

Laporan Tugas Besar 3 IF2211 Strategi Algoritma
Pemanfaatan *Pattern Matching* dalam Membangun Sistem Deteksi Individu
Berbasis Biometrik Melalui Citra Sidik Jari
Semester II Tahun 2023/2024



Disusun oleh:

Kelompok 16 - Albertus

- | | | |
|----|-------------------------|----------|
| 1. | Andi Marihot Sitorus | 13522138 |
| 2. | Nicholas Reymond Sihite | 13522144 |
| 3. | Muhammad Dzaki Arta | 13522149 |

INSTITUT TEKNOLOGI BANDUNG

2024

Daftar Isi

Daftar Isi	2
BAB I Deskripsi Masalah	3
BAB II Landasan Teori	4
2.1. Dasar Teori	4
2.1.1. Algoritma KMP	4
2.1.2. Algoritma BM	4
2.1.3. Regular Expression	5
2.2. Pengukuran Persentase Kemiripan dengan Hamming Distance	6
2.3. Penjelasan Singkat Aplikasi dan Basis Data	7
BAB III Analisis Pemecahan Masalah	8
3.1. Langkah Pemecahan Masalah	8
3.1.1. Pemodelan Masalah dalam Algoritma KMP	8
3.1.2. Pemodelan Masalah dalam Algoritma BM	9
3.1.3. Pemodelan Masalah dalam Regular Expression	9
3.2. Arsitektur dan Fitur Aplikasi	10
3.3. Contoh Ilustrasi Kasus	11
3.3.1. Ilustrasi Kasus dengan Algoritma KMP	11
3.3.2. Ilustrasi Kasus dengan Algoritma BM	12
BAB IV Implementasi dan Pengujian	13
4.1. Spesifikasi Teknis Program	13
4.2. Penjelasan Cara Penggunaan Aplikasi	22
4.3. Hasil Pengujian	24
4.4. Analisis Hasil Pengujian	41
Bab V Kesimpulan	43
5.1. Kesimpulan	43
5.2. Saran	43
Lampiran	44
Daftar Pustaka	44

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.

Di dalam Tugas Besar 3 ini, mahasiswa 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. Dasar Teori

2.1.1. Algoritma KMP

Algoritma KMP (*Knuth-Morris-Pratt*) adalah salah satu algoritma pencocokan *string* yang mencocokkan pola dari kiri ke kanan. Dasar pembuatan algoritma ini adalah bagaimana cara melakukan pergeseran pola tanpa melakukan perbandingan yang boros seperti algoritma *brute force*. Jawabannya adalah: misalkan pola merupakan sebuah larik karakter $P[0..n]$, geser pola sampai sufiks terbesar dari $P[1..n]$ yang sama dengan infiks $P[0..n]$. Algoritma ini memanfaatkan sebuah fungsi pinggiran (*border function*) untuk mempermudah pergeseran pola ketika ketidakcocokan terjadi. Fungsi pinggiran merupakan sebuah larik berukuran $n - 1$ yang berisi ukuran sufiks-infiks untuk setiap panjang upalarik dari 0 sampai $n - 1$. Berikut merupakan contoh fungsi pinggiran dari pola “*abcab*”.

Tabel 2.1.1.1. Fungsi Pinggiran Pola “*abcab*”

$k (0..n - 1)$	0	1	2	3
$b(k)$	0	1	0	1

Misalkan pola tersebut ingin dicari pada *string* “*abcaabcab*”. Berikut merupakan perbandingannya.

Tabel 2.1.1.2. Ilustrasi Pencocokan dengan Algoritma KMP

a	b	c	a	a	b	c	a	b
a	b	c	a	b				
geser pola ke indeks $b(4-1) = 1$								
geser pola ke $b(1-1) = 0$			a	b	c	a	b	
pola ditemukan!								

2.1.2. Algoritma BM

Algoritma BM (*Boyer-Moore*) adalah salah satu algoritma pencocokan *string* yang mencocokkan pola dari kanan ke kiri. Algoritma ini juga berusaha menjawab pertanyaan bagaimana cara melakukan pergeseran pola tanpa melakukan perbandingan yang boros. Jawabannya adalah dengan: jika karakter

yang tidak cocok ada dan berada di sebelah kiri pola, geser ke indeks tersebut; jika karakter yang tidak cocok ada dan berada di sebelah kanan pola, geser satu kali ke kanan; atau jika karakter yang tidak cocok tidak ada pada pola, geser pola sampai tepat 1 karakter melewati karakter yang tidak cocok tersebut. Algoritma ini memanfaatkan sebuah tabel *last occurrence* untuk mempermudah pergeseran. Tabel ini menyimpan informasi kapan terakhir kali sebuah karakter muncul dalam pola, lalu pola akan bergeser ke indeks tersebut. Berikut merupakan contoh tabel *last occurrence* dari pola “*abcab*”.

Tabel 2.1.2.1. Tabel Last Occurrence Pola “abcab”

karakter	a	b	c
last occurrence	3	4	2

Misalkan pola tersebut ingin dicari pada *string* “*abcaabcab*”. Berikut merupakan perbandingannya.

Tabel 2.1.1.2. Ilustrasi Pencocokan dengan Algoritma BM

a	b	c	a	a	b	c	a	b
a	b	c	a	b	geser ke <i>last occurrence</i> a, yaitu 3			
	a	b	c	a	b	geser ke kanan sebanyak 1		
		a	b	c	a	b	geser ke 2	
				a	b	c	a	b
							pola ditemukan!	

2.1.3. Regular Expression

Regular Expression merupakan sebuah metode *string matching* melalui sekumpulan kombinasi karakter dengan pola tertentu. Kombinasi karakter yang dibentuk akan dicocokkan dengan *string* yang ingin dicari. Keuntungan penggunaan *regular expression* adalah dapat menemukan kecocokan pada beberapa *string* yang berbeda, tetapi memiliki pola yang sama. Sebagai contoh, misalnya ada dua *string*, yaitu “*kota*” dan “*KOTA*” dan kita ingin mencari *string* “*Kota*”. Ketika melakukan *string matching* dengan metode lain, kedua string tersebut akan dianggap berbeda sehingga tidak akan ditemukan kecocokan. Namun, dengan penggunaan *regular expression*, dapat dirancang kombinasi karakter sehingga hasil pencarian “*Kota*” akan memunculkan kedua *string* tersebut.

Berikut merupakan tabel yang berisi beberapa notasi/sintaks yang umumnya digunakan dalam *regular expression*.

Tabel 2.1.3.1. Beberapa Notasi dalam Regular Expression

Notasi	Penjelasan
.	Karakter apapun kecuali <i>newline</i>
\.	Karakter titik
^	Awal <i>string</i>
\$	Akhir <i>string</i>
[abc]	Karakter a, b, atau c
[a-z]	Karakter a sampai z
[^abc]	Karakter apapun selain a, b, atau c
a b	Karakter a atau b
?	0 atau 1 dari elemen sebelumnya
*	0 atau lebih dari elemen sebelumnya
+	Minimal 1 dari elemen sebelumnya
{n}	Tepat n dari elemen sebelumnya
{n, }	Minimal n dari elemen sebelumnya
{m, n}	Antara m sampai n (inklusif) dari elemen sebelumnya

2.2. Pengukuran Persentase Kemiripan dengan *Hamming Distance*

Hamming Distance adalah sebuah metode untuk menghitung besarnya perbedaan antara dua *string* atau vektor yang panjangnya sama. Metode ini menghitung secara literal ada berapa karakter yang berbeda antara kedua *string* yang dibandingkan. Sebagai contoh, *string* “abgdf” dan “abcde” berbeda sebanyak dua karakter, maka *Hamming Distance*-nya adalah 2, atau dapat dinormalisasi dengan membaginya dengan panjang *string*, yaitu 0,4.

Dalam aplikasi Albertus Fingerprint Matcher ini, *hamming distance* akan digunakan apabila algoritma *KMP* dan *BM* gagal menemukan kecocokan antara sidik jari masukan dengan sidik jari pada basis data. Batasan persentase yang dipilih adalah 70% artinya sidik jari hanya akan dianggap mirip apabila persentase kemiripannya lebih besar atau sama dengan 70%. Batas ini dipilih sebab setelah melakukan beberapa kali percobaan untuk mencocokkan gambar berkategori “Altered-Hard” dengan gambar pada basis data, hasil yang didapat rata-rata adalah di atas 70%.

2.3. Penjelasan Singkat Aplikasi dan Basis Data

Aplikasi Albertus Fingerprint Matcher merupakan aplikasi dengan fungsionalitas utama mencari sidik jari yang paling mirip dengan sidik jari masukan pengguna dari daftar sidik jari yang ada pada basis data. Ketika ditemukan, akan tampil informasi biodata individu dengan sidik jari yang paling mirip dengan sidik jari masukan.

Data sidik jari yang digunakan berasal dari [Sokoto Coventry Fingerprint Dataset](#) pada folder “Real” dengan total 6000 sidik jari sementara biodata untuk 600 orang dengan sidik jari tersebut dibuat dengan memanfaatkan situs [Mockaroo](#). Implementasi basis data (insersi data) dilakukan dengan memanfaatkan program yang dibuat dengan Bahasa Python untuk mempermudah pekerjaan. Program Python tersebut juga sekaligus mengubah nama-nama pada tabel biodata ke dalam Bahasa Alay dengan mengadaptasikan algoritma yang dipakai pada situs [Alay Generator](#).

BAB III

Analisis Pemecahan Masalah

3.1. Langkah Pemecahan Masalah

Aplikasi ini dirancang untuk menyelesaikan masalah pencocokan sidik jari. Urutan langkah-langkahnya adalah sebagai berikut.

1. Pengguna memasukkan sidik jari yang ingin dicocokkan
2. Pengguna memilih algoritma (*KMP* atau *BM*) yang ingin digunakan untuk mencocokkan sidik jari
3. Aplikasi akan mengambil 60 pixel dari gambar masukan lalu memanfaatkan algoritma *KMP* atau *BM* (dijelaskan pada bagian 3.1.1. dan 3.1.2.) untuk mencari sidik jari yang paling mirip
4. Aplikasi memanfaatkan *regular expression* (dijelaskan pada bagian 3.1.3.) untuk mencari biodata individu dengan sidik jari yang paling mirip
5. Aplikasi menampilkan hasil pencarian pada layar

Pengambilan 60 pixel dilakukan dengan mengubah gambar menjadi rangkaian karakter ASCII, lalu mengambil 60 karakter ASCII yang berada di tengah gambar. Alasan pemilihan 60 pixel adalah karena setelah beberapa percobaan, ditemukan bahwa pengambilan 60 pixel memberikan hasil yang cukup baik tanpa memakan waktu yang banyak. Selain itu, gambar yang ada pada basis data merupakan gambar *grayscale*, artinya 1 pixel berukuran 1 byte, sehingga 1 pixel dapat diwakili oleh 1 karakter ASCII yang berukuran 1 byte juga.

3.1.1. Pemodelan Masalah dalam Algoritma *KMP*

Algoritma yang kami buat untuk menyelesaikan permasalahan dengan *KMP* harus memiliki *string* dan *pattern*. *String* yang kami gunakan merupakan representasi dari gambar yang berada di basis data. Algoritma *KMP* ini akan me-*resize* gambar yang ada di basis data menjadi 60x60 pixel yang kemudian diubah menjadi biner dan menjadi ASCII. Pattern yang dibandingkan merupakan representasi dari gambar input yang diubah menjadi ASCII dan diambil 60

pixelnya. Algoritma KMP membandingkan pattern dengan string secara iteratif dengan menggunakan fungsi pinggiran. Fungsi pinggiran akan menerima pattern dan mengembalikan *list of integer* yang merepresentasikan hasil dari fungsi pinggiran *Knuth-Morris-Pratt Algorithm*. Setelah menerima *string* dan *pattern*, Algoritma KMP akan memanggil fungsi pinggiran dengan parameter *pattern*, lalu melakukan iterasi dengan bantuan fungsi pinggiran tersebut sesuai dengan *Knuth-Morris-Pratt Algorithm*.

3.1.2. Pemodelan Masalah dalam Algoritma BM

Algoritma *Boyer Moore* yang kami buat untuk menyelesaikan permasalahan pencocokan *string* harus menerima dua masukan berupa *String* yaitu *text* dan *pattern*. Masukan *text* merupakan representasi dari gambar yang berada dalam *database*. Gambar akan di-*resize* menjadi ukuran 60x60 piksel yang kemudian diubah menjadi biner dan karakter ASCII. Masukan *pattern* yang dibandingkan dengan *text* merupakan 60 piksel yang diambil dari gambar masukan yang telah di-*resize* menjadi 60x60 dan diubah menjadi karakter ASCII. Algoritma BM membandingkan *pattern* dengan *text* secara iteratif sesuai dengan fungsi *Last Occurrence*. Fungsi *Last Occurrence* menerima masukan *pattern* dan mengembalikan *list of integer* yang merupakan jarak indeks setiap karakter dalam *pattern* terhadap karakter paling akhir. Dengan menggunakan fungsi *Last Occurrence*, *pattern* akan diiterasi sepanjang *text* untuk menemukan *string* yang cocok sesuai dengan konsep algoritma *Boyer Moore*.

3.1.3. Pemodelan Masalah dalam *Regular Expression*

Tabel sidik_jari pada basis data menyimpan data *path* menuju lokasi gambar sidik jari dan nama pemilik sidik jari tersebut. Permasalahan yang dihadapi adalah nama pada tabel biodata merupakan nama yang sudah dienkripsi ke Bahasa Alay. Oleh karena itu, aplikasi memanfaatkan *regular expression* untuk mencocokkan nama pada tabel sidik_jari dan biodata. *Converter Bahasa Alay* yang digunakan memiliki beberapa kemungkinan perubahan huruf. Berikut merupakan perubahan huruf yang mungkin terjadi.

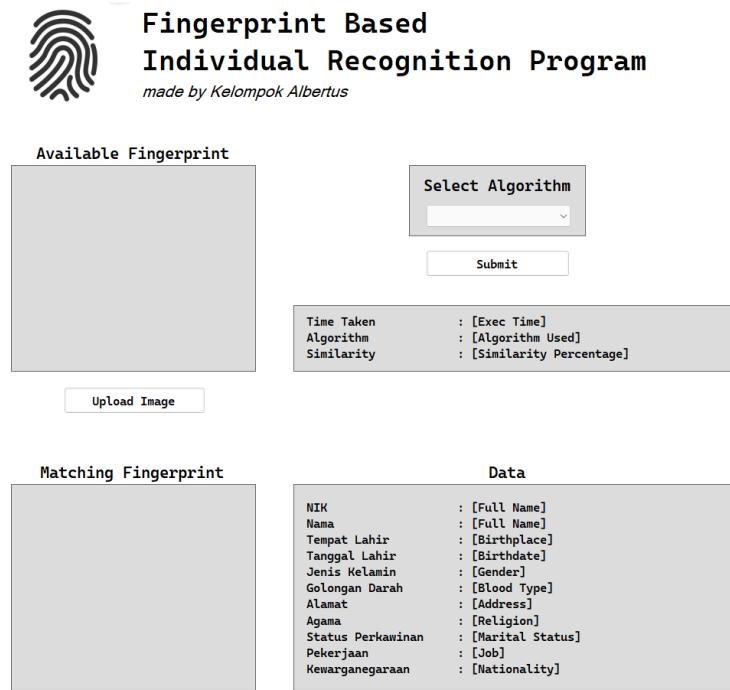
Tabel 3.1.3.1. Kemungkinan Enkripsi Huruf-Huruf ke B

Bahasa Normal	Bahasa Alay
a atau A	a, A, 4, atau dihilangkan (?)
e atau E	e, E, 3, atau dihilangkan (?)
g atau G	g, G, atau 6
i atau I	i, I, 1, atau dihilangkan (?)
o atau O	o, O, 0, atau dihilangkan (?)
s atau S	s, S, atau 5
z atau Z	z, Z, atau 2
u atau U	u, U, atau dihilangkan (?)
Lainnya	Huruf kecil atau kapital huruf tersebut

Oleh karena itu, dibuat sebuah *map* yang memetakan huruf-huruf tersebut ke hasil konversi yang mungkin. Untuk setiap huruf yang ada pada nama asli, akan dibentuk *regular expression* dari *map* yang telah dibuat. Sebagai contoh, dari nama “Albertus” akan dibuat sebuah *regular expression*, yaitu “^ [aA4]?[IL][bB][eE3]?[rR][tT][uU]?[sS5]\$”. *Regular expression* ini kemudian akan dibandingkan dengan nama-nama yang ada pada basis data. Ketika ada nama yang cocok dengan *regular expression* tersebut, aplikasi akan mengambil semua data individu tersebut dan menampilkannya pada layar.

3.2. Arsitektur dan Fitur Aplikasi

Aplikasi yang telah dibangun merupakan aplikasi *GUI* berbasis *desktop* yang dibuat dalam Bahasa C# dengan *framework* .NET dan basis data MySQL. Pengembangan aplikasi dilakukan menggunakan *IDE* Visual Studio 2022 dan Visual Studio Code. Fitur utama aplikasi adalah mencocokkan sebuah sidik jari masukan dengan daftar sidik jari yang ada pada basis data. Jika ada sidik jari yang mirip, akan ditampilkan informasi biodata individu dengan sidik jari tersebut. Berikut merupakan tampilan aplikasinya.



Gambar 3.2.1. Tampilan Aplikasi

3.3. Contoh Ilustrasi Kasus

Sebagai ilustrasi, misalkan terdapat sebuah gambar yang telah dikonversi menjadi karakter ASCII dan pada bagian tengahnya terdapat 5 karakter ASCII “y&:y&”. Selain itu, terdapat gambar pada basis data yang diwakili oleh karakter ASCII “30y&3jy&:y&8s1#”. Akan dilakukan perbandingan dengan algoritma *KMP* dan *BM* (sebagai catatan, skala diperkecil untuk mempermudah ilustrasi).

3.3.1. Ilustrasi Kasus dengan Algoritma KMP

Tabel 3.3.1.1 Fungsi Pinggiran

$k(0..n-1)$	0	1	2	3
$b(k)$	0	0	0	1

Tabel 3.3.1.2. Ilustrasi Kasus dengan Algoritma KMP

3.3.2. Ilustrasi Kasus dengan Algoritma *BM*

Tabel 3.3.1.1 Fungsi Pinggiran

i(0..n-1)	0	1	2	3
char(i)	o	n	t	a
L(i)	0	0	0	1

Tabel 3.3.1.2. Ilustrasi Kasus dengan Algoritma KMP

s	f	q	t	r	o	n	t	a	d	w	3	n	l	u
o	n	t	a											
	o	n	t	a										
					o	n	t	a						

BAB IV

Implementasi dan Pengujian

4.1. Spesifikasi Teknis Program

Tabel 4.1.1. Spesifikasi Fungsi dan Prosedur Program

No	Nama	Deskripsi
1	<pre>main() { namespace Albertus_FingerprintMatcher { internal static class Program { [STAThread] static void Main() { Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new Form()); } } } }</pre>	Prosedur ini digunakan untuk menjalankan program utama.
2	<pre>FindMatch(String input, String algoritma) → List<String></pre>	Fungsi ini digunakan untuk mencari sidik jari yang cocok. Parameter input merupakan <i>string</i> yang berisi <i>path</i> sidik jari masukan dan parameter algoritma merupakan <i>string</i> yang berisi algoritma yang dipilih. Fungsi akan mengembalikan sebuah <i>List<String></i> yang berisi informasi <i>path</i> menuju sidik jari paling mirip, nama pemilik sidik jari, dan persentase kemiripannya. <p><code>public static List<String> FindMatch(String input, String algoritma){</code></p> <p><code>public static List<String> FindMatch(String input, String algoritma){</code></p> <p><code> bool find = false;</code></p> <p><code> List<String> result = new List<String>();</code></p> <p><code> List<String> path = database.GetPath();</code></p> <p><code> List<String> name = database.GetName();</code></p> <p><code> String pattern = imgMethod.imgToASCII_60Px1(input);</code></p> <p><code> if (algoritma.Equals("KMP"))</code></p> <p><code> {</code></p> <p><code> for (int i = 0; i < path.Count; i++)</code></p> <p><code> {</code></p> <p><code> String currPath = "../../../../../" + path[i];</code></p> <p><code> String text = imgMethod.imgToASCII(currPath);</code></p> <p><code> find = KMP.KMPSearch(text, pattern);</code></p> <p><code> if (find)</code></p> <p><code> {</code></p> <p><code> result.Add(path[i]);</code></p> <p><code> result.Add(name[i]);</code></p> <p><code> result.Add("100%");</code></p> <p><code> result.Add("True");</code></p>

```
                break;
            }
        }
    }
    else
    {
        Console.WriteLine("Masuk BM");
        for (int i = 0; i < path.Count; i++)
        {
            String currPath = "../../../../../" + path[i];
            String text = imgMethod.imgToASCII(currPath);
            find = BM.BMMMatch(text, pattern);
            if (find)
            {
                result.Add(path[i]);
                result.Add(name[i]);
                result.Add("100%");
                result.Add("True");
                break;
            }
        }
        if (!find)
        {
            double percentage = 0;
            int idx = 0;
            Bitmap bitmapPattern = imgMethod.ResizeImage(input, 100, 100);
            String fullPattern = imgMethod.ResizedImgToASCII(bitmapPattern);
            for (int i = 0; i < path.Count; i++)
            {
                String currPath = "../../../../../" + path[i];
                Bitmap bitmaptext = imgMethod.ResizeImage(currPath, 100, 100);
                String text = imgMethod.ResizedImgToASCII(bitmaptext);
                double current = HammingDistance.ComputeHammingDistance(text,
fullPattern);
                if (current > percentage)
                {
                    percentage = current;
                    idx = i;
                }
            }
            result.Add(path[idx]);
            result.Add(name[idx]);
            result.Add((percentage * 100).ToString("F2") + "%");
            if (percentage * 100 >= 70)
            {
                result.Add("True");
            }
            else
            {
                result.Add("False");
            }
        }
    }
}
```

	<pre> return result; } </pre>	
3	<pre> FindBiodata(String namaAsli) → List<String> </pre>	Fungsi ini digunakan untuk mencari biodata orang dengan sidik jari yang paling mirip dengan masukan. Parameter namaAsli merupakan <i>string</i> yang berisi nama asli individu dengan sidik jari yang paling mirip. Fungsi akan mengembalikan List<String> yang berisi informasi biodata orang tersebut.
	<pre> public static List<String> FindBiodata(String namaAsli) { List<String> result = new List<String>(); List<String> encryptedNames = database.GetNamaLengkap(); int idx = -1; String regex = RegEx.generateRegEx(namaAsli); for (int i = 0; i < encryptedNames.Count; i++) { if (RegEx.IsMatch(encryptedNames[i], regex)) { idx = i; break; } } if (idx != -1) { result.Add(database.GetNIK()[idx]); result.Add(namaAsli); result.Add(database.GetTempatLahir()[idx]); result.Add(database.GetTanggalLahir()[idx]); result.Add(database.GetJenisKelamin()[idx]); result.Add(database.GetGolonganDarah()[idx]); result.Add(database.GetAlamat()[idx]); result.Add(database.GetAgama()[idx]); result.Add(database.GetStatusPerkawinan()[idx]); result.Add(database.GetPekerjaan()[idx]); result.Add(database.GetKewarganegaraan()[idx]); } return result; } </pre>	
4	<pre> generateRegEx(String input) → String </pre>	Fungsi ini merupakan fungsi yang digunakan untuk membuat <i>regular expression</i> berdasarkan nama asli individu. Parameter input merupakan <i>string</i> yang berisi nama asli individu yang ingin dibuat menjadi <i>regular expression</i> . Fungsi mengembalikan <i>string</i> yang berisi <i>regular expression</i> dari nama input.
	<pre> public static string generateRegEx(string input){ string regex = ""; foreach (char c in input.ToLower()) { regex += c.ToString(); } return regex; } </pre>	

	<pre> if (wordPossibleMap.ContainsKey(c)){ regex += wordPossibleMap[c]; } else { regex += c; } } regex = "^" + regex + "\$"; return regex; } </pre>	
5	<code>isMatch(String input, String regEx) → bool</code> <pre> public static bool isMatch(string input, string regEx){ return Regex.IsMatch(input, regEx); } </pre>	Fungsi ini merupakan fungsi yang digunakan untuk mencocokkan nama dengan Bahasa Alay dengan <i>regular expression</i> . Parameter input merupakan <i>string</i> yang berisi nama dalam Bahasa Alay dan parameter regEx merupakan <i>string</i> yang berisi <i>regular expression</i> . Fungsi mengembalikan <i>true</i> jika cocok dan <i>false</i> jika tidak.
6	<code>ResizeImage(string filePath, int width, int height) → Bitmap</code> <pre> public static Bitmap ResizeImage(string filePath, int width, int height) { using (Image originalImage = Image.FromFile(filePath)) { Bitmap resizedImage = new Bitmap(width, height); resizedImage.SetResolution(originalImage.HorizontalResolution, originalImage.VerticalResolution); using (Graphics graphics = Graphics.FromImage(resizedImage)) { graphics.InterpolationMode = InterpolationMode.HighQualityBicubic; graphics.DrawImage(originalImage, 0, 0, width, height); } return resizedImage; } } </pre>	Fungsi ini merupakan fungsi yang digunakan untuk mengubah ukuran gambar sesuai parameter <i>width</i> dan <i>height</i> .
7	<code>imgToASCII_60PxI(string filePath) → string</code> <pre> public static string imgToASCII_60PxI(string filePath) { Bitmap resized = ResizeImage(filePath, 60, 60); </pre>	Fungsi ini merupakan fungsi yang digunakan untuk mengambil 60 pixel dari gambar masukan dengan mengubahnya menjadi karakter ASCII. Parameter filePath merupakan <i>string</i> yang berisi <i>path</i> menuju lokasi gambar masukan. Fungsi mengembalikan <i>string</i> yang berisi karakter ASCII dari 30 pixel gambar masukan.

	<pre> byte[] bytes = BitmapToByteArray(resized); string binaryString = Encoding.ASCII.GetString(bytes); binaryString = binaryString.Substring(binaryString.Length / 2 - 30, 60); return binaryString; } </pre>	
8	<pre> imgToASCII(string filePath) → string </pre> <p> public static string imgToASCII(string filepath) { Bitmap resized = ResizeImage(filepath, 60, 60); byte[] bytes = BitmapToByteArray(resized); string binaryString = Encoding.ASCII.GetString(bytes); return binaryString; } </p>	Fungsi ini merupakan fungsi yang digunakan untuk mengubah sebuah gambar menjadi karakter ASCII. Parameter filePath merupakan <i>string</i> yang berisi <i>path</i> menuju lokasi gambar dari basis data. Fungsi mengembalikan karakter ASCII yang mewakili gambar pada basis data.
9	<pre> KMPSearch(string text, string pattern) → bool </pre> <p> public static bool KMPSearch(string text, string pattern) { int n = text.Length; int m = pattern.Length; int[] b = ComputeBorder(pattern); int i = 0; int j = 0; while (i < n) { if (pattern[j] == text[i]) { if (j == m - 1) return true; i++; j++; } else if (j > 0) { j = b[j - 1]; } else { i++; } } } </p>	Fungsi ini merupakan fungsi yang digunakan untuk mencari kecocokan antara sebuah pola dan sebuah teks dengan algoritma <i>KMP</i> . Parameter text merupakan <i>string</i> yang berisi karakter ASCII gambar pada basis data dan parameter pattern merupakan <i>string</i> yang berisi 60 pixel karakter ASCII dari gambar masukan. Fungsi mengembalikan <i>true</i> jika cocok, <i>false</i> jika tidak.

	<pre> } } return false; } </pre>	
10	<p>ComputeBorder(string pattern) → int[]</p>	Fungsi ini merupakan fungsi yang digunakan untuk membuat fungsi pinggiran yang digunakan algoritma KMP. Parameter pattern merupakan <i>string</i> yang berisi 60 pixel karakter ASCII dari gambar masukan. Fungsi mengembalikan <i>array of integer</i> yang mewakili fungsi pinggiran.
	<pre> public static int[] ComputeBorder(string pattern) { int[] b = new int[pattern.Length]; b[0] = 0; int m = pattern.Length; int j = 0; int i = 1; while (i < m) { if (pattern[j] == pattern[i]) { // j+1 chars match b[i] = j + 1; i++; j++; } else if (j > 0) { // j follows matching prefix j = b[j - 1]; } else { // no match b[i] = 0; i++; } } return b; } </pre>	
11	<p>BMMatch(string text, string pattern) → bool</p>	Fungsi ini merupakan fungsi yang digunakan untuk mencari kecocokan antara sebuah pola dan sebuah teks dengan algoritma BM. Parameter text merupakan <i>string</i> yang berisi karakter ASCII gambar pada basis data dan parameter pattern merupakan <i>string</i> yang berisi 60 pixel karakter ASCII dari gambar masukan. Fungsi mengembalikan <i>true</i> jika cocok, <i>false</i> jika tidak.
	<pre> public static bool BMMatch(string text, string pattern) { </pre>	

	<pre> int n = text.Length; int m = pattern.Length; int[] bmTable = ComputeTable(pattern); int i = m - 1; while (i < n) { int shift = bmTable[(int)text[i]]; int k = 0; int j = m - 1; while (j >= 0 && pattern[j] == text[i]) { k++; i--; j--; } if (j < 0) { return true; } else { i += k; i += shift; } } return false; } </pre>	
12	ComputeTable(string pattern) → int[]	Fungsi ini merupakan fungsi yang digunakan untuk membuat tabel <i>last occurrence</i> yang digunakan algoritma <i>BM</i> . Parameter pattern merupakan <i>string</i> yang berisi 60 pixel karakter ASCII dari gambar masukan. Fungsi mengembalikan <i>array of integer</i> yang mewakili tabel <i>last occurrence</i> .
	<pre> public static int[] ComputeTable(string pattern){ int m = pattern.Length; int[] bmTable = new int[256]; for (int i = 0; i < 256; i++){ bmTable[i] = m; } for (int i = 0; i < m - 1; i++){ bmTable[pattern[i]] = m - 1 - i; } return bmTable; } </pre>	
13	ComputeHammingDistance(string text, string pattern) → double	Fungsi ini merupakan fungsi yang digunakan untuk menghitung banyaknya perbedaan huruf dari string yang dibuat dari gambar, yaitu antara gambar masukan dan gambar pada basis data. Parameter text merupakan

