

LAPORAN TUGAS BESAR 3

IF2211 - STRATEGI ALGORITMA

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



Disusun oleh :

Raffael Boymian Siahaan	13522046
Nabila Shikoofa Muida	13522069
Novelya Putri Ramadhani	13522096

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

2024

DAFTAR ISI

DESKRIPSI TUGAS.....	3
LANDASAN TEORI.....	5
2.1 Pattern Matching.....	5
2.1.1 Algoritma Knuth-Morris-Pratt.....	5
2.1.2 Algoritma Boyer-Moore.....	6
2.1.3 Regular Expression (Regex).....	7
2.2 Penjelasan Aplikasi Desktop yang dibangun.....	8
ANALISIS PEMECAHAN MASALAH.....	9
3.1 Langkah-Langkah Pemecahan Masalah.....	9
3.2 Proses Penyelesaian Solusi dengan Algoritma KMP dan BMP.....	9
3.3 Fitur Fungsional dan Arsitektur Aplikasi Desktop yang Dibangun.....	12
IMPLEMENTASI DAN PENGUJIAN.....	14
4.1 Spesifikasi Teknis Program.....	14
4.2 Tata Cara Penggunaan Program (interface program, fitur-fitur yang disediakan program, dan sebagainya).....	15
4.3 Hasil Pengujian.....	16
4.4 Analisis hasil pengujian.....	16
KESIMPULAN.....	20
5.1 Kesimpulan.....	20
5.2 Saran.....	20
5.3 Komentar.....	20
5.4 Refleksi.....	20
LAMPIRAN.....	21
DAFTAR PUSTAKA.....	21

BAB 1

DESKRIPSI TUGAS



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, kami 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 2

LANDASAN TEORI

2.1 Pattern Matching

Pada tugas besar ini, kami 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, serta *regular expression* sesuai dengan yang diajarkan pada materi dan salindia kuliah.

Pattern matching, atau pencocokan pola, adalah proses mencari keberadaan suatu string spesifik (pola) di dalam string lain yang lebih panjang (teks). Dalam konteks ini, teks diwakili oleh simbol T yang merupakan string panjang dengan karakter sebanyak n , sedangkan pola yang dicari diwakili oleh simbol P dan merupakan string dengan panjang m karakter, dengan asumsi bahwa m jauh lebih kecil daripada n . Tujuan utama dari *pattern matching* adalah untuk menemukan lokasi pertama di dalam teks T dimana pola P muncul.

Untuk mengilustrasikan definisi tersebut, misalkan teks T adalah "the rain in spain stays mainly on the plain" dan pola P adalah "main". Dalam kasus ini, *pattern matching* berhasil ketika menemukan substring "main" dalam kata "mainly" yang merupakan bagian dari teks T . Lokasi substring ini menjadi titik dimana pola P pertama kali muncul dalam teks T .

Selain itu, terdapat beberapa konsep dasar terkait string yang sering digunakan dalam studi *pattern matching*. Misalnya, konsep prefix dan suffix dalam sebuah string. Prefix dari sebuah string S adalah substring yang dimulai dari karakter pertama hingga karakter k , sedangkan suffix adalah substring yang dimulai dari karakter k hingga akhir string. Misalnya, untuk string S "andrew", semua kemungkinan prefix adalah "a", "an", "and", "andr", "andre", dan "andrew", sedangkan semua kemungkinan suffix adalah "w", "ew", "rew", "drew", "ndrew", dan "andrew".

2.1.1 Algoritma Knuth-Morris-Pratt

Algoritma Knuth-Morris-Pratt (KMP) merupakan sebuah teknik dalam pencocokan string yang mengatasi kelemahan algoritma brute force dalam hal efisiensi waktu. KMP mencari pola dalam teks dari kiri ke kanan serupa dengan pendekatan brute force tetapi

melakukan ini dengan lebih cerdas. Ketika ketidakcocokan (mismatch) terjadi, KMP memindahkan pola secara efisien berdasarkan informasi yang sudah dikumpulkan sebelumnya sehingga menghindari perbandingan yang tidak perlu.

