

**OPTIMASI K-MEANS DENGAN LOCAL OUTLIER FACTOR  
UNTUK MENGATASI DATA OUTLIER  
Studi Kasus: Klasterisasi Performa Akademik Mahasiswa**

**TUGAS AKHIR**



**Disusun Oleh :**  
**MUHAMMAD ANJAR HARIMURTI RAHADI**  
**123180056**

**PROGRAM STUDI INFORMATIKA  
JURUSAN INFORMATIKA  
FAKULTAS TEKNIK INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
YOGYAKARTA  
2023**

**OPTIMASI K-MEANS DENGAN LOCAL OUTLIER FACTOR  
UNTUK MENGATASI DATA OUTLIER  
Studi Kasus: Klasterisasi Performa Akademik Mahasiswa**

**TUGAS AKHIR**

Tugas Akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana S-1 di Program Studi Informatika, Jurusan Informatika, Fakultas Teknik Industri, Universitas Pembangunan Nasional “Veteran” Yogyakarta



**Disusun Oleh :**

**MUHAMMAD ANJAR HARIMURTI RAHADI**

**123180056**

**PROGRAM STUDI INFORMATIKA  
JURUSAN INFORMATIKA  
FAKULTAS TEKNIK INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
YOGYAKARTA  
2023**

## HALAMAN PENGESAHAN PEMBIMBING

### OPTIMASI K-MEANS DENGAN LOCAL OUTLIER FACTOR UNTUK MENGATASI DATA OUTLIER Studi Kasus: Klasterisasi Performa Akademik Mahasiswa

Disusun Oleh:

Muhammad Anjar Harimurti Rahadi

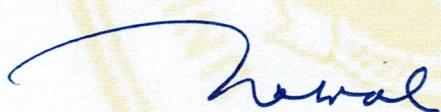
123180056

Telah diuji dan dinyatakan lulus pada tanggal .....

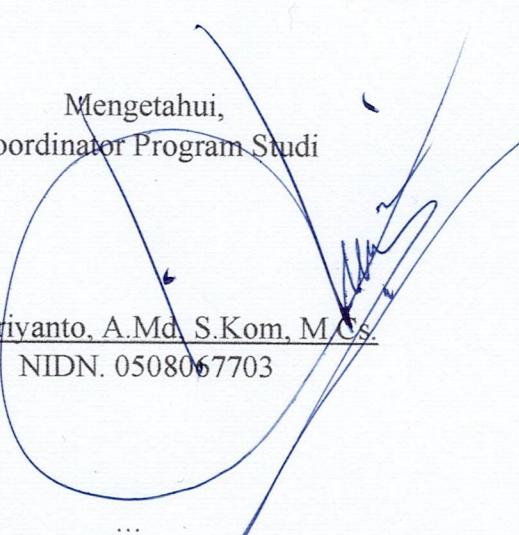
Menyetujui,

Pembimbing I

Pembimbing II

  
Nur Heri Cahyana, S.T, M.Kom.  
NIDN. 0022096003

  
Dr. Herlina Jayadianti, S.T, M.T  
NIDN. 0527087701

Mengetahui,  
Koordinator Program Studi  
  
Dr. Heriyanto, A.Md, S.Kom, M.GS.  
NIDN. 0508067703

## HALAMAN PENGESAHAN PENGUJI

### OPTIMASI K-MEANS DENGAN LOCAL OUTLIER FACTOR UNTUK MENGATASI DATA OUTLIER Studi Kasus: Klasterisasi Performa Akademik Mahasiswa

Disusun Oleh:

Muhammad Anjar Harimurti Rahadi

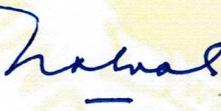
123180056

Telah diuji dan dinyatakan lulus pada tanggal .....

Menyetujui,

Penguji I

Penguji II

  
Nur Heri Cahyana, S.T, M.Kom.

NIDN. 0022096003

  
Dr. Herlina Jayadianti, S.T, M.T

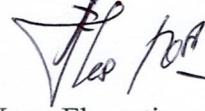
NIDN. 0527087701

Penguji III

Penguji IV

  
Vynska Amalia Permadji, S.Kom, M.Kom

NIDN. 0025089302

  
Mangaras Yanu Florestiyanto, S.T., M.Eng

NIDN. 0521018201

## **PERNYATAAN BEBAS PLAGIASI**

Saya yang bertanda tangan di bawah ini :

Nama : Muhammad Anjar Harimurti Rahadi  
NIM : 123180056  
Fakultas/Prodi : Teknik Industri/Informatika

dengan ini saya menyatakan bahwa judul Tugas Akhir

### **Optimasi K-Means Dengan Local Outlier Factor Untuk Mengatasi Data Outlier Studi Kasus: Klasterisasi Performa Akademik Mahasiswa**

adalah hasil kerja saya sendiri dan benar bebas dari plagiasi kecuali cuplikan serta ringkasan yang terdapat di dalamnya telah saya jelaskan sumbernya (Situs) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No 17 Tahun 2010 dan Peraturan Perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Yogyakarta, 1 Februari 2023....

Yang membuat pernyataan,



Muhammad Anjar Harimurti Rahadi

NIM. 123180056

## **SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR**

Sebagai mahasiswa Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta, yang bertanda tangan dibawah ini, saya:

Nama: Muhammad Anjar Harimurti Rahadi

NIM: 123180056

Menyatakan bahwa karya ilmiah saya yang berjudul:

**Optimasi K-Means Dengan Local Outlier Factor Untuk Mengatasi Data Outlier Studi Kasus: Klasterisasi Performa Akademik Mahasiswa**

merupakan karya asli saya dan belum pernah dipublikasikan dimanapun. Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apa pun yang diberikan Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta  
Pada tanggal : 1 Februari 2023

Yang menyatakan,



Muhammad Anjar Harimurti Rahadi

NIM. 123180056

## **ABSTRAK**

Dalam *machine learning*, *clustering* adalah metode untuk menganalisis data statistik dalam *unsupervised learning*. K-Means adalah metode yang sering digunakan untuk mengelompokkan data dengan cepat dan sederhana dengan mencari kombinasi variabel dan atribut objek. Namun, metode ini akan mengalami masalah jika data memiliki *outlier*.

Dalam penelitian ini, metode *Local Outlier Factor* (LOF) digunakan untuk mengatasi masalah anomali *outlier* yang diangkat oleh K-Means dengan mendeteksi *outlier* berbasis kepadatan dengan menghitung *local variance* dari titik data tertentu. LOF akan diterapkan setelah *preprocessing* data dan sebelum masuk proses *clustering* data K-Means.

Data dari kuisioner performa akademik mahasiswa sebanyak 210 data yang terbagi dalam tiga *cluster*, digunakan untuk pengujian. Dengan terdeteksi sebanyak 38 data atau 18% data *outlier*. Penerapan LOF meningkatkan *silhouette score* sebesar 10,23%. *Elbow method* juga digunakan dengan *silhouette score* untuk mendapatkan nilai K optimal = 3.

Kata kunci: *K-Means*, *Local Outlier Factor*, Klasterisasi, *Outlier*

## **ABSTRACT**

*In machine learning, clustering is a method for analyzing statistical data in unsupervised learning. K-Means is a well-known method for quickly and simply clustering data by searching for combinations of variables and object attributes. The algorithm will have issues if the data has outliers.*

*In this study, the Local Outlier Factor (LOF) method was used to solve the outlier anomaly concerns raised by K-Means by detecting density-based outliers by calculating the local variance of a particular data. Before the K-Means data clustering process, LOF will be applied.*

*Data from the distribution of the student academic performance questionnaire, totaling 210 data that were separated into three clusters, were used for the testing. With up to 38 data detected, this represents for 18% outlier data. The LOF approach increased the silhouette score by 10.23%. The elbow method is used with the silhouette score to obtain an optimal K value = 3.*

*Keywords:* *K-Means, Local Outlier Factor, Clustering, Outlier*

## KATA PENGANTAR

Dengan mengucapkan puji dan syukur kehadiran Tuhan Yang Maha Esa, penulis dapat menyelesaikan tugas akhir yang berjudul " Optimasi *K-Means* Dengan *Local Outlier Factor* Untuk Mengatasi Data *Outlier* Studi Kasus: Klasterisasi Performa Akademik Mahasiswa". Tugas akhir ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan program studi Informatika di Universitas Pembangunan Nasional "Veteran" Yogyakarta.

Selama proses penggerjaan tugas akhir ini, penulis menyadari akan bantuan yang penulis terima. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih atas semua bantuan, bimbingan, maupun dukungan yang penulis terima kepada seluruh pihak yang terlibat, yaitu :

1. Bapak Dr. Awang Hendrianto Pratomo, S.T., M.T selaku Ketua Jurusan Informatika UPN "Veteran" Yogyakarta,
2. Bapak Dr. Heriyanto, A.Md, S.Kom, M.Cs. selaku Koordinator Program Studi Informatika UPN "Veteran" Yogyakarta,
3. Bapak Nur Heri Cahyana, S.T, M.Kom dan Ibu Dr. Herlina Jayadianti, S.T, M.T selaku dosen pembimbing atas bantuan, bimbingan, arahan, serta saran yang sudah diberikan kepada penulis,
4. Ibu Vynska Amalia Permadi, S.Kom, M.Kom dan Mangaras Yanu Florestiyanto, S.T., M.Eng selaku dosen penguji atas bantuan, arahan, serta saran yang sudah diberikan kepada penulis,
5. Bapak, Ibu, saudara dan saudari penulis atas doa, dukungan serta motivasi selama proses penggerjaan tugas akhir ini,
6. Sahabat-sahabat penulis, Rama, Allyandaru, Ekky, Isti, Fiqup, Vania, Rifka, Meri, Isna, Zahra, Satya, dan Haris yang selalu memberikan *support* kepada penulis dalam suka maupun duka, serta telah menemani penulis selama perkuliahan.
7. Keluarga besar IF 2018 serta *Information Technology Club* (ITC) yang telah menjadikan penulis orang yang bertanggung jawab serta tempat untuk menempa diri menjadi orang yang lebih baik.

Akhir kata, semoga skripsi ini dapat memberikan manfaat bagi pembaca dan berkontribusi dalam pengembangan ilmu pengetahuan. Penulis juga menyadari bahwa tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan agar penulis dapat menyempurnakan karya ini di masa yang akan datang.

Yogyakarta, 1 Februari 2023  
Penulis

## DAFTAR ISI

|  |             |
|--|-------------|
| <b>HALAMAN JUDUL.....</b>                            | <b>ii</b>   |
| <b>HALAMAN PENGESAHAN PEMBIMBING .....</b>           | <b>iii</b>  |
| <b>HALAMAN PENGESAHAN PENGUJI.....</b>               | <b>iv</b>   |
| <b>PERNYATAAN BEBAS PLAGIASI.....</b>                | <b>v</b>    |
| <b>SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR .....</b> | <b>vi</b>   |
| <b>ABSTRAK.....</b>                                  | <b>vii</b>  |
| <b>ABSTRACT .....</b>                                | <b>viii</b> |
| <b>KATA PENGANTAR .....</b>                          | <b>ix</b>   |
| <b>DAFTAR ISI .....</b>                              | <b>x</b>    |
| <b>DAFTAR GAMBAR .....</b>                           | <b>xii</b>  |
| <b>DAFTAR TABEL .....</b>                            | <b>xiii</b> |
| <b>DAFTAR PERSAMAAN .....</b>                        | <b>xiv</b>  |
| <b>DAFTAR MODUL PROGRAM .....</b>                    | <b>xv</b>   |
| <b>BAB I PENDAHULUAN .....</b>                       | <b>1</b>    |
| 1.1    Latar Belakang.....                           | 1           |
| 1.2    Rumusan Masalah .....                         | 2           |
| 1.3    Batasan Masalah.....                          | 2           |
| 1.4    Tujuan Penelitian.....                        | 2           |
| 1.5    Manfaat Penelitian.....                       | 3           |
| 1.6    Tahapan Penelitian .....                      | 3           |
| 1.6.1    Metodologi Penelitian .....                 | 3           |
| 1.6.2    Metodologi Pengembangan Sistem .....        | 4           |
| 1.7    Sistematika Penulisan.....                    | 5           |
| <b>BAB II TINJAUAN LITERATUR .....</b>               | <b>7</b>    |
| 2.1    Landasan Teori .....                          | 7           |
| 2.1.1 <i>Machine Learning</i> .....                  | 7           |
| 2.1.2 <i>Clustering</i> .....                        | 7           |
| 2.1.3 <i>K-Means</i> .....                           | 8           |
| 2.1.4 <i>Local Outlier Factor (LOF)</i> .....        | 9           |
| 2.1.5 <i>Hyperparameter Tuning</i> .....             | 9           |
| 2.1.6 <i>Elbow Method</i> .....                      | 10          |
| 2.1.7 <i>Silhouette Method</i> .....                 | 10          |
| 2.2    Studi Literatur.....                          | 11          |

|  |           |
|--|-----------|
| <b>BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM .....</b> | <b>16</b> |
| 3.1    Metodologi Penelitian .....                                 | 16        |
| 3.1.1    Identifikasi Masalah .....                                | 16        |
| 3.1.2    Studi Literatur.....                                      | 17        |
| 3.1.3    Pengumpulan Data.....                                     | 17        |
| 3.1.4    Preprocessing Data .....                                  | 19        |
| 3.1.5    Outlier Detection .....                                   | 22        |
| 3.1.6 <i>Data Clustering</i> .....                                 | 26        |
| 3.1.7 <i>Model Evaluation</i> .....                                | 30        |
| 3.2    Metodologi Pengembangan Sistem .....                        | 32        |
| 3.2.1    Analisis Kebutuhan Sistem.....                            | 32        |
| 3.2.2    Perancangan Sistem.....                                   | 33        |
| 3.2.3    Pengujian Sistem .....                                    | 41        |
| <b>BAB IV HASIL DAN PEMBAHASAN.....</b>                            | <b>43</b> |
| 4.1    Hasil Penelitian.....                                       | 43        |
| 4.1.1 <i>Data Initialization</i> .....                             | 44        |
| 4.1.2 <i>Preprocessing Data</i> .....                              | 44        |
| 4.1.3 <i>Outlier Detection</i> .....                               | 46        |
| 4.1.4 <i>Data Clustering</i> .....                                 | 48        |
| 4.1.5 <i>Model Evaluation</i> .....                                | 51        |
| 4.1.6 <i>System Implementation</i> .....                           | 53        |
| 4.1.7 <i>System Testing</i> .....                                  | 57        |
| 4.2    Pembahasan .....  | 58        |
| <b>BAB V PENUTUP .....</b>   | <b>60</b> |
| 5.1    Kesimpulan.....   | 60        |
| 5.2    Saran .....   | 60        |
| <b>DAFTAR PUSTAKA .....</b>  | <b>61</b> |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 1.1 <i>Waterfall Process</i> (Sumber: Sommerville, 2016).....                 | 4  |
| Gambar 2.1 Ilustrasi dari Algoritma K-Means (Sumber: Muttaqin & Defriani, 2020)..... | 9  |
| Gambar 2.2 <i>Elbow Method</i> (Sumber: Shi et al., 2021).....                       | 10 |
| Gambar 3.1 Alur Tahapan Penelitian .....   | 16 |
| Gambar 3.2 Kuisioner .....   | 18 |
| Gambar 3.3 <i>Flowchart Utama</i> .....  | 19 |
| Gambar 3.4 <i>Flowchart Remove Unused and Null Data</i> .....                        | 20 |
| Gambar 3.5 <i>Flowchart Encoding Data</i> .....                                      | 21 |
| Gambar 3.6 <i>Flowchart Outlier Detection</i> .....                                  | 22 |
| Gambar 3.7 <i>Flowchart Clustering Data</i> .....                                    | 27 |
| Gambar 3.8 <i>Flowchart Model Evaluation</i> .....                                   | 30 |
| Gambar 3.9 Perancangan Arsitektur Sistem.....  | 33 |
| Gambar 3.10 Proses DFD Level 0.....  | 34 |
| Gambar 3.11 Proses DFD Level 1.....  | 34 |
| Gambar 3.12 Proses DFD Level 2 – <i>Remove Unused and Null Data</i> .....            | 35 |
| Gambar 3.13 Proses DFD Level 2 – <i>Encoding Data</i> .....                          | 36 |
| Gambar 3.14 Proses DFD Level 2 – <i>Outlier Detection</i> .....                      | 36 |
| Gambar 3.15 Proses DFD Level 2 – <i>Clustering Process</i> .....                     | 37 |
| Gambar 3.16 Proses DFD Level 2 – <i>Model Evaluation</i> .....                       | 38 |
| Gambar 3.17 Rancangan Halaman <i>Raw Data</i> .....                                  | 38 |
| Gambar 3.18 Rancangan Halaman <i>Data Cleansing</i> .....                            | 39 |
| Gambar 3.19 Rancangan Halaman <i>Data Preparation</i> .....                          | 39 |
| Gambar 3.20 Rancangan Halaman <i>Modelling</i> .....                                 | 40 |
| Gambar 3.21 Rancangan Halaman <i>Clustering Comparation</i> .....                    | 40 |
| Gambar 3.22 Rancangan Halaman <i>Cluster Analysis</i> .....                          | 41 |
| Gambar 4.1 Tampilan Aplikasi Klasterisasi Performa Akademik Mahasiswa .....          | 43 |
| Gambar 4.2 Proses <i>Removing Unused and Null Data</i> .....                         | 45 |
| Gambar 4.3 Proses <i>Encoding Data</i> .....   | 46 |
| Gambar 4.4 Hasil <i>Outlier Detection</i> .....                                      | 48 |
| Gambar 4.5 Hasil Visualisasi <i>Elbow Method</i> .....                               | 49 |
| Gambar 4.6 Hasil Visualisasi <i>Silhouette Score</i> .....                           | 50 |
| Gambar 4.7 Hasil <i>Clustering Data</i> .....  | 51 |
| Gambar 4.8 <i>Silhouette Visualization</i> .....                                     | 52 |
| Gambar 4.9 Halaman <i>Raw Data</i> .....   | 53 |
| Gambar 4.10 Halaman <i>Data Cleansing – Remove Unused and Null Data</i> .....        | 53 |
| Gambar 4.11 Halaman <i>Data Cleansing – Encoding Data</i> .....                      | 54 |
| Gambar 4.12 Halaman <i>Data Preparation</i> .....                                    | 54 |
| Gambar 4.13 Halaman <i>Modelling – Hyperparameter Tuning</i> .....                   | 55 |
| Gambar 4.14 Halaman <i>Modelling – Clustering Result</i> .....                       | 55 |
| Gambar 4.15 Halaman <i>Clustering Comparation – Before Optimization</i> .....        | 56 |
| Gambar 4.16 Halaman <i>Clustering Comparation – After Optimization</i> .....         | 56 |
| Gambar 4.17 Halaman <i>Cluster Analysis – Final Analysis</i> .....                   | 56 |
| Gambar 4.18 Halaman <i>Cluster Analysis – Model Evaluation</i> .....                 | 57 |

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 2.1 Tabel Nilai <i>Silhouette</i> .....                   | 10 |
| Tabel 2.4 Tabel <i>State of The Art</i> .....                   | 13 |
| Tabel 2.5 Tabel <i>State of The Art</i> Lanjutan.....           | 14 |
| Tabel 3.1 Daftar Pertanyaan yang Diajukan.....                  | 17 |
| Tabel 3.3 Parameter yang digunakan pada penelitian .....        | 18 |
| Tabel 3.4 Tabel Sampel Data .....                               | 23 |
| Tabel 3.5 Tabel perhitungan jarak antar data .....              | 23 |
| Tabel 3.6 Tabel Nilai <i>reachdist3</i> .....                   | 25 |
| Tabel 3.7 Tabel LOF <i>Score</i> .....                          | 26 |
| Tabel 3.8 Tabel Titik Pusat <i>Cluster</i> .....                | 27 |
| Tabel 3.9 Tabel Hasil <i>Clustering</i> Iterasi Satu.....       | 28 |
| Tabel 3.10 Tabel Pusat <i>Cluster</i> Satu Baru.....            | 28 |
| Tabel 3.11 Tabel Pusat <i>Cluster</i> Dua Baru .....            | 28 |
| Tabel 3.12 Tabel Pusat <i>Cluster</i> Tiga Baru .....           | 29 |
| Tabel 3.13 Perbandingan Pusat <i>Cluster</i> Baru dan Lama..... | 29 |
| Tabel 3.14 Tabel Hasil Akhir Proses <i>Clustering</i> .....     | 29 |
| Tabel 3.15 Tabel <i>Silhouette Score</i> .....                  | 31 |
| Tabel 3.16 Spesifikasi Kebutuhan Perangkat Keras.....           | 32 |
| Tabel 3.17 Spesifikasi Kebutuhan Perangkat Lunak.....           | 32 |
| Tabel 3.18 Rancangan Pengujian Sistem .....                     | 41 |
| Tabel 3.19 Rancangan Pengujian Sistem Lanjutan .....            | 42 |
| Tabel 4.1 <i>Silhouette Score</i> .....                         | 50 |
| Tabel 4.2 Pembagian Cluster Hasil Proses <i>K-Means</i> .....   | 51 |
| Tabel 4.3 Hasil Pengujian Sistem.....                           | 57 |
| Tabel 4.4 Hasil Pengujian Sistem.....                           | 58 |

## **DAFTAR PERSAMAAN**

|   |    |
|---|----|
| Persamaan 2.1 <i>Euclidean Distance</i> .....                               | 8  |
| Persamaan 2.2 Persamaan Cluster Baru .....                                  | 8  |
| Persamaan 2.3 <i>Local Outlier Factor</i> .....                             | 9  |
| Persamaan 2.4 <i>Local Reachable Density</i> .....                          | 9  |
| Persamaan 2.5 <i>Silhouete Coefficient</i> .....                            | 11 |
| Persamaan 2.6 Jarak rata-rata antara titik dalam cluster yang sama .....    | 11 |
| Persamaan 2.7 Jarak rata-rata antara titik dalam cluster yang berbeda ..... | 11 |

## **DAFTAR MODUL PROGRAM**

|   |    |
|---|----|
| Modul Program 4.1 Proses Import Data .....                          | 44 |
| Modul Program 4.2 Proses <i>Removing Unused and Null Data</i> ..... | 44 |
| Modul Program 4.3 Proses <i>Encoding Data</i> .....                 | 45 |
| Modul Program 4.4 <i>Local Outlier Factor</i> .....                 | 46 |
| Modul Program 4.5 <i>Outlier Detection</i> .....                    | 47 |
| Modul Program 4.6 <i>Outlier Removal</i> .....                      | 47 |
| Modul Program 4.7 <i>Elbow Method</i> .....                         | 48 |
| Modul Program 4.8 <i>Silhouette Score</i> .....                     | 49 |
| Modul Program 4.9 <i>K-Means Model</i> .....                        | 51 |
| Modul Program 4.10 <i>Silhouette Visualization</i> .....            | 52 |

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

*Clustering* adalah metode umum untuk analisis data statistik dalam *Machine Learning* dalam *unsupervised learning* (Shi et al., 2021) yang melibatkan pengidentifikasi struktur dalam kumpulan data yang tidak berlabel (Madhulatha, 2012). Algoritma dalam *clustering* yang paling umum digunakan adalah algoritma *K-Means*. Algoritma ini memberikan setiap titik ke cluster yang pusatnya disebut sebagai centroid terdekat (Madhulatha, 2012). Selain *K-Means* terdapat algoritma lain yang kerap digunakan yaitu *K-Medoids*. Algoritma ini menggunakan objek pada sekumpulan objek untuk mewakili sebuah *cluster* yang disebut dengan medoid (Firzada & Yunus, 2021). Metode-metode ini banyak digunakan dalam penelitian-penelitian sebelumnya.

Penelitian-penelitian sebelumnya mengenai *clustering* telah banyak dilakukan sebelumnya, salah satunya adalah penelitian yang dilakukan oleh Vhallah (2018) yang menyatakan bahwa *K-Means* dapat menemukan karakteristik-karakteristik dari sebuah objek. Selain itu, metode metode ini dapat melakukan proses pengelompokan data dengan cepat dan sederhana (Rosmini et al., 2018). Namun, metode ini sangat dipengaruhi dan sensitif terhadap *noise* atau anomali pada data (Ariawan, 2019). Lain halnya dengan *K-Medoids*, metode ini tidak bergantung dengan urutan masuk pada dataset (Irawan et al., 2020). Selain itu, dalam proses perhitungan metode ini meminimalkan jumlah *dissimilarities* berpasangan, bukan berdasarkan jumlah kuadrat jarak (Ramadhan et al., 2021). Namun, metode ini memiliki kinerja komputasi yang cukup tinggi jika dibandingkan dengan *K-Means* (Qomariyah & Siregar, 2022).

