

## BAB 3 METODOLOGI PENELITIAN

### 3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini melibatkan beberapa tahapan. Tahap pertama adalah studi literatur, yang melibatkan pencarian, pembacaan, analisis, dan evaluasi sumber-sumber ilmiah yang relevan tentang implementasi algoritma *Boyer-Moore*. Studi literatur ini membantu dalam mengidentifikasi penelitian, merumuskan hipotesis, memilih metode yang sesuai, dan menyusun kerangka penelitian. Tahap kedua adalah pengumpulan data, yang melibatkan pencarian dataset terkait yang dapat digunakan dalam penelitian. Data yang digunakan dalam penelitian ini adalah data lowongan pekerjaan yang diambil dari *kaggle* dengan nama *Job Dataset*. Tahap ketiga adalah analisis kebutuhan, yang melibatkan identifikasi dan definisi kebutuhan pengguna dan sistem. Tahap keempat adalah perancangan, yang melibatkan pembuatan rancangan sistem, penentuan kebutuhan pengguna dan sistem, pembuatan sitemap sistem, dan pembuatan *low-fidelity prototype*. Tahap kelima adalah implementasi, yang melibatkan realisasi rancangan sistem menjadi sistem yang siap digunakan. Tahap keenam adalah pengujian, yang melibatkan pengukuran ketepatan sistem yang telah dikembangkan. Tahap terakhir adalah evaluasi, yang melibatkan penilaian sejauh mana sistem yang telah dikembangkan memenuhi spesifikasi, kebutuhan, dan kriteria keberhasilan yang telah ditetapkan.

### 3.2 Studi Literatur

Studi literatur bertujuan untuk mengetahui perkembangan pengetahuan, teori, konsep, metode, dan temuan-temuan terkini yang berkaitan dengan masalah yang ingin diteliti. Studi literatur meliputi mencari, membaca, menganalisis, dan mengevaluasi sumber-sumber ilmiah yang relevan tentang implementasi algoritma *Boyer-Moore* terhadap sebuah sistem. Studi literatur juga dapat membantu untuk mengidentifikasi penelitian, merumuskan hipotesis, memilih metode yang sesuai, dan menyusun kerangka penelitian. Adapun referensi penelitian terdahulu tentang implementasi algoritma *Boyer-Moore* yang diambil dari sebuah jurnal yang ditulis oleh Sara Nasr [32].

Pada penelitian terdahulu tentang proses pencarian menggunakan algoritma

*Boyer Moore* di perpustakaan digital. Penelitian ini menerapkan algoritma pencarian string *Boyer-Moore* untuk mencari buku secara efektif dalam perpustakaan digital dan membandingkannya dengan algoritma *Horspool* untuk membuktikan efisiensi kinerja algoritma *Boyer-Moore* dalam proses pencarian. Dengan menggunakan algoritma *Boyer-Moore*, pengguna dapat mencari buku yang diinginkan dengan mengetik *sub-string* dari judul buku yang cocok dengan bagian dari judul buku asli. Dengan diimplementasikannya algoritma *Boyer-Moore* dapat membantu meningkatkan efektivitas pencarian di perpustakaan digital, memudahkan siswa dan pembaca dalam menemukan buku yang diinginkan [33].

Pada penelitian terdahulu lainnya yang membahas tentang implementasi algoritma *Boyer-Moore* dalam kamus istilah komputer dan informatika berbasis web. Penelitian ini dilakukan dengan menggunakan 1501 kata-kata istilah komputer dan informatika, dimana implementasi algoritma *Boyer-Moore* dalam menerjemahkan istilah-istilah tersebut memiliki waktu pemrosesan rata-rata sebesar 29,240106 milidetik, dan penggunaan memori rata-rata sebesar 0,392 megabita. Hasil ini menunjukkan bahwa algoritma *Boyer-Moore* dapat diimplementasikan secara efisien dalam sistem berbasis web untuk membantu pengguna dalam menerjemahkan dan memahami istilah-istilah teknis yang kompleks [34].

### 3.3 Pengumpulan Data

Pengumpulan data adalah proses mendapatkan data dari sumber-sumber yang relevan, baik primer maupun sekunder. Pengumpulan data harus dilakukan dengan cara yang valid, reliabel, dan etis. Pengumpulan data meliputi mencari dataset terkait yang dapat digunakan dalam penelitian. Dalam penelitian ini digunakan dataset untuk data lowongan pekerjaan. Data tersebut diambil dari *kaggle* dengan nama *Job Dataset* [35]. Dari dataset tersebut, data-data yang digunakan dalam penelitian ini adalah *company name, city, country, job title, role, salary, description, skills, dan requirements*.

### 3.4 Analisis Kebutuhan

Analisis kebutuhan adalah proses mengidentifikasi dan mendefinisikan apa yang diinginkan dan dibutuhkan oleh pengguna dari sistem yang akan dikembangkan. Analisis kebutuhan meliputi menentukan kebutuhan pengguna, menentukan spesifikasi sistem, dan menentukan kriteria keberhasilan produk.

### 3.4.1 User Requirement dan System Requirement

Ada beberapa kebutuhan pengguna yang diperlukan dalam implementasi algoritma *Boyer-Moore* terhadap sistem. Pertama, user dapat melihat seluruh data job posting. User dapat melakukan Scan CV dengan job spesifik yang dipilih. Lalu user dapat melakukan *matching data* CV nya dengan seluruh data (deskripsi pekerjaan dan *skills requirements*) untuk semua list pekerjaan yang ada. Setelah proses *matching* dilakukan, akan ditampilkan 5 *match percentage* tertinggi.

Ada beberapa kebutuhan sistem yang diperlukan dalam implementasi algoritma *Boyer-Moore* terhadap sistem. Pertama, sistem harus menyediakan fitur untuk melihat semua *data job posting*. Lalu sistem harus dapat melakukan pencocokan data CV user dengan semua data (deskripsi pekerjaan dan *skills requirements*) untuk semua daftar pekerjaan yang ada. Proses pencocokan dilakukan dengan mengekstraksi keyword menggunakan *YAKE (Yet Another Keyword Extractor)* dari (deskripsi pekerjaan dan *skills requirements*) yang ada, lalu diproses dengan algoritma *Boyer-Moore* untuk mendapatkan hasil kecocokan. Setelah proses pencocokan, sistem harus menampilkan hasil dari persentase pencocokan dengan pekerjaan yang dipilih dan menampilkan alternatif 5 *match percentage* tertinggi.

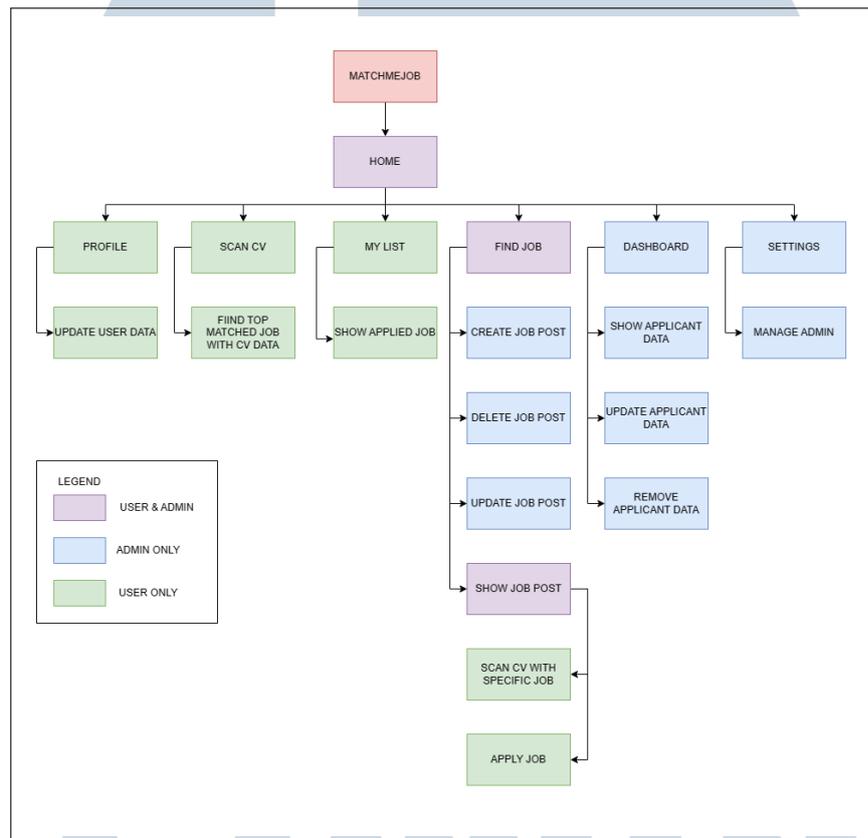
### 3.5 Perancangan

Perancangan adalah proses membuat rencana atau sketsa tentang fungsi, fitur, dan teknologi yang digunakan dalam sistem yang akan dikembangkan. Perancangan harus dilakukan dengan cara yang kreatif, inovatif, dan sesuai dengan spesifikasi dan kebutuhan pengguna. Perancangan meliputi membuat rancangan sistem, membuat *requirement* untuk pengguna dan sistem, membuat *sitemap* sistem, *flowchart* sistem, serta *low-fidelity prototype*.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### 3.5.1 Sitemap Diagram

Sistem yang dilakukan implementasi algoritma *Boyer-Moore* ini akan berbasis website. Maka dari itu dibutuhkan *sitemap* diagram. *Sitemap* diagram adalah diagram yang menunjukkan struktur halaman web. Berikut adalah sitemap diagramnya.

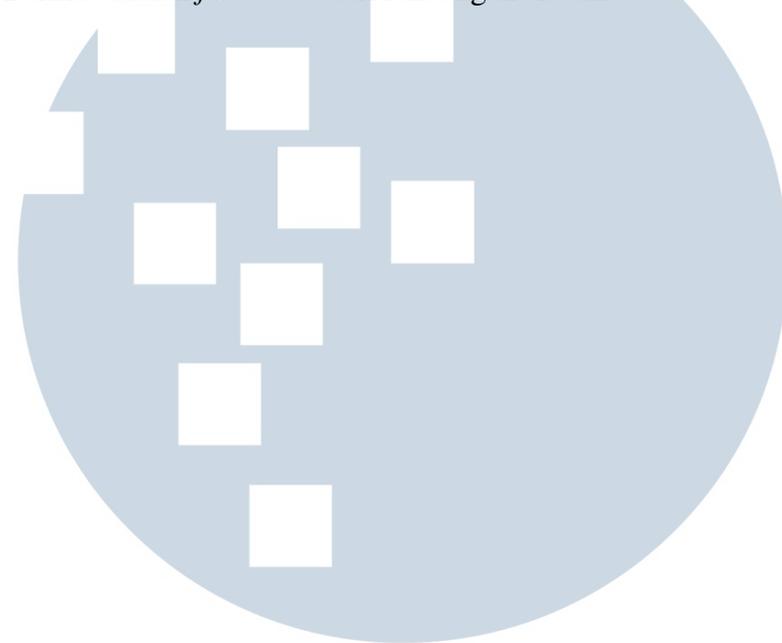


Gambar 3.1. *Sitemap* Diagram

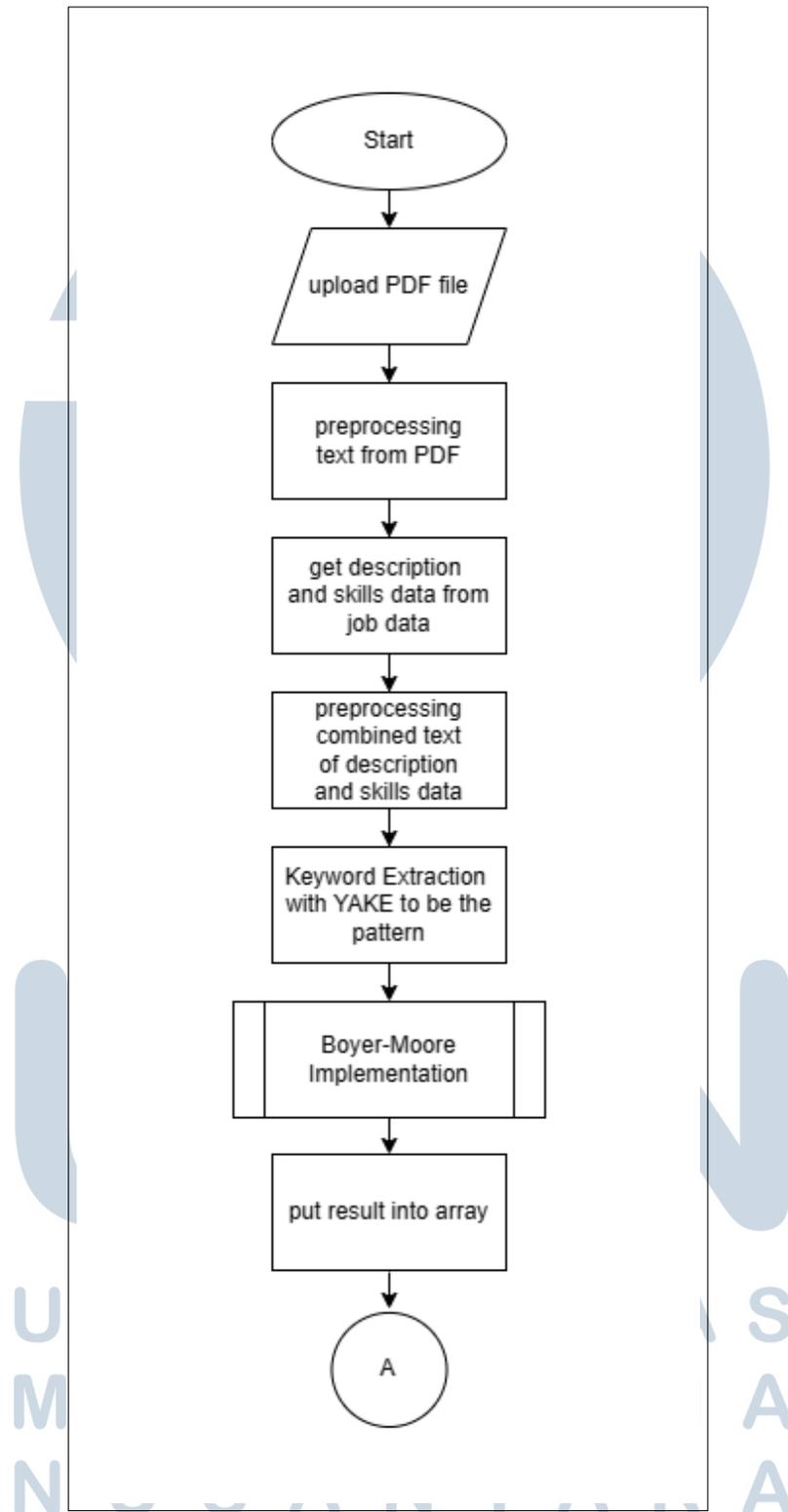
Pada gambar 3.1 *Sitemap* Diagram, dijelaskan bahwa website yang akan dibangun akan memiliki 7 halaman utama yaitu, *index*, *profile*, *scan CV*, *My List*, *Job List*, *Dashboard*, dan *Settings*. Dalam website ini juga terdapat 2 role yaitu pengguna dan admin. Pengguna hanya dapat mengakses halaman *index*, *profile*, *scan CV*, *My List*, serta *Job List*. Admin dapat mengakses halaman *Job List*, *Dashboard*, serta *Settings*. Pada gambar 3.1 *Sitemap* Diagram juga terdapat Legend yang menunjukkan setiap warna pada sitemap memiliki hak akses yang berbeda. Site yang berwarna ungu artinya dapat diakses oleh pengguna dan admin. Site yang berwarna biru hanya dapat diakses oleh admin. Site yang berwarna hijau hanya dapat diakses oleh pengguna.

### 3.5.2 Flowchart Diagram

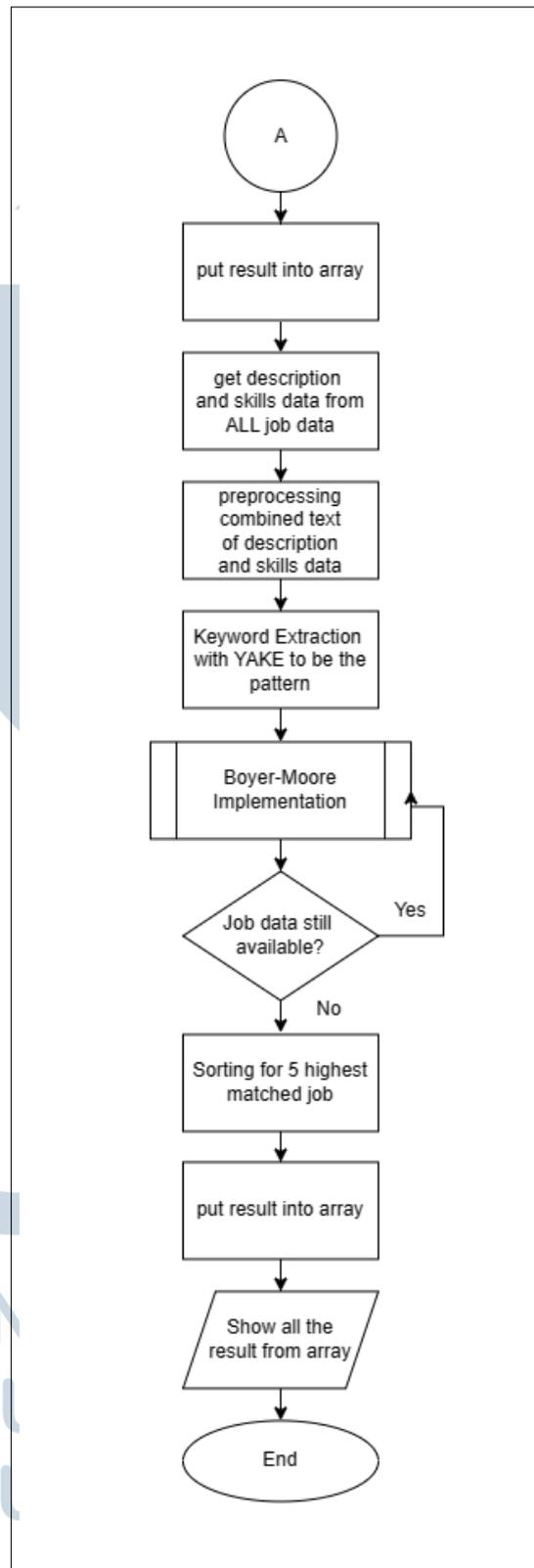
Dalam melakukan perancangan, terdapat tahapan-tahapan dari proses yang dilakukan oleh sistem. Proses tersebut akan didefinisikan dengan menggunakan *flowchart*. Berikut adalah *flowchart* dari rancangan sistem.



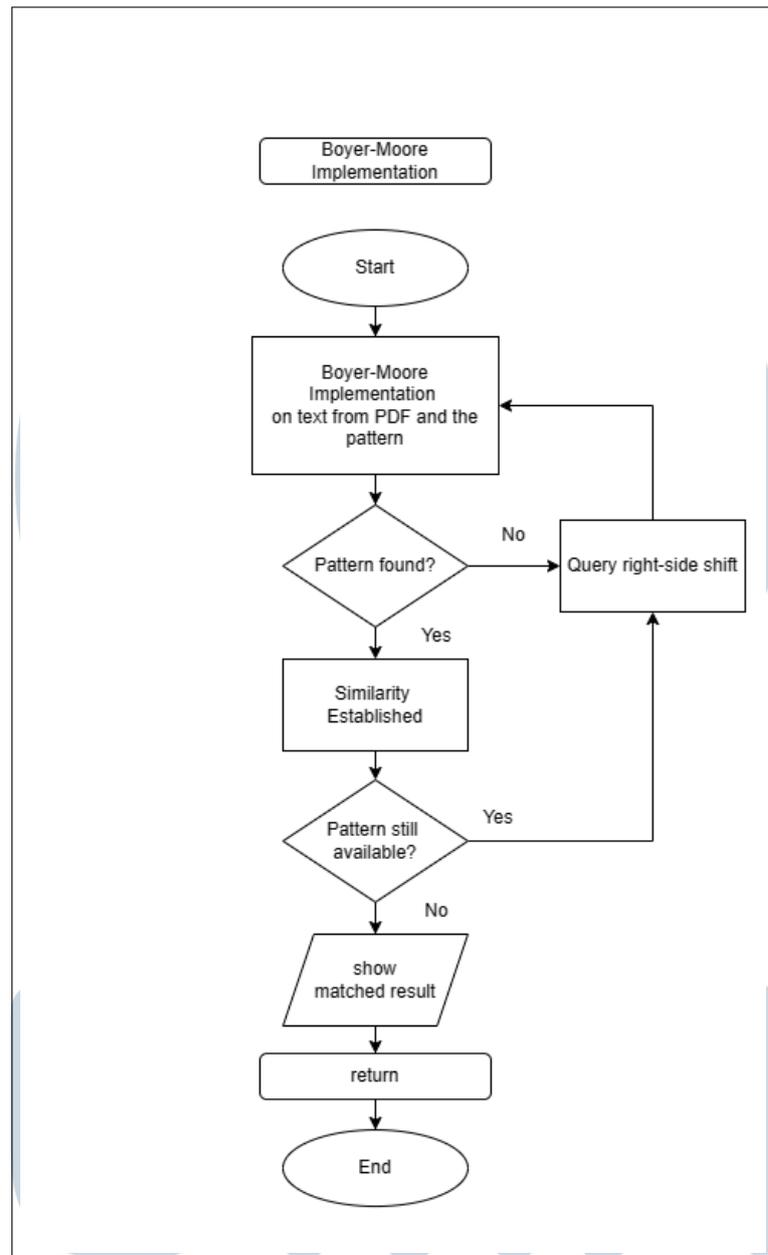
UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.2. *Flowchart* Rancangan Sistem



Gambar 3.3. Lanjutan *Flowchart* Rancangan Sistem



Gambar 3.4. Predefined Process Implementasi *Boyer-Moore*

Proses yang dilakukan dalam *flowchart* rancangan sistem diatas adalah pertama dengan meminta *input* dari pengguna berupa file dengan format PDF. Lalu text dalam PDF akan di ekstraksi dengan dilakukan proses tokenisasi menjadi *string token* dan dilakukan *pre-processing*. Dalam tahap *pre-processing*, dilakukan penghapusan dari *string* terhadap *string non-alphanumeric* dan text diubah menjadi *uppercase*. Selanjutnya, berdasarkan data pekerjaan yang dipilih, digabungkan antara data deskripsi pekerjaan dengan data *skills*. Kemudian teks yang telah

digabungkan dilakukan proses tokenisasi menjadi *string token* dan dilakukan *pre-processing* juga.

Untuk melakukan proses *string matching* dengan menggunakan algoritma *Boyer-Moore* dibutuhkan *text* dan *pattern*. *Text* diambil dari data yang di ekstraksi dan telah dilakukan *pre-processing* dari PDF. *Pattern* diambil dari teks yang digabungkan antara data deskripsi pekerjaan dengan data *skills*. Maka dari itu, untuk mendapatkan *pattern* dilakukan *keyword extraction* dengan menggunakan YAKE. Setelah *text* dan *pattern* sudah didapat, proses *string matching* dengan algoritma *Boyer-Moore* berjalan. Jika mendapatkan kecocokan *pattern*, ditambahkan persentase kecocokan kedalam perhitungan. Jika tidak mendapat kecocokan, akan dilakukan *query right-side shift*. Dalam algoritma *Boyer-Moore*, *query right-side shift* adalah operasi untuk menggeser semua *pattern* ke arah kanan sebanyak 1 posisi. Sehingga *pattern* akan terus digeser hingga terjadi kecocokan *pattern*. Jika telah digeser hingga *text* habis namun belum mendapat kecocokan *pattern*, artinya tidak ada kecocokan antara *pattern* tersebut dengan teks nya. Proses ini akan terus dilakukan hingga *pattern* habis. Setelah proses *Boyer-Moore* selesai, jumlah kecocokan akan ditampilkan dalam bentuk persentase. Misalkan dari *keyword extraction* terdapat 20 *keyword*. Setelah dilakukan *matching*, hanya 15 saja yang cocok. Sehingga sistem akan menampilkan persentase kecocokan dengan angka 75%. Angka tersebut akan di simpan kedalam *array*.

Dalam implementasi algoritma *sorting* pada sistem rekomendasi yang dibangun, setelah proses pencocokan string menggunakan algoritma *Boyer-Moore* selesai, langkah selanjutnya adalah mengurutkan hasil kecocokan tersebut untuk menentukan rekomendasi terbaik. Proses sortasi ini penting karena memungkinkan sistem untuk menampilkan hasil yang paling relevan kepada pengguna. Proses sistem masih berlanjut untuk mendapatkan 5 rekomendasi lainnya. Sortasi dilakukan dengan mengambil persentase kecocokan yang dihasilkan oleh algoritma *Boyer-Moore* dan mengurutkannya dari yang tertinggi ke terendah. Sistem akan mengambil semua data lowongan pekerjaan dan dilakukan proses yang sama mulai dari penggabungan *text* hingga *pre-processing* sampai menjadi *pattern*. Lalu proses *Boyer-Moore* dijalankan untuk teks yang dilakukan *matching* dengan masing-masing lowongan pekerjaan. Hasil kecocokan juga akan dilakukan dalam persentase. Setelah itu, dilakukan *sorting* untuk mendapatkan 5 persentase tertinggi untuk menjadi 5 rekomendasi lainnya. Kemudian 5 hasil ini juga di simpan kedalam *array*. Setelah semua proses selesai, hasil dari *array* ditampilkan.

### 3.5.3 Pre-processing

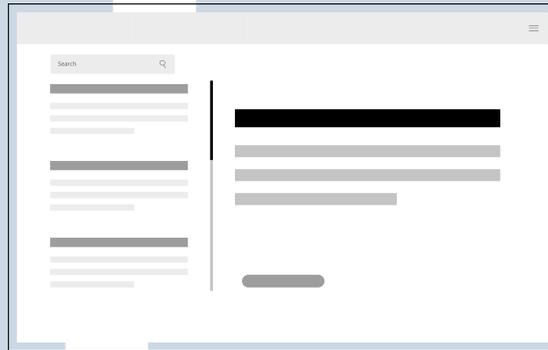
Proses *preprocessing* adalah tahap penting dalam pengolahan teks yang bertujuan untuk membersihkan dan menyiapkan data agar sesuai untuk analisis lebih lanjut. Proses preprocessing yang dilakukan adalah cleansing data, remove english stop words, case folding, dan tokenizing. Berikut adalah penjelasan singkat dari setiap langkah:

1. *Cleansing Data*: Langkah ini melibatkan penghapusan karakter non-alfanumerik yang tidak diperlukan, seperti tanda baca dan simbol khusus, yang dapat mengganggu proses analisis.
2. *Remove English Stop Words*: Ini adalah proses menghilangkan kata-kata umum dalam bahasa Inggris yang tidak menambahkan nilai signifikan ke dalam analisis teks. Kata-kata ini dikenal sebagai ‘stop words’ dan termasuk kata-kata seperti “the”, “is”, “in”, dan “on”. Penghapusan stop words membantu dalam meningkatkan efisiensi proses pencocokan string dan analisis teks dengan mengurangi jumlah token yang perlu diproses.
3. *Case Folding*: Proses ini mengubah semua teks menjadi huruf besar (uppercase) atau huruf kecil (lowercase) untuk memastikan konsistensi dan menghindari duplikasi selama pencocokan string karena perbedaan kapitalisasi.
4. *Tokenizing*: Tahap ini memecah teks menjadi unit-unit yang lebih kecil, biasanya kata-kata, yang disebut token. Ini memudahkan identifikasi dan analisis pola dalam teks.

Dengan mengintegrasikan semua langkah *preprocessing* ini, sistem akan dapat memproses teks dengan lebih efektif, memastikan bahwa data yang digunakan untuk pencocokan string dengan algoritma *Boyer-Moore* adalah data yang telah dioptimalkan untuk mendapatkan hasil yang akurat dan relevan.

### 3.5.4 Low-Fidelity Prototype

*Low-Fidelity Prototype* adalah prototype sederhana yang menunjukkan konsep desain awal. *Low-Fidelity Prototype* digunakan untuk menguji ide dengan cepat, mengidentifikasi kesenjangan dan kekurangan. Berikut adalah *Low-Fidelity Prototype* yang akan diimplementasikan kedalam sistem.



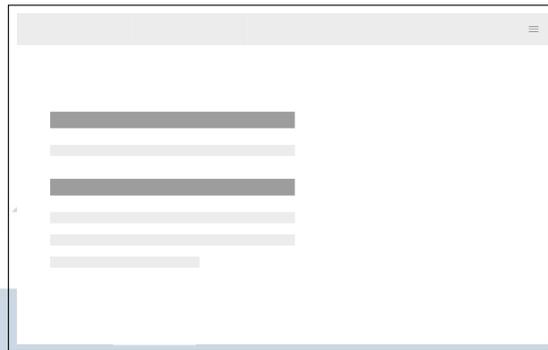
Gambar 3.5. *Prototype* Tampilan List Lowongan Pekerjaan

Pada gambar 3.5 *Prototype* Tampilan List Lowongan Pekerjaan, akan ditampilkan list lowongan pekerjaan disebelah kiri layar. Ketika sebuah pekerjaan di pilih, akan ditampilkan informasi detail dari pekerjaan tersebut. Pada bagian bawah, terdapat sebuah button yang akan *redirect* ke halaman untuk melakukan scan CV.



Gambar 3.6. *Prototype* Halaman Upload File PDF

Pada gambar 3.6 *Prototype* Halaman Upload File PDF, akan ditampilkan informasi mengenai pekerjaan yang dipilih. Dibawahnya terdapat sebuah kolom untuk *upload* file PDF. Lalu dibawah kolom tersebut terdapat sebuah button untuk melakukan proses pencocokan.



Gambar 3.7. *Prototype* Halaman Hasil

Pada gambar 3.7 *Prototype* Halaman Hasil, akan ditampilkan hasil dari pencocokan yang terdapat hasil pencocokan dengan lowongan pekerjaan yang dipilih, serta menampilkan 5 rekomendasi pekerjaan dengan persentase tertinggi. Hasil yang ditampilkan adalah nama perusahaan, lokasi perusahaan, serta hasil dari kecocokan antara CV dengan data lowongan pekerjaan. Rekomendasi pekerjaan lainnya ditampilkan dengan format yang sama seperti di atasnya. Rekomendasi pekerjaan ini ditampilkan berdasarkan persentase kecocokan dari yang paling tinggi, dengan tujuan untuk memberikan alternatif pilihan kepada penggunanya.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 3.6 Implementasi

Implementasi adalah proses merealisasikan rancangan sistem menjadi sistem yang siap digunakan. Implementasi harus dilakukan dengan cara yang akurat, efektif, dan efisien. Implementasi meliputi pengaplikasian algoritma *Boyer-Moore* ke dalam sistem lowongan pekerjaan dengan tujuan untuk melihat persentase hasil kesesuaian antara data dari CV yang diunggah dan data dari deskripsi pekerjaan dengan *skills requirement* dari sebuah lowongan pekerjaan.

### 3.7 Pengujian

Pengujian adalah proses mengukur ketepatan sistem yang telah dikembangkan. Pengujian harus dilakukan dengan cara yang objektif, sistematis, dan kritis. Salah satu teknik pengujian yang dapat digunakan adalah pengujian manual terhadap string matching menggunakan algoritma *Boyer-Moore*. Dengan demikian, pengujian dapat memastikan bahwa sistem bekerja seperti yang diharapkan.

### 3.8 Evaluasi

Evaluasi adalah proses menilai sejauh mana sistem yang telah dikembangkan memenuhi spesifikasi, kebutuhan, dan kriteria keberhasilan yang telah ditetapkan. Nilai evaluasi yang dihitung adalah kecepatan dari proses algoritma *Boyer-Moore* melakukan *string matching*. Evaluasi yang dilakukan adalah menguji kecepatan perhitungan dari hasil persentase kesesuaian data CV dengan seluruh data pekerjaan yang tersimpan dalam sistem mulai dari 100 data pekerjaan, 200 data pekerjaan, 500 data pekerjaan, hingga 1000 data pekerjaan yang digunakan dalam mencari rekomendasi tertinggi.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA