

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, pencarian kerja menjadi tantangan bagi banyak individu, terutama bagi lulusan baru. Salah satu kesulitan utama yang dihadapi adalah menemukan lowongan yang sesuai dengan kualifikasi mereka (Mendez & Bulanadi, 2020). Faktor lain yang memperumit pencarian kerja adalah tingginya persaingan di dunia kerja. Setiap tahun, jumlah lulusan baru terus meningkat, sehingga persaingan untuk mendapatkan pekerjaan semakin ketat (Puspasari et al., 2021).

Dalam menghadapi tantangan tersebut, berbagai *platform* pencarian kerja berbasis digital hadir sebagai solusi untuk membantu pencari kerja menemukan peluang yang sesuai dengan lebih efisien. *Platform* ini menyediakan fitur pencarian berdasarkan kategori pekerjaan, lokasi, hingga kisaran gaji untuk mempermudah pencocokan antara pencari kerja dan lowongan yang tersedia. Selain itu, beberapa *platform* juga dilengkapi dengan sistem rekomendasi yang bertujuan untuk menyajikan lowongan kerja yang relevan berdasarkan preferensi dan riwayat pencarian pengguna.

Salah satu *platform* pencarian kerja yang banyak digunakan adalah *JobStreet*. *JobStreet* merupakan portal lowongan kerja yang menyediakan berbagai informasi terkait lowongan pekerjaan di berbagai bidang industri, mulai dari tingkat *entry-level* hingga profesional. Dengan jutaan pengguna aktif, *JobStreet* memainkan peran penting dalam membantu pencari kerja mendapatkan pekerjaan yang sesuai dengan kualifikasi dan preferensi mereka.

Meskipun *JobStreet* telah menyediakan fitur pencarian berdasarkan kategori pekerjaan, lokasi, dan kisaran gaji, tampilan sistem sering kali membuat pengguna enggan menyaring banyak data yang disajikan dalam satu informasi pekerjaan. Akibatnya, pencari kerja sering kali harus menyaring banyak lowongan yang tidak relevan dan mengalami kesulitan dalam menemukan keterkaitan antara posisi yang mereka inginkan dengan peluang yang tersedia di pasar kerja (Kamaruddin et al., 2019).

Untuk mengatasi keterbatasan dalam sistem pencarian tersebut, diperlukan solusi yang dapat membantu pencari kerja memperoleh rekomendasi yang lebih relevan. Seiring dengan perkembangan teknologi, sistem rekomendasi berbasis data telah banyak diterapkan sebagai solusi untuk meningkatkan efisiensi pencarian informasi, termasuk dalam dunia kerja. Sistem rekomendasi adalah algoritma yang digunakan untuk memberikan saran berdasarkan preferensi pengguna dengan cara mengumpulkan dan menganalisis data untuk menghasilkan rekomendasi yang lebih akurat (Burr et al., 2018).

Namun, sering kali keefektifan sistem rekomendasi masih terhambat oleh keterbatasan dalam pemrosesan data yang beragam serta kurangnya pemahaman terhadap pola pencarian pengguna secara lebih mendalam. Akibatnya, rekomendasi yang diberikan tidak selalu sesuai dengan ekspektasi pencari kerja, sehingga mereka tetap harus melakukan pencarian tambahan untuk menemukan pekerjaan yang benar-benar relevan.

Salah satu metode yang dapat digunakan untuk meningkatkan efektivitas sistem rekomendasi dalam pencarian kerja adalah *clustering*. *Clustering* merupakan teknik dalam pembelajaran tanpa pengawasan yang digunakan untuk mengelompokkan data berdasarkan kesamaan pola tertentu (Santosa et al., 2023). *Clustering* dapat membantu mengelompokkan berbagai kategori pekerjaan berdasarkan faktor seperti jenis industri, lokasi.

Berbagai penelitian telah menerapkan metode *clustering* dalam sistem rekomendasi kerja. Namun, sebagian besar penelitian masih berfokus pada pengelompokan pekerjaan berdasarkan deskripsi teks atau kata kunci tanpa mempertimbangkan pola aplikasi kerja pengguna secara lebih spesifik. Misalnya, penelitian yang dilakukan oleh Zhang et al. (2021) menggunakan pendekatan *K-Means clustering* untuk mengelompokkan lowongan kerja berdasarkan teks deskripsi pekerjaan.

Meskipun metode ini dapat membantu dalam klasifikasi pekerjaan, hasil rekomendasi masih kurang optimal karena tidak mempertimbangkan pola aplikasi kerja individu. Selain itu, penelitian oleh Liu et al. (2022) mengembangkan sistem rekomendasi berbasis *clustering* dengan memanfaatkan untuk mengekstrak topik dari deskripsi pekerjaan, namun sistem tersebut masih bersifat umum dan kurang memperhitungkan preferensi pencari kerja di lokasi tertentu.

Penelitian ini hadir untuk membuat sebuah sistem rekomendasi kerja berbasis *K-Means Clustering* yang dirancang untuk menyajikan data secara langsung. Dengan pendekatan Metode *Elbow* untuk menentukan jumlah *cluster* yang optimal, sistem ini mengelompokkan pekerjaan berdasarkan jenis pekerjaan, lokasi, dan klasifikasi. Selain itu, sistem ini juga dilengkapi dengan fitur tambahan, seperti informasi jumlah pelamar pada tiap lowongan kerja, pelacakan status lamaran untuk mengetahui apakah suatu pekerjaan sudah dilamar atau belum, serta persentase kecocokan pelamar terhadap suatu *cluster* pekerjaan. Fitur pelacakan lamaran ini memberikan manfaat signifikan bagi pencari kerja, seperti membantu mereka mengelola riwayat lamaran dengan lebih terstruktur, serta menghindari pengiriman lamaran ganda. Selain itu, pengguna dapat melihat jumlah pesaing yang telah menyimpan atau melamar pekerjaan yang sama, sehingga memberikan wawasan lebih dalam dalam proses pengambilan keputusan dan strategi mencari pekerjaan.

1.2 Rumusan Masalah

Berdasarkan identifikasi masalah diatas, rumusan masalah yang akan dibahas yaitu :

- a. Bagaimana membuat *clustering* yang optimal dari data lowongan kerja dengan *algoritma k-means clustering* yang sesuai dengan kebutuhan aplikasi?
- b. Bagaimana merancang *dashboard* sistem rekomendasi yang efektif dalam menemukan lowongan kerja?

