

Laporan Tugas Besar IF2211 Strategi Algoritma

**Pemanfaatan Pattern Matching untuk Membangun Sistem ATS
(Applicant Tracking System) Berbasis CV Digital**



Disusun oleh:
info

Andri Nurdianto	(13523145)
Arlow Emmanuel Hergara	(13523161)
Fachriza Ahmad Setiyono	(13523162)

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2025**

Daftar Isi

Daftar Isi	2
Bab 1	4
Bab 2	5
2.1 Dasar Teori	5
2.1.1 String Matching	5
2.1.2 Algoritma Knuth-Moriss-Pratt	5
2.1.3 Algoritma Boyer-Moore	5
2.1.4 Algoritma Aho-Corasick	6
2.2 Penjelasan Aplikasi	7
2.2.1 Python	7
2.2.2 PyQt	7
2.2.3 PyPDF	7
2.2.4 Docker	7
2.2.5 MySQL	8
2.2.6 MySQL Connector Python	8
2.2.7 UV	8
Bab 3	9
3.1 Langkah-Langkah Pemecahan Masalah	9
3.2 Proses Pemetaan Masalah Menjadi Elemen-Elemen Algoritma KMP dan BM	10
3.3 Fitur Fungsional dan Arsitektur Aplikasi yang dibangun	10
3.4 Contoh Ilustrasi Kasus	10
Bab 4	12
4.1 Spesifikasi Teknis Program	12
4.1.1 Searcher	12
4.1.2 PDFReader	12
4.1.3 LevenshteinFuzzy	12
4.1.4 SectionScraper	13
4.1.5 StringMatcher	13
4.1.6 KnuthMorrisPratt	13
4.1.7 BoyerMoore	13
4.1.8 AhoCorasick	13
4.2 Tata Cara Penggunaan Program	14
4.3 Hasil pengujian	14
4.4 Analisis hasil pengujian	22
Bab 5	23
Kesimpulan	23
Saran	23
Refleksi	23
Lampiran	24
Pranala Repository Github	24

Pranala video Youtube	24
Tabel Ketercapaian	24
Daftar Pustaka	25

Bab 1

Deskripsi Persoalan

Di era digital ini, keamanan data dan akses menjadi semakin penting. Perkembangan proses rekrutmen tenaga kerja telah mengalami perubahan signifikan dengan memanfaatkan teknologi untuk meningkatkan efisiensi dan akurasi. Salah satu inovasi yang menjadi solusi utama adalah Applicant Tracking System (ATS), yang dirancang untuk mempermudah perusahaan dalam menyaring dan mencocokkan informasi kandidat dari berkas lamaran, khususnya Curriculum Vitae (CV). ATS memungkinkan perusahaan untuk mengelola ribuan dokumen lamaran secara otomatis dan memastikan kandidat yang relevan dapat ditemukan dengan cepat.

Meskipun demikian, salah satu tantangan besar dalam pengembangan sistem ATS adalah kemampuan untuk memproses dokumen CV dalam format PDF yang tidak selalu terstruktur. Dokumen seperti ini memerlukan metode canggih untuk mengekstrak informasi penting seperti identitas, pengalaman kerja, keahlian, dan riwayat pendidikan secara efisien. Pattern matching menjadi solusi ideal dalam menghadapi tantangan ini.

Pattern matching adalah teknik untuk menemukan dan mencocokkan pola tertentu dalam teks. Dalam konteks ini, algoritma Boyer-Moore dan Knuth-Morris-Pratt (KMP) sering digunakan karena keduanya menawarkan efisiensi tinggi untuk pencarian teks di dokumen besar. Algoritma ini memungkinkan sistem ATS untuk mengidentifikasi informasi penting dari CV pelamar dengan kecepatan dan akurasi yang optimal.

Di dalam Tugas Besar 3 ini, Anda diminta untuk mengimplementasikan sistem yang dapat melakukan deteksi informasi pelamar berbasis dokumen CV digital. Metode yang akan digunakan untuk melakukan deteksi pola dalam CV adalah Boyer-Moore dan Knuth-Morris-Pratt. Selain itu, sistem ini akan dihubungkan dengan identitas kandidat melalui basis data sehingga harapannya terbentuk sebuah sistem yang dapat mengenali profil pelamar secara lengkap hanya dengan menggunakan CV digital.

Bab 2

Landasan Teori

2.1 Dasar Teori

2.1.1 String Matching

String matching adalah proses untuk menemukan pola yang cocok pada suatu string dengan string yang lain. String matching dapat digunakan untuk membantu menemukan kata kunci tertentu pada sebuah dokumen, website, atau database. Ada 3 algoritma yang biasanya dipakai dalam string matching, yaitu brute force, knuth-morris-pratt, dan boyer moore.

2.1.2 Algoritma Knuth-Moriss-Pratt

Algoritma Knuth-Morris-Pratt (KMP) adalah algoritma pencocokan string yang digunakan untuk menemukan posisi kemunculan suatu pola dalam sebuah teks secara efisien dengan berjalan dari kiri ke kanan. Untuk meningkatkan KMP melakukan tahap pra-pemrosesan terhadap pola sebelum proses pencocokan dilakukan. Tahap ini menghasilkan sebuah struktur data bernama fungsi batas atau sering disebut prefix function (sering disimpan dalam array `lps[]` atau longest prefix suffix), yang mencerminkan kecocokan sebagian pola terhadap dirinya sendiri.

Fungsi batas ini menyimpan informasi tentang panjang prefiks terbesar dari $P[0..k]$ yang juga merupakan sufiks dari $P[0..k]$, dan bukan dari $P[1..k]$. Nilai ini digunakan untuk menentukan sejauh mana pola dapat digeser tanpa mengulang perbandingan karakter yang sudah terbukti cocok sebelumnya. Misalnya, jika terjadi ketidakcocokan pada posisi j dalam pola, maka pencocokan tidak perlu dimulai ulang dari indeks nol, melainkan dapat dilanjutkan dari posisi `lps[j-1]`. Di sini, $k = j - 1$ adalah posisi sebelum terjadinya mismatch.

