LAPORAN TUGAS BESAR #3
Pemanfaatan Pattern Matching untuk Membangun Sistem ATS (Applicant Tracking System) Berbasis CV Digital

IF2211– Strategi Algoritma

Kelompok "*kerjalembut*"

Anggota Kelompok:

Muhammad Adli Arindra	18222089
Benedictus Nelson	13523150
Stefan Mattew Susanto	13523020

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2024

Daftar Isi

1. Deskripsi Tugas

Curriculum Vitae (CV) adalah dokumen vital yang merangkum kualifikasi dan pengalaman kandidat. Ini adalah representasi utama pelamar, membantu perusahaan memahami kompetensi dan kesesuaian mereka untuk pekerjaan. CV penting untuk membuat kesan pertama dan menentukan kemajuan kandidat dalam proses seleksi.

Di era digital, keamanan data dan akses sangat penting. Proses rekrutmen telah berubah dengan teknologi untuk efisiensi. Perusahaan sering menerima ribuan CV, yang sulit ditinjau manual. Ini menyebabkan proses yang lambat dan hilangnya kandidat. Applicant Tracking System (ATS) adalah solusi inovatif. ATS mempermudah penyaringan dan pencocokan informasi kandidat dari CV. Ini memungkinkan pengelolaan ribuan lamaran secara otomatis dan menemukan kandidat relevan dengan cepat.

Namun, ATS menghadapi tantangan dalam memproses CV PDF yang tidak terstruktur. Dokumen ini memerlukan metode canggih untuk mengekstrak informasi penting seperti identitas, pengalaman, keahlian, dan pendidikan. Pattern matching adalah solusi ideal untuk tantangan ini. Pattern matching adalah teknik mencari dan mencocokkan pola dalam teks. Algoritma Boyer-Moore dan Knuth-Morris-Pratt (KMP) sering digunakan karena efisiensi tinggi dalam pencarian teks besar. Algoritma ini membantu ATS mengidentifikasi informasi CV dengan kecepatan dan akurasi optimal.

Tugas besar ini mengimplementasikan sistem deteksi informasi pelamar berbasis CV digital. Metode deteksi pola yang digunakan adalah Boyer-Moore dan KMP. Sistem ini akan terhubung dengan identitas kandidat melalui basis data. Tujuannya adalah membangun sistem yang dapat mengenali profil pelamar secara lengkap hanya dengan CV digital.

2. Landasan Teori

Pada bagian berikut akan dibahas dasar-dasar teoritis yang menjadi landasan pengembangan sistem Applicant Tracking System (ATS) berbasis CV digital. Akan dijelaskan secara rinci algoritma *pattern matching* utama seperti Knuth-Morris-Pratt (KMP), Boyer-Moore, dan Aho-Corasick yang digunakan dalam proses pencarian. Selain itu, bagian ini juga akan memberikan gambaran umum mengenai arsitektur dan fungsionalitas proyek yang dibangun.

2.1. Dasar Teori

Algoritma *pattern matching* menjadi krusial dalam pembangunan sistem Applicant Tracking System (ATS) ini. Teknik ini memungkinkan pencarian efisien terhadap *keyword* tertentu dalam ribuan dokumen CV digital. Dengan demikian, sistem dapat dengan cepat mengidentifikasi CV yang paling relevan dengan kriteria pencarian yang ditetapkan oleh pengguna atau tim HR. Tujuannya adalah mempercepat proses penyaringan kandidat.

Algoritma Knuth-Morris-Pratt (KMP) adalah salah satu metode *pattern matching* yang efisien. KMP bekerja dengan menghindari pengecekan ulang karakter yang sudah diketahui cocok. Ini dilakukan dengan membangun tabel *prefix function* atau *LPS array* dari pola yang dicari. Tabel ini memandu pergeseran pola ketika terjadi ketidakcocokan, sehingga tidak perlu membandingkan setiap karakter satu per satu dari awal. Efisiensi KMP membuatnya cocok untuk pencarian teks di dokumen besar.

Berbeda dengan KMP, algoritma Boyer-Moore (BM) memulai pencarian dari akhir pola menuju awal. BM memanfaatkan dua heuristik utama: *bad character rule* dan *good suffix rule*. Aturan ini memungkinkan algoritma untuk melompati banyak karakter saat terjadi ketidakcocokan, seringkali menghasilkan kinerja yang lebih cepat dibandingkan KMP pada teks yang panjang. Meskipun keduanya efisien, BM cenderung lebih cepat dalam banyak kasus praktis karena kemampuan lompatan yang lebih besar.

Algoritma Aho-Corasick adalah perbaikan dari KMP untuk masalah pencarian multi-pola. Algoritma ini memungkinkan sistem untuk mencari semua *keyword* sekaligus dalam satu kali traversal teks. Berbeda dengan KMP dan BM yang mencari satu pola dalam satu waktu, Aho-Corasick membangun *finite automaton* dari semua pola yang dicari. Ini menjadikannya sangat efisien ketika ada banyak *keyword* yang perlu dicocokkan secara bersamaan. Algoritma ini merupakan opsi bonus untuk pencarian *multi-pattern* yang lebih cepat.

Apabila tidak ditemukan kecocokan *exact match* menggunakan KMP atau Boyer-Moore, sistem akan beralih ke *fuzzy matching*. Untuk ini, algoritma Levenshtein Distance digunakan. Levenshtein Distance mengukur tingkat kemiripan antara dua string dengan menghitung jumlah operasi (penyisipan, penghapusan, atau penggantian karakter) minimum yang diperlukan untuk mengubah satu string menjadi string lain. Ini sangat berguna untuk mengatasi kesalahan pengetikan (*typo*) pada *keyword* yang dimasukkan pengguna. Dengan ambang batas kemiripan tertentu, sistem tetap dapat menampilkan hasil yang relevan.