KMP menggunakan apa yang disebut sebagai "*border function*" untuk menentukan seberapa jauh pola harus digeser ketika ketidakcocokan terjadi. *Border function* menghitung dan menyimpan ukuran dari prefix terpanjang pada pola yang juga merupakan suffix. Nilai ini disimpan dalam sebuah tabel yang digunakan untuk memutuskan seberapa jauh pola harus digeser ketika ketidakcocokan terjadi. Dengan kata lain, algoritma ini memanfaatkan informasi dari perbandingan yang telah dilakukan untuk meminimalkan jumlah perbandingan di masa mendatang.

Sebagai contoh, jika pola "ababa" memiliki ketidakcocokan setelah karakter kelima, KMP akan memeriksa tabel border dan melihat bahwa dua karakter terakhir dari string (yaitu, "ba") merupakan awalan yang juga merupakan akhiran. Oleh karena itu, pola digeser sehingga "ba" di awal pola sejajar dengan "ba" yang telah cocok di teks, sehingga perbandingan bisa dilanjutkan dari titik ini.

Dengan menggunakan *border function* ini, KMP mengurangi jumlah perbandingan yang perlu dilakukan dibandingkan dengan pencocokan pola brute force, membuatnya sangat efisien terutama untuk pencarian string yang panjang atau ketika pola yang dicari relatif panjang dan kompleks.

2.1.2 Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan metode efisien untuk pencocokan pola dalam teks, terutama berkat dua teknik utama yang digunakannya: teknik "looking-glass" dan "character-jump".

Teknik *looking-glass* memulai pencarian dari akhir pola, bukan dari awal. Dengan memulai dari akhir, algoritma ini dapat mengidentifikasi ketidakcocokan lebih awal dan kemudian menggunakan informasi ini untuk melompat lebih jauh dalam teks tanpa membandingkan setiap karakter.

Ketika terjadi ketidakcocokan antara karakter di teks dan karakter di pola, algoritma ini menentukan seberapa jauh pola harus digeser berdasarkan posisi terakhir kemunculan

karakter yang tidak cocok ini dalam pola. Jika karakter tidak ditemukan dalam pola, pola dapat digeser melewati posisi karakter tersebut dalam teks.

Kasus pemindahan pola yang mungkin terjadi adalah :

1. **Kasus Pertama:** Jika karakter yang tidak cocok di teks ditemukan di suatu tempat dalam pola, pola digeser sehingga kejadian terakhir karakter ini dalam pola diselaraskan dengan karakter di teks.
2. **Kasus Kedua:** Jika karakter di teks tidak ada dalam pola, pola digeser seluruhnya melewati karakter ini.
3. **Kasus Ketiga:** Jika karakter yang tidak cocok adalah unik dalam pola, maka pola digeser sejauh mungkin tanpa melewati kemungkinan kecocokan selanjutnya.

Sebagai bagian dari persiapannya, algoritma Boyer-Moore membangun sebuah tabel yang dikenal sebagai fungsi 'last occurrence' untuk setiap karakter dalam alfabet yang digunakan. Tabel ini mencatat posisi terakhir setiap karakter dalam pola, yang memungkinkan algoritma untuk melompat lebih jauh dalam teks selama fase pencocokan.

Boyer-Moore dikenal sangat efisien terutama ketika alfabet karakter dalam teks yang besar dan pola yang panjang, menjadikannya lebih cepat dibandingkan dengan metode brute force. Walaupun dalam kasus terburuknya, waktu yang dibutuhkan dapat sebanding dengan metode pencarian tradisional, dalam praktiknya sering kali jauh lebih cepat karena keefektifan dalam melompati bagian teks yang tidak perlu dibandingkan secara detail.

2.1.3 Regular Expression (Regex)

Regular Expression (Regex) adalah sebuah metode dalam bidang pemrograman dan pengolahan data untuk melakukan pencarian, penggantian, atau manipulasi teks berdasarkan pattern atau aturan terdefinisi. Regex memungkinkan pengguna untuk mengidentifikasi string yang kompleks dalam sebuah teks besar dengan cepat dan efisien. Adapun tujuan utama penggunaan Regex dalam kode yang kami buat adalah untuk mengidentifikasi dan memodifikasi elemen-elemen bahasa alay ini agar dapat dikonversi ke format yang lebih standar atau untuk keperluan analisis lebih lanjut.