2.1.3 Algoritma Boyer-Moore

Algoritma Boyer-Moore merupakan salah satu algoritma pencocokan pola efisien yang digunakan untuk mencari keberadaan suatu pola P dalam teks T . Keunggulan algoritma ini terletak pada cara pencocokannya yang tidak selalu memeriksa setiap karakter teks secara berurutan, melainkan melompati bagian-bagian teks yang dianggap tidak mungkin menjadi titik awal kemunculan pola. Algoritma ini didasarkan pada dua prinsip utama:

1. Teknik Looking-Glass:

Proses pencocokan dilakukan dari kanan ke kiri pada pola. Artinya, ketika mencocokkan P terhadap teks T , algoritma memulai perbandingan dari ujung kanan pola, bukan dari awal. Pendekatan ini memungkinkan algoritma untuk mendeteksi ketidakcocokan lebih awal pada karakter yang lebih penting dalam pencocokan.

2. Teknik Character-Jump:

Ketika terjadi ketidakcocokan antara karakter dalam teks $T[i]$ dan karakter dalam pola $P[j]$, algoritma tidak langsung berpindah satu karakter ke kanan seperti pada algoritma naive. Sebaliknya, algoritma menggunakan informasi dari pra-pemrosesan untuk menentukan berapa jauh pola bisa digeser agar pencocokan berikutnya lebih menjanjikan.

Untuk memungkinkan lompatan yang cerdas ini, Boyer-Moore melakukan tahap praproses terhadap pola P untuk membentuk struktur yang disebut last occurrence function atau fungsi kemunculan terakhir. Fungsi ini didefinisikan sebagai $L(x)$ adalah indeks tertinggi i sehingga $P[i] = x$ atau -1 jika karakter x tidak muncul dalam pola.

Fungsi $L(x)$ memetakan setiap karakter x dalam alfabet A ke indeks terakhir kemunculannya dalam pola P . Informasi ini digunakan saat terjadi ketidakcocokan: jika karakter dari teks $T[i]$ tidak cocok dengan karakter dalam pola $P[j]$, pola akan digeser sehingga karakter $T[i]$ disejajarkan dengan kemunculan terakhirnya dalam pola, atau jika tidak ada, digeser melewati posisi itu seluruhnya.

2.1.4 Algoritma Aho-Corasick

Algoritma Aho-Corasick adalah algoritma pencocokan pola seperti algoritma KMP dan BM yang menggunakan automaton dalam pemecahan masalah. Algoritma ini memiliki keunggulan terhadap algoritma KMP dan BM yaitu mampu mencari lebih dari satu pola dalam sekali pencarian. Algoritma ini bekerja menggunakan dua tahap berikut:

1. Pembuatan Automaton

Automaton dibuat menggunakan trie hasil semua pola yang dicari sebagai dasar. Automaton berdasarkan trie ini menggunakan node sebagai state automaton dan edges setiap node sebagai transisi dari satu state ke state lain berdasarkan next character dalam string. Trie ini dilengkapi menjadi automaton utuh dengan menambahkan suffix links dan failure links pada setiap node. Kedua link ini adalah transisi baru yang ditambahkan pada trie agar semua next character memiliki transisi untuk menjalankan automaton. Failure links adalah link yang menyambung pada root ketika menemukan next character yang tidak dikenal. Sementara itu, suffix links adalah link yang menyambung pada node lain

2. Pencarian Pola

Pencarian pola dilakukan dengan menjalankan automaton dengan character dalam string sebagai input yang mengubah state. Jika state merupakan state pola yang dicari, maka pola tersebut ditemukan. Ketika suatu character tidak ditemukan sebagai transisi pada suatu node, maka transisi dari link node dilihat juga. Proses tersebut dilakukan hingga suatu transisi untuk character tersebut ditemukan atau sampai ke

root node. Penjalanan automaton dilakukan hingga semua character dalam string telah dimasukkan dalam automaton.

2.2 Penjelasan Aplikasi

2.2.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat interpreted, multi-paradigma, dan memiliki sintaks yang sederhana serta mudah dipahami. Python mendukung paradigma pemrograman prosedural, berorientasi objek, dan fungsional. Bahasa ini populer karena fleksibilitasnya dan banyak digunakan dalam pengembangan aplikasi desktop, web, sains data, kecerdasan buatan, dan otomasi.

Dalam tugas besar ini, Python digunakan sebagai bahasa utama dalam membangun sistem Applicant Tracking System (ATS), khususnya karena Python memiliki pustaka ekosistem yang baik untuk pemrosesan teks seperti re untuk Regex, pengelolaan file PDF (PyPDF2), pembuatan GUI (PyQt5), serta koneksi ke basis data (mysql-connector-python).

2.2.2 PyQt

PyQt adalah sebuah pustaka Python yang menyediakan binding terhadap Qt, yaitu framework GUI lintas platform yang dikembangkan oleh Qt Company. PyQt memungkinkan pembuatan GUI dengan fitur-fitur seperti layout dinamis, event-driven interaction, dan komponen visual yang lengkap.

Dalam tugas besar ini ini, PyQt digunakan untuk membangun Graphical User Interface (GUI) tempat pengguna dapat memasukkan kata kunci pencarian, memilih algoritma pencocokan (KMP/BM), serta menampilkan hasil pencarian CV dalam bentuk interaktif seperti summary dan preview file CV.

2.2.3 PyPDF

PyPDF adalah pustaka Python yang memungkinkan manipulasi dan ekstraksi data dari file PDF. Fungsi utama dari pustaka ini dalam konteks ATS adalah mengonversi dokumen CV dalam format PDF menjadi string teks yang bisa dianalisis lebih lanjut menggunakan algoritma pattern matching.

2.2.4 Docker

Docker adalah platform open-source yang digunakan untuk mengembangkan, mengemas, dan menjalankan aplikasi di dalam container. Container adalah unit terisolasi yang membungkus aplikasi beserta seluruh dependensinya, sehingga dapat dijalankan secara konsisten di berbagai lingkungan, tanpa perlu khawatir tentang perbedaan sistem operasi atau konfigurasi perangkat lunak.