		<p><i>string</i> yang berisi karakter ASCII dari gambar pada basis data dan parameter pattern merupakan <i>string</i> yang berisi karakter ASCII dari gambar masukan. Fungsi mengembalikan <i>double</i> dalam rentang 0 sampai 1 yang mewakili kemiripan kedua gambar.</p>
	<pre>public static double ComputeHammingDistance(string text, string pattern) { int len = Math.Min(text.Length, pattern.Length); int maxLen = Math.Max(text.Length, pattern.Length); int distance = 0; for (int i = 0; i < len; i++) { if (text[i] != pattern[i]) { distance++; } } distance += maxLen - len; return 1.0 - (double)distance / maxLen; }</pre>	
14	Connect()	Prosedur ini digunakan untuk menghubungkan aplikasi ke basis data
	<pre>public void Connect() { try { connection.Open(); if (connection.State == System.Data.ConnectionState.Open) { Console.WriteLine("Connected to SQL Server database"); } } catch (MySqlException ex) { Console.WriteLine("Error while connecting to SQL Server: " + ex.Message); } }</pre>	
15	Disconnect()	Prosedur ini digunakan untuk memutus hubungan aplikasi ke basis data
	<pre>public void Disconnect() { try { connection.Close(); if (connection.State == System.Data.ConnectionState.Closed) { Console.WriteLine("Disconnected from SQL Server database"); } } }</pre>	

	<pre> catch (MySqlException ex) { Console.WriteLine("Error while disconnecting from SQL Server: " + ex.Message); } } </pre>	
16	<pre>Execute(string query) → DataTable</pre>	Fungsi ini merupakan fungsi yang digunakan untuk mengeksekusi <i>query</i> SQL dari aplikasi ke basis data. Parameter <i>query</i> merupakan <i>string</i> yang berisi perintah SQL yang akan dieksekusi. Fungsi mengembalikan <i>DataTable</i> yang berisi data-data hasil eksekusi perintah SQL.
	<pre> public DataTable Execute(string query) { DataTable dataTable = new DataTable(); try { if (connection.State != System.Data.ConnectionState.Open) { Console.WriteLine("Connection is not open"); return dataTable; } MySqlCommand cmd = new MySqlCommand(query, connection); MySqlDataAdapter adapter = new MySqlDataAdapter(cmd); adapter.Fill(dataTable); Console.WriteLine("Query executed successfully"); } catch (MySqlException ex) { Console.WriteLine("Error executing query: " + ex.Message); } return dataTable; } </pre>	
17	<pre>ResizedImgToASCII(Bitmap bitmap) → String</pre>	Fungsi ini merupakan fungsi yang digunakan untuk mengubah sebuah gambar menjadi karakter ASCII. Parameter <i>bitmap</i> merupakan <i>bitmap</i> yang berisi gambar yang telah diubah ukurannya. Fungsi mengembalikan karakter ASCII yang mewakili gambar pada basis data.
	<pre> public static string ResizedImgToASCII(Bitmap bitmap) { using (MemoryStream memoryStream = new MemoryStream()) { bitmap.Save(memoryStream, System.Drawing.Imaging.ImageFormat.Bmp); byte[] bytes = memoryStream.ToArray(); string asciiString = Encoding.ASCII.GetString(bytes); return asciiString; } } </pre>	

4.2. Penjelasan Cara Penggunaan Aplikasi

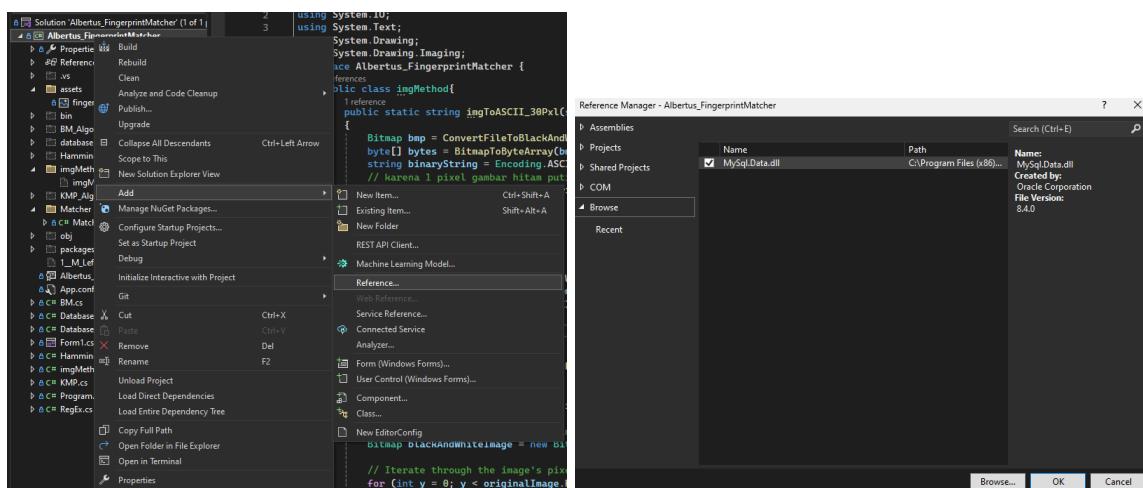
Aplikasi yang dibuat tidak di-*build* menjadi *file executable*. Oleh karena itu, untuk menjalankan aplikasi, dibutuhkan *download* dan instalasi beberapa *requirements*, yaitu Visual Studio 2022 dengan .NET Desktop Development, MySQLConnector untuk Visual Studio 2022, dan SQL DBMS, misalnya MariaDB.

Hal pertama yang perlu dilakukan adalah menyiapkan basis data aplikasi. Masukkan file SQL bernama “tubes3_albertus.sql” yang berada pada folder “src/Albertus_FingerprintMatcher/database” ke sebuah basis data, misalnya “tubes3_albertus”.

```
MariaDB [(none)]> use tubes3_albertus;
Database changed
MariaDB [tubes3_albertus]> source tubes3_albertus.sql;
```

Gambar 4.2.1. Insersi Basis Data

Setelah memasukkan basis data, buka *solution* “Albertus_FingerprintMatcher.sln” yang berada pada *folder* “src/Albertus_FingerprintMatcher” pada Visual Studio 2022. Klik kanan pada *file* “Albertus_FingerprintMatcher.cs”, Add > Reference... > Browse, lalu tambahkan *file* .msi MySQLConnector yang telah diunduh.



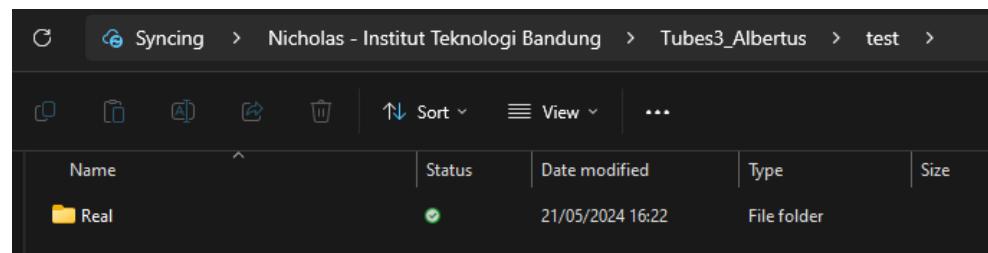
Gambar 4.2.2. Instalasi MySQLConnector

Selanjutnya, buka file “DatabaseManager.cs” lalu ganti nama Database, Uid, dan Pwd pada konstruktor DatabaseManager sesuai dengan basis data yang telah dibuat.

```
public DatabaseManager()
{
    string connectionString = "Server=localhost;Database=tubes3_albertus;Uid=root;Pwd=[password_anda];";
    connection = new MySqlConnection(connectionString);
}
```

Gambar 4.2.3. Modifikasi Variabel DatabaseManager

Setelah itu, lakukan *unzip* terhadap “Real.zip” yang berada pada folder “test”. Pastikan hasil *unzip* menghasilkan struktur folder “test/Real/[isi].BMP”.



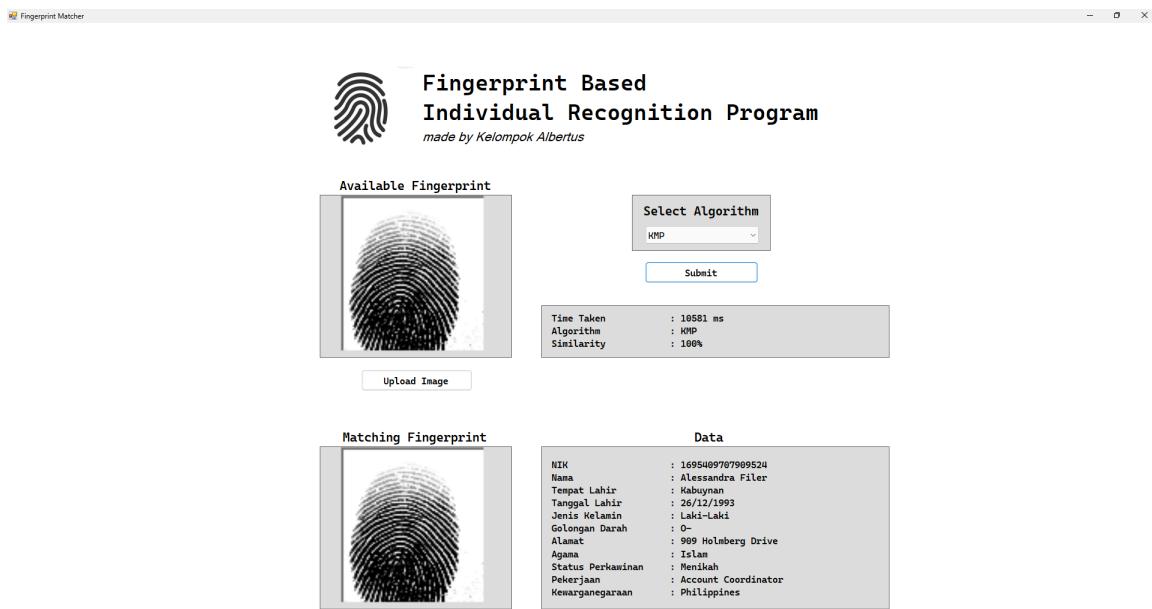
Gambar 4.2.4. Insersi Folder “Real” SOCOFing ke Folder “test”

Jalankan aplikasi dengan menekan tombol “Start” pada Visual Studio 2022. Tampilan awal aplikasi adalah sebagai berikut.