Regex dapat digunakan untuk mengidentifikasi pola di mana huruf besar dan kecil dicampur secara acak dalam kata. Dalam bahasa alay, angka sering digunakan untuk

menggantikan huruf yang mirip dalam penampilan (misalnya, '4' digunakan untuk 'A'). Regex bisa digunakan untuk mengidentifikasi pola ini dan menggantinya dengan huruf yang sesuai. Regex juga dapat membantu dalam mengidentifikasi dan mengembalikan penyingkatan ke bentuk penuhnya jika aturan penyingkatan diketahui. Kita dapat menggunakan Regex untuk menulis sebuah script yang mengintegrasikan semua aturan ini dalam satu fungsi pemrosesan teks, yang menangani kombinasi huruf besar-kecil, penggunaan angka, dan penyingkatan dalam satu langkah.

2.2 Penjelasan Aplikasi Desktop yang dibangun

Aplikasi desktop yang kami kembangkan dirancang khusus untuk keperluan identifikasi sidik jari, memanfaatkan teknologi pemrosesan citra dan pencocokan pola yang canggih. Antarmuka pengguna aplikasi ini terdiri dari beberapa komponen utama yang dirancang untuk memaksimalkan efisiensi dan keakuratan dalam pengenalan sidik jari. Aplikasi identifikasi sidik jari yang kami kembangkan dibangun menggunakan bahasa pemrograman C# dengan memanfaatkan .NET Framework.

Aplikasi ini dilengkapi dengan pilihan algoritma pencocokan sidik jari yang dapat dipilih sesuai dengan kebutuhan pengguna:

- Boyer-Moore (BM): Algoritma ini terkenal dengan efisiensinya dalam pencocokan string, yang telah kami adaptasi untuk pencocokan pola sidik jari.
- Knuth-Morris-Pratt (KMP): Algoritma ini juga telah kami adaptasi untuk meningkatkan efisiensi pencocokan sidik jari, terutama dalam mengidentifikasi pengulangan pola dalam sidik jari.

BAB 3

ANALISIS PEMECAHAN MASALAH

3.1 Langkah-Langkah Pemecahan Masalah

Sebelum memulai penerapan algoritma Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM) dalam aplikasi identifikasi sidik jari, ada beberapa langkah penting yang harus dilakukan untuk memastikan bahwa kedua algoritma dapat diimplementasikan secara efektif.

Berikut adalah langkah-langkah awal yang diperlukan:

- 1) Mengumpulkan data sidik jari yang cukup dari berbagai individu untuk membentuk database yang komprehensif dan representatif. Data ini harus mencakup berbagai jenis sidik jari dengan variasi dalam pola, ukuran, dan orientasi.
- 2) Mengubah fitur-fitur yang telah diekstrak menjadi string atau array data yang dapat diolah oleh algoritma pencocokan.
- 3) Membuat struktur database untuk menyimpan sidik jari yang telah diproses ini dengan cara yang memungkinkan pencarian cepat dan efisien.
- 4) Mengkodekan algoritma KMP dan BM dalam bahasa pemrograman yang dipilih, mengadaptasi mereka untuk bekerja dengan data sidik jari.
- 5) Melakukan *unit testing* untuk memverifikasi bahwa setiap komponen algoritma bekerja sesuai dengan spesifikasi dan dapat secara akurat menemukan cocokan dalam data sidik jari.

3.2 Proses Penyelesaian Solusi dengan Algoritma KMP dan BMP

Proses kerja algoritma Knuth-Morris-Pratt (KMP) dalam konteks pencocokan string terdiri dari dua bagian utama: pembuatan tabel border (juga dikenal sebagai tabel lompatan atau tabel kegagalan) dan proses pencocokan string itu sendiri.

Pembuatan tabel border adalah langkah pertama dan krusial dalam algoritma KMP, yang bertujuan untuk meminimalkan jumlah pengulangan yang tidak perlu ketika pencocokan gagal. Proses ini melibatkan analisis pattern untuk menentukan seberapa jauh algoritma dapat melompati pattern tanpa kehilangan potensi kecocokan yang valid. Pada pembuatan tabel border menggunakan BorderFunctionTable, sistem menyiapkan sebuah

array di mana setiap index i memiliki nilai maksimal panjang dari substring yang adalah prefix dan suffix. Fungsi ini memanggil `BorderFunction` untuk setiap index untuk mendapatkan panjang prefix yang cocok yang juga suffix. Dalam fungsi `KMPMatch`, algoritma iteratif melintasi text dan mencoba mencocokkan pattern menggunakan informasi dari tabel border. Ketika terjadi ketidakcocokan, algoritma tidak kembali ke awal pattern, tetapi melanjutkan dari titik pencocokan terakhir yang memungkinkan, berdasarkan nilai di tabel border.