Docker membantu dalam pengembangan perangkat lunak karena menyediakan lingkungan yang portable, ringan, dan reproducible. Dalam konteks proyek ATS

(Applicant Tracking System), Docker dapat digunakan untuk menyediakan lingkungan yang konsisten untuk menjalankan aplikasi Python beserta dependensinya, menjalankan MySQL server dalam container tersendiri tanpa perlu instalasi manual, dan menyederhanakan proses deployment aplikasi ke server atau mesin lain, hanya dengan menjalankan perintah docker-compose up.

2.2.5 MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang bersifat open-source dan digunakan secara luas dalam berbagai aplikasi. MySQL mengorganisasi data dalam bentuk tabel yang saling terhubung melalui relasi. MySQL mendukung operasi CRUD (Create, Read, Update, Delete), transaksi, serta indeks untuk pencarian cepat.

Dalam sistem ATS, MySQL digunakan untuk menyimpan informasi pelamar seperti nama, kontak, posisi yang dilamar, serta path ke file CV. Dua tabel utama yang digunakan adalah ApplicantProfile dan ApplicationDetail, di mana satu pelamar bisa memiliki lebih dari satu lamaran.

2.2.6 MySQL Connector Python

MySQL Connector/Python adalah pustaka dari Oracle yang memungkinkan program Python untuk terhubung dan berinteraksi langsung dengan basis data MySQL. Pustaka ini mendukung berbagai fitur seperti eksekusi query SQL, transaksi, dan manajemen koneksi basis data secara efisien. Dalam proyek ini, mysql-connector-python digunakan untuk menyisipkan hasil ekstraksi informasi dari CV ke dalam tabel, mengambil data profil pelamar dari database untuk ditampilkan di GUI, dan mengelola relasi antara pelamar dan file CV-nya. Dengan pustaka ini, proses integrasi antara sistem ATS dan penyimpanan data menjadi lebih terstruktur dan konsisten.

2.2.7 UV

UV adalah manajer paket dan manajer virtual environment yang digunakan pada bahasa pemrograman Python. UV dikembangkan sebagai solusi serba guna yang menggabungkan fungsi dari beberapa alat populer seperti pip, venv, dan virtualenv dalam satu alat. Beberapa fitur utama dari uv diantaranya, pembuatan dan pengelolaan virtual environment (uv venv), instalasi dan penghapusan paket Python seperti pip (uv pip install), dan manajemen dependensi proyek melalui pyproject.toml. UV dapat digunakan untuk setup proyek ATS dan menjaga agar semua dependensi seperti PyQt, MySQL Connector, dan PyPDF terkelola dalam satu environment yang terpisah dari sistem global pada perangkat.

Bab 3

Analisis Pemecahan Masalah

3.1 Langkah-Langkah Pemecahan Masalah

Dalam mengembangkan sistem Applicant Tracking System (ATS) berbasis CV digital, dilakukan hal-hal seperti berikut:

1. Ekstraksi Teks dari CV PDF

File CV yang diunggah pengguna dengan format .pdf dikonversi menjadi string panjang yang merepresentasikan seluruh isi teks dari dokumen tersebut. Proses ini dilakukan menggunakan pustaka PyPDF.

2. Pembersihan dan Normalisasi Teks

Teks hasil ekstraksi dibersihkan dari karakter non-alfabet, diubah ke huruf kecil, dan direpresentasikan dalam bentuk yang sama untuk mempermudah proses pencocokan pola.

3. Input Kata Kunci oleh Pengguna

Pengguna memasukkan satu atau lebih kata kunci yang ingin dicari, dipisahkan dengan koma, serta memilih algoritma pencarian (KMP/BM).

4. Exact Matching dengan KMP/BM

Algoritma pencocokan string Knuth-Morris-Pratt (KMP) atau Boyer-Moore (BM) dijalankan terhadap setiap keyword untuk seluruh CV yang tersedia.

5. Fuzzy Matching

Jika tidak ditemukan exact match, sistem akan menggunakan Levenshtein Distance untuk mencari CV yang paling mirip berdasarkan ambang batas kemiripan tertentu.

6. Pencatatan dan Perangkingan Hasil

CV yang memiliki kecocokan tertinggi akan diurutkan berdasarkan jumlah dan frekuensi keyword yang cocok, lalu ditampilkan kepada pengguna.

7. Penampilan Informasi Summary

Informasi penting seperti nama, skill, pengalaman kerja, dan pendidikan ditampilkan melalui ekstraksi dengan Regex dan data dari basis data MySQL.

3.2 Proses Pemetaan Masalah Menjadi Elemen-Elemen Algoritma KMP dan BM

Pemetaan masalah menjadi elemen-elemen algoritma KMP dan BM terlihat jelas pada permasalahan utama program, yaitu pencarian CV berdasarkan keyword tertentu. Isi dari suatu CV dapat di-*parse* menjadi satu string yang utuh. Setelah string ini didapatkan, mencari *keyword* menjadi permasalahan menjadi *pattern* dalam string. Algoritma KMP dan BM adalah algoritma yang khusus dibuat untuk permasalahan itu dan tidak perlu dimodifikasi sama sekali agar dapat digunakan. Dari hasil matching ini dihitung jumlah match yang ditemukan dan dibandingkan dengan CV lain agar dapat menunjukkan CV dengan matches terbanyak.

3.3 Fitur Fungsional dan Arsitektur Aplikasi yang dibangun

Program dibuat dengan arsitektur *monolithic* yang membuat bagian-bagian antar program saling terikat. Arsitektur ini dipilih untuk memungkinkan pembuatan program dengan cepat dan juga membuat program sendiri sangkil. Secara lebih rinci, program dapat dibagi menjadi tiga bagian: koneksi database, alur program, dan fungsi-fungsi *matching*. Program ini dibuat dengan fitur-fitur selayak ATS sederhana. Secara singkat, program dibangun dengan fitur-fitur berikut:

- Membaca file CV dalam format PDF dan mengekstraksi teks secara otomatis.
- Penyimpanan data CV seperti nama penulis, tanggal lahir, dan lokasi file CV dalam *database*.
- Pencarian CV dengan *keyword matching* menggunakan algoritma Knuth-Morris-Pratt, Boyer-Moore, dan Aho-Corasick.
- Pencarian CV sekunder ketika tidak ada *matching keyword* menggunakan *fuzzy search*.
- Penampilan summary CV yang menampilkan daftar *skill*, *experience*, dan *education* menggunakan *regex*.

