

**LAPORAN TUGAS BESAR 03**  
**IF2211 STRATEGI ALGORITMA**

Kelompok C b is Justice



Disusun oleh:

Muhammad Yusuf Rafi (13522009)

Devinzen (13522064)

Emery Fathan Zwagery (13522079)

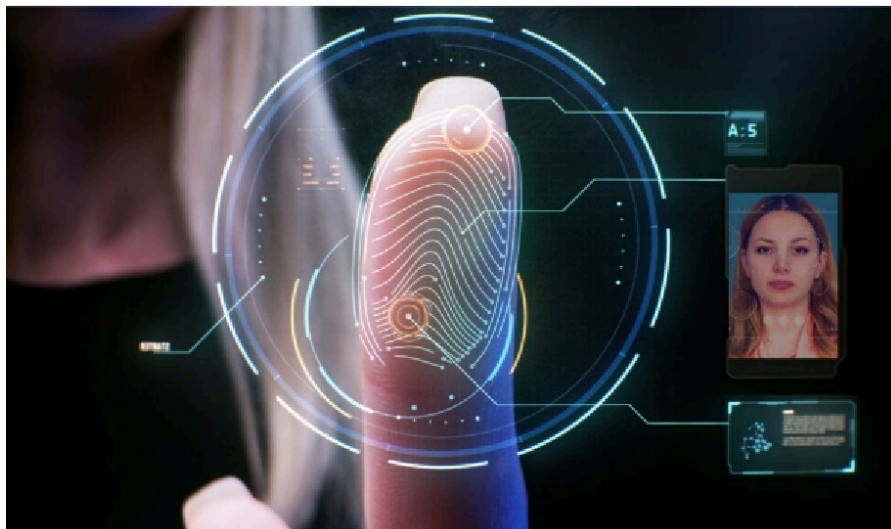
**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**  
**INSTITUT TEKNOLOGI BANDUNG**  
**BANDUNG**

**2023**

## **BAB I**

### **DESKRIPSI MASALAH**

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.



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

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.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Algoritma KMP, BM, dan Regular Expression**

Algoritma Knuth-Morris-Pratt dan Boyer-Moore digunakan untuk mencocokkan string. Algoritma KMP menggunakan tabel prefiks atau tabel border function yang menghitung prefiks terbesar di substring  $[0..i]$  yang juga suffix dari substring itu. Kompleksitas waktu algoritma ini linear.

Algoritma Boyer-Moore ditemukan oleh Robert Stephen Boyer dan J Strother Moore. Algoritma ini mencari polanya dari belakang, melakukan komparasi dari kanan ke kiri. Sebelum mulai di match, polanya di preprocess untuk membuat bad character table (last occurrence function). Ini menentukan bagaimana pola digeser ketika terjadi mismatch.

Regular Expression (regex) adalah sekuens karakter yang mendeskripsikan pola untuk dicari didalam teks. Ada beberapa karakter spesial di regex seperti '.', '+', '\*', '?', dll. Tanda kurung [] digunakan untuk mencocokkan dengan salah satu karakter di dalamnya. '\*' untuk mencocokkan 0 atau lebih ekspresi sebelumnya, '+' untuk mencocokkan 1 atau lebih ekspresi sebelumnya, '?' untuk mencocokkan 0 atau 1 kali ekspresi sebelumnya, dan lain-lain.

#### **2.2 Pengukuran persentase kemiripan**

Untuk mencari tingkat kemiripan string digunakan Longest Common Subsequence (LCS), yaitu mencari panjang subsequence terbesar yang ada di kedua string. Proses mencari LCS pada program yang dibuat menggunakan dynamic programming.

#### **2.3 Aplikasi Desktop yang akan dibangun**

Aplikasi desktop yang akan dibangun akan menerima berkas citra sidik jari dan akan menampilkan sidik jari paling mirip yang ada di database, serta biodata pemilik sidik jari tersebut.

## **BAB III**

### **ANALISIS PEMECAHAN MASALAH**

#### **3.1 Langkah-Langkah Pemecahan Masalah**

Pertama-tama dari citra sidik jari diubah menjadi grayscale. Setelah itu data-data pixel diubah menjadi bentuk binary. Pixel yang grayscale valuenya lebih dari 127 menjadi 1 dan yang kurang dari 127 menjadi 0. Karena proses string matching pada binary lambat, maka data binary dikelompokkan menjadi ASCII 8 bit.