Setelah tabel border telah dibangun, algoritma Knuth-Morris-Pratt (KMP) siap untuk pencocokan string dengan inisialisasi dua indeks, i untuk text dan j untuk pattern, keduanya diatur ke 0. Proses dimulai dengan iterasi melalui text, berlanjut selama i masih kurang dari panjang text. Selama iterasi, `text[i]` dibandingkan dengan `pattern[j]`; jika terjadi kecocokan, kedua i dan j diinkremen. Jika j mencapai panjang pattern, ini menandakan bahwa kecocokan berhasil ditemukan, dengan lokasi kecocokan ditentukan oleh $i - j$. Jika terjadi mismatch dan j tidak nol, j diatur kembali ke `border[j - 1]`, menghindari pencocokan ulang dari awal dan memungkinkan pencocokan untuk melanjutkan dari kecocokan terakhir yang mungkin valid. Jika j adalah 0, yang berarti tidak ada kecocokan sebelumnya, pencocokan hanya melanjutkan dengan menginkremen i . Proses ini berulang sampai seluruh text telah dilalui atau sebuah kecocokan ditemukan.

Algoritma berhenti ketika mencapai akhir text atau menemukan kecocokan. Jika pencarian selesai tanpa menemukan kecocokan, hasilnya adalah negatif. Kompleksitas waktu dari implementasi ini adalah $O(n + m)$, dengan n adalah panjang pattern dan m adalah panjang text. Kompleksitas ini didasarkan pada dua proses:

- **$O(n)$** untuk pembuatan tabel border.
- **$O(m)$** untuk proses pencocokan di text.

Sedangkan proses kerja algoritma Boyer-Moore memanfaatkan 'last occurrence function' yang sangat meningkatkan efisiensi algoritma dengan mengurangi jumlah perbandingan yang diperlukan dalam skenario pencocokan gagal. Langkah-langkah dalam pembuatan last occurrence function adalah sebagai berikut.

Pertama, buatlah tabel `lastOccurrence` yang besar cukup untuk menyimpan indeks kemunculan terakhir untuk setiap karakter yang mungkin muncul di teks. Tabel ini diindeks berdasarkan nilai ASCII karakter, dan setiap nilai diinisialisasi dengan -1, menandakan bahwa karakter tersebut belum muncul di pattern. Kemudian dilakukan iterasi melalui pattern dari awal hingga akhir. Untuk setiap karakter di pattern, perbarui tabel `lastOccurrence` di indeks yang sesuai dengan nilai ASCII karakter tersebut dengan posisi terbaru karakter tersebut di pattern. Ini menandakan posisi terakhir setiap karakter muncul dalam pattern.

Proses pencocokan dalam algoritma Boyer-Moore dimulai dari akhir pattern dan mengikuti langkah-langkah ini:

- 1) Menetapkan indeks i untuk teks dan j untuk pattern, dengan keduanya diatur pada posisi terakhir dari pattern.
- 2) Selama i lebih kecil dari panjang teks, lakukan perbandingan karakter:
 - Jika karakter pada `text[i]` cocok dengan `pattern[j]`, dekremen j (bergerak mundur melalui pattern) dan i (bergerak mundur melalui teks). Jika j mencapai 0, kecocokan berhasil ditemukan, dengan lokasi kecocokan ditentukan oleh $i + 1$.
 - Jika tidak cocok, tabel last occurrence digunakan untuk menentukan seberapa jauh lompatan yang harus dilakukan: jika karakter yang tidak cocok ditemukan dalam pattern, lompatan ditentukan berdasarkan posisi terakhir karakter tersebut dalam pattern; jika karakter tersebut terletak sebelum posisi j di pattern, maka pattern digeser sehingga karakter ini sejajar dengan karakter saat ini di teks, memungkinkan pencocokan untuk melanjutkan dari posisi ini.
- 3) Teruskan dengan menyesuaikan posisi i dan j berdasarkan hasil perbandingan dan informasi dari tabel `lastOccurrence`.