3.4 Contoh Ilustrasi Kasus

Pengguna ingin mencari pelamar yang menguasai teknologi "Python", "HTML", dan "C++". Pengguna akan melakukan hal-hal sebagai berikut:

1. Pengguna memasukkan kata kunci: Python, HTML, C++.
2. Memilih algoritma pencarian: KMP
3. Sistem mengekstrak teks dari seluruh CV dalam direktori.
4. Algoritma KMP mencari keberadaan kata kunci dalam teks tiap CV.
5. Sistem mengurutkan CV berdasarkan banyaknya keyword yang ditemukan:
 - CV1: Python (3x), HTML (1x), C++ (2x) → Total match(es): 6
 - CV2: Python (1x), C++ (1x) → Total match(es): 2

- CV3: HTML (1x) → Total match(es): 1

6. Hasil ditampilkan kepada pengguna dalam bentuk kartu informasi.

7. Pengguna bisa klik Summary untuk melihat detail ringkasan dan View CV untuk membuka file PDF asli.

Bab 4

Implementasi dan Pengujian

4.1 Spesifikasi Teknis Program

Program dimulai dari `main.py` yang menyalakan koneksi ke database, melakukan *loading* PDF secara asinkron dan membuka aplikasi GUI. Direktori `pages` berisi *python script* yang mengatur *main program logic* serta tampilan dari aplikasi. Direktori `functions` berisi *python script* yang menggunakan algoritma-algoritma khusus untuk mengimplementasikan fitur-fitur dalam aplikasi, seperti *pattern matcher*, *regex matcher*, *fuzzy matcher*, dan *PDF reader*. Direktori `util` berisi *script* yang membantu dalam setup awal. Direktori `connection` berisi *script* yang mengurus koneksi dengan database. Berikut adalah rincian kelas-kelas penting yang ada dalam program.

4.1.1 Searcher

Kelas ini adalah kelas yang melakukan searching pada seluruh CV yang disimpan dalam program. Berikut adalah method-method dalam kelas ini:

- `set_algorithm`:
mengubah algoritma yang dipakai oleh searcher dengan mengubah object string matcher yang disimpan.
- `search`:
melakukan searching dengan exact match terlebih dahulu dan dilanjutkan dengan fuzzy match ketika jumlah matches kurang. Exact match memiliki bobot 300 kali lipat dari fuzzy match.

4.1.2 PDFReader

Kelas ini adalah kelas yang berfungsi membaca PDF dan menyediakan isi text dari PDF sebagai string. Berikut adalah method-method dalam kelas ini:

- `open_pdf`:
membuka pdf dan menyimpan cache untuk isi PDF tersebut.
- `preload_pdf`:
memanggil `open_pdf` dalam background thread.

4.1.3 LevenshteinFuzzy

Kelas ini berfungsi dalam menghitung levenshtein distance suatu pola terhadap seluruh teks. Berikut adalah method-method dalam kelas ini:

- `fuzzy_search`:
melakukan fuzzy search dengan threshold max distance, mengembalikan daftar index dimana distance lebih kecil dari threshold.

4.1.4 SectionScraper

Kelas ini berfungsi untuk mengambil informasi dari PDF untuk summary yang dilakukan dengan regex. Berikut adalah method-method dalam kelas ini:

- `scrape_skills`:
Menghasilkan comma separated list dari skill yang ditemukan dalam skills section.
- `scrape_experience`:
Menghasilkan comma separated list dari experience yang ditemukan dalam experience sections (terdapat beberapa section yang terdefinisi sebagai experience section) dengan mencari keyword yang ditentukan.
- `scrape_education`:
Menghasilkan comma separated list dari education yang ditemukan dalam education sections (terdapat beberapa section yang terdefinisi sebagai education section) dengan mencari keyword yang ditentukan.

4.1.5 StringMatcher

Kelas ini adalah kelas abstrak yang mendefinisikan method-method untuk string matcher yang digunakan pada searcher. Berikut adalah method-method dalam kelas ini:

- `preprocessPattern`:
melakukan preprocessing pada semua pattern.
- `search`:
melakukan pencarian dengan pattern dan text dan menghasilkan daftar index pattern ditemukan.

4.1.6 KnuthMorrisPratt

Kelas ini adalah kelas yang mengimplementasikan algoritma KMP sebagai string matcher. Berikut adalah method-method dalam kelas ini:

- `preprocessPattern`:
generate lps.
- `search`:
melakukan pencarian dengan pattern dan text dan menghasilkan daftar index pattern ditemukan.

4.1.7 BoyerMoore

Kelas ini adalah kelas yang mengimplementasikan algoritma BM sebagai string matcher. Berikut adalah method-method dalam kelas ini:

- `preprocessPattern`:
generate last_occurence.
- `search`:
melakukan pencarian dengan pattern dan text dan menghasilkan daftar index pattern ditemukan.

4.1.8 AhoCorasick

Kelas ini adalah kelas yang mengimplementasikan algoritma AC sebagai string matcher. Berikut adalah method-method dalam kelas ini:

- preprocessPattern:
generate automaton.
- search:
melakukan pencarian dengan pattern dan text dan menghasilkan daftar index pattern ditemukan.