Gambar 4.2.5. Tampilan Awal Aplikasi

Masukkan gambar yang ingin dicocokkan beserta algoritma yang ingin digunakan, lalu tekan tombol “Submit”. Hasil akan ditampilkan sesaat setelah pencarian selesai dilakukan. Hasil pencarian berupa waktu eksekusi, jenis algoritma yang digunakan, persentase kemiripan, gambar sidik jari yang cocok, beserta biodata individu dengan sidik jari yang cocok.



Gambar 4.2.6. Hasil Pencarian Sidik Jari

4.3. Hasil Pengujian

Tabel 4.3.1. Hasil Pengujian Program

No	Kasus
1	Input : test/Real/1__M_Left_index_finger.BMP

	<p style="text-align: center;"> Fingerprint Based Individual Recognition Program <i>made by Kelompok Albertus</i></p> <p>Available Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: auto;"></div> <p>Matching Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: auto;"></div> <p>Upload Image</p> <p>Select Algorithm</p> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content; margin: 2px auto; text-align: center;">KMP</div> <p>Submit</p> <p>Time Taken : 964 ms Algorithm : KMP Similarity : 100%</p> <p>Data</p> <table border="0"><tr><td>NIK</td><td>:</td><td>1695409707909524</td></tr><tr><td>Nama</td><td>:</td><td>Alessandra Filer</td></tr><tr><td>Tempat Lahir</td><td>:</td><td>Kabuynan</td></tr><tr><td>Tanggal Lahir</td><td>:</td><td>26/12/1993</td></tr><tr><td>Jenis Kelamin</td><td>:</td><td>Laki-Laki</td></tr><tr><td>Golongan Darah</td><td>:</td><td>O-</td></tr><tr><td>Alamat</td><td>:</td><td>909 Holmberg Drive</td></tr><tr><td>Agama</td><td>:</td><td>Islam</td></tr><tr><td>Status Perkawinan</td><td>:</td><td>Menikah</td></tr><tr><td>Pekerjaan</td><td>:</td><td>Account Coordinator</td></tr><tr><td>Kewarganegaraan</td><td>:</td><td>Philippines</td></tr></table>	NIK	:	1695409707909524	Nama	:	Alessandra Filer	Tempat Lahir	:	Kabuynan	Tanggal Lahir	:	26/12/1993	Jenis Kelamin	:	Laki-Laki	Golongan Darah	:	O-	Alamat	:	909 Holmberg Drive	Agama	:	Islam	Status Perkawinan	:	Menikah	Pekerjaan	:	Account Coordinator	Kewarganegaraan	:	Philippines
NIK	:	1695409707909524																																
Nama	:	Alessandra Filer																																
Tempat Lahir	:	Kabuynan																																
Tanggal Lahir	:	26/12/1993																																
Jenis Kelamin	:	Laki-Laki																																
Golongan Darah	:	O-																																
Alamat	:	909 Holmberg Drive																																
Agama	:	Islam																																
Status Perkawinan	:	Menikah																																
Pekerjaan	:	Account Coordinator																																
Kewarganegaraan	:	Philippines																																



Fingerprint Based Individual Recognition Program

made by Kelompok Albertus

Available Fingerprint



Upload Image

Select Algorithm

BM

Submit

Time Taken : 992 ms
Algorithm : BM
Similarity : 100%

Matching Fingerprint



Data

NIK : 1695409707909524
Nama : Alessandra Filer
Tempat Lahir : Kabuynan
Tanggal Lahir : 26/12/1993
Jenis Kelamin : Laki-Laki
Golongan Darah : O-
Alamat : 909 Holmberg Drive
Agama : Islam
Status Perkawinan : Menikah
Pekerjaan : Account Coordinator
Kewarganegaraan : Philippines

2 Input : test/Real/35_M_Left_little_finger.BMP

Hasil	<p style="text-align: center;"> Fingerprint Based Individual Recognition Program <i>made by Kelompok Albertus</i></p> <p>Available Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; text-align: center;"></div> <p>Matching Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; text-align: center;"></div> <p>Select Algorithm</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">KMP</div> <p>Submit</p> <p>Time Taken : 2618 ms Algorithm : KMP Similarity : 100%</p> <p>Data</p> <table border="0" style="width: 100%;"><tr><td style="width: 50%; vertical-align: top;">NIK</td><td>: 4658180504106174</td></tr><tr><td>Nama</td><td>: Port Chiverton</td></tr><tr><td>Tempat Lahir</td><td>: Zevgolatei </td></tr><tr><td>Tanggal Lahir</td><td>: 16/08/1978</td></tr><tr><td>Jenis Kelamin</td><td>: Laki-Laki</td></tr><tr><td>Golongan Darah</td><td>: A-</td></tr><tr><td>Alamat</td><td>: 6 Tennessee Pass</td></tr><tr><td>Agama</td><td>: Buddha</td></tr><tr><td>Status Perkawinan</td><td>: Menikah</td></tr><tr><td>Pekerjaan</td><td>: Senior Financial Analyst</td></tr><tr><td>Kewarganegaraan</td><td>: Greece</td></tr></table>	NIK	: 4658180504106174	Nama	: Port Chiverton	Tempat Lahir	: Zevgolatei	Tanggal Lahir	: 16/08/1978	Jenis Kelamin	: Laki-Laki	Golongan Darah	: A-	Alamat	: 6 Tennessee Pass	Agama	: Buddha	Status Perkawinan	: Menikah	Pekerjaan	: Senior Financial Analyst	Kewarganegaraan	: Greece
NIK	: 4658180504106174																						
Nama	: Port Chiverton																						
Tempat Lahir	: Zevgolatei																						
Tanggal Lahir	: 16/08/1978																						
Jenis Kelamin	: Laki-Laki																						
Golongan Darah	: A-																						
Alamat	: 6 Tennessee Pass																						
Agama	: Buddha																						
Status Perkawinan	: Menikah																						
Pekerjaan	: Senior Financial Analyst																						
Kewarganegaraan	: Greece																						



Fingerprint Based Individual Recognition Program

made by Kelompok Albertus

Available Fingerprint



Upload Image

Select Algorithm

BM

Submit

Time Taken : 2541 ms
Algorithm : BM
Similarity : 100%

Matching Fingerprint



Data

NIK	:	4658180504106174
Nama	:	Port Chiverton
Tempat Lahir	:	Zevgolatei
Tanggal Lahir	:	16/08/1978
Jenis Kelamin	:	Laki-Laki
Golongan Darah	:	A-
Alamat	:	6 Tennessee Pass
Agama	:	Buddha
Status Perkawinan	:	Menikah
Pekerjaan	:	Senior Financial Analyst
Kewarganegaraan	:	Greece

3 Input : test/Real/526_F_Right_index_finger.BMP

Hasil	<p style="text-align: center;"> Fingerprint Based Individual Recognition Program <i>made by Kelompok Albertus</i></p> <p>Available Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; text-align: center;"></div> <p>Select Algorithm</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">KMP</div> <p style="text-align: center;">Submit</p> <p>Time Taken : 4218 ms Algorithm : KMP Similarity : 100%</p> <p>Upload Image</p> <p>Matching Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; text-align: center;"></div> <p>Data</p> <table border="0"><tr><td>NIK</td><td>:</td><td>8012228319666584</td></tr><tr><td>Nama</td><td>:</td><td>Pepito Dewdney</td></tr><tr><td>Tempat Lahir</td><td>:</td><td>Muṣṭaradah</td></tr><tr><td>Tanggal Lahir</td><td>:</td><td>01/08/1994</td></tr><tr><td>Jenis Kelamin</td><td>:</td><td>Perempuan</td></tr><tr><td>Golongan Darah</td><td>:</td><td>O-</td></tr><tr><td>Alamat</td><td>:</td><td>44466 Sauthoff Crossing</td></tr><tr><td>Agama</td><td>:</td><td>Buddha</td></tr><tr><td>Status Perkawinan</td><td>:</td><td>Belum Menikah</td></tr><tr><td>Pekerjaan</td><td>:</td><td>Senior Quality Engineer</td></tr><tr><td>Kewarganegaraan</td><td>:</td><td>Syria</td></tr></table>	NIK	:	8012228319666584	Nama	:	Pepito Dewdney	Tempat Lahir	:	Muṣṭaradah	Tanggal Lahir	:	01/08/1994	Jenis Kelamin	:	Perempuan	Golongan Darah	:	O-	Alamat	:	44466 Sauthoff Crossing	Agama	:	Buddha	Status Perkawinan	:	Belum Menikah	Pekerjaan	:	Senior Quality Engineer	Kewarganegaraan	:	Syria
NIK	:	8012228319666584																																
Nama	:	Pepito Dewdney																																
Tempat Lahir	:	Muṣṭaradah																																
Tanggal Lahir	:	01/08/1994																																
Jenis Kelamin	:	Perempuan																																
Golongan Darah	:	O-																																
Alamat	:	44466 Sauthoff Crossing																																
Agama	:	Buddha																																
Status Perkawinan	:	Belum Menikah																																
Pekerjaan	:	Senior Quality Engineer																																
Kewarganegaraan	:	Syria																																

