaitu:

1. The looking-glass technique

Teknik ini akan mencari pola/pattern pada text secara terbalik dari bagian akhir text.

2. The character-jump technique

Ketika terjadi perbedaan pada indeks ke- $i$  dari  $T(\text{text})$  dengan karakter di pattern  $P$  pada indeks ke- $j$  terdapat tiga kemungkinan yang akan dilakukan. Pertama, jika dalam  $P$  terkandung karakter yang sesuai dengan  $T[i]$  maka akan dilakukan pergeseran ke kanan sehingga kemunculan terakhir dari karakter tersebut sejajar dengan  $T[i]$ . Kedua, jika dalam  $P$  terdapat karakter yang sesuai dengan  $T[i]$  tetapi tidak dimungkinkan ke pergeseran kanan sehingga sejajar, maka geser pattern  $P$  ke kanan satu langkah. Ketiga, jika kasus satu dan dua tidak ada yang memenuhi maka sejajarkan  $P[0]$  dengan  $T[i+1]$

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

Algoritma Knuth-Morris-Pratt adalah salah satu algoritma pencocokan string atau pencarian string yang terkenal saat ini. Algoritma ini ditemukan oleh Donald E. Knuth, James H. Morris dan Vaughan R. Pratt. Algoritma ini akan bekerja dengan mencocokkan pattern dari kiri ke kanan seperti teknik brute force. Namun, yang membedakan adalah pergeseran ketika ditemukan ketidakcocokan. Dalam hal ini digunakan fungsi pinggiran atau border function sebagai tolak ukur. Fungsi Pinggiran ini akan memproses prefix terpanjang yang juga merupakan suffix terpanjang. Dengan fungsi ini, pergeseran akan ditentukan dengan memasukkan indeks pada pattern yang tidak cocok saat pencocokan.

### **2.1.3. Regular Expression**

Regular expression merupakan sebuah notasi yang dapat digunakan untuk mendeskripsikan pola pada suatu string atau text. Konsep tentang regex atau regular expression ini dicetuskan oleh Stephen Cole Kleene pada tahun 1951. Regex ini sering digunakan dalam hal pencocokan, pencarian, dan manipulasi teks.

## **2.2. Deskripsi Singkat Aplikasi Web**

Web pada kelompok kami dibuat menggunakan framework sebagai berikut :

- Frontend : React JS
- Backend : Node JS
- Database : MySQL

Web ini terdiri atas 3 fitur menu yakni menu pertama adalah menu untuk menambah database penyakit, menu kedua adalah untuk melakukan tes sequence dna, dan menu ketiga adalah untuk menampilkan riwayat dari hasil tes dna yang telah dilakukan. Web ini menggunakan arsitektur client server. Server digunakan sebagai backend yang berhubungan dengan database dan client sebagai frontend yang berhubungan dengan user.

## **BAB 3**

### **ANALISIS PEMECAHAN MASALAH**

#### **3.1 Langkah Penyelesaian Masalah**

- Fitur menambah data penyakit

Pada fitur ini hasil yang diharapkan adalah menerima masukan berupa nama penyakit dan sequence dna untuk penyakit tersebut, lalu memasukkannya pada database. Pertama-tama dibuat form untuk menerima masukan nama penyakit dengan diketik dan menerima masukan sequence dna berupa file txt dari upload file. Lalu akan dilakukan pengecekan pada masukan-masukan ini terutama pada sequence dna apakah memenuhi format dna yakni hanya berisi karakter A/C/G/T saja. Jika masukan valid, maka akan dilakukan insert pada database dengan value sesuai dengan masukan dan jika tidak maka akan mengeluarkan pesan error.

- Fitur Tes DNA

Pada tes DNA, masukan yang diharapkan adalah file yang berisi untaian rantai DNA, nama pengguna, nama penyakit, dan pilihan algoritma pattern matching yang ingin digunakan. Masukan pengguna berupa file akan dicek menggunakan regex apakah memenuhi syarat, seperti tidak ada huruf kecil, huruf yang ada hanya A,C,G,T, dan tidak ada spasi. Kemudian setelah menerima nama pengguna dan nama penyakit maka akan dicari pattern penyakit yang sesuai dengan nama penyakit masukan di database. Jika penyakit tidak ditemukan di database maka hasil pengecekan akan Not Found. Kemudian jika penyakit ada di database, maka akan diambil pattern DNA-nya dan kemudian dicek menggunakan algoritma pilihan baik BM ataupun KMP. Hasilnya akan ditampilkan di layar dengan format {Tanggal-NamaPengguna-NamaPenyakit-True/False}.

- Fitur Menampilkan Pencarian Riwayat Tes DNA

Data hasil tes DNA beserta data penyakit yang berada di database akan dijoin dan disimpan dalam struktur data list agar dapat menampilkan riwayat pengecekan penyakit. Mula mula akan disediakan sebuah input berupa teks yang menerima masukan dari pengguna. Melalui masukan pengguna, akan dilakukan filtering terhadap list dari database. Terdapat dua cara yang diterapkan untuk dapat melakukan filtering. Filtering pertama adalah aplikasi akan menampilkan data riwayat yang sesuai dengan urutan huruf



masukan pengguna. Filtering kedua adalah dengan menggunakan regex. Filtering regex hanya dilakukan pada tanggal pengecekan dan nama penyakit. Data riwayat yang akan muncul adalah data yang memiliki nama penyakit atau tanggal yang huruf-hurufnya terdapat pada masukan pengguna. Misalkan terdapat riwayat pemeriksaan penyakit cholera di database. Dengan menggunakan regex, meskipun masukan pengguna adalah chalero, aplikasi tetap akan memunculkan riwayat dari data tersebut. Hal ini tidak berlaku apabila masukan pengguna adalah mencari nama orang yang pernah melakukan pemeriksaan. Hal ini disebabkan fitur regex tidak diterapkan pada pencarian nama.