4.2 Tata Cara Penggunaan Program

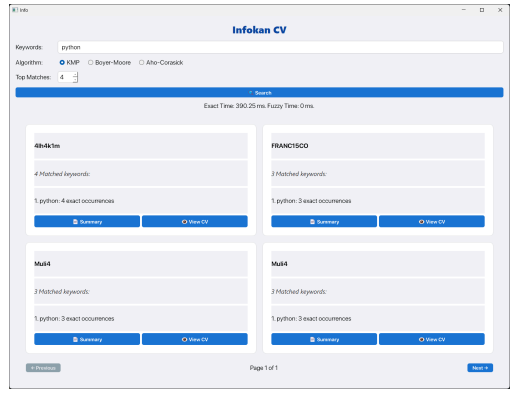
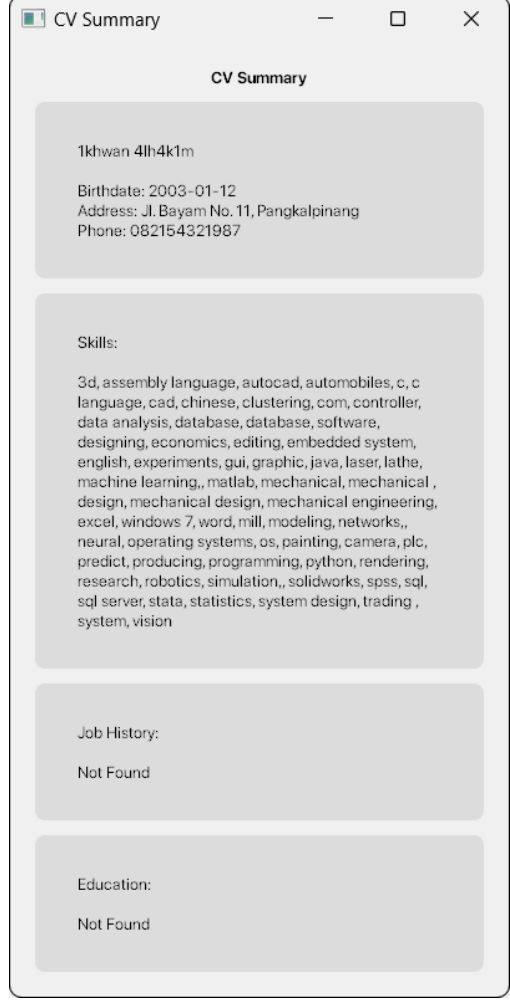
Untuk menjalankan program ini, diperlukan docker dan uv sebagai ketergantungan program. Program dijalankan dengan menyalakan docker compose dan script main.py pada direktori src. Docker compose diperlukan untuk memulai database dan mengisinya dengan data yang diperlukan, sementara main.py adalah *script* yang menjalankan program utama. Docker compose dinyalakan dengan perintah *docker compose up* sementara main.py dijalankan dengan perintah *uv run main.py*.

Setelah menjalankan program, pengguna dapat memasukkan keyword-keyword yang dicari dalam CV pada search bar dalam bentuk comma separated list. Setelah itu, pengguna dapat mengganti algoritma pencocokan pola yang digunakan dengan memilih pada radio button antara KMP, BM, atau AC. Pengguna juga dapat menentukan jumlah CV yang ingin dikembalikan dalam hasil pencarian. Setelah semua penyetelan sudah selesai, pengguna dapat memulai pencarian dengan menekan tombol cari.

Setelah pencarian selesai dilakukan, daftar semua CV yang memiliki keyword dan jumlah kemunculan keyword akan ditampilkan kepada pengguna. Pada tiap CV, pengguna dapat memilih untuk melihat ringkasan CV atau melihat CV-nya sendiri. Jika pengguna memilih untuk melihat ringkasan, sebuah window baru berisi ringkasan dari CV akan ditampilkan pada pengguna. Jika pengguna memilih untuk melihat CV, CV akan ditampilkan pada window baru menggunakan PDF viewer pengguna.

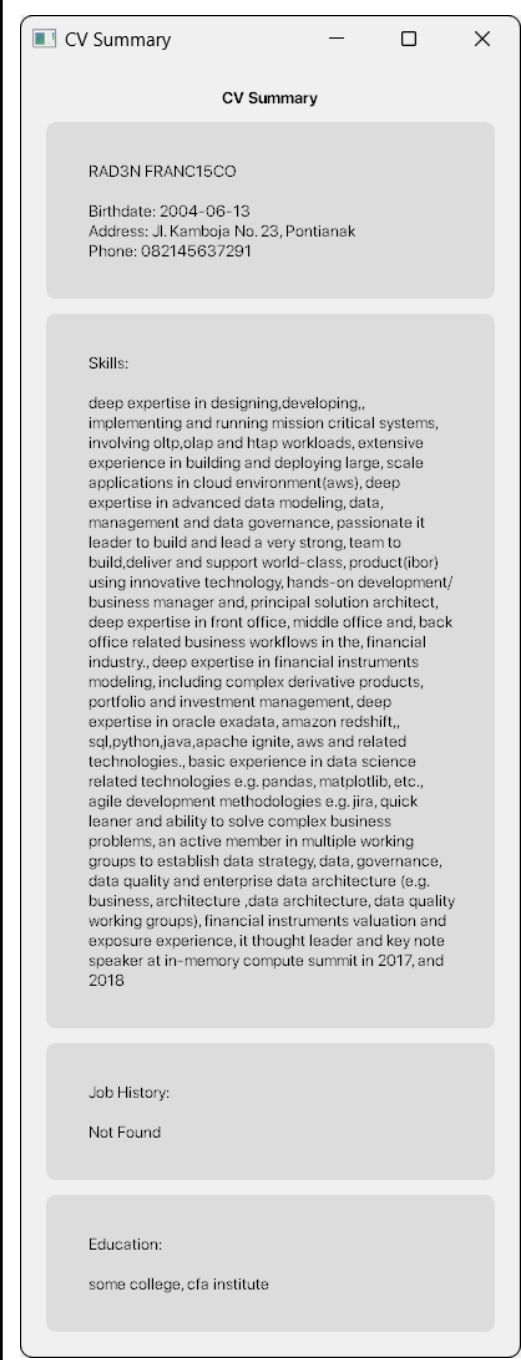
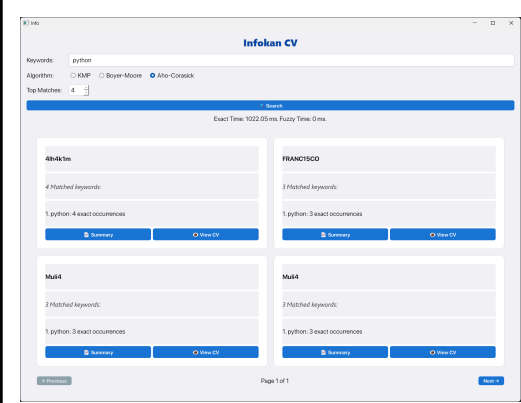
4.3 Hasil pengujian