Bentuk ASCII 8 bit inilah yang akan dibandingkan dengan sidik jari pada database. Jika tidak ditemukan sidik jari yang benar-benar sama, maka akan dicari sidik jari yang paling mirip dengan kemiripan di atas 70%.

#### **3.2 Penyelesaian solusi dengan algoritma KMP dan BM**

Untuk algoritma KMP, dicari dulu border function pada pattern. Fungsi pinggiran  $b(k)$  itu panjang prefix terbesar di  $P[0..k]$  yang juga suffix di  $P[1..k]$ . Setelah ditentukan nilai border function untuk setiap  $k$  bisa mulai mencari pola. Pencarian mulai dilakukan dari kiri. Jika ketemu yang tidak sama (mismatch) maka akan menggeser sesuai dengan nilai dari border functionnya.

Untuk algoritma BM, dicari dulu fungsi last occurrence. Pada implementasi dibuat array `charTable` dengan ukuran 256 untuk setiap karakter. Dibuat juga array `offsetTable` untuk menghitung berapa banyak pergeserannya ketika terjadi mismatch. Ketika terjadi mismatch saat pencarian, maka akan menggeser sesuai dengan nilai pada `charTable` dan `offsetTable`.

#### **3.3 Fitur Fungsional dan Arsitektur Aplikasi**

Pada aplikasi yang dibuat pengguna bisa memasukkan gambar citra sidik jari yang akan dibandingkan. Pengguna juga bisa memilih algoritma yang ingin digunakan dalam pencarian (KMP atau BM). Setelah pencarian selesai, program akan menampilkan sidik jari yang paling mirip beserta biodata pemilik sidik jari itu, waktu eksekusi dan tingkat kemiripan.

Program ditulis dengan bahasa C# dengan menggunakan WPF untuk GUInya. Program juga membuat koneksi ke database sql yang digunakan untuk menyimpan biodata orang.

Program menggunakan algoritma KMP dan BM untuk mencocokkan string yang didapat dari citra sidik jari. Jika tidak ada yang exact match, maka akan dicari yang terdekat menggunakan algoritma LCS. Untuk menyimpan dan mengembalikan biodata pada database menggunakan skema enkripsi-dekripsi AES. Regular expression digunakan untuk mencocokkan nama yang bisa saja menggunakan bahasa alay.

### **3.4 Contoh Ilustrasi Kasus**

Citra sidik jari bisa citra asli maupun yang rusak. Untuk kategori rusak ada yang rusak kecil, sedang, dan parah. Kami menggunakan teknik pengambilan sampel yang baik untuk menangani gambar yang rusak parah. Kami menggunakan LRE dan transisi frekuensi( perubahan antara character-i dan character-i+1. Jika tidak ada yang match 100% digunakan LCS.

## **BAB IV IMPLEMENTASI DAN PENGUJIAN**

### **4.1 Spesifikasi Teknis Program**

#### **4.1.1 Backend**

##### **4.1.1.1 Import**

<b>Nama Import</b>	<b>Deskripsi</b>
--------------------	------------------

Org.BouncyCastle.Utilities.Collections	Menyediakan berbagai koleksi utilitas yang digunakan dalam konteks kriptografi, seperti tabel hash dan array dinamis.
System	Menyediakan kelas fundamental dan base class seperti tipe data dasar (string, int, dll.), pengecualian (exceptions), dan lainnya yang diperlukan untuk aplikasi dasar.
System.Collections.Generic	Menyediakan kelas koleksi generik seperti <code>List&lt;T&gt;</code> , <code>Dictionary&lt;TKey, TValue&gt;</code> , <code>Queue&lt;T&gt;</code> , dan <code>Stack&lt;T&gt;</code> yang lebih efisien dan aman dari tipe dibandingkan koleksi non-generik.
System.Diagnostics	Menyediakan alat untuk logging, pelacakan (tracing), dan pengukuran performa aplikasi. Contohnya <code>Debug</code> , <code>Trace</code> , dan <code>Process</code> .
System.Drawing	Menyediakan kelas untuk bekerja dengan gambar, warna, font, dan grafis lainnya dalam aplikasi berbasis Windows Forms.
System.Linq	Menyediakan kemampuan query yang kuat terhadap koleksi, database, XML, dan data lain menggunakan sintaks yang konsisten dan terintegrasi dalam bahasa C#.
System.Text	Menyediakan kelas untuk bekerja dengan teks, encoding, dan manipulasi string yang