### **3.2. Fitur Fungsional dan Arsitektur Aplikasi Web**

Fitur fungsional aplikasi web ini adalah sebagai berikut.

1. Aplikasi dapat menerima input sequence DNA dalam bentuk file dengan melalui pemeriksaan menggunakan regex. Hasil input akan dimasukkan ke dalam database.
2. Aplikasi dapat memprediksi seseorang apakah menderita penyakit tertentu atau tidak berdasarkan input file sequence DNA yang dimiliki. Pemeriksaan keabsahan input juga menggunakan regex. Pencocokan pattern sequence DNA dapat menggunakan algoritma KMP serta Boyer-Moore.
3. Aplikasi dapat menampilkan hasil riwayat pengecekan penyakit dengan menerima masukan tanggal dan nama.

Komponen arsitektur aplikasi web dasar di antaranya adalah sebagai berikut.

1. Client : melakukan request atau mengakses halaman, dalam hal ini adalah web browser
2. Web server : melakukan dukungan protokol web agar dapat memproses permintaan client
3. Database server : menyediakan data dalam bentuk terstruktur
4. Content management server : menangani content untuk melayani aplikasi
5. Aplikasi server : melayani permintaan akses dari client

Konsep pemrograman three-tier atau yang biasa disebut client server programming digunakan pada aplikasi ini. Gambaran secara umum tiga fungsionalitas sistem tersebut adalah sebagai berikut.

- Komponen klien : berjalan di atas local komputer user (tier 1). Akses tidak bisa sampai pada RDBMS jika tidak ada koneksi pada aplikasi server.

- Aplikasi server : berjalan pada remoter server yang berfungsi sebagai koneksi antara klien dengan database system (tier 2).
- Aplikasi RDBMS : kumpulan database (tier 3). Tidak bisa diakses secara langsung dari tier 1 jika tidak terkoneksi dengan tier 2.

## **BAB 4**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1. Spesifikasi Teknis Program**

- **Fitur Menambah Data Penyakit**

Struktur data yang digunakan untuk kelas ini hanyalah reader yang digunakan untuk membaca detail dari file input dari upload file.

Fungsi atau prosedur yang digunakan untuk kelas ini adalah sebagai berikut:

- `handleDiseaseChange` : prosedur untuk menghandle event pada perubahan input nama penyakit
- `handleSubmit` : prosedur untuk menghandle event pada pengeklikan tombol submit, berisi juga handle untuk melakukan post request insert into database
- `onFileChange` : prosedur untuk menghandle event pada perubahan input upload file txt
- `fileData` : fungsi untuk menampilkan detail dari file tentang masukan dari upload file txt
- `isValid` : fungsi yang digunakan untuk mengecek apakah input sequence dna apakah valid atau tidak

- **Fitur Tes DNA**

Struktur data yang digunakan untuk komponen FileUploader adalah file, penyakit, pengguna, hasil, dan radio.

Fungsi atau prosedur yang digunakan untuk komponen FileUploader adalah sebagai berikut:

- `setFile(file)` : prosedur untuk mengubah file
- `setPenyakit(penyakit)` : prosedur untuk mengubah variabel penyakit
- `setPengguna`: prosedur untuk mengubah variabel pengguna
- `setHasil`: prosedur untuk mengubah variabel hasil
- `setRadio`: prosedur untuk mengubah variabel radio
- `onFileChange` : prosedur yang dipanggil saat terdapat perubahan file yang diupload

- onPenyakitChange: prosedur yang dipanggil saat terdapat perubahan nama penyakit yang dimasukkan
- onPenggunaChange: prosedur yang dipanggil saat terdapat perubahan nama pengguna yang dimasukkan.
- onSubmit: prosedur yang dipanggil ketika tombol submit ditekan yang akan memberikan data masukan ke server dan menerima hasil pengecekan oleh server.

Fungsi atau prosedur yang digunakan dalam server terkait tes DNA:

- upload : prosedur ini akan menyimpan file yang telah diupload ke website
  - border(pattern) : fungsi yang menghasilkan sebuah list yang isinya merupakan nilai dari setiap fungsi pinggiran yang terdapat dalam pattern.
  - kmpMatching(text, pattern) : fungsi yang melakukan algoritma KMP
  - lastOccurence(pattern) : fungsi yang menghasilkan list indeks yang menandakan kemunculan terakhir setiap karakter pada pattern
  - bmMatching : fungsi yang melakukan algoritma BM
- Fitur menampilkan riwayat tes dna

Struktur data yang digunakan adalah list untuk menyimpan hasil fetch dari database, yaitu riwayatList dan filteredList untuk menyimpan data yang telah melalui proses filtering.