No	Algorithm	Keyword(s)	Top Matches	Result
----	-----------	------------	-------------	--------

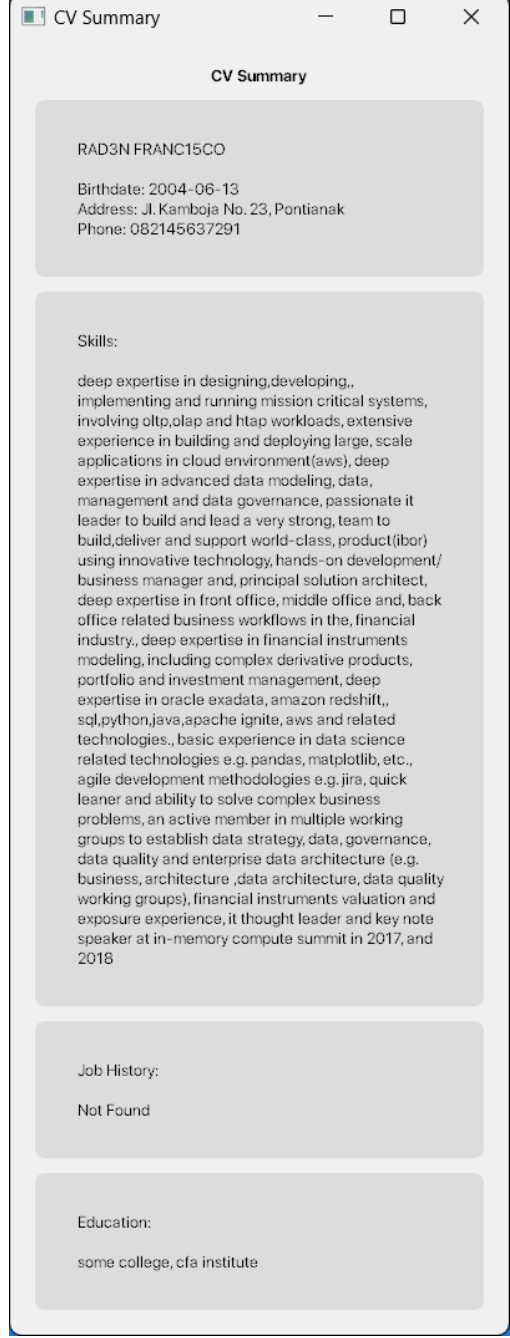
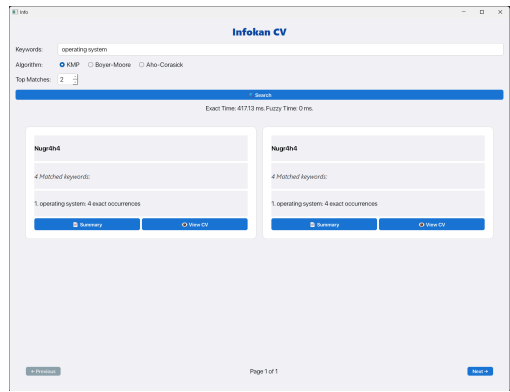
1.	KMP	python	5	 
----	-----	--------	---	--

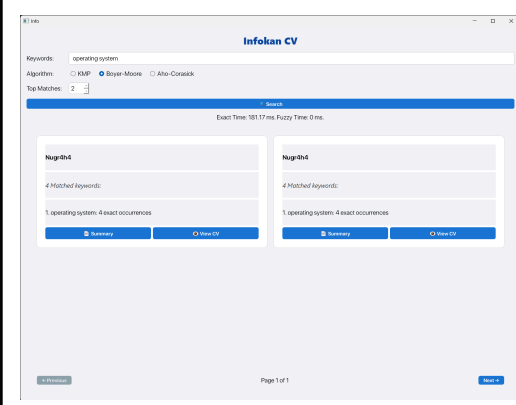
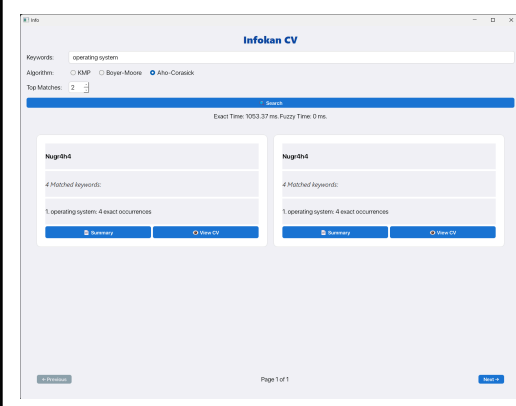
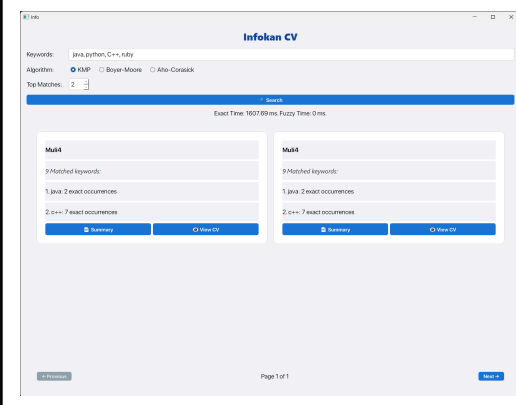
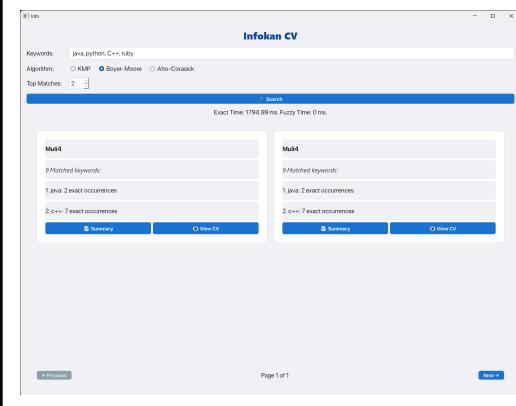
				<div><div>CV Summary</div><div><div>CV Summary</div><div><div>RAD3N FRANCISCO</div><div>Birthdate: 2004-06-13 Address: Jl. Kamboja No. 23, Pontianak Phone: 082145637291</div></div><div><div>Skills:</div><div>deep expertise in designing,developing,, implementing and running mission critical systems, involving oltp,olap and htap workloads, extensive experience in building and deploying large, scale applications in cloud environment(aws), deep expertise in advanced data modeling, data, management and data governance, passionate it leader to build and lead a very strong, team to build,deliver and support world-class, product(ibor) using innovative technology, hands-on development/ business manager and, principal solution architect, deep expertise in front office, middle office and, back office related business workflows in the, financial industry,, deep expertise in financial instruments modeling, including complex derivative products, portfolio and investment management, deep expertise in oracle exadata, amazon redshift,, sql,python,java,apache ignite, aws and related technologies,, basic experience in data science related technologies e.g. pandas, matplotlib, etc., agile development methodologies e.g. jira, quick leaner and ability to solve complex business problems, an active member in multiple working groups to establish data strategy, data, governance, data quality and enterprise data architecture (e.g. business, architecture ,data architecture, data quality working groups), financial instruments valuation and exposure experience, it thought leader and key note speaker at in-memory compute summit in 2017, and 2018</div></div><div><div>Job History:</div><div>Not Found</div></div><div><div>Education:</div><div>cfa institute, some college</div></div></div></div>
2.	BM	python	5	<div><div>Infekan CV</div><div><div>Keywords: python</div><div>Algorithms: <input type="checkbox"/> BSM <input checked="" type="checkbox"/> Boyer-Moore <input type="checkbox"/> Aho-Corasick</div><div>Top Matches: 4</div><div>Exact Time: 32716ms Fuzzy Time: 0ms</div><div><div><div>Albda-Tm</div><div>4 Matched keywords: 1 python: 4 exact occurrences</div><div><div>Summary</div><div>View CV</div></div></div><div><div>FRANCISCO</div><div>3 Matched keywords: 1 python: 3 exact occurrences</div><div><div>Summary</div><div>View CV</div></div></div><div><div>Multi4</div><div>3 Matched keywords: 1 python: 3 exact occurrences</div><div><div>Summary</div><div>View CV</div></div></div><div><div>Multi4</div><div>3 Matched keywords: 1 python: 3 exact occurrences</div><div><div>Summary</div><div>View CV</div></div></div></div><div><div>Page: 1 of 1</div><div>Next ></div></div></div></div>