Kompleksitas waktu algoritma Boyer-Moore bervariasi tergantung pada struktur data dan inputnya, tetapi umumnya lebih efisien daripada pencarian brute-force, terutama untuk teks yang panjang dengan karakter yang sering berulang. Dalam kasus terbaik, kompleksitasnya bisa sangat efisien, mendekati $O(n/m)$, di mana n adalah panjang teks dan m adalah panjang pattern.

3.3 Fitur Fungsional dan Arsitektur Aplikasi Desktop yang Dibangun



Fitur fungsional dari aplikasi desktop yang kami kembangkan berfokus pada kemampuan interaktif dan proses otomatis untuk memudahkan pengguna dalam mengelola dan menganalisis sidik jari:

a) Pengunggahan Citra Sidik Jari

Pengguna dapat dengan mudah mengunggah gambar sidik jari melalui antarmuka yang user-friendly. Sistem mendukung berbagai format gambar dan memproses gambar untuk memastikan kualitasnya memenuhi standar analisis.

b) Tombol Pencarian Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM)

Untuk memulai proses pencocokan, pengguna dapat memilih antara algoritma KMP atau BM melalui tombol pencarian yang tersedia. Pengguna dapat menyesuaikan pencocokan sidik jari sesuai dengan kebutuhan spesifik mereka, tergantung pada sifat data atau kebutuhan kecepatan pencocokan.

c) Pencocokan Sidik Jari Otomatis

Setelah sidik jari diunggah, aplikasi menggunakan algoritma KMP dan BM untuk mencari kecocokan dalam database. Proses ini otomatis dan dilakukan dengan cepat, menyediakan hasil dalam hitungan detik tergantung pada ukuran database.

Pada area tampilan sidik jari, bagian kiri adalah area di mana sidik jari yang baru diunggah oleh pengguna ditampilkan. Sedangkan bagian kanan adalah area yang menampilkan sidik jari dari database yang sesuai dengan sidik jari yang telah diunggah, setelah proses pencocokan dilakukan menggunakan algoritma pencocokan yang terintegrasi dalam sistem.

Di sisi kanan antarmuka, terdapat sebuah panel yang menampilkan hasil pencarian yang mencakup "List Biodata". Informasi yang ditampilkan dalam list ini mencakup nama, tempat lahir, tanggal lahir, jenis kelamin, golongan darah, alamat, agama, status perkawinan, pekerjaan, dan kewarganegaraan dari individu yang sidik jarinya cocok. Panel ini juga menampilkan "Waktu Pencarian", yang mencatat durasi yang diperlukan untuk menyelesaikan pencarian dan pencocokan sidik jari, serta "Persentase Kecocokan", yang memberikan persentase kecocokan antara sidik jari yang diinput dengan sidik jari yang ada di database, sehingga memberikan gambaran mengenai keakuratan pencocokan.