Dari penelitian-penelitian yang sudah dilakukan sebelumnya, metode *K-Means* dinilai tepat dan berguna untuk pengelompokan karena metode ini dapat melakukan proses pengelompokan data dengan cepat dan sederhana. Selain itu, metode ini juga mampu mencari kombinasi antar variabel (Rahmawati et al., 2019) dan juga dapat menemukan karakteristik-karakteristik dari sebuah objek (Vhallah et al., 2018). Namun, metode ini sangat dipengaruhi oleh *noise* atau anomali *outlier* pada data (Ariawan, 2019). *Outlier* adalah sebuah kejadian atau item pada data yang mencurigakan dan berbeda dari data lainnya pada sebuah *dataset* (Alghushairy et al., 2020). Untuk mengatasi kelemahan tersebut, dapat diterapkan algoritma yang dapat membuat data menjadi bersih, bebas *noise*, dan konsisten (Ariawan, 2019). Salah satu algoritma yang dapat digunakan adalah algoritma *Local Outlier Factor*. Algoritma ini merupakan metode deteksi *outlier* berbasis kepadatan dengan menghitung *local variance* dari titik data yang diberikan (Cheng et al., 2019). Metode ini cocok untuk deteksi *outlier* pada dataset dengan distribusi yang tidak merata serta penentuan outlier berdasarkan kepadatan antara setiap titik data dengan tetangganya (Cheng et al., 2019).

Dalam penelitian ini, dibutuhkan data untuk menguji penambahan metode *Local Outlier Factor* dalam mengatasi kelemahan pada *K-Means*. Data yang digunakan adalah data performa akademik mahasiswa di UPN “Veteran” Yogyakarta. Data ini digunakan karena pada proses *clustering*, *K-Means* menggunakan sistem partisi dan menghitung jarak diantara data. Oleh karena itu, dibutuhkan data dengan tipe numerik seperti atribut indeks prestasi dan golongan Uang Kuliah Tunggal (UKT) pada data performa akademik. Dikarenakan data yang digunakan bertipe kategorikal, sehingga diperlukan perubahan data menjadi tipe numerik seperti atribut data organisasi dan pekerjaan. Data tersebut akan diklasterisasi menjadi beberapa *cluster* yang nantinya akan terbentuk sebuah *cluster warning*. *Cluster* ini berisikan mahasiswa yang membutuhkan prioritas lebih dalam proses *monitoring*.

Dengan begitu, penelitian ini akan menerapkan metode *Local Outlier Factor* (LOF) dengan tujuan untuk mengatasi kekurangan pada algoritma *K-Means* yang sangat dipengaruhi oleh *noise* atau anomali *outlier* pada data. Sehingga diharapkan gabungan kedua metode ini dapat menghasilkan data yang bersih, struktur yang baik, serta membuat kinerja algoritma *K-Means* menjadi lebih baik daripada hanya menggunakan algoritma *K-Means*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka dapat dirumuskan masalah sebagai berikut,

1. *K-Means sensitive* terhadap *noise* atau anomali *outlier* pada data.
2. Bagaimana hasil struktur *cluster* setelah *K-Means* dioptimasi dengan *Local Outlier Factor*?
3. Bagaimana analisis perbandingan performa antara algoritma *K-Means* dan algoritma *K-Means* yang telah dioptimasi dengan *Local Outlier Factor*?
4. Mencari *cluster warning* yang dapat membantu proses monitoring performa akademik mahasiswa.

## 1.3 Batasan Masalah

Batasan masalah pada penelitian ini yaitu:

1. Data yang digunakan adalah data hasil kuisioner dari sampel mahasiswa Program Studi Informatika dan Sistem Informasi UPN “Veteran” Yogyakarta tahun 2017 – 2019 berjumlah 210 data.
2. Kriteria yang digunakan adalah Indeks Prestasi Semester (IPS) genap tahun ajaran 2020/2021, keikutsertaan dalam organisasi serta status pekerjaan.

## 1.4 Tujuan Penelitian

Tujuan dari penelitian ini yaitu:

1. Penambahan algoritma *Local Outlier Factor* yang dapat mengatasi masalah anomali *outlier* pada *K-Means*.

2. Mendapatkan hasil struktur *cluster* setelah K-Means dioptimasi dengan *Local Outlier Factor*.
3. Melakukan perbandingan antara algoritma K-Means dan algoritma K-Means yang dioptimasi dengan LOF dari segi pengelompokkan data.
4. Mendapatkan *cluster warning* yang dapat membantu proses monitoring mahasiswa.

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah dapat digunakan oleh Universitas Pembangunan Nasional Veteran Yogyakarta untuk membantu monitoring hasil akademik mahasiswa.

## 1.6 Tahapan Penelitian

Tahapan penelitian merupakan teknik yang dilakukan dalam membuat tahapan penelitian yang dibagi ke dalam beberapa tahapan yang memiliki fokus target pencapaian masing – masing.

### 1.6.1 Metodologi Penelitian

Tahapan-tahapan pada penelitian yang akan dilakukan ini sebagai berikut:

#### 1. Identifikasi Masalah

Tahapan pertama dari penelitian ini adalah identifikasi masalah. Tahapan ini bertujuan untuk mendapatkan informasi yang akan berguna untuk penelitian ini. Informasi yang akan digunakan sebagai dasar dan pendukung bersumber dari buku, jurnal, dan penelitian sebelumnya yang memiliki keterkaitan dengan penelitian yang akan dikerjakan.

#### 2. Pengumpulan data

Tahapan selanjutnya dari penelitian ini adalah pengumpulan data. Data yang digunakan dalam penelitian ini adalah data primer yang didapatkan melalui penyebaran kuisioner yang berisi beberapa pertanyaan yang dapat berguna untuk penelitian ini. Data tersebut berupa Indeks Prestasi Semester (IPS) genap tahun ajaran 2020/2021, keikutsertaan organisasi dan pekerjaan.

#### 3. *Preprocessing Data*

Data penelitian yang sudah diperoleh dari tahap sebelumnya akan dilakukan preprocessing berupa pemilahan dan pembersihan data seperti penghapusan nama, nim, serta email.

#### 4. *Outlier Detection*

Dari hasil seleksi data yang sudah diperoleh, akan dilakukan deteksi dan pembersihan data outlier dengan algoritma *Local Outlier Factor* pada data yang tersedia.

## 5. Data Clustering

Setelah didapatkan data yang bersih dan bebas *noise*, proses selanjutnya dapat dilakukan pelatihan data untuk mengelompokkan data menjadi beberapa *cluster* dengan menggunakan algoritma *K-Means*.

## 6. Model Evaluation

Dilakukan evaluasi terhadap performa model dengan mencari nilai *Silhouette Score* serta melihat analisis dari penggunaan nilai K terbaik pada model yang telah dibangun dengan algoritma *K-Means*

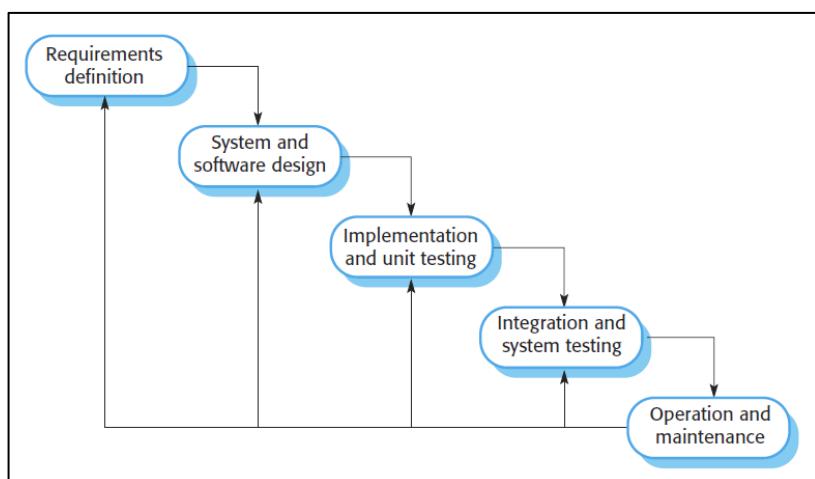
## 7. Deployment

Setelah model telah berhasil diimplementasikan, langkah selanjutnya adalah tahapan *deployment*. Sistem yang dibangun ini akan dikembangkan dan akan di-deploy dalam bentuk *website*. Setelah itu, akan dilakukan pengujian sistem dengan menggunakan metode pengujian *black box*.

## 8. Hasil Penelitian

Setelah seluruh tahapan sudah dilakukan, maka akan didapatkan kesimpulan berdasarkan hasil penelitian mengenai performa akademik mahasiswa dengan metode *K-Means* yang dioptimasi oleh algoritma *Local Outlier Factor*.

### 1.6.2 Metodologi Pengembangan Sistem



Gambar 1.1 Waterfall Process (Sumber: Sommerville, 2016)

Pada tahap pengembangan sistem, model yang digunakan yaitu model Waterfall. Model waterfall atau biasa disebut dengan classic life cycle menggunakan pendekatan sistematis dan berurutan dalam mengembangkan perangkat lunak (Sommerville, 2016). Berdasarkan referensi yang digunakan, metode Waterfall terdiri dari lima tahapan sebagai berikut :

#### 1. Requirements analysis and definition

Tahap ini adalah merupakan tahapan untuk menentukan karakteristik, kendala, dan tujuan dari suatu sistem melalui konsultasi dengan pengguna sistem. Semua ini didefinisikan dengan baik sehingga dapat berfungsi sebagai spesifikasi sistem.

## 2. *System and software design*

Pada tahap ini akan dilakukan perancangan arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Perancangan membantu dalam menentukan perangkat keras (hardware) dan sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

## 3. *Implementation and Unit Testing*

Pada tahap ini akan dilakukan pembangunan sistem dari hasil perancangan menggunakan Bahasa koding yang akan digunakan. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit testing.

## 4. *Intergration and system testing*

Dalam tahap Integration and System Testing ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

## 5. *Operation and maintenance*

Tahapan Operation and Maintenance ini, sistem sudah dapat mulai digunakan. Selain itu juga memperbaiki error yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

## 1.7 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam menyusun laporan penelitian ini yaitu sebagai berikut:

### 1. BAB I Pendahuluan

Menjelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, tahapan penelitian, dan sistematika penulisan terkait klasterisasi performa akademik mahasiswa. Bab ini membahas mengenai masalah dari *K-Means* berupa anomali pada data dengan tujuan untuk menerapkan metode *Local Outlier Factor* pada *K-Means* dalam mengatasi outlier data pada data performa akademik mahasiswa.

### 2. BAB II Tinjauan Literatur

Menjelaskan mengenai teori-teori metode *K-Means* dan *Local Outlier Factor* yang mendasari penelitian secara terperinci yang memuat tentang landasan teori yang akan dibahas pada penelitian, hasil dari penelitian sebelumnya, dan *gap research* yang akan dilakukan.

### 3. BAB III Metodologi Penelitian

Menjelaskan mengenai metode yang akan digunakan pada penelitian ini dalam menyelesaikan masalah yang diangkat terdiri dari desain penelitian, jenis dan sumber data, prosedur pengumpulan data, dan teknik analisis.

**4. BAB IV Hasil dan Pembahasan**

Menjelaskan mengenai analisis dan pembahasan dari hasil yang didapatkan pada sistem yang telah dibangun. Hasil penelitian mencakup tampilan dari program dan hasil dari klasifikasi yang dibuat berdasarkan rancangan yang ada pada metodologi penelitian.

**5. BAB V Kesimpulan dan Saran**

Menjelaskan mengenai kesimpulan yang didapatkan dari hasil dan saran pada penelitian ini yang dapat digunakan untuk bekal pada penelitian selanjutnya.

## **BAB II**

### **TINJAUAN LITERATUR**

#### **2.1 Landasan Teori**

##### **2.1.1 *Machine Learning***

Pembelajaran mesin adalah cabang dari *computer science* yang secara luas bertujuan untuk memungkinkan komputer “belajar” tanpa diprogram secara langsung (Bi et al., 2019). Cabang ini akan terus berkembang dan bertujuan untuk meniru kecerdasan manusia dengan belajar dari lingkungan (el Naqa & Murphy, 2015). Kemajuan dalam ML telah memungkinkan munculnya sistem cerdas baru-baru ini dengan kapasitas kognitif seperti manusia yang menembus bisnis dan kehidupan pribadi kita (Janiesch et al., 2021). Metode pembelajaran mesin telah digunakan secara efektif di berbagai industri, termasuk perbankan, hiburan, biomedis, pengenalan pola, visi komputer, teknik pesawat ruang angkasa, dan biologi komputasi (el Naqa & Murphy, 2015). Tujuan pembelajaran mesin adalah untuk meniru bagaimana manusia (dan makhluk hidup lainnya) belajar memproses data sensorik untuk mencapai suatu tujuan (el Naqa & Murphy, 2015).

Terdapat tiga tipe dari *Machine Learning* yang umum digunakan yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Semi-Supervised Learning*. *Supervised Learning* adalah sebuah metode pembelajaran di mana setiap contoh pelatihan dari data input dipasangkan dengan label klasifikasi yang diketahui. Hal ini memungkinkan model untuk menangani persamaan dan perbedaan ketika objek yang akan diklasifikasikan memiliki banyak sifat variabel dalam kelas mereka sendiri tetapi masih memiliki kualitas mendasar yang mengidentifikasi mereka (el Naqa & Murphy, 2015). Berbeda dengan *Unsupervised Learning*, metode ini berusaha mengungkap korelasi dan kelompok alami di antara data tanpa memperhatikan hasil apa pun atau "jawaban yang benar" (Bi et al., 2019). Oleh karena itu, model menganalisis data dengan menentukan seberapa dekat polanya dengan data yang sudah ada sebelumnya (el Naqa & Murphy, 2015). Metode terakhir adalah *Semi-Supervised Learning* yang dimana metode ini menyesuaikan model dengan data berlabel dan tidak berlabel. Pelabelan data seringkali memakan waktu dan mahal, terutama untuk kumpulan data yang besar (Bi et al., 2019). Dalam skenario seperti itu, bagian berlabel dapat digunakan untuk membantu pembelajaran bagian yang tidak berlabel dengan tujuan meningkatkan kinerja model (el Naqa & Murphy, 2015).

##### **2.1.2 *Clustering***

Clustering adalah metode umum untuk analisis data statistik dalam pembelajaran mesin yang telah digunakan secara luas di berbagai domain dan menempati posisi signifikan dalam unsupervised learning (Shi et al., 2021). Seperti semua masalah lain dari jenis pembelajaran ini melibatkan pengidentifikasiannya struktur dalam kumpulan data yang tidak berlabel. Oleh karena itu, cluster adalah sekelompok objek yang "mirip" satu sama lain dan "berbeda" dengan yang ditemukan di cluster lain. (Madhulatha, 2012).

Algoritma untuk pengelompokan data dapat dibagi menjadi dua yaitu partitional dan hierarchical. Sementara algoritma partitional memutuskan semua klaster sekaligus, teknik hierarchical menemukan klaster berikutnya dengan memanfaatkan klaster yang telah ditetapkan sebelumnya (Madhulatha, 2012). Ada dua jenis algoritma hierarchical: agglomerative (bottom-up) dan divisive (top-down). Algoritme agglomerative menggabungkan elemen ke dalam kluster yang lebih besar secara bertahap dengan memulai setiap elemen sebagai kluster tunggal sedangkan algoritma divisive dimulai dengan seluruh himpunan dan memisahkannya menjadi kelompok-kelompok yang lebih kecil (Madhulatha, 2012).

### 2.1.3 *K-Means*

K-Means adalah teknik data mining yang mengelompokkan data menggunakan skema partisi dan melakukan prosedur pemodelan tanpa pengawasan (unsupervised) (Yunita, 2018). Algoritma ini memberikan setiap titik ke cluster yang pusatnya disebut sebagai centroid terdekat. Pusat adalah rata-rata dari semua titik dalam cluster yaitu, koordinatnya adalah rata-rata aritmatika untuk setiap dimensi secara terpisah atas semua titik dalam cluster (Madhlulatha, 2012). Metode ini adalah pendekatan pengelompokan data yang dikenal baik karena kecepatannya dalam mengelompokkan data (Vhallah et al., 2018), mudah untuk diimplementasikan dan hanya membutuhkan waktu yang singkat untuk dipelajari (Muttaqin & Defriani, 2020). Langkah-langkah dari algoritma K-means secara singkat sebagai berikut.

- 1) Tentukan nilai k sebagai jumlah klaster yang ingin dibentuk.
  - 2) Inisialisasi k sebagai jumlah centroid
  - 3) Hitung jarak setiap data ke masing-masing centroid menggunakan persamaan *Euclidean Distance* yaitu sebagai berikut,

### Keterangan:

d = Jarak

j = Banyak Data

c = Centroid

x = Data

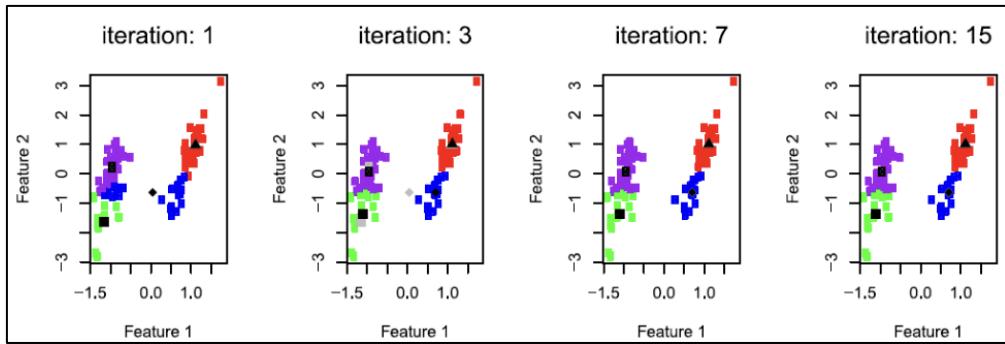
- 4) Kelompokkan setiap data berdasarkan jarak terdekat antara data dengan centroidnya.
  - 5) Tentukan posisi centroid baru sebagai berikut,

## Keterangan:

c = Centroid

x = Data

- 6) Kembali ke langkah 3 jika posisi centroid baru dengan centroid lama tidak sama.



**Gambar 2.1 Ilustrasi dari Algoritma K-Means (Sumber: Muttaqin & Defriani, 2020)**

#### **2.1.4 Local Outlier Factor (LOF)**

Local Outlier Factor menghitung faktor outlier berdasarkan ukuran kepadatan relatif dari setiap titik data relatif terhadap titik sekitarnya, yang disebut sebagai nilai LOF dan digunakan untuk mewakili derajat outlier dalam data (Cheng et al., 2019). Hal ini sangat penting karena data abnormal sendiri tidak dapat dievaluasi secara efektif dan data outlier bukanlah target cluster dalam proses clustering. Anomali dalam suatu data ditentukan dengan menghitung deviasi data terhadap data lain, dan kepadatan terendah data tidak normal (Budiarto et al., 2019).

## Keterangan:

$LOF_{MinPts}(p)$  = Local Outlier Factor dari p

$N_{MinPts}(p)$  = Jumlah minimum point tetangga terdekat data p

$$LRD_{MinPts}(p) = 1 / \left( \frac{\sum_{o \in N_{MinPts}(p)} \text{reach-dist}_{MinPts}(p,o)}{|MinPts(p)|} \right) \dots \quad (2.4)$$

## Keterangan:

$lrd_{MinPts}(p)$  = Local reachability density dari p

**MinPts(p)** = Jumlah minimum titik data p

$reach-dist_{MinPts}(p, o)$  = Jangkauan jarak terjauh dari titik p

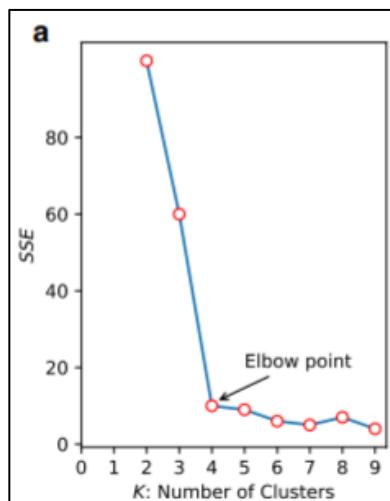
### 2.1.5 *Hyperparameter Tuning*

*Hyperparameter Tuning* adalah parameter tingkat yang lebih tinggi yang diatur secara manual sebelum memulai pelatihan, yang didasarkan pada properti seperti karakteristik data dan kapasitas algoritma untuk dipelajari dalam algoritma *Machine Learning* (Agrawal, 2021). Nilai *learning rate* ( $\alpha$ ) yang besar akan mengubah *loss* secara drastis, sehingga terjadi *overshooting* dan menyebabkan *divergence*, namun jika ditemukan nilai optimal dari  $\alpha$ , maka akan mencapai *convergence* dalam waktu yang lebih singkat dan tanpa *overshoot* (Agrawal, 2021). Oleh karena itu, diperlukannya penyetelan nilai  $\alpha$  yang

paling efisien, penyetelan ini dapat disebut dengan *hyperparameter tuning*. Sehingga *hyperparameter tuning* adalah salah satu aspek penting dalam membangun model yang efisien (Agrawal, 2021).

### 2.1.6 Elbow Method

Metode Elbow adalah metode tertua untuk membedakan dan menganalisis jumlah potensial cluster yang optimal dalam kumpulan data dengan cara menentukan nilai awal K=2 sebagai jumlah optimal awal cluster K, dan kemudian terus meningkatkan K hingga maksimum dengan menaikkan nilai K sebanyak 1 langkah (Shi et al., 2021). Jumlah cluster yang optimal dibedakan berdasarkan fakta sebelum mencapai nilai K. Nilai dari jumlah kuadrat dari Euclidean Distances (SSE) akan cepat menurun seiring bertambahnya nilai K. Nilai yang optimal akan didapatkan apabila selisih nilai SSE hampir tidak berubah seperti yang tertera pada Gambar 2.2.



Gambar 2.2 Elbow Method (Sumber: Shi et al., 2021)

### 2.1.7 Silhouette Method

Metode Silhouette digunakan untuk mengukur seberapa efektif anggota cluster dikelompokkan bersama. Semakin besar koefisien Silhouette, semakin baik pembentukan cluster (B. N. Sari, 2016). Metode yang menggunakan jarak rata-rata antara satu titik data dan lainnya dalam cluster yang sama (cohesion measure) dan jarak rata-rata antara cluster yang berbeda (separation measure) untuk menilai hasil clustering (D. M. Saputra et al., 2020; Shi et al., 2021). Nilai dari pengelompokan sebuah data lebih efektif apabila nilai S lebih dekat ke 1, dan data harus ditempatkan di cluster yang berbeda apabila nilai S lebih dekat ke -1 (Shi et al., 2021).

Tabel 2.1 Tabel Nilai Silhouette

| Nilai Silhouette Coefficient | Struktur          |
|------------------------------|-------------------|
| $0.7 < SC \leq 1$            | Kuat              |
| $0.5 < SC \leq 0.7$          | Sedang            |
| $0.25 < SC \leq 0.5$         | Lemah             |
| $SC \leq 0.25$               | Tidak Terstruktur |

## Keterangan:

a : Jarak rata-rata antara satu titik dan lainnya dalam cluster yang sama

b : Jarak rata-rata antara cluster yang berbeda

## Keterangan:

C = Jumlah data dalam cluster

$d(i, j)$  = Euclidean Distance

$$b(i) = \min_{|C_k|} \sum_{j \in C, i \neq j} d(i, j) \dots \quad (2.7)$$

## Keterangan:

C = Jumlah data dalam cluster

$d(i,j)$  = Euclidean Distance

## 2.2 Studi Literatur

Penelitian mengenai permasalahan performa akademik mahasiswa sudah pernah dilakukan sebelumnya dengan mengimplementasikan berbagai metode. Salah satunya adalah penelitian serupa yang dilakukan oleh Vhallah (2018). Vhallah menerapkan metode K-Means yang digunakan untuk melakukan pengelompokan pada data. Atribut yang digunakan adalah Sistem Kredit Semester Kumulatif, Indeks Prestasi Kumulatif, dan Semester Kumulatif. Penelitian dimulai dengan analisis masalah dan dilanjutkan dengan pengambilan data. Metode pengumpulan data yang digunakan yaitu wawancara dan mengambil data akademik Mahasiswa pada bagian Administrasi Akademik data untuk mahasiswa Angkatan 2014, 2015, 2016 dan 2017 periode semester Ganjil 2017/2018 yang memiliki IPK  $\leq 2,50$  dari 1696 Mahasiswa Aktif. Penelitian ini menghasilkan cluster dari angkatan 2014 berada pada cluster 0 sekitar 30,77% dari sampel, angkatan 2015 berada pada cluster 1 dan 2 sekitar 66,7% sampel, angkatan 2016 berada pada cluster 0 dan 1 sekitar 50% dari sampel, dan angkatan 2017 berada pada cluster 2 sekitar 22,22% dari sampel serta pada angkatan 2017 terdapat 4 mahasiswa yang berpotensi drop out.