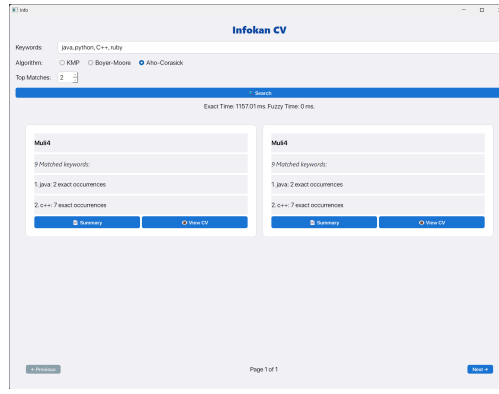
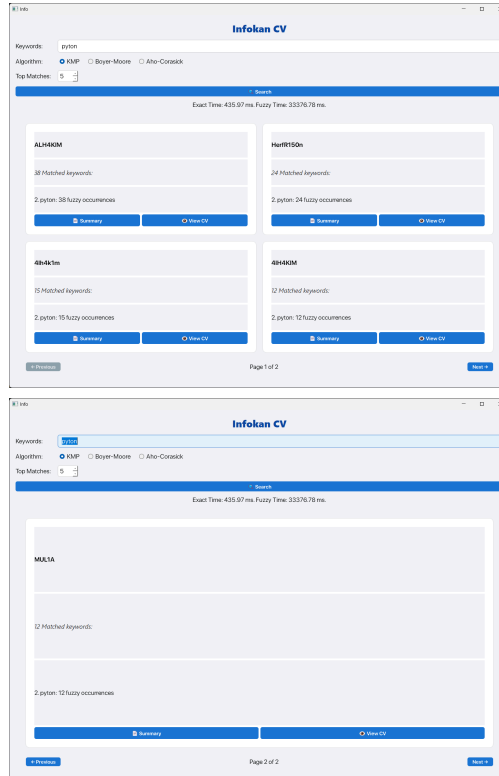
				<div> <div>CV Summary</div> <div>CV Summary</div> <div> 1khwan 4lh4k1m Birthdate: 2003-01-12 Address: Jl. Bayam No. 11, Pangkalpinang Phone: 082154321987 </div> <div> Skills: 3d, assembly language, autocad, automobiles, c, c language, cad, chinese, clustering, com, controller, data analysis, database, database, software, designing, economics, editing, embedded system, english, experiments, gui, graphic, java, laser, lathe, machine learning,, matlab, mechanical, mechanical , design, mechanical design, mechanical engineering, excel, windows 7, word, mill, modeling, networks,, neural, operating systems, os, painting, camera, plc, predict, producing, programming, python, rendering, research, robotics, simulation,, solidworks, spss, sql, sql server, stata, statistics, system design, trading , system, vision </div> <div> Job History: Not Found </div> <div> Education: Not Found </div> </div>
--	--	--	--	---

				
3.	Aho-Corasick	python	5	

				<div> <div>CV Summary</div> <div>CV Summary</div> <div> 1khwan 4lh4k1m Birthdate: 2003-01-12 Address: Jl. Bayam No. 11, Pangkalpinang Phone: 082154321987 </div> <div> Skills: 3d, assembly language, autocad, automobiles, c, c language, cad, chinese, clustering, com, controller, data analysis, database, database, software, designing, economics, editing, embedded system, english, experiments, gui, graphic, java, laser, lathe, machine learning,, matlab, mechanical, mechanical , design, mechanical design, mechanical engineering, excel, windows 7, word, mill, modeling, networks,, neural, operating systems, os, painting, camera, plc, predict, producing, programming, python, rendering, research, robotics, simulation,, solidworks, spss, sql, sql server, stata, statistics, system design, trading , system, vision </div> <div> Job History: Not Found </div> <div> Education: Not Found </div> </div>
--	--	--	--	---