BAB 4

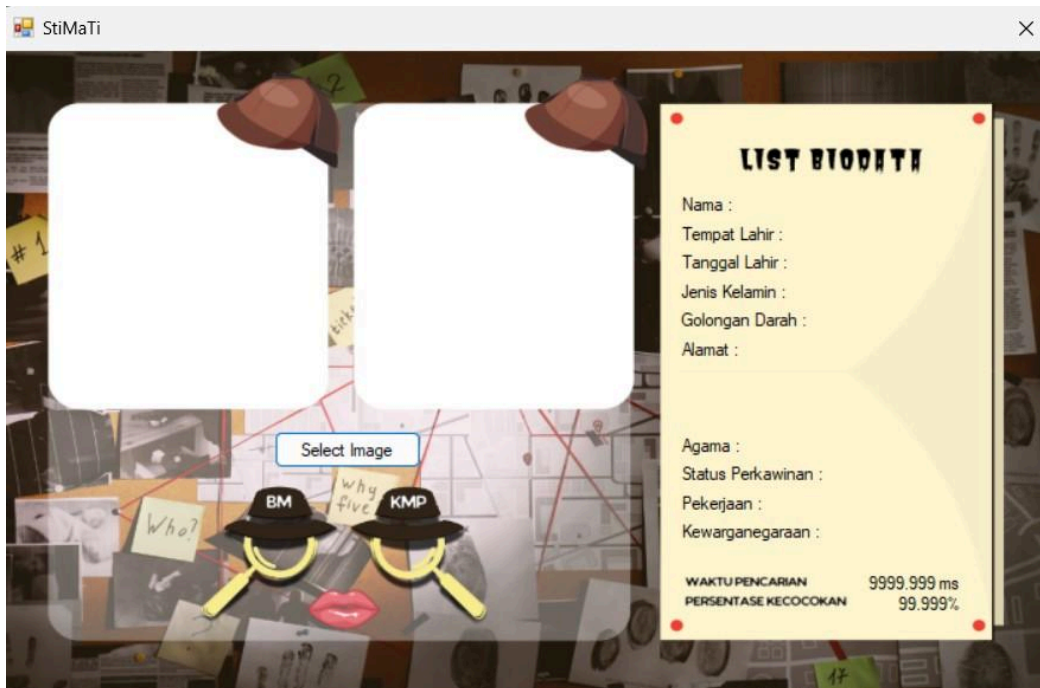
IMPLEMENTASI DAN PENGUJIAN

4.1 Spesifikasi Teknis Program

Program ini terstruktur dalam beberapa direktori utama: Assets, Doc, Src, dan Test. Direktori utama, Src, berisi tiga subdirektori: algo, database, dan frontend, di mana kode sumber utama program berada. Semua gambar yang digunakan sebagai studi kasus dalam pengujian program tersimpan dalam direktori Test. Dokumentasi terkait, termasuk laporan, ditempatkan dalam direktori Doc. Di dalam direktori algo, terdapat beberapa file, termasuk BiodataProcessor.cs, BM.cs, FingerprintProcessor.cs, KMP.cs, dan Levenshtein.cs, yang masing-masing menyimpan skrip yang berhubungan dengan komponen algoritmik dari sistem. Semua hal yang berhubungan dengan database termasuk sqlnya tersimpan di dalam folder database, begitu pula semua hal yang berhubungan dengan frontend tersimpan dalam folder Frontend.

Di dalam BiodataProcessor.cs, kami memproses dan mengambil biodata berdasarkan nama dari daftar biodata. Fungsi ini terintegrasi dengan beberapa algoritma pencocokan untuk memverifikasi kecocokan nama, mencerminkan penerapan strategi yang solid dalam identifikasi dan verifikasi data. GetBiodata(string name, List<Biodata> biodatas) bertujuan untuk mengambil objek Biodata dari sebuah daftar berdasarkan nama yang diberikan. IsAMatch(string name, List<string>? names) mengecek apakah nama yang diberikan cocok dengan salah satu dari daftar nama yang mungkin.

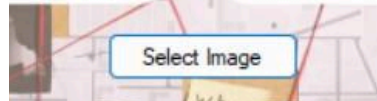
4.2 Tata Cara Penggunaan Program (interface program, fitur-fitur yang disediakan program, dan sebagainya).



1. Pengguna memulai dengan mengunggah citra sidik jari yang ingin dicari biodatanya.
2. Setelah citra diunggah, pengguna memiliki pilihan untuk memilih algoritma yang akan digunakan untuk proses pencocokan. Dua opsi yang tersedia adalah:
 - KMP (Knuth-Morris-Pratt)
 - BM (Boyer-Moore)
3. Program akan menampilkan list biodata jika ditemukan citra sidik jari yang memiliki kemiripan minimal 70%. Biodata ini bisa mencakup nama, alamat, status, dan informasi lain yang tersimpan di database. Program juga dapat mengeluarkan informasi tidak ada sidik jari yang mirip jika semua citra dalam basis data tidak memiliki kemiripan dengan masukan.
4. Program juga menyediakan informasi tentang waktu eksekusi dari pencocokan, yang berguna untuk menganalisis efisiensi algoritma yang dipilih.
5. Persentase kemiripan juga ditampilkan, memberikan insight tentang seberapa dekat citra sidik jari masukan dengan citra dalam database.

Adapun fitur-fitur yang kami sediakan antara lain

- a) Tombol Unggah Gambar



b) Tombol Pencarian BM/KMP



Untuk memulai proses pencocokan, pengguna diberikan opsi untuk memilih antara algoritma KMP atau BM. Kami menggunakan simbol kaca pembesar dengan sebuah topi untuk menggambarkan masing-masing BM dan KMP.

c) List Biodata

4.3 Hasil Pengujian

4.4 Analisis hasil pengujian.

Test Case	Waktu Eksekusi (ms)		Persentase Kemiripan	
	KMP	BM	KMP	BM
1: Sidik Jari Sama 4_M_Left_index_finger.BMP	7404,4626	1776,514	100	100