	lebih canggih seperti <b>StringBuilder</b> , <b>Encoding</b> , dan lainnya.
System.Threading.Tasks	Menyediakan kelas dan tipe untuk membuat dan mengelola tugas asinkron, seperti <b>Task</b> , <b>Task&lt;T&gt;</b> , dan <b>Parallel</b> .
System.Windows.Media.Animation	Menyediakan kelas dan metode untuk membuat animasi pada elemen UI dalam aplikasi WPF, seperti <b>Storyboard</b> , <b>DoubleAnimation</b> , dan <b>Timeline</b> .
WpfApp1.Utilities	Menyediakan kelas utilitas yang didefinisikan pengguna untuk mendukung berbagai fungsionalitas dalam aplikasi, seperti konversi gambar, pemrosesan data, atau logika aplikasi khusus lainnya.

#### 4.1.1.2 Struktur Data

Struktur Data	Deskripsi
<pre> public class Foto {     public string     AsciiRepresentation { get;     set; }     public string Path {     get; set; }     public string Name {     get; set; } } </pre>	<p>Kelas <b>Foto</b> berfungsi untuk menyimpan informasi terkait sebuah foto, dengan tiga properti utama: <b>AsciiRepresentation</b>, <b>Path</b>, dan <b>Name</b>.</p> <p>Properti <b>AsciiRepresentation</b> menyimpan representasi ASCII dari foto. Properti <b>Path</b> menyimpan jalur file dari foto menunjukkan lokasi file foto di sistem file, yakni folder 'fingerprint'.</p> <p>Properti <b>Name</b> menyimpan nama foto yang didapatkan dari database.</p>



<pre> public class Biodata {     public string NIK { get; set; } // varchar(16), primary key, not null     public string Nama { get; set; } // varchar(100), nullable     public string TempatLahir { get; set; } // varchar(50), nullable     public string TanggalLahir { get; set; } // date, nullable, disimpan sebagai string     public string JenisKelamin { get; set; } // enum('Laki-Laki','Perempuan'), nullable     public string GolonganDarah { get; set; } // varchar(5), nullable     public string Alamat { get; set; } // varchar(255), nullable     public string Agama { get; set; } // varchar(50), nullable     public string StatusPerkawinan { get; set; } // enum('Belum Menikah','Menikah','Cerai'), nullable     public string Pekerjaan { get; set; } // varchar(100), nullable     public string Kewarganegaraan { get; set; } // varchar(50), nullable } </pre>	<p>Kelas `Biodata` digunakan untuk menyimpan informasi pribadi seseorang dalam aplikasi. Properti `NIK` adalah kunci utama yang unik dan tidak boleh null, menyimpan Nomor Induk Kependudukan. `Nama`, `TempatLahir`, `TanggalLahir`, `JenisKelamin`, `GolonganDarah`, `Alamat`, `Agama`, `StatusPerkawinan`, `Pekerjaan`, dan `Kewarganegaraan` adalah properti lainnya yang menyimpan detail biodata seperti nama, tempat dan tanggal lahir, jenis kelamin, golongan darah, alamat, agama, status pernikahan, pekerjaan, dan kewarganegaraan. Properti ini memungkinkan pengorganisasian dan pengelolaan data pribadi secara terstruktur dalam aplikasi.</p>
--	--

#### 4.1.1.3 Fungsi

Nama Fungsi (parameter)	Deskripsi
public int KMPSearch(string text, string pattern)	Fungsi ini untuk mencari indeks pertama dimana

	string cocok dengan pola dengan algoritma KMP.
private static int[] BorderKMP(string pattern)	Fungsi ini untuk menghitung border function KMP dari pola dan memasukkan ke sebuah array.
public int BoyerMooreSearch(string text, string pattern)	Fungsi ini untuk mencari indeks pertama dimana string cocok dengan pola dengan algoritma BM.
private int[] MakeCharTable(string pattern)	Fungsi ini digunakan untuk menghitung posisi terakhir setiap huruf dalam pola (last occurrence) untuk algoritma BM.
private int[] MakeOffsetTable(string pattern)	Fungsi ini digunakan untuk membuat tabel good suffix untuk algoritma BM.
private bool IsPrefix(string pattern, int p)	Mengecek apakah substring yang dimulai pada index p pada pattern merupakan prefiks dari pattern.
private int SuffixLength(string pattern, int p)	Menghitung panjang substring dari index p ke depan yang juga merupakan suffix dari pattern.
public int _lcs(string x, string y, int m, int n)	Fungsi ini menghitung LCS (Largest Common Subsequence) dari kedua string.
public string AlayTransform(string namaAlay)	Mengubah bahasa alay menjadi normal
public string NormalToAlay(string namaNormal)	Mengubah bahasa normal menjadi alay untuk generate database
private string ShortenWord(string word)	Memendekkan kata dengan menghilangkan semua huruf vokal
public static string aes128_encrypt(string s, byte[] key)	Fungsi untuk mengenkripsikan data string dengan AES.