1.3 Batasan Masalah

- a. Fokus dari penelitian ini hanya untuk mengukur akurasi dari rekomendasi pekerjaan hasil *clustering* menggunakan algoritma k-means tanpa mempertimbangkan algoritma lain.
- b. Mengevaluasi pola jenis pekerjaan hanya berdasarkan hasil *clustering*.
- c. Hasil dari rekomendasi pekerjaan yang telah *diclustering* hanya berlaku untuk data dari *webscrapping* yang dilakukan pada platform *JobStreet*.

1.4 Tujuan

Adapun tujuan dari penelitian ini yaitu :

- a. Menganalisis pola pencarian kerja pengguna pada *JobStreet* dengan melihat fitur-fitur seperti kategori pekerjaan, lokasi, dan rentang gaji.
- b. Memberikan informasi yang berguna bagi pengguna dalam merancang strategi yang lebih tepat berdasarkan proses hasil *clustering*.

1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

- a. Membantu memahami pola dan preferensi pencari kerja, yang dapat memberikan wawasan lebih dalam mengenai faktor-faktor yang memengaruhi keputusan pencarian pekerjaan.

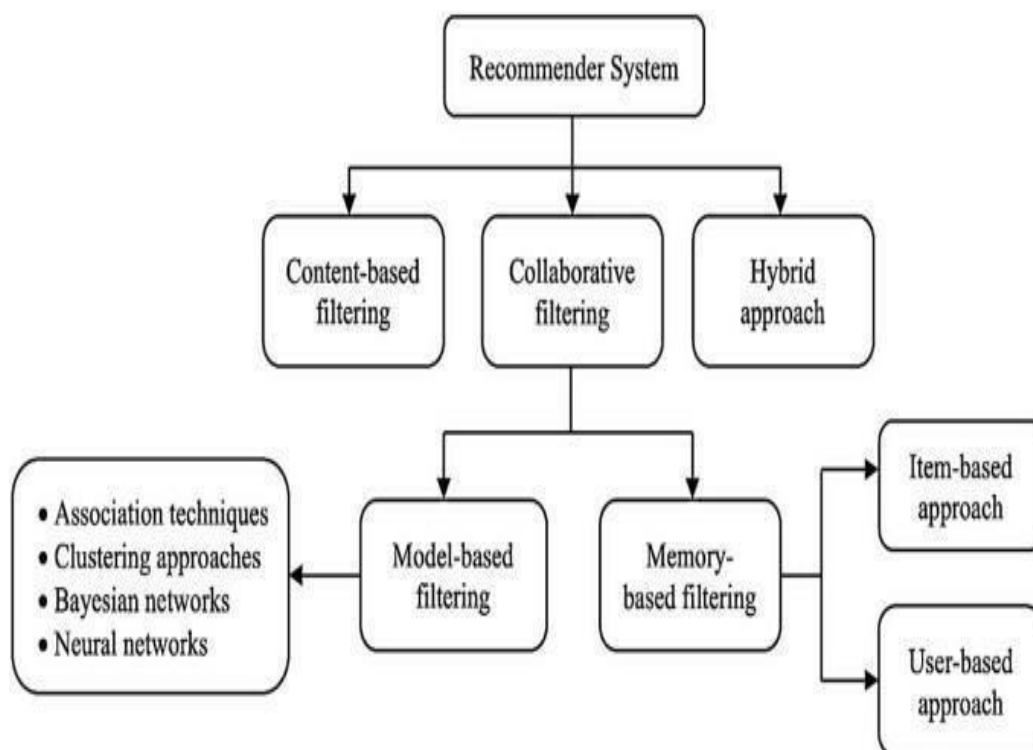
- b. Meningkatkan relevansi rekomendasi pekerjaan agar lebih sesuai dan relevan bagi pengguna.
- c. Berkontribusi pada pengembangan analisis data terutama dalam penggunaan algoritma *k-means clustering*.

1.6 Landasan Teori

1.6.1 Sistem rekomendasi

Jumlah pencari kerja meningkat seiring waktu, setiap pekerjaan menerima sejumlah lamaran, dan di antaranya, banyak yang relevan dengan posisi yang disebutkan. Hal ini menimbulkan masalah besar bagi perekrut pekerjaan karena mereka harus memilih profil/resume yang paling memenuhi syarat dari kumpulan resume, dimana kadang ada yang mendaftar pekerjaan tanpa memperhatikan kualifikasi yang dibutuhkan. Proses pencocokan resume kandidat dengan deskripsi pekerjaan akan memakan waktu yang cukup lama jika terjadi hal seperti itu (Elsevier B.V., 2020).

Sistem rekomendasi telah muncul untuk mengatasi masalah ini, dengan tujuan membantu pengguna menemukan apa yang benar-benar relevan dengan kebutuhan mereka. Sistem rekomendasi adalah perangkat lunak untuk membantu pengguna dalam proses pengambilan keputusan dengan menerapkan penyaringan informasi, penggalan data, dan algoritma prediksi (Urdaneta-Ponte et al., 2021). Sistem rekomendasi merupakan alat yang efisien untuk menyaring informasi daring, yang tersebar luas karena kebiasaan pengguna komputer yang berubah, tren personalisasi, dan akses internet yang semakin berkembang. Meskipun sistem rekomendasi terkini unggul dalam memberikan rekomendasi yang tepat, sistem ini memiliki berbagai keterbatasan dan tantangan seperti skalabilitas, *cold-start*, kelangkaan, dll (Roy & Dutta, 2022).



Gambar 1. Klasifikasi metode dalam sistem rekomendasi (Roy & Dutta, 2022).

1.6.2 Flask API

Flask adalah pustaka aplikasi web standar untuk *Python*. *Flask* menyediakan antarmuka berorientasi objek yang canggih ke server web. *Flask* banyak digunakan dalam pengembangan sistem berbasis web karena kemampuannya dalam menangani permintaan HTTP, menyajikan data dalam format yang fleksibel, serta integrasi yang mudah dengan berbagai pustaka data dan *machine learning* (Wang et al., 2020). *Flask* juga telah dikembangkan sebagai API untuk mendukung komunikasi antara aplikasi yang digunakan dengan model prediksi. Sebagai API, *flask* memainkan peranan penting untuk menyediakan *endpoint* dalam sistem pencarian kerja guna untuk mengakses prediksi dan rekomendasi pekerjaan.

1.6.3 React js

React adalah *framework* web yang memiliki fitur yang lebih baik dibandingkan dengan *framework* sejenis lainnya seperti *Angular*, *Vue*. Hal ini dikarenakan implementasi Virtual DOM yang bertujuan untuk meningkatkan kemampuan aplikasi secara keseluruhan (Gowda & Rangaswamy, 2020). *React js* unggul dalam arsitektur berbasis komponen, yang mendukung penggunaan ulang dan pemeliharaan. *DOM* virtual *React* secara efisien memperbarui *UI*, meningkatkan kinerja dalam pengembangan aplikasi web (Garg et al., 2023). Penyediaan fitur-fitur penting yang dilakukan oleh aplikasi yang menggunakan *React js* dapat memungkinkan penyederhanaan proyek secara keseluruhan, dimana sangat berguna untuk keefisienan dalam produktivitas aplikasi (Nalbantov et al., 2023).

1.6.4 Unified Modeling Language

UML merupakan singkatan dari *unified modeling language*. *UML* merupakan bahasa pemodelan serba guna dalam bidang rekayasa perangkat lunak yang dirancang untuk menyediakan cara standar untuk memvisualisasikan desain sistem perangkat lunak seperti yang terlihat pada Tabel 1. Diagram *UML* digunakan untuk menggambarkan struktur arsitektur, dinamis, dan statis suatu sistem perangkat lunak (Modi et al., 2021).

Tabel 1. Penilaian alat dari UML (Modi et al., 2021)

No	Tools	Supportive range of diagram	wide UML	Export/import XML file format	Open Source	Java code Generator	Reverse Engineering
1	<i>Agro UML</i>	1		0	1	1	0
2	<i>Modelio</i>	1		1	1	1	1
3	<i>Star UML</i>	1		1	1	1	1
4	<i>Rational Rose</i>	1		0	0	1	0

Dalam Tabel 1, angka 1 dan 0 digunakan untuk menunjukkan apakah sebuah fitur tersedia atau tidak dalam masing-masing alat *UML*.

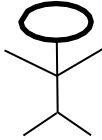
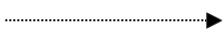
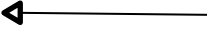
- 1 berarti fitur tersebut tersedia dalam alat *UML* yang bersangkutan.
- 0 berarti fitur tersebut tidak tersedia dalam alat *UML* yang bersangkutan.

1.6.5 Use Case Diagram

Use case diagram merupakan diagram perilaku dalam *Unified Modeling Language (UML)*. Use case diagram menggambarkan persyaratan fungsional perangkat lunak. Use case diagram dapat digunakan untuk memahami cara kerja sistem. Pemodelan Use Case menyediakan representasi grafis dari persyaratan sistem perangkat lunak. Ini adalah alat yang berguna untuk penggalian persyaratan. Elemen kunci dalam model use case adalah aktor (entitas eksternal), dan use case itu sendiri. Use case adalah unit layanan atau fungsionalitas (persyaratan), dalam sistem.

Pemodelan use case juga secara efektif telah menjadi teknik analisis yang dapat dipraktikkan dengan terus menyempurnakan pendekatan analisis sistem ini hingga saat ini, dan kini telah diformalkan sebagai bagian dari UML, meskipun, pemodelan use case tidak jauh berbeda dalam strategi dan tujuannya dari teknik sebelumnya seperti analisis sudut pandang terstruktur (Ahmed & Daleel, 2020). Elemen-elemen utama dalam use case diagram, seperti aktor, use case, *dependency*, *generalization*, *include*, *extend*, *association*, sistem, *collaboration*, dan note, dijelaskan lebih rinci pada Tabel 2.

Tabel 2. Komponen pada use case diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	Aktor adalah entitas (bisa berupa pengguna, sistem lain, atau perangkat keras) yang berinteraksi dengan sistem. Aktor digambarkan sebagai figur <i>stickman</i> dalam Use Case Diagram.
	<i>Include</i>	<i>Include</i> digunakan ketika suatu use case selalu membutuhkan use case lain untuk berjalan.
	<i>Extend</i>	<i>Extend</i> digunakan untuk menunjukkan bahwa suatu use case dapat memperluas perilaku use case lain jika kondisi tertentu terpenuhi.
	<i>System</i>	Sistem merepresentasikan batas dari aplikasi atau layanan yang dikembangkan, biasanya digambarkan sebagai kotak besar yang berisi use case.
	<i>Use Case</i>	Use Case merepresentasikan suatu fungsi atau proses yang dilakukan dalam sistem, biasanya digambarkan sebagai oval.

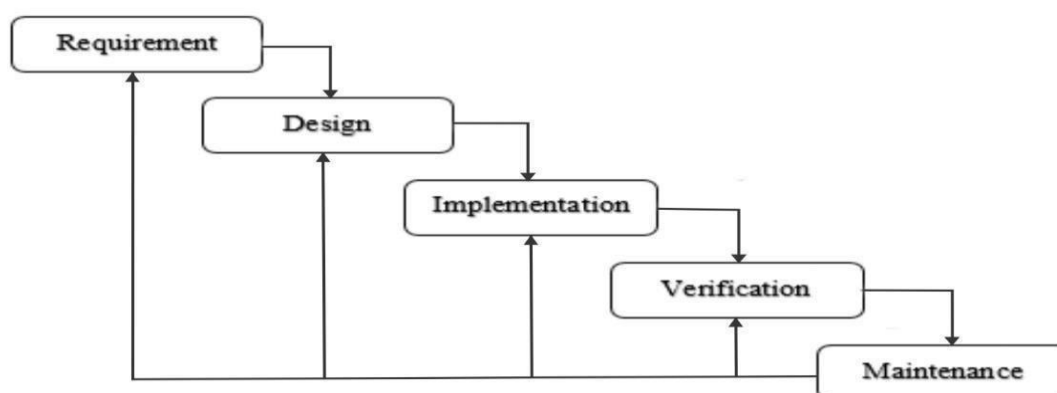
1.6.6 JobStreet

JobStreet adalah platform pencarian kerja yang populer di Asia Tenggara, yang menyediakan layanan untuk menghubungkan pencari kerja dengan perusahaan yang mencari karyawan. Dengan antarmuka yang ramah pengguna, *JobStreet* memungkinkan pengguna untuk mencari lowongan pekerjaan berdasarkan kriteria tertentu seperti lokasi, industri, dan jenis pekerjaan. *JobStreet* juga menawarkan fitur untuk mengunggah resume, melamar pekerjaan secara langsung, dan mendapatkan rekomendasi pekerjaan yang sesuai dengan kualifikasi pengguna.

JobStreet berfungsi sebagai jembatan antara pencari kerja dan perusahaan, memfasilitasi proses rekrutmen dengan menyediakan informasi yang relevan dan terkini tentang lowongan pekerjaan. Platform ini juga dilengkapi dengan alat analisis yang membantu pengguna memahami tren pasar kerja, gaji, dan keterampilan yang dibutuhkan dalam industri tertentu. Dengan demikian, *JobStreet* tidak hanya berperan sebagai situs pencarian kerja, tetapi juga sebagai sumber informasi yang berharga bagi pencari kerja dan perusahaan.

1.6.7 Metode Waterfall

Metode *waterfall* adalah salah satu metode yang dapat ditempuh dalam menerapkan tahapan pengembangan sistem informasi (Cahyono et al., 2022). Studi-studi terdahulu menunjukkan bahwa sistem informasi yang dikembangkan menggunakan model *waterfall* berhasil karena telah dievaluasi dan menghasilkan hasil yang dapat diandalkan. Dengan menggunakan model *waterfall*, pengembangan perangkat lunak dilakukan secara bertahap, dimulai dari analisis kebutuhan, desain sistem, implementasi, pengujian, penyebaran, dan pemeliharaan (Amron et al., 2022). Dalam metode *waterfall* terdapat lima tahapan yang saling terkait mempengaruhi satu dengan yang lain seperti yang terlihat pada Gambar 2.



Gambar 2. Tahapan dari metode *waterfall*

1.6.8 Clustering

Clustering adalah proses pengelompokan data berdasarkan kesamaan propertinya. *Clustering* dalam prosesnya mengkategorikan sekumpulan data ke dalam kelompok-kelompok yang serupa (*cluster*), dan elemen-elemen dalam setiap *cluster* memiliki kesamaan, di mana kesamaan antar elemen dalam *cluster* yang sama harus cukup kecil dibandingkan dengan kesamaan antar elemen *cluster* yang berbeda (Ghazal et al., 2021).

Terdapat tingkat kesamaan yang tinggi dalam setiap klaster data yang diklasifikasikan oleh algoritma *clustering*, dan kesamaan antara berbagai klaster data relatif besar. Algoritma *clustering* biasanya mencakup metode hierarkis, metode berbasis kepadatan, dan metode partisi. Dari metode tersebut, algoritma *clustering* kebanyakan dipakai dalam berbagai dataset yang mencakup jumlah yang banyak, seperti pada aplikasi rekomendasi pekerjaan (Cui, 2020). Hal ini disebabkan oleh kemampuan algoritma *clustering* untuk menangani data yang kompleks dan tidak berlabel, sehingga dapat mengidentifikasi pola tersembunyi di dalam data.

1.6.9 K-Means Clustering

K-means adalah prosedur iteratif yang membagi N objek menjadi K *cluster* yang terpisah. *K-means* mungkin merupakan metode pengelompokan yang paling banyak digunakan, dan khususnya yang paling terkenal dari metode pengelompokan berbasis partisi yang menggunakan *centroid* untuk presentasi *cluster*. Kualitas pengelompokan *k-means* diukur melalui kriteria kesalahan kuadrat dalam *cluster*. Algoritma *K-means* digunakan untuk meminimalkan masalah, dan *k-means* memiliki banyak varian yang akan dibahas selanjutnya, tetapi untuk dapat menggunakan algoritma *k-means*, jumlah *cluster* yang ada dalam data perlu diketahui dalam beberapa kali percobaan atau uji coba akan diperlukan untuk menemukan jumlah *cluster* terbaik (Oti et al., 2021).

Untuk menerapkan algoritma *K-Means*, terdapat beberapa langkah utama yang dilakukan secara iteratif hingga hasil optimal tercapai. Proses ini bertujuan untuk mengelompokkan data berdasarkan kesamaan fitur sehingga setiap *cluster* memiliki karakteristik yang serupa. Adapun langkah-langkahnya adalah sebagai berikut:

1. Menentukan jumlah *cluster* (K) dan *centroid* awal
 - Tentukan jumlah *cluster* K yang akan digunakan.
 - Pilih K *centroid* awal secara acak dari data.
2. Menghitung jarak dan mengelompokkan data
Hitung jarak setiap data ke *centroid* menggunakan rumus jarak *Euclidean Distance* seperti pada rumus dibawah ini :

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Keterangan:

i : index dari atribut.

n : jumlah data.

x_i : atribut dari data ke- i , ($i = 1, 2, 3, \dots, n$).

y_i : atribut dari pusat *Cluster* ke- i , ($i = 1, 2, 3, \dots, n$).

3. Menghitung ulang *centroid* baru
Perbarui posisi *centroid* dengan mengambil rata-rata dari semua data dalam *cluster* tersebut seperti pada rumus dibawah:

$$v = \frac{\sum_{i=1}^n x_i}{n}; i = 1, 2, 3, \dots, n$$

Keterangan:

v : *Centroid* pada *Cluster*.

n : banyaknya data pada *Cluster*.

x_i : data ke- i , ($i = 1, 2, 3, \dots, n$).

4. Mengulangi proses hingga konvergensi
Ulangi langkah pengelompokan dan perhitungan *centroid* hingga tidak ada perubahan signifikan atau konvergensi tercapai.
5. Evaluasi hasil *clustering*
Gunakan metode evaluasi seperti *Elbow Method* atau *Silhouette Score* untuk menentukan jumlah *cluster* terbaik.

1.6.10 Elbow Method

Elbow Method adalah salah satu teknik yang digunakan untuk menentukan jumlah kluster optimal dalam *K-Means Clustering* dengan menganalisis perubahan nilai *Within-Cluster Sum of Squares (WCSS)* terhadap jumlah *cluster* (K). Total *WCSS* dihitung dengan menjumlahkan jarak kuadrat dari setiap titik ke *centroid cluster* terdekatnya seperti yang terlihat pada rumus dibawah :

$$WCSS = \sum_{k=1}^K \sum_{x \in C_k} \|x - \mu_k\|^2$$

Keterangan:

K : jumlah *cluster*.

C_k : himpunan data dalam *cluster* ke- k .

x : titik data dalam *cluster*.

μ_k : pusat (*centroid*) *cluster* ke- k .

Metode ini bekerja dengan memplot nilai *WCSS* pada berbagai jumlah *cluster* dan mencari titik di mana kurva mulai melandai atau membentuk sudut seperti siku (*elbow*), yang menunjukkan bahwa menambah jumlah *cluster* setelah titik tersebut tidak lagi memberikan pengurangan *WCSS* yang signifikan (Syakur et al., 2018).

1.6.11 Blackbox Testing

Menurut buku *The Art of Testing* yang ditulis oleh Glenford J. Myers (2004), *blackbox testing* merupakan salah satu strategi pengujian perangkat lunak yang berfokus pada menemukan kondisi di mana program tidak berjalan sesuai spesifikasinya. Metode ini tidak berkonsentrasi pada perilaku internal atau struktur program, melainkan pada pengujian fungsionalitas perangkat lunak berdasarkan spesifikasi yang telah ditentukan. *Blackbox testing* mulai diperkenalkan pada tahun 1979. Blackbox testing sering digunakan untuk menguji berbagai aspek perangkat lunak, seperti validasi input dan output, respons sistem terhadap berbagai skenario, serta pengujian batas dan kesalahan

1.6.12 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) adalah pengujian interaksi antara end-user dan sistem secara langsung yang berfungsi untuk memverifikasi bahwa fitur telah berjalan sesuai dengan kebutuhan user tersebut. Pengujian *UAT* termasuk fase terakhir dalam proses pengujian pada sistem, yang dimana sistem telah selesai melalui tahap pengembangan. *UAT* menjadi salah satu rangkaian pengujian final dari perangkat lunak dan dilakukan sebelum dikembangkan dan diluncurkan.

[illegible]

2.2.2 Tempat Penelitian

Penelitian dilakukan secara daring dengan pengumpulan data melalui proses *web scraping* dari situs *JobStreet*, yang kemudian dianalisis menggunakan algoritma *K-Means Clustering*.

2.3 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *waterfall*. Ada beberapa tahap yang merupakan bagian dari metode ini. Tahapan – tahapan tersebut adalah sebagai berikut :

1. *Requirements* (Analisis Kebutuhan)

Tahapan ini menganalisis kebutuhan sistem yang diperlukan untuk merancang sistem pencarian kerja berbasis *clustering* pada platform *JobStreet*. Berbagai kebutuhan data diperoleh dari hasil observasi dan studi literatur, dengan fokus pada pola-pola tertentu seperti nama pekerjaan, kategori pekerjaan, lokasi, dan informasi lainnya. *Requirements* ini bertujuan agar nantinya dapat memahami fitur pada sistem, untuk menampilkan hasil *clustering* sesuai kebutuhan.

2. *Design* (Perancangan)

Pada tahapan ini, dirancang elemen-elemen sebagai berikut :

- *Use Case Diagram* : Untuk memberikan gambaran alur sistem, termasuk proses pencarian berdasarkan hasil *clustering*.
- *Rancangan UI/UX* : Merancang tampilan antarmuka sistem agar dapat mudah digunakan oleh pengguna yang nantinya memakai sistem, termasuk nantinya digunakan untuk visualisasi hasil *clustering* daftar pekerjaan.

3. *Implementation* (Implementasi)

Untuk tahapan ini, rancangan sistem yang diimplementasikan menggunakan teknologi berbasis web. Teknologi yang digunakan adalah :

- *Python* yang berfungsi untuk pemrosesan data dan algoritma *k-means clustering*.
- *Flask* sebagai *framework backend* yang menangani proses *clustering* dan pengelolaan data.
- *React js* yang digunakan sebagai frontend yang berfungsi untuk membangun antarmuka pengguna yang interaktif
- *Microsoft Excel* digunakan sebagai media penyimpanan untuk mendukung hasil dari proses *webscraping* dan *clustering*.

4. *Testing* (Pengujian)

Black Box Testing digunakan untuk memeriksa fitur-fitur dalam sistem agar dapat berfungsi dengan baik.

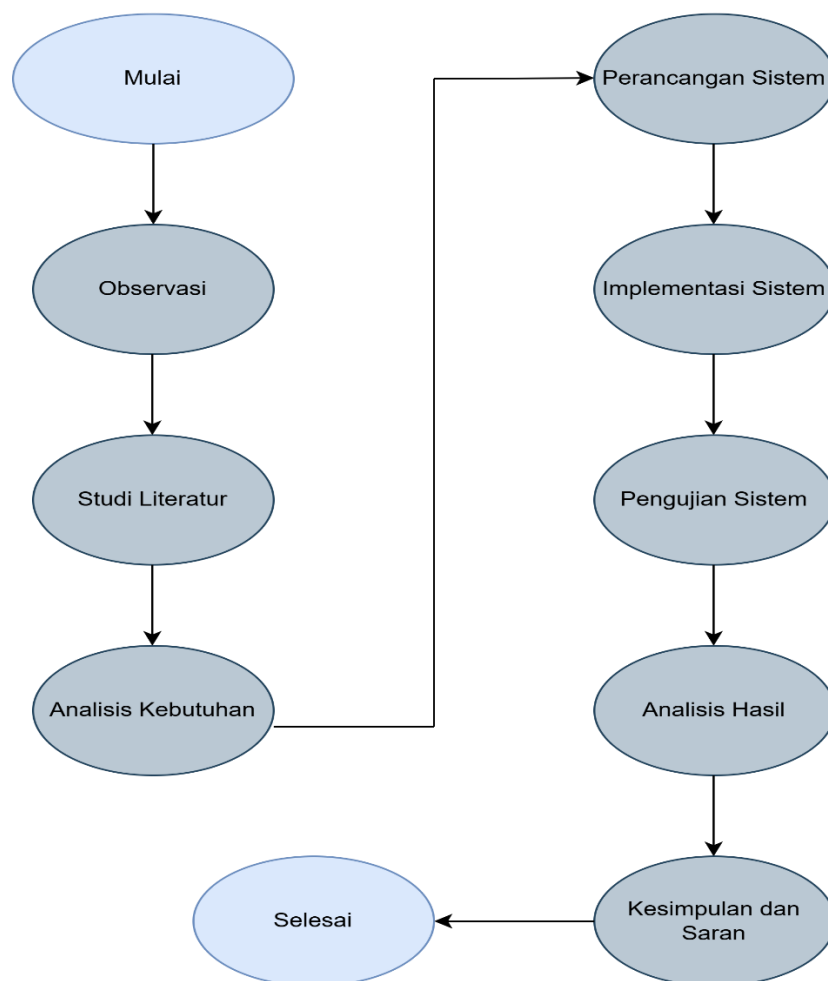
5. *Maintenance* (Pemeliharaan)

Tahap ini meliputi pemeliharaan sistem secara berkala untuk memastikan bahwa sistem pencarian kerja berbasis *clustering* tetap berfungsi optimal. Pemeliharaan ini mencakup pembaruan dataset, pengoptimalan algoritma *clustering*, dan penambahan fitur baru berdasarkan kebutuhan pengguna.

2.4 Tahapan Penelitian

Berikut adalah rangkuman urutan tahapan penelitian yang dilakukan dalam penelitian ini, dimulai dari pengumpulan data dan studi literatur hingga tahap evaluasi dan perbaikan sistem. Tahapan-tahapan ini disusun secara sistematis untuk memastikan bahwa setiap aspek kebutuhan dan pengembangan sistem telah terpenuhi, serta data yang diperoleh selalu akurat dan relevan. Urutan tahapan penelitian dapat dilihat pada Gambar 3.

1. Pengumpulan Data dan Studi Literatur
 - Melakukan observasi dan studi literatur untuk mengumpulkan informasi awal.
 - Menggunakan metode *web scraping* untuk mengambil data secara otomatis dari *JobStreet* melalui *API* dan *network request*.
 - Mengolah data hasil *web scraping* untuk mendalami pola jenis pekerjaan.
2. Requirements
Melakukan analisis kebutuhan sistem berdasarkan hasil observasi dan studi literatur.
3. Design
Merancang sistem termasuk pembuatan user interface beserta pemrograman dari aplikasi yang akan digunakan.
4. Implementation (Implementasi)
 - Mengimplementasikan rancangan yang telah dibuat ke dalam aplikasi *web* sesuai dengan spesifikasi yang ditentukan.
 - Mengintegrasikan data hasil *web scraping* ke dalam aplikasi sehingga informasi ditampilkan secara dinamis.
5. Testing (Pengujian)
Melakukan pengujian sistem menggunakan metode *blackbox testing* untuk memastikan aplikasi berjalan dengan baik dan sesuai kebutuhan.
6. Evaluasi dan Penyempurnaan
Jika sistem tidak memenuhi kebutuhan, tahapan akan diulang ke tahap *requirements*.



Gambar 3.Tahapan penelitian.

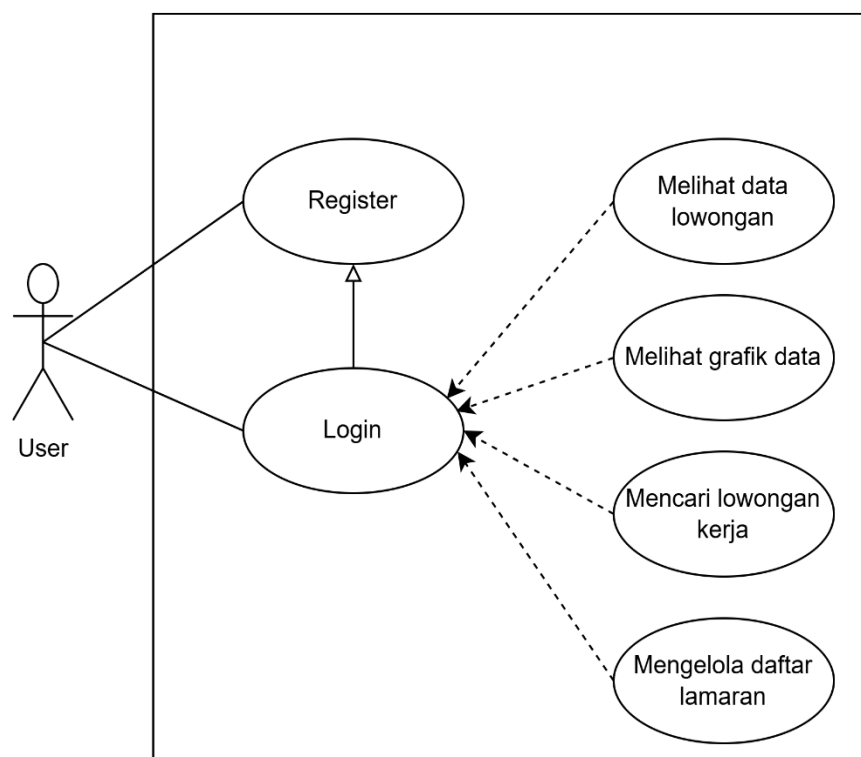
2.5 Rancangan Sistem Aplikasi

Pada tahapan perancangan sistem aplikasi, peneliti menggunakan sistem *use case diagram* dalam menggambarkan aktivitas yang dilakukan oleh *user* di dalam aplikasi. Sistem rancangan dikelola untuk mendukung pengguna dalam menjelajah data-data pada lowongan kerja, mengelola lamaran, dan agar dapat memanfaatkan dengan baik hasil dari data-data yang telah melalui klasifikasi dengan algoritma berbasis *k-means clustering*.

Sebagaimana ditunjukkan dalam Gambar 4, dalam rancangannya hanya terdapat satu aktor, yaitu *user* yang memiliki berbagai peran sebagai berikut :

- Melakukan registrasi ke aplikasi.
- Melakukan login ke aplikasi.
- Mengelola daftar lamaran yang telah dipilih pada lowongan kerja yang tersedia.
- Melakukan pencarian lowongan berdasarkan *cluster*.
- Melihat visualisasi jumlah lowongan pada *dashboard*.

Dalam perancangannya sistem ini fokus pada kebutuhan *user* dalam mempermudah mengakses informasi lowongan kerja secara cepat dan akurat. Fitur yang dirancang memberikan pengalaman yang lebih menarik dalam mencari lowongan kerja berdasarkan *clustering*. Visualisasi data dalam bentuk grafik membantu dalam memahami ketersediaan lowongan kerja yang ada, sehingga *user* dapat membuat keputusan yang efisien. Dengan tampilan antarmuka yang ramah pengguna, aplikasi ini bertujuan untuk memberikan aksesibilitas yang maksimal bagi *user* tanpa perlu proses administrasi tambahan.



Gambar 4. Use case diagram

2.6 Rancangan Tampilan Web

Rancangan tampilan web yang dikembangkan pada aplikasi ini bertujuan untuk membangun sistem pencarian kerja *JobStreet* berdasarkan pola jenis pekerjaan yang sudah melalui proses *clustering* dengan menggunakan algoritma *k-means clustering*. Aplikasi dibuat dengan menggunakan *Flask* sebagai *backend* dan *Reactjs* untuk *frontendnya*. Data yang diambil melalui proses *web scraping* dilakukan secara otomatis, sehingga dalam aplikasi ini tidak memerlukan adanya admin untuk pengelolaan data.

Aplikasi *web* ini terdiri dari tiga halaman, yaitu :

1. Halaman *Dashboard*

Halaman ini menampilkan visual dari data lowongan kerja hasil *web scraping* yang sudah melalui proses *clustering*.

Dalam melakukan visualisasi data, data divisualisasikan dengan menggunakan bentuk grafik garis yang memberikan gambaran mengenai distribusi lowongan kerja.

Detail Tampilan :

- a. Grafik garis menunjukkan jumlah lowongan kerja yang sudah melalui proses *clustering* dan sudah disimpan pada *backend*, dimana grafiknya berdasarkan klasifikasi pekerjaan.
- b. Sumbu x pada grafik mempresentasikan klasifikasi tiap pekerjaan.
- c. Sumbu y pada grafik mempresentasikan jumlah lowongan pekerjaan berdasarkan klasifikasi.

2. Halaman Lamaran

Pada halaman lamaran menampilkan daftar pekerjaan yang telah dipilih oleh *user* di halaman lowongan. Hal-hal yang ditampilkan pada halaman lamaran yaitu :

- a. Perusahaan.
- b. Lokasi perusahaan.
- c. Link lamaran.
- d. Tanggal lamaran.

3. Halaman lowongan

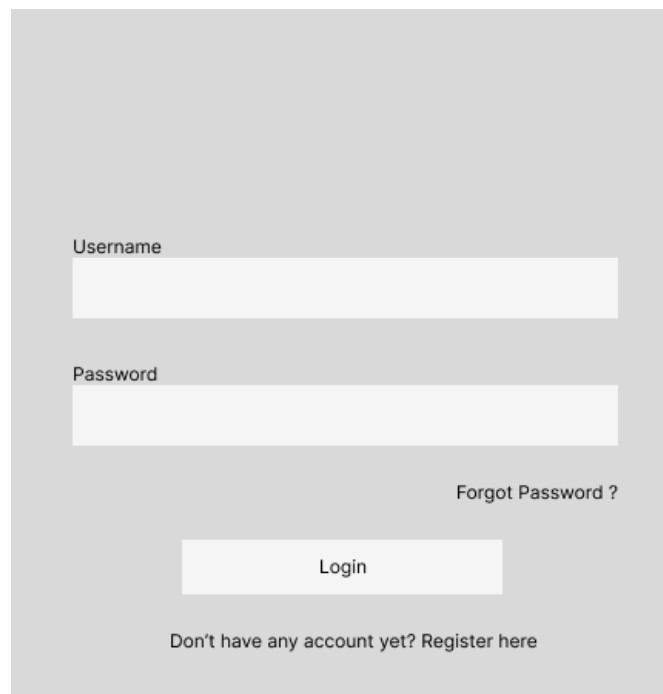
Halaman lowongan menampilkan data-data hasil dari *web scarping* yang terdapat pada *backend*. Data-data yang ditampilkan seperti :

- a. Perusahaan.
- b. Nama pekerjaan.
- c. Tanggal lamaran.
- d. Tipe pekerjaan.
- e. Klasifikasi pekerjaan.
- f. Link pekerjaan.
- g. Sub klasifikasi pekerjaan.
- h. Penjelasan pekerjaan.

Aplikasi *web* ini dirancang untuk memudahkan pengguna agar dapat memahami klasifikasi lowongan kerja,menjelajahi data-data lowongan, dan mengelola lamaran mereka dengan baik.

2.6.1 Halaman *Login*

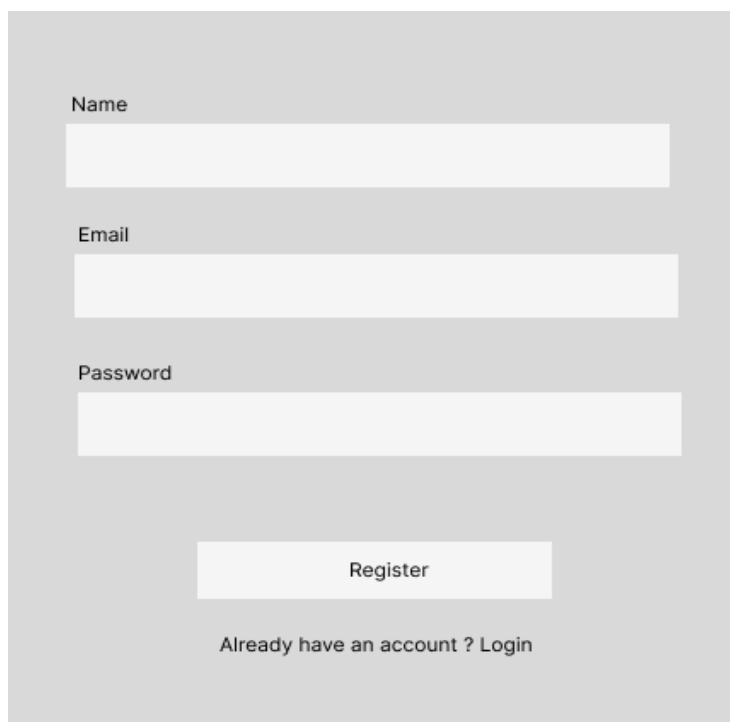
Halaman *login* adalah halaman saat user harus memasukkan username dan password terlebih dahulu untuk dapat mengakses fitur-fitur yang tersedia di dalam website sebagaimana terlihat pada Gambar 5. Proses ini memastikan bahwa hanya pengguna yang terdaftar yang dapat menggunakan layanan yang disediakan oleh sistem.

A login form with a light gray background. It features two input fields: 'Username' and 'Password'. Below the password field is a link 'Forgot Password ?'. A 'Login' button is centered below the link. At the bottom, there is a link 'Don't have any account yet? Register here'.

Gambar 5. Halaman *login*

2.6.2 Halaman *Register*

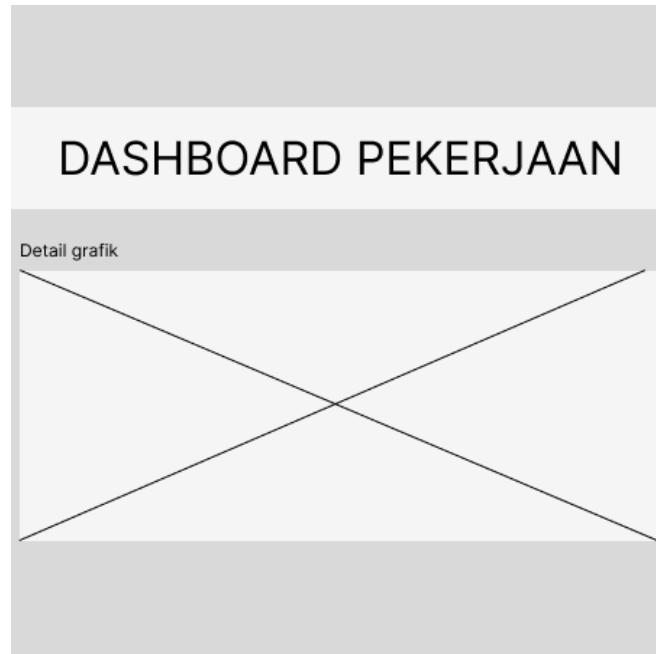
Halaman register adalah halaman yang berfungsi untuk membuat akun baru bagi user yang ingin menggunakan website. Sebagaimana ditunjukkan pada Gambar 6, halaman register menjadi langkah awal sebelum user dapat mengakses fitur-fitur utama dalam sistem, seperti melihat data lowongan kerja berdasarkan *K- Means Clustering*, mencari pekerjaan yang sesuai, serta mengelola daftar lamaran.

A register form with a light gray background. It features three input fields: 'Name', 'Email', and 'Password'. Below the password field is a 'Register' button. At the bottom, there is a link 'Already have an account ? Login'.

Gambar 6. Halaman *register*

2.6.3. Halaman *Dashboard*

Halaman dashboard menampilkan data-data lowongan pekerjaan yang telah melalui proses *clustering* dengan visualisasi grafik. Fungsi dari halaman dashboard ini adalah memberikan gambaran kepada pengguna mengenai distribusi atau jumlah lowongan yang masih tersedia berdasarkan kategori tertentu. Rancangan halaman dashboard bisa dilihat pada Gambar 7.



Gambar 7. Halaman *dashboard*

2.6.4 Halaman Utama

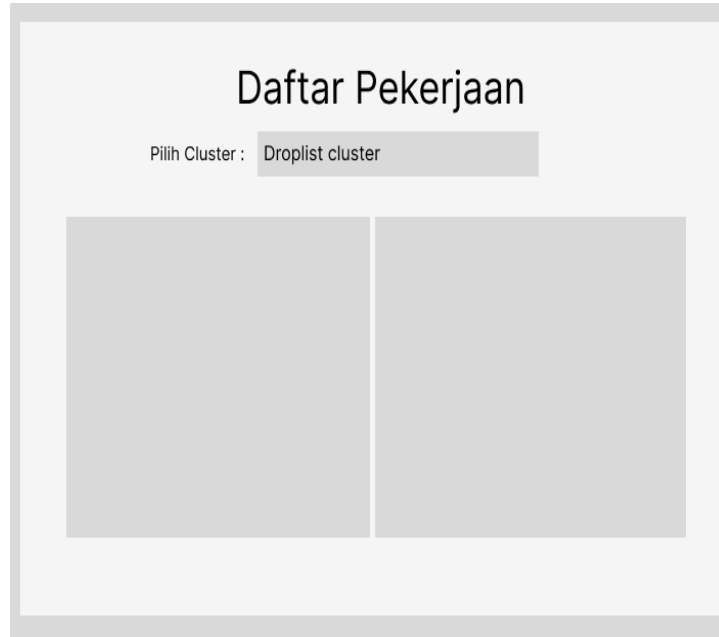
Halaman utama adalah tampilan awal yang dapat dilihat oleh pengguna setelah mereka *login*. Pada halaman ini pengguna dapat melihat berbagai fitur yang disajikan dengan pesan-pesan yang menarik seperti yang terlihat pada Gambar 8. Tujuan dari halaman ini adalah untuk memberikan gambaran singkat kepada pengguna tentang tujuan dan manfaat dari situs yang dibuat.



Gambar 8. Halaman Utama.

2.6.5 Halaman lowongan

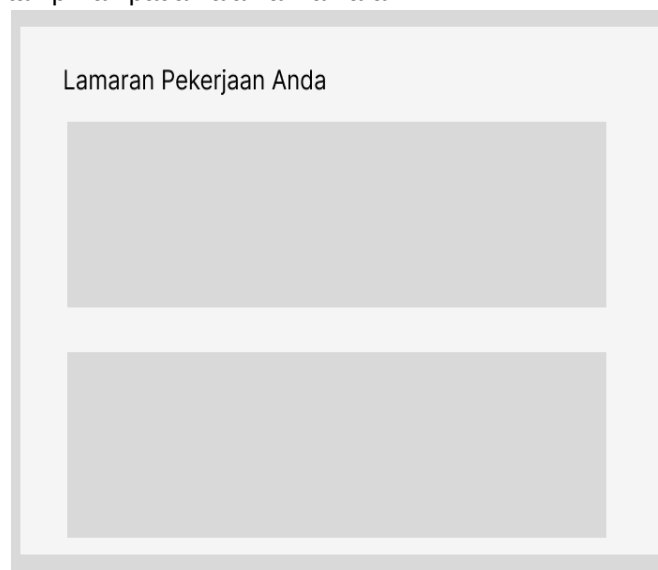
Halaman lowongan adalah halaman yang memungkinkan pengguna untuk memilih daftar pekerjaan berdasarkan hasil *clustering* yang tersedia seperti yang terlihat pada Gambar 9. Pengguna memilih *cluster* yang tersedia berdasarkan minat dan keahlian mereka.



Gambar 9. Halaman lowongan

2.6.6 Halaman Lamaran

Halaman lamaran adalah halaman yang menyimpan pekerjaan yang diminati oleh pengguna di halaman lowongan seperti yang terlihat pada Gambar 10. Pada halaman lamaran hanya menyimpan data-data penting yang ada pada halaman lowongan, jadi tidak semua data pada halaman lowongan ditampilkan pada halaman lamaran.



Gambar 10. Halaman Lamaran

2.7 Analisis Kebutuhan Sistem

Dalam mengembangkan sistem ini terdapat berbagai macam kebutuhan agar proses penelitian dapat berjalan dengan lancar dan implementasi dari sistem dapat memenuhi kebutuhan pengguna. Kebutuhan itu dibagi menjadi dua yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*).

2.7.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan selama menjalani penelitian, yaitu :

Tabel 4. Perangkat keras

No	Perangkat keras	Spesifikasi
1	Prosesor	<i>Intel® Core™ i3-10110U CPU @ 2.10GHz</i>
2	<i>RAM</i>	<i>8 GB</i>
3	<i>GPU</i>	<i>NVIDIA GeForce MX130</i>

2.7.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan selama penelitian, yaitu :

Tabel 5. Perangkat lunak

No	Perangkat lunak	Fungsi
1	<i>Google Colaboratory</i>	<i>Jupyter notebook</i> yang disediakan oleh <i>Google</i> dengan fitur TPU gratis
2	<i>Matplotlib</i>	<i>Library</i> untuk visualisasi data
3	<i>Pandas</i>	<i>Library</i> untuk mengelola data
4	<i>Visual Studio Code</i>	Editor kode yang digunakan untuk pengembangan dan implementasi.
5	<i>DashWind</i>	<i>Framework</i> untuk pengembangan <i>frontend web</i> .
6	<i>Flask</i>	Sebagai <i>backend</i> dari aplikasi sistem.