2: Sidik Jari Sama 1_M_Left_index_finger.BMP	8975,4967	2821,8752	100	100
3: Sidik Jari Sama 6_M_Left_middle_finger.BMP	5686,1272	2108,0497	100	100
4: Sidik Jari Berbeda test.BMP	100283,467	54358,6589	0	0

Pemanfaatan pattern matching dalam membangun sistem deteksi individu berbasis biometrik melalui citra sidik jari dapat menjadi komponen krusial dalam teknologi keamanan dan forensik di masa yang akan datang. Sistem deteksi menggunakan teknik pattern matching seperti algoritma Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM) memberikan metode yang efisien dan akurat untuk mengidentifikasi individu berdasarkan sidik jari mereka. Proses ini dimulai dengan akuisisi citra sidik jari yang kemudian diolah dan diubah menjadi format yang dapat diinterpretasikan oleh sistem, seperti ASCII atau format gambar yang dipecah menjadi segmen-segmen.

Dalam penggunaannya, algoritma KMP memanfaatkan tabel border untuk mempercepat proses pencocokan dengan menghindari perbandingan yang tidak perlu, sedangkan BM menggunakan last occurrence function untuk mengoptimalkan lompatan dalam pencarian, memungkinkan peningkatan signifikan dalam efisiensi waktu pencarian terutama pada dataset besar. Kedua teknik ini memungkinkan sistem untuk mencari kemiripan sidik jari dengan cepat dan akurat, menyediakan feedback berupa kemiripan dalam persentase dan menampilkan biodata individu yang terkait jika ada kecocokan.

Penggunaan pattern matching dalam sistem deteksi berbasis biometrik juga mendukung keamanan yang lebih baik dengan meminimalisir kesalahan pengenalan dan memastikan bahwa hanya individu yang terverifikasi yang dapat mengakses area atau informasi yang sensitif. Seiring dengan kemajuan teknologi pemrosesan citra dan peningkatan algoritma pencocokan, pemanfaatan pattern matching diharapkan menjadi lebih inovatif, mengintegrasikan kecerdasan buatan dan pembelajaran mesin untuk lebih meningkatkan akurasi dan efisiensi sistem deteksi biometrik berbasis sidik jari.

Algoritma KMP memiliki kompleksitas waktu $O(n + m)$, di mana n adalah panjang teks dan m adalah panjang pattern. Hal ini membuat algoritma KMP sangat efisien untuk pencocokan string, karena ia hanya memerlukan waktu linear. KMP memiliki kompleksitas waktu $O(n + m)$, di mana n adalah panjang teks dan m adalah panjang pattern. Hal ini membuat algoritma KMP sangat efisien untuk pencocokan string, karena ia hanya memerlukan waktu linear. Namun, KMP bisa kurang efektif jika digunakan pada string dengan set karakter yang sangat besar, karena banyaknya kemungkinan lompatan yang tidak efisien. Pembuatan tabel border membutuhkan waktu tambahan di awal, meskipun ini hanya terjadi sekali dan kemudian dapat digunakan untuk semua pencocokan berikutnya. Sedangkan algoritma BM dikenal sebagai algoritma pencocokan string tercepat untuk penggunaan umum, terutama karena ia biasanya melakukan lebih sedikit perbandingan daripada KMP dalam praktiknya. Algoritma ini menggunakan dua heuristik inti, yaitu good suffix shift dan bad character shift, yang secara signifikan mengurangi jumlah perbandingan karakter dengan lompatan yang jauh lebih besar daripada yang mungkin di KMP.

Dalam analisis hasil pengujian sistem pencocokan sidik jari menggunakan algoritma Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM), kami telah mengamati beberapa parameter kinerja, yaitu waktu eksekusi dan persentase kemiripan, berdasarkan empat kasus tes yang berbeda. Berikut ini merupakan analisis mendetail dari hasil yang didapatkan.

Pada ketiga kasus dimana citra sidik jari yang diuji adalah sama, algoritma BM menunjukkan waktu eksekusi yang secara konsisten lebih cepat dibandingkan dengan KMP. Sehingga BM lebih efisien dalam menangani pencocokan sidik jari yang identik karena kecenderungan BM untuk membuat lompatan yang lebih besar dan mengurangi jumlah perbandingan karakter. Contohnya, pada kasus pertama, waktu eksekusi untuk BM adalah sekitar tiga kali lebih cepat daripada KMP (1776 ms vs. 4626 ms), menunjukkan efisiensi yang signifikan.

