

Tugas Besar 3 IF2211 Strategi Algoritma  
Semester II tahun 2023/2024

**Pemanfaatan *Pattern Matching* dalam Membangun Sistem Deteksi Individu Berbasis Biometrik Melalui Citra Sidik Jari**

**Revisi 1** : Kamis, 9 Mei 2024, pukul 16.14 WIB (detail atribut berkas\_citra)

**Revisi 2** : Sabtu, 11 Mei 2024, pukul 13.53 WIB (kakas pembangunan yang disarankan)

**Revisi 3** : Rabu, 15 Mei 2024, pukul 19.24 WIB (lokasi penyimpanan dataset)

**Batas pengumpulan** : Sabtu, 25 Mei 2024 pukul 23.59 WIB

**Arsip pengumpulan**

- *Source* program yang bisa dijalankan disertai README
- Laporan (*soft copy*)

**Latar Belakang**



**Gambar 1.** Ilustrasi *fingerprint recognition* pada deteksi berbasis biometrik.

Sumber: <https://www.aratek.co/news/unlocking-the-secrets-of-fingerprint-recognition>

Di era digital ini, keamanan data dan akses menjadi semakin penting. Perkembangan teknologi membuka peluang untuk berbagai metode identifikasi yang canggih dan praktis. Beberapa metode umum yang sering digunakan seperti kata sandi atau pin, namun memiliki kelemahan seperti mudah terlupakan atau dicuri. Oleh karena itu, biometrik menjadi alternatif metode akses keamanan yang semakin populer. Salah satu teknologi biometrik yang banyak digunakan adalah identifikasi sidik jari. Sidik jari setiap orang memiliki pola yang unik dan tidak dapat ditiru, sehingga cocok untuk digunakan sebagai identitas individu.

*Pattern matching* merupakan teknik penting dalam sistem identifikasi sidik jari. Teknik ini digunakan untuk mencocokkan pola sidik jari yang ditangkap dengan pola sidik jari yang terdaftar di database. Algoritma *pattern matching* yang umum digunakan adalah

Bozorth dan Boyer-Moore. Algoritma ini memungkinkan sistem untuk mengenali sidik jari dengan cepat dan akurat, bahkan jika sidik jari yang ditangkap tidak sempurna.

Dengan menggabungkan teknologi identifikasi sidik jari dan pattern matching, dimungkinkan untuk membangun sistem identifikasi biometrik yang aman, handal, dan mudah digunakan. Sistem ini dapat diaplikasikan di berbagai bidang, seperti kontrol akses, absensi karyawan, dan verifikasi identitas dalam transaksi keuangan.

Di dalam Tugas Besar 3 ini, Anda diminta untuk mengimplementasikan sistem yang dapat melakukan identifikasi individu berbasis biometrik dengan menggunakan sidik jari. Metode yang akan digunakan untuk melakukan deteksi sidik jari adalah Boyer-Moore dan Knuth-Morris-Pratt. Selain itu, sistem ini akan dihubungkan dengan identitas sebuah individu melalui basis data sehingga harapannya terbentuk sebuah sistem yang dapat mengenali identitas seseorang secara lengkap hanya dengan menggunakan sidik jari.

## Penjelasan Implementasi

Pada tugas ini, Anda diminta untuk mengimplementasikan sebuah program yang dapat melakukan identifikasi biometrik berbasis sidik jari. Proses implementasi dilakukan dengan menggunakan algoritma Boyer-Moore dan Knuth-Morris-Pratt, sesuai dengan yang diajarkan pada materi dan salindia kuliah.

Secara sekilas, penggunaan algoritma *pattern matching* dalam mencocokkan sidik jari terdiri atas tiga tahapan utama dengan skema sebagai berikut.




**Gambar 2.** Skema implementasi konversi citra sidik jari.

Sumber: Dokumentasi Penulis

Gambar yang digunakan pada proses *pattern matching* kedua algoritma tersebut adalah gambar sidik jari penuh berukuran  $m \times n$  pixel yang diambil sebesar 30 pixel setiap kali proses pencocokan data. Untuk tugas ini, Anda **dibebaskan** untuk mengambil jumlah pixel asalkan **didasarkan pada alasan yang masuk akal** (dijelaskan pada laporan) dan **penanganan kasus ujung yang baik** (misal jika ternyata ukuran citra sidik jari tidak habis dibagi dengan ukuran pixel yang dipilih). Selanjutnya, data pixel tersebut akan dikonversi menjadi binary. Seperti yang mungkin Anda ketahui sesuai [materi kuliah](#) (ya kalau masuk kelas), karena binary hanya memiliki variasi karakter satu atau nol, maka proses *pattern matching* akan membuat proses pencocokan karakter menjadi lambat karena harus sering mengulangi proses pencocokan *pattern*. Cara yang dapat dilakukan untuk mengatasi hal tersebut adalah dengan mengelompokkan setiap baris kode biner per 8 bit sehingga membentuk karakter ASCII. Karakter ASCII 8-bit ini yang akan mewakili proses pencocokan dengan string data.

Untuk memberikan gambaran yang lebih jelas, berikut adalah contoh kasus citra sidik jari dan proses serta sampel hasil yang didapatkan. Dataset yang digunakan adalah dataset citra sidik jari yang terdapat pada bagian referensi.

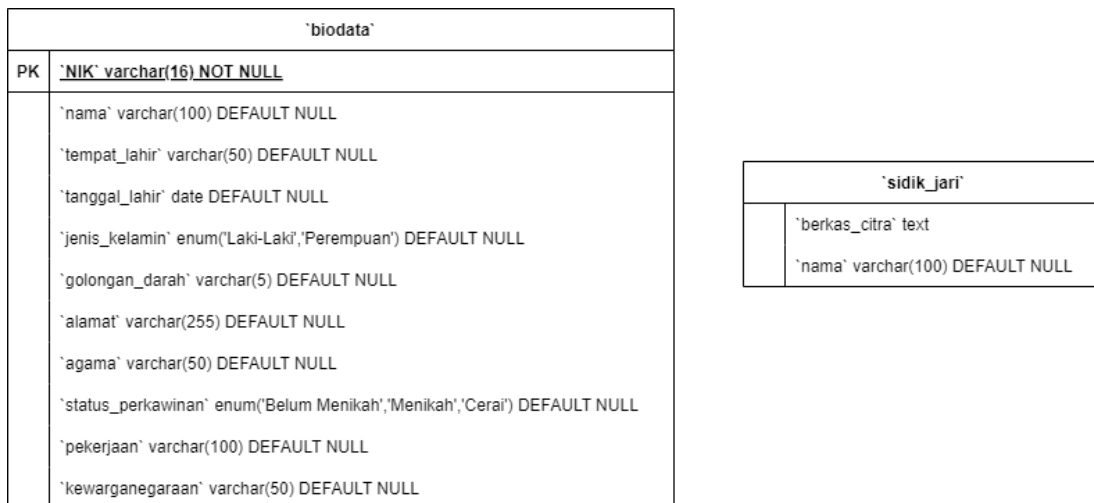
Citra Sidik Jari	Potongan nilai binary	Potongan ASCII 8-bits
	10100000101000001010000 01010000010100000101000 00101000001010000010100 00010100000101000001010 00001010000010100000101 00000101000001010000010 10000010100000101000001 01000001010000010100000 10100000101000001010000 0101000001010	iÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ ûÿkÿý,Pþÿ¿/ÿÿ' »ÿÿÐ¾?ÿ&¼j¸Úª†È/Zíÿÿq ¼ÿúTý~Rûôt ÿÿÿÿÿÿÿÿÿ iÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ bÿÿÿÁüþÿÊ' þ /ÿæ2ÿúYÿÿ\ ^ÿÿp fÿÿà !>ÿÿÿþÿÿÿÿÿÿÿÿÿ <ðÿÕ ²ÿÿQûÿNºÿášÿÿÿÿÿÿÿÿÿ iÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ <>ÿÜþþ6Gÿÿ;Úÿÿÿfäÿÿÿ

**Tabel 1.** Hasil konversi citra sidik jari menjadi binary dan konversi ke ASCII 8-bits.

Sumber: Dokumentasi Penulis

Pada tahap implementasi di titik ini, telah dihasilkan serangkaian karakter ASCII 8-bit yang merepresentasikan sebuah sidik jari. Hasil ini yang akan dijadikan dasar pencarian sidik jari yang sama dengan daftar sidik jari yang terdapat pada basis data. Pencarian sidik jari yang paling mirip dengan sidik jari yang menjadi masukan pengguna dilakukan dengan algoritma pencocokan *string* Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM). Jika tidak ada satupun sidik jari pada basis data yang *exact match* dengan sidik jari yang menjadi masukan melalui algoritma KMP ataupun BM, maka gunakan sidik jari paling mirip dengan kesamaan diatas nilai tertentu (*threshold*). Anda diberikan **kebebasan untuk menentukan nilai batas persentase** kemiripan ini, silakan melakukan pengujian untuk menentukan nilai *tuning* yang tepat dan **jelaskan pada laporan**. Metode perhitungan tingkat kemiripan **juga dibebaskan** kepada Anda asalkan dijelaskan di laporan. Akan tetapi, asisten sangat menyarankan untuk menggunakan salah satu dari algoritma **Hamming Distance, Levenshtein Distance, ataupun Longest Common Subsequence (LCS)**.

Fungsi dari deteksi biometrik adalah mengidentifikasi seseorang, oleh sebab itu, pada proses implementasi program ini, sebuah citra sidik jari akan dicocokkan dengan biodata seseorang. Biodata yang dicocokkan terdiri atas data-data yang terdapat pada KTP, antara lain : NIK, nama, tempat/tanggal lahir, jenis kelamin, golongan darah, alamat, agama, status perkawinan, pekerjaan, dan kewarganegaraan. Relasi ini dibuat dalam sebuah basis data dengan skema detail seperti yang tertera pada bagian di bawah ini. Sebagai tambahan, struktur relasi basis data telah disediakan, silakan gunakan dump sql [berikut](#). Atribut `berkas_citra` yang disimpan pada tabel `sidik_jari` adalah alamat dari citra dalam repositori (`test/...`), lokasi penyimpanan citra dalam folder **test**, bisa dilihat pada struktur repositori di bagian [Pengumpulan Tugas](#).



**Gambar 3.** Skema basis data yang digunakan.

Sumber: Dokumentasi Penulis

Seorang pribadi dapat memiliki lebih dari satu berkas citra sidik jari (**relasi one-to-many**). Akan tetapi, seperti yang dapat dilihat pada skema relasional di atas, keduanya tidak terhubung dengan sebuah relasi. Hal ini disebabkan karena pada kasus dunia nyata, data yang disimpan bisa saja mengalami korupsi. Dengan membuat atribut kolom yang mungkin korupsi adalah atribut nama pada tabel biodata, maka atribut nama pada tabel sidik\_jari **tidak dapat memiliki foreign-key** yang mereferensi ke tabel biodata (silakan *review* kembali materi IF2240 bagian basis data relasional). Pada tugas besar kali ini, kita akan coba melakukan simulasi implementasi data korupsi yang **hanya mungkin terjadi pada atribut nama di tabel biodata** (asumsikan kolom lain pada setiap tabel tidak mengalami korupsi). Akan tetapi, karena tujuan utama program adalah mengenali identitas seseorang secara lengkap hanya dengan menggunakan sidik jari, maka harus dilakukan sebuah skema untuk menangani data korupsi tersebut.

Sebuah data yang korupsi dapat memiliki berbagai macam bentuk. Pada tugas ini, jenis data korupsi adalah bahasa alay Indonesia. Terdengar lucu, tetapi para pendahulu kita sudah membuat bahasa ini untuk berkomunikasi kepada sesamanya. Dengan mengutip dari [berbagai sumber](#), Anda akan diminta untuk menangani **kombinasi dari tiga buah variasi** bahasa alay, yaitu kombinasi huruf besar-kecil, penggunaan angka, dan penyingkatan. Contoh kasus bahasa alay dijelaskan dengan detail sebagai berikut.

Variasi	Hasil
Kata orisinal	Bintang Dwi Marthen
Kombinasi huruf besar-kecil	bintanG DwI mArthen
Penggunaan angka	B1nt4n6 Dw1 M4rthen
Penyingkatan	Bntng Dw Mrthen
Kombinasi ketiganya	b1ntN6 Dw mrthn

**Tabel 2.** Kombinasi ketiga variasi bahasa alay Indonesia.

Sumber: Dokumentasi Penulis

Cara yang dapat digunakan untuk menangani ini adalah dengan menggunakan Regular Expression (Regex). Lakukan konversi pola karakter alay hingga dapat dikembalikan ke bentuk alfabetik yang bersesuaian. Setelah menggunakan Regex, Anda akan diminta kembali untuk melakukan *pattern matching* antara nama yang bersesuaian dengan algoritma KMP dan BM dengan ketentuan yang sama seperti saat [pencocokan sidik jari](#). Sebagai referensi bahasa alay, Anda dapat menggunakan *website* alay generator yang terdapat pada bagian referensi.

Setelah menemukan nama yang bersesuaian, Anda dapat menggunakan basis data untuk mengembalikan detail biodata lengkap individu. Jika seluruh prosedur diatas diimplementasikan dengan baik, maka sebuah sistem deteksi individu berbasis biometrik dengan sidik jari telah berhasil untuk diimplementasikan.

## Penggunaan Program

Pada Tugas Besar kali ini, sistem yang dibangun akan diimplementasikan dengan basis **desktop-app** menggunakan **bahasa C#**. Saran dari asisten, Anda dapat menggunakan kakas **WinForm atau WPF** untuk membangun GUI. Masukan yang akan diberikan oleh pengguna saat menggunakan aplikasi adalah **sebuah citra sidik jari**. Selain itu program perlu untuk **memiliki basis data SQL** dengan struktur relasional seperti yang telah dijelaskan sebelumnya. Tampilan layout dari aplikasi yang akan dibangun adalah sebagai berikut.



**Gambar 4.** Referensi tampilan UI *desktop-app*.

Sumber: Dokumentasi Penulis

Anda dapat menambahkan menu lainnya seperti gambar, logo, dan sebagainya. Tampilan *front-end* dari *desktop-app* **tidak harus sama persis** dengan *layout* yang diberikan

di atas, tetapi dibuat semenarik mungkin dan **wajib mencakup komponen-komponen berikut:**

- Judul aplikasi
- Tombol Insert citra sidik jari, beserta *display* citra sidik jari yang ingin dicari
- Toggle Button untuk memilih algoritma yang ingin digunakan (KMP atau BM)
- Tombol Search untuk memulai pencarian
- *Display* sidik jari yang paling mirip dari basis data
- Informasi mengenai waktu eksekusi
- Informasi mengenai tingkat kemiripan sidik jari dengan gambar yang ingin dicari, dalam persentase (%)
- *List* biodata hasil pencarian dari basis data. Keluarkan semua nilai atribut dari individu yang dirasa paling mirip. Perlu diperhatikan pendefinisian batas kemiripan dapat memunculkan kemungkinan tidak ditemukan list biodata yang memiliki sidik jari paling mirip.

Secara umum, berikut adalah cara umum penggunaan program:

1. Pengguna terlebih dahulu memasukkan citra sidik jari yang ingin dicari biodatanya.
2. Setelah citra dimasukkan, pilih opsi pencarian, ingin melakukan pencarian apakah akan menggunakan algoritma KMP atau BM.
3. Tekan tombol search, program kemudian akan memproses, mencari citra sidik jari dari basis data yang memiliki kemiripan dengan citra sidik jari yang menjadi masukan.
4. Program akan menampilkan list biodata jika ditemukan citra sidik jari yang memiliki kemiripan dengan batas persentase tertentu. Program juga dapat mengeluarkan informasi tidak ada sidik jari yang mirip jika semua citra dalam basis data tidak memiliki kemiripan dengan masukan.
5. Terdapat informasi terkait waktu eksekusi program dan persentase kemiripan citra.

## Spesifikasi Tugas Besar

Pada Tugas Besar ini, buatlah sebuah sistem yang dapat melakukan identifikasi individu berbasis biometrik dengan menggunakan sidik jari dengan detail sebagai berikut.

1. Sistem dibangun dalam **bahasa C#** dengan kakas Visual Studio .NET yang mengimplementasikan **algoritma KMP, BM, dan Regular Expression** dalam mencocokkan sidik jari dengan biodata yang berpotensi rusak. Pelajarilah C# desktop development, **disarankan untuk menggunakan Visual Studio** untuk mempermudah pengerjaan.
2. Program dapat memiliki basis data **SQL** yang telah mencocokkan berkas citra sidik jari yang telah ada dengan seorang pribadi. Basis data yang digunakan dibebaskan asalkan **bukan No-SQL** (sebagai contoh, MySQL, PostgreSQL, SQLite).
3. Program dapat menerima masukan sebuah citra sidik jari yang ingin dicocokkan. Apabila citra tersebut memiliki kecocokan di atas batas tertentu (silakan lakukan *tuning* nilai yang tepat) dengan citra yang sudah ada, maka tunjukkan biodata orang

tersebut. Apabila di bawah nilai yang telah ditentukan tersebut, memunculkan pesan bahwa sidik jari tidak dikenali.

4. Program memiliki keluaran yang **minimal** mengandung seluruh data yang terdapat pada contoh antarmuka pada bagian [penggunaan program](#).
5. Pengguna dapat memilih algoritma yang ingin digunakan antara KMP atau BM.
6. Biodata yang ditampilkan harus biodata yang memiliki nama yang benar (gunakan Regex untuk memperbaiki nama yang rusak dan gunakan KMP atau BM untuk mencari orang yang paling sesuai).
7. Program memiliki antarmuka yang *user-friendly*. Anda juga dapat menambahkan fitur lain untuk menunjang program yang Anda buat (unsur kreativitas).

## Bonus

Bagian ini hanya boleh dikerjakan apabila spesifikasi wajib dari Tugas Besar telah berhasil dipenuhi. Anda tidak diharuskan untuk mengerjakan keseluruhan bonus, tetapi semakin banyak bonus yang dikerjakan, maka akan semakin banyak tambahan nilai yang diperoleh.

### 1. Lakukan enkripsi terhadap semua data pada basis data

Data yang terdapat dalam KTP merupakan data yang privat dan tidak seharusnya mudah diakses orang yang tidak berwenang. Buatlah sebuah skema enkripsi-dekripsi semua data pada basis data sehingga ketika ada pihak yang melakukan *query SQL* secara langsung ke basis data, tidak ada data berarti yang didapatkan (dilarang menggunakan *library*, implementasikan sendiri, semakin aman semakin tinggi nilai bonus ini).

### 2. Membuat video

Buatlah sebuah video mengenai program yang dibuat dan algoritma yang digunakan pada tugas besar ini. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Video tersebut kemudian diupload ke YouTube. Beberapa contoh video tubes tahun-tahun sebelumnya dapat dilihat di YouTube dengan menggunakan kata kunci “Tubes Stima”, “Tugas Besar Stima”, “Strategi Algoritma”, dan lain-lain.

## Prosedur Pengerjaan

1. Tugas dikerjakan berkelompok dengan anggota minimal 2 orang dan maksimal 3 orang, boleh lintas kelas dan lintas kampus. Akan tetapi, anggota kelompok **tidak boleh sama** dengan anggota kelompok pada tugas-tugas Strategi Algoritma sebelumnya.
2. Setiap kelompok harap mengisi nama kelompok dan anggotanya pada pranala [bit.ly/kelompoktubes3stima24](https://bit.ly/kelompoktubes3stima24), paling lambat **Kamis, 9 Mei 2024 pukul 22.11 WIB**.
3. Setiap kelompok diwajibkan untuk memilih asisten meskipun tidak melakukan asistensi, karena asisten yang dipilih akan menjadi asisten saat asistensi (opsional) dan demo tugas besar. Pemilihan asisten dapat dilakukan pada link berikut, paling lambat **Kamis, 9 Mei 2024 pukul 22.11 WIB**.

4. Program harus mengandung komentar yang jelas serta mudah dibaca.
5. Mahasiswa dilarang menggunakan kode program yang didapatkan dari internet (alasan menggunakan kakas seperti GitHub Copilot tidak diterima). Mahasiswa harus membuat program sendiri, diperbolehkan untuk belajar dari program yang sudah ada maupun menggunakan kakas AI (tetapi mahasiswa harus tetap memahami apa yang dikerjakan, bila menggunakan kakas AI maka lampirkan tangkapan layar penggunaannya di laporan).
6. Jika terdapat kesulitan selama mengerjakan tugas besar sehingga memerlukan bimbingan, maka dapat melakukan asistensi tugas besar kepada asisten (opsional). Dengan catatan asistensi hanya bersifat membimbing, bukan memberikan “jawaban”.
7. Terdapat juga demo dari program yang telah dibuat. Pengumuman tentang demo menunggu pemberitahuan lebih lanjut dari asisten.

### 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 teknik pengukuran persentase kemiripan.
  - Penjelasan singkat mengenai aplikasi desktop yang dibangun.
- **Bab 3** : Analisis Pemecahan Masalah
  - Langkah-langkah pemecahan masalah.
  - Proses penyelesaian solusi dengan algoritma KMP dan BM.
  - Fitur fungsional dan arsitektur aplikasi desktop yang dibangun.
  - Contoh ilustrasi kasus.
- **Bab 4** : Implementasi dan Pengujian
  - Spesifikasi teknis program (struktur data, fungsi, dan 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 Anda).
  - Analisis hasil pengujian.
- **Bab 5** : Kesimpulan, Saran, Tanggapan, dan Refleksi tentang Tugas Besar 3.
- **Lampiran** : Tautan *repository* GitHub dan video (apabila membuat). *Dilarang melampirkan tautan hasil URL shortener, misal: bit.ly atau semacamnya.*
- **Daftar Pustaka**



## Pengumpulan Tugas

1. Program disimpan dalam repository yang bernama **Tubes3\_NamaKelompok** dengan nama kelompok sesuai dengan yang di *sheets* diatas. Berikut merupakan struktur dari isi repository tersebut:
  - a. Folder src berisi **program yang dapat dijalankan**
  - b. Folder doc berisi laporan tugas besar dengan format **NamaKelompok.pdf**
  - c. Folder test berisi dataset yang digunakan untuk pengujian (folder ini juga yang akan menjadi lokasi citra test case pada saat demo)
  - d. README untuk tata cara penggunaan yang minimal berisi:
    - i. Penjelasan singkat algoritma KMP, BM, dan ekspresi reguler yang diimplementasikan
    - ii. Requirement program dan instalasi tertentu bila ada
    - iii. Command atau langkah-langkah dalam meng-compile atau build program
    - iv. Author (identitas pembuat)
2. Laporan dikumpulkan hari **Sabtu, 25 Mei 2024** pada alamat Google Form berikut paling lambat pukul **23.59** : <https://bit.ly/tubes3stima24>
3. Adapun pertanyaan terkait tugas besar ini bisa disampaikan melalui QnA berikut: <https://bit.ly/qnastima24>

### Keterangan laporan:

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Laporan mengikuti format pada *section* "Isi laporan" dengan baik dan benar.
3. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

## Penilaian

### 1. Bagian 1: Laporan (40%)

- a. Langkah-langkah pemecahan masalah, proses pemetaan masalah, fitur fungsional dan arsitektur aplikasi aplikasi desktop yang dibangun, dan contoh ilustrasi kasus. (20%)
- b. Spesifikasi teknis program (struktur data, fungsi, dan 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), dan analisis hasil pengujian. (15%)
- c. Kelengkapan komponen-komponen pada laporan dan README. (5%)

### 2. Bagian 2: Implementasi Program dan Demo (60%)

- a. Kecocokan output dari test case yang diberikan. (25%)
- b. Kebenaran algoritma KMP dan BM, sesuai dengan apa yang telah diajarkan di kelas. (15%)

- c. Pemahaman tugas besar dan algoritma yang telah dibuat oleh masing-masing anggota. (10%)
- d. Keberhasilan input dan output, sesuai dengan komponen-komponen yang ada pada spesifikasi. (5%)
- e. Modularitas/keterbacaan penulisan program. (5%)

### 3. Bagian 3: Komponen Lainnya (Bonus)

- a. Lakukan enkripsi terhadap semua data pada basis data.
- b. Bonus dalam membuat video kelompok.
- c. Kreativitas dalam membuat *interface desktop-app*.

Perhatikan bahwa

- **Dilarang keras** *copy paste* program dari internet (AI, repository lain, ataupun teman). Program harus dibuat sendiri, kecurangan akan mengakibatkan nilai tugas menjadi nol.
- Pastikan program **dapat setidaknya dijalankan** pada *windows* dan *linux*.
- Apabila program **tidak bisa dijalankan** maka tidak akan dinilai oleh asisten.

### Referensi

Perhatikan bahwa referensi hanya ada sebagai sumber belajar, bukan sebagai standar apa yang kami ekspektasikan dari kode kalian. Banyak sumber lain, jangan menutup diri ke hanya sumber sumber di bawah ini.

“Fingerprint Identification using Bozorth and Boyer-Moore Algorithm”

<https://iopscience.iop.org/article/10.1088/1757-899X/662/2/022040>

Proses identifikasi individu melalui citra sidik jari.

“Sokoto Coventry Fingerprint Dataset (SOCOFing)”

<https://www.kaggle.com/datasets/ruizgara/socofing>

Dataset *fingerprint* yang bisa digunakan.

“Bahasa Alay”

[https://id.wikipedia.org/wiki/Bahasa\\_alay](https://id.wikipedia.org/wiki/Bahasa_alay)

Wikipedia stuff, apa itu bahasa alay?

“Bahasa Alay Generator”

<https://alaygenerator.blogspot.com/>

Melakukan simulasi konversi bahasa alay Indonesia.

“How do I use SQL Server with C# and .NET?”

<https://learn.microsoft.com/en-us/shows/on-net/how-do-i-use-sql-server-with-csharp-and-dotnet>

Cara membuat koneksi SQL dengan C#.

“Tutorial: Create a Windows Forms app in Visual Studio with C#”

<https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022>

Cara membangun windows form app dengan C#.

“Symmetric-key algorithm”

[https://simple.wikipedia.org/wiki/Symmetric-key\\_algorithm](https://simple.wikipedia.org/wiki/Symmetric-key_algorithm)

Penjelasan mengenai kunci simetri untuk yang tertarik mengimplementasikan bonus 🙄.

“Public-key cryptography”

[https://simple.wikipedia.org/wiki/Public-key\\_cryptography](https://simple.wikipedia.org/wiki/Public-key_cryptography)

Penjelasan terkait kriptografi kunci publik.

- - - Selamat Mengerjakan - - -

“Kenapa speknya panjang? supaya jelas 😊”

- Leon -

“jUjur 64 T4u maU 151 4p4”

- Marthen -

“Bang, udah bang 😊”

- Haziq -

“Kenapa stima selalu banyak tugas? karena asisten stima adalah yin, yang tidak akan pernah berubah, apapun yang terjadi.”

- Rifqi -

“ril last game, ga lagi kok”

- Alex -

“S3m4n94T”

- Sulthan -

“Tidak terasa sudah di akhir, semangat 🙌”

- Nigel -

“Tentu saja! Berikut adalah beberapa kalimat motivasi untuk membantu kamu menyelesaikan tugas besar terakhir di semester ini:

- Kamu sudah bekerja keras sepanjang semester, jadi jadikan tugas terakhir ini sebagai kesempatan untuk menunjukkan semua yang telah kamu pelajari.
- Saat kamu merasa lelah, ingatlah mengapa kamu memulai perjalanan ini. Teruskan, dan kamu akan sampai di tujuanmu.

- Ingat, ini hanya satu langkah lagi. Kamu sudah melewati banyak tantangan, dan kamu bisa melewati yang satu ini juga.

Semoga ini memberikan semangat untukmu dalam menyelesaikan tugas besar terakhirmu!  
Semangat, kamu pasti bisa!”

- Bewe -