Regular Expression (Regex) adalah urutan karakter yang membentuk pola pencarian. Regex sangat efektif untuk mengekstrak informasi penting dari teks CV secara otomatis. CV seringkali memiliki struktur yang bervariasi, namun informasi kunci seperti nama, kontak, keahlian, pengalaman kerja, dan riwayat pendidikan sering mengikuti pola tertentu. Dengan Regex, sistem dapat mendefinisikan pola-pola ini dan secara akurat mengidentifikasi serta mengekstraksi *keypoints* tersebut dari teks CV yang tidak terstruktur.

2.2. Penjelasan Projek

Proyek ini adalah implementasi Applicant Tracking System (ATS) berbasis CV digital yang dikembangkan menggunakan bahasa pemrograman Python. Antarmuka pengguna (frontend) dibangun dengan customtkinter. Pemilihan customtkinter didasarkan pada kemampuannya untuk menyediakan aplikasi desktop yang modern dan intuitif. Pustaka ini memungkinkan pengembangan antarmuka yang menarik tanpa mengorbankan performa.

Basis data MySQL digunakan untuk menyimpan profil pelamar dan informasi terkait aplikasi. Untuk memastikan portabilitas dan kemudahan *deployment*, basis data ini di-*dockerize*. Pendekatan ini dipilih karena Docker memungkinkan enkapsulasi basis data beserta dependensinya ke dalam sebuah kontainer. Hal ini memastikan konsistensi lingkungan, sehingga basis data dapat dijalankan dengan mudah dan tanpa masalah di berbagai sistem operasi atau server yang berbeda.

Inti dari sistem ini adalah kemampuan *pattern matching* dan *path finding*. Semua algoritma *pathfinding* diimplementasikan secara mandiri, tanpa menggunakan pustaka bawaan Python. Sistem ini mendukung pencarian CV berdasarkan *keyword* yang dimasukkan pengguna, dengan pilihan algoritma Knuth-Morris-Pratt (KMP) dan Boyer-Moore untuk *exact matching*. Selain itu, jika tidak ada kecocokan sempurna, sistem akan melakukan *fuzzy matching* menggunakan algoritma Levenshtein Distance untuk memperhitungkan kemungkinan *typo* atau variasi penulisan.

Setelah pencarian, pengguna dapat melihat ringkasan CV. Ringkasan ini menampilkan informasi penting yang diekstrak menggunakan Regular Expression (Regex), seperti ringkasan pelamar, keahlian, pengalaman kerja, dan riwayat pendidikan. Fitur ini membantu pengguna (HR atau rekruter) mendapatkan gambaran cepat tentang kandidat tanpa perlu membuka seluruh dokumen CV asli. Sistem ini dirancang untuk menyediakan pengalaman pengguna yang responsif dan efisien.

3. Analisis Pemecahan Masalah

Bagian ini menguraikan pendekatan sistematis yang digunakan dalam mengembangkan solusi Applicant Tracking System (ATS) berbasis CV digital. Pembahasan akan mencakup langkah-langkah pemecahan masalah, pemetaan komponen proyek, fitur fungsional yang tersedia, serta contoh ilustrasi kasus. Ini bertujuan untuk memberikan pemahaman mendalam tentang perancangan dan implementasi sistem.

3.1. Langkah Pemecahan Masalah

Pengembangan sistem Applicant Tracking System (ATS) ini diawali dengan analisis mendalam terhadap permasalahan dalam proses rekrutmen. Tahap awal melibatkan identifikasi tantangan yang dihadapi perusahaan. Tantangan utamanya adalah volume lamaran kerja yang tinggi serta format CV yang bervariasi, khususnya PDF yang tidak selalu terstruktur. Hal ini menyulitkan peninjauan manual oleh tim HR dan mengakibatkan proses yang tidak efisien.

Langkah selanjutnya adalah perancangan arsitektur sistem. Kami merancang alur kerja yang dimulai dengan konversi CV dari format PDF menjadi satu *string* panjang yang memuat seluruh teks. Representasi ini mempermudah pencocokan pola menggunakan algoritma *string matching*. Kemudian, algoritma KMP dan Boyer-Moore diimplementasikan untuk melakukan pencarian *keyword* secara *exact match*. Apabila tidak ditemukan kecocokan, sistem akan menerapkan algoritma Levenshtein Distance untuk *fuzzy matching*, mempertimbangkan kemungkinan *typo* pada *keyword*. Selain itu, Regular Expression (Regex) digunakan untuk mengekstrak informasi penting dari teks CV.

3.2. Pemetaan Masalah

Dalam sistem ini, setiap file CV dalam format PDF akan diubah menjadi sebuah objek. Objek ini akan menyimpan teks CV sebagai sebuah *string* panjang untuk keperluan algoritma *pattern matching*. Selain itu, objek tersebut juga akan menyimpan representasi XML dari CV dalam bentuk *string*. Hal ini memungkinkan Regular Expression (Regex) untuk memanfaatkan struktur XML dalam mengekstraksi informasi. Representasi linear teks ini menjadi dasar dalam mencari CV yang paling relevan dengan *keyword* yang dimasukkan oleh pengguna. Tujuan konversi ini adalah mempermudah pencocokan pola menggunakan algoritma *string matching*, sehingga setiap *keyword* dapat dicari secara efisien.

