

Tugas Besar III IF2211 Strategi Algoritma  
Semester II Tahun 2020/2021  
**Penerapan String Matching dan Regular Expression dalam  
DNA Pattern Matching**

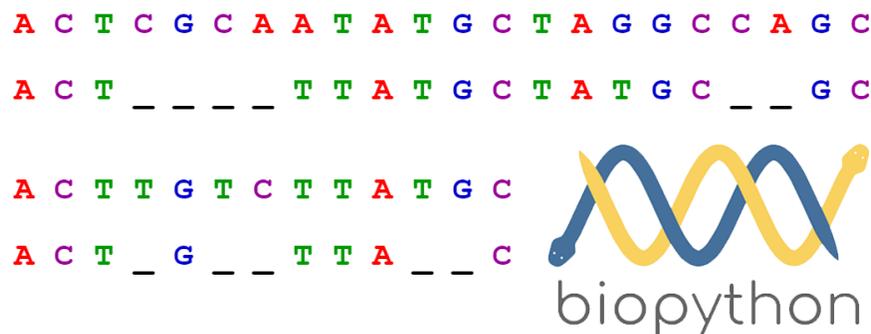
**Batas pengumpulan : 29 April 2022 pukul 23.59 WIB**

**Arsip pengumpulan :**

- Source program yang bisa dijalankan disertai *readme.txt*
- Laporan (*soft copy*)

**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: 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.

### Deskripsi tugas:

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

### Fitur-Fitur Aplikasi:

1. Aplikasi dapat menerima *input* penyakit baru berupa nama penyakit dan *sequence* DNA-nya (dan dimasukkan ke dalam *database*).
  - a. Implementasi *input sequence* DNA dalam bentuk *file*.
  - b. Dilakukan sanitasi *input* menggunakan **regex** untuk memastikan bahwa masukan merupakan *sequence* DNA yang valid (tidak boleh ada huruf kecil, tidak boleh ada huruf selain AGCT, dan tidak ada spasi).
  - c. Contoh *input* penyakit:

**Tambahkan Penyakit**

Nama Penyakit:

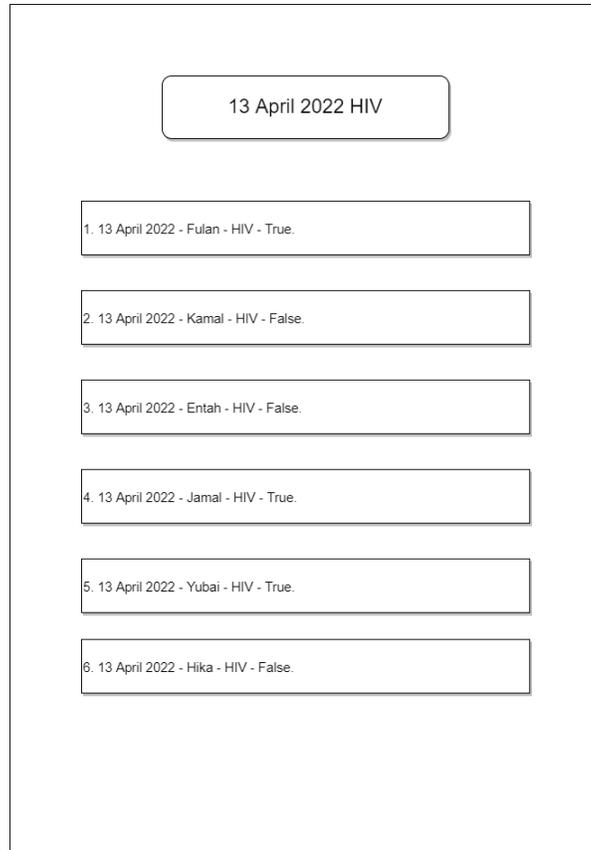
Sequence DNA:

*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:

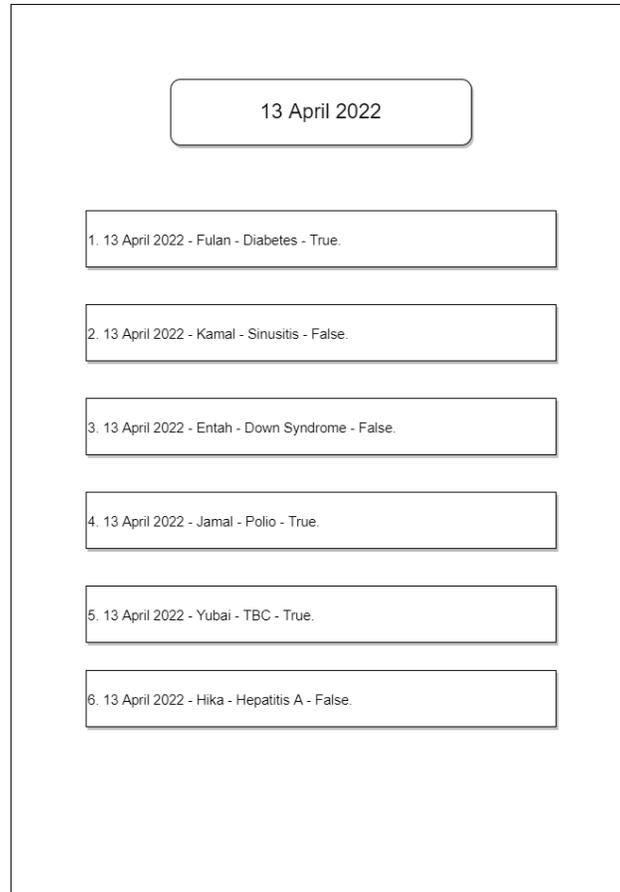
Gambar 3. Ilustrasi Prediksi

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: <tanggal\_prediksi><spasi><nama\_penyakit>, contoh “13 April 2022 HIV”. **Format penanggalan dibebaskan**, jika bisa menerima >1 format lebih baik.
  - b. Kolom pencarian dapat menerima masukan hanya tanggal ataupun hanya nama penyakit. Fitur ini diimplementasikan menggunakan **regex**.
  - c. Contoh ilustrasi:
    - i. Masukan tanggal dan nama penyakit



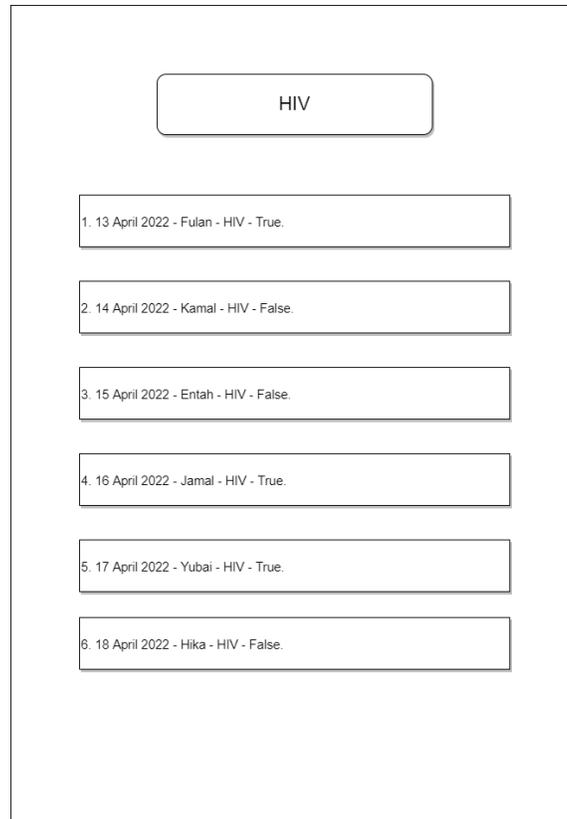
*Gambar 4. Ilustrasi Interaksi 1*

- ii. Masukan hanya tanggal



*Gambar 5. Ilustrasi Interaksi 2*

- iii. Masukkan hanya nama penyakit



*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:

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.

### Lain-lain:

1. Anda dapat menambahkan fitur-fitur lain yang menunjang program yang anda buat (unsur kreativitas).

2. Tugas dikerjakan berkelompok, minimal 2 orang dan maksimal 3 orang, boleh lintas kelas namun **tidak boleh sekelompok dengan orang yang sama dengan tubes stima sebelumnya.**
3. Semua kelompok harap mengisi data kelompok mereka pada link <https://bit.ly/KelompokStima12022>
4. Batas akhir pengisian anggota kelompok adalah **17 April 2022**. Mahasiswa yang belum mendapatkan kelompok setelah tanggal ini akan diacak kelompoknya.
5. Anda harus membuat aplikasi dan program ini sendiri kecuali library file dan regex, tetapi belajar dari contoh-contoh program serupa yang sudah ada tidak dilarang (tidak boleh melakukan plagiasi source code dari program orang lain). Program harus dibuat sendiri, tidak boleh sama dengan teman.
6. Program harus modular dan mengandung komentar yang jelas.
7. Dilarang menggunakan kode program yang diunduh dari Internet. Mahasiswa harus membuat program sendiri, tetapi belajar dari program yang sudah ada tidak dilarang.
8. Batas akhir pengumpulan tugas adalah **29 April 2022 23:59 WIB**. Keterlambatan dalam mengumpulkan akan diberi penalti pengurangan skor yang cukup signifikan.
9. Semua pertanyaan menyangkut tugas ini dapat dikomunikasikan lewat QnA yang bisa diakses pada <https://bit.ly/QnAStima2022>
10. **Bonus (maksimal 15 poin):**
  - a. Mendeploy aplikasi web yang telah dibangun (hosting provider dibebaskan). Deployment website harus dipertahankan sampai demo tugas besar.
  - b. Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke Youtube. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Pada waktu demo aplikasi di depan asisten, mahasiswa mengakses video Youtube tersebut dan memutarinya di depan asisten. Beberapa contoh video tubes tahun-tahun sebelumnya dapat dilihat di YouTube dengan menggunakan kata kunci “Tubes Stima”, “Tugas besar stima”, “strategi algoritma”, dll.
  - c. Membuat fitur tingkat kemiripan DNA pengguna dengan DNA penyakit pada tes DNA.
11. Demo akan dilakukan, tunggu informasi lanjut setelah waktu pengerjaan tugas berakhir.
12. Setiap anggota kelompok harus memahami seluruh program, termasuk bagian yang bukan bagian mereka.
13. Program disimpan dalam folder **Tubes3\_NIM** (jika menggunakan *repository* dapat mengubah nama *repository*-nya) dengan NIM merupakan NIM anggota terkecil. Berikut merupakan struktur dari isi folder tersebut.
  - a. Folder **src** berisi **source code**.
  - b. Folder **doc** berisi **laporan tugas besar** dengan format **nama\_kelompok.pdf**
  - c. Folder **test** berisi **data-data** atau **pengetahuan** awal yang dimiliki oleh aplikasi serta contoh file .txt sequence DNA penyakit. Apabila menggunakan database lokal, hasil dumpnya dapat disimpan di folder ini. Data-data awal maupun hasil dump database diberi prefix **seed-** pada nama file nya.