	<p> Fingerprint Based Individual Recognition Program <i>made by Kelompok Albertus</i></p> <p>Available Fingerprint</p>  <p>Select Algorithm</p> <p>BM</p> <p>Submit</p> <p>Time Taken : 4086 ms Algorithm : BM Similarity : 100%</p> <p>Matching Fingerprint</p>  <p>Data</p> <table><tbody><tr><td>NIK</td><td>:</td><td>8012228319666584</td></tr><tr><td>Nama</td><td>:</td><td>Pepito Dewdney</td></tr><tr><td>Tempat Lahir</td><td>:</td><td>Muñir-adah</td></tr><tr><td>Tanggal Lahir</td><td>:</td><td>01/08/1994</td></tr><tr><td>Jenis Kelamin</td><td>:</td><td>Perempuan</td></tr><tr><td>Golongan Darah</td><td>:</td><td>O-</td></tr><tr><td>Alamat</td><td>:</td><td>44466 Sauthoff Crossing</td></tr><tr><td>Agama</td><td>:</td><td>Buddha</td></tr><tr><td>Status Perkawinan</td><td>:</td><td>Belum Menikah</td></tr><tr><td>Pekerjaan</td><td>:</td><td>Senior Quality Engineer</td></tr><tr><td>Kewarganegaraan</td><td>:</td><td>Syria</td></tr></tbody></table>	NIK	:	8012228319666584	Nama	:	Pepito Dewdney	Tempat Lahir	:	Muñir-adah	Tanggal Lahir	:	01/08/1994	Jenis Kelamin	:	Perempuan	Golongan Darah	:	O-	Alamat	:	44466 Sauthoff Crossing	Agama	:	Buddha	Status Perkawinan	:	Belum Menikah	Pekerjaan	:	Senior Quality Engineer	Kewarganegaraan	:	Syria
NIK	:	8012228319666584																																
Nama	:	Pepito Dewdney																																
Tempat Lahir	:	Muñir-adah																																
Tanggal Lahir	:	01/08/1994																																
Jenis Kelamin	:	Perempuan																																
Golongan Darah	:	O-																																
Alamat	:	44466 Sauthoff Crossing																																
Agama	:	Buddha																																
Status Perkawinan	:	Belum Menikah																																
Pekerjaan	:	Senior Quality Engineer																																
Kewarganegaraan	:	Syria																																
4	Input : test/Real/32__M_Right_little_finger.BMP																																	

Hasil	<p style="text-align: center;"> Fingerprint Based Individual Recognition Program <i>made by Kelompok Albertus</i></p> <p>Available Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; text-align: center;"> Upload Image</div> <p>Matching Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; text-align: center;"></div> <p>Select Algorithm</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">KMP</div> <p style="text-align: center;">Submit</p> <p>Time Taken : 2299 ms Algorithm : KMP Similarity : 100%</p> <p>Data</p> <table border="0" style="width: 100%;"><tr><td style="width: 50%;">NIK</td><td>: 4087189544497622</td></tr><tr><td>Nama</td><td>: Nonnah Sturdey</td></tr><tr><td>Tempat Lahir</td><td>: Veltruby</td></tr><tr><td>Tanggal Lahir</td><td>: 24/12/1988</td></tr><tr><td>Jenis Kelamin</td><td>: Laki-Laki</td></tr><tr><td>Golongan Darah</td><td>: A+</td></tr><tr><td>Alamat</td><td>: 03372 Longview Terrace</td></tr><tr><td>Aqama</td><td>: Kong Hu Chu</td></tr><tr><td>Status Perkawinan</td><td>: Belum Menikah</td></tr><tr><td>Pekerjaan</td><td>: Pharmacist</td></tr><tr><td>Kewarganegaraan</td><td>: Czech Republic</td></tr></table>	NIK	: 4087189544497622	Nama	: Nonnah Sturdey	Tempat Lahir	: Veltruby	Tanggal Lahir	: 24/12/1988	Jenis Kelamin	: Laki-Laki	Golongan Darah	: A+	Alamat	: 03372 Longview Terrace	Aqama	: Kong Hu Chu	Status Perkawinan	: Belum Menikah	Pekerjaan	: Pharmacist	Kewarganegaraan	: Czech Republic
NIK	: 4087189544497622																						
Nama	: Nonnah Sturdey																						
Tempat Lahir	: Veltruby																						
Tanggal Lahir	: 24/12/1988																						
Jenis Kelamin	: Laki-Laki																						
Golongan Darah	: A+																						
Alamat	: 03372 Longview Terrace																						
Aqama	: Kong Hu Chu																						
Status Perkawinan	: Belum Menikah																						
Pekerjaan	: Pharmacist																						
Kewarganegaraan	: Czech Republic																						



Fingerprint Based Individual Recognition Program

made by Kelompok Albertus

Available Fingerprint



Upload Image

Select Algorithm

BM

Submit

Time Taken : 2316 ms
Algorithm : BM
Similarity : 100%

Matching Fingerprint



Data

NIK : 4087189544497622
Nama : Nonnah Sturdey
Tempat Lahir : Veltruby
Tanggal Lahir : 24/12/1988
Jenis Kelamin : Laki-Laki
Golongan Darah : A+
Alamat : 03372 Longview Terrace
Agama : Kong Hu Chu
Status Perkawinan : Belum Menikah
Pekerjaan : Pharmacist
Kewarganegaraan : Czech Republic

5 | Input : test/Real/466_F_Right_index_finger.BMP

	<p> Fingerprint Based Individual Recognition Program <i>made by Kelompok Albertus</i></p> <p>Available Fingerprint</p>  <p>Select Algorithm</p> <p>KMP</p> <p>Submit</p> <p>Time Taken : 3397 ms Algorithm : KMP Similarity : 100%</p> <p>Upload Image</p> <p>Matching Fingerprint</p>  <p>Data</p> <table><tbody><tr><td>NIK</td><td>:</td><td>6565378517505586</td></tr><tr><td>Nama</td><td>:</td><td>Delmer Bouts</td></tr><tr><td>Tempat Lahir</td><td>:</td><td>Bayan</td></tr><tr><td>Tanggal Lahir</td><td>:</td><td>15/01/1982</td></tr><tr><td>Jenis Kelamin</td><td>:</td><td>Laki-Laki</td></tr><tr><td>Golongan Darah</td><td>:</td><td>AB-</td></tr><tr><td>Alamat</td><td>:</td><td>9 Saint Paul Alley</td></tr><tr><td>Agama</td><td>:</td><td>Kepercayaan lainnya</td></tr><tr><td>Status Perkawinan</td><td>:</td><td>Belum Menikah</td></tr><tr><td>Pekerjaan</td><td>:</td><td>Executive Secretary</td></tr><tr><td>Kewarganegaraan</td><td>:</td><td>Mongolia</td></tr></tbody></table>	NIK	:	6565378517505586	Nama	:	Delmer Bouts	Tempat Lahir	:	Bayan	Tanggal Lahir	:	15/01/1982	Jenis Kelamin	:	Laki-Laki	Golongan Darah	:	AB-	Alamat	:	9 Saint Paul Alley	Agama	:	Kepercayaan lainnya	Status Perkawinan	:	Belum Menikah	Pekerjaan	:	Executive Secretary	Kewarganegaraan	:	Mongolia
NIK	:	6565378517505586																																
Nama	:	Delmer Bouts																																
Tempat Lahir	:	Bayan																																
Tanggal Lahir	:	15/01/1982																																
Jenis Kelamin	:	Laki-Laki																																
Golongan Darah	:	AB-																																
Alamat	:	9 Saint Paul Alley																																
Agama	:	Kepercayaan lainnya																																
Status Perkawinan	:	Belum Menikah																																
Pekerjaan	:	Executive Secretary																																
Kewarganegaraan	:	Mongolia																																
Hasil																																		

	 <h2>Fingerprint Based Individual Recognition Program</h2> <p><i>made by Kelompok Albertus</i></p> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> <p>Available Fingerprint</p>  <p>Upload Image</p> </div> <div style="flex: 1;"> <p>Select Algorithm</p> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content;">BM</div> <p>Submit</p> </div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>Time Taken : 3702 ms Algorithm : BM Similarity : 100%</p> </div> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="flex: 1;"> <p>Matching Fingerprint</p>  </div> <div style="flex: 1;"> <p>Data</p> <table border="0"> <tbody> <tr><td>NIK</td><td>:</td><td>6565378517505586</td></tr> <tr><td>Nama</td><td>:</td><td>DeLmer Bouts</td></tr> <tr><td>Tempat Lahir</td><td>:</td><td>Bayan</td></tr> <tr><td>Tanggal Lahir</td><td>:</td><td>15/01/1982</td></tr> <tr><td>Jenis Kelamin</td><td>:</td><td>Laki-Laki</td></tr> <tr><td>Golongan Darah</td><td>:</td><td>AB-</td></tr> <tr><td>Alamat</td><td>:</td><td>9 Saint Paul Alley</td></tr> <tr><td>Agama</td><td>:</td><td>Kepercayaan lainnya</td></tr> <tr><td>Status Perkawinan</td><td>:</td><td>Belum Menikah</td></tr> <tr><td>Pekerjaan</td><td>:</td><td>Executive Secretary</td></tr> <tr><td>Kewarganegaraan</td><td>:</td><td>Mongolia</td></tr> </tbody> </table> </div> </div>	NIK	:	6565378517505586	Nama	:	DeLmer Bouts	Tempat Lahir	:	Bayan	Tanggal Lahir	:	15/01/1982	Jenis Kelamin	:	Laki-Laki	Golongan Darah	:	AB-	Alamat	:	9 Saint Paul Alley	Agama	:	Kepercayaan lainnya	Status Perkawinan	:	Belum Menikah	Pekerjaan	:	Executive Secretary	Kewarganegaraan	:	Mongolia
NIK	:	6565378517505586																																
Nama	:	DeLmer Bouts																																
Tempat Lahir	:	Bayan																																
Tanggal Lahir	:	15/01/1982																																
Jenis Kelamin	:	Laki-Laki																																
Golongan Darah	:	AB-																																
Alamat	:	9 Saint Paul Alley																																
Agama	:	Kepercayaan lainnya																																
Status Perkawinan	:	Belum Menikah																																
Pekerjaan	:	Executive Secretary																																
Kewarganegaraan	:	Mongolia																																
6	Input : test/11_M_Left_index_finger_Obl.BMP																																	

	 <h2>Fingerprint Based Individual Recognition Program</h2> <p><i>made by Kelompok Albertus</i></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Available Fingerprint</p>  <p>Upload Image</p> </div> <div style="width: 45%;"> <p>Select Algorithm</p> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content;">KMP</div> <p>Submit</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> Time Taken : 10020 ms Algorithm : Hamming Distance (not found using KMP) Similarity : 90,95% </div> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="width: 45%;"> <p>Matching Fingerprint</p>  </div> <div style="width: 45%;"> <p>Data</p> <table border="0"> <tbody> <tr><td>NIK</td><td>:</td><td>0202860407098953</td></tr> <tr><td>Nama</td><td>:</td><td>Emalia Bernadzki</td></tr> <tr><td>Tempat Lahir</td><td>:</td><td>Qiganjide</td></tr> <tr><td>Tanggal Lahir</td><td>:</td><td>15/04/1997</td></tr> <tr><td>Jenis Kelamin</td><td>:</td><td>Perempuan</td></tr> <tr><td>Golongan Darah</td><td>:</td><td>B+</td></tr> <tr><td>Alamat</td><td>:</td><td>5 Anthes Trail</td></tr> <tr><td>Agama</td><td>:</td><td>Kepercayaan lainnya</td></tr> <tr><td>Status Perkawinan</td><td>:</td><td>Cerai</td></tr> <tr><td>Pekerjaan</td><td>:</td><td>Operator</td></tr> <tr><td>Kewarganegaraan</td><td>:</td><td>China</td></tr> </tbody> </table> </div> </div>	NIK	:	0202860407098953	Nama	:	Emalia Bernadzki	Tempat Lahir	:	Qiganjide	Tanggal Lahir	:	15/04/1997	Jenis Kelamin	:	Perempuan	Golongan Darah	:	B+	Alamat	:	5 Anthes Trail	Agama	:	Kepercayaan lainnya	Status Perkawinan	:	Cerai	Pekerjaan	:	Operator	Kewarganegaraan	:	China
NIK	:	0202860407098953																																
Nama	:	Emalia Bernadzki																																
Tempat Lahir	:	Qiganjide																																
Tanggal Lahir	:	15/04/1997																																
Jenis Kelamin	:	Perempuan																																
Golongan Darah	:	B+																																
Alamat	:	5 Anthes Trail																																
Agama	:	Kepercayaan lainnya																																
Status Perkawinan	:	Cerai																																
Pekerjaan	:	Operator																																
Kewarganegaraan	:	China																																
Hasil	7 Input : test/547_M_Left_index_finger_Obl.BMP																																	

Hasil	<p>Fingerprint Based Individual Recognition Program</p> <p><i>made by Kelompok Albertus</i></p> <p>Available Fingerprint</p>  <p>Select Algorithm</p> <p>KMP</p> <p>Submit</p> <p>Time Taken : 10015 ms Algorithm : Hamming Distance (not found using KMP) Similarity : 91,47%</p> <p>Upload Image</p> <p>Matching Fingerprint</p>  <p>Data</p> <table border="0"> <tbody> <tr> <td>NIK</td> <td>:</td> <td>8405093654671440</td> </tr> <tr> <td>Nama</td> <td>:</td> <td>Briny Turton</td> </tr> <tr> <td>Tempat Lahir</td> <td>:</td> <td>Chisong</td> </tr> <tr> <td>Tanggal Lahir</td> <td>:</td> <td>24/03/2024</td> </tr> <tr> <td>Jenis Kelamin</td> <td>:</td> <td>Laki-Laki</td> </tr> <tr> <td>Golongan Darah</td> <td>:</td> <td>B-</td> </tr> <tr> <td>Alamat</td> <td>:</td> <td>2 Morning Avenue</td> </tr> <tr> <td>Agama</td> <td>:</td> <td>Katolik</td> </tr> <tr> <td>Status Perkawinan</td> <td>:</td> <td>Cerai</td> </tr> <tr> <td>Pekerjaan</td> <td>:</td> <td>Recruiting Manager</td> </tr> <tr> <td>Kewarganegaraan</td> <td>:</td> <td>China</td> </tr> </tbody> </table>	NIK	:	8405093654671440	Nama	:	Briny Turton	Tempat Lahir	:	Chisong	Tanggal Lahir	:	24/03/2024	Jenis Kelamin	:	Laki-Laki	Golongan Darah	:	B-	Alamat	:	2 Morning Avenue	Agama	:	Katolik	Status Perkawinan	:	Cerai	Pekerjaan	:	Recruiting Manager	Kewarganegaraan	:	China
NIK	:	8405093654671440																																
Nama	:	Briny Turton																																
Tempat Lahir	:	Chisong																																
Tanggal Lahir	:	24/03/2024																																
Jenis Kelamin	:	Laki-Laki																																
Golongan Darah	:	B-																																
Alamat	:	2 Morning Avenue																																
Agama	:	Katolik																																
Status Perkawinan	:	Cerai																																
Pekerjaan	:	Recruiting Manager																																
Kewarganegaraan	:	China																																
8	Input : test/505_M_Left_index_finger_Obl.BMP																																	

	<p style="text-align: center;">  Fingerprint Based Individual Recognition Program <i>made by Kelompok Albertus</i> </p> <div style="margin-top: 20px;"> <p>Available Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; text-align: center;">  </div> <p style="margin-top: 10px;">Upload Image</p> <p>Matching Fingerprint</p> <div style="border: 1px solid #ccc; padding: 5px; text-align: center;">  </div> </div> <div style="margin-top: 20px;"> <p>Select Algorithm</p> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content; margin-bottom: 5px;"> KMP </div> <p style="text-align: center;">Submit</p> </div> <div style="margin-top: 20px; border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>Time Taken : 14588 ms Algorithm : Hamming Distance (not found using KMP) Similarity : 80,11%</p> </div> <div style="display: flex; justify-content: space-between;"> <p>Matching Fingerprint</p> <p>Data</p> </div> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <table> <tbody> <tr><td>NIK</td><td>:</td><td>7591073217848241</td></tr> <tr><td>Nama</td><td>:</td><td>Edd Hrihorovich</td></tr> <tr><td>Tempat Lahir</td><td>:</td><td>Do-üba</td></tr> <tr><td>Tanggal Lahir</td><td>:</td><td>07/06/1975</td></tr> <tr><td>Jenis Kelamin</td><td>:</td><td>Perempuan</td></tr> <tr><td>Golongan Darah</td><td>:</td><td>AB-</td></tr> <tr><td>Alamat</td><td>:</td><td>08239 Manufacturers Terrace</td></tr> <tr><td>Agama</td><td>:</td><td>Kong Hu Chu</td></tr> <tr><td>Status Perkawinan</td><td>:</td><td>Belum Menikah</td></tr> <tr><td>Pekerjaan</td><td>:</td><td>Biostatistician III</td></tr> <tr><td>Kewarganegaraan</td><td>:</td><td>Pakistan</td></tr> </tbody> </table> </div>	NIK	:	7591073217848241	Nama	:	Edd Hrihorovich	Tempat Lahir	:	Do-üba	Tanggal Lahir	:	07/06/1975	Jenis Kelamin	:	Perempuan	Golongan Darah	:	AB-	Alamat	:	08239 Manufacturers Terrace	Agama	:	Kong Hu Chu	Status Perkawinan	:	Belum Menikah	Pekerjaan	:	Biostatistician III	Kewarganegaraan	:	Pakistan
NIK	:	7591073217848241																																
Nama	:	Edd Hrihorovich																																
Tempat Lahir	:	Do-üba																																
Tanggal Lahir	:	07/06/1975																																
Jenis Kelamin	:	Perempuan																																
Golongan Darah	:	AB-																																
Alamat	:	08239 Manufacturers Terrace																																
Agama	:	Kong Hu Chu																																
Status Perkawinan	:	Belum Menikah																																
Pekerjaan	:	Biostatistician III																																
Kewarganegaraan	:	Pakistan																																
Hasil	9 Input : test/433_M_Left_thumb_finger_Obl.BMP																																	

	 <h2>Fingerprint Based Individual Recognition Program</h2> <p><i>made by Kelompok Albertus</i></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Available Fingerprint</p>  <p>Upload Image</p> </div> <div style="width: 45%;"> <p>Select Algorithm</p> <div style="border: 1px solid #ccc; padding: 2px; width: fit-content;">BM</div> <p>Submit</p> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> Time Taken : 11963 ms Algorithm : Hamming Distance (not found using BM) Similarity : 84,81% </div> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <div style="width: 45%;"> <p>Matching Fingerprint</p>  </div> <div style="width: 45%;"> <p>Data</p> <table border="0"> <tbody> <tr><td>NIK</td><td>:</td><td>6021566594770114</td></tr> <tr><td>Nama</td><td>:</td><td>Hale Maud</td></tr> <tr><td>Tempat Lahir</td><td>:</td><td>Huurch</td></tr> <tr><td>Tanggal Lahir</td><td>:</td><td>06/02/1985</td></tr> <tr><td>Jenis Kelamin</td><td>:</td><td>Laki-Laki</td></tr> <tr><td>Golongan Darah</td><td>:</td><td>O+</td></tr> <tr><td>Alamat</td><td>:</td><td>5 Towne Place</td></tr> <tr><td>Agama</td><td>:</td><td>Kepercayaan lainnya</td></tr> <tr><td>Status Perkawinan</td><td>:</td><td>Menikah</td></tr> <tr><td>Pekerjaan</td><td>:</td><td>Mechanical Systems Engineer</td></tr> <tr><td>Kewarganegaraan</td><td>:</td><td>Mongolia</td></tr> </tbody> </table> </div> </div>		NIK	:	6021566594770114	Nama	:	Hale Maud	Tempat Lahir	:	Huurch	Tanggal Lahir	:	06/02/1985	Jenis Kelamin	:	Laki-Laki	Golongan Darah	:	O+	Alamat	:	5 Towne Place	Agama	:	Kepercayaan lainnya	Status Perkawinan	:	Menikah	Pekerjaan	:	Mechanical Systems Engineer	Kewarganegaraan	:	Mongolia
NIK	:	6021566594770114																																	
Nama	:	Hale Maud																																	
Tempat Lahir	:	Huurch																																	
Tanggal Lahir	:	06/02/1985																																	
Jenis Kelamin	:	Laki-Laki																																	
Golongan Darah	:	O+																																	
Alamat	:	5 Towne Place																																	
Agama	:	Kepercayaan lainnya																																	
Status Perkawinan	:	Menikah																																	
Pekerjaan	:	Mechanical Systems Engineer																																	
Kewarganegaraan	:	Mongolia																																	
Hasil	10 Input : test/Real/2_F_Left_little_finger.BMP																																		

	<p> Fingerprint Based Individual Recognition Program <i>made by Kelompok Albertus</i></p> <p>Available Fingerprint</p>  <p>Select Algorithm</p> <p>KMP</p> <p>Submit</p> <table border="1"><tr><td>Time Taken</td><td>:</td><td>1 ms</td></tr><tr><td>Algorithm</td><td>:</td><td>KMP</td></tr><tr><td>Similarity</td><td>:</td><td>100%</td></tr></table> <p>Upload Image</p> <p>Matching Fingerprint</p>  <p>Data</p> <table border="1"><tr><td>NIK</td><td>:</td><td>0015644960623967</td></tr><tr><td>Nama</td><td>:</td><td>Nikolaos Kristoffersson</td></tr><tr><td>Tempat Lahir</td><td>:</td><td>Katav-Ivanovsk</td></tr><tr><td>Tanggal Lahir</td><td>:</td><td>03/10/2007</td></tr><tr><td>Jenis Kelamin</td><td>:</td><td>Laki-Laki</td></tr><tr><td>Golongan Darah</td><td>:</td><td>O-</td></tr><tr><td>Alamat</td><td>:</td><td>14793 Sherman Plaza</td></tr><tr><td>Agama</td><td>:</td><td>Islam</td></tr><tr><td>Status Perkawinan</td><td>:</td><td>Menikah</td></tr><tr><td>Pekerjaan</td><td>:</td><td>Web Developer III</td></tr><tr><td>Kewarganegaraan</td><td>:</td><td>Russia</td></tr></table>	Time Taken	:	1 ms	Algorithm	:	KMP	Similarity	:	100%	NIK	:	0015644960623967	Nama	:	Nikolaos Kristoffersson	Tempat Lahir	:	Katav-Ivanovsk	Tanggal Lahir	:	03/10/2007	Jenis Kelamin	:	Laki-Laki	Golongan Darah	:	O-	Alamat	:	14793 Sherman Plaza	Agama	:	Islam	Status Perkawinan	:	Menikah	Pekerjaan	:	Web Developer III	Kewarganegaraan	:	Russia
Time Taken	:	1 ms																																									
Algorithm	:	KMP																																									
Similarity	:	100%																																									
NIK	:	0015644960623967																																									
Nama	:	Nikolaos Kristoffersson																																									
Tempat Lahir	:	Katav-Ivanovsk																																									
Tanggal Lahir	:	03/10/2007																																									
Jenis Kelamin	:	Laki-Laki																																									
Golongan Darah	:	O-																																									
Alamat	:	14793 Sherman Plaza																																									
Agama	:	Islam																																									
Status Perkawinan	:	Menikah																																									
Pekerjaan	:	Web Developer III																																									
Kewarganegaraan	:	Russia																																									
Hasil																																											

	<p> Fingerprint Based Individual Recognition Program <i>made by Kelompok Albertus</i></p> <p>Available Fingerprint</p>  <p>Select Algorithm</p> <p>BM</p> <p>Submit</p> <p>Time Taken : 2 ms Algorithm : BM Similarity : 100%</p> <p>Upload Image</p> <p>Matching Fingerprint</p>  <p>Data</p> <table><tbody><tr><td>NIK</td><td>: 0015644960623967</td></tr><tr><td>Nama</td><td>: Nikolaos Kristoffersson</td></tr><tr><td>Tempat Lahir</td><td>: Katav-Ivanovsk</td></tr><tr><td>Tanggal Lahir</td><td>: 03/10/2007</td></tr><tr><td>Jenis Kelamin</td><td>: Laki-Laki</td></tr><tr><td>Golongan Darah</td><td>: O-</td></tr><tr><td>Alamat</td><td>: 14793 Sherman Plaza</td></tr><tr><td>Agama</td><td>: Islam</td></tr><tr><td>Status Perkawinan</td><td>: Menikah</td></tr><tr><td>Pekerjaan</td><td>: Web Developer III</td></tr><tr><td>Kewarganegaraan</td><td>: Russia</td></tr></tbody></table>	NIK	: 0015644960623967	Nama	: Nikolaos Kristoffersson	Tempat Lahir	: Katav-Ivanovsk	Tanggal Lahir	: 03/10/2007	Jenis Kelamin	: Laki-Laki	Golongan Darah	: O-	Alamat	: 14793 Sherman Plaza	Agama	: Islam	Status Perkawinan	: Menikah	Pekerjaan	: Web Developer III	Kewarganegaraan	: Russia	
NIK	: 0015644960623967																							
Nama	: Nikolaos Kristoffersson																							
Tempat Lahir	: Katav-Ivanovsk																							
Tanggal Lahir	: 03/10/2007																							
Jenis Kelamin	: Laki-Laki																							
Golongan Darah	: O-																							
Alamat	: 14793 Sherman Plaza																							
Agama	: Islam																							
Status Perkawinan	: Menikah																							
Pekerjaan	: Web Developer III																							
Kewarganegaraan	: Russia																							
11	Input : test/ipin gosong.bmp																							

11 Input : test/ipin gosong.bmp

Hasil	<p align="center">Fingerprint Based Individual Recognition Program</p> <p><i>made by Kelompok Albertus</i></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Available Fingerprint</p>  <p>Upload Image</p> </div> <div style="width: 45%;"> <p>Select Algorithm</p> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">KMP</div> <p>Submit</p> <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc; margin-top: 10px;"> Time Taken : 21126 ms Algorithm : Not Found! Similarity : Not Found! </div> </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Matching Fingerprint</p>  </div> <div style="width: 45%;"> <p>Data</p> <table border="0" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 30%;">NIK</td><td>:</td><td>No Match</td></tr> <tr><td>Nama</td><td>:</td><td>No Match</td></tr> <tr><td>Tempat Lahir</td><td>:</td><td>No Match</td></tr> <tr><td>Tanggal Lahir</td><td>:</td><td>No Match</td></tr> <tr><td>Jenis Kelamin</td><td>:</td><td>No Match</td></tr> <tr><td>Golongan Darah</td><td>:</td><td>No Match</td></tr> <tr><td>Alamat</td><td>:</td><td>No Match</td></tr> <tr><td>Agama</td><td>:</td><td>No Match</td></tr> <tr><td>Status Perkawinan</td><td>:</td><td>No Match</td></tr> <tr><td>Pekerjaan</td><td>:</td><td>No Match</td></tr> <tr><td>Kewarganegaraan</td><td>:</td><td>No Match</td></tr> </table> </div> </div>	NIK	:	No Match	Nama	:	No Match	Tempat Lahir	:	No Match	Tanggal Lahir	:	No Match	Jenis Kelamin	:	No Match	Golongan Darah	:	No Match	Alamat	:	No Match	Agama	:	No Match	Status Perkawinan	:	No Match	Pekerjaan	:	No Match	Kewarganegaraan	:	No Match
NIK	:	No Match																																
Nama	:	No Match																																
Tempat Lahir	:	No Match																																
Tanggal Lahir	:	No Match																																
Jenis Kelamin	:	No Match																																
Golongan Darah	:	No Match																																
Alamat	:	No Match																																
Agama	:	No Match																																
Status Perkawinan	:	No Match																																
Pekerjaan	:	No Match																																
Kewarganegaraan	:	No Match																																

4.4. Analisis Hasil Pengujian

Berikut merupakan hasil pengujian aplikasi berdasarkan waktu eksekusi pada bagian 4.3.

Tabel 4.4.1. Hasil Pengujian Aplikasi

Test Case	KMP		BM		Hamming Distance	
	Waktu (ms)	Kemiripan (%)	Waktu (ms)	Kemiripan (%)	Waktu (ms)	Kemiripan (%)
1	964	100	992	100	-	
2	2618	100	2541	100	-	
3	4218	100	4086	100	-	
4	2299	100	2316	100	-	
5	3397	100	3702	100	-	
6	Not Found		Not Found		10020	90,95
7	Not Found		Not Found		10015	91,47
8	Not Found		Not Found		14588	80,11
9	Not Found		Not Found		11963	84,81

10	1	100 (<i>false</i>)	2	100 (<i>false</i>)	-	-
----	---	----------------------	---	----------------------	---	---

Berdasarkan hasil pengujian tersebut, algoritma *KMP* dan *BM* hanya dapat mencocokkan dua sidik jari yang sama persis (100% sama) atau kebetulan memiliki subpixel berukuran 60 yang sama, tetapi kenyataannya berbeda. Jika dianalisis, penyebabnya adalah sebagai berikut: Misalkan terdapat sebuah gambar yang memiliki subpixel dalam ASCII “...L81@[53 karakter lain]hYt...” (misalkan dalam posisi 20-80) dan gambar kedua yang memiliki subpixel yang sama dalam posisi tengah, gambar masukan adalah gambar kedua dan gambar pada basis data adalah kedua gambar. Basis data yang telah dibuat telah mengurutkan secara leksikografis *path* menuju gambar sehingga apabila gambar pertama dicocokkan pertama, program akan menerima bahwa gambar tersebut *exact match*. Untuk mengurangi kesalahan seperti ini, sebenarnya ukuran pixel yang diambil dari gambar masukan dapat diperbanyak, tetapi waktu eksekusi algoritma menjadi terlalu lama. Oleh karena itu, kami memilih untuk tetap menggunakan 60 pixel.

Selain itu, *Hamming Distance* terbukti dapat menjadi alternatif yang baik untuk mencocokkan dua sidik jari yang sama, dengan salah satu sidik jari diubah sedikit. Dari empat percobaan yang ditampilkan pada laporan ini (dan beberapa yang tidak ditampilkan), *hamming distance* selalu memberikan hasil yang akurat dengan persentase di atas 70%.

Bab V

Kesimpulan

5.1. Kesimpulan

Algoritma *KMP* dan *BM* dapat digunakan untuk mencocokkan sidik jari dua individu dengan tingkat kemiripan 100%. Jika tidak ditemukan sidik jari individu dengan tingkat kemiripan 100%, pencocokan dapat dilakukan dengan memanfaatkan *Hamming Distance*. *Regular Expression* dapat digunakan untuk mencocokkan nama asli individu dengan nama yang telah dienkripsi dengan Bahasa Alay.

5.2. Saran

Aplikasi *desktop* ini diharapkan dapat menjadi salah satu alternatif pencocokan sidik jari antar individu. Untuk kemudahan, sebaiknya aplikasi di-*deploy* menjadi aplikasi yang dapat berdiri sendiri tanpa harus bergantung pada *IDE* dan *DBMS* lokal.

Lampiran

Link repository GitHub:

https://github.com/nicholasrs05/Tubes3_Albertus

Daftar Pustaka

Munir, Rinaldi. 2024. “Pencocokan String (*String/Pattern Matching*)” di <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf> (diakses 23 Mei 2024).

Munir, Rinaldi. 2024. “String Matching dengan Regular Expression” di <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/String-Matching-dengan-Regex-2019.pdf> (diakses 23 Mei 2024).