Keyword pencarian yang dimasukkan oleh pengguna akan berfungsi sebagai *pattern* yang akan dicocokkan dengan *string* teks CV. Proses pencarian ini dilakukan menggunakan algoritma pencocokan *string* Knuth-Morris-Pratt (KMP) dan Boyer-Moore (BM). Pengguna dapat memilih salah satu dari dua algoritma pencarian tersebut melalui tombol *toggle* pada antarmuka. Selain itu, algoritma Aho-Corasick juga dapat digunakan sebagai alternatif metode pencarian *keyword* yang efisien untuk *multi-pattern matching*.

Regular Expression (Regex) digunakan untuk mengekstrak informasi penting dari teks CV secara otomatis. Informasi yang diekstraksi meliputi ringkasan pelamar, keahlian, pengalaman kerja (seperti tanggal dan jabatan), dan riwayat pendidikan (seperti tanggal kelulusan, universitas, dan gelar). Ekstraksi ini penting untuk menampilkan ringkasan profil yang relevan kepada pengguna.

Apabila tidak ditemukan kecocokan *exact match* menggunakan algoritma KMP, Boyer-Moore, maupun Aho-Corasick, sistem akan beralih ke pencarian *fuzzy match*. Untuk setiap *keyword* yang belum ditemukan, sistem akan mencari CV yang paling mirip berdasarkan tingkat kemiripan di atas ambang batas tertentu (*threshold*). Metode perhitungan tingkat kemiripan ini menggunakan algoritma Levenshtein Distance. Hal ini mempertimbangkan kemungkinan adanya kesalahan pengetikan (*typo*) oleh pengguna atau HR saat memasukkan kata kunci. Pengembang memiliki keleluasaan untuk menentukan nilai ambang batas persentase kemiripan ini, dengan syarat dilakukan pengujian untuk menemukan nilai *tuning* yang optimal dan dijelaskan secara rinci dalam laporan.

3.3. Fitur Fungsional

Sistem ini menyediakan fitur utama untuk pencarian CV berdasarkan *keyword*. Pengguna dapat memasukkan satu atau lebih *keyword* yang diinginkan. Mereka juga dapat memilih algoritma pencocokan yang akan digunakan untuk *exact matching* antara Knuth-Morris-Pratt (KMP), Boyer-Moore (BM), atau Aho-Corasick. Pilihan algoritma ini memengaruhi cara sistem memindai dan mencocokkan *keyword* dengan isi CV.

Selain *exact matching*, sistem juga mendukung pencarian *fuzzy matching*. Fitur ini diaktifkan apabila tidak ditemukan satupun kecocokan *exact match* untuk suatu *keyword*. Sistem akan mencari CV yang paling mirip berdasarkan tingkat kemiripan menggunakan algoritma Levenshtein Distance. Ini membantu pengguna tetap mendapatkan hasil relevan meskipun ada kesalahan penulisan pada *keyword*.

Sistem memungkinkan pengguna untuk melihat ringkasan (*summary*) dari CV yang cocok. Ringkasan ini menampilkan informasi penting dari pelamar. Informasi tersebut meliputi identitas (nama, kontak, dan informasi pribadi lainnya) yang diperoleh dari basis data. Data lain seperti ringkasan pelamar, keahlian, pengalaman kerja, dan riwayat pendidikan diekstrak menggunakan Regular Expression (Regex).

Pengguna juga dapat melihat file PDF CV secara keseluruhan langsung dari program. Setiap kartu hasil pencarian menyediakan tombol "View CV". Fitur ini memungkinkan pengguna untuk meninjau dokumen asli CV secara lengkap.

3.4. Contoh Ilustrasi Kasus

Untuk memberikan pemahaman yang lebih konkret, berikut disajikan contoh kasus penerapan sistem CV ATS beserta prosesnya dan contoh *output* yang dihasilkan. Misalkan seorang rekruter sedang mencari kandidat untuk posisi "Software Engineer" yang memiliki keahlian "Python", "React", dan "SQL". Rekruter akan memasukkan *keyword* tersebut ke dalam kolom input "Keywords" pada antarmuka aplikasi, dipisahkan dengan koma.

Selanjutnya, rekruter akan memilih algoritma pencarian yang diinginkan, misalnya "KMP", melalui tombol *toggle* yang tersedia. Rekruter juga dapat menentukan jumlah CV teratas yang ingin ditampilkan, misalnya "5", menggunakan "Top Matches Selector". Setelah semua kriteria dimasukkan, rekruter akan menekan "Search Button" untuk memulai proses pencarian.

Sistem akan memindai *string* teks dari semua CV yang ada di basis data menggunakan algoritma KMP. Setiap file CV dalam format PDF telah dikonversi menjadi satu *string* panjang yang memuat seluruh teksnya. Setelah proses pencarian *exact match* selesai, sistem akan menampilkan waktu eksekusi pencarian tersebut. Jika ada *keyword* yang tidak ditemukan secara *exact match*, sistem akan melakukan *fuzzy matching* menggunakan algoritma Levenshtein Distance dan menampilkan waktu pencarian *fuzzy match* juga.

Hasil pencarian akan ditampilkan dalam bentuk tampilan beberapa kartu CV. Setiap kartu memuat informasi seperti nama kandidat, jumlah kecocokan *keyword* yang ditemukan, serta daftar *keyword* yang cocok beserta frekuensi kemunculannya. Contohnya, kartu "Farhan" mungkin menampilkan "Matched keywords: 1. React: 1 occurrence, 2. Express: 2 occurrences, 3. HTML: 1 occurrence". Dari hasil ini, rekruter dapat mengklik tombol "Summary" untuk melihat ekstraksi informasi dari CV seperti identitas, keahlian, pengalaman kerja, dan riwayat pendidikan yang didapatkan dari Regex. Atau, mereka dapat menekan tombol "View CV" untuk melihat langsung file CV asli kandidat.

4. Implementasi dan Pengujian

Bab ini akan menguraikan proses implementasi dan pengujian sistem Applicant Tracking System (ATS) berbasis CV digital. Pembahasan mencakup spesifikasi teknis program, panduan penggunaan aplikasi, serta hasil dan analisis pengujian yang dilakukan. Bagian ini bertujuan untuk memberikan gambaran lengkap mengenai fungsionalitas dan kinerja sistem yang sudah kami buat.

4.1. Spesifikasi Teknis Program

Struktur Data

1. **ApplicantProfile** – Menyimpan data pribadi pelamar seperti nama, tanggal lahir, alamat, dan nomor telepon
2. **ApplicationDetail** – Menyimpan detail lamaran, termasuk role yang dilamar dan path ke file CV PDF
3. **ApplicationPDF** – Menampung teks hasil ekstraksi PDF beserta raw HTML-nya
4. **SearchResult** – Menggabungkan ApplicantProfile, ApplicationDetail, ApplicationPDF, dan hasil pencocokan kata kunci. Saat objek dibuat, program mengekstrak ringkasan CV menggunakan Regex.extract_summary
5. **Summary** – Struktur data hasil ekstraksi bagian ringkasan, pendidikan, pengalaman, dsb. dari CV

Fungsi / Prosedur Utama

- **PDFReader:** read_text dan read_raw untuk membaca teks atau HTML dari file PDF menggunakan pdfminer.six
- **PatternMatching:** Mengimplementasikan algoritma KMP, Boyer-Moore, Levenshtein Distance, dan Aho-Corasick untuk mencari kata kunci pada teks CV
- **Regex.extract_summary:** Membuang tag HTML, memotong setiap bagian (experience, education, skills, dsb.), lalu mengembalikan objek Summary
    - Daftar ALLOWED_SKILLS memastikan hanya kata kunci keterampilan yang diizinkan yang diekstrak dari bagian “Skills”
    - Fungsi _get_section digunakan untuk mengambil isi setiap bagian hingga header berikutnya
- **ApplicantDatabase**: Menyediakan berbagai operasi CRUD untuk tabel ApplicantProfile, ApplicationDetail, dan ApplicationPDF di MySQL
- **Homepage:** Komponen GUI utama berbasis customtkinter. Terdapat pilihan algoritma (KMP, Boyer-Moore, atau Aho-Corasick), kolom kata kunci, dan jumlah hasil yang ingin ditampilkan
    - Metode prefetch memuat data CV dari basis data ke memori terlebih dahulu sebelum pencarian berlangsung
    - Metode _on_search menjalankan pencarian kata kunci pada setiap CV dengan algoritma yang dipilih. Hasil dengan skor tertinggi ditampilkan dalam bentuk CvCard

- **CvCard:** Widget kartu hasil pencarian yang menampilkan nama pelamar serta dua tombol: “Summary” dan “View CV”. Tombol “Summary” menampilkan jendela ringkasan, sedangkan “View CV” membuka PDF melalui aplikasi bawaan sistem operasi

Alur Proses

1. **Prefetch** – Data CV dan metadata dimuat dari database ke prefetched_data untuk mempercepat pencarian
2. **Pencarian** – Saat tombol Search diklik, _on_search melakukan pencocokan kata kunci dengan algoritma yang dipilih dan menghitung skor ketercocokan. Hanya hasil dengan kata kunci cocok yang ditampilkan
3. **Tampilan Hasil** – Setiap SearchResult diwujudkan menjadi sebuah CvCard, dimana pengguna dapat melihat ringkasan atau membuka file PDF asli

Metode Optimisasi

1. **Parallelization** – Proses pencocokan kata kunci terhadap seluruh data dilakukan secara paralel menggunakan ThreadPoolExecutor. Setiap entri diproses oleh thread terpisah untuk mempercepat pencarian ketika data berjumlah besar.
2. **Preprocessing –** Kata kunci dari input pengguna dibersihkan dan diseragamkan (diubah ke huruf kecil, spasi dihapus, duplikat dibuang) sebelum proses pencocokan dilakukan, sehingga meningkatkan efisiensi dan akurasi algoritma pencarian.
3. **Prefetching** – Data CV (cv_text) dan metadata (profil pelamar, detail aplikasi) dimuat ke dalam memori terlebih dahulu saat inisialisasi agar siap digunakan saat pencocokan dimulai. Hal ini menghindari delay saat runtime.

4.2. Cara Penggunaan Program

1. **Persiapan Lingkungan**

- Pastikan sudah terpasang Python dan Docker.
- Buat lingkungan virtual dan instal dependensi:

```
python -m venv venv
./venv/Scripts/activate
pip install -r requirements.txt
python -m src.utils.sql  # seeding basis data SQL
```

2. **Menjalankan Aplikasi**

- Jalankan layanan database MySQL melalui Docker lalu buka program:

```
docker compose up -d db
python -m src.main
```

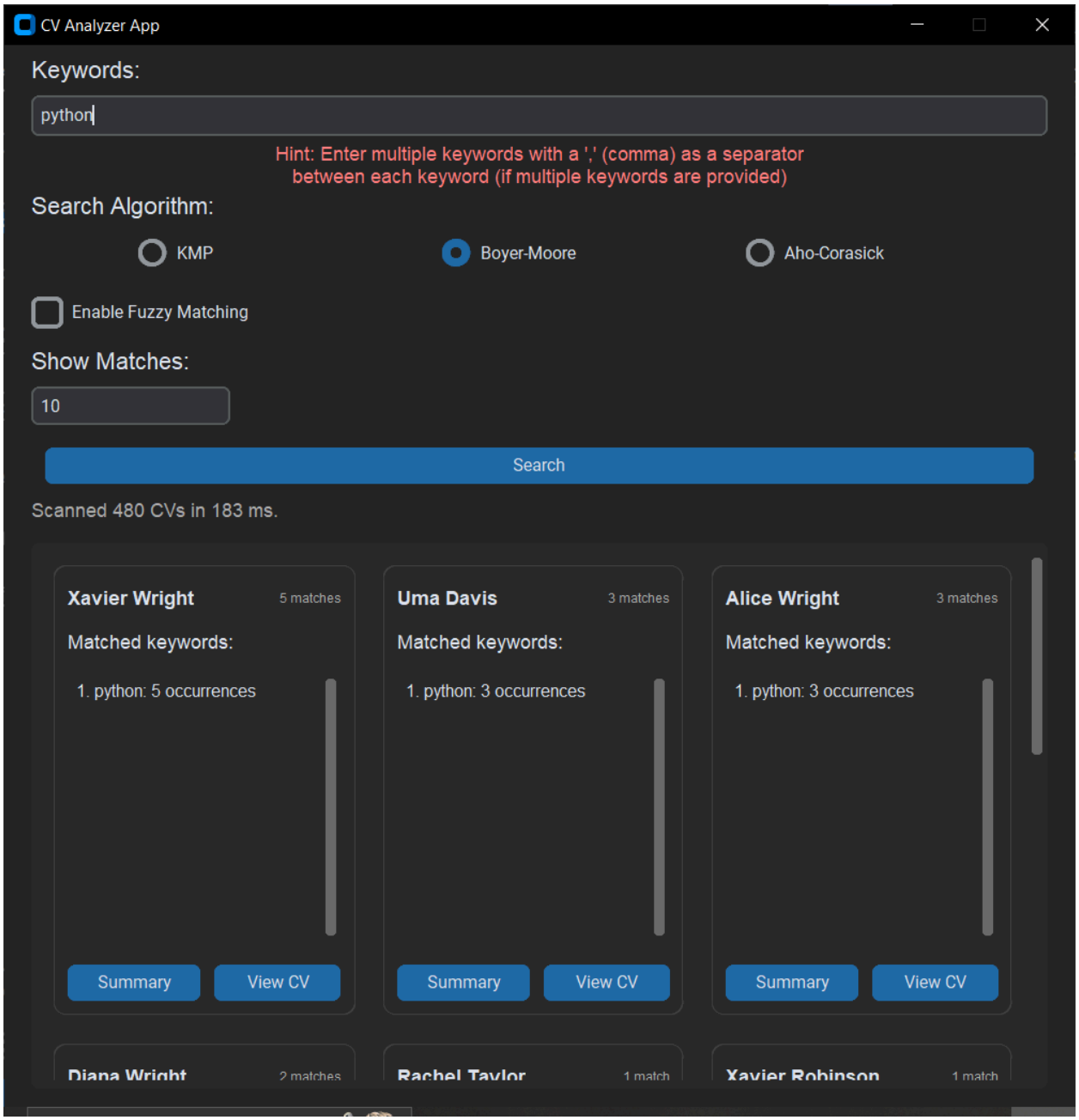
- Untuk menghentikan Docker:

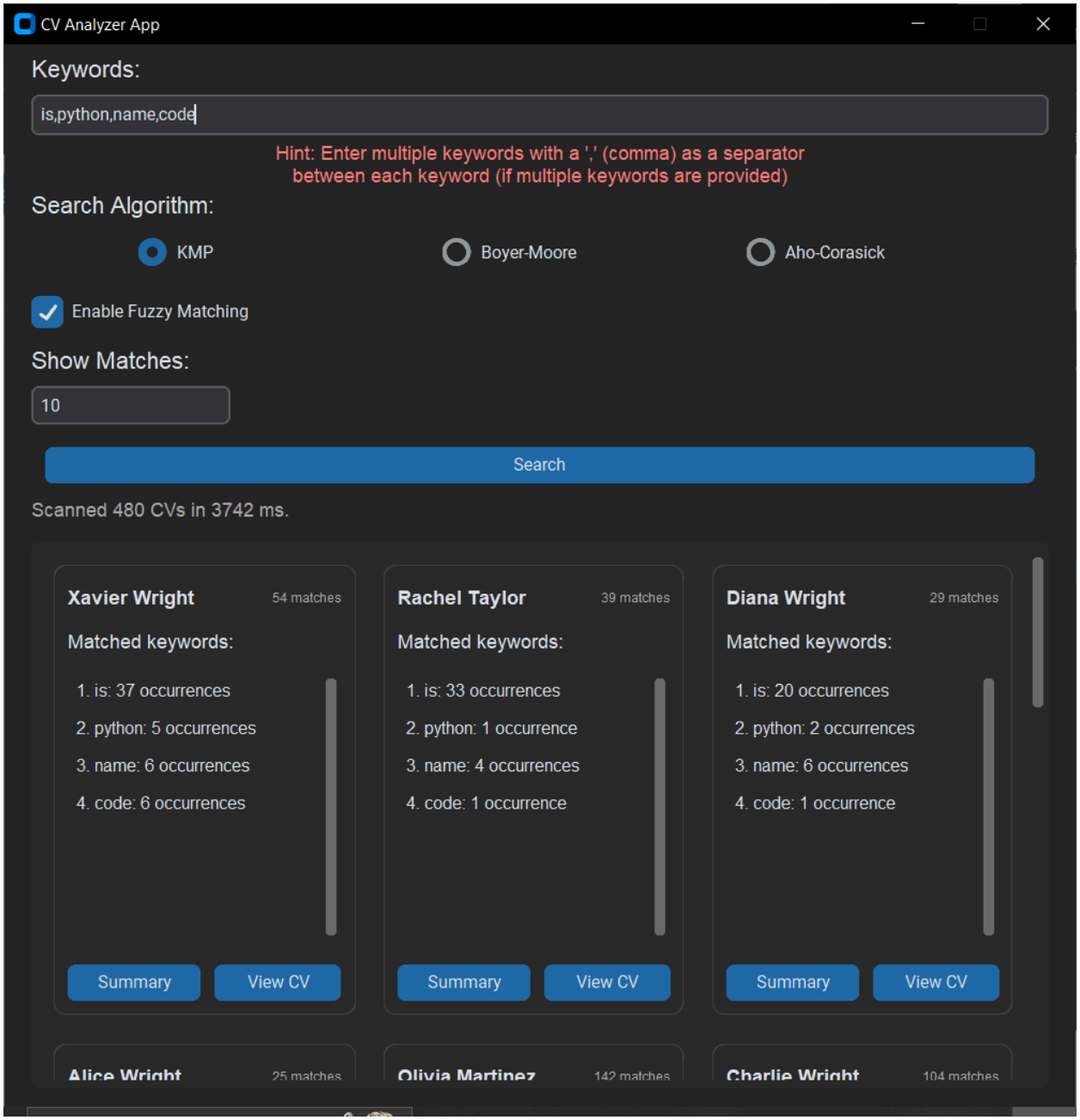
```
docker compose down
```

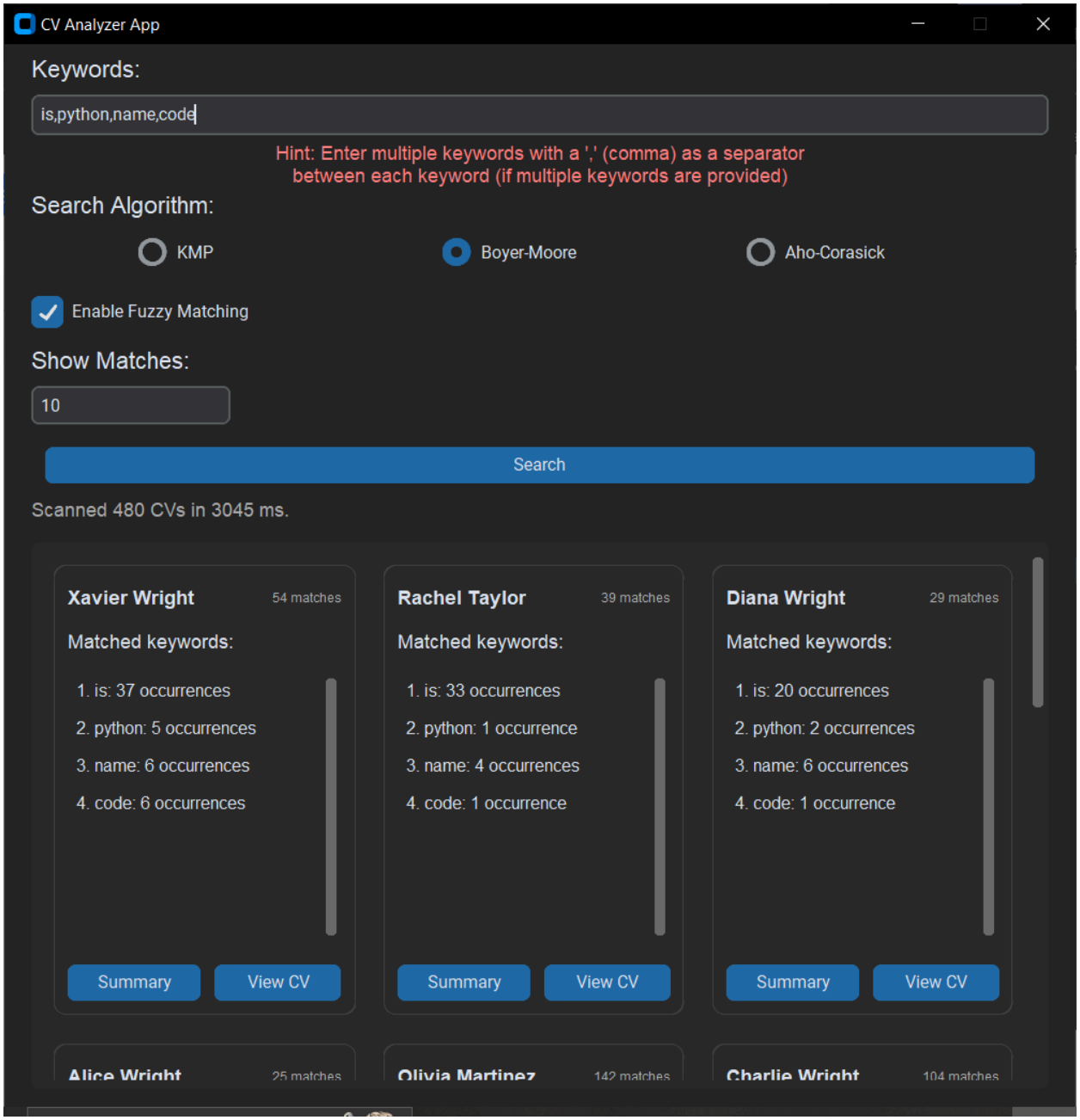
3. Menggunakan Aplikasi

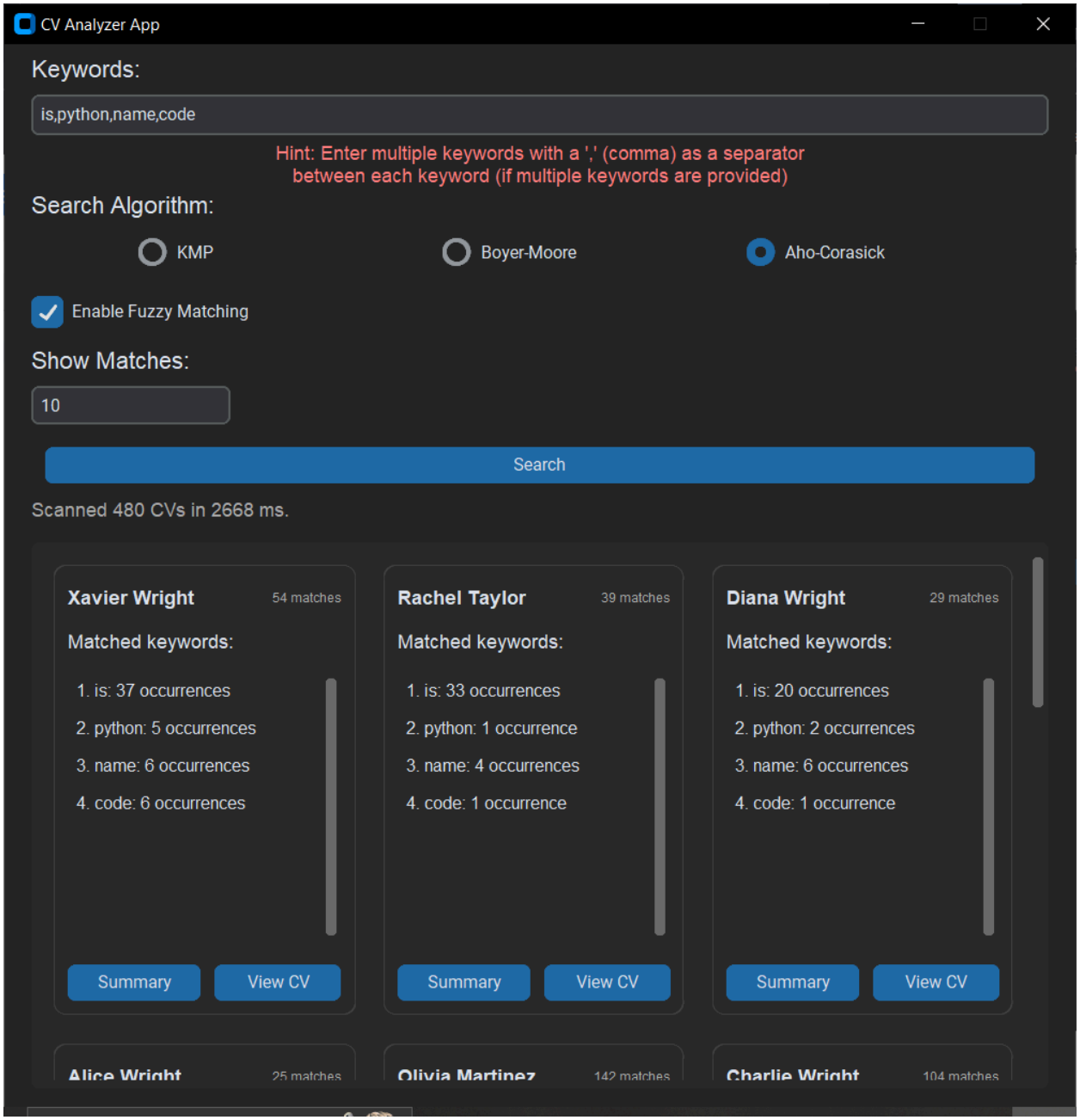
- Setelah program berjalan, pengguna memasukkan kata kunci pada kolom “Keywords”.
- Pilih algoritma pencocokan di antara “KMP”, “Boyer-Moore”, atau “Aho-Corasick”.
- Tentukan berapa banyak hasil yang ingin ditampilkan (misalnya 10).
- Klik tombol Search untuk menjalankan pencarian.
- Hasil berupa daftar CV dengan skor tertinggi akan muncul dalam bentuk kartu. Masing-masing kartu memiliki tombol Summary untuk melihat ringkasan serta View CV untuk membuka file PDF CV.

4.3. Hasil Pengujian




CV Analyzer App
Keywords:
is,python,name,code
Hint: Enter multiple keywords with a ',' (comma) as a separator
between each keyword (if multiple keywords are provided)
Search Algorithm:
KMP
Boyer-Moore
Aho-Corasick
Enable Fuzzy Matching
Show Matches:
10
Search
Scanned 480 CVs in 3742 ms.
Xavier Wright
54 matches
Matched keywords:
1. is: 37 occurrences
2. python: 5 occurrences
3. name: 6 occurrences
4. code: 6 occurrences
Summary
View CV
Rachel Taylor
39 matches
Matched keywords:
1. is: 33 occurrences
2. python: 1 occurrence
3. name: 4 occurrences
4. code: 1 occurrence
Summary
View CV
Diana Wright
29 matches
Matched keywords:
1. is: 20 occurrences
2. python: 2 occurrences
3. name: 6 occurrences
4. code: 1 occurrence
Summary
View CV
Alice Wright
25 matches
Olivia Martinez
142 matches
Charlie Wright
104 matches


CV Analyzer App
Keywords:
is,python,name,code
Hint: Enter multiple keywords with a ',' (comma) as a separator
between each keyword (if multiple keywords are provided)
Search Algorithm:
KMP
Boyer-Moore
Aho-Corasick
Enable Fuzzy Matching
Show Matches:
10
Search
Scanned 480 CVs in 3045 ms.
Xavier Wright
54 matches
Matched keywords:
1. is: 37 occurrences
2. python: 5 occurrences
3. name: 6 occurrences
4. code: 6 occurrences
Summary
View CV
Rachel Taylor
39 matches
Matched keywords:
1. is: 33 occurrences
2. python: 1 occurrence
3. name: 4 occurrences
4. code: 1 occurrence
Summary
View CV
Diana Wright
29 matches
Matched keywords:
1. is: 20 occurrences
2. python: 2 occurrences
3. name: 6 occurrences
4. code: 1 occurrence
Summary
View CV
Alice Wright
25 matches
Olivia Martinez
142 matches
Charlie Wright
104 matches



4.4. Analisis Hasil Pengujian

Berdasarkan hasil pengujian visual terhadap sistem ATS (applicant tracking system) , dapat disimpulkan bahwa sistem telah bekerja sesuai dengan spesifikasi pencocokan keyword

menggunakan berbagai algoritma yang berbeda. Pengujian dilakukan dengan berbagai konfigurasi keyword dan algoritma pencarian, serta melibatkan 480 CV dalam basis data.

Pada pengujian pertama, saat hanya memasukkan satu keyword yaitu python dan menggunakan algoritma Boyer-Moore tanpa fuzzy matching, sistem berhasil menampilkan hasil dalam waktu sangat cepat, yaitu 183 ms. Ini menunjukkan efisiensi Boyer-Moore pada pencarian single pattern, sesuai dengan sifat heuristiknya yang menghindari pencocokan karakter secara linear.

Pada pengujian kedua hingga keempat, sistem diuji dengan menggunakan fitur fuzzy matching dengan pengetesan menggunakan keyword is,python, name, code. Pengujian dilakukan dengan menggunakan tiga algoritma yang berbeda untuk mendapatkan , yakni KMP, Boyer-Moore, dan Aho-Corasick. Waktu eksekusi yang dihasilkan adalah:

- KMP: 3742 ms
- Boyer-Moore: 3045 ms
- Aho-Corasick: 2668 ms

Perbandingan antara KMP dengan BM, waktu eksekusi keduanya mirip untuk dataset kecil. Sedangkan BM sedikit lebih cepat saat pola panjang atau teks besar sesuai teori.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan hasil pengerjaan dan proses pengembangan sistem ATS ini, kami semakin memahami bagaimana algoritma pencocokan string bekerja dalam konteks dunia nyata. Kami mempelajari penerapan berbagai strategi algoritma, khususnya algoritma Knuth-Morris-Pratt (KMP) dan Boyer-Moore, yang mampu mempercepat pencarian informasi pada data teks besar seperti CV digital. Selain itu, implementasi Levenshtein Distance memberi kami wawasan baru tentang bagaimana pendekatan fuzzy matching dapat meningkatkan fleksibilitas sistem dalam menghadapi input yang tidak sempurna. Proses ini memperdalam pemahaman kami tentang pemilihan dan penerapan strategi algoritma yang efisien untuk membangun sistem pencarian yang responsif dan relevan dalam proses rekrutmen digital.

5.2. Saran

Pada program kami, sistem ATS ini dapat dikembangkan menjadi aplikasi berbasis web menggunakan framework seperti Flask atau Django agar lebih mudah diakses dan mendukung kolaborasi lintas perangkat. Dari sisi performa, algoritma Levenshtein Distance sebagai pencocokan fuzzy dapat dioptimalkan dengan mempertimbangkan penggunaan pustaka yang lebih efisien seperti RapidFuzz, terutama saat menangani dataset yang besar. Selain itu, disarankan menambahkan fitur visualisasi hasil pencarian, seperti penyorotan keyword atau grafik frekuensi kemunculan, untuk meningkatkan pengalaman pengguna. Untuk menjaga keandalan sistem, pengelolaan error dan fitur logging juga perlu diintegrasikan agar proses debugging dan pemeliharaan lebih terstruktur.

6. Lampiran

Tautan GitHub

Tautan Video YouTube

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	✓	

7. Daftar Pustaka

Lesson: Regular Expressions. (n.d.). Oracle. Retrieved from https://docs.oracle.com/javase/tutorial/essential/regex/

Jurafsky, D., & Martin, J. H. (n.d.). *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Chapter 2).

Sedgewick, R. (1992). *Algorithms in C++* (Chapter 19, String Searching). Addison-Wesley.