- d. README selengkap mungkin (dapat menjelaskan cara menjalankan *backend* dan *frontend* di lokal). Referensi README dapat diakses pada <https://github.com/ritaly/README-cheatsheet> atau referensi lain yang serupa.
14. Folder tersebut **di-zip** dengan format yang sama dengan nama folder dan dikumpulkan melalui <https://bit.ly/SubmisiStima3>

#### Isi laporan:

- **Cover:** Cover laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo “gajah” ganesha.
- **Bab 1:** Deskripsi tugas (dapat menyalin spesifikasi tugas ini).
- **Bab 2:** Landasan Teori.
  - Deskripsi singkat algoritma KMP, BM, dan Regex
  - Penjelasan singkat mengenai aplikasi web yang dibangun
- **Bab 3:** Analisis Pemecahan Masalah.
  - Langkah penyelesaian masalah setiap fitur
  - Fitur fungsional dan arsitektur aplikasi web yang dibangun
- **Bab 4:** Implementasi dan pengujian.
  - Spesifikasi teknis program (struktur data, fungsi, prosedur yang dibangun)
  - Penjelasan tata cara penggunaan program (interface program, fitur-fitur yang disediakan program, dan sebagainya)
  - Hasil pengujian (*screenshot* antarmuka dan skenario yang memperlihatkan berbagai kasus yang mencakup seluruh fitur pada aplikasi DNA Sequence Matching)
  - Analisis hasil pengujian
- **Bab 5:** Kesimpulan, saran, dan komentar/refleksi tentang tugas besar 3 ini.
- Daftar Pustaka.

#### Keterangan laporan:

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

### Penilaian:

1. **Bagian 1** : Laporan (25%)
  - a. Langkah penyelesaian masalah setiap fitur (10%)
  - b. Hasil pengujian dan analisis algoritma (10%)
  - c. Komponen-komponen lain dalam laporan (5%)
2. **Bagian 2** : Implementasi Program (75%)
  - a. Kebenaran program (30%)
  - b. Pemahaman terhadap cara kerja program (25%)
  - c. Interface, Features, dan Unsur Kreativitas (20%)
3. **Bagian 3** : Bonus (15%)
  - a. Melakukan *deployment* ke dalam website
  - b. Membuat video demonstrasi program
  - c. Membuat fitur tingkat kemiripan DNA pengguna dengan DNA penyakit

### Referensi Belajar:

- [Dasar-dasar Go](#)
- [Framework Backend Go Echo](#)
- [Framework Backend Go Fiber](#)
- [Framework Backend Go Gin](#)
- [Framework Backend Node.js Express](#)
- [Framework Backend Node.js Fastify](#)
- [Framework Backend Node.js Hapi](#)
- [Framework Frontend Vue.js](#)
- [Framework Frontend Angular](#)
- [Framework Frontend React](#)
- [Framework Frontend Next.js](#)
- [Connect to MySQL from Node.js](#)
- [Connect to MySQL from Node.js using Sequelize](#)
- [Connect to SQL from Go](#)
- [API Call using Axios](#)
- [Hamming Distance Explanation](#)
- [Levenshtein Distance Explanation](#)
- [LCS Explanation](#)

--- Selamat Mengerjakan! ---

“Aku ketika melihat spesifikasi program: :D *outside* D: *inside*”  
– Tito –

“Panik gak? Panik lah, masa enggak”  
– Hokki –

“String matching ya gais bukan Tinder matching”  
– Jojo –

“Semua badai pasti akan berlalu”  
– Girvin –

“ 5'– AGCACGATTATGGCA –3 ”  
– Christo –

“Kukira kita Adenine dan Thymine :(”  
– Dzaki –

“Jangan daftar IRK.”  
– Dimas –

“~~Success~~ Tubes is in our DNA 😎”  
– Shafira –