<pre>public static string aes128_decrypt(string s, byte[] key)</pre>	Fungsi untuk mendekripsikan string base64 ke string dengan AES.
--	---

## 4.1.2 Frontend

### 4.1.2.1 MainWindow.xaml

Kode XAML tersebut mendefinisikan tata letak dan tampilan untuk jendela dalam aplikasi WPF. Jendela ini memiliki dua bagian utama: bagian untuk menampilkan gambar sidik jari dan bagian untuk menampilkan informasi terkait pencocokan sidik jari, serta bagian untuk menampilkan informasi biodata.

Di dalam jendela, terdapat tiga grid yang menampung elemen-elemen UI:

Pertama, grid untuk placeholder gambar sidik jari, yang berisi gambar placeholder dan teks untuk memasukkan sidik jari.

Kedua, grid untuk gambar sidik jari yang cocok, yang berisi kontrol gambar dan teks untuk informasi sidik jari yang cocok.

Ketiga, grid untuk menampilkan informasi biodata, yang berisi beberapa TextBlock yang menampilkan detail biodata seperti NIK, nama, tempat lahir, tanggal lahir, dan sebagainya.

Selain itu, terdapat juga tombol untuk melakukan pencarian dan memilih gambar, serta tombol untuk mengaktifkan atau menonaktifkan algoritma tertentu. Data pada UI diikat (bind) ke properti-properti pada `MainWindowViewModel` untuk mengatur tampilan dan interaksi dengan data.

### 4.1.2.2 App.xaml

File `App.xaml` adalah bagian penting dari proyek aplikasi WPF yang menentukan konfigurasi global dan sumber daya yang akan digunakan di seluruh aplikasi. Di dalamnya, terdapat definisi aplikasi serta sumber daya yang diperlukan:

Atribut `Application` menetapkan kelas aplikasi WPF dan file XAML yang akan dimulai saat aplikasi dijalankan. Dalam contoh ini, `MainWindow.xaml` akan menjadi titik awal ketika aplikasi berjalan.

Dalam blok `<Application.Resources>`, sumber daya aplikasi didefinisikan. Dua file sumber daya yang digabungkan (`ButtonTheme.xaml` dan `StackPanelTheme.xaml`) akan memberikan gaya dan tata letak yang konsisten untuk tombol dan panel tumpukan di seluruh aplikasi.

Penggunaan file `App.xaml` memungkinkan manajemen sumber daya secara terpusat, memudahkan dalam mengatur gaya dan tata letak secara konsisten di seluruh aplikasi.

#### **4.1.2.3 MainWindow.xaml.cs**

Kelas `MainWindow` didefinisikan dalam namespace `WpfApp1.View` dan merupakan bagian dari aplikasi WPF. Kelas ini mengimpor beberapa pustaka dari .NET dan WPF yang diperlukan untuk berbagai fungsi seperti manipulasi file, operasi IO, tugas asinkron, dan elemen UI. Pustaka yang diimpor mencakup `Microsoft.Win32`, `System`, `System.Collections.Generic`, `System.IO`, dan beberapa lainnya yang menyediakan fungsionalitas seperti penanganan file, operasi string, serta kontrol dan elemen UI.

Terdapat konstruktor `public MainWindow()`, yang berfungsi untuk menginisialisasi jendela utama. Di dalam konstruktor ini, metode `InitializeComponent()` dipanggil untuk menginisialisasi komponen-komponen yang didefinisikan dalam file XAML yang terkait. Baris komentar menunjukkan bahwa pada awalnya ada inisialisasi `MainWindowViewModel`, tetapi saat ini dikomentari, mungkin untuk tujuan pengujian atau pengembangan lebih lanjut. Kelas `MainWindow` ini ditujukan untuk mengelola logika interaksi dari antarmuka pengguna yang didefinisikan dalam `MainWindow.xaml`.