Adapun fungsi dan prosedur yang dibangun adalah sebagai berikut.

- setRiwayatList(riwayatList) : prosedur untuk mengubah variabel riwayatList
- setSearchInput(searchInput) : prosedur untuk mengubah variabel searchInput
- setFilteredResults(filteredResults) : prosedur untuk mengubah variabel filteredResults
- searchItems(searchValue) : prosedur ini bertanggung jawab untuk melakukan filtering terhadap RiwayatList berdasarkan masukan parameter searchValue dari pengguna.
- checkRegex(dataItem, strInput) : memeriksa dataItem yaitu sebuah data riwayat dengan regex dari stringInput

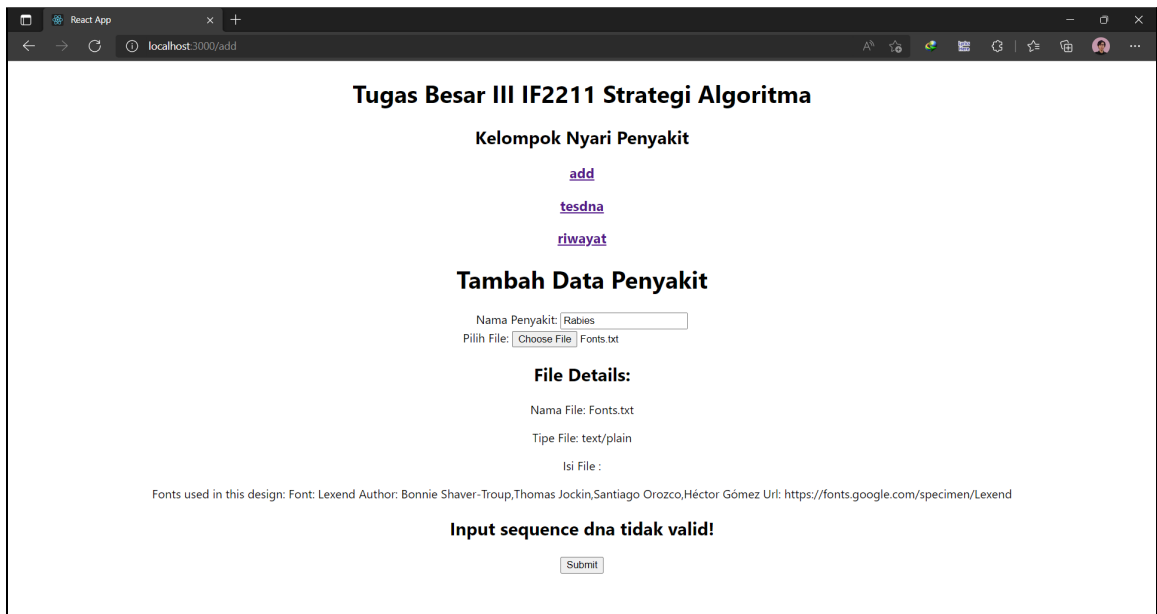
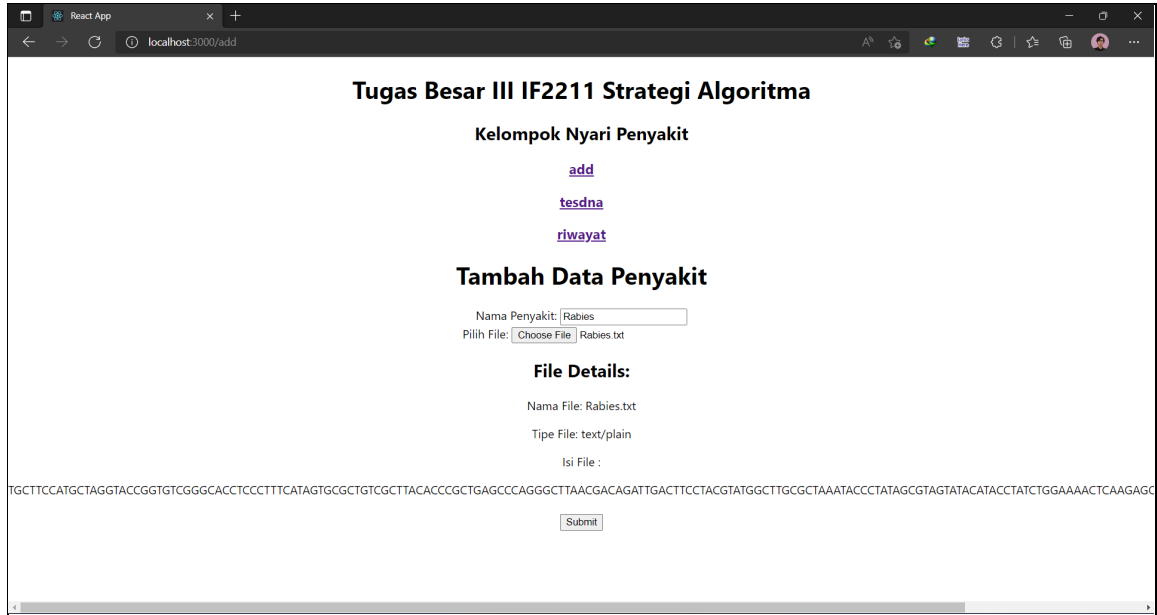
## 4.2. Cara Penggunaan Program

Untuk cara penggunaan webnya dapat dilihat pada readme repository.

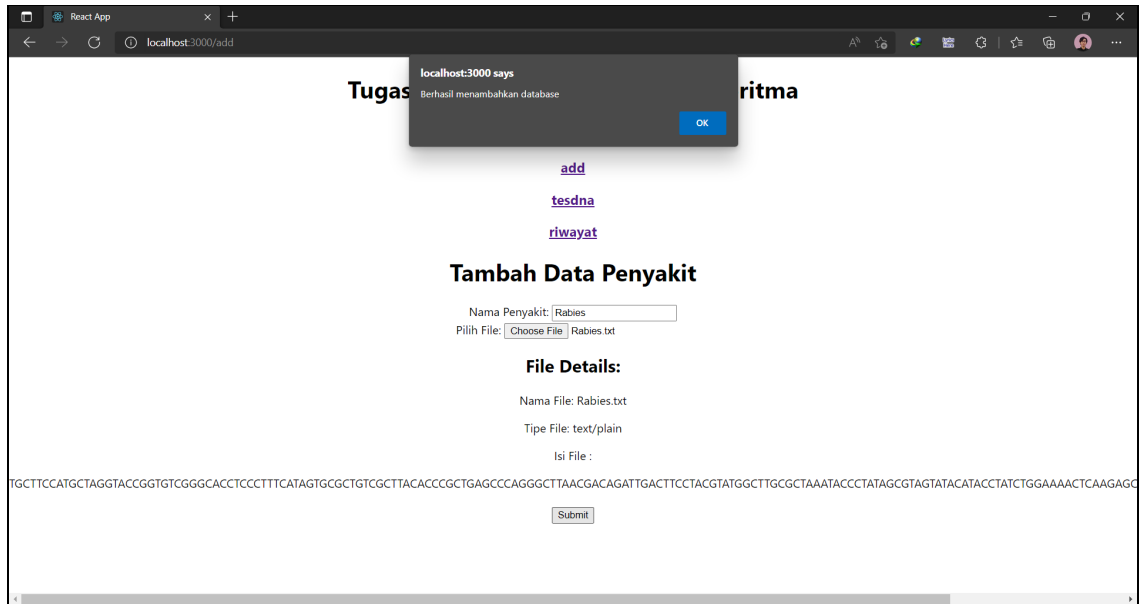
- Fitur menambah data penyakit
  1. Pilih menu add
  2. Pengguna dapat memasukkan input berupa nama penyakit dan upload file berupa txt
  3. Jika masukan telah valid, lalu pengguna menekan tombol submit, maka database akan terupdate berdasarkan entry penyakit baru sesuai masukan pengguna
- Fitur Tes DNA
  1. Pilih menu tesdna
  2. Pengguna memasukkan file yang berisi rantai DNA, nama pengguna, dan nama penyakit
  3. Jika masukan file valid sesuai regex dan nama penyakit terdapat di database maka akan dilakukan pengecekan dan akan keluar hasilnya True atau False. Namun jika salah satunya tidak terpenuhi maka akan muncul keterangan yang berbeda
- Fitur menampilkan riwayat tes dna
  1. Pilih menu riwayat
  2. Pengguna dapat memasukkan input berupa tanggal, nama orang, dan nama penyakit pada kolom input.
  3. Hasil pencarian akan muncul di bawah kolom input.

## 4.3. Hasil Testing

- Fitur Menambah Data Penyakit



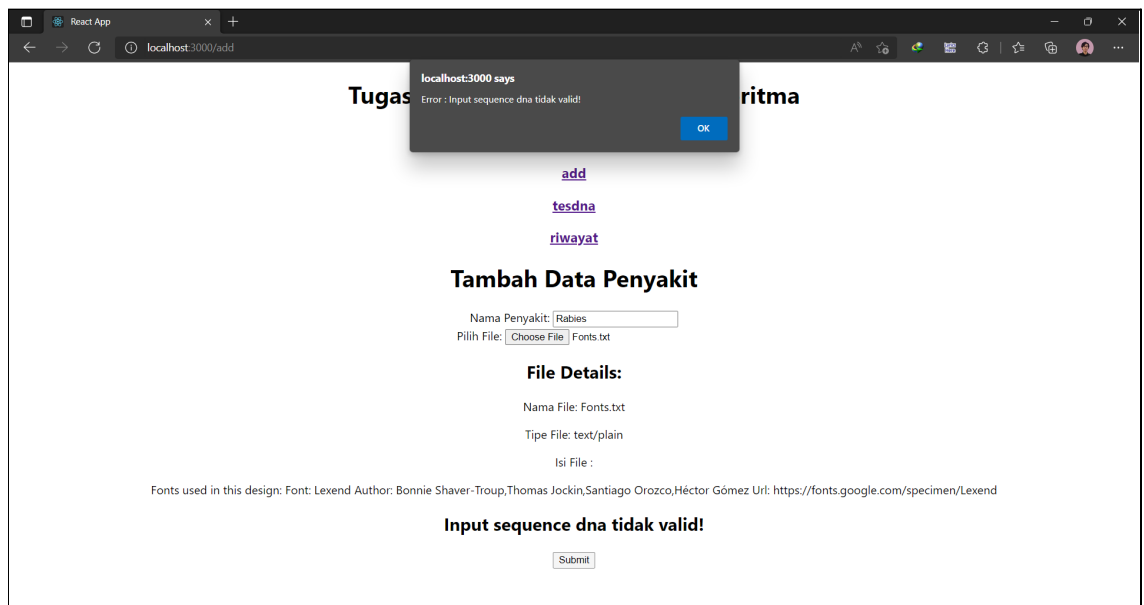
Gambar di atas menunjukkan jika masukan yang diterima sesuai dengan yang diharapkan, yaitu file sequence dna nya berisi sequence dna yang valid. Sedangkan untuk gambar kedua menunjukkan sebaliknya, jika masukan tidak valid. Sehingga jika ditekan submit untuk kedua case akan menghasilkan pesan sebagai berikut. Dan juga terlihat bahwa pada database telah mengalami perubahan, yakni penambahan entry untuk masukan 'Rabies' dengan sequence DNA-nya.



```

MariaDB [dna]> select * from dna_disease where nama = 'Rabies';
+-----+-----+-----+-----+-----+-----+
| id | nama | sequence_dna |
+-----+-----+-----+-----+
| 260 | Rabies | TGCTTCATGCTAGGTACCGGTGTCGGGCACCTCCCTTTCATAGTGCCTGTCGCTTACACCCGCTGAGCCAGGGCTTAACGACAGATTGACTTCTACGTA |
|    |      | TGCGTTGCGCTAAATACCCCTATAGCGTAGTATACATACCTATCTGGAAAACTCAAGAGCGCCCAACCGTGTGGCAGAACTCTTCGTC |
+-----+-----+-----+-----+
1 row in set (0.001 sec)

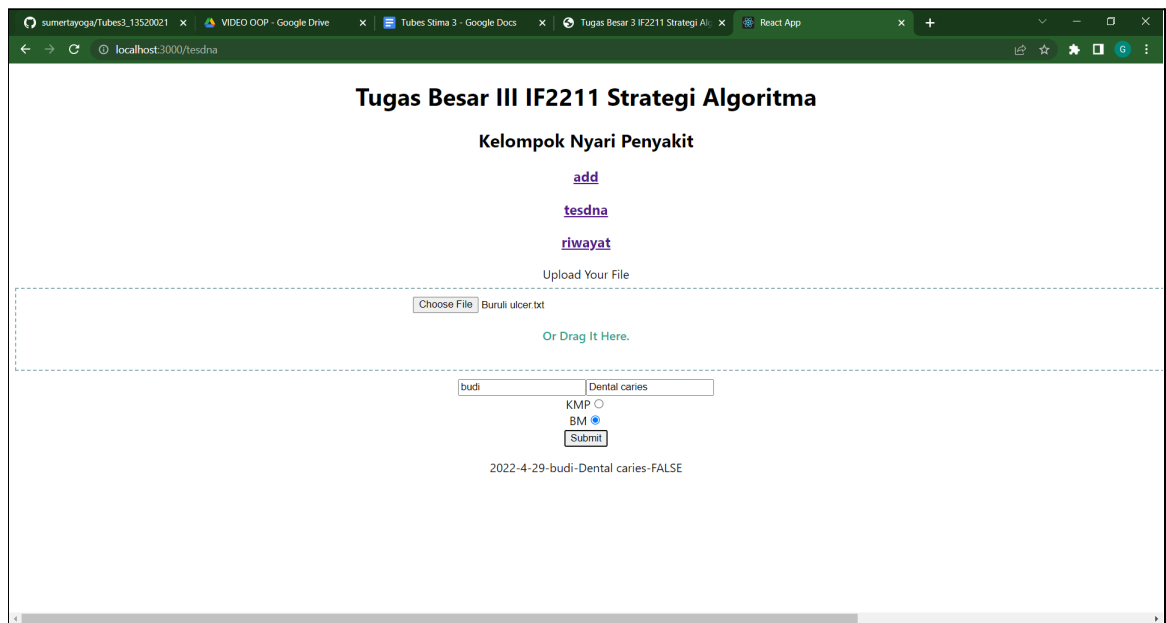
```



- Fitur Tes DNA



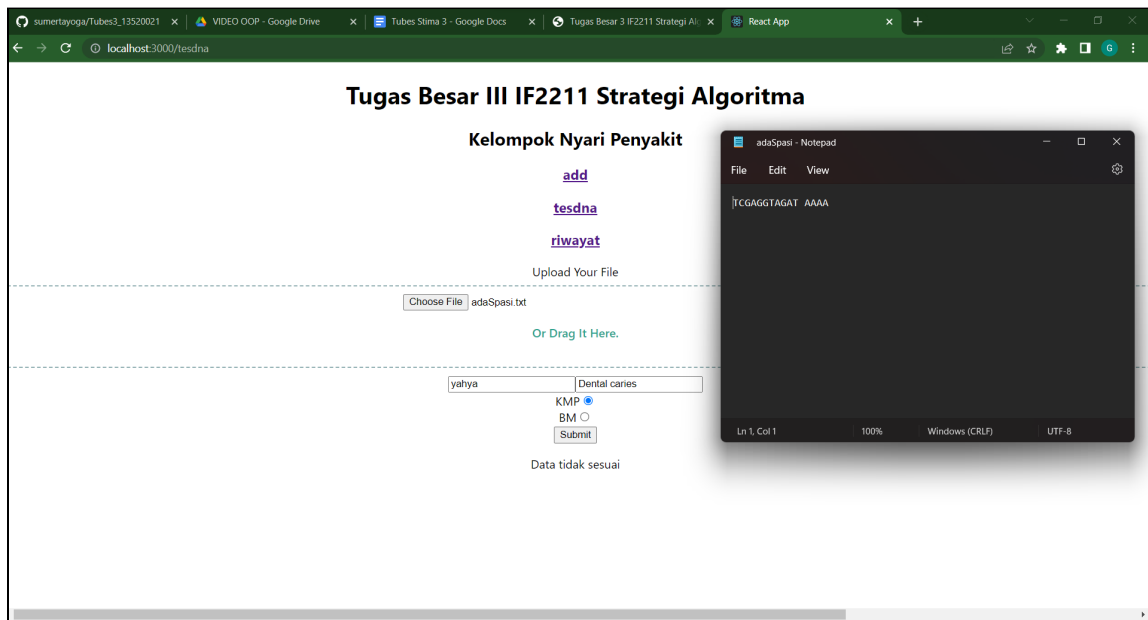
Gambar di atas merupakan pengujian fitur tes dna dengan menggunakan file test case yang sesuai dengan nama penyakitnya. Dapat dilihat disana terdapat file Dental caries.txt yang didalamnya terdapat pattern dari penyakit Dental caries. Hasil pengecekan akan muncul di bawah tombol submit dan menunjukkan hasil True sesuai keinginan.



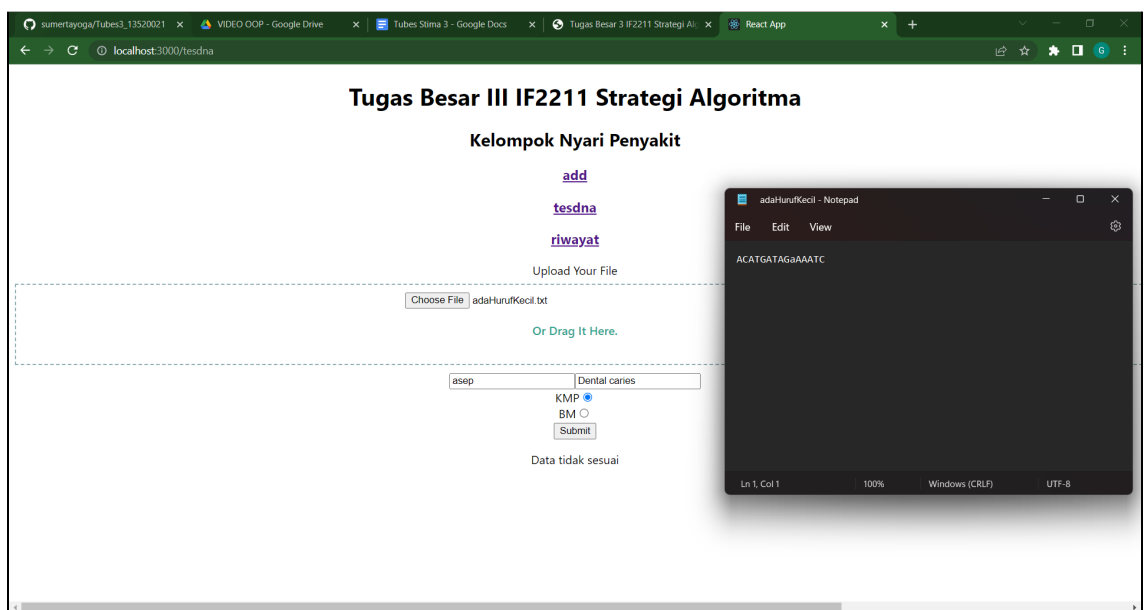
Gambar di atas merupakan pengujian ketika mendapatkan hasil false. File yang diupload adalah Buruli ulcer.txt yang mengandung pattern penyakit Buruli ulcer didalamnya.



Namun, pengecekan dilakukan dengan nama penyakit Dental caries sehingga menghasilkan False pada bagian hasil pengecekan.



Gambar di atas adalah pengecekan regex dari isi file yang diupload. Terlihat file tersebut mengandung untaian DNA tetapi terdapat spasi didalamnya sehingga akan menampilkan keterangan data tidak sesuai.



Gambar di atas adalah pengecekan ketika file berisi huruf kecil. Saat pengecekan regex masukan berupa huruf kecil ini akan ditolak sehingga menampilkan keterangan data tidak sesuai.

- Fitur menampilkan riwayat tes dna
  1. Masukan berupa tanggal dan nama penyakit

## Tugas Besar III IF2211 Strategi Algoritma

### Kelompok Nyari Penyakit

[add](#)

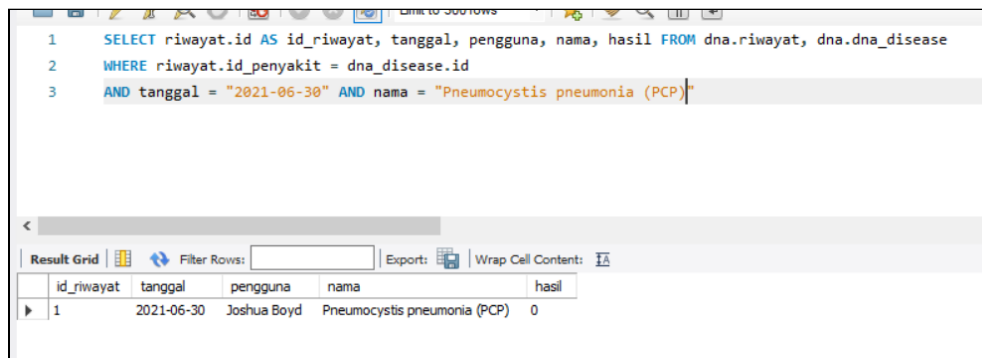
[tesdna](#)

[riwayat](#)

### Pencarian Riwayat

1	30 June 2021	Joshua Boyd	Pneumocystis pneumonia (PCP)	FALSE
---	--------------	-------------	------------------------------	-------

Pengecekan pada database



```
1 SELECT riwayat.id AS id_riwayat, tanggal, pengguna, nama, hasil FROM dna.riwayat, dna.dna_disease
2 WHERE riwayat.id_penyakit = dna_disease.id
3 AND tanggal = "2021-06-30" AND nama = "Pneumocystis pneumonia (PCP)"
```

	id_riwayat	tanggal	pengguna	nama	hasil
▶	1	2021-06-30	Joshua Boyd	Pneumocystis pneumonia (PCP)	0

2. Masukan berupa tanggal

## Pencarian Riwayat

2	29 June 2021	sulton	Bubonic plague	TRUE
---	--------------	--------	----------------	------

3	29 June 2021	chiko	Bubonic plague	FALSE
---	--------------	-------	----------------	-------

6	29 June 2021	dragon-san	Angiostrongyliasis	FALSE
---	--------------	------------	--------------------	-------

7	29 June 2021	nami swan	Cholera	TRUE
---	--------------	-----------	---------	------

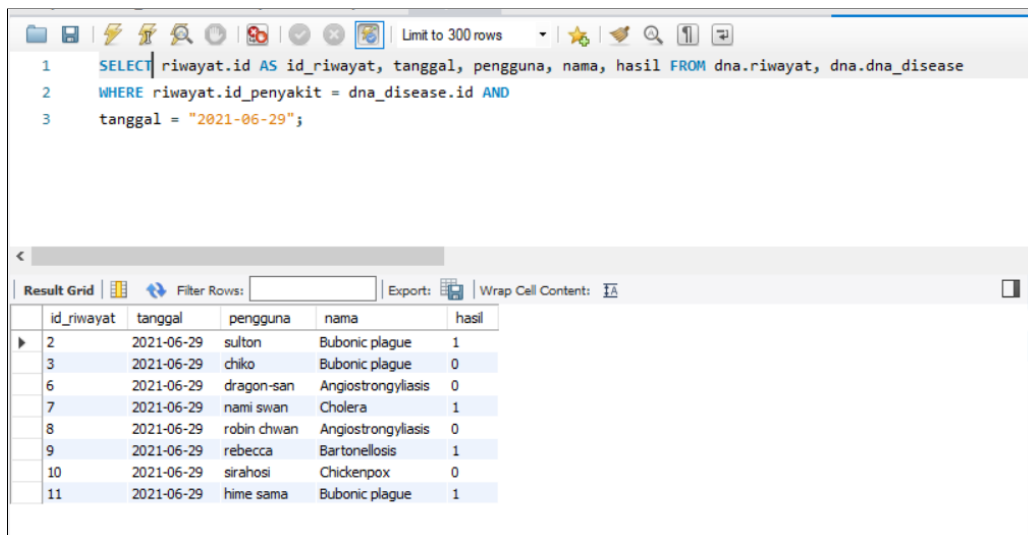
8	29 June 2021	robin chwan	Angiostrongyliasis	FALSE
---	--------------	-------------	--------------------	-------

9	29 June 2021	rebecca	Bartonellosis	TRUE
---	--------------	---------	---------------	------

10	29 June 2021	sirahosi	Chickenpox	FALSE
----	--------------	----------	------------	-------

11	29 June 2021	hime sama	Bubonic plague	TRUE
----	--------------	-----------	----------------	------

### Pemeriksaan melalui database



The screenshot shows a database query interface. At the top, there is a toolbar with various icons and a 'Limit to 300 rows' dropdown. Below the toolbar, a SQL query is entered in a text area:

```
1 SELECT riwayat.id AS id_riwayat, tanggal, pengguna, nama, hasil FROM dna.riwayat, dna.dna_disease
2 WHERE riwayat.id_penyakit = dna_disease.id AND
3 tanggal = "2021-06-29";
```

Below the query, there is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The result grid displays the following data:

	id_riwayat	tanggal	pengguna	nama	hasil
▶	2	2021-06-29	sulton	Bubonic plague	1
	3	2021-06-29	chiko	Bubonic plague	0
	6	2021-06-29	dragon-san	Angiostrongyliasis	0
	7	2021-06-29	nami swan	Cholera	1
	8	2021-06-29	robin chwan	Angiostrongyliasis	0
	9	2021-06-29	rebecca	Bartonellosis	1
	10	2021-06-29	sirahosi	Chickenpox	0
	11	2021-06-29	hime sama	Bubonic plague	1

### 3. Masukan berupa nama penyakit

# Tugas Besar III IF2211 Strategi Algoritma

## Kelompok Nyari Penyakit

[add](#)

[tesdna](#)

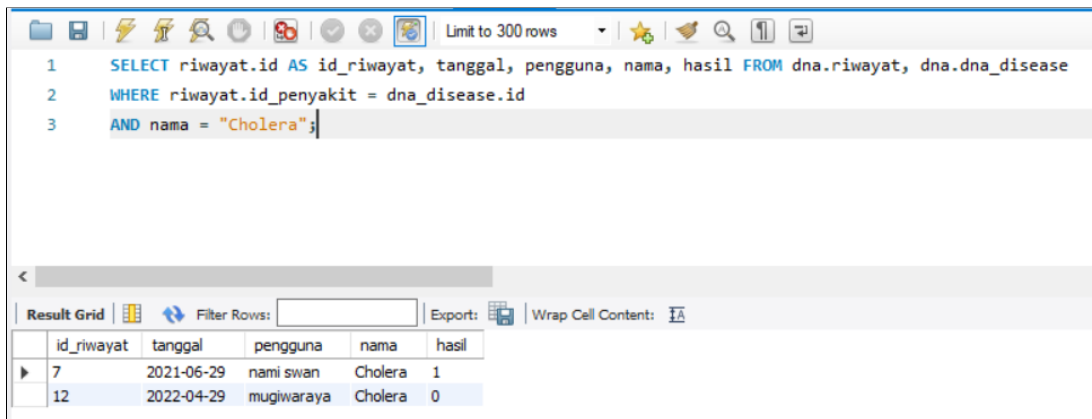
[riwayat](#)

## Pencarian Riwayat

7	29 June 2021	nami swan	Cholera	TRUE
---	--------------	-----------	---------	------

12	29 April 2022	mugiwaraya	Cholera	FALSE
----	---------------	------------	---------	-------

Pemeriksaan melalui database



The screenshot shows a database query interface. The SQL query is:

```
1 SELECT riwayat.id AS id_riwayat, tanggal, pengguna, nama, hasil FROM dna.riwayat, dna.dna_disease
2 WHERE riwayat.id_penyakit = dna_disease.id
3 AND nama = "Cholera";
```

The results are displayed in a table with the following columns: id\_riwayat, tanggal, pengguna, nama, and hasil.

id_riwayat	tanggal	pengguna	nama	hasil
7	2021-06-29	nami swan	Cholera	1
12	2022-04-29	mugiwaraya	Cholera	0

4. Masukan berupa nama penyakit dengan filter regex

## Tugas Besar III IF2211 Strategi Algoritma

Kelompok Nyari Penyakit

[add](#)

[tesdna](#)

[riwayat](#)

### Pencarian Riwayat

7	29 June 2021	nami swan	Cholera	TRUE
---	--------------	-----------	---------	------

12	29 April 2022	mugiaraya	Cholera	FALSE
----	---------------	-----------	---------	-------

Aplikasi tetap menampilkan penyakit cholera meskipun masukan merupakan choare yang tidak begitu cocok

## **BAB 5**

### **PENUTUP**

#### **5.1. Kesimpulan**

1. Algoritma pencocokan string Boyer-Moore dan Knutt-Morris-Pratt dapat diterapkan pada pencarian sequence DNA manusia terhadap sequence penyakit.
2. Regular Expression dapat digunakan untuk proses validasi sebuah sequence DNA serta sebagai filtering dalam proses pencarian riwayat.

#### **5.2. Saran**

1. Tidak semua penyakit berada pada DNA, tetapi juga pada RNA. Oleh karena itu, dapat diberikan fitur tambahan untuk pengecekan pola pada RNA
2. Sejauh ini, aplikasi web yang telah dibangun secara fungsionalitas dapat berjalan dengan baik. Namun, pada segi tampilan masih kurang menarik. Pengembang supaya dapat membuat aplikasi yang tampilannya lebih menarik.

#### **5.3. Komentar dan Refleksi**

Dengan diberikan tugas besar ini, kami sadar bahwa suatu algoritma dapat diterapkan untuk menyelesaikan masalah pada disiplin ilmu lain yang pada awalnya kami kira tidak ada hubungannya dengan keinformatikaan, seperti biologi. Penulis juga akhirnya dapat mencoba dan banyak belajar mengenai bagaimana cara pembuatan aplikasi web yang memiliki bagian frontend dan backend di dalamnya, seperti bagaimana dari pihak client dan server dapat mengambil data yang tersimpan pada database dan bagaimana frontend dan backend dapat terhubung satu sama lain.

## DAFTAR PUSTAKA

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

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

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Tugas-Besar-3-IF2211-Strategi-Algoritma-2022.pdf>

<http://ernapuspitasr.blogspot.com/2015/05/arsitektur-web.html> diakses pada 29 April 2022

## LAMPIRAN

Link video: <https://youtu.be/EoMRESz8vCw>

Link repository: [https://github.com/sumertayoga/Tubes3\\_13520021.git](https://github.com/sumertayoga/Tubes3_13520021.git)