Ketika sidik jari yang diuji berbeda, ditemukan bahwa KMP memakan waktu yang jauh lebih lama dibandingkan dengan BM (54358 ms vs. 6589 ms). Hal ini juga menegaskan bahwa BM jauh lebih efektif dalam mengidentifikasi ketidakcocokan lebih

awal dan menghentikan proses lebih cepat daripada KMP, yang terus mencari hingga akhir teks meskipun tidak ada peluang untuk mencocokkan lebih lanjut.

Dari segi persentase kemiripan, baik KMP maupun BM berhasil mengidentifikasi kemiripan 100% untuk semua kasus di mana sidik jari adalah sama sehingga kedua algoritma tersebut sangat akurat dalam menemukan kecocokan sempurna dalam database sidik jari. Untuk kasus 4, baik KMP maupun BM mencatatkan persentase kemiripan sebesar 0% untuk sidik jari yang berbeda, yang sesuai dengan harapan karena citra yang diuji sama sekali tidak cocok dengan yang ada di database. Hal ini menunjukkan bahwa kedua algoritma tersebut efektif dalam mengeliminasi kecocokan palsu ketika tidak ada kemiripan yang sebenarnya.

Dari hasil pengujian ini, dapat disimpulkan bahwa algoritma Boyer-Moore secara umum lebih unggul dari Knuth-Morris-Pratt dalam hal efisiensi waktu, terutama dalam skenario dimana tidak ada kecocokan yang ditemukan. Kedua algoritma tersebut menunjukkan tingkat akurasi yang tinggi dalam mendeteksi kemiripan ketika sidik jari yang sama diuji. Namun, untuk penggunaan dalam sistem deteksi sidik jari di mana waktu eksekusi cukup krusial dan terdapat potensi untuk *mismatch*, Boyer-Moore merupakan pilihan yang lebih baik.

BAB 5

KESIMPULAN

5.1 Kesimpulan

Pemanfaatan teknik pattern matching dalam sistem deteksi individu berbasis biometrik melalui citra sidik jari, seperti dengan menggunakan algoritma Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM) memungkinkan pencocokan sidik jari yang cepat dan tepat dengan mengurangi jumlah perbandingan yang tidak perlu melalui penggunaan tabel border (KMP) dan fungsi last occurrence (BM), memaksimalkan efisiensi pencarian dalam database besar. Keberhasilan dalam mencocokkan sidik jari tidak hanya menghasilkan identifikasi yang akurat tetapi juga meningkatkan keamanan dengan memastikan bahwa hanya individu yang terverifikasi yang dapat mengakses area sensitif.

5.2 Saran

- Fokus pada optimasi kode dan algoritma untuk mengurangi waktu pemrosesan tanpa mengorbankan akurasi.
- Teruslah mengeksplorasi dan menguji coba kombinasi dari berbagai algoritma pencocokan string dan pattern matching untuk menemukan metode yang lebih efektif dan efisien.

5.3 Komentar

Tugas ini telah membuka jendela baru bagi kami untuk mengeksplorasi lebih dalam tentang materi-materi yang tidak sepenuhnya tercakup dalam perkuliahan. Terima kasih yang sebesar-besarnya kepada asisten lab, dosen pengajar, dan semua pihak yang telah mendukung kami sepanjang proses pembelajaran ini. Melalui tugas ini, kami tidak hanya memperoleh pengetahuan teknis tentang penerapan algoritma pencocokan seperti Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM) dalam konteks biometrik sidik jari, tetapi juga kesempatan untuk terus tumbuh dan berkembang bersama.

5.4 Refleksi

- Jaga kesehatan, jangan sampai dirawat 7 hari apalagi dimasa kritis menjelang *deadline* pengumpulan tubes ^^

LAMPIRAN

Tautan Repository GitHub : https://github.com/novelxv/Tubes3_StiMaTi

DAFTAR PUSTAKA

- [1] R. Munir, Pencocokan String - informatika.stei.itb.ac.id, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pencocokan-string-2021.pdf> (accessed Mei. 20, 2024).
- [2] R. Munir, String Matching dengan Regex - informatika.stei.itb.ac.id, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2022-2023/String-Matching-dengan-Regex-2019.pdf> (accessed Mei. 20, 2024).