				
4.	KMP	operating system	2	

5.	BM	operating system	2	
6.	Aho-Corasick	operating system	2	
7.	KMP	java, python, C++, ruby	2	
8.	BM	java, python, C++, ruby	2	

9.	Aho-Corasick	java, python, C++, ruby	2	
10.	KMP	pyton	5	

4.4 Analisis hasil pengujian

Pada percobaan 1 - 3, terlihat untuk satu *keyword*, algoritma KMP dan BM berjalan dengan waktu yang relatif sama dan lebih cepat dari Aho-Corasick. Ini karena proses pembuatan Trie dalam algoritma Aho-Corasick membutuhkan waktu lebih lama dari KMP dan BM. Pada percobaan 4 - 6, dengan *keyword* yang lebih panjang, kesimpulan pengujian juga sama dengan percobaan sebelumnya. Untuk pengujian 7 - 9 dengan *multiple keywords*, terlihat bahwa Aho-Corasick lebih unggul dari algoritma KMP dan BM. Ini karena Aho-Corasick dapat mencari banyak *pattern* hanya dalam sekali jalan, membuat *overhead* pembuatan Trie tidak terasa. Terakhir pada percobaan 10, dilakukan *searching* dengan *keyword* yang *typo* yaitu “pyton”. Ditemukan *keyword* dengan *fuzzy matching*, walau lebih lama dari algoritma lainnya.

Bab 5

Kesimpulan, Saran, dan Refleksi

Kesimpulan

Program berhasil berjalan sebagai sebuah *Application Tracking System* untuk mengelola banyak CV. Algoritma KMP, BM, dan AC dapat digunakan sebagai algoritma pencocokan pola untuk mencari keyword dalam banyak CV secara sangkil. Algoritma AC dapat berjalan lebih cepat daripada KMP dan BM karena dapat mencari semua keyword dalam satu *pass*. Pencocokan regex juga dapat digunakan dalam mencari informasi dalam CV. Namun, penggunaan ini sangat terbatas karena isi CV sangat fleksibel dan banyak format yang dapat muncul dalam CV. Hal ini menyebabkan sangat sulit untuk menulis regex yang dapat menangkap informasi dengan format yang beragam. Penggunaan regex untuk mencari informasi sebaiknya digunakan pada format yang lebih seragam, seperti ATS tetapi dibatasi pada satu bidang tertentu.

Saran

- Lakukan normalisasi pada data CV
- Membaca dan mempelajari dokumentasi
- Menggunakan pustaka yang tepat agar pengerjaan berjalan dengan sangkil

Refleksi

Tugas besar 3 mata kuliah strategi algoritma memberikan pelajaran berharga dalam memahami *string matching*, terutama *string matching* dengan algoritma Knutt Moris Pratt dan Boyer Moore. Kami mengetahui bahwa kedua algoritma tersebut memiliki keunggulan masing-masing dalam hal efisiensi pencarian pola. KMP unggul dalam kestabilan performa pada teks yang memiliki banyak pengulangan, sedangkan Boyer-Moore sangat efisien untuk teks panjang dengan alfabet yang besar karena dapat melompati lebih banyak karakter saat pencarian.

Melalui implementasi nyata dalam sistem Applicant Tracking System (ATS), kami dapat melihat secara langsung bagaimana algoritma-algoritma ini bekerja dalam proses pencarian kata kunci di dokumen CV. Selain itu, kami juga belajar tentang Aho-Corasick, algoritma yang dapat pencarian banyak pola sekaligus, dan memiliki performa lebih unggul dalam kasus multi-keyword matching.

Secara keseluruhan, tugas besar ini memberikan pelajaran bagi kami untuk dalam membuat suatu aplikasi dengan menerapkan algoritma-algoritma yang telah kami pelajari pada kelas kuliah.

Lampiran

Pranala Repository Github

https://github.com/drianto/Tubes3_info

Pranala video Youtube

▶ Infokan CV - Tubes Stima

Tabel Ketercapaian

No	Poin	Ya	Tidak
1	Aplikasi dapat dijalankan.	✓	
2	Aplikasi menggunakan basis data berbasis SQL dan berjalan dengan lancar.	✓	
3	Aplikasi dapat mengekstrak informasi penting menggunakan Regular Expression (Regex).	✓	
4	Algoritma Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM) dapat menemukan kata kunci dengan benar.	✓	
5	Algoritma Levenshtein Distance dapat mengukur kemiripan kata kunci dengan benar.	✓	
6	Aplikasi dapat menampilkan summary CV applicant.	✓	
7	Aplikasi dapat menampilkan CV applicant secara keseluruhan.	✓	
8	Membuat laporan sesuai dengan spesifikasi.	✓	
9	Membuat bonus enkripsi data profil applicant.		✓
10	Membuat bonus algoritma Aho-Corasick.	✓	
11	Membuat bonus video dan diunggah pada Youtube.	✓	

Daftar Pustaka

- Hutabarat, R. A. P., Hutapea, J. S., Lubis, M. M. 2023. PENERAPAN ALGORITMA STRING MATCHING DALAM PENCOCOKAN DATA STRING. Diakses melalui <https://ejurnal.ulbi.ac.id/index.php/informatika/article/download/2998/1193/6164>
- Khodra, Masayu Leylia. 2025. String Matching dengan Regular Expression. Diakses melalui [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/02-Algoritma-Brute-Force-\(2025\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/02-Algoritma-Brute-Force-(2025)-Bag1.pdf)
- Kuhlman, Dave. 2025. A Python Book: Beginning Python, Advanced Python, and Python Exercises. Diakses melalui https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html
- Munir, Rinaldi. 2025. Pencocokan string (string matching) dengan algoritma brute force, KMP, Boyer-Moore. Diakses melalui [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-\(2025\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/23-Pencocokan-string-(2025).pdf)