## **4.2 Tata Cara Penggunaan Program**

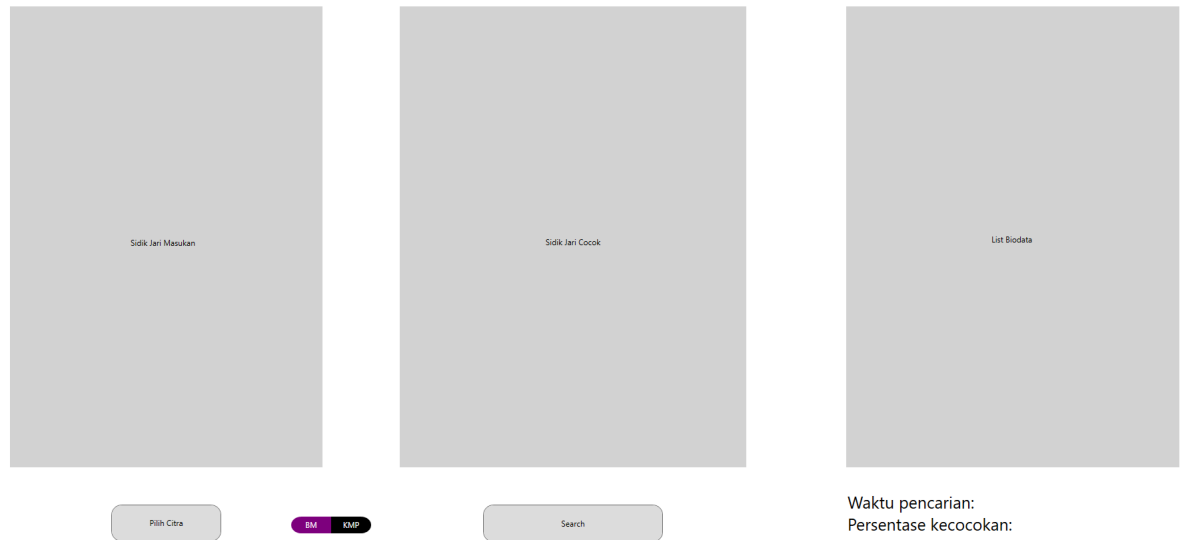
### **4.2.1 Cara menjalankan program**

Gunakan visual studio, pastikan dependency terinstall, klik run pada tombol run di visual studio

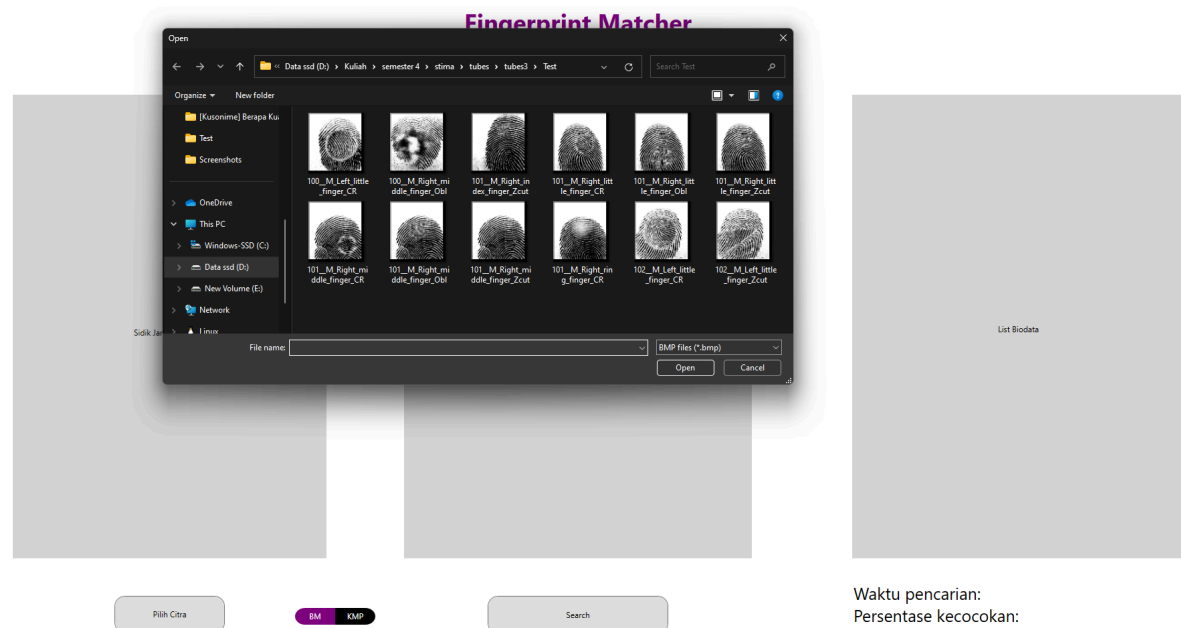
## 4.2.2 Fitur Utama

1. Pilih algoritma yang diinginkan, KMP/BM.

### Fingerprint Matcher



2. Pilih citra sidik jari yang ingin dicocokkan.



3. Tunggu proses pencocokan
4. Hasil akan ditampilkan dihalaman yang sama

## Fingerprint Matcher



Pilih Citra

BM KMP

Search

NIK: 5618459654496051

Nama: Rosa Mulyadi

Tempat Lahir: Ambon

Tanggal Lahir: 1994-01-15

Jenis Kelamin: Perempuan

Golongan Darah: AB

Alamat: Jl. Panglima Sudirman, Semarang

Agama: Konghucu

Status Perkawinan: Belum Menikah

Pekerjaan: Peneliti

Kewarganegaraan: WNI

Waktu pencarian: 2484 ms  
Persentase kecocokan: 100%

## 4.3 Hasil Pengujian

(togel ungu mengarah pada algoritma dipilih)

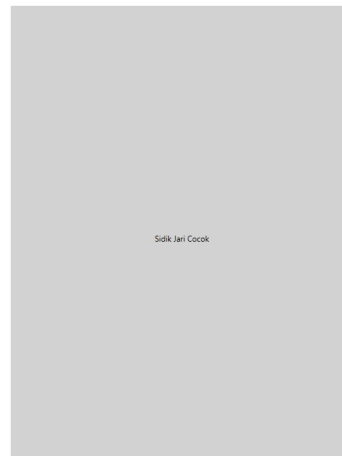
### 4.3.1 KMP

1. Tampilan awal

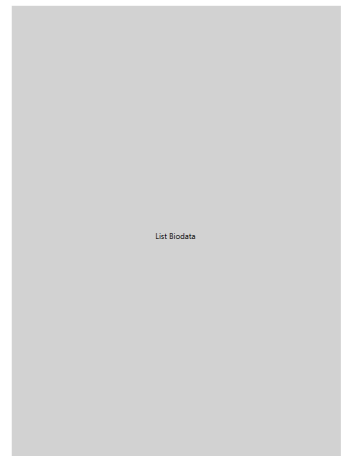
## Fingerprint Matcher



Sidik Jari Masukkan



Sidik Jari Cocok



List Biodata

Pilih Citra



BM KMP

Search

Waktu pencarian:  
Persentase kecocokan:

## 2. Image Input yang Real

### Fingerprint Matcher



Pilih Citra

BM LAMP



Search

NIK: 0390849568117203  
Nama: Rosa Puspitasari  
Tempat Lahir: Tangerang  
Tanggal Lahir: 1988-12-14  
Jenis Kelamin: Perempuan  
Golongan Darah: A  
Alamat: Jl. Asia Afrika, Yogyakarta  
Agama: Hindu  
Status Perkawinan: Belum Menikah  
Pekerjaan: Pengusaha  
Kewarganegaraan: WNI

Waktu pencarian: 3166 ms  
Persentase kecocokan: 100%

## 3. Image Input yang Rusak medium

### Fingerprint Matcher



Pilih Citra

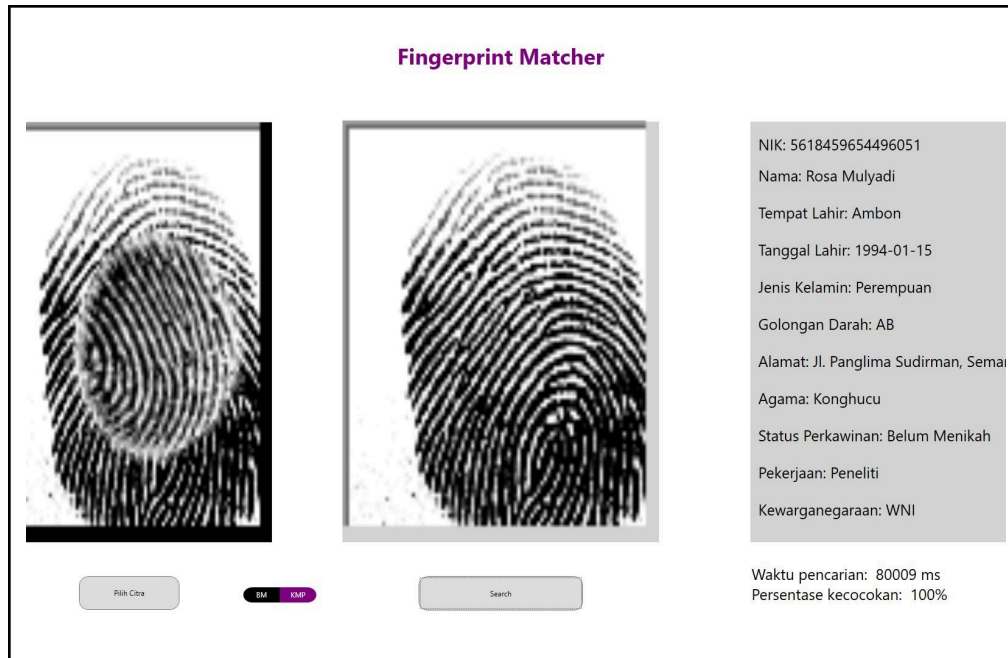
BM LAMP

Search

NIK: 3224419557442282  
Nama: Surya Anggraini  
Tempat Lahir: Bandung  
Tanggal Lahir: 1967-11-16  
Jenis Kelamin: Laki-Laki  
Golongan Darah: A  
Alamat: Jl. Sukajadi, Banda Aceh  
Agama: Konghucu  
Status Perkawinan: Belum Menikah  
Pekerjaan: Mahasiswa  
Kewarganegaraan: WNI

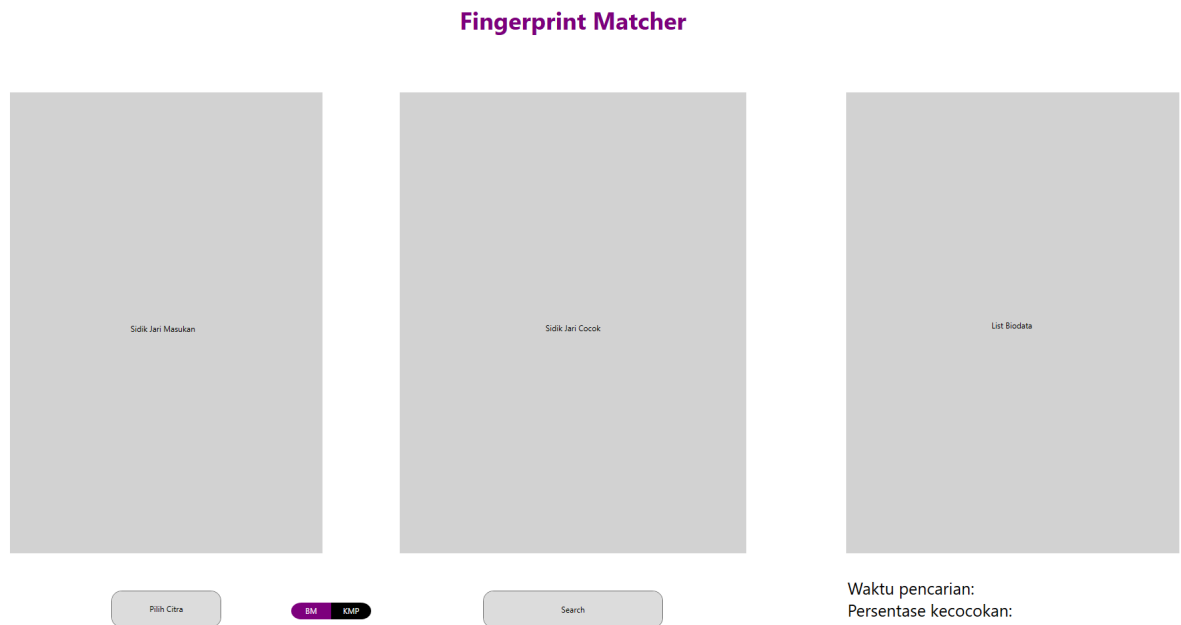
Waktu pencarian: 83849 ms  
Persentase kecocokan: 100%

## 4. Image Input yang Rusak parah



#### 4.3.2 BM

##### 1. Tampilan Awal

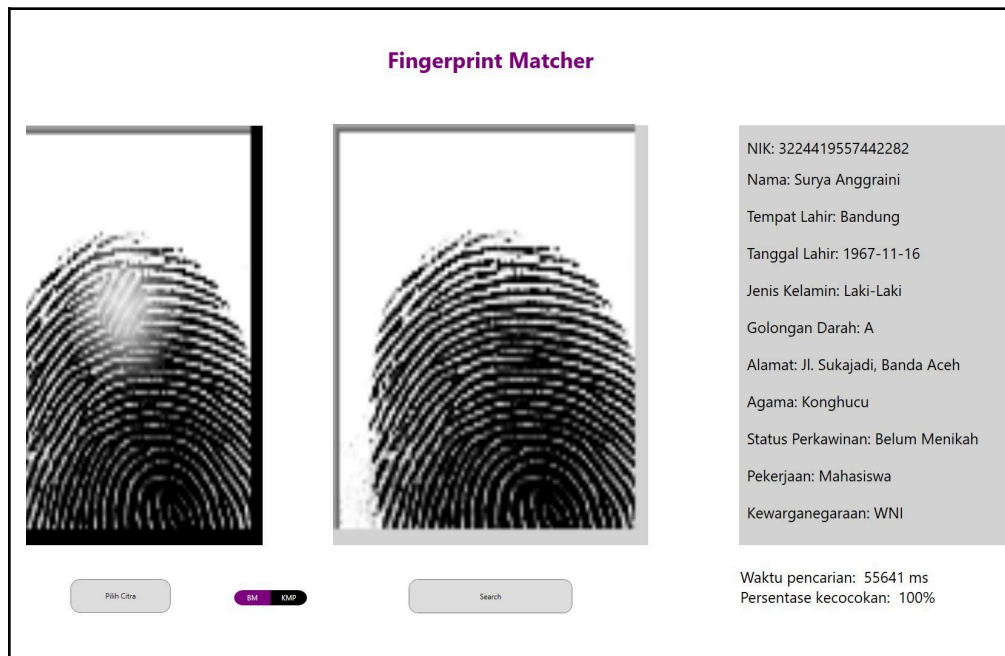


##### 2. StartURL valid dan TargetURL tidak valid





### 3. StartURL tidak valid dan TargetURL valid



## 4.4 Analisis Hasil Pengujian

Dari Pengujian yang telah kami coba , dengan menggunakan KM dan BMP kami dapat mencocokkan beberapa sidik jari yang rusak. Ini karena kita menggunakan teknik

pemilihan bagian citra sebanyak sekitar 128 bit berurutan yang kemudian dievaluasi nilai frequency transisi nya dan RLE nya. Sehingga semua hasil uji di atas match 100% karena menggunakan teknik pengambilan sampel, bukan pencocokan keseluruhan. Untuk bound LCS jaga jaga kalau hasil kemiripan tidak ada yang 100%. Namun hasil kemiripan di atas 100% semua karena kami menggunakan teknik pengambilan sampel. Kami juga dapat mencari biodata dengan pencocokan KMP dan BM dengan bantuan regex.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dalam tugas besar 3 ini kami menyimpulkan bahwa pencocokan string dapat dilakukan dengan boyer moore dan knuth morris pratt dengan teknik pengambilan sampel pada citra masukan dan full citra pada database. Ini dilakukan dengan cara mengubah image menjadi binary, lalu dari binary diubah menjadi ASCII karakter, saat menjadi ASCII karakter barulah dicocokkan dengan ASCII di database.

#### **5.2 Saran**

Tugas nya seru, untuk spesifikasi tolong diperjelas

#### **5.3 Refleksi**

Seharusnya Kami jangan deadliner.

#### **5.4 Ruang Perbaikan atau Pengembangan**

Program yang kami buat bisa lebih cepat dan efisien dengan multithreading.

#### **5.5 Link Repository**

[https://github.com/mrsuiii/Tubes3\\_C-is-Justice](https://github.com/mrsuiii/Tubes3_C-is-Justice)

## **5.6 Link Video**

[https://youtu.be/Z\\_KbQQHbUL0?si=UDZTVn0I6vq9okGa](https://youtu.be/Z_KbQQHbUL0?si=UDZTVn0I6vq9okGa)

## **DAFTAR PUSTAKA**