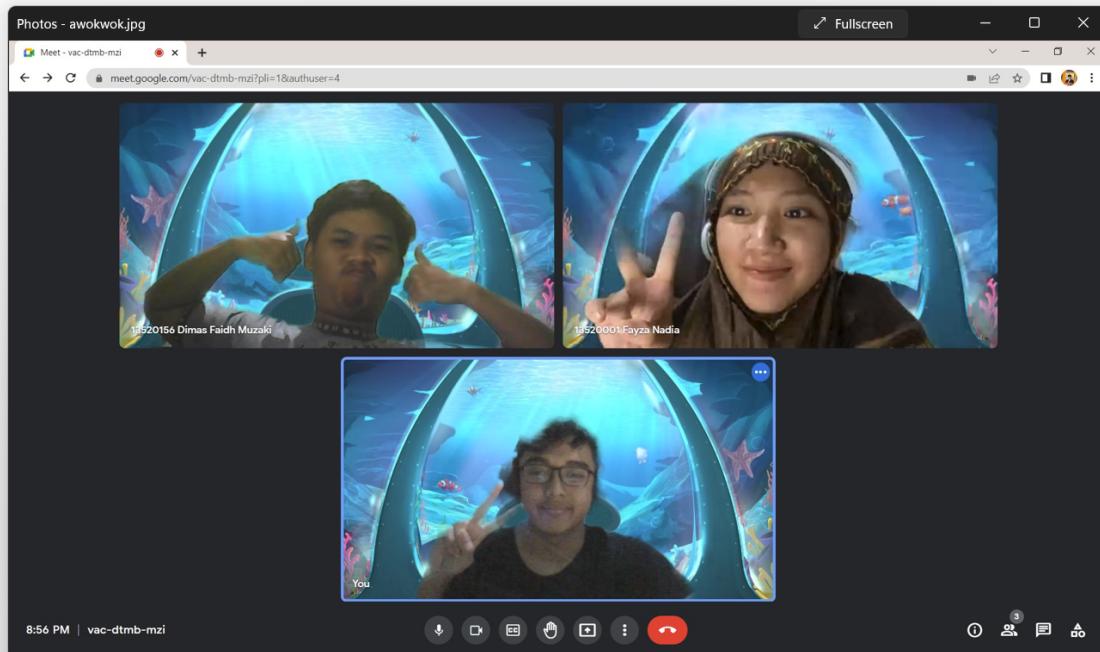


LAPORAN TUGAS BESAR III
IF2211 STRATEGI ALGORITMA

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



Disusun oleh:
Kelompok Istighfar Ikhtiar Tawakal
13520001 Fayza Nadia
13520026 Muhammad Fajar Ramadhan
13520156 Dimas Faidh Muzaki

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
SEMESTER 2 TAHUN 2021/2022

DAFTAR ISI

BAB I	2
BAB II	3
Knuth Morris Pratt Algorithm	3
Boyer Moore Algorithm	3
Regular Expression (Regex)	4
Aplikasi Web	4
BAB III	6
Pemeriksaan DNA	6
Penambahan Penyakit	8
Penampilan Riwayat	8
BAB IV	10
Spesifikasi Program	10
Tata Cara Penggunaan Program	16
Hasil Pengujian dan Analisis	19
Pengujian Penambahan Penyakit	19
Pengujian Pemeriksaan DNA	21
Pengujian Penampilan Riwayat	23
BAB V	26
Kesimpulan	26
Saran	26
TAUTAN REPOSITORY GITHUB	27
TAUTAN VIDEO DEMO	27
DAFTAR PUSTAKA	27

BAB I

DESKRIPSI TUGAS

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

Salah satu jenis tes DNA yang sangat berkaitan dengan dunia bioinformatika adalah DNA sequence analysis. DNA sequence analysis adalah sebuah cara yang dapat digunakan untuk memprediksi berbagai macam penyakit yang tersimpan pada database berdasarkan urutan sekuens DNA-nya. Sebuah sekuens DNA adalah suatu representasi string of nucleotides yang disimpan pada suatu rantai DNA, sebagai contoh: ATTCTGTAACCTAGTAAGTTA. Teknik pattern matching memegang peranan penting untuk dapat menganalisis sekuens DNA yang sangat panjang dalam waktu singkat.

Dalam Tugas Besar ini dibuat sebuah aplikasi DNA Pattern Matching berbasis web. Dengan memanfaatkan algoritma String Matching dan Regular Expression yang telah dipelajari. Aplikasi yang dibuat bersifat interaktif dan dapat digunakan untuk mendeteksi apakah seorang pasien mempunyai penyakit genetik tertentu. Hasil prediksi tersebut dapat disimpan pada basis data untuk kemudian ditampilkan berdasarkan query pencarian

BAB II

LANDASAN TEORI

Knuth Morris Pratt Algorithm

Knuth Morris Pratt Algorithm merupakan salah satu algoritma *pattern searching/pattern matching* yang disusun oleh James H Morris dan ditemukan beberapa minggu kemudian oleh Donald Knuth pada teori automata. Algoritma Knuth Morris Pratt mempunyai dasar yang sama pada string matching menggunakan *brute force*. Perbedaan pada algoritma ini adalah ketika terjadi *mismatch* pada perbandingan string, seberapa besar pergeseran perbandingan dari *pattern* yang bisa dilakukan untuk pencocokkan character selanjutnya berdasarkan informasi yang sudah didapat sehingga mengurangi jumlah perbandingan yang sia-sia.

Secara sistematis, langkah-langkah yang dilakukan algoritma Knuth Morris Pratt pada saat pencocokan string adalah sebagai berikut:

1. Algoritma melakukan *pre-process* untuk membuat *lps* atau prefix terpanjang pada pattern yang bisa menjadi suffix
2. Algoritma mulai mencocokkan pattern dengan text mulai dari kiri hingga ditemukan pattern
3. Jika terjadi *mismatch* pada perbandingan maka lakukan kondisi berikut:
 - a. Perbandingan selanjutnya pada pattern dan text adalah perbandingan pattern dimulai dari posisi $lps[j-1]$ dan dilanjutkan jika perbandingan terus cocok
 - b. Jika terjadi *mismatch* kembali ke langkah a
 - c. Jika perbandingan sudah mencapai tidak adanya *lps* ($lps[j-1]$ sama dengan 0) maka lakukan perbandingan dari awal pattern dan text pada posisi selanjutnya dan kembali ke langkah a jika terjadi *mismatch*

Boyer Moore Algorithm

Algoritma Boyer-Moore adalah salah satu algoritma pencarian string yang dipublikasikan oleh Robert S. Boyer dan J. Strother Moore pada tahun 1977. Algoritma Boyer Moore dianggap

sebagai salah satu algoritma yang paling efisien untuk aplikasi pencocokan pola umum. Ia mampu mengenali dan melewati area tertentu dalam teks yang tidak ditemukan kecocokan. Ide dibalik algoritma ini adalah bahwa dengan memulai pencocokan karakter dari kanan, dan bukan dari kiri, maka akan lebih banyak informasi yang didapat.

Secara sistematis, langkah-langkah yang dilakukan algoritma Boyer-Moore pada saat pencocokan string adalah:

1. Algoritma Boyer-Moore mulai mencocokkan pattern pada awal teks.
2. Dari kanan ke kiri, algoritma ini akan mencocokkan karakter per karakter pattern dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
 - a. Karakter di pattern dan di teks yang dibandingkan tidak cocok (mismatch).
 - b. Semua karakter di pattern cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser pattern dengan memaksimalkan nilai pergeseran good-suffix dan pergeseran bad-character, lalu mengulangi langkah 2 sampai pattern berada di ujung teks.

Regular Expression (Regex)

Regular expression merupakan istilah yang digunakan untuk kodifikasi metode pencarian yang ditemukan oleh ahli ilmu Matematika Amerika Stephen Kleene. Secara singkat, dapat dijabarkan bahwa regular expression merupakan cara cepat untuk pencarian pattern dalam sebuah teks. Regular expressions dapat digunakan untuk mencari pattern tertentu dalam teks dan jika sudah ditemukan, pattern dapat dimodifikasi dengan berbagai cara.

Aplikasi Web

Aplikasi web adalah sebuah software berbentuk client-side dan server-side dimana client dapat menjalankan atau *make-request* kepada sebuah browser. Dalam kata lain, aplikasi web merupakan sebuah program komputer yang memanfaatkan teknologi web browser untuk melakukan sebuah proses atau pekerjaan pada internet. Pada tugas besar ini, dibuat sebuah aplikasi web yang dapat memprediksi sebuah DNA seseorang apakah terdapat penyakit pada DNA tersebut. Aplikasi web

ini menerima input rantai seseorang dan melakukan *pattern matching* pada rantai penyakit yang terdapat di database dengan hasil output adalah tingkat kemiripan *pattern* dan apakah orang tersebut mengidap suatu penyakit.

BAB III

ANALISIS PEMECAHAN MASALAH

Aplikasi web yang dibangun akan memiliki struktur yang terdiri atas backend berupa web server yang terhubung dengan basis data dan frontend yang menjadi client untuk digunakan oleh pengguna. Pada aplikasi backend akan diimplementasikan algoritma string matching Knuth Morris Pratt Algorithm serta Boyer Moore Algorithm untuk merealisasikan penemuan rantai DNA penyakit dalam rantai DNA Manusia. Aplikasi frontend akan digunakan oleh pengguna dalam memasukkan informasi diri dan rantai DNA untuk selanjutnya data tersebut dikirim ke backend untuk dilakukan pencarian. Selain itu pengguna juga dapat menambahkan data penyakit baru dengan memasukkan nama dan rantai DNA. Terakhir, pengguna dapat melihat riwayat pemeriksaan rantai DNA dengan menggunakan query. Riwayat yang ditampilkan dapat dikelompokkan berdasarkan penyakit yang diperiksa, tanggal dilakukan pemeriksaan, ataupun keduanya berdasarkan query yang dimasukkan.

Pemeriksaan DNA

Dalam melaksanakan pemeriksaan rantai DNA, aplikasi akan meminta pengguna untuk memasukkan nama diri, nama penyakit yang ingin diperiksa, dan rantai DNA dirinya dalam format file txt. Sebelum diteruskan ke server, aplikasi client akan memeriksa terlebih dahulu isi dari file rantai DNA pengguna. Untuk dapat melanjutkan pemeriksaan, rantai DNA harus memenuhi syarat sebagai berikut:

1. Tidak boleh ada huruf kecil,
2. Tidak boleh ada huruf selain AGCT, dan
3. Tidak ada spasi

Tes DNA

Periksa apakah jangan-jangan DNA Anda mengidap penyakit..?

Nama Pengguna:

Prediksi Penyakit:

Sequence DNA:

Choose File No file chosen

* File yang diperbolehkan adalah .txt
* Tidak ada huruf kecil
* Tidak boleh ada huruf selain AGCT
* Tidak ada spasi

Submit!

Gambar 1

UI pada Halaman Fitur Pemeriksaan DNA

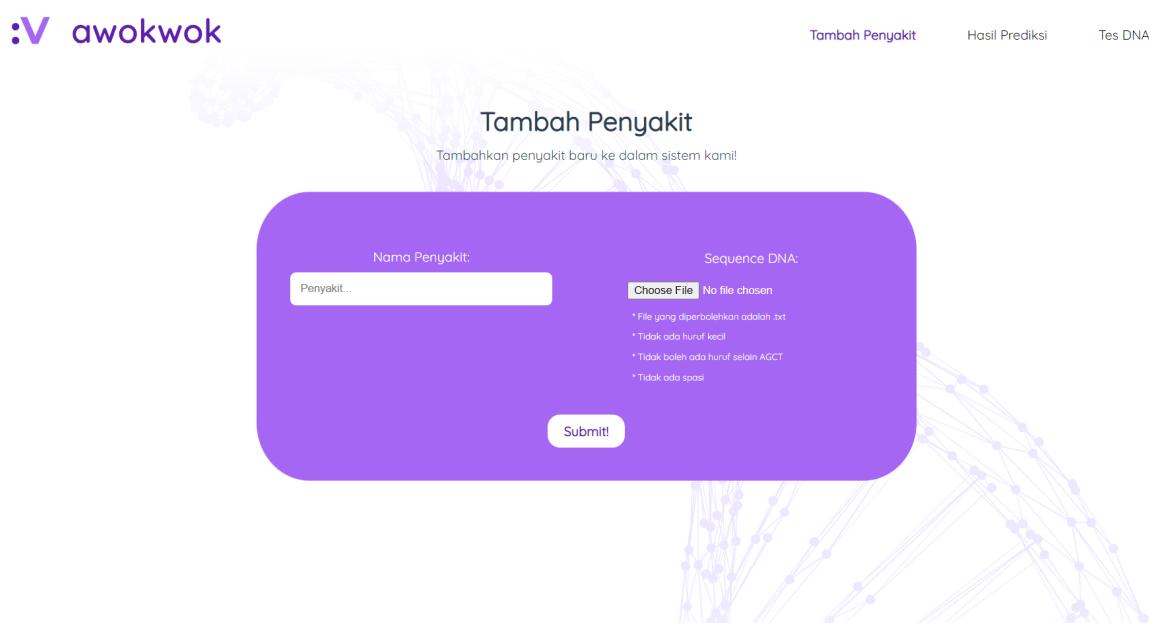
Untuk memeriksa kevalidasian rantai dalam memenuhi syarat-syarat tersebut, aplikasi client memanfaatkan regular expression untuk memeriksa isi file txt. Regular expression digunakan untuk menemukan pola yang dilarang oleh syarat. Apabila pola ditemukan, rantai DNA yang dimasukkan tidak dapat dipakai dan tidak akan diteruskan ke server. Pola ekspresi regular yang dipakai untuk tiap syarat adalah sebagai berikut:

1. Mencari huruf kecil: /[a-z]/g;
2. Mencari huruf selain AGCT: /[^AGCT]/g
3. Mencari spasi: /\s/g;

Rantai DNA yang memenuhi syarat dapat diteruskan ke server untuk dilakukan pemeriksaan. Server akan menerapkan algoritma KMP pada rantai DNA pengguna dan rantai DNA penyakit. Namun, Server akan memeriksa terlebih dahulu ketersediaan DNA penyakit yang ingin dicocokkan pada basis data server. Apabila tidak ditemukan, server akan memberikan pesan error pada client. Selain memeriksa kecocokan, server juga akan memeriksa persentase kemiripan rantai DNA pengguna dan penyakit. Terakhir, server akan memberikan client data hasil pemeriksaan untuk dilihat oleh pengguna. Selain itu, data akan disimpan ke dalam basis data.

Penambahan Penyakit

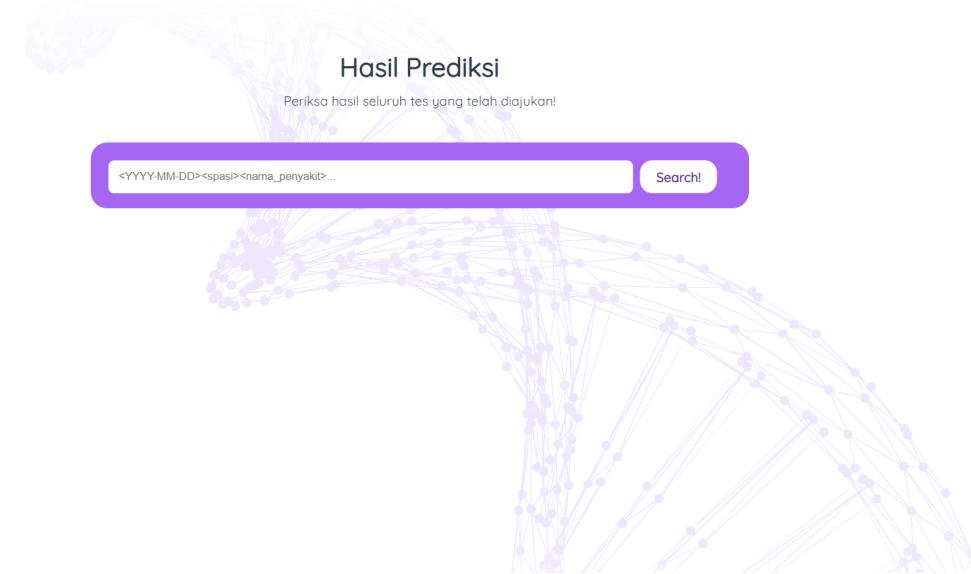
Untuk penambahan basis data penyakit, pengguna dapat memasukkan nama penyakit dan rantai DNA penyakit dalam format file txt. Sama seperti sebelumnya, rantai DNA harus memenuhi syarat-syarat dengan memanfaatkan regular expression. Rantai DNA yang valid akan diteruskan ke server beserta namanya. Apabila basis data belum memiliki data untuk nama penyakit tersebut, data penyakit akan ditambahkan ke dalam basis data.



Gambar 2
UI pada Halaman Fitur Penambahan Penyakit

Penampilan Riwayat

Dalam hal menampilkan riwayat pemeriksaan, aplikasi menyediakan kolom teks untuk pengguna memasukkan query. Query yang diterima berbentuk tanggal dengan format YYYY-MM-DD dan/atau nama penyakit. Untuk menampilkan data riwayat sesuai yang diinginkan pengguna, aplikasi client memanfaatkan regular expression pada query untuk menentukan keinginan pengguna.



Gambar 3
UI pada Halaman Fitur Pencarian Riwayat

Aplikasi dibangun dengan pembagian backend menggunakan bahasa pemrograman Golang dan frontend menggunakan framework VueJs. Aplikasi dibangun sesuai dengan kebutuhan fungsional seperti yang sudah disebutkan sebelumnya.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Spesifikasi Program

Aplikasi memiliki sebuah basis data yang berkomunikasi dengan backend. Basis data dibuat pada DBMS postgres dengan skema basis data sebagai berikut:

- Penyakits = (nama, rantai)
- Pemeriksaans = (id, nama, tanggal, prediksi, hasil, rantai, penyakit)

Basis data diakses oleh backend sesuai dengan request dari client. Method dan url yang disediakan oleh backend untuk mengakses data antara lain:

1. GET(/penyakit/get)
Berfungsi untuk mengambil semua data penyakit yang ada
2. GET(/penyakit/get/{nama})
Berfungsi untuk mengambil data penyakit dengan nama yang sesuai
3. POST(/penyakit/create) Body : {nama, rantai}
Berfungsi untuk menambahkan penyakit baru pada basis data
4. GET(/pemeriksaan/get)
Berfungsi untuk mengambil semua data pemeriksaan yang ada
5. GET(/pemeriksaan/get/when?tanggal=)
Berfungsi untuk mengambil data pemeriksaan yang dilakukan pada tanggal tertentu
6. GET(/pemeriksaan/get/what?penyakit=)
Berfungsi untuk mengambil data pemeriksaan dengan nama penyakit tertentu
7. GET(/pemeriksaan/get/whenwhat?penyakit=&tanggal=)
Berfungsi untuk mengambil data pemeriksaan dengan nama penyakit dan pada tanggal tertentu
8. POST(/pemeriksaan/create) Body : {nama,rantai,penyakit}
Berfungsi untuk melakukan pemeriksaan rantai DNA pengguna dengan rantai DNA penyakit pada sisi backend dan menambahkannya pada basis data

Fungsi KMP dan Boyer Moore diimplementasikan pada sisi backend dan digunakan pada controller/handler untuk HTTP request ke-8.

Secara garis besar, aplikasi terbagi menjadi dua bagian utama:

1. Frontend

Pada bagian frontend, aplikasi dapat didekomposisi menjadi beberapa bagian utama yang saling terhubung.

a. index.js

Nama	Deskripsi
router	Fungsi yang digunakan untuk menyimpan routing path dari masing-masing laman pada aplikasi

b. Main.vue

Bagian ini berisi landing page yang dapat melakukan redirect page ke laman pemeriksaan DNA. Selain itu, pada laman ini terdapat pula profil singkat masing-masing anggota kelompok sebagai pembuat dari aplikasi Awokwok DNA Matching ini.

c. AddPenyakit.vue

Bagian ini berisi laman untuk menambahkan penyakit baru pada basis data.

Nama	Deskripsi
resetFlags	Melakukan reset pada variabel boolean ketentuan sequence DNA
submitPenyakit	Melakukan submisi penyakit baru dari masukan pengguna dan menghubungkan dengan backend
readUploadedFileAsText	Melakukan pemeriksaan terhadap tipe file dan mengubahnya menjadi teks
upload	Melakukan pemeriksaan terhadap ketentuan sequence DNA dan mengunggahnya

d. Prediction.vue

Bagian ini berisi laman untuk melakukan pemeriksaan DNA berdasarkan basis data penyakit yang dimiliki serta mengunggah hasil pemeriksaan ke basis data.

Nama	Deskripsi
resetFlags	Melakukan reset pada variabel boolean ketentuan sequence DNA
parseDate	Mengubah format waktu pemeriksaan pada basis data menjadi format YYYY-MM-DD untuk ditampilkan pada hasil
submitPemeriksaan	Melakukan submisi pemeriksaan baru dari masukan pengguna dan menghubungkan dengan backend
readUploadedFileAsText	Melakukan pemeriksaan terhadap tipe file dan mengubahnya menjadi teks
upload	Melakukan pemeriksaan terhadap ketentuan sequence DNA dan mengunggahnya

e. History.vue

Bagian ini berisi laman untuk melihat riwayat pemeriksaan DNA yang telah dilakukan sesuai dengan basis data.

Nama	Deskripsi
submit	Melakukan submisi sesuai query masukan pengguna dan menghubungkan dengan backend untuk kemudian ditampilkan

f. Query.vue

Bagian ini berisi komponen yang digunakan untuk menyimpan suatu hasil pemeriksaan sesuai query yang dimasukkan pengguna. Komponen ini dipanggil pada laman History.vue dan memiliki properti:

- nomor
 - id
 - nama
 - tanggal
 - penyakit
 - prediksi
 - hasil
- g. Test.vue
- Bagian ini berisi komponen yang digunakan untuk menyimpan suatu hasil pemeriksaan. Komponen ini dipanggil pada laman Pemeriksaan.vue dan memiliki properti:
- tanggal
 - nama
 - penyakit
 - prediksi
 - hasil

2. Backend

Pada bagian backend, aplikasi didekomposisi menjadi beberapa bagian yang saling terhubung.

a. controllers

Bagian ini berisi fungsi dan prosedur untuk mengambil atau membuat data pada database

1) pemeriksaanControllers.go

Bagian ini terhubung dengan database pemeriksaan ketika dilakukan request

Nama	Deskripsi
CreatePemeriksaan	Membuat record pemeriksaan baru pada database
GetAllPemeriksaan	Mengambil semua data pada tabel pemeriksaan
GetPemeriksaanByPenyakit	Mengambil semua data pada tabel pemeriksaan berdasarkan penyakit yang dicari
GetPemeriksaanByTanggal	Mengambil semua data pada tabel pemeriksaan berdasarkan tanggal yang dicari
GetPemeriksaanByTang	Mengambil semua data pada tabel pemeriksaan

galAndPenyakit	berdasarkan tanggal dan penyakit yang dicari
----------------	--

2) penyakitController.go

Bagian ini terhubung dengan database penyakit ketika dilakukan request

Nama	Deskripsi
GetAllPenyakit	Mengambil semua data pada tabel penyakit
CreatePenyakit	Menambah record penyakit baru pada database

b. database

1) connection.go

Bagian ini berisi fungsi untuk melakukan otomasi pembuatan database berdasarkan entity pemeriksaan dan penyakit atau koneksi jika database sudah ada

Nama	Deskripsi
Connect	Fungsi koneksi terhadap database
MigratePenyakit	Migrasi entity pemeriksaan menjadi tabel pada database
MigratePemeriksaan	Migrasi entity penyakit menjadi tabel pada database

2) database.go

Bagian ini merupakan config dalam pembuatan dan koneksi terhadap database

Nama	Deskripsi
GetConnectionString	Melakukan koneksi terhadap database berdasarkan config

c. entity

1) pemeriksaan.go

Bagian ini berisikan struct untuk menjadi dasar entity atau tabel pemeriksaan pada database.

2) penyakit.go

Bagian ini berisikan struct untuk menjadi dasar entity atau tabel penyakit pada database.

d. stringMatching

1) bmSearch.go

Bagian ini berisi fungsi untuk melakukan string matching dengan algoritma Boyer-Moore.

Nama	Deskripsi
bmMatch	Melakukan string matching dengan algoritma Boyer-Moore
min	Mengembalikan bilangan integer yang lebih kecil
buildLast	Membentuk last occurrence array sebelum melakukan algoritma Boyer-Moore

2) kmpSearch.go

Bagian ini berisi fungsi untuk melakukan string matching dengan algoritma Knuth–Morris–Pratt.

Nama	Deskripsi
KmpMatch	Melakukan string matching dengan algoritma Knuth-Morris-Pratt
ComputeLps	Menghitung lps atau prefix terpanjang yang juga menjadi suffix pada sebuah pattern

e. main.go

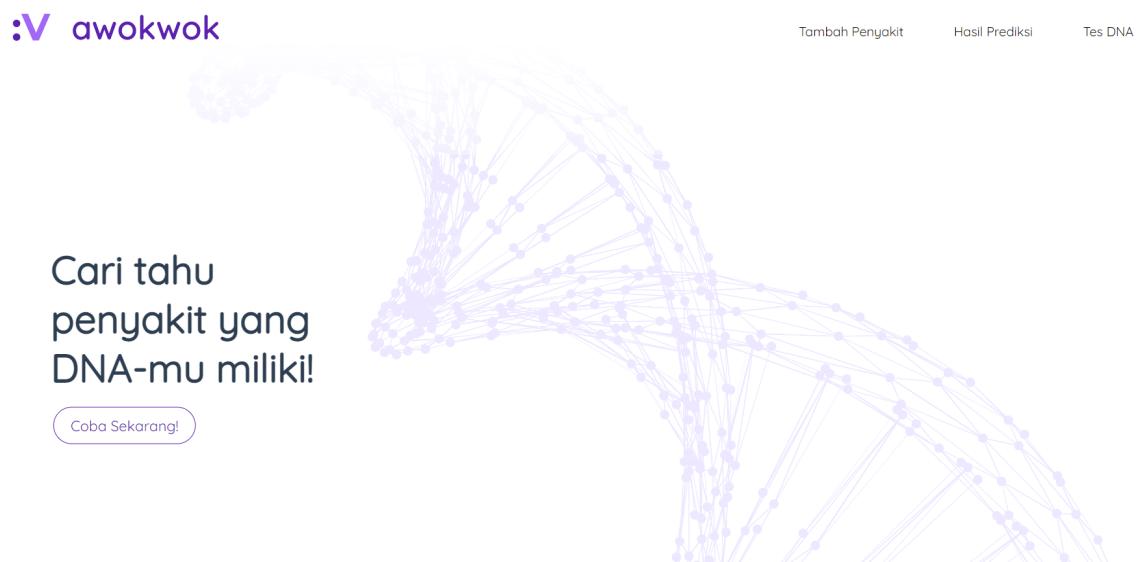
Bagian ini fungsi utama yang dijalankan dalam program.

Nama	Deskripsi

main	Memanggil fungsi serta prosedur lain
initialiseHandlers	Menginisiasi router handler untuk modifikasi basis data
initDB	Inisialisasi basis data

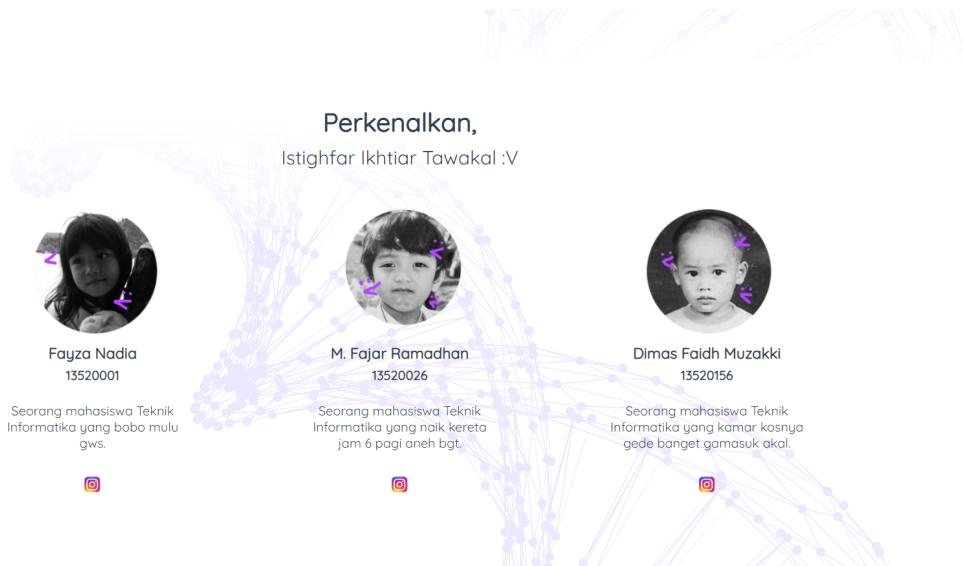
Tata Cara Penggunaan Program

Aplikasi client dibangun menggunakan framework VueJs dengan tampilan awal sebagai berikut:



Gambar 4

Landing Page



Gambar 5

Profil Kelompok

Menambahkan data penyakit baru pada basis data dapat dilakukan di menu Tambah Penyakit.

Tambah Penyakit

Tambahkan penyakit baru ke dalam sistem kami!

Nama Penyakit:

Sequence DNA:

Choose File

* File yang diperbolehkan adalah .txt
 * Tidak ada huruf kecil
 * Tidak boleh ada huruf selain AGCT
 * Tidak ada spasi

Submit!

Gambar 6

Menambahkan Penyakit Baru

Pada menu ini kita dapat menginput nama penyakit yang ingin ditambahkan serta rantai DNA dalam bentuk file txt. Pada saat mengupload file txt, ada beberapa teks yang merupakan syarat

yang harus dipenuhi oleh isi dari file txt. Teks berwarna hijau memiliki arti bahwa rantai DNA telah memenuhi syarat tersebut. Rantai DNA perlu memenuhi semua syarat untuk dapat disubmit. Penyakit yang sudah ditambahkan ke dalam basis data dapat digunakan untuk melakukan pemeriksaan pada menu Tes DNA.

:V awokwok

Tambah Penyakit Hasil Prediksi Tes DNA

Tes DNA

Periksa apakah jangan-jangan DNA Anda mengidap penyakit...?

Nama Pengguna:

Nama Pengguna...

Prediksi Penyakit:

Prediksi Penyakit...

Sequence DNA:

Choose File No file chosen

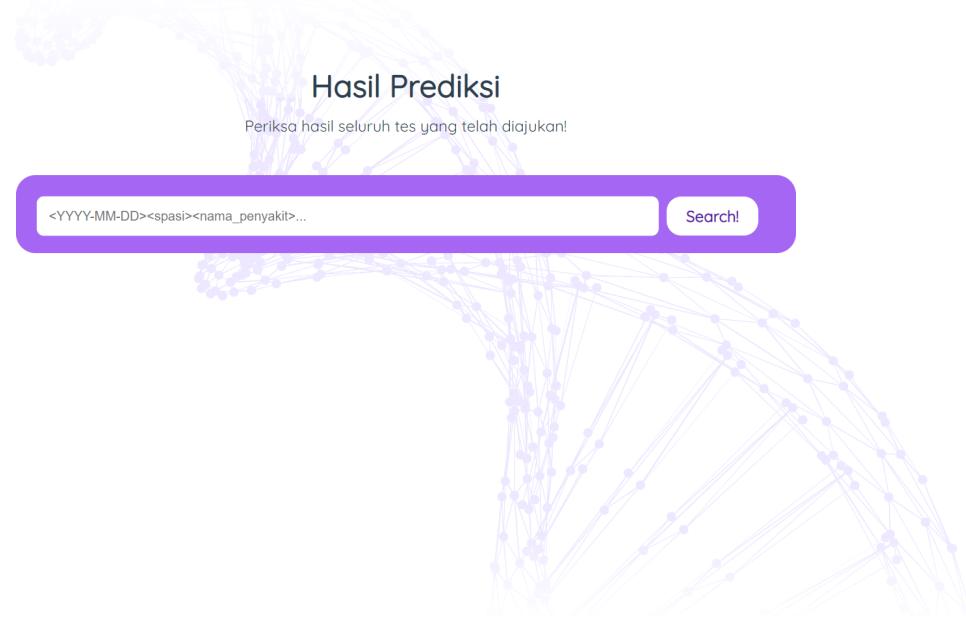
- * File yang diperbolehkan adalah .txt
- * Tidak ada huruf kecil
- * Tidak boleh ada huruf selain AGCT
- * Tidak ada spasi

Submit!

Gambar 7

Melakukan pemeriksaan DNA

Sama seperti sebelumnya, untuk melakukan tes DNA kita perlu memasukkan nama kita dan penyakit yang ingin diperiksa. Selain itu, kita juga perlu mengupload file txt yang berisi rantai DNA kita untuk diperiksa. Apabila kolom sudah diisi, rantai DNA sudah benar, dan penyakit terdapat pada basis data, pengguna dapat submit untuk memulai pemeriksaan dan hasil pemeriksaan akan ditampilkan pada bagian bawah layar. Pemeriksaan yang pernah dilakukan akan masuk ke dalam basis data aplikasi. Pengguna dapat melihat riwayat pemeriksaan yang pernah dilakukan dengan mengakses halaman Hasil Prediksi.

**Gambar 8**

Melakukan pencarian riwayat prediksi

Pada halaman ini, pengguna dapat memilih untuk:

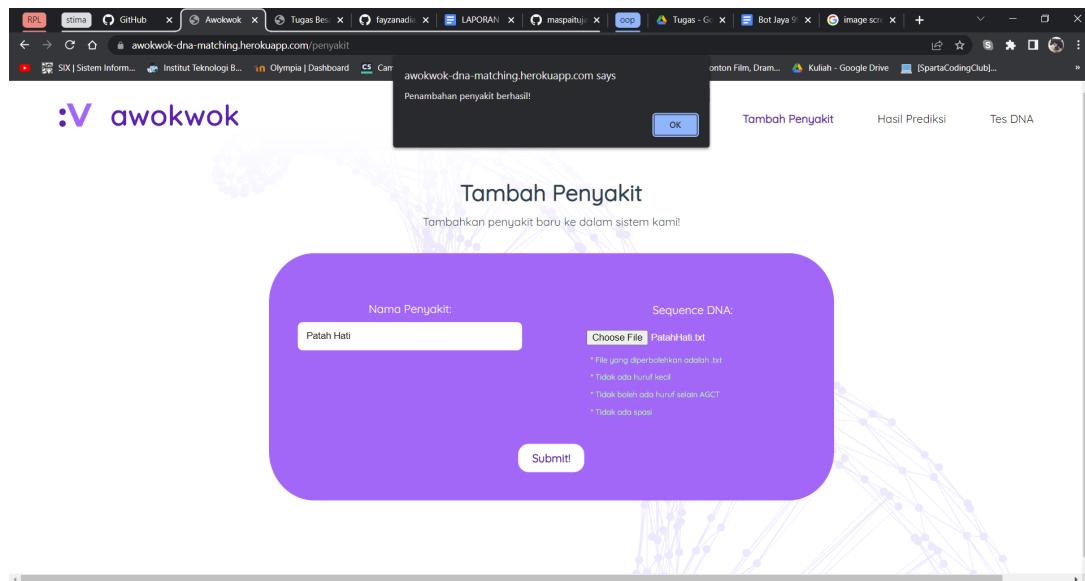
1. Melihat semua riwayat pemeriksaan
2. Melihat riwayat pemeriksaan yang dilakukan pada tanggal tertentu
3. Melihat riwayat pemeriksaan untuk penyakit tertentu
4. Melihat riwayat pemeriksaan untuk penyakit tertentu yang dilakukan pada tanggal tertentu

Untuk poin satu, pengguna dapat melakukannya dengan mengklik tombol “Search!” tanpa mengisi query. Selain itu, pengguna harus mengisi query sesuai yang diinginkan. Query dapat ditulis dengan format “<tanggal><spasi><nama_penyakit>”, “<nama_penyakit><spasi><tanggal>”, “<nama_penyakit>”, atau “<tanggal>” dengan catatan query tanggal harus ditulis dalam format YYYY-MM-DD.

Hasil Pengujian dan Analisis

- a. Pengujian Penambahan Penyakit

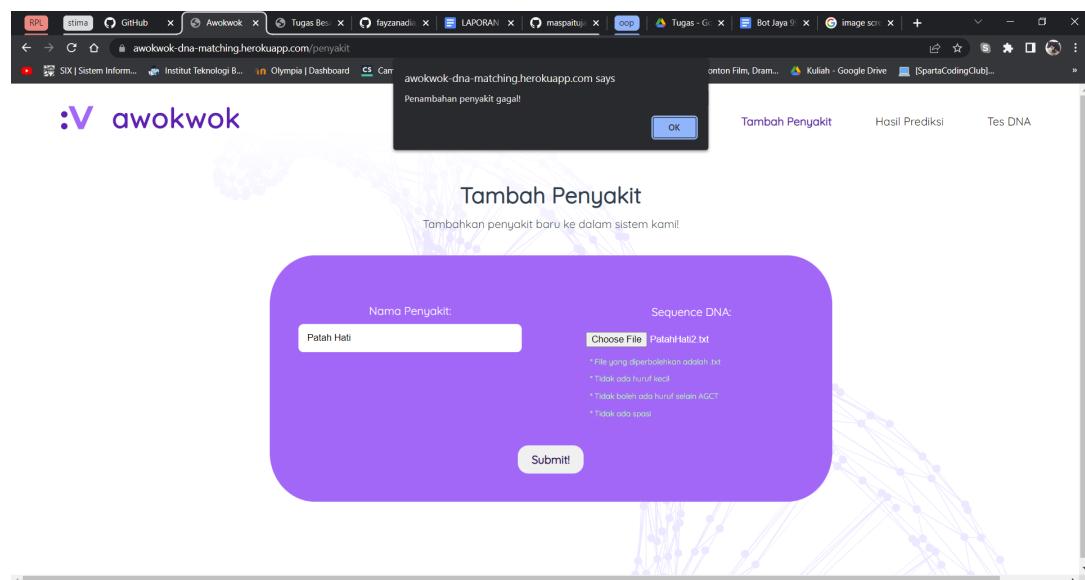
Kasus 1



Gambar 9
Penambahan Penyakit Baru Berhasil

Pada pengujian pertama, kami melakukan penambahan penyakit dengan masukan nama penyakit serta sequence DNA yang sesuai. Nama penyakit belum terdaftar pada basis data sehingga dapat diterima oleh program. File masukan yang berisi sequence DNA juga memenuhi empat kriteria sehingga dapat diterima oleh program. Kedua masukan tersebut kemudian ditambahkan sebagai penyakit baru pada basis data, ditandai dengan munculnya pop up message bertuliskan “Penambahan penyakit berhasil!”

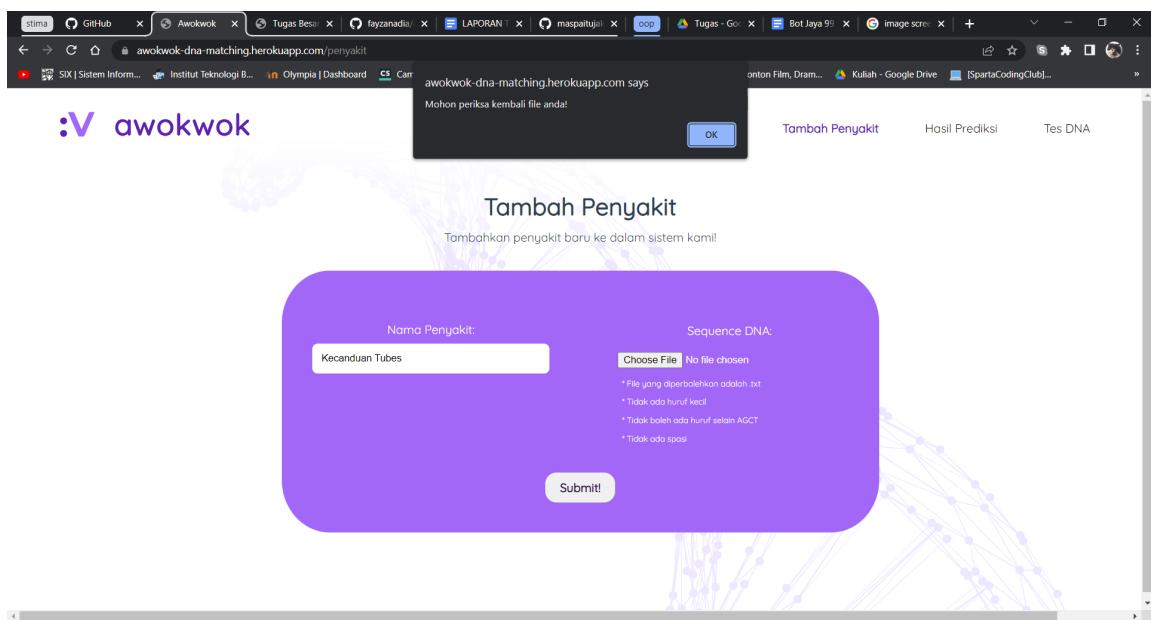
Kasus 2



Gambar 10
Penambahan Penyakit Baru Gagal 1

Pada pengujian kedua, kami melakukan penambahan penyakit dengan contoh masukan nama penyakit yang sama seperti nama penyakit yang telah terdaftar sebelumnya. Walaupun file masukan sequence DNA yang diberikan telah memenuhi keempat syarat, program tetap menolak masukan tersebut. Program kemudian akan menampilkan pop up message bertuliskan “Penambahan penyakit gagal!”

Kasus 3

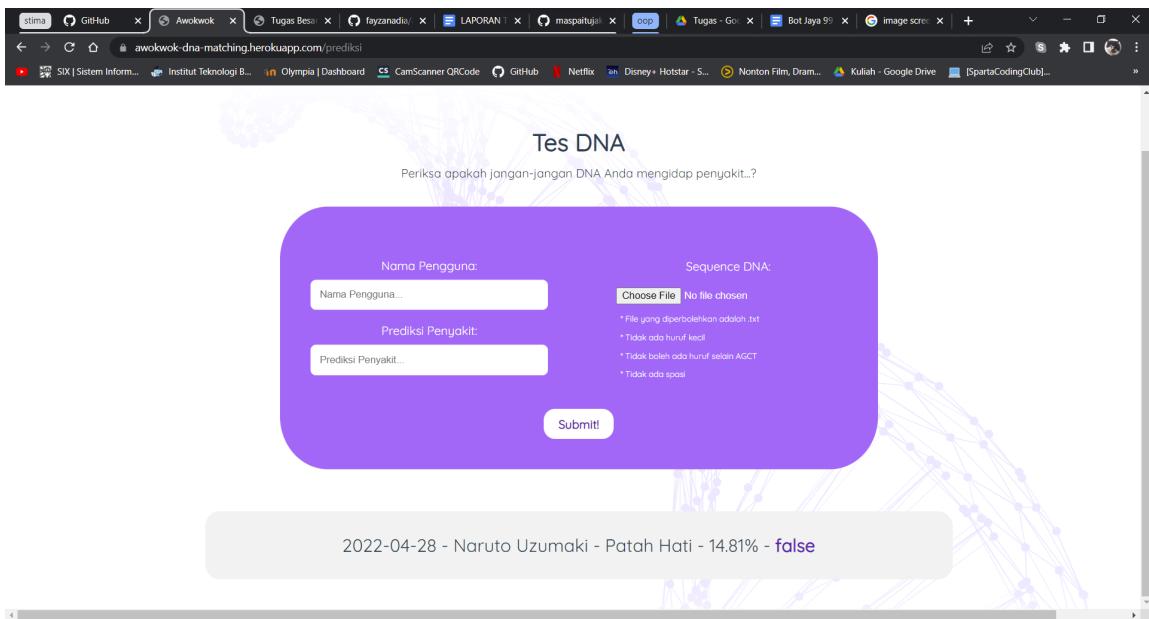


Gambar 11
Penambahan Penyakit Baru Gagal 2

Pada pengujian ketiga, kami melakukan penambahan penyakit dengan contoh masukan yang tidak benar. Dalam hal ini, file masukan tidak diberikan sehingga keempat kriteria tersebut tidak terpenuhi, ditandai dengan warna teks yang tidak menyala hijau. Sehingga, program menolak masukan tersebut dengan sebuah pop up message bertuliskan “Mohon periksa kembali file anda!”

b. Pengujian Pemeriksaan DNA

Kasus 1



Gambar 12

Pemeriksaan DNA Berhasil dengan Hasil False

Pada pengujian pertama, kami melakukan pemeriksaan DNA dengan contoh masukan nama penyakit serta sequence DNA pengguna yang sesuai.

Sequence DNA penyakit Patah Hati:

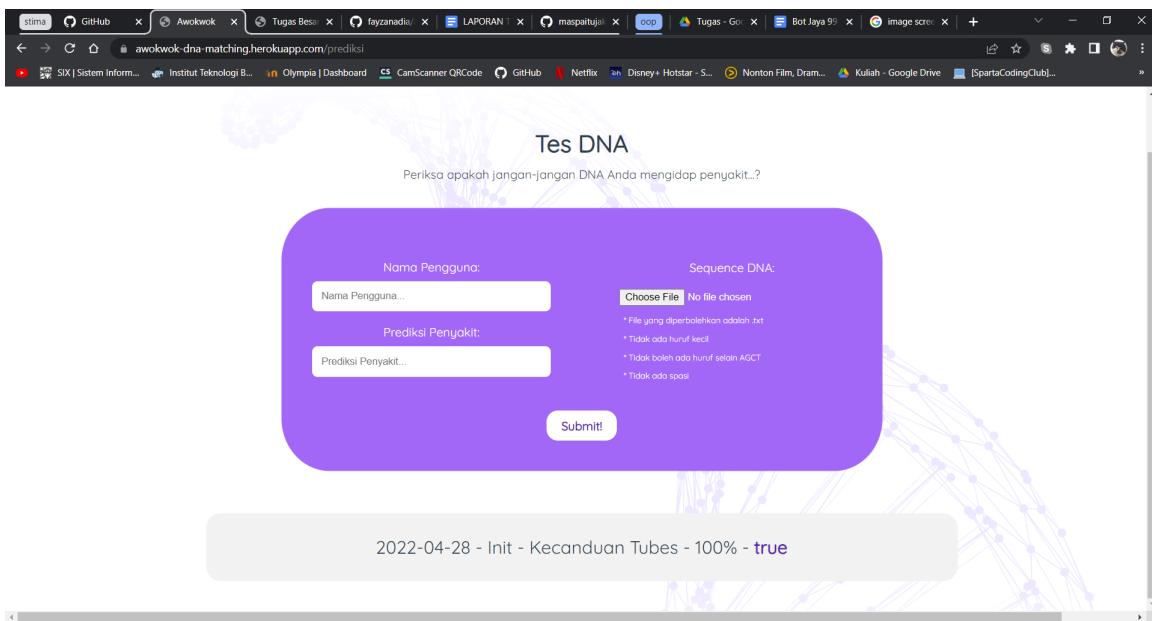
ATGCGTACATAGCAGATACGATCAGAT

Sequence DNA pengguna:

CATAGCGAAATGATGATCGATGCATCGATCGATCGATCGATGCTAGCTGA

Dari kedua masukan sequence DNA tersebut, didapatkan hasil sebagaimana tertera pada layar, yaitu similarity senilai 14.81% dengan hasil false.

Kasus 2



Gambar 13

Pemeriksaan DNA Berhasil dengan Hasil True

Pada pengujian kedua, kami melakukan pemeriksaan DNA dengan contoh masukan nama penyakit serta sequence DNA pengguna yang sesuai.

Sequence DNA penyakit Kecanduan Tubes:

GGATCTACTAATTCAGATCGATCG

Sequence DNA pengguna:

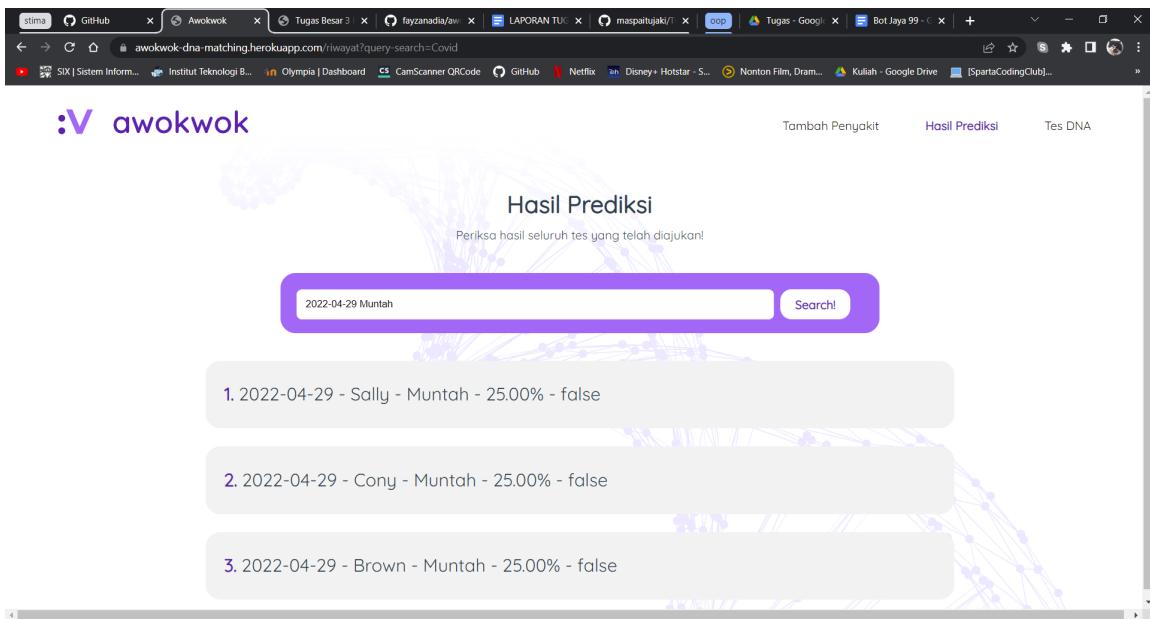
CTACGGGATCTACTAATTCAGATCGATCGAGTG

Dari kedua masukan sequence DNA tersebut, didapatkan hasil sebagaimana tertera pada layar, yaitu similarity senilai 100% dengan hasil true.

c. Pengujian Penampilan Riwayat

Kasus 1

Pada pengujian pertama, kami melakukan pencarian riwayat tes dengan contoh masukan lengkap tanggal pengujian dan nama penyakit.

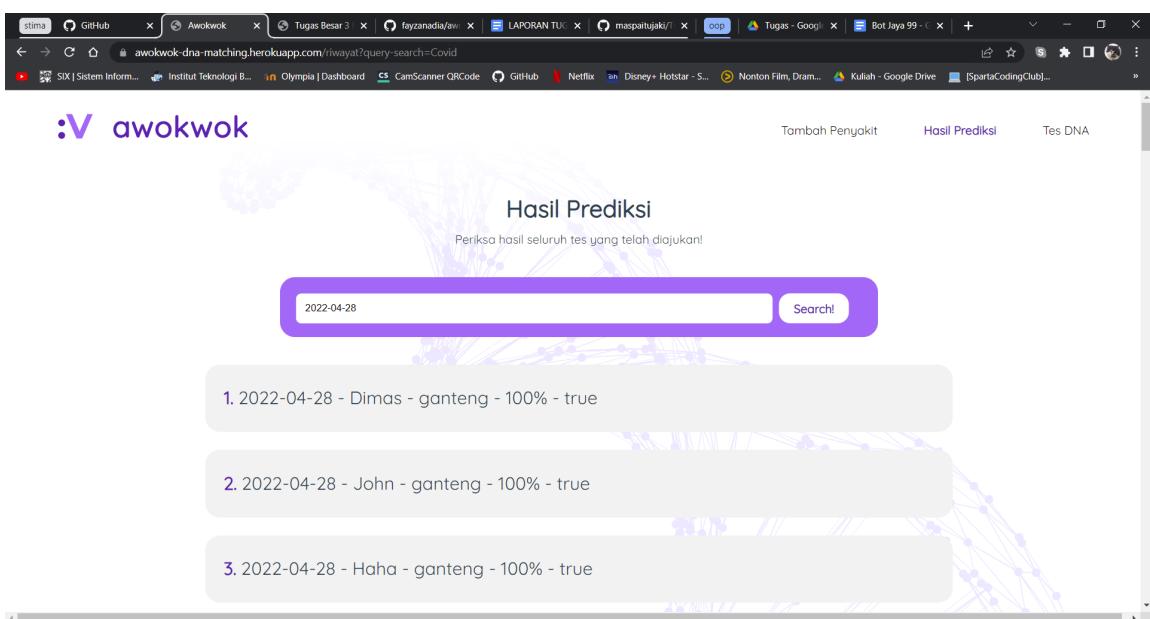


Gambar 14

Pencarian Riwayat Tes dengan Input Tanggal dan Nama Penyakit

Kasus 2

Pada pengujian kedua, kami melakukan pencarian riwayat tes dengan contoh masukan berupa tanggal saja.

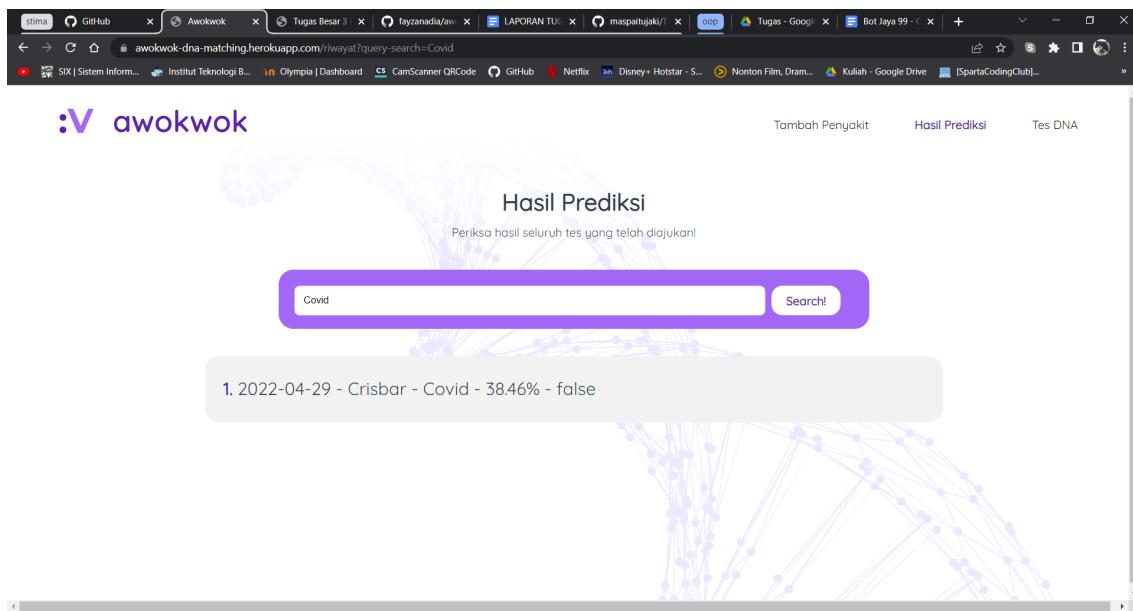


Gambar 15

Pencarian Riwayat Tes dengan Input Tanggal

Kasus 3

Pada pengujian ketiga, kami melakukan pencarian riwayat tes dengan contoh masukan berupa nama penyakit saja.



Gambar 16

Pencarian Riwayat Tes dengan Input Nama Penyakit

BAB V

KESIMPULAN DAN SARAN

Kesimpulan

Dengan menggunakan framework VueJs untuk frontend dan Golang untuk backend, kami telah membuat aplikasi berbasis Web untuk melakukan pendekripsi penyakit terhadap rantai DNA seseorang. Setelah dilakukan uji coba serta studi kasus, dapat disimpulkan bahwa aplikasi yang kami buat dapat melakukan pendekripsi pada rantai seseorang dengan menggunakan algoritma *string/pattern matching* untuk menentukan apakah terdapat penyakit di dalamnya sebagaimana yang telah diharapkan.

Saran

Setelah menyelesaikan pengembangan tugas besar ini, beberapa saran yang dapat kami berikan adalah lebih mempelajari bagaimana memprogram dalam bahasa Golang untuk membuat algoritma pada backend aplikasi web yang kami buat. Lebih memahami bagaimana menggunakan framework VueJs dalam membuat bagian frontend sebuah aplikasi web. Lebih memahami tentang deployment baik untuk backend maupun frontend. Jika ketiga saran di atas dapat diperbaiki kedepannya maka pengembangan tugas dan proses debugging akan menjadi lebih cepat.

TAUTAN REPOSITORY GITHUB

- Link github spesifikasi
https://github.com/maspaitujaki/Tubes3_13520001
- Link github backend for deployment
<https://github.com/maspaitujaki/dna-matching-api>
- Link github frontend for deployment
<https://github.com/fayzanadia/awokwok-dna-matching>
- Link deployment
<https://awokwok-dna-matching.herokuapp.com>

TAUTAN VIDEO DEMO

https://youtu.be/Wtx9P_odyN4

DAFTAR PUSTAKA

Munir, R. (2021). Pencocokan string (String matching/pattern matching)[PDF]. Institut Teknologi Bandung. Diakses pada 28 April 2022 pukul 23.08 WIB.
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf>

Munir, R. (2021). Pencocokan string dengan Regular Expression (Regex)[PDF]. Institut Teknologi Bandung. Diakses pada 28 April 2022 pukul 23.08 WIB.
<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2018-2019/String-Matching-dengan-Regex-2019.pdf>