Jika pada penelitian sebelumnya Vhallah hanya menggunakan metode K-Means, Ariawan (2019) melakukan penelitian yang sedikit berbeda yaitu dengan menambahkan algoritma Local Outlier Factor (LOF) yang digunakan untuk mendeteksi anomali pada data atau data outlier yang merupakan salah satu kelemahan dari algoritma K-Means. Tujuan dilaksanakan penelitian ini adalah untuk menguji dan membandingkan kinerja dari jumlah iterasi, nilai SSE dan waktu proses pada algoritma K-Means setelah dan sebelum ditambahkan dengan algoritma LOF. Penelitian ini dimulai dengan pengambilan data dan

menggunakan beberapa fitur seperti nomor induk pegawai, nilai perilaku, nilai kehadiran, dan nilai kinerja. Tahapan selanjutnya adalah dilakukan proses deteksi outlier dengan menentukan nilai minpts, jumlah tetangga terdekat untuk menghitung nilai outlier, dan nilai batas untuk memilih suatu anomali pada data. Hasil dari penelitian ini adalah metode Local Outlier Factor dapat menghasilkan hasil dari segi jumlah iterasi, waktu proses, dan kualitas cluster menjadi lebih baik dan dapat menemukan data outlier sebesar 22,8%.

Lain halnya dengan penelitian yang dilakukan oleh Ridwan (2013), Ridwan menggunakan algoritma Naïve Bayes Classifier (NBC) untuk mengetahui evaluasi kinerja akademik dari mahasiswa. Tahapan awal yang dilakukan adalah pengambilan data penelitian yang berupa data akademik mahasiswa Angkatan 2005-2009 yang telah dinyatakan lulus. Penelitian ini menggunakan data yang dibagi menjadi tiga jenis yaitu data training dan testing, data target, serta data riwayat kuliah. Data training dan testing yang digunakan yaitu data dengan atribut NIM, jenis kelamin, asal sekolah, jalur masuk, nilai ujian nasional, gaji orang tua, IP semester 1-4, IPK semester 1-4, dan keterangan lulus. Selain itu, data target yang digunakan yaitu sampel data mahasiswa angkatan 2010-2011 yang diasumsikan belum lulus. Pengujian pada data mahasiswa dengan algoritma NBC menghasilkan nilai precision, recall, dan accuracy masing-masing 83%, 50%, dan 70%. Nilai dari persentase tingkat precision, recall, dan accuracy juga sangat dipengaruhi oleh penentuan data training karena pola data training tersebut akan dijadikan sebagai rule untuk menentukan kelas.

Selain penggunaan algoritma K-Means, penggabungan K-Means dengan Local Outlier Factor, dan Naïve Bayes Classifier. Penelitian yang dilaksanakan oleh Himawan (2014) menggunakan Iterative Dichotomiser 3 (ID3) untuk mengklasifikasi tingkat kelulusan mahasiswa. Penelitian ini dimulai dengan pengumpulan data mahasiswa Angkatan 2008, 2009, dan 2010. Proses klasifikasi dilakukan dari node paling atas yaitu akar pohon (root). Dilanjutkan ke bawah melalui cabang-cabang sampai dihasilkan node daun (leafes) dimana node daun ini menunjukkan hasil akhir klasifikasi. Sebuah objek yang diklasifikasi dalam pohon harus dites nilai entropinya, dimana entropi adalah ukuran dari teori informasi yang dapat mengetahui karakteristik impurity dan homogeneity dari kumpulan data. Penelitian ini menghasilkan kesimpulan dimana semakin sedikit data training yang digunakan maka hasil klasifikasinya menunjukkan ketidakakuratan yang tinggi. Sebaliknya jika data training yang digunakan semakin banyak hasilnya akan semakin akurat.

Dari penelitian-penelitian yang telah dijabarkan sebelumnya, maka dapat diringkas menjadi suatu tabel state of the art yang dapat dilihat pada Tabel 2.2 berikut ini.

**Tabel 2.2 Tabel State of The Art**

| No | Penulis                            | Judul  | Metode                                  | Hasil  |
|----|------------------------------------|--|---|--|
| 1. | (Ariawan, 2019)                    | Optimasi Pengelompokan Data Pada Metode K-Means dengan Analisis Outlier  | K-Means dan <i>Local Outlier Factor</i> | Saat mengelompokkan data kinerja pegawai dengan metode K-Means, metode <i>Local Outlier Factor</i> dapat menghasilkan hasil yang lebih baik dari segi jumlah iterasi, waktu proses, dan kualitas cluster. Dapat menemukan data outlier sebesar 22,8%.  |
| 2. | (Vhallah et al., 2018)             | Pengelompokan Mahasiswa Potensial Drop Out menggunakan Metode Clustering K-Means                                     | K-Means                                 | Hasil dari <i>clustering</i> angkatan 2014 berada pada cluster 0 sekitar 30,77% dari sampel, angkatan 2015 berada pada cluster 1 dan 2 sekitar 66,7% sampel, angkatan 2016 berada pada cluster 0 dan 1 sekitar 50% dari sampel, dan angkatan 2017 berada pada cluster 2 sekitar 22,22% dari sampel serta pada angkatan 2017 terdapat 4 mahasiswa yang berpotensi drop out. |
| 3. | (V. N. Sari et al., 2018)          | Penerapan Metode K-Means Clustering dalam Menentukan Predikat Kelulusan Mahasiswa Untuk Menganalisa Kualitas Lulusan | K-Means                                 | Kelompok mahasiswa yang memiliki nilai rata-rata IPK tertinggi dari ketiga cluster terdapat pada cluster 2 dengan rata-rata IPK 3.3967 sehingga dapat disimpulkan bahwa cluster 2 merupakan lulusan mahasiswa yang memiliki kualitas terbaik.  |
| 4. | (Ridwan et al., 2013)              | Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier         | <i>Naive Bayes Classifier</i> (NBC)     | Pengujian pada data mahasiswa dengan NBC menghasilkan nilai <i>precision</i> , <i>recall</i> , dan <i>accuracy</i> masing-masing 83%, 50%, dan 70%. Nilai tersebut juga sangat dipengaruhi oleh penentuan data training karena pola tersebut akan dijadikan sebagai <i>rule</i> untuk menentukan kelas.  |
| 5. | (Sulistiyawati & Supriyanto, 2021) | Implementasi Algoritma K-Means Clustering dalam Penetuan Siswa Kelas Unggulan  | K-Means                                 | Implementasi algoritma <i>clustering</i> k-means memberikan hasil pengelompokan data yang efektif. Hasil dari klaster 6 kelas yang diujikan sebanyak 192, jumlah siswa yang masuk kelas unggulan sebanyak 96 dan siswa yang tidak masuk sebanyak 96.   |

**Tabel 2.3 Tabel State of The Art Lanjutan**

| No | Penulis                     | Judul  | Metode  | Hasil  |
|----|-----------------------------|--|---|--|
| 6. | (Himawan, 2014)             | Aplikasi Data Mining Menggunakan Algoritma ID3 Untuk Mengklasifikasi Kelulusan Mahasiswa Pada Universitas Dian Nuswantoro Semarang | <i>Iterative Dichotomiser 3 (ID3)</i>         | Semakin sedikit data training yang digunakan maka hasil klasifikasinya menunjukkan ketidakakuratan yang tinggi. Sebaliknya jika data training yang digunakan semakin banyak hasilnya akan semakin akurat.  |
| 7. | (Pradnyana & Permana, 2018) | Sistem Pembagian Kelas Kuliah Mahasiswa Dengan Metode K-Means dan K-Nearest Neighbors Untuk Meningkatkan Kualitas Pembelajaran     | <i>K-Means</i> dan <i>K-Nearest Neighbors</i> | Jumlah cluster dan jumlah data yang digunakan mempengaruhi kualitas cluster yang dibentuk oleh metode yang digunakan. Kualitas cluster dinilai berdasarkan nilai <i>Silhouette Indeks</i> . Nilai <i>Silhouette Indeks</i> tertinggi diperoleh saat menggunakan 100 data dengan jumlah cluster 10 sebesar 0,534 ( <i>medium structure</i> ). |
| 8. | (Kurniadi & Sugiyono, 2020) | Pengelompokan Data Akademik Menggunakan Algoritma K-Means Pada Data Akademik Unissula  | K-Means                                       | Algoritma K-Means menghasilkan 3 cluster dengan masing-masing anggota pada cluster 1 terdapat 31% anggota, pada cluster 2 terdapat 53% anggota dan cluster 3 terdapat 16% anggota. Sistem ini mampu menampilkan jumlah siswa yang berisiko <i>drop out</i> (DO) dan jumlah mahasiswa yang perlu diperhatikan (warning)                       |

Berdasarkan penelitian-penelitian sebelumnya, penelitian ini akan memiliki kesamaan dalam hal objek penelitian yang dilakukan yaitu klasterisasi performa akademik mahasiswa. Perbedaan antara penelitian ini dengan penelitian sebelumnya terdapat pada metode, parameter atau atribut, serta lokasi studi kasus yang digunakan.

Berbagai perbedaan dengan penelitian sebelumnya terdapat dalam beberapa hal. Perbedaan pertama terdapat dalam metode yang digunakan seperti *K-Means* (Budiarto et al., 2019; Nur et al., 2018; Rahmawati et al., 2019; Rosadi et al., 2016; Sulistiawati & Supriyanto, 2021; Vhallah et al., 2018), *K-Nearest Neighbors* (Pradnyana & Permana, 2018; Rustam & Annur, 2019), *Naive Bayes Classifier (NBC)* (Ridwan et al., 2013; H. K. Saputra, 2018), dan *Iterative Dichotomiser 3 (ID3)* (Amalia & Naf'an, 2017; Himawan, 2014).

Selain terletak pada metode yang digunakan, perbedaan juga terdapat pada parameter atau atribut yang digunakan. Penelitian yang dilakukan oleh Vhallah (2018) menggunakan SKS Kumulatif, Indeks Prestasi Kumulatif, dan Semester Kumulatif dari

mahasiswa Angkatan 2014, 2015, 2016 dan 2017 periode semester Ganjil 2017/2018 yang memiliki IPK  $\leq 2,50$ . Sedangkan data penelitian yang digunakan oleh Ariawan (2019) meliputi nomor induk pegawai, nilai perilaku, nilai kehadiran, dan nilai kinerja. Perbedaan terakhir adalah lokasi studi kasus penelitian yang bertempat di Jurusan Informatika UPN “Veteran” Yogyakarta.

Dari beberapa penelitian sebelumnya, untuk mendapatkan hasil performa akademik mahasiswa masih terdapat beberapa masalah seperti data yang masih dipengaruhi oleh pemilihan data saat melakukan proses *training* (Ridwan et al., 2013), jumlah data yang digunakan (Himawan, 2014), serta sensitif terhadap *noise* atau anomali pada data (Ariawan, 2019). Dari hal tersebut, penelitian ini menggunakan metode *K-Means* yang dinilai dapat membantu proses klasterisasi karena metode ini mampu mencari kombinasi antar variabel (Rahmawati et al., 2019) serta dapat menemukan karakteristik-karakteristik dari sebuah objek (Vhallah et al., 2018).

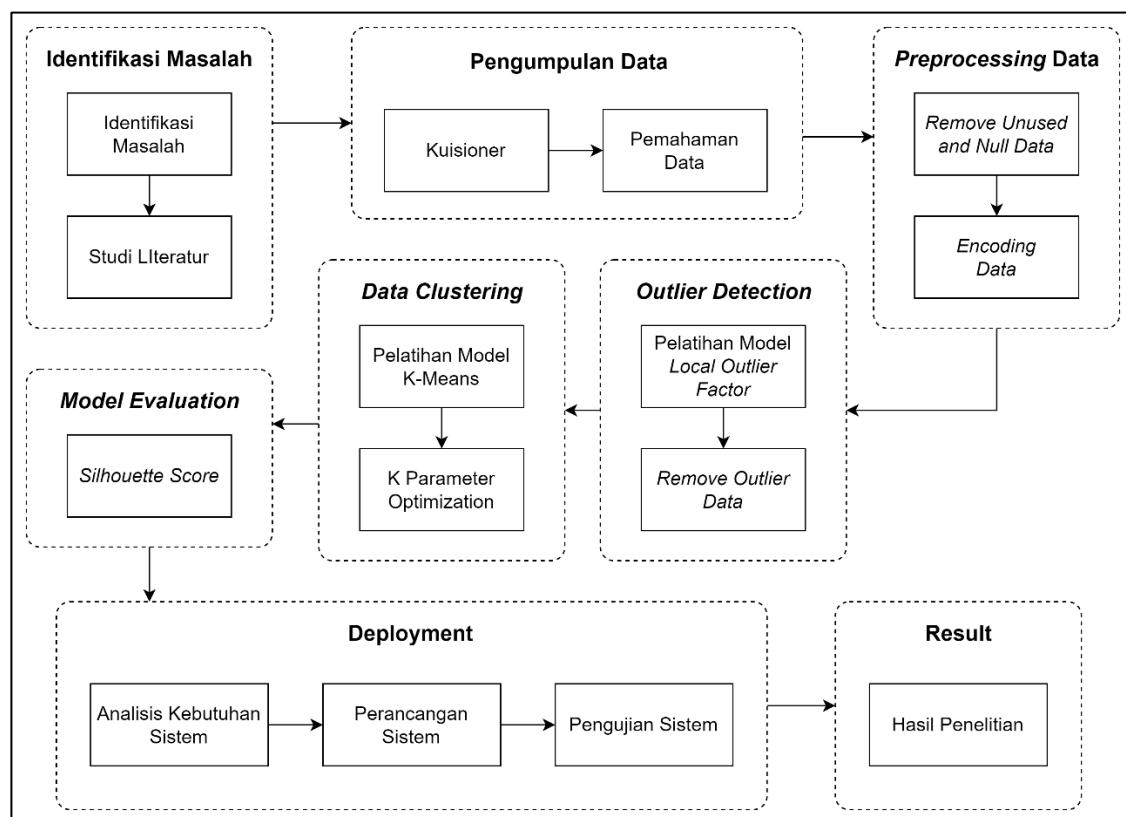
Namun, metode K-Means sendiri masih memiliki kelemahan terhadap *noise* atau anomali pada data (Ariawan, 2019). Oleh karena itu, diperlukan penerapan algoritma yang dapat membuat data menjadi bersih, bebas *noise*, serta konsisten (Ariawan, 2019). Salah satu algoritma yang dapat digunakan adalah algoritma *Local Outlier Factor*. Metode ini cocok untuk deteksi *outlier* pada dataset dengan distribusi yang tidak merata serta penentuan outlier berdasarkan kepadatan antara setiap titik data dengan tetangganya (Cheng et al., 2019). Penerapan dari metode ini telah teruji dari penelitian yang dilaksanakan oleh Alghushairy (2020) dan Ariawan (2019) yang menghasilkan *cluster* yang lebih baik dari segi jumlah iterasi, waktu proses, dan kualitas cluster serta dapat menemukan data outlier sebesar 22,8%.

## BAB III

### METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

#### 3.1 Metodologi Penelitian

Pada bagian ini akan dibahas mengenai metodologi yang akan dilakukan dalam penelitian ini, yaitu untuk melakukan optimasi pada *K-Means* dengan *Local Outlier Factor* untuk klasterisasi performa akademik mahasiswa. Penelitian ini akan dilakukan dengan menggunakan metode kuantitatif dengan *dataset* yang digunakan berupa data primer berdasarkan hasil dari kuisioner yang sudah dilakukan sebelumnya. Metode kuantitatif adalah pendekatan penelitian yang berkonsentrasi pada evaluasi data numerik dengan teknik statistik yang lebih didasarkan pada pengujian statistik dan hipotesis daripada penalaran ilmiah (Hardani et al., 2020). Tahapan penelitian dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 Alur Tahapan Penelitian

##### 3.1.1 Identifikasi Masalah

Penelitian ini dimulai dengan melakukan identifikasi masalah yang dilakukan untuk mencari masalah yang akan diangkat pada penelitian ini. Permasalahan yang diangkat dapat bersumber dari sebuah algoritma dan juga dunia nyata. Pada penelitian ini, masalah algoritma yang diangkat adalah masalah pada algoritma *K-Means* yang memiliki kelemahan pada data *outlier* saat melakukan proses pengelompokan pada data mahasiswa.

### **3.1.2 Studi Literatur**

Studi literatur dilakukan untuk mencari informasi yang akan digunakan sebagai dasar ataupun pendukung pada penelitian ini. Tahapan ini dilakukan dengan mencari dari beberapa sumber seperti beberapa buku, jurnal, dan penelitian sebelumnya yang relevan dengan penelitian ini. Informasi yang didapatkan digunakan sebagai dasar dalam penyelesaian masalah penelitian yang diangkat. Informasi mengenai penelitian-penelitian terdahulu dapat dilihat pada Tabel 2.2 Tabel *State of The Art*.

### **3.1.3 Pengumpulan Data**

#### **1) Kuisioner**

Data yang digunakan dalam penelitian ini merupakan data primer yang didapatkan melalui proses penyebaran kuisioner mulai tanggal 23 September 2022 hingga 14 Oktober 2022 dengan target responden adalah mahasiswa Informatika dan Sistem Informasi UPN Veteran Yogyakarta Angkatan 2017 hingga 2019. Kuisioner yang dilakukan dapat diakses melalui tautan berikut <https://s.id/SkripsiAnjar>. Data yang didapatkan berjumlah 210 data dimana data tersebut dinilai dapat mencakup kinerja atau performa akademik dari mahasiswa. Untuk pertanyaan yang diajukan dapat dilihat pada Tabel 3.1 berikut ini.

**Tabel 3.1 Daftar Pertanyaan yang Diajukan**

| No | Pertanyaan                         | Deskripsi   | Keterangan   |
|----|------------------------------------|---|--|
| 1. | Nama Lengkap                       | Nama lengkap dari responden   | Berupa data dengan tipe <i>string</i>  |
| 2. | Nomor Induk Mahasiswa (NIM)        | Nomor induk mahasiswa dari responden  | Berupa data dengan tipe <i>integer</i> dengan panjang karakter 9                                   |
| 3. | Angkatan                           | Angkatan dari responden   | Berupa data dengan tipe <i>integer</i> dengan rentang data 2017 - 2021                             |
| 4. | Golongan Uang Kuliah Tunggal (UKT) | Golongan UKT dari responden   | Berupa data dengan tipe <i>integer</i> dengan rentang data 0 - 9                                   |
| 5. | Indeks Prestasi Semester (IPS)     | Nilai akhir yang didapatkan oleh mahasiswa pada semester genap ajaran 2020/2021                   | Berupa data dengan tipe <i>float</i> dimana data tersebut memiliki rentang nilai antara 0 hingga 4 |
| 6. | Keikutsertaan Organisasi           | Status keikutsertaan mahasiswa dalam sebuah organisasi pada semester genap tahun ajaran 2020/2021 | Berupa data dengan tipe <i>string</i> berupa pilihan Ya atau Tidak                                 |
| 7. | Status Pekerjaan                   | Status pekerjaan mahasiswa pada semester genap ajaran 2020/2021                                   | Berupa data dengan tipe <i>string</i> berupa pilihan Ya atau Tidak                                 |

Penyebaran kuisioner ini dinilai perlu untuk dilakukan karena data yang diambil adalah data-data aktivitas non-akademik dari mahasiswa seperti keikutsertaan dalam organisasi dan status pekerjaan.

**Gambar 3.2 Kuisioner**

## 2) Pemahaman Data

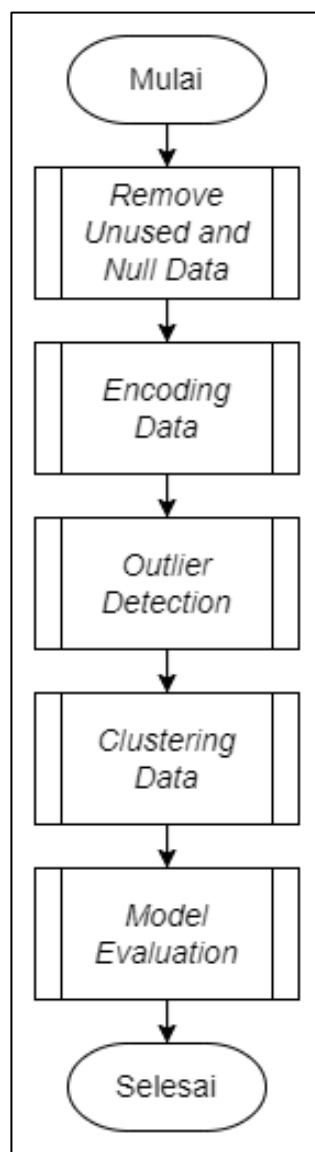
Parameter yang digunakan pada penelitian ini terdiri dari data Indeks Prestasi Semester (IPS) genap tahun ajaran 2020/2021, keikutsertaan dalam organisasi serta status pekerjaan yang akan digunakan pada algoritma K-Means. Detail dari setiap parameter yang digunakan dapat dilihat pada Tabel 3.2 berikut.

**Tabel 3.2 Parameter yang digunakan pada penelitian**

| No | Parameter                      | Deskripsi   | Keterangan  |
|----|--------------------------------|---|---|
| 1. | Indeks Prestasi Semester (IPS) | Nilai akhir yang didapatkan oleh mahasiswa pada semester genap tahun ajaran 2020/2021             | Berupa data dengan tipe <i>float</i> . Statistik Data:<br>a. Jumlah: 210<br>b. Rata-Rata: 3.49<br>c. Standar Deviasi: 0.41<br>d. Nilai minimal: 1.62<br>e. Nilai Maksimal: 4.00<br>f. Modus: 4.00 |
| 2. | Keikutsertaan Organisasi       | Status keikutsertaan mahasiswa dalam sebuah organisasi pada semester genap tahun ajaran 2020/2021 | Berupa data dengan tipe <i>string</i> berupa pilihan Ya atau Tidak  |
| 3. | Status Pekerjaan               | Status pekerjaan mahasiswa pada semester genap dan ajaran 2020/2021                               | Berupa data dengan tipe <i>string</i> berupa pilihan Ya atau Tidak  |

### 3.1.4 Preprocessing Data

Data yang telah didapatkan dari kuisioner masih perlu dilakukan proses terlebih dahulu sebelum digunakan dalam pembuatan serta pelatihan model. Beberapa proses yang dapat dilakukan seperti menghapus data-data yang tidak diperlukan, menghapus data kosong, dan *encoding* data. Alur tahapan *preprocessing* data dapat dilihat pada *flowchart* utama Gambar 3.3.

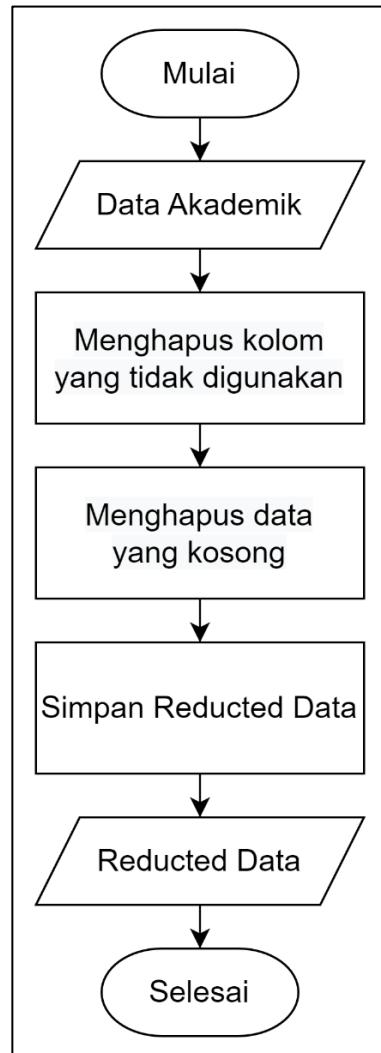


Gambar 3.3 Flowchart Utama

Berdasarkan *flowchart* utama pada Gambar 3.3, terdapat lima buah proses utama yaitu *removing unused and null data*, *encoding data*, *outlier detection*, *clustering data*, dan *model evaluation*. Proses-proses tersebut dapat diturunkan menjadi proses yang lebih detail dan dapat dijelaskan lebih lanjut sebagai berikut.

### 1) Remove Unused and Null Data

Proses pada tahapan ini secara ringkas dapat dilihat pada *flowchart* Gambar 3.4 berikut.

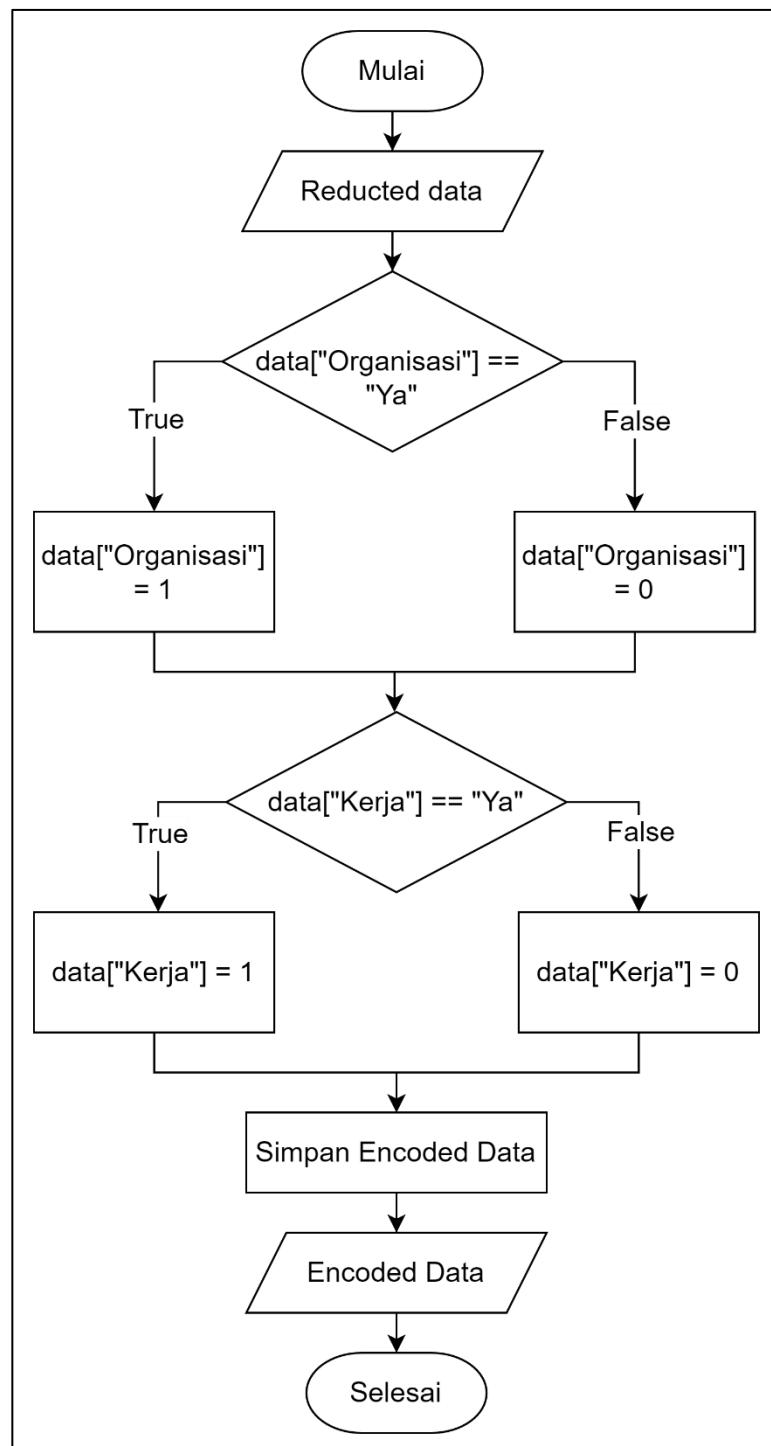


**Gambar 3.4 Flowchart Remove Unused and Null Data**

Proses pertama pada tahapan ini adalah dengan melakukan penyesuaian pada data yang telah didapatkan. Hal ini diperlukan karena tidak semua kolom pada dataset akan digunakan dalam pembuatan model. Penyesuaian yang dilakukan meliputi penghapusan kolom-kolom yang tidak akan digunakan seperti timestamp, email, nama lengkap, Nomor Induk Mahasiswa (NIM), dan angkatan. Selain penghapusan kolom, penghapusan data juga diperlukan karena terdapat beberapa data yang memiliki nilai kosong (*null value*). Setelah proses ini data yang telah bersih akan disimpan.

### 2) Encoding Data

Proses pada tahapan ini secara ringkas dapat dilihat pada *flowchart* Gambar 3.5 berikut.

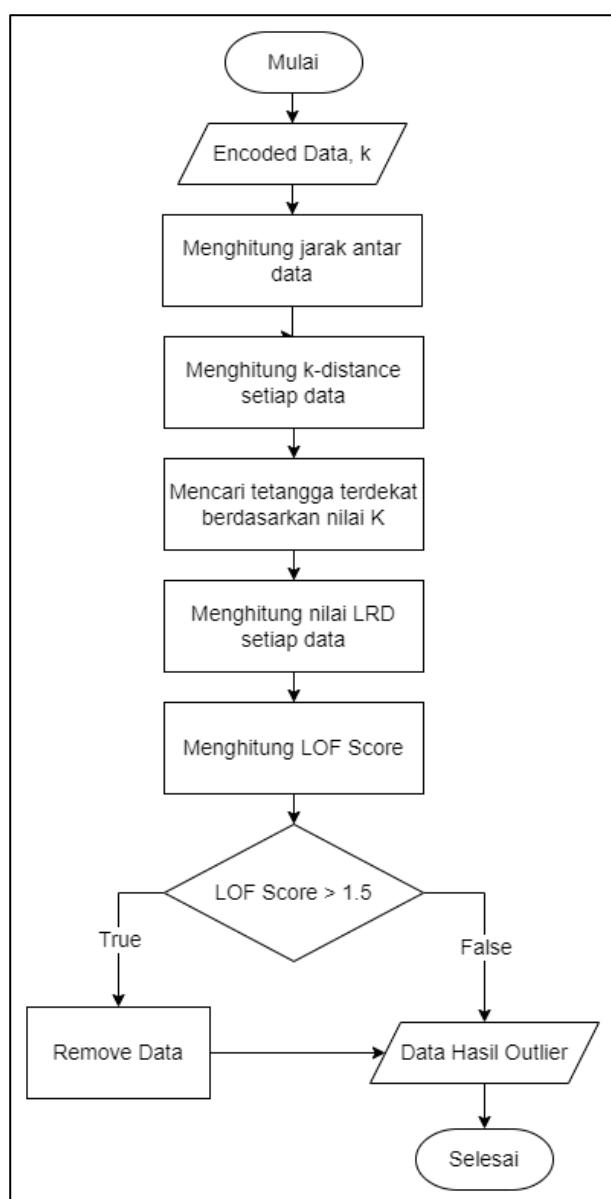


**Gambar 3.5 Flowchart Encoding Data**

Setelah dilakukan penghapusan data, akan dilakukan proses *encoding* pada data. Proses ini diperlukan karena pada saat pembuatan model dari algoritma yang digunakan memerlukan data dalam bentuk numerik. Data yang diubah terdapat pada parameter organisasi dan pekerjaan yang memiliki *value* “Tidak” dan “Ya”. Parameter yang memiliki *value* “Tidak” akan diubah menjadi angka 0 dan parameter yang memiliki *value* “Ya” akan diubah menjadi angka 1. Setelah proses ini, data akan disimpan dan dilanjutkan ke proses selanjutnya.

### 3.1.5 Outlier Detection

Data yang telah dilakukan proses *encoding data* akan digunakan untuk melakukan proses pendekripsi *outlier*. Pada proses ini akan menggunakan algoritma *Local Outlier Factor*. Proses dari algoritma ini adalah dengan menghitung jarak dan *density* setiap data dan mencari data yang berada diluar jangkauan. Proses akhir dari algoritma ini akan menghasilkan nilai yang dapat digunakan untuk mendekripsi apakah titik data tersebut termasuk sebagai *outlier*. Untuk sebuah data dapat dikatakan *outlier* atau tidak dapat dilihat dari LOF score dengan acuan berupa batasan (*threshold*) LOF score lebih besar atau sama dengan 1.5. Nilai tersebut didapatkan dari hasil percobaan pada penelitian sebelumnya yang dilakukan oleh Breunig (2000). Proses pada tahapan ini secara ringkas dapat dilihat pada *flowchart* Gambar 3.6 berikut.



Gambar 3.6 Flowchart Outlier Detection

Proses pertama yang dilakukan adalah dengan menginisialisasikan parameter yang dibutuhkan dalam perhitungan yaitu  $k = 3$ . Dalam melakukan perhitungan ini diperlukan beberapa sampel data seperti Tabel 3.3 berikut ini.

**Tabel 3.3 Tabel Sampel Data**

| Data | IP   | Organisasi | Kerja |
|------|------|------------|-------|
| D1   | 3.94 | 1          | 0     |
| D2   | 3.48 | 0          | 1     |
| D3   | 2.83 | 0          | 0     |
| D4   | 4    | 1          | 0     |
| D5   | 3.77 | 1          | 1     |
| D6   | 3.92 | 0          | 0     |
| D7   | 3.48 | 1          | 0     |
| D8   | 2.75 | 0          | 0     |
| D9   | 3.68 | 0          | 1     |

Langkah selanjutnya adalah menghitung jarak antar masing-masing data yang berada pada Tabel 3.3 dengan menggunakan Persamaan 2.1 sebagai berikut.

$$d(D_1, D_1) = -$$

$$d(D_1, D_2) = \sqrt{(3.94 - 3.48)^2 + (1 - 0)^2 + (0 - 1)^2} = 1.487144916$$

$$d(D_1, D_3) = 1.494021419$$

$$d(D_1, D_4) = 0.06$$

$$d(D_1, D_5) = 1.014347081$$

$$d(D_1, D_6) = 1.00019998$$

$$d(D_1, D_7) = 0.46$$

$$d(D_1, D_8) = 1.554380906$$

$$d(D_1, D_9) = 1.437915157$$

Berikut hasil perhitungan jarak antar data dapat dilihat di Tabel 3.4 ketika semua data telah selesai dalam proses perhitungan.

**Tabel 3.4 Tabel perhitungan jarak antar data**

| O  | X    | Y | Z | A1   | A2   | A3   | A4   | A5   | A6   | A7   | A8   | A9   |
|----|------|---|---|------|------|------|------|------|------|------|------|------|
| D1 | 3.94 | 1 | 0 | -    | 1.49 | 1.49 | 0.06 | 1.01 | 1.00 | 0.46 | 1.55 | 1.44 |
| D2 | 3.48 | 0 | 1 | 1.49 | -    | 1.19 | 1.51 | 1.04 | 1.09 | 1.41 | 1.24 | 0.20 |
| D3 | 2.83 | 0 | 0 | 1.49 | 1.19 | -    | 1.54 | 1.70 | 1.09 | 1.19 | 0.08 | 1.31 |
| D4 | 4    | 1 | 0 | 0.06 | 1.51 | 1.54 | -    | 1.03 | 1.00 | 0.52 | 1.60 | 1.45 |
| D5 | 3.77 | 1 | 1 | 1.01 | 1.04 | 1.70 | 1.03 | -    | 1.42 | 1.04 | 1.74 | 1.00 |
| D6 | 3.92 | 0 | 0 | 1.00 | 1.09 | 1.09 | 1.00 | 1.42 | -    | 1.09 | 1.17 | 1.03 |
| D7 | 3.48 | 1 | 0 | 0.46 | 1.41 | 1.19 | 0.52 | 1.04 | 1.09 | -    | 1.24 | 1.43 |
| D8 | 2.75 | 0 | 0 | 1.55 | 1.24 | 0.08 | 1.60 | 1.74 | 1.17 | 1.24 | -    | 1.37 |
| D9 | 3.68 | 0 | 1 | 1.44 | 0.20 | 1.31 | 1.45 | 1.00 | 1.03 | 1.43 | 1.37 | -    |

Setelah didapatkan jarak antar data, Langkah selanjutnya adalah menghitung *k-distance* pada setiap data dengan cara mencari nilai minimal sesuai dengan nilai parameter k.

$$\text{dist}_3(D_1) = d(D_1, D_6) = 1.0002 \rightarrow d(D_1, D_6) \text{ merupakan tetangga terdekat ketiga}$$

$$\text{dist}_3(D_2) = d(D_2, D_6) = 1.092520023$$

$$\text{dist}_3(D_3) = d(D_3, D_2) = 1.192686044$$

$$\text{dist}_3(D_4) = d(D_4, D_6) = 1.003194896$$

$$\text{dist}_3(D_5) = d(D_5, D_4) = 1.026109156$$

$$\text{dist}_3(D_6) = d(D_6, D_9) = 1.028396811$$

$$\text{dist}_3(D_7) = d(D_7, D_5) = 0.52$$

$$\text{dist}_3(D_8) = d(D_8, D_2) = 1.238103388$$

$$\text{dist}_3(D_9) = d(D_9, D_6) = 1.028396811$$

Setelah didapatkan nilai *k-distance* pada setiap data, Langkah selanjutnya adalah mencari titik tetangga terdekat sesuai dengan nilai parameter k.

$$N_3(D_1) = \{D_4, D_7, D_6\}$$

$$N_3(D_2) = \{D_9, D_5, D_6\}$$

$$N_3(D_3) = \{D_8, D_6, D_2\}$$

$$N_3(D_4) = \{D_1, D_7, D_6\}$$

$$N_3(D_5) = \{D_9, D_1, D_4\}$$

$$N_3(D_6) = \{D_1, D_4, D_9\}$$

$$N_3(D_7) = \{D_1, D_4, D_5\}$$

$$N_3(D_8) = \{D_3, D_6, D_2\}$$

$$N_3(D_9) = \{D_2, D_5, D_6\}$$

Setelah mendapatkan jarak antar data, nilai *k-distance*, dan kelompok titik tetangga terdekat. Proses berikutnya adalah mencari nilai *reach distance* dengan melihat nilai maksimum antara jarak antar titik dan nilai *k-distance*.

$$\text{reachdist}_3(D_4 \leftarrow D_1) = \max\{\text{dist}_3(D_4), d(D_4, D_1)\} = \max \{1.003, 0.06\} = 1.003$$

$$\text{reachdist}_3(D_7 \leftarrow D_1) = 1.041201229$$

$$\text{reachdist}_3(D_6 \leftarrow D_1) = 1.028396811$$

Berikut hasil perhitungan nilai *reach distance* dapat dilihat di Tabel 3.5 ketika semua data telah selesai dalam proses perhitungan.

**Tabel 3.5 Tabel Nilai *reachdist*<sub>3</sub>**

| O  | X    | Y | Z | reachdist <sub>3</sub> (O' ← O) |                          |                          |
|----|------|---|---|---------------------------------|--------------------------|--------------------------|
|    |      |   |   | 1 <sup>st</sup> neighbor        | 2 <sup>nd</sup> neighbor | 3 <sup>rd</sup> neighbor |
| D1 | 3.94 | 1 | 0 | 1.003194896                     | 1.041201229              | 1.028396811              |
| D2 | 3.48 | 0 | 1 | 1.028396811                     | 1.041201229              | 1.092520023              |
| D3 | 2.83 | 0 | 0 | 1.238103388                     | 1.09                     | 1.192686044              |
| D4 | 4    | 1 | 0 | 1.00019998                      | 1.041201229              | 1.028396811              |
| D5 | 3.77 | 1 | 1 | 1.028396811                     | 1.014347081              | 1.026109156              |
| D6 | 3.92 | 0 | 0 | 1.00019998                      | 1.003194896              | 1.028396811              |
| D7 | 3.48 | 1 | 0 | 1.00019998                      | 1.003194896              | 1.041201229              |
| D8 | 2.75 | 0 | 0 | 1.192686044                     | 1.17                     | 1.238103388              |
| D9 | 3.68 | 0 | 1 | 1.092520023                     | 1.026109156              | 1.028396811              |

Setelah didapatkan nilai dari  $\text{reachdist}_k$  seperti pada Tabel 3.5, proses selanjutnya adalah menghitung nilai kepadatan  $\text{LRD}_k$  (*Local Reachability Density*), tingkat kepadatan titik data ke data lain, berdasarkan nilai dari setiap tetangga terdekat dengan menggunakan Persamaan 2.4 sebagai berikut.

$$\text{LRD}_3(D1) = 1 / \left( \frac{\text{reachdist}_3(D_4 \leftarrow D_1) + \text{reachdist}_3(D_7 \leftarrow D_1) + \text{reachdist}_3(D_6 \leftarrow D_1)}{k} \right)$$

$$\text{LRD}_3(D1) = 1 / \left( \frac{1.003194896 + 1.041201229 + 1.028396811}{3} \right) = 0.976310497$$

$$\text{LRD}_3(D_2) = 0.94873118$$

$$\text{LRD}_3(D_3) = 0.85208163$$

$$\text{LRD}_3(D_4) = 0.977262993$$

$$\text{LRD}_3(D_5) = 0.977563915$$

$$\text{LRD}_3(D_6) = 0.989513895$$

$$\text{LRD}_3(D_7) = 0.985352374$$

$$\text{LRD}_3(D_8) = 0.833150634$$

$$\text{LRD}_3(D_9) = 0.953280974$$

Proses terakhir dalam pendeksiian *outlier* adalah dengan menghitung LOF score yang dapat dilakukan dengan menggunakan Persamaan 2.3 sebagai berikut.

$$\text{LOF}_3(D1) = 1 / \left( \frac{\frac{\text{LRD}_3(D_4) + \text{LRD}_3(D_6) + \text{LRD}_3(D_7)}{\text{LRD}_3(D_1)}}{k} \right)$$

$$LOF(D1) = 1 / \left( \frac{\frac{0.977262993 + 0.989513895 + 0.985352374}{0.976310497}}{3} \right) = 1.0079$$

Berikut hasil perhitungan LOF *score* dapat dilihat di Tabel 3.6 ketika semua data telah selesai dalam proses perhitungan.

**Tabel 3.6 Tabel LOF Score**

| Data      | IP          | Organisasi | Kerja    | LOF Score          |
|-----------|-------------|------------|----------|--------------------|
| D1        | 3.94        | 1          | 0        | 1.007920216        |
| D2        | 3.48        | 0          | 1        | 1.026057696        |
| D3        | 2.83        | 0          | 0        | 1.084166747        |
| <b>D4</b> | <b>4</b>    | <b>1</b>   | <b>0</b> | <b>1.506612955</b> |
| D5        | 3.77        | 1          | 1        | 0.991189909        |
| D6        | 3.92        | 0          | 0        | 0.979219689        |
| D7        | 3.48        | 1          | 0        | 0.991569948        |
| <b>D8</b> | <b>2.75</b> | <b>0</b>   | <b>0</b> | <b>1.616375435</b> |
| D9        | 3.68        | 0          | 1        | 1.01956963         |

Jika dilihat hasil perhitungan dari Tabel 3.6, maka dapat disimpulkan dari sampel data yang ada terdeteksi dua buah data yang tergolong sebagai *outlier* yaitu terletak pada data D4 dan D8. Kedua data tersebut dapat digolongkan sebagai data *outlier* karena memiliki LOF *score* lebih dari atau sama dengan 1.5. Untuk *threshold* atau batasan LOF *score* dapat dikatakan sebagai *outlier*. Sehingga kedua data tersebut dapat dihapus untuk hasil proses *clustering* yang lebih optimal.

### 3.1.6 Data Clustering

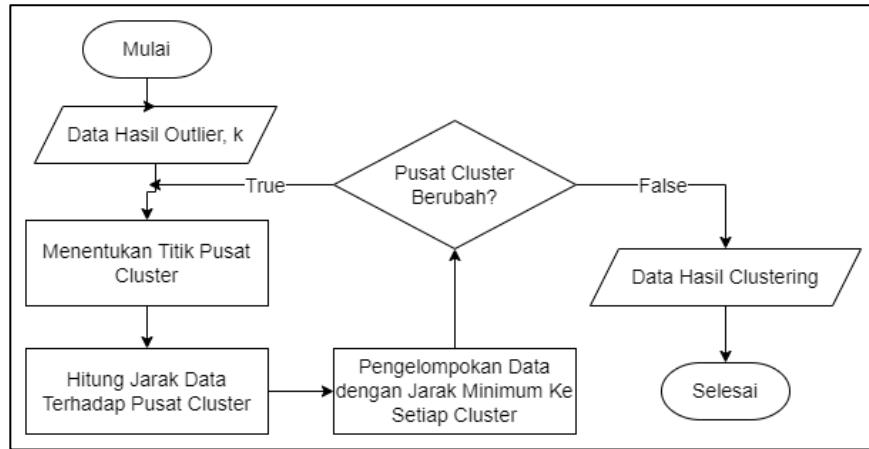
Tahapan ini akan dilakukan pembuatan model *machine learning* dengan melakukan pelatihan terhadap semua data yang telah melalui *preprocessing* dan penghapusan *outlier*. Pada tahapan ini juga dilakukan optimasi terhadap model untuk mencari model dengan performa terbaik dengan cara menguji parameter penting dalam pelatihan model.

#### 1) Pelatihan Model K-Means

Penelitian ini menggunakan *K-Means* dalam pembuatan model *machine learning*. Proses pelatihan pada model *K-Means* menggunakan *dataset* berupa data numerik berjumlah 164 data dengan 3 parameter. Data yang digunakan tersebut merupakan data telah melalui *preprocessing* dan penghapusan *outlier* dengan menggunakan *Local Outlier Factor*.

Proses perhitungan pada saat pelatihan model *K-Means* dilakukan dengan menghitung jarak dari setiap data ke setiap titik pusat *cluster*. Jarak yang telah didapatkan akan digunakan untuk mengelompokkan data sesuai dengan jarak minimal terhadap pusat *cluster*. Proses ini akan menghasilkan data yang telah dikelompokkan sesuai dengan jumlah *cluster* yang sudah

ditentukan sebelumnya. Proses pada tahapan ini dapat dilihat pada *flowchart* Gambar 3.7 berikut.



**Gambar 3.7 Flowchart Clustering Data**

Proses pertama yang dilakukan adalah dengan menginisialisasikan parameter yang dibutuhkan dalam perhitungan yaitu  $k = 3$ . Dalam melakukan perhitungan ini akan digunakan data sampel seperti pada Tabel 3.3. Langkah selanjutnya adalah dengan menentukan titik pusat *cluster* sesuai dengan jumlah parameter  $k$  yang akan digunakan seperti pada Tabel 3.7 berikut ini.

**Tabel 3.7 Tabel Titik Pusat Cluster**

| Cluster | IP   | Organisasi | Kerja |
|---------|------|------------|-------|
| 1       | 3.48 | 0          | 1     |
| 2       | 3.77 | 1          | 1     |
| 3       | 3.68 | 0          | 1     |

Setelah didapatkan titik pusat awal untuk setiap *cluster* selanjutnya dilakukan perhitungan pencarian jarak masung-masing data ke setiap titik pusat cluster dengan Persamaan 2.1 sebagai berikut.

$$d(D_1, C_1) = \sqrt{(D_{1x} - C_{1x})^2 + (D_{1y} - C_{1y})^2 + (D_{1z} - C_{1z})^2}$$

$$d(D_1, C_1) = \sqrt{(3.48 - 3.94)^2 + (0 - 1)^2 + (1 - 0)^2}$$

$$d(D_1, C_1) = 1.487144916$$

$$d(D_1, C_2) = 1.014347081$$

$$d(D_1, C_3) = 1.437915157$$

Dari perhitungan tersebut, dapat dilihat bahwa data pertama masuk kedalam *cluster* kedua. Hal itu dapat terjadi karena diantara tiga jarak tersebut, data tersebut memiliki jarak terpendek jika diukur dari pusat *cluster* kedua yaitu 1.014347081. Ketika semua data telah dihitung, semua hasil dapat dilihat di Tabel 3.8 berikut.

**Tabel 3.8 Tabel Hasil Clustering Iterasi Satu**

| Data | C <sub>1</sub> | C <sub>2</sub> | C <sub>3</sub> | Hasil |
|------|----------------|----------------|----------------|-------|
| D1   | 1.487144916    | 1.014347081    | 1.437915157    | C2    |
| D2   | 0              | 1.041201229    | 0.2            | C1    |
| D3   | 1.192686044    | 1.698116604    | 1.312440475    | C1    |
| D4   | 1.506784656    | 1.026109156    | 1.449965517    | C2    |
| D5   | 1.041201229    | 0              | 1.004041832    | C3    |
| D6   | 1.092520023    | 1.422146265    | 1.028396811    | C3    |
| D7   | 1.414213562    | 1.041201229    | 1.428285686    | C2    |
| D8   | 1.238103388    | 1.743674282    | 1.365613415    | C1    |
| D9   | 0.2            | 1.004041832    | 0              | C3    |

Setelah perhitungan semua jarak dan pengelompokan data, langkah selanjutnya adalah melakukan perhitungan titik pusat cluster baru dengan menggunakan Persamaan 2.2 sebagai berikut.

a. Pusat Cluster 1

**Tabel 3.9 Tabel Pusat Cluster Satu Baru**

| Cluster | IP   | Organisasi | Kerja |
|---------|------|------------|-------|
| D2      | 3.48 | 0          | 1     |
| D3      | 2.83 | 0          | 0     |
| D8      | 2.75 | 0          | 0     |

Pusat Baru IP :

$$C_{1_{IP}} = \frac{3.48+2.83+2.75}{3} = 3.02$$

$$C_{1_{Organisasi}} = \frac{0+0+0}{3} = 0$$

$$C_{1_{Kerja}} = \frac{1+0+0}{3} = 0.33$$

b. Pusat Cluster 2

**Tabel 3.10 Tabel Pusat Cluster Dua Baru**

| Cluster | IP   | Organisasi | Kerja |
|---------|------|------------|-------|
| D1      | 3.94 | 1          | 0     |
| D4      | 4    | 1          | 0     |
| D5      | 3.77 | 1          | 1     |
| D7      | 3.48 | 1          | 0     |

Pusat Baru IP :

$$C_{2_{IP}} = \frac{3.94+4+3.77+3.48}{4} = 3.79$$

$$C_{2_{Organisasi}} = \frac{1+1+1+1}{4} = 1$$

$$C_{2_{Kerja}} = \frac{0+0+1+0}{4} = 0.25$$

c. Pusat Cluster 3

**Tabel 3.11 Tabel Pusat Cluster Tiga Baru**

| Cluster | IP   | Organisasi | Kerja |
|---------|------|------------|-------|
| D6      | 3.92 | 0          | 0     |
| D9      | 3.68 | 0          | 1     |

Pusat Baru IP :

$$C_{3IP} = \frac{3.92+3.68}{2} = 3.8$$

$$C_{2Organisasi} = \frac{0+0}{2} = 0$$

$$C_{3Kerja} = \frac{0+1}{2} = 0.5$$

**Tabel 3.12 Perbandingan Pusat Cluster Baru dan Lama**

| <b>Pusat Cluster Lama</b> |        |            |             |
|---------------------------|--------|------------|-------------|
| Cluster                   | IP     | Organisasi | Kerja       |
| 1                         | 3.48   | 0          | 1           |
| 2                         | 3.77   | 1          | 1           |
| 3                         | 3.68   | 0          | 1           |
| <b>Pusat Cluster Baru</b> |        |            |             |
| Cluster                   | IP     | Organisasi | Kerja       |
| 1                         | 3.02   | 0          | 0.333333333 |
| 2                         | 3.7975 | 1          | 0.25        |
| 3                         | 3.8    | 0          | 0.5         |

Jika dilihat pada Tabel 3.12, antara pusat *cluster* lama dan pusat *cluster* baru mengalami perubahan nilai. Sehingga proses iterasi akan dilanjutkan hingga tidak ada perubahan pusat *cluster* atau jumlah iterasi maksimal dari model sudah terpenuhi. Setelah dilakukan beberapa iterasi, hasil akhir dari proses *clustering* dapat dilihat di Tabel 3.13 berikut ini.

**Tabel 3.13 Tabel Hasil Akhir Proses Clustering**

| Data | IP   | Organisasi | Kerja | Hasil |
|------|------|------------|-------|-------|
| D1   | 3.94 | 1          | 0     | C2    |
| D2   | 3.48 | 0          | 1     | C3    |
| D3   | 2.83 | 0          | 0     | C1    |
| D4   | 4    | 1          | 0     | C2    |
| D5   | 3.77 | 1          | 1     | C2    |
| D6   | 3.92 | 0          | 0     | C3    |
| D7   | 3.48 | 1          | 0     | C2    |
| D8   | 2.75 | 0          | 0     | C1    |
| D9   | 3.68 | 0          | 1     | C3    |

2) *K Parameter Optimization*

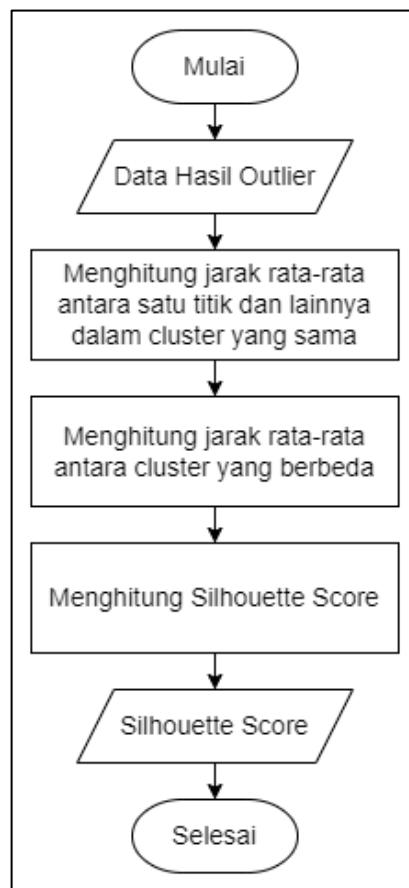
Optimasi model dilakukan dengan menggabungkan *Elbow* dan *Silhouette Method*. Kedua metode tersebut merupakan metode dalam *hyperparameter tuning* yang dapat membantu dalam pencarian model terbaik dengan melakukan optimasi pada parameter k. *Elbow Method* memiliki nilai akhir dari jumlah kuadrat dari Euclidean Distances (SSE) pada setiap kenaikan

nilai parameter k, sedangkan *Silhouette* menggunakan jarak rata-rata antara satu titik data dan lainnya dalam cluster yang sama (*cohesion measure*) dan jarak rata-rata antara cluster yang berbeda (*separation measure*). Optimasi pada model ini sangat diperlukan karena *K-Means* dipengaruhi oleh nilai parameter k sebagai penentu jumlah *cluster* yang akan dibuat.

Pada penelitian ini, *Silhouette Method* akan digunakan untuk mendukung hasil dari *Elbow Method*. Skenario optimasi parameter yang akan dilakukan adalah dengan melakukan perhitungan *Elbow* dan *Silhouette* untuk nilai parameter k mulai dari  $k = 2$  hingga  $k = 8$ . Setelah itu akan dihasilkan irisan nilai dari kedua metode tersebut yang akan digunakan sebagai nilai parameter K dalam proses *K-Means clustering*.

### 3.1.7 Model Evaluation

Tahapan evaluasi pada penelitian ini dilakukan dengan menguji model yang telah dibangun sebelumnya. Pengujian dilakukan dengan mencari nilai *silhouette score* dan melihat analisis dari penggunaan nilai K terbaik pada model yang telah dibangun. Proses perhitungan *silhouette score* pada tahapan ini secara ringkas dapat dilihat pada *flowchart* Gambar 3.8 berikut.



Gambar 3.8 Flowchart Model Evaluation

Proses pertama yaitu dengan melakukan proses perhitungan jarak rata-rata antar titik dalam *cluster* yang sama pada data sampel yang tertera di Tabel 3.3. Proses perhitungan dilakukan dengan menggunakan dengan Persamaan 2.6 berikut ini.

$$a(3) = \frac{\sqrt{(c_{8x} - c_{3x})^2 + (c_{8y} - c_{3y})^2 + (c_{8z} - c_{3z})^2}}{c_i}$$

$$a(3) = (\sqrt{(2.75 - 2.83)^2 + (0 - 0)^2 + (0 - 0)^2})/1 = 0.08$$

Setelah didapatkan jarak antar titik di dalam cluster yang sama, langkah selanjutnya dilanjutkan dengan menghitung jarak minimum antar titik dalam cluster yang berbeda dengan Persamaan 2.7 seperti berikut.

$$d(3,1) = -$$

$$\begin{aligned} d(3,2) &= (\sqrt{(3.94 - 2.83)^2 + (1 - 0)^2 + (0 - 0)^2} + \\ &\quad \sqrt{(4 - 2.83)^2 + (1 - 0)^2 + (0 - 0)^2} + \\ &\quad \sqrt{(3.77 - 2.83)^2 + (1 - 0)^2 + (1 - 0)^2} + \\ &\quad \sqrt{(3.48 - 2.83)^2 + (1 - 0)^2 + (0 - 0)^2})/4 = 1.480986798 \end{aligned}$$

$$\begin{aligned} d(3,3) &= (\sqrt{(3.48 - 2.83)^2 + (0 - 0)^2 + (1 - 0)^2} + \\ &\quad \sqrt{(3.92 - 2.83)^2 + (0 - 0)^2 + (0 - 0)^2} + \\ &\quad \sqrt{(3.68 - 2.83)^2 + (0 - 0)^2 + (1 - 0)^2})/4 = 1.198375506 \end{aligned}$$

$$b(3) = \min(d(3,1), d(3,2), d(3,3)) = 1.198375506$$

Ketika sudah didapatkan nilai antar *cluster* dalam *cluster* yang sama, selanjutnya menghitung nilai *silhouette score* dengan Persamaan 2.5 sebagai berikut.

$$S(i) = \frac{b-a}{\max(a,b)}$$

$$S(i) = \frac{1.198375506 - 0.08}{\max(1.198375506, 0.08)} = \frac{1.118375506}{1.198375506} = 0.933242961$$

Setelah itu, proses yang sama dapat diterapkan ke semua titik yang tersedia. Untuk mendapatkan nilai akhir untuk *silhouette score* dapat dilakukan dengan melakukan rata-rata pada setiap data. Semua hasil perhitungan *silhouette score* dapat dilihat di Tabel 3.14 berikut ini.

**Tabel 3.14 Tabel Silhouette Score**

| Data                                | X    | Y | Z | a(i) | d(i,1) | d(i,2) | d(i,3) | b(i)              | S(i)        |
|-------------------------------------|------|---|---|------|--------|--------|--------|-------------------|-------------|
| D1                                  | 3.94 | 1 | 0 | 0.08 | -      | 1.48   | 1.19   | 1.19              | 0.933242961 |
| D2                                  | 3.48 | 0 | 1 | 0.08 | -      | 1.53   | 1.25   | 1.25              | 0.936402223 |
| D3                                  | 2.83 | 0 | 0 | 0.51 | 1.52   | -      | 1.30   | 1.30              | 0.609109445 |
| D4                                  | 4    | 1 | 0 | 0.53 | 1.56   | -      | 1.31   | 1.31              | 0.594411254 |
| D5                                  | 3.77 | 1 | 1 | 1.02 | 1.72   | -      | 1.15   | 1.15              | 0.111245616 |
| D6                                  | 3.92 | 0 | 0 | 0.67 | 1.21   | -      | 1.31   | 1.21              | 0.445666717 |
| D7                                  | 3.48 | 1 | 0 | 0.64 | 3.74   | 1.36   | -      | 1.36              | 0.525623658 |
| D8                                  | 2.75 | 0 | 0 | 1.06 | 3.67   | 1.12   | -      | 1.12              | 0.061138503 |
| D9                                  | 3.68 | 0 | 1 | 0.61 | 3.80   | 1.33   | -      | 1.33              | 0.538214759 |
| <b>Nilai silhouette score K = 3</b> |      |   |   |      |        |        |        | <b>0.52833946</b> |             |

Dapat dilihat pada Tabel 3.14, nilai *silhouette score* untuk parameter  $k = 3$  adalah 0.52833946 yang didapatkan melalui rata-rata *silhouette score* setiap data. Jika merujuk pada Tabel 2.1, untuk nilai tersebut dapat disimpulkan bahwa struktur *cluster* yang dihasilkan adalah struktur sedang dan merupakan struktur yang cukup baik karena memiliki nilai *silhouette score* diantara 0.5 dan 0,7.

## 3.2 Metodologi Pengembangan Sistem

### 3.2.1 Analisis Kebutuhan Sistem

Pada tahap ini akan dilaksanakan analisis terhadap hal-hal yang berkaitan dengan kebutuhan untuk mengembangkan sistem pada penelitian ini. Analisis kebutuhan sistem ini terdapat dua bagian yaitu kebutuhan fungsional dan kebutuhan perangkat.

#### 1) Kebutuhan Fungsional

Kebutuhan fungsional yaitu kebutuhan yang berkaitan dengan sistem yang dibuat. Sehingga mencakup gambaran proses-proses yang terdapat dalam sistemnya. Kebutuhan fungsional dalam sistem ini sebagai berikut :

- a. Sistem dapat menyajikan *dataset*
- b. Sistem dapat menyajikan data hasil *preprocessing*
- c. Sistem dapat menyajikan grafik *Elbow method*
- d. Sistem dapat menyajikan data hasil clustering dari *K-Means* dalam bentuk 3 dimensi

#### 2) Kebutuhan Non-Fungsional

Kebutuhan non-fungsional ini yaitu kebutuhan yang tidak berkaitan langsung dengan sistem, pada hal ini yaitu kebutuhan perangkat. Perangkat sendiri terdapat dua yaitu perangkat keras dan perangkat lunak. Kebutuhan perangkat mencakup mengenai spesifikasi kebutuhan perangkat yang digunakan untuk membangun sistem. Kebutuhan perangkat sendiri meliputi perangkat keras dan perangkat lunak. Berikut merupakan spesifikasi kebutuhan perangkat lunak dan perangkat keras yang digunakan dalam penelitian ini :

**Tabel 3.15 Spesifikasi Kebutuhan Perangkat Keras**

| Perangkat Keras | Spesifikasi                   |
|-----------------|-------------------------------|
| Tipe Sistem     | Tipe Sistem 64-bit            |
| Prosesor        | Intel Core i7-11370H, 3.30GHz |
| Memori/RAM      | 16.00 GB                      |
| SSD             | 1 TB                          |

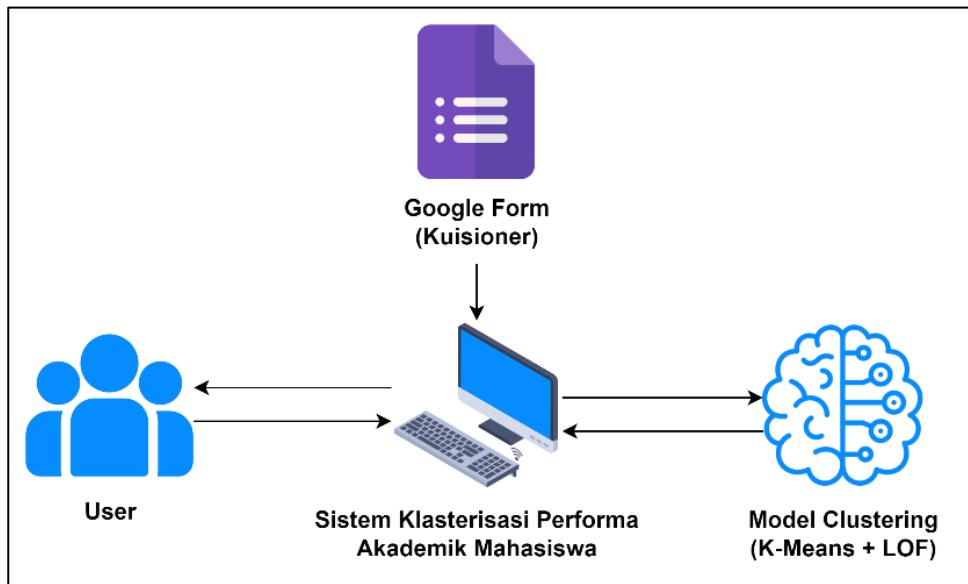
**Tabel 3.16 Spesifikasi Kebutuhan Perangkat Lunak**

| Perangkat Lunak    | Spesifikasi                          |
|--------------------|--------------------------------------|
| Sistem Operasi     | Windows 11                           |
| Bahasa Pemrograman | Python 3.11                          |
| Web Browser        | Opera GX                             |
| GUI                | Python Dash                          |
| IDE                | Jupyter Notebook, Visual Studio Code |

### 3.2.2 Perancangan Sistem

Pada bagian perancangan sistem akan mengupas mengenai rancangan sistem yang akan dibuat dalam penelitian. Perancangan sistem pada penelitian ini mencakup perancangan arsitektur sistem, perancangan proses, perancangan antarmuka.

#### 1) Perancangan Arsitektur

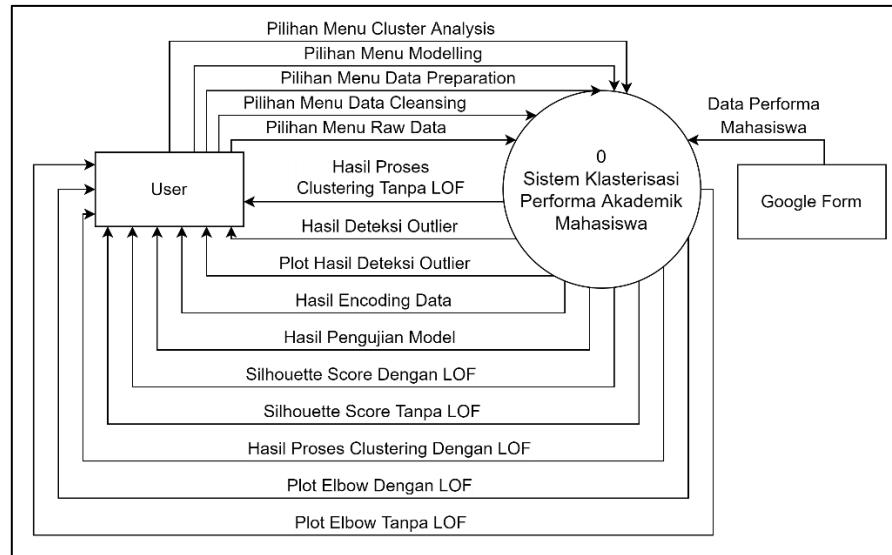


Gambar 3.9 Perancangan Arsitektur Sistem

Arsitektur sistem yang dibuat pada penelitian ini meliputi *user* dari aplikasi, sistem klasterisasi performa akademik mahasiswa, model clustering, dan dataset. *User* pada arsitektur ini merupakan orang yang akan mengoperasikan sistem ini. Sistem ini dimulai dari pengambilan data dengan melakukan penyebaran kuisioner. Langkah selanjutnya adalah data akan masuk ke proses *clustering* dimana pada proses ini akan dilakukan proses analisis outlier dengan menggunakan *Local Outlier Factor* dan *clustering data* menggunakan model *K-Means* yang sudah dilatih untuk mendapatkan hasil pengelompokan pada data yang dapat dilihat oleh *user*.

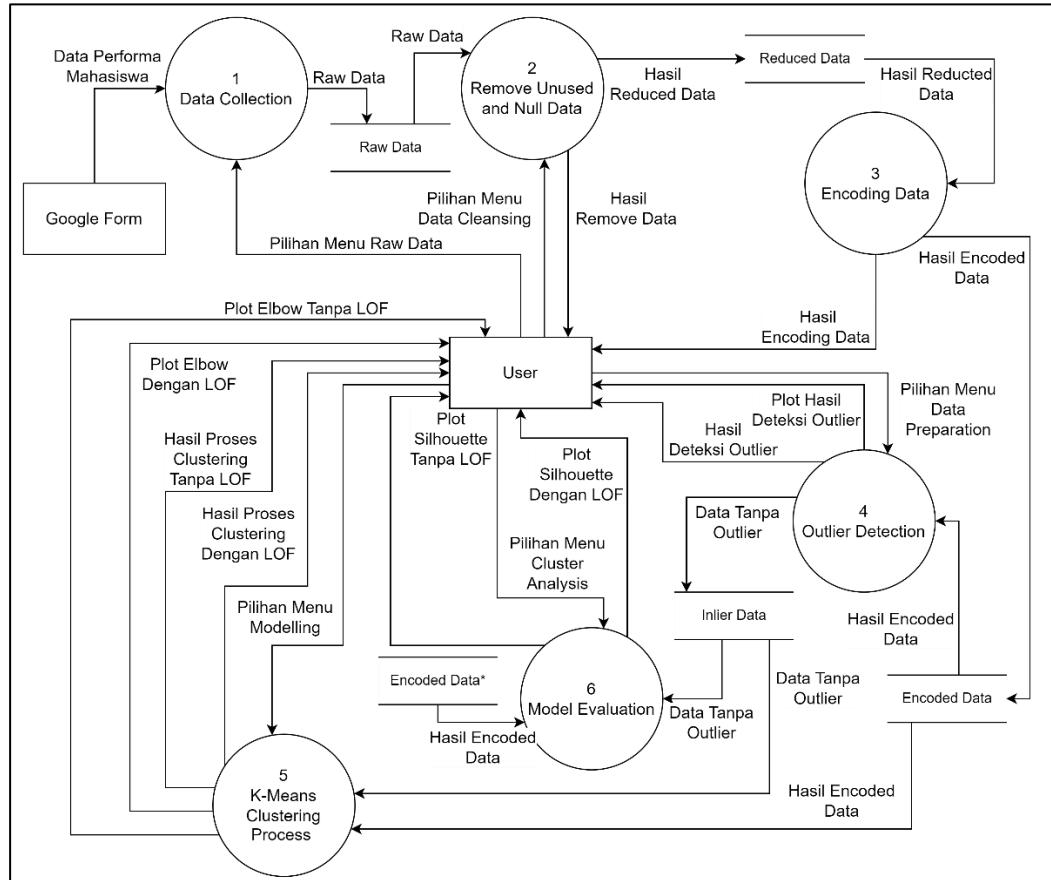
#### 2) Perancangan Proses

Perancangan proses sistem pada penelitian ini dilakukan dengan membuat aliran data pada sistem yang akan dibangun. Perancangan proses yang dibuat digambarkan dengan menggunakan *Data Flow Diagram* (DFD) level 0, level 1, dan level 2.



**Gambar 3.10 Proses DFD Level 0**

Gambar 3.10 merupakan hasil perancangan DFD level 0 dimana terdapat dua entitas dan 1 proses yang saling berinteraksi. Pada proses ini sistem akan menerima *input* berupa data akademik dari mahasiswa yang akan digunakan dalam proses *clustering*. *Output* yang akan diterima oleh *user* adalah *user* dapat melihat hasil dari pengolahan data, hasil dari deteksi *outlier*, hasil dari proses *clustering*, dan hasil dari pengujian model.

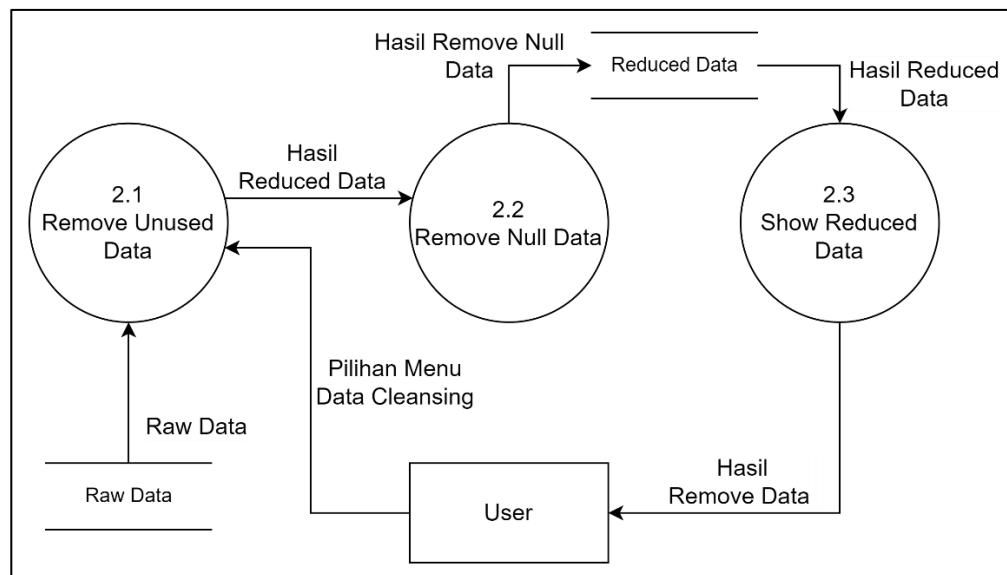


**Gambar 3.11 Proses DFD Level 1**

Setelah perancangan DFD level 0, maka akan dilanjutkan pada perancangan DFD level 1, Gambar 3.11, dimana pada proses ini alur data akan dibuat menjadi lebih detail. Alur data pada DFD level 1 terdapat enam cabang yaitu data collection, remove unused and null data, encoding data, outlier detection, dan proses clustering dengan K-Means, dan model evaluation.

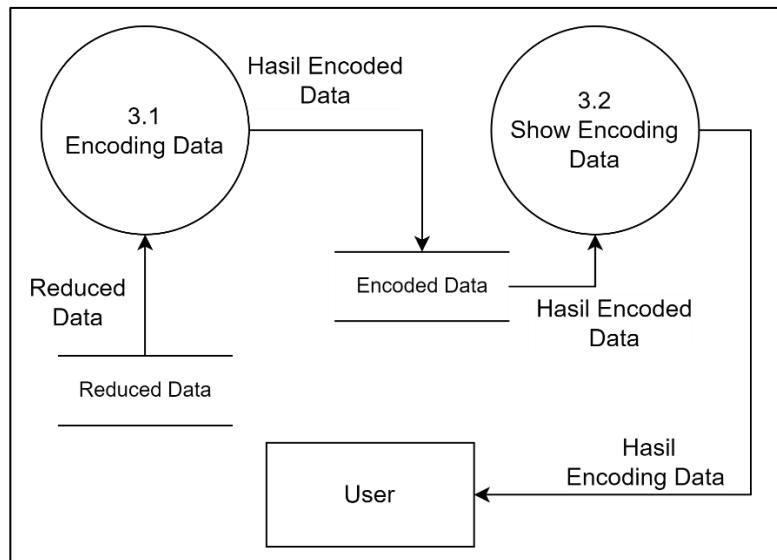
Pada proses pertama adalah data collection dimana pada proses ini akan mengambil data dari platform Google Form. Data yang diambil merupakan data yang sebelumnya telah dilakukan penyebaran kuisioner pada mahasiswa Informatika dan Sistem Informasi UPN Veteran Yogyakarta. Pada proses kedua, removed unused and null data, akan menerima input data akademik mahasiswa. Tahapan ini dilakukan untuk menghapus data-data yang kosong dan juga menghapus kolom pada data yang tidak digunakan. Ketika proses pertama selesai akan langsung dilanjutkan ke proses kedua yaitu encoding data. Tahapan ini akan mengubah data yang bersifat kategorikal dengan value “Ya” menjadi angka 1 dan data dengan value “Tidak” diubah menjadi angka 0.

Pada proses selanjutnya akan dilakukan proses pendekripsi outlier dimana pada proses ini akan menghasilkan data yang bebas dari outlier. Data-data yang sudah dibersihkan dari outlier akan digunakan pada dua proses yaitu proses clustering dan juga proses evaluasi model. Pada proses clustering akan dilakukan pengelompokan data dengan menggunakan algoritma K-Means yang sudah dilatih sebelumnya. Sedangkan pada proses evaluasi model akan dilakukan penghitungan silhouette score untuk mengetahui hasil dari model dengan menaikkan nilai parameter k. Dari keenam proses yang terdapat pada DFD level 1 akan diilustrasikan lebih lanjut pada DFD level 2.



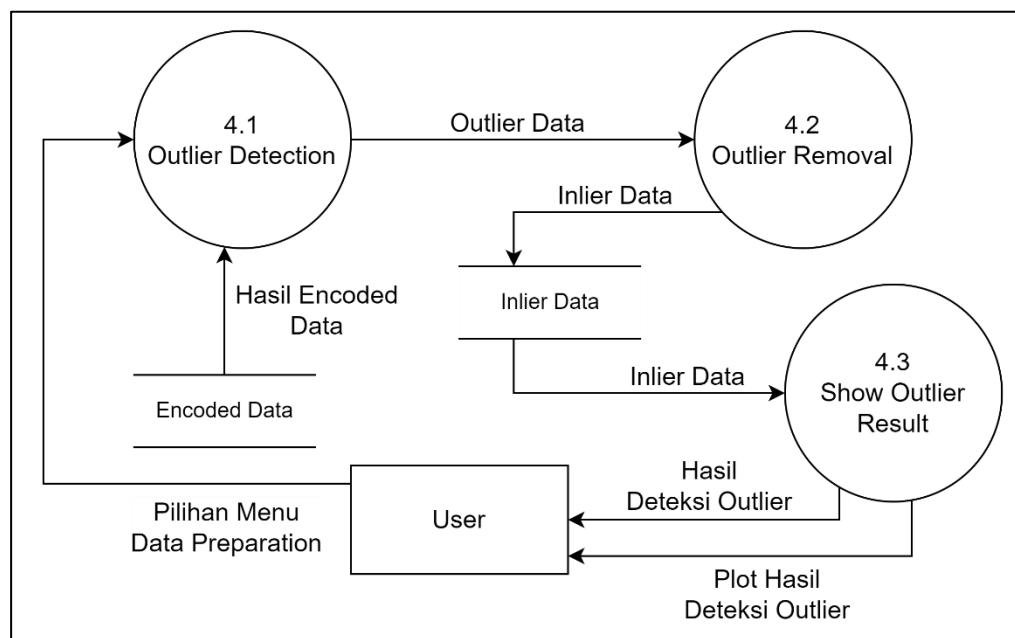
**Gambar 3.12 Proses DFD Level 2 – Remove Unused and Null Data**

DFD level 2 pada Gambar 3.12 terlihat bahwa proses sebelumnya dapat dipecah menjadi tiga buah proses yang lebih kecil lagi. Proses dimulai dari penghapusan data yang tidak digunakan dari data akademik mahasiswa dan kemudian dilanjutkan dengan menghapus data kosong. Data yang telah dilakukan proses dapat dilihat langsung oleh *user*.



**Gambar 3.13 Proses DFD Level 2 – Encoding Data**

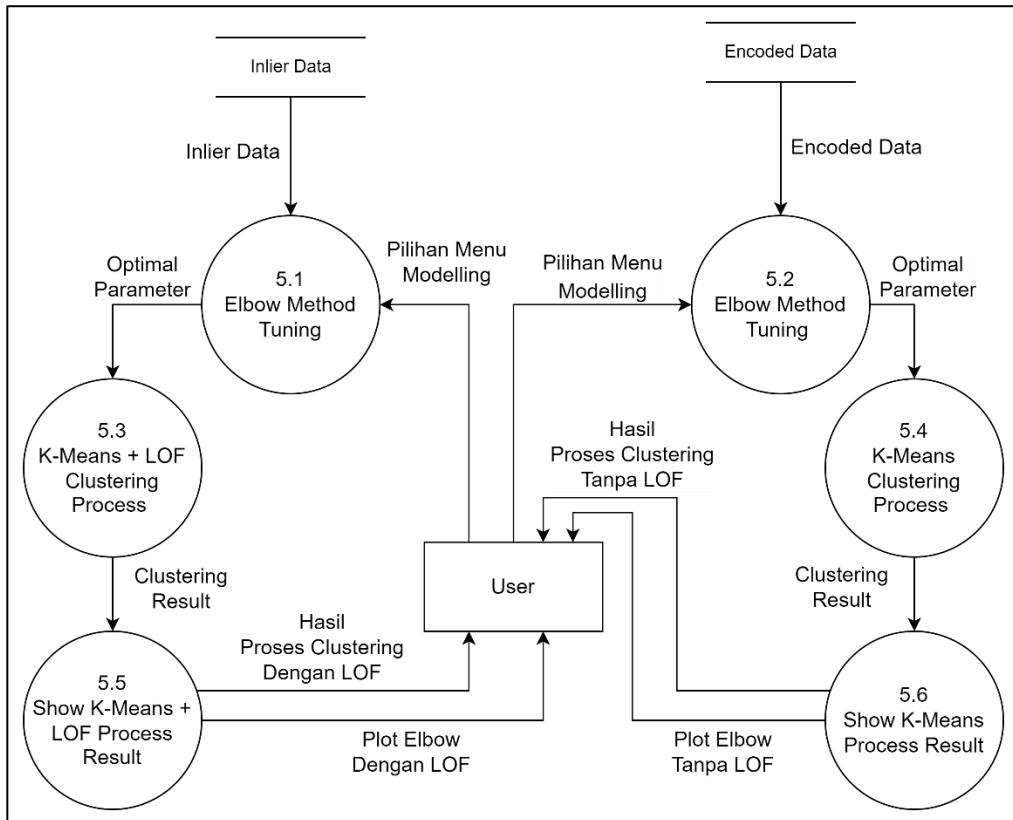
DFD level 2 pada Gambar 3.13 terdapat dua buah proses yang dimulai dari pengubahan data yang bersifat kategorikal dengan *value* “Ya” menjadi angka 1 dan data dengan *value* “Tidak” diubah menjadi angka 0. Data yang telah melalui proses *encoding* dapat dilihat oleh *user*.



**Gambar 3.14 Proses DFD Level 2 – Outlier Detection**

DFD level 2 pada Gambar 3.14 merupakan tahap pendekripsi *outlier* dimana data yang sebelumnya sudah diproses akan langsung didekripsi dengan

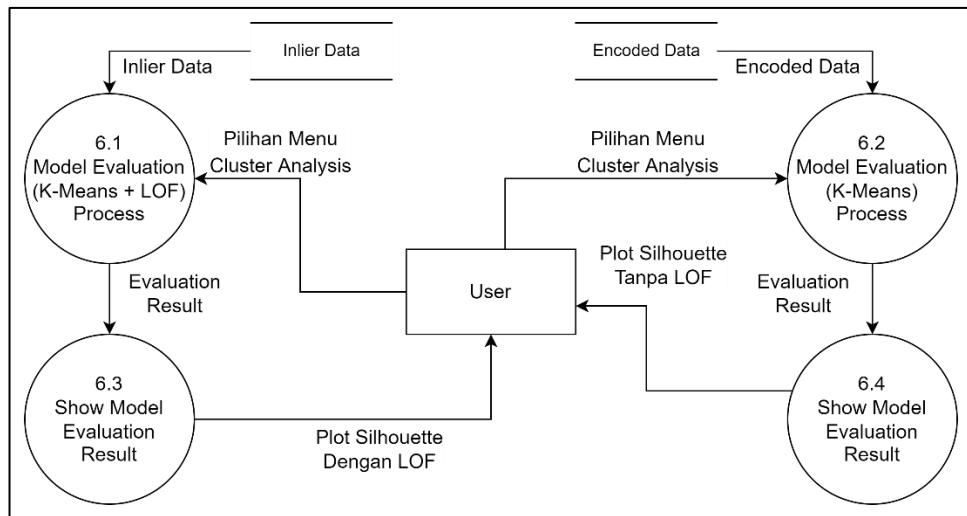
menggunakan algoritma *Local Outlier Factor*. Setelah didapatkan data-data yang terdeteksi sebagai *outlier*, data tersebut akan dihapus dan hasil dari proses ini akan dapat dilihat oleh *user*.



**Gambar 3.15 Proses DFD Level 2 – *Clustering Process***

DFD level 2 pada Gambar 3.15 terbagi menjadi dua buah proses besar, yaitu proses clustering dengan menggunakan K-Means + Local Outlier Factor dan K-Means saja. Sebelum proses pengelompokan dimulai akan dilakukan proses optimasi yaitu pengambilan parameter k terbaik yang akan digunakan sebagai jumlah cluster pada saat pengelompokan telah selesai dilakukan. Perbedaan kedua proses tersebut terdapat pada data yang digunakan, untuk proses pengelompokan dengan K-Means + LOF menggunakan data hasil deteksi outlier sedangkan pada proses pengelompokan dengan K-Means menggunakan data sebelum pendekripsi outlier. Ketika proses clustering telah selesai, hasil data serta plot Elbow akan ditampilkan dan dapat dilihat oleh user.

DFD level 2 pada Gambar 3.16 merupakan tahap terakhir dari proses pada sistem ini. Sama seperti proses sebelumnya, pada tahapan ini akan dilakukan dua buah proses, yaitu evaluasi pada model clustering yang menggunakan K-Means + Local Outlier Factor dan K-Means saja. Evaluasi yang digunakan adalah evaluasi Silhouette Method yang menguji model dengan menaikkan nilai parameter k. Hasil dari metode yang digunakan ini adalah silhouette score yang dapat ditampilkan dalam bentuk plot dan dapat dilihat langsung oleh user.



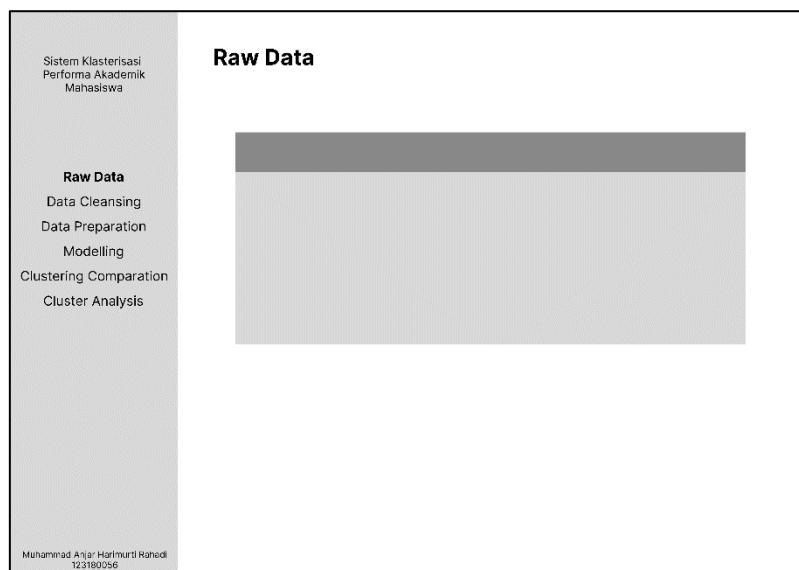
**Gambar 3.16 Proses DFD Level 2 – Model Evaluation**

### 3) Perancangan Antarmuka

Perancangan antarmuka akan menggambarkan bagaimana sistem yang akan dibangun. Tahapan ini akan menjadi acuan dalam pembuatan UI saat proses *development* dimulai. Beberapa rincian dari rancangan antarmuka adalah sebagai berikut.

#### a. Rancangan Halaman Raw Data

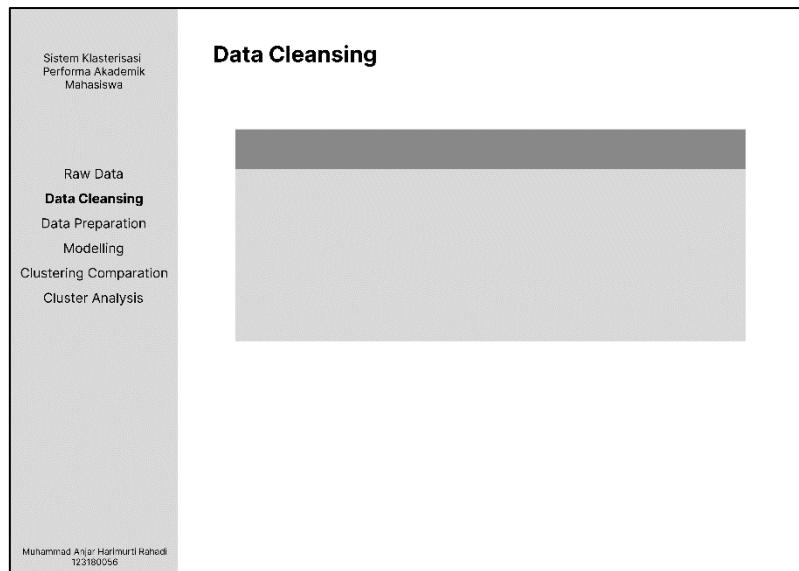
Pada halaman *raw data* akan ditampilkan data asli yang belum dilakukan perubahan atau proses. Data ini langsung didapatkan dari *dataset* yang telah diperoleh sebelumnya.



**Gambar 3.17 Rancangan Halaman Raw Data**

b. Rancangan Halaman *Data Cleansing*

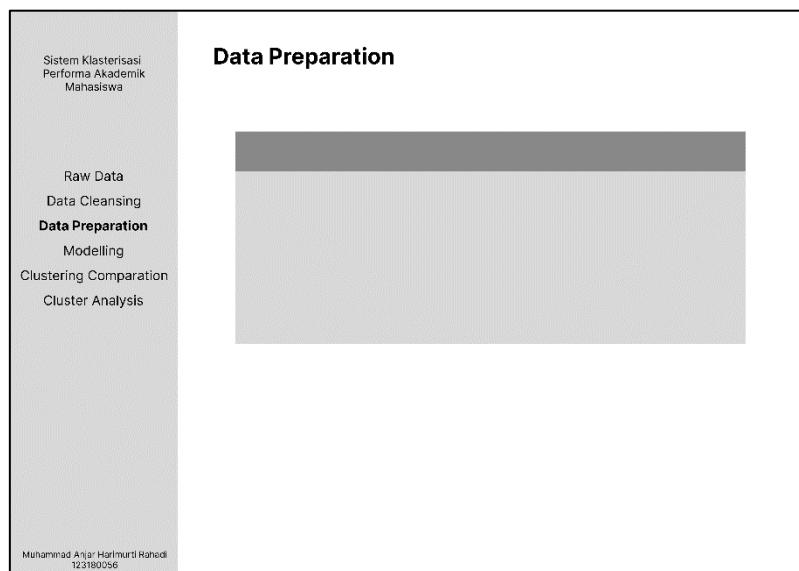
Pada halaman *data cleansing* akan ditampilkan proses pembersihan data seperti penghilangan kolom-kolom pada data yang tidak digunakan, menghilangkan data kosong, dan melakuakan proses *encoding* pada data.



Gambar 3.18 Rancangan Halaman *Data Cleansing*

c. Rancangan Halaman *Data Preparation*

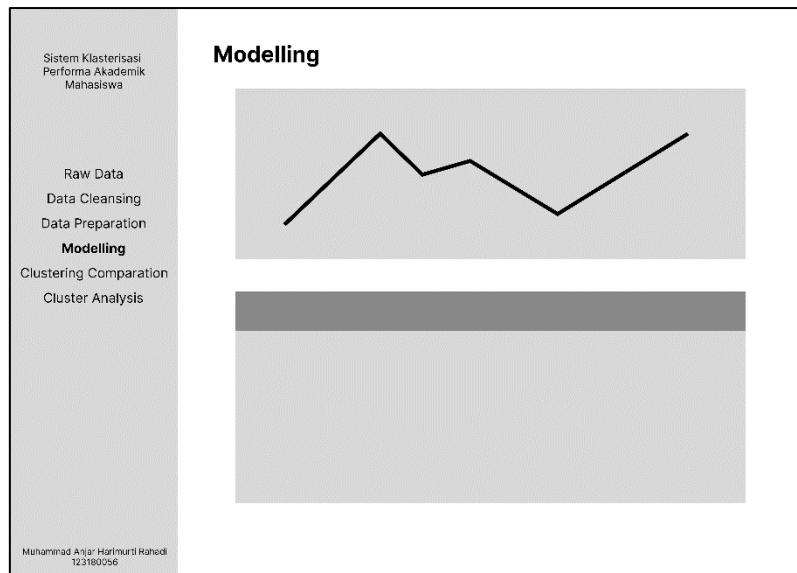
Pada halaman *data preparation* akan ditampilkan proses persiapan data sebelum masuk ke dalam model seperti pendekesan dan penghapusan *outlier* pada data.



Gambar 3.19 Rancangan Halaman *Data Preparation*

d. Rancangan Halaman *Modelling*

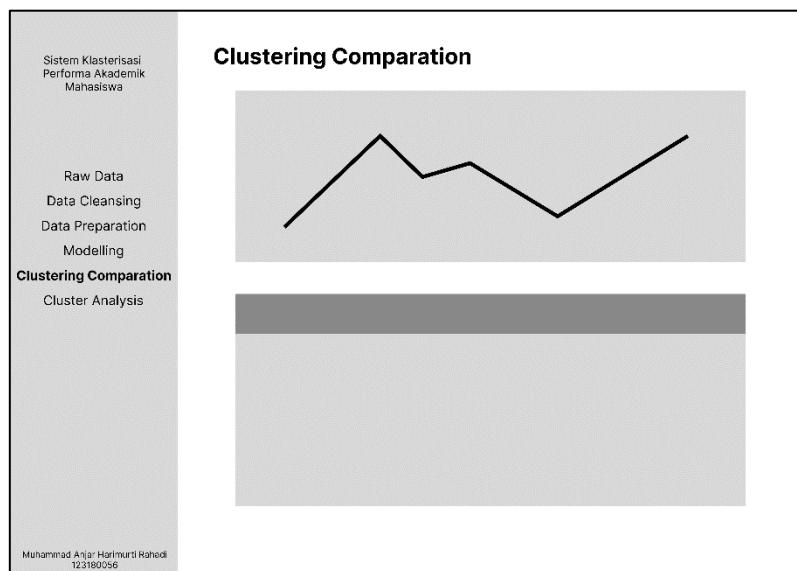
Pada halaman *modelling* akan ditampilkan proses pemilihan nilai parameter k terbaik dengan menggunakan *hyperparameter tuning* dan menampilkan hasil proses *clustering* pada data.



Gambar 3.20 Rancangan Halaman *Modelling*

e. Rancangan Halaman *Clustering Comparation*

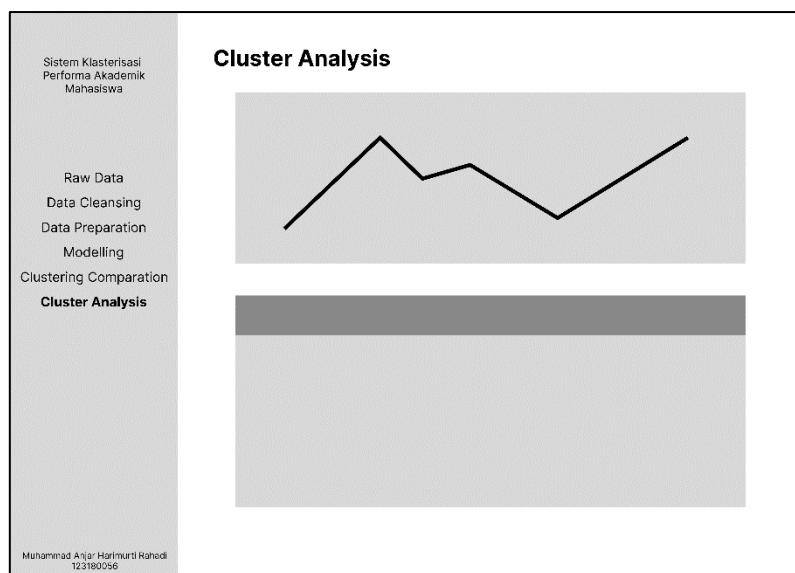
Pada halaman *clustering comparation* akan ditampilkan hasil perbandingan antara algoritma K-Means dan algoritma K-Means yang telah dioptimasi dengan *Local Outlier Factor*.



Gambar 3.21 Rancangan Halaman *Clustering Comparation*

#### f. Rancangan Halaman *Cluster Analysis*

Pada halaman *clustering analysis* akan ditampilkan hasil dari penelitian seperti interpretasi dari masing masing *cluster* yang terbentuk dan juga akan ditampilkan hasil evaluasi model dengan *Silhouette Score*.



**Gambar 3.22 Rancangan Halaman *Cluster Analysis***

#### 3.2.3 Pengujian Sistem

Pada penelitian ini, tahapan pengujian ini dilakukan agar aplikasi yang dibangun berfungsi secara penuh tanpa adanya error saat dijalankan. Dalam penelitian ini, pengujian dilakukan dengan mengukur kinerja dan fungsionalitas dari aplikasi yang dibuat. Berikut merupakan skenario dalam pengujian fungsionalitas aplikasi yang dibangun seperti Tabel 3.18 berikut,

**Tabel 3.17 Rancangan Pengujian Aplikasi**

| No | Halaman                 | Pengujian   | Hasil    |       |
|----|-------------------------|---|----------|-------|
|    |                         |   | Berhasil | Gagal |
| 1. | <i>Data Raw</i>         | Menampilkan 10 data asli  |          |       |
| 2. | <i>Data Cleansing</i>   | Menampilkan 10 data setelah dilakukan penghapusan kolom yang tidak terpakai |          |       |
| 3. |                         | Menampilkan 10 data setelah dilakukan proses <i>encoding data</i>           |          |       |
| 4. | <i>Data Preparation</i> | Menampilkan plot data yang terdeteksi sebagai <i>outlier</i>                |          |       |
| 5. |                         | Menampilkan 10 data setelah dilakukan pendekripsi <i>outlier</i>            |          |       |

**Tabel 3.18 Rancangan Pengujian Aplikasi Lanjutan**

| No  | Halaman                       | Pengujian   | Hasil    |       |
|-----|-------------------------------|---|----------|-------|
|     |                               |   | Berhasil | Gagal |
| 6.  | <i>Modelling</i>              | Menampilkan plot dari <i>Elbow Method</i>   |          |       |
| 7.  |                               | Menampilkan tabel data setelah proses <i>clustering</i>                             |          |       |
| 8.  |                               | Menampilkan plot dari <i>Silhouette Score</i>                                       |          |       |
| 9.  | <i>Clustering Comparation</i> | Menampilkan plot dari <i>Elbow Method</i> sebelum dan setelah optimasi              |          |       |
| 10. |                               | Menampilkan plot dari <i>Silhouette Score</i> sebelum dan setelah optimasi          |          |       |
| 11. |                               | Menampilkan plot data setelah proses <i>clustering</i> sebelum dan setelah optimasi |          |       |
| 12. | <i>Cluster Analysis</i>       | Menampilkan hasil interpretasi <i>cluster</i>                                       |          |       |
| 13. |                               | Menampilkan plot <i>Silhouette Visualization</i> sesuai dengan nilai K Terbaik      |          |       |

## BAB IV

# HASIL DAN PEMBAHASAN

### 4.1 Hasil Penelitian

Pada bagian sub bab implementasi ini akan dijelaskan secara lengkap mengenai proses rancangan pada bab sebelumnya yang kemudian akan menjadi sebuah aplikasi. Proses perancangan tersebut akan diimplementasikan dalam bentuk kode program. Berikut merupakan tampilan dari aplikasi klasterisasi performa akademik mahasiswa yang dapat dilihat pada Gambar 4.1.

| NIM            | GolonganUKT | IPGanjil | OrganisasiGanjil | KerjaGanjil | IPGenap | OrganisasiGenap | KerjaGenap |
|----------------|-------------|----------|------------------|-------------|---------|-----------------|------------|
| filter data... |             |          |                  |             |         |                 |            |
| 123180001      | 4           | 3.88     | Tidak            | Tidak       | 3.37    | Tidak           | Tidak      |
| 123170041      | 4           | 3.83     | Ya               | Tidak       | 3.8     | Ya              | Tidak      |
| 123210103      | 5           | 3.9      | Tidak            | Tidak       | 3.7     | Ya              | Tidak      |
| 123180004      | 2           | 3.89     | Tidak            | Tidak       | 3.57    | Tidak           | Tidak      |
| 123180007      | 8           | 4        | Ya               | Ya          | 4       | Ya              | Ya         |
| 123180008      | 3           | 3.8      | Ya               | Tidak       | 3.08    | Tidak           | Tidak      |
| 123200065      | 0           | 3.9      | Tidak            | Tidak       | 3.6     | Tidak           | Tidak      |
| 123200056      | 7           | 3.94     | Tidak            | Tidak       | 3.62    | Ya              | Tidak      |
| 123200023      | 5           | 3.88     | Ya               | Ya          | 3.77    | Ya              | Ya         |
| 123180009      | 4           | 3.62     | Tidak            | Tidak       | 3.7     | Tidak           | Tidak      |

Gambar 4.1 Tampilan Aplikasi Klasterisasi Performa Akademik Mahasiswa

Pada Gambar 4.1 dapat dilihat bahwa aplikasi yang dibuat terdapat enam menu yang masing-masing mewakili proses clustering dari performa akademik mahasiswa. Pada menu “Raw Data” akan menampilkan data asli yang didapatkan secara langsung melalui kuisioner. Menu kedua adalah “Data Cleansing” yang berisi proses pembersihan data seperti penghilangan kolom-kolom pada data yang tidak digunakan. Selain itu, pada menu ini juga akan menampilkan proses pengubahan data yang sebelumnya berupa data kategorikal seperti “Ya” dan “Tidak” menjadi data numerik seperti 0 dan 1. Pada menu selanjutnya adalah “Data Preparation” yang akan menampilkan data-data mana saja yang tegolong sebagai data outlier dan inlier.

Jika pada ketiga menu sebelumnya masih berhubungan tentang pengubahan data yang akan digunakan, menu keempat yaitu “Modelling” akan menampilkan proses optimasi pada nilai k dan hasil dari model clustering dari data sebelumnya. Menu selanjutnya adalah “Cluster Analysis”, pada menu ini akan menampilkan analisis dari hasil proses clustering seperti interpretasi cluster yang terbentuk dan hasil evaluasi dari model.

Menu terakhir dari aplikasi yang dibangun adalah “Clustering Comparation” yang akan menampilkan hasil perbandingan antara algoritma K-Means dan algoritma K-Means yang telah dioptimasi dengan Local Outlier Factor.

Pada implementasinya, kode program yang dibuat menggunakan beberapa library dalam pembuatan model dan aplikasi yang dibangun. Berikut merupakan library yang digunakan dalam pembuatan aplikasi.

#### 4.1.1 *Data Initialization*

Tahapan pertama ini memiliki beberapa proses di dalamnya yaitu proses pengambilan dan *import* data. Pada proses pertama pengambilan data dilakukan dengan cara penyebaran kuisioner melalui platform *Google Form* yang terdiri dari beberapa pertanyaan yang akan digunakan sebagai parameter dalam penelitian ini. Setelah data dari kuisioner didapatkan, perlu dilakukan proses *import* pada data agar dapat digunakan dalam pembuatan model. Proses *import* atau pengambilan data kedalam aplikasi dapat dilakukan sebagai berikut.

---

##### ***Pseudocode 1: Proses Import Data***

---

```
SET data = pd.read_csv('dataset/datasetFull.csv')
```

---

#### **Modul Program 4.1 Proses Import Data**

#### 4.1.2 *Preprocessing Data*

Tahapan selanjutnya dari aplikasi ini adalah *preprocessing data*. Pada tahapan ini akan terbagi menjadi dua buah proses yaitu *removing unused and null data* serta *encoding data*.

##### 1) *Removing Unused and Null Data*

Tahapan awal dari *preprocessing data* adalah penghapusan kolom-kolom yang tidak digunakan dan data kosong. Tahapan ini diperlukan karena tidak semua data yang diambil dari kuisioner akan digunakan untuk pelatihan model seperti email, nim dan nama. Selain kolom, data kosong juga dapat mengganggu proses pelatihan model sehingga hasilnya menjadi kurang baik. Oleh karena itu, pada tahapan awal penelitian ini akan menghapus kolom-kolom yang tidak digunakan dan juga data kosong. Proses penghapusan kedua hal tersebut dapat dilakukan sebagai berikut.

---

##### ***Pseudocode 2: Proses Removing Unused and Null Data***

---

```
SET dataFilter=data[['IPGenap','OrganisasiGenap','KerjaGenap']]  
SET dataFilter = dataFilter.dropna()
```

---

#### **Modul Program 4.2 Proses *Removing Unused and Null Data***

Pada modul program diatas data yang akan digunakan adalah IPGenap, OrganisasiGenap, dan KerjaGenap. Setelah dilakukan pemilihan kolom data yang akan digunakan, akan dilakukan proses penghapusan data yang bernilai kosong. Berikut merupakan hasil dari proses *removing unused and null data* dapat dilihat pada Gambar 4.2.

|     | IPGenap | OrganisasiGenap | KerjaGenap |
|-----|---------|-----------------|------------|
| 0   | 3.96    | Tidak           | Tidak      |
| 1   | 3.80    | Ya              | Tidak      |
| 2   | 3.70    | Ya              | Tidak      |
| 3   | 2.83    | Tidak           | Tidak      |
| 4   | 3.70    | Tidak           | Tidak      |
| ... | ...     | ...             | ...        |
| 205 | 3.48    | Tidak           | Ya         |
| 206 | 3.60    | Tidak           | Tidak      |
| 207 | 3.11    | Ya              | Tidak      |
| 208 | 4.00    | Ya              | Tidak      |
| 209 | 3.55    | Ya              | Tidak      |

**Gambar 4.2 Proses *Removing Unused and Null Data***

## 2) Encoding Data

Tahapan selanjutnya adalah mengubah data yang memiliki tipe kategorikal menjadi tipe numerik. Hal ini perlu untuk dilakukan karena dalam model *K-Means* yang akan dibangun hanya menerima data-data dengan tipe numerik. Oleh karena itu, tahapan ini juga penting untuk dilakukan agar algoritma *K-Means* dapat bekerja dengan baik. Proses pengubahan tipe data atau *encoding data* dapat dilakukan sebagai berikut.

---

### **Pseudocode 3: Proses Encoding Data**

---

```
SET yesNoIndex = {'Ya':1,'Tidak':0}
SET dataFilter = dataFilter.replace(yesNoIndex)
```

---

## **Modul Program 4.3 Proses Encoding Data**

Setelah modul program diatas dijalankan, maka semua data kategorikal yang sebelumnya bernilai “Tidak” dan “Ya” akan diubah menjadi bentuk numerik 0 dan 1. Berikut merupakan hasil dari proses *encoding data* dapat dilihat pada Gambar 4.3.

|     | IPGenap | OrganisasiGenap | KerjaGenap |
|-----|---------|-----------------|------------|
| 0   | 3.96    | 0               | 0          |
| 1   | 3.80    | 1               | 0          |
| 2   | 3.70    | 1               | 0          |
| 3   | 2.83    | 0               | 0          |
| 4   | 3.70    | 0               | 0          |
| ... | ...     | ...             | ...        |
| 205 | 3.48    | 0               | 1          |
| 206 | 3.60    | 0               | 0          |
| 207 | 3.11    | 1               | 0          |
| 208 | 4.00    | 1               | 0          |
| 209 | 3.55    | 1               | 0          |

Gambar 4.3 Proses *Encoding Data*

#### 4.1.3 Outlier Detection

Setelah didapatkan data yang bersih dari proses sebelumnya, selanjutnya akan dilakukan proses pendekripsi data-data yang bersifat *outlier*. Proses ini penting untuk dilakukan karena *K-Means* dinilai *sensitive* terhadap data-data *outlier* pada saat pengelompokan data. Sehingga hal ini dapat menyebabkan model *K-Means* yang dibagun mendapati kesalahan pada saat mengelompokkan data. Pada penelitian ini, metode *Local Outlier Factor* (LOF) akan digunakan untuk pendekripsi data-data yang bersifat *outlier*. Metode ini diimplementasikan dengan menggunakan *library sklearn*. Proses pendekripsi *outlier* dengan menggunakan LOF dapat dilakukan sebagai berikut.

---

##### Pseudocode 4: Local Outlier Factor

---

```

function localOutlierFactor(dataset)
    Input: dataset: DataFrame
    Output: dataWithoutOutlier: DataFrame

    set clf = LocalOutlierFactor(n_neighbors=20, contamination="auto")
    set X = dataset[['IP','Organisasi','Kerja']].values
    set y_pred = clf.fit_predict(X)
    set X_scores = clf.negative_outlier_factor_
    set round_off_values = np.around(X_scores, decimals =2)
    set new = round_off_values*(-1)
    set outlierData = detectOutlier(new)
    set dataWithoutOutlier = removeOutlier(dataset, outlierData)
    return dataWithoutOutlier
end function

```

---

#### Modul Program 4.4 Local Outlier Factor

Modul program diatas merupakan implementasi dari pendekripsi data *outlier* dengan menggunakan *Local Outlier Factor*. Pada fungsi tersebut akan menerima *input* berupa *dataset* yang akan dideteksi dan akan mengembalikan *dataset* yang telah bersih dari *outlier*. Pada tahapan pertama akan dipanggil metode dari LOF dengan beberapa parameter berupa *n\_neighbors* dan *contamination*. Kedua parameter itu merupakan parameter utama dimana *n\_neighbors* merupakan jumlah yang dianggap sebagai tetangga oleh sebuah titik data, sedangkan *contamination* merupakan tingkat kontaminasi outlier pada data. Pada penelitian ini, digunakan jumlah *n\_neighbors* = 20 dan *contamination* = “auto”. Setelah didapatkan hasil dari LOF, maka akan dilanjutkan pendekripsi data yang terdeteksi sebagai *outlier*. Untuk tahapan selanjutnya dapat dilakukan sebagai berikut.

---

#### **Pseudocode 5: Outlier Detection**

---

```
function detectOutlier(new)
    Input: new: list
    Output: outlierData: DataFrame

    set datas = pd.DataFrame(new)
    set outlier = []
    set i = 0
    for score in datas[0] do
        if score >= 1.5 then
            compute outlier.append(i)
        endif
        increment i
    endfor
    return outlierData
end function
```

---

#### **Modul Program 4.5 Outlier Detection**

Ketika sudah didapat data-data yang terdeteksi sebagai *outlier*, maka Langkah selanjutnya adalah dilakukan proses penghapusan data. Untuk tahapan penghapusan data dapat dilakukan sebagai berikut.

---

#### **Pseudocode 6: Outlier Removal**

---

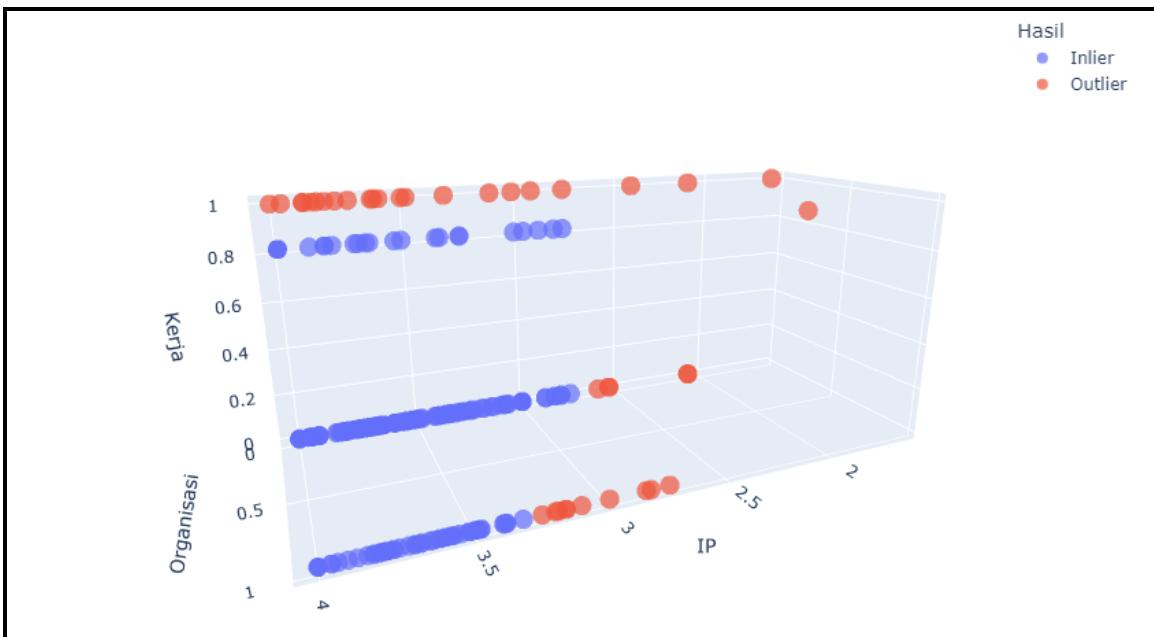
```
function removeOutlier(dataset, outlierData)
    Input: dataset, outlierData: DataFrame
    Output: dataWithoutOutlier: DataFrame

    compute dataset.drop(outlierData, inplace=True)
    set dataWithoutOutlier = dataset.reset_index(drop=True)
    return dataWithoutOutlier
end function
```

---

#### **Modul Program 4.6 Outlier Removal**

Setelah tahapan deteksi *outlier* telah selesai dilakukan, data akan terbagi menjadi dua yaitu data yang tergolong sebagai *outlier* dan *inlier*. Berikut merupakan hasil *plot* dari proses deteksi *outlier* pada Gambar 4.4.



**Gambar 4.4 Hasil Outlier Detection**

#### 4.1.4 Data Clustering

Setelah tahapan *preprocessing* dan *outlier detection*, maka data yang akan digunakan sudah bersih dan siap digunakan dalam pembuatan model. Model yang akan digunakan merupakan model *K-Means*. Namun, sebelum pembuatan model, *hyperparameter tuning* akan dilakukan terlebih dahulu untuk mendapatkan nilai parameter k terbaik agar didapatkan hasil pengelompokkan data yang terbaik.

---

##### Pseudocode 7: Elbow Method

---

```

function elbowMethod(dataset):
    Input: dataset: DataFrame
    Output: fig_elbow: plotly.express
    Output: optimumValue: int
    Output: optimumScore: double

    set optimumValue = 0
    set optimumScore = 0
    set ssd = []
    set range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
    for num_clusters in range_n_clusters do
        set kmeans = KMeans(n_clusters=num_clusters, max_iter=300)
        compute kmeans.fit(dataset)
        compute ssd.append(kmeans.inertia_)
        if (kmeans.inertia_ > optimumScore) then
            set optimumScore = kmeans.inertia_
            set optimumValue = num_clusters
    end for
    set fig_elbow = px.line(x=range_n_clusters, y=ssd, labels={'x': 'Cluster', 'y': 'Elbow SSE(Sum of Square Error)'})
    return fig_elbow, optimumValue, optimumScore
end function

```

---

##### Modul Program 4.7 Elbow Method

---

**Pseudocode 8: Silhouette Score**

---

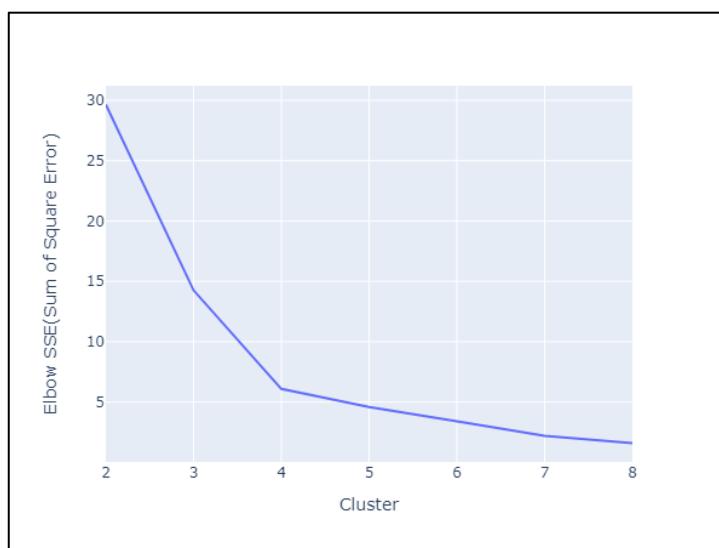
```
function silhouetteMethod(dataset)
    Input: dataset: DataFrame
    Output: figSilhouette: plotly.express
    Output: optimumSilhouetteValue: int
    Output: optimumSilhouetteScore: double

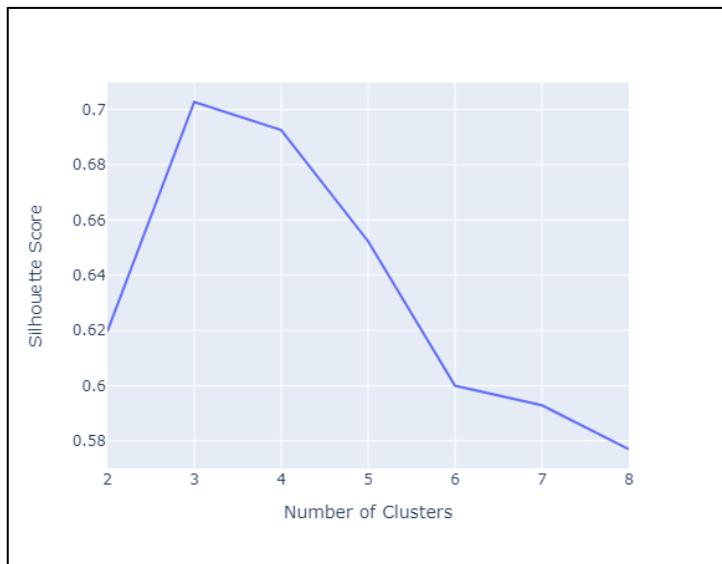
    set range_n_clusters = [2, 3, 4, 5, 6, 7, 8]
    set optimumScore = 0
    set optimumValue = 0
    set results_sil = {}
    for num_clusters in range_n_clusters do
        set kmeans = KMeans(n_clusters=num_clusters, max_iter=300)
        compute kmeans.fit(dataset)
        compute cluster_labels = kmeans.labels_
        set silhouette_avg= silhouette_score(dataset, cluster_labels)
        compute results_sil.update({num_clusters: silhouette_avg})
        if silhouette_avg > optimumScore then
            set optimumValue = num_clusters
            set optimumScore = silhouette_avg
    set fig_silhouette = px.line( x= list(results_sil.keys()),
y=list(results_sil.values()), labels={'x' : 'Number of Clusters', 'y' : 'Silhouette Score'})
    return fig_silhouette, optimumValue, optimumScore
end function
```

---

**Modul Program 4.8 Silhouette Score**

Proses dari *hyperparameter tuning* dapat dilakukan dengan menerapkan *Elbow Method* yang didukung dengan bantuan *Silhouette Score*. Hal ini dilakukan karena hasil dari *Elbow Method* terkadang kurang menunjukkan nilai parameter k terbaik. Oleh karena itu, hasil dari *Elbow Method* akan dilakukan pencocokan dengan hasil perhitungan dari *Silhouette Score*. Proses dari *hyperparameter tuning* dapat dilakukan seperti pada Modul Program 4.7 dan Modul Program 4.8.

**Gambar 4.5 Hasil Visualisasi Elbow Method**



**Gambar 4.6 Hasil Visualisasi Silhouette Score**

Setelah proses dari *Elbow Method* dan *Silhouette Score* telah dijalankan, akan dihasilkan nilai optimal dan *plot* untuk setiap proses. Hasil visualisasi setiap proses dalam bentuk *plot* pada Gambar 4.5 dan Gambar 4.6.

**Tabel 4.1 Silhouette Score**

| K        | Silhouette Score |
|----------|------------------|
| 2        | 0.6174223        |
| <b>3</b> | <b>0.7024898</b> |
| 4        | 0.6887586        |
| 5        | 0.6806782        |
| 6        | 0.6292535        |
| 7        | 0.5884822        |
| 8        | 0.5709499        |

Dari Gambar 4.5 dan Gambar 4.6 dapat dilihat hasil dari *Elbow Method* yaitu parameter k yang optimal terletak pada nilai k = 3 atau k = 4. Sedangkan hasil dari *Silhouette Score* seperti pada Tabel 4.1, nilai parameter K yang mendapatkan nilai tertinggi terdapat pada nilai k = 3 dengan score 0.7027509. Dari hasil kedua metode tersebut dapat disimpulkan nilai parameter K yang terbaik didapatkan dengan mengambil irisan hasil yaitu nilai k = 3 yang dapat digunakan dalam pembuatan model *K-Means*.

Setelah didapatkan nilai parameter k yang dinilai optimal, maka proses selanjutnya adalah dilakukan pembuatan model menggunakan metode *K-Means*. Berikut adalah implementasi kode program yang dapat dilakukan sebagai berikut.

---

**Algoritma 9: K-Means Model**

---

```
function kmeansMethod(dataset, k)
    Input dataset: DataFrame
    Input k: int
    Output kmeans: sklearn.cluster
    Output labels, samplesCentroids: DataFrame

    set kmeans = KMeans(n_clusters=k, max_iter=300, random_state= 56)
    compute kmeans.fit(dataset)
    set labels = kmeans.predict(dataset)
    set samplesCentroids = kmeans.cluster_centers_[labels]
    return kmeans, labels, samplesCentroids
end function
```

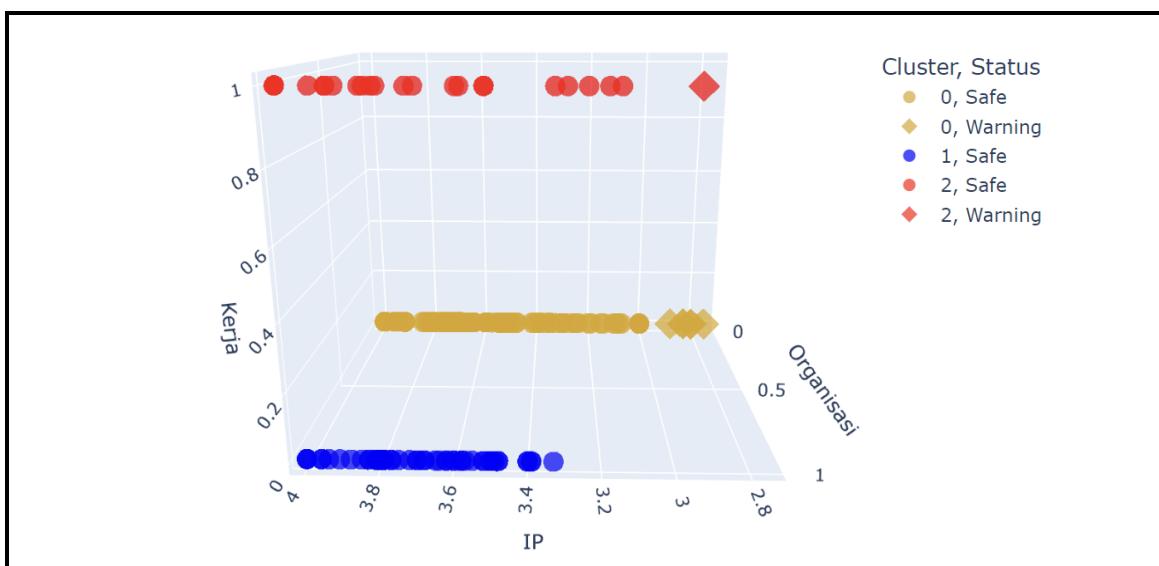
---

**Modul Program 4.9 K-Means Model**

Modul program diatas merupakan tahapan dalam pembuatan model *K-Means* pada proses *clustering* data. Dalam proses pembuatan model dibutuhkan input berupa nilai parameter k yang telah optimal dan dataset. Setelah model telah dibuat, proses pelatihan akan dilakukan dengan menggunakan fungsi *fit* terhadap dataset. Setelah proses selesai, akan dihasilkan label untuk setiap data yang menjelaskan akan pembagian cluster untuk setiap data. Berikut merupakan hasil dari pembagian cluster dari dataset yang telah dilatih dan dapat dilihat pada Gambar 4.7 dan Tabel 4.2 berikut.

**Tabel 4.2 Pembagian Cluster Hasil Proses K-Means**

| Cluster | Count |
|---------|-------|
| 0       | 99    |
| 1       | 51    |
| 2       | 22    |

**Gambar 4.7 Hasil Clustering Data****4.1.5 Model Evaluation**

Pada penelitian ini, tahap evaluasi dilakukan dengan menguji model yang telah dibangun sebelumnya. Proses evaluasi model pada tahapan ini akan menggunakan

*silhouette score* dengan nilai parameter k terbaik. Pada skenario pengujian dilakukan dengan mencari jarak antar data dalam *cluster* yang berbeda dan juga jarak antar data dalam *cluster* yang sama. Skenario pengujian yang dilakukan adalah dengan menguji model dalam melakukan proses *clustering* dengan menggunakan nilai parameter k terbaik. Proses evaluasi dengan menggunakan *silhouette score* dapat dilakukan seperti pada Modul Program 4.8. Pada modul tersebut, digunakan fungsi *silhouette\_score* yang akan menghasilkan nilai dari rentang -1 hingga 1.

Selain dengan *silhouette score*, pada penelitian ini juga digunakan visualisasi akan persebaran data di dalam *cluster* untuk mengetahui nilai *silhouette score* pada setiap data. Visualisasi untuk *silhouette score* dapat dilakukan dengan menerapkan kode program seperti berikut ini.

---

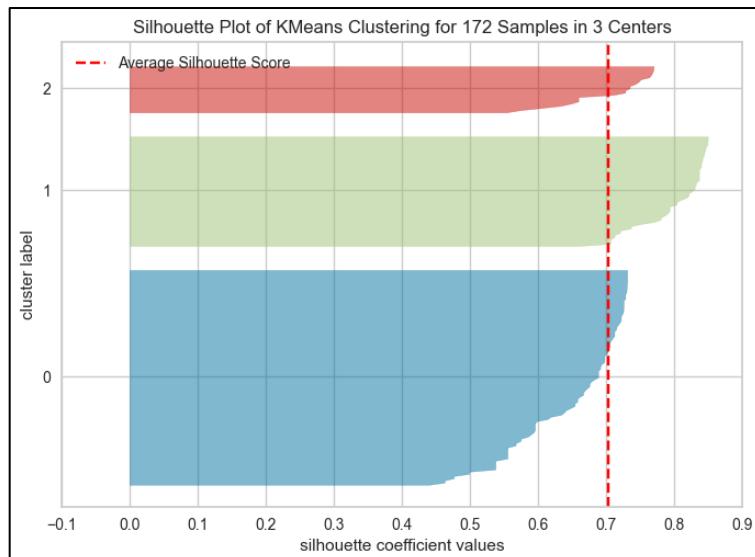
#### Algoritma 10: Silhouette Visualization

---

```
set dataEvaluasi = dataFinal[['IP','Organisasi','Kerja']]
set visualizer = SilhouetteVisualizer(kmeansModel,
colors='yellowbrick')
compute visualizer.fit(dataEvaluasi.to_numpy())
compute visualizer.finalize()
```

---

#### Modul Program 4.10 Silhouette Visualization



Gambar 4.8 Silhouette Visualization

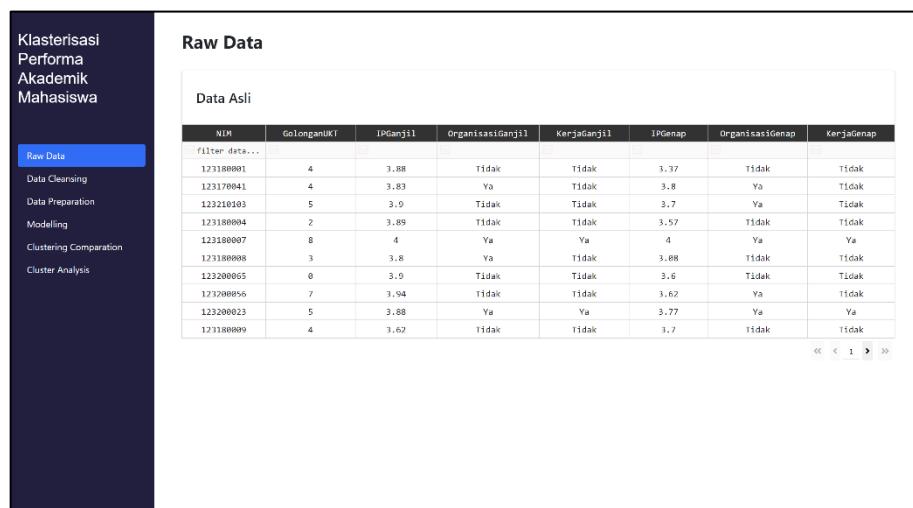
Untuk melakukan evaluasi pada model, dilakukan dengan cara menggunakan fungsi *SilhouetteVisualizer()* dengan parameter berupa model *K-Means* yang telah dibangun sebelumnya. Setelah itu akan dilakukan proses *visualizer.fit()* untuk menerapkan data pada visualisasi yang telah dibuat. Selanjutnya, *visualizer.finalize()* berguna untuk menampilkan visualisasi dalam bentuk akhir (*final*) seperti penambahan *title* serta *label x* dan *y*. Untuk hasil visualisasi yang telah dibangun dapat dilihat pada Gambar 4.8.

Pada penelitian ini, nilai parameter K yang digunakan dalam pembuatan model *clustering* adalah K = 3. Pada Gambar 4.8, setiap data yang berada di *cluster* 0, 1, maupun 2 memiliki *silhouette score* yang berada diatas 0.4 dengan nilai rata-rata sebesar 0.7024898.

Jika mengacu pada Tabel 2.1, maka nilai tersebut sudah tergolong sangat baik dalam evaluasi model struktur sebuah *cluster* karena tergolong sebagai struktur yang kuat.

#### 4.1.6 System Implementation

Aplikasi yang telah dirancang sebelumnya akan dikembangkan dengan menggunakan *platform* berbasis *website*. Aplikasi yang dibangun menggunakan bahasa pemrogramman *Python* dengan menggunakan *framework* *Python Dash*. *Framework* tersebut digunakan untuk membangun *user interface* dari aplikasi yang telah dibangun. Pada penelitian ini, dibangun 6 buah halaman yang akan menampilkan setiap proses dari pembuatan aplikasi ini. Untuk hasil dari *user interface* yang telah dibangun dapat dilihat pada gambar-gambar berikut ini.



The screenshot shows a dark-themed user interface for a web application. On the left, a sidebar menu lists: Klasterisasi Performa Akademik Mahasiswa, Raw Data (selected), Data Cleaning, Data Preparation, Modelling, Clustering Comparison, and Cluster Analysis. The main content area is titled 'Raw Data' and contains a sub-section 'Data Asli'. It displays a table with 10 rows of data, each representing a student record with columns: NIM, GolonganKTI, IPGenjil, OrganisasiGenjil, KerjaGenjil, IPGenap, OrganisasiGenap, and KerjaGenap. The data includes various numerical and categorical values. At the bottom right of the table, there are navigation buttons for pagination: <<, <, 1, >, >>.

Gambar 4.9 Halaman *Raw Data*

Pada halaman *Raw Data*, akan ditampilkan data asli yang belum dikenakan proses dan merupakan data yang secara langsung didapatkan dari penyebaran kuisioner. Data akan ditampilkan dalam bentuk *pagination* dengan setiap halaman akan diwakilkan dengan 10 data seperti pada Gambar 4.9.



The screenshot shows the 'Data Cleaning' section of the application. The sidebar remains the same. The main content area is titled 'Data Cleaning' and contains two sub-sections: 'Remove Unused and Null Data' and 'Encoding Data'. The 'Remove Unused and Null Data' section displays a table with 10 rows of data, likely representing cleaned data, with columns: IPGenap, OrganisasiGenap, and KerjaGenap. The 'Encoding Data' section displays a table with 2 rows of data, showing the encoding of categorical variables, with columns: IPGenap, OrganisasiGenap, and KerjaGenap. Both sections include a 'filter data...' button at the top and navigation buttons at the bottom right.

Gambar 4.10 Halaman *Data Cleaning – Remove Unused and Null Data*

Pada halaman *Cleansing Data*, akan ditampilkan data yang telah dilakukan *preprocessing* seperti penghapusan kolom-kolom yang tidak akan digunakan dalam penelitian ini seperti *timestamp*, *email*, nama lengkap, Nomor Induk Mahasiswa (NIM), dan angkatan. Selain itu, akan diterapkan pula proses *encoding data* yang dimana akan mengubah data yang bersifat kategorikal menjadi tipe numerik. Untuk itu, setiap data yang bernilai “Tidak” akan diubah menjadi angka 0 dan data yang bernilai “Ya” akan diubah menjadi angka 1 seperti pada Gambar 4.10 dan Gambar 4.11.

The screenshot shows a user interface for data processing. On the left, a sidebar menu lists "Klasterisasi Performa Akademik Mahasiswa" and several sub-options: "Raw Data", "Data Cleaning" (which is highlighted in blue), "Data Preparation", "Modelling", "Clustering Comparison", and "Cluster Analysis".

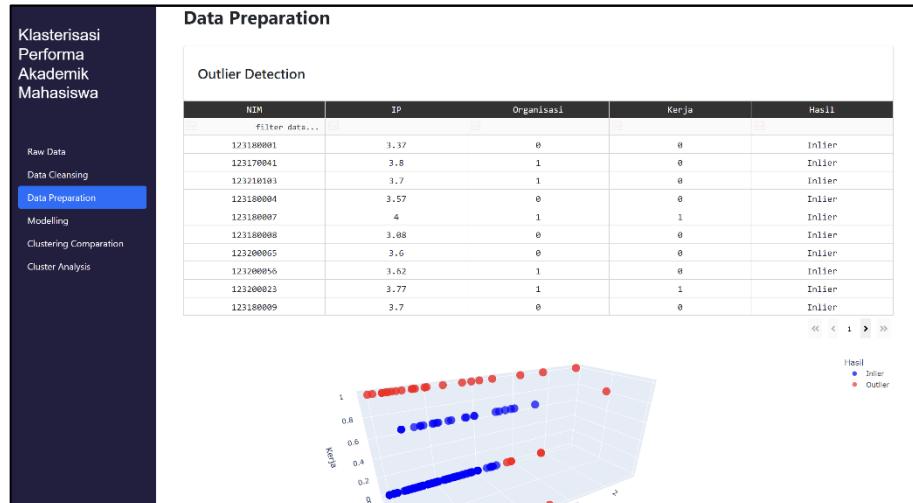
**Raw Data:**

| NIM  | IP    | Kerja | Hasil |
|------|-------|-------|-------|
| 3.7  | Ya    | Tidak | Tidak |
| 3.57 | Tidak | Ya    | Tidak |
| 4    | Ya    | Ya    | Tidak |
| 3.08 | Tidak | Tidak | Tidak |
| 3.6  | Tidak | Tidak | Tidak |
| 3.62 | Ya    | Tidak | Tidak |
| 3.77 | Ya    | Ya    | Tidak |
| 3.7  | Tidak | Tidak | Tidak |

**Encoding Data:**

| IPGenap        | OrganisasiGenap | KerjaGenap |
|----------------|-----------------|------------|
| filter data... | 0               | 0          |
| 3.37           | 0               | 0          |
| 3.8            | 1               | 0          |
| 3.7            | 1               | 0          |
| 3.57           | 0               | 0          |
| 4              | 1               | 1          |
| 3.08           | 0               | 0          |
| 3.6            | 0               | 0          |
| 3.62           | 1               | 0          |
| 3.77           | 1               | 1          |
| 3.7            | 0               | 0          |

Gambar 4.11 Halaman *Data Cleaning – Encoding Data*



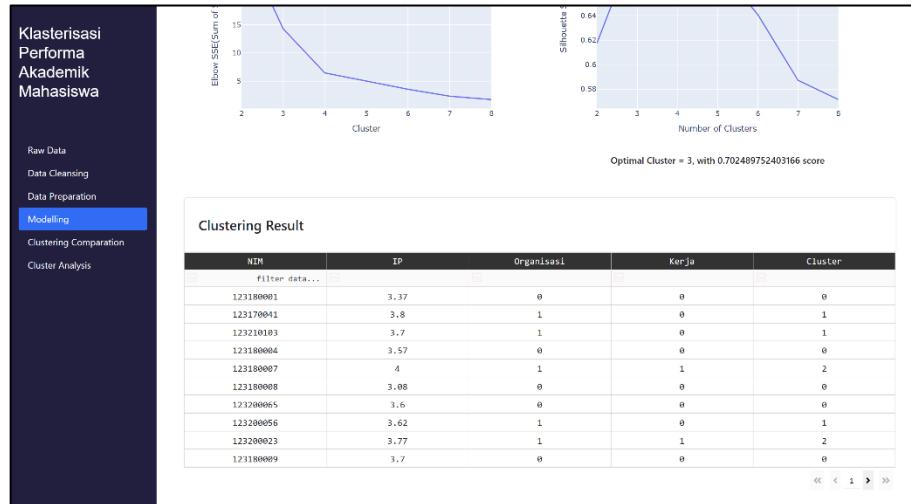
Gambar 4.12 Halaman *Data Preparation*

Pada halaman *Data Preparation*, akan ditampilkan hasil pendektsian anomali *outlier* pada setiap data yang tersedia. Proses pendektsian ini akan menghasilkan 2 buah kategori yaitu data *inlier* dan data *outlier*. Data tersebut akan ditampilkan dalam bentuk *pagination* dengan setiap halaman akan diwakilkan dengan 10 data seperti pada Gambar 4.12.



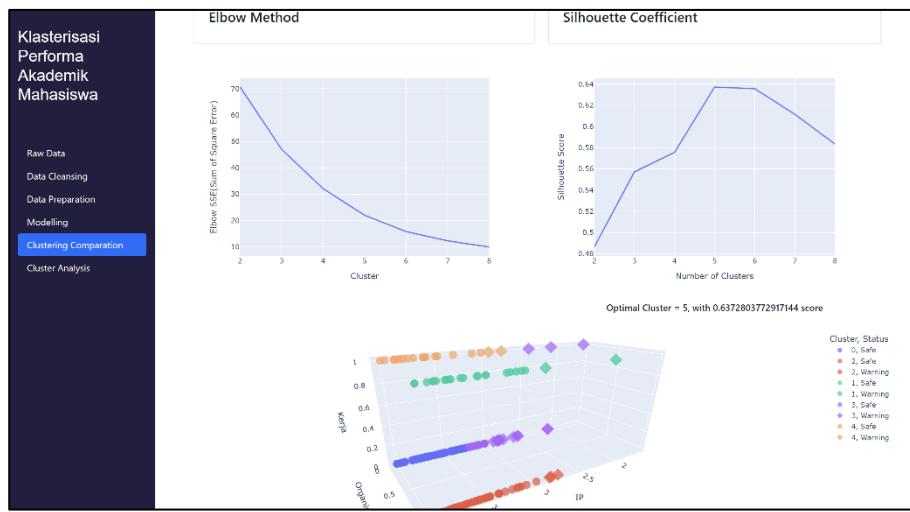
Gambar 4.13 Halaman *Modelling – Hyperparameter Tuning*

Pada halaman *Modelling*, akan ditampilkan dua buah *plot* yaitu *plot* hasil dari *Elbow Method* dan *Silhouette Score*. Selain itu juga akan ditampilkan hasil pengelompokan data dengan menggunakan *K-Means* yang telah terbagi menjadi beberapa *cluster*. Visualisasi dari halaman ini dapat dilihat pada Gambar 4.13 dan Gambar 4.14.

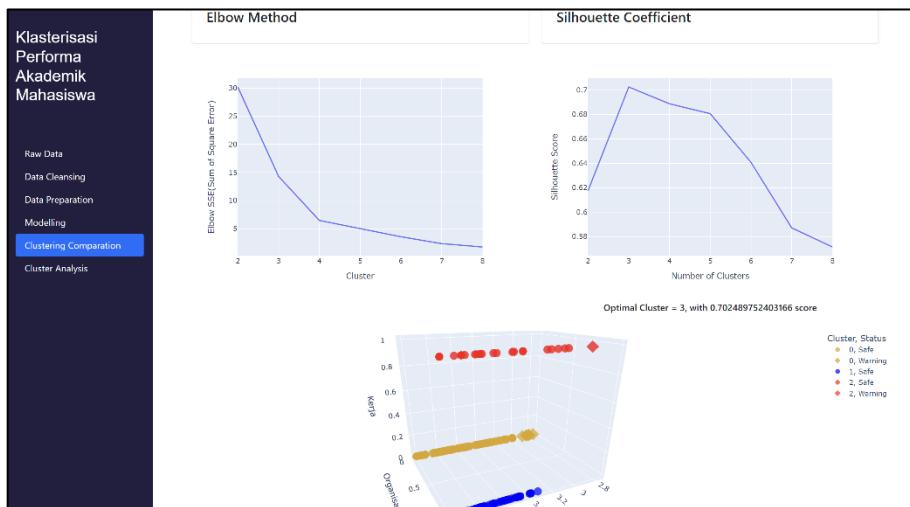


Gambar 4.14 Halaman *Modelling – Clustering Result*

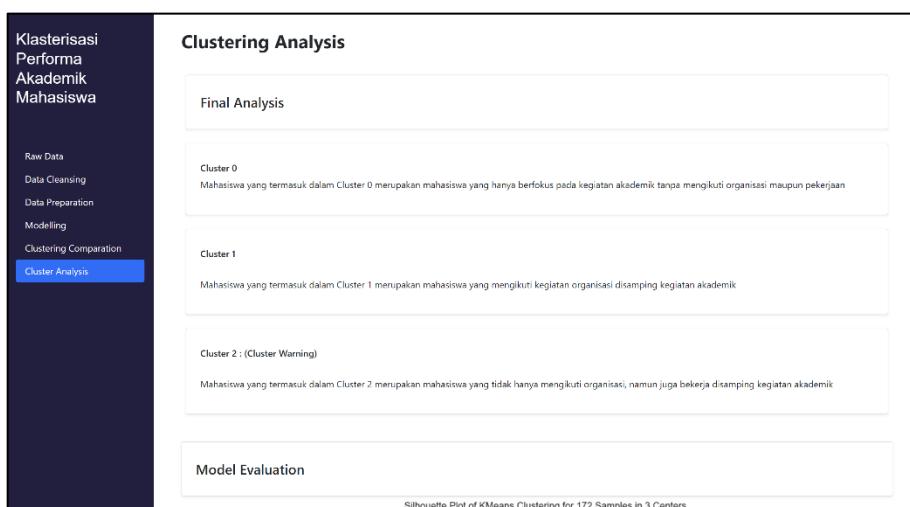
Pada halaman *Clustering Comparation*, akan ditampilkan perbandingan antara proses pengelompokan data dengan dan tanpa menggunakan optimasi *Local Outlier Factor*. Untuk setiap proses akan ditampilkan dua buah *plot* yaitu *plot* hasil dari *Elbow Method* dan *Silhouette Score*. Selain itu juga akan ditampilkan hasil pengelompokan data dengan menggunakan *K-Means* yang telah terbagi menjadi beberapa *cluster* dalam bentuk *scatter plot*. Visualisasi dari halaman ini dapat dilihat pada Gambar 4.15 dan Gambar 4.16.



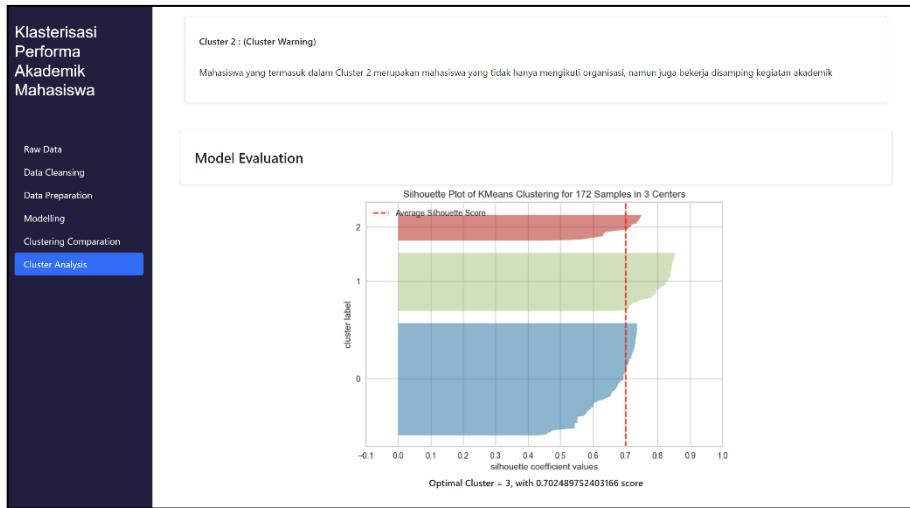
Gambar 4.15 Halaman *Clustering Comparation – Before Optimization*



Gambar 4.16 Halaman *Clustering Comparation – After Optimization*



Gambar 4.17 Halaman *Cluster Analysis – Final Analysis*



**Gambar 4.18 Halaman Cluster Analysis – Model Evaluation**

Pada halaman *Cluster Analysis*, akan ditampilkan hasil dari interpretasi setiap *cluster* yang terbentuk dari proses *clustering*. Selain itu, pada halaman ini juga akan ditampilkan hasil dari pengujian model dengan menampilkan *Silhouette Visualization*. Untuk visualisasi dari halaman ini dapat dilihat pada Gambar 4.17 dan Gambar 4.18.

#### 4.1.7 System Testing

Setelah aplikasi selesