

# **LAPORAN TUGAS BESAR 3**

## **IF2211 STRATEGI ALGORITMA**

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



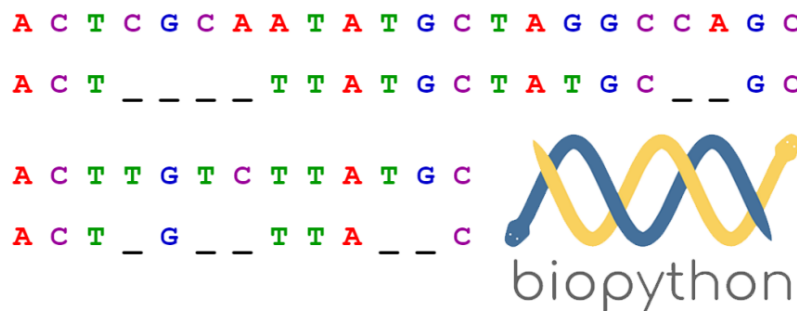
### **Kelompok dnAAAAA**

Hilya Fadhilah Imania	13520024
Jova Andres Riski Sirait	13520072
Adelline Kania Setiyawan	13520084

**Program Studi Sarjana Teknik Informatika**  
**Sekolah Teknik Elektro dan Informatika**  
**Institut Teknologi Bandung**  
**2022**

## 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.



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: ATTCGTAAGTAAAGTTA. 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.

## II. Landasan Teori

### A. Deskripsi Singkat Algoritma KMP, BM, dan Regex

#### *Algoritma KMP (Knuth-Morris-Pratt)*

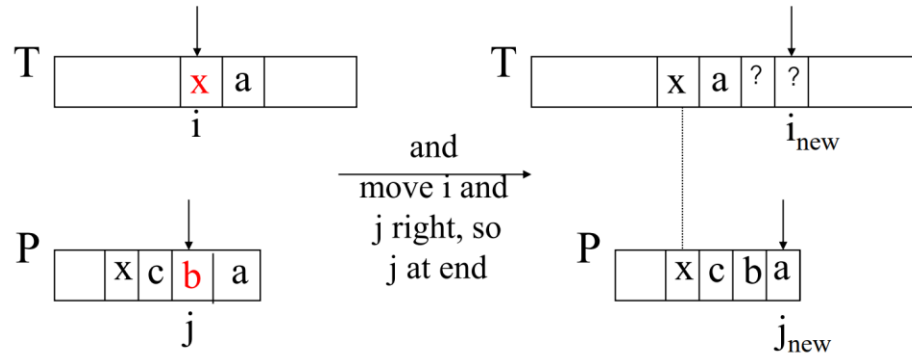
Algoritma KMP adalah salah satu algoritma *pattern matching* yang mencari suatu *pattern* dengan *left-to-right order*. Algoritma KMP sama dengan algoritma brute force dalam pola pencarian *pattern*-nya. Namun, jika pada algoritma *brute force* perpindahan untuk melakukan pengecekan *pattern* selalu hanya 1, pada algoritma KMP ini, perpindahannya bisa lebih dari satu sehingga menghemat ruang pencarian dan jauh lebih baik dari algoritma *brute force*.

Perpindahan pengecekan ini bergantung pada *mismatch* yang terdapat antara text dan pattern. Misal, *mismatch* terjadi pada *pattern* P pada posisi j sehingga jumlah perpindahan yang dilakukan adalah sesuai dengan panjang prefix  $P[0..j-1]$  terpanjang yang merupakan sufix  $P[1..j-1]$ . Sebelum melakukan pencarian dengan algoritma KMP, terlebih dahulu dapat dibuat suatu tabel yang menampung panjang prefix yang merupakan sufix terpanjang untuk setiap prefix *pattern*.

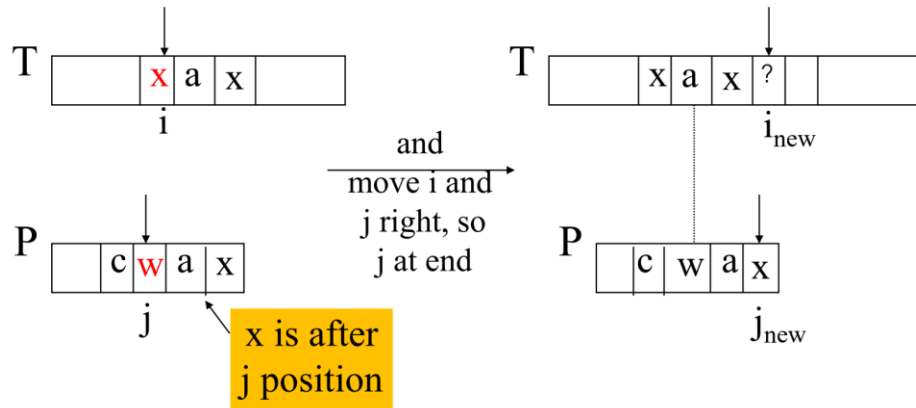
#### *Algoritma BM (Boyer-Moore)*

Algoritma BM adalah salah satu algoritma *pattern matching* yang menggunakan 2 teknik, yaitu:

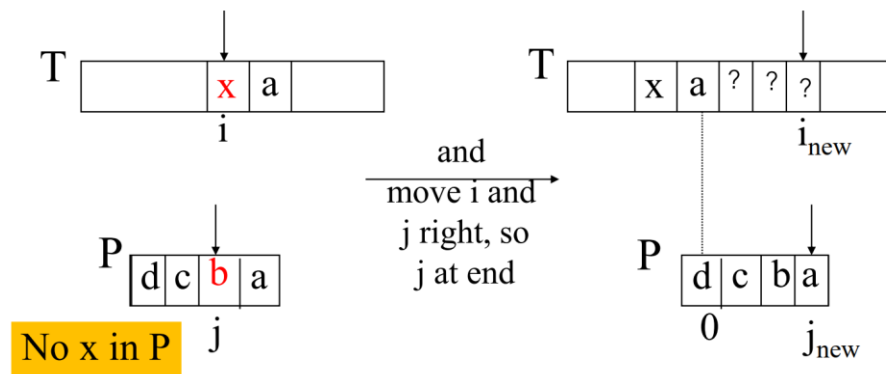
1. The *looking-glass* technique, yaitu mencari *pattern* pada teks dengan pola yang bergerak dari akhir *pattern*
2. The *character-jump* technique, yaitu teknik yang dilakukan ketika terdapat karakter pada *pattern* yang tidak sama dengan teks. Terdapat 3 skema, yaitu
  - Jika terdapat karakter x yang tidak sesuai dengan pattern, namun terdapat pada pattern, maka pattern akan digeser hingga karakter yang sama tersebut sejajar



- Jika terdapat karakter x yang tidak sesuai dengan pattern, namun terdapat pada pattern, tetapi tidak memungkinkan untuk melakukan pergeseran, maka akan digeser hanya 1 karakter



- Jika kedua kasus tersebut tidak terpenuhi, maka *pattern* akan digeser sebanyak karakter pada *pattern*



### ***Regex (Regular Expression)***

Regex adalah notasi pada suatu *pattern* yang mendefinisikan sebuah pola pencarian sehingga dapat membantu untuk melakukan *pattern matching*. Terdapat berbagai tanda-

tanda baca yang mengidentifikasika suatu notasi dari regex. Proses pencocokan atau *pattern matching* menggunakan regex bukanlah pencocokan yang menghasilkan hasil yang *exact* sehingga regex bukanlah *exact matching*. Namun, hal ini dapat memberikan hasil pencocokan yang lebih luas.

## **B. Penjelasan Singkat Mengenai Aplikasi Web yang Dibangun**

Aplikasi yang dibuat terdiri dari *frontend* dan *backend*. Implementasi *frontend* menggunakan Vue, sedangkan implementasi *backend* menggunakan Golang. Penyimpanan data menggunakan basis data MongoDB. *Frontend* mencakup tampilan website kepada pengguna dan meminta input dari pengguna untuk melakukan penambahan penyakit, tes DNA, dan pencarian filter untuk menampilkan hasil tes DNA yang ingin dicari.

### III. Analisis Pemecahan Masalah

#### C. Langkah Penyelesaian Masalah Setiap Fitur

1. Aplikasi dapat menerima input penyakit baru berupa nama penyakit dan sequence DNA-nya (dan dimasukkan ke dalam database).

Pada *frontend*, pengguna dapat menambahkan data penyakit baru yang terdiri dari nama penyakit dan sequence DNA-nya dalam bentuk file txt. Kemudian, sebelum melakukan penyimpanan ke basis data, program di *backend* akan melakukan pengecekan terlebih dahulu apakah sequence DNA pada file tersebut valid atau tidak. Suatu sequence DNA dikatakan valid jika tidak mengandung huruf kecil, spasi antar hruuf (spasi untuk *new line* dianggap valid), serta sequence DNA hanya mengandung huruf A, G, C, dan T saja.

Apabila sequence DNA valid, program di *backend* akan menyimpan data penyakit tersebut ke basis data dan mengirimkan respons kepada *frontend* bahwa sequence DNA valid dan data berhasil disimpan pada basis data. Namun, apabila sequence DNA tidak valid, program *backend* akan mengirimkan respons bahwa sequence DNA tidak valid. *Frontend* akan menampilkan respons yang diberikan dari *backend*.

2. Aplikasi dapat memprediksi seseorang menderita penyakit tertentu berdasarkan sequence DNA-nya.

Pada *frontend*, pengguna dapat melakukan tes DNA dengan memberikan input berupa nama pengguna, prediksi penyakit yang ingin dilakukan tes, metode apa yang ingin digunakan (Boyer Moore atau KMP), dan sequence DNA pasien dalam bentuk file txt. Pengguna juga dapat menentukan jenis algoritma *string matching* apa yang ingin dilakukan, KMP atau Boyer Moore. Kemudian, sama seperti pada fitur pertama, sebelum dilakukan tes DNA, program di *backend* akan melakukan pengecekan terlebih dahulu apakah sequence DNA pada file tersebut valid atau tidak. Suatu sequence DNA dikatakan valid jika tidak mengandung huruf kecil, spasi antar hruuf (spasi untuk *new line* dianggap valid), serta sequence DNA hanya mengandung huruf A, G, C, dan T saja.

Apabila sequence DNA valid, program di *backend* akan melakukan pencocokan *string matching* antara sequence DNA pasien dan penyakit yang ingin di tes

menggunakan algoritma yang diinginkan. Sequence DNA pasien diasumsikan selalu lebih panjang dari pada sequence DNA penyakit. Setelah berhasil melakukan pencocokan, program di *backend* akan memberikan respons kepada *frontend* apakah tes DNA yang dilakukan cocok atau tidak.

Namun, apabila sequence DNA tidak valid, program *backend* akan mengirimkan respons bahwa sequence DNA tidak valid dan tes DNA tidak berhasil dilakukan. *Frontend* akan menampilkan respons yang diberikan dari *backend*.

3. Aplikasi memiliki halaman yang menampilkan urutan hasil prediksi dengan kolom pencarian di dalamnya. Kolom pencarian bekerja sebagai filter dalam menampilkan hasil.

Pada *frontend*, terdapat suatu halaman dimana pengguna dapat melakukan query atau pencarian hasil tes DNA berdasarkan tanggal tes atau nama penyakit. Pencarian dapat menerima masukan dengan struktur `<tanggal_prediksi><spasi><nama_penakit>`, contoh “13 April 2022 HIV”. Program juga dapat menerima berupa masukan tanggal prediksi atau nama penyakit saja.

Untuk melakukan pencarian, query yang dimasukkan pengguna akan dikirimkan sebagai parameter ke *backend*. Program akan mengecek terlebih dahulu apakah struktur pencarian sudah memenuhi menggunakan implementasi dari regex. Apabila sudah sesuai, program akan melakukan query terhadap dokumen-dokumen tes DNA yang terdapat pada basis data kemudian mengirimkan data-data tersebut ke *frontend* untuk ditampilkan kepada pengguna.

4. **(Bonus)** Menghitung tingkat kemiripan DNA pengguna dengan DNA penyakit pada tes DNA

Untuk melakukan perhitungan tingkat kemiripan DNA pengguna dengan DNA penyakit digunakan algoritma LCS atau *least common subsequence*. Perhitungan menggunakan algoritma ini akan dilakukan ketika hasil prediksi seseorang menderita penyakit tertentu berdasarkan sequence DNA-nya pada poin 2 bernilai salah. Apabila bernilai benar, program tidak akan melakukan pengecekan menggunakan algoritma



LCS lagi karena dapat dipastikan bahwa tingkat kemiripan DNA pengguna dengan DNA penyakit bernilai sama dengan 1.

#### **D. Fitur Fungsional dan Arsitektur Aplikasi Web yang Dibangun**

Fitur fungsional dari Aplikasi Web yang dibangun adalah:

- 1) Fitur 1 – Menambahkan data penyakit baru
- 2) Fitur 2 – Melakukan Tes DNA
- 3) Fitur 3 – Menampilkan hasil Tes DNA yang dilakukan
- 4) Fitur 4 – Menampilkan hasil Tes DNA berdasarkan query yang dimasukkan

Aplikasi yang dibuat terdiri dari *frontend* dan *backend*. Implementasi *frontend* menggunakan Vue, sedangkan implementasi *backend* menggunakan Golang. Penyimpanan data menggunakan basis data MongoDB. *Frontend* mencakup tampilan website kepada pengguna dan meminta input dari pengguna untuk melakukan penambahan penyakit, tes DNA, dan pencarian filter untuk menampilkan hasil tes DNA yang ingin dicari.

## IV. Implementasi dan Pengujian

### A. Spesifikasi Teknis Program (struktur data, fungsi, prosedur yang ada)

#### 1) Struktur Data

Struktur data utama dari aplikasi web ini adalah struktur *database* yang berupa dokumen atau *collection*. *Database* memiliki dua *collections*, yaitu *collection* penyakit dan tes\_dna. *Collection* penyakit menyimpan data berupa id penyakit, nama penyakit, dan sequence DNA penyakit. Sedangkan *collection* tes\_dna menyimpan data berupa id, nama pengguna yang melakukan tes DNA, prediksi penyakit, tanggal tes, sequence DNA pengguna, hasil tes DNA, dan persentase kemiripan.

#### 2) Fungsi dan Prosedur

##### a. Fungsi dan Prosedur untuk Mengelola Basis Data

##### 1. CreatePenyakit

Fungsi ini akan menerima request berupa *form data* yang berisi nama penyakit dan sequence DNA (berupa file). Fungsi ini kemudian akan mengecek terlebih dahulu apakah sequenceDNA yang diberikan merupakan sequence DNA yang valid. Fungsi ini akan melakukan penyimpanan data tersebut ke database. Apabila semuanya berhasil, maka fungsi ini akan mengirimkan respons berupa pesan sukses beserta data penyakit yang dibuat.

##### 2. CreateTesDNA

Fungsi ini akan menerima request berupa *form data* yang berisi nama pengguna, prediksi penyakit, metode *string matching*, dan sequence DNA (berupa file). Fungsi ini terlebih dahulu akan mengecek apakah sequenceDNA yang diberikan merupakan sequence DNA yang valid. Kemudian, akan dicari sequence DNA penyakit berdasarkan prediksi penyakit. Sequence DNA penyakit dan pengguna akan dicocokkan melalui algoritma *string matching* berdasarkan metode yang dipilihnya sehingga akan mengembalikan hasil tes DNA dan persentase kemiripannya. Fungsi ini akan melakukan penyimpanan data tersebut ke database. Apabila semuanya berhasil, maka fungsi ini akan mengirimkan respons berupa pesan sukses beserta hasil tes DNA yang dilakukan.

##### 3. GetPenyakit

Fungsi ini akan mengirimkan respons berupa data penyakit yang diambil dari database berdasarkan nama penyakit yang dicarinya.

#### 4. GetAllTesDNA

Fungsi ini akan mengirimkan respons berupa semua data penyakit yang diambil dari database berdasarkan query yang diberikan. Query akan dicek terlebih dahulu apakah query memenuhi format yang dibolehkan dengan menggunakan regex.

### b. Fungsi dan Prosedur untuk Regex

#### 1. splitWord

Fungsi ini akan mengembalikan sebuah array hasil pemisahan suatu kata yang dijadikan parameternya serta dipisahkan untuk setiap “-”, “ ”, atau “.”

#### 2. ReadQuery

Fungsi ini akan mengembalikan sebuah boolean beserta pesan kesalahan query atau tanggal dan nama penyakit berdasarkan string query yang diberikan. Fungsi ini akan terlebih dahulu melakukan splitWord terhadap suatu query kemudian menentukan apakah query tersebut memenuhi format yang diperbolehkan atau tidak.

#### 3. ValidateDNASequence

Fungsi ini akan memvalidasi sequence DNA yang diberikan. Suatu sequence DNA dikatakan valid apabila tidak mengandung huruf kecil, spasi antar huruf (spasi untuk *new line* dianggap valid), serta sequence DNA hanya mengandung huruf A, G, C, dan T saja

### c. Fungsi dan Prosedur untuk *String/Pattern Matching*

#### 1. BmStringMatching dan BuildLast

Fungsi BmStringMatching adalah fungsi yang melakukan string matching terhadap dua buah pattern/string dengan menerapkan algoritma Boyer-Moore. Fungsi BmStringMatching akan menggunakan fungsi BuildLast untuk mencatat kemunculan dari tiap char dalam representasi ASCII yang terdapat dalam pattern. Fungsi BmStringMatching ini akan me-return sebuah boolean yang menyatakan hasil pencocokan string matching, integer yang menyatakan index pertama pattern sequence DNA pasien dan penyakit cocok (akan bernilai

0 jika tidak cocok), serta integer yang menyatakan jumlah perbandingan yang dilakukan.

## 2. StringMatching dan BorderFunction

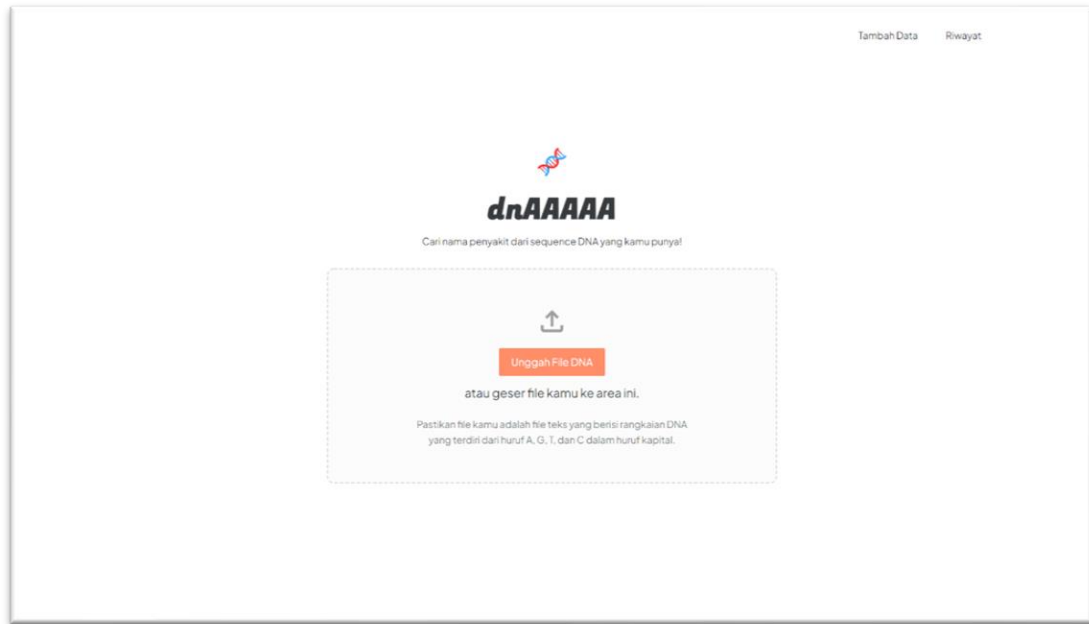
Fungsi StringMatching adalah fungsi yang melakukan string matching terhadap dua buah pattern/string dengan menerapkan algoritma Knuth-Morris-Pratt. Fungsi StringMatching akan menggunakan fungsi BorderFunction untuk menentukan nilai fungsi border setiap sufiks patternnya. Fungsi StringMatching ini akan me-*return* sebuah boolean yang menyatakan hasil pencocokan *string matching*, integer yang menyatakan index pertama pattern sequence DNA pasien dan penyakit cocok (akan bernilai 0 jika tidak cocok), serta integer yang menyatakan jumlah perbandingan yang dilakukan.

## 3. LongestCommonSubsequence dan SequenceSimilarity

Fungsi SequenceSimilarity adalah fungsi yang mengembalikan sebuah nilai float yang menyatakan kemiripan antara pattern sequence DNA dan pasien. Fungsi ini akan menggunakan fungsi LongestCommonSubsequence yang mengembalikan panjang pattern terpanjang termirip antara sequence DNA dan pasien. Nilai dari fungsi LongestCommonSubsequence ini kemudian akan dibagi dengan panjang maksimal antara sequence DNA pasien dan penyakit.

# B. Penjelasan Tata Cara Penggunaan Program

1. Jalankan program secara lokal sesuai instruksi pada Readme atau kunjungi tautan <https://dnaaaaa.jovaandreas.me/> untuk mengakses program secara online. Berikut adalah tampilan program pada saat pertama kali dibuka.



- Pengguna dapat langsung mengunggah file yang menyimpan sequence DNA dengan menekan tombol ataupun melakukan *drag-and-drop* ke kotak yang telah disediakan.
2. Setelah mengunggah file yang sesuai, pengguna akan diarahkan ke laman pengisian data yang dibutuhkan untuk melakukan tes DNA.

The screenshot shows the 'Tes DNA' form. It has four main input fields: 'Nama \*' with the value 'Dipsy', 'Prediksi Penyakit \*' with a dropdown menu showing 'hiv', 'Metode Pencocokan String \*' with a dropdown menu showing 'Boyer Moore', and 'File Sequence DNA \*' with a text input showing 'mydna.txt' and a 'Pilih File' button. At the bottom right is a blue button with a magnifying glass icon and the text 'Cocokkan DNA'.

Data yang akan diisi selanjutnya adalah nama pengguna, prediksi penyakit, dan metode pencocokan string yang tersedia yaitu Knuth-Morris-Pratt atau Boyer Moore. Setelah menekan tombol **Cocokkan DNA**, maka pengguna akan mendapatkan hasil tes DNA seperti berikut ini.



Hasil yang akan dikembalikan adalah hasil tes (Positif/Negatif) beserta tingkat kemiripan sequence DNA-nya.

3. Pengguna juga dapat menambahkan penyakit baru pada dengan menekan tombol **Tambah Data** pada navigasi.

Tambahkan DNA Penyakit Baru

Nama Penyakit \*

hemofilia

File Sequence DNA \*

mydna.txt

Pilih File

Tambahkan

Pengguna akan memasukkan nama penyakit baru dan juga file yang menyimpan sequence DNA dari penyakit tersebut. Jika data yang diberikan sesuai, akan tampil pesan seperti berikut.

Tambahkan DNA Penyakit Baru

Berhasil

Data penyakit berhasil dimasukkan

Tanggal : 4/28/2022

Penyakit : hemofilia

- Pengguna juga dapat mengecek hasil tes DNA yang pernah dilakukan dengan menekan tombol **Riwayat** pada navigasi. Hasil yang ditampilkan adalah seperti berikut.

Masukkan pencarian...				
Tanggal	Nama	Prediksi	Hasil	Kemiripan
April 14, 2022	tes nih ya4	hiv	Negatif	?
April 14, 2022	tes nih ya4	paru-paru	Negatif	?
April 23, 2022	tes nih ya4	paru-paru	Positif	?
April 23, 2022	tes nih ya4	paru-paru	Positif	?
April 23, 2022	tes nih ya4	paru-paru	Negatif	?
April 24, 2022	tes nih ya4	paru-paru	Negatif	?
April 24, 2022	tes nih ya4	paru-paru	Negatif	9.14%
April 25, 2022	Adelline	Asma	Positif	89.47%
April 25, 2022	Jova	Asma	Positif	85.00%
April 27, 2022	Hilya	COVID 19 Varian Deltacron	Positif	100.00%

Pengguna juga dapat mencari riwayat Cek DNA secara spesifik dengan memasukkan query pada box yang telah disediakan. Contoh query yang dapat diproses dengan benar oleh program adalah <Tanggal> saja, <Nama Penyakit> saja, atau <Tanggal> <Nama Penyakit>. Berikut adalah contoh hasil pencariannya.

Asma				
Tanggal	Nama	Prediksi	Hasil	Kemiripan
April 25, 2022	Adelline	Asma	Positif	89.47%
April 25, 2022	Jova	Asma	Positif	85.00%

## C. Hasil Pengujian

### 1. Menambah data penyakit

Sequence DNA valid

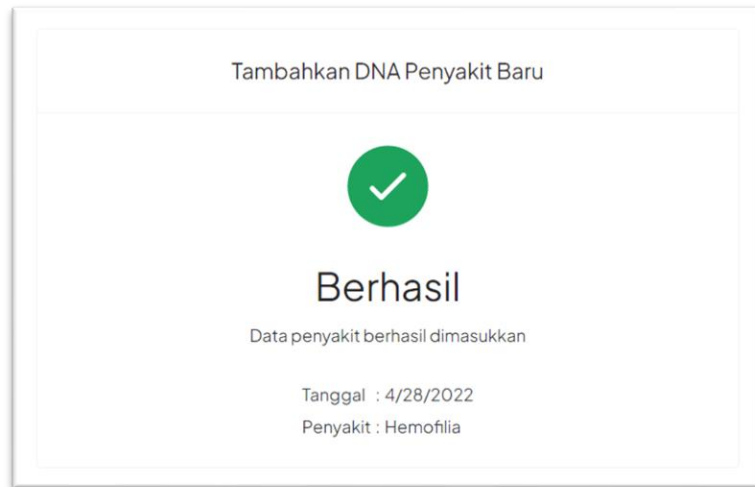


Penyakit: Hemofilia

Sequence:

GCAGTCAGCTAGCGATGCTAGCTACGTACGTACGTAGCTAGCTAGCTACG  
GCGGGCGATCTACTACTTCTACACCAAAAATTT

Hasil yang didapatkan:



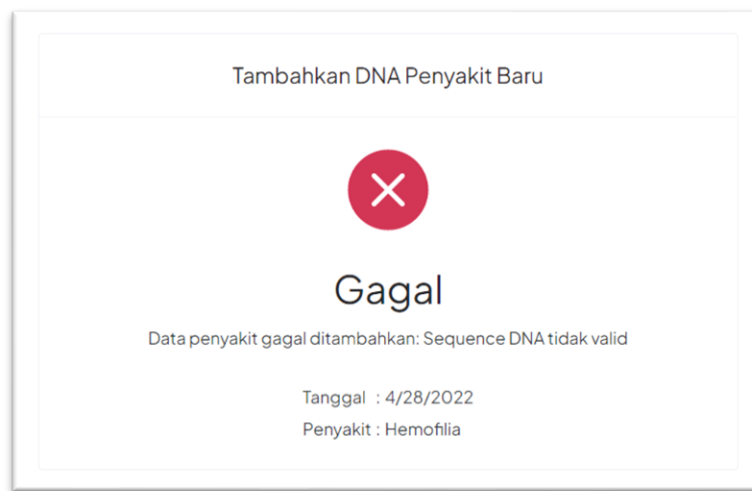
Sequence DNA tidak valid

Penyakit: Hemofilia

Sequence:

GCAGTCAGCTAGCGATGCTAGCTACGTACGTACGTAGCTFGCTAGCTACG  
GCGGGCGATCTACTACTTCTACACCAAAAATTT

Hasil yang didapatkan:



## 2. Mencocokkan DNA pengguna dengan penyakit

Penyakit : Hemofilia (Kasus Positif)

Metode : Knutt-Morris-Pratt

Hasil yang didapatkan:



Penyakit : hiv (Kasus Negatif)

Metode : Knutt-Morris-Pratt

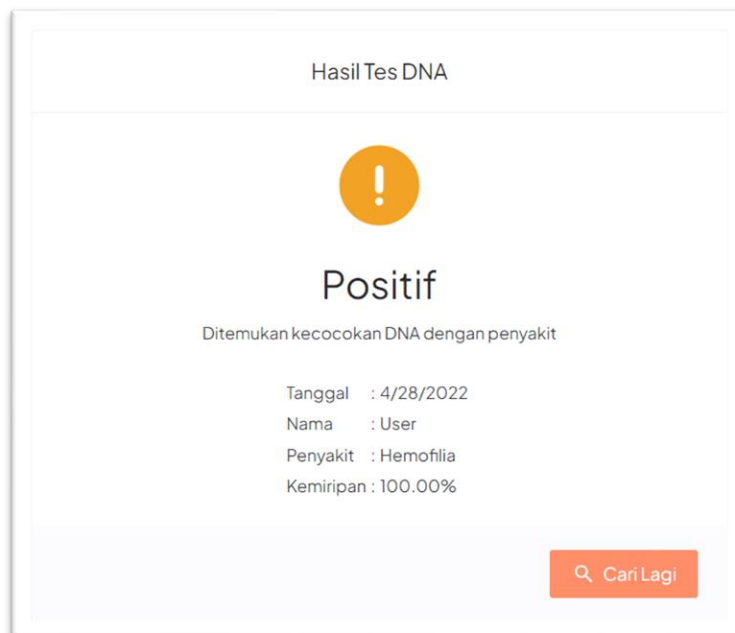
Hasil yang didapatkan:



Penyakit : Hemofilia (Kasus Positif, Kemiripan 100%)

Metode : Boyer Moore

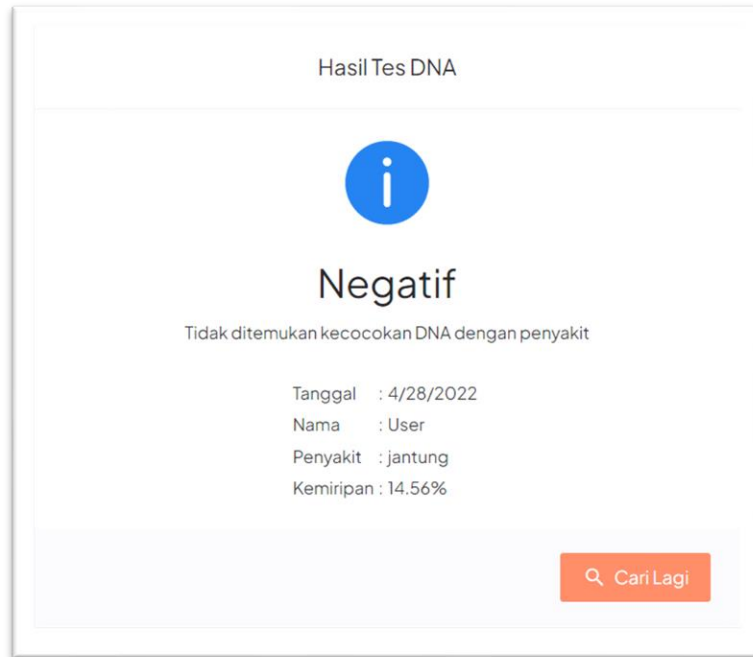
Hasil yang didapatkan:



Penyakit : jantung (Kasus Negatif)

Metode : Boyer Moore

Hasil yang didapatkan:



Kasus Sequence DNA salah



### 3. Mencari riwayat tes DNA dengan query

Query: 25 April 2022 (hanya tanggal)

Riwayat Cek DNA				
25 April 2022				
Tanggal	Nama	Prediksi	Hasil	Kemiripan
April 25, 2022	Adelline	Asma	Positif	89.47%
April 25, 2022	Jova	Asma	Positif	85.00%

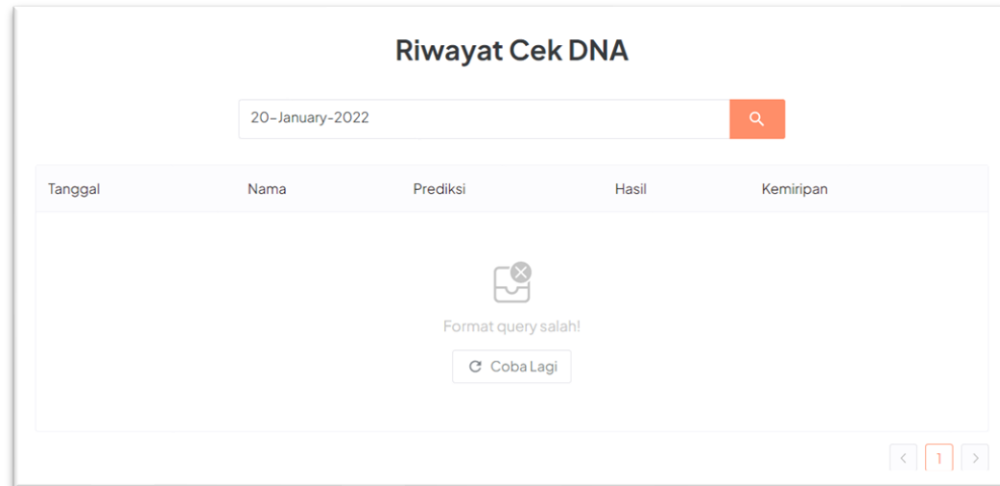
Query: jantung (hanya nama penyakit)

Riwayat Cek DNA				
jantung				
Tanggal	Nama	Prediksi	Hasil	Kemiripan
April 27, 2022	Hilya	jantung	Negatif	46.30%
April 28, 2022	User	jantung	Negatif	14.56%

Query: 24-04-2022 paru-paru (tanggal dan nama penyakit)

Riwayat Cek DNA				
24-04-2022 paru-paru				
Tanggal	Nama	Prediksi	Hasil	Kemiripan
April 24, 2022	tes nih ya4	paru-paru	Negatif	30.23%
April 24, 2022	tes nih ya4	paru-paru	Negatif	9.14%

Query: 20-January-2022 (Format query salah)



#### D. Analisis Hasil Pengujian

Berdasarkan hasil pengujian, didapatkan analisis sebagai berikut

1. Aplikasi website berhasil melakukan penambahan data penyakit untuk sequence DNA yang valid, yaitu sequence yang tidak mengandung huruf kecil, spasi antar huruf (spasi untuk *new line* dianggap valid), serta sequence DNA hanya mengandung huruf A, G, C, dan T saja. Pada hasil pengujian poin ke (1) dengan sequence DNA tidak valid, akan didapatkan pesan error seperti yang terlihat pada *screenshot* bagian tersebut.
2. Aplikasi website berhasil melakukan pencocokan DNA antara penyakit dan pasien dengan metode Knutt-Morris-Pratt ataupun Boyer Moore. Hasil tes DNA dapat bernilai positif maupun negatif serta menampilkan persentase kemiripan sequence DNA antara penyakit dan pasien. Hasil tes dinyatakan positif jika persentase kemiripannya adalah di atas 80%. Namun, akan dilakukan pengecekan terlebih dahulu terhadap sequence DNA pasien, apakah valid atau tidak. Jika tidak valid, maka akan terdapat pesan error seperti pada *screenshot* poin ke (2) bagian terakhir, kasus sequence DNA salah.
3. Aplikasi website berhasil menampilkan semua riwayat tes DNA yang telah dilakukan. Selain itu, pengguna dapat memberikan input query kemudian aplikasi website akan menampilkan semua riwayat tes DNA berdasarkan query yang dimasukkan. Apabila query yang dimasukkan salah, maka akan terdapat pesan berupa "Format query salah!".

## **V. Kesimpulan, Saran, dan Komentar**

Dari pengerjaan Tugas Besar 3 Strategi Algoritma ini, kami telah berhasil membuat aplikasi berbasis web yang dapat melakukan pencocokan sequence DNA manusia dengan sequence DNA penyakit untuk mendeteksi adanya penyakit pada manusia tersebut. Pencocokan DNA penyakit dan DNA manusia dilakukan dengan algoritma Knutt-Morris-Pratt ataupun Boyer Moore. Aplikasi juga dapat menghitung tingkat kemiripannya dengan algoritma LCS. Aplikasi juga dapat menambahkan data penyakit dengan memasukkan nama dan file text berisi sequence DNA yang valid. Untuk mengecek hasil tes DNA yang pernah dilakukan, terdapat halaman riwayat dengan memasukkan filter tanggal, nama penyakit ataupun keduanya. Aplikasi juga telah berhasil dideploy dan dapat diakses dari tautan yang diberikan pada panduan penggunaan aplikasi.

Untuk pengembangan kedepannya, disarankan untuk penanganan regex dapat ditingkatkan untuk menambah format query yang dapat diterima aplikasi sehingga dapat lebih fleksibel. Tampilan website juga masih dapat ditingkatkan lagi supaya lebih enak dilihat dan digunakan.

Dari Tugas Besar 3 kali ini kami banyak mempelajari hal baru seperti proses pembuatan website dari pembuatan frontend, backend, integrasi, sampai melakukan setup untuk deploy ke cloud.

## **VI. Daftar Pustaka**

- [1] Slide Kuliah IF2211 Strategi Algoritma – Pencocokan String (String Matching/Pattern Matching)
- [2] Slide Kuliah IF2211 Strategi Algoritma – Pencocokan String dengan Regular Expression (Regex)

## **VII. Lampiran**

- 1. Link Github : [https://github.com/adellinekania/Tubes3\\_dnAAAAA](https://github.com/adellinekania/Tubes3_dnAAAAA)
- 2. Link Website : <https://dnaaaaa.jovaandreas.me>
- 3. Link Video : <https://www.youtube.com/watch?v=F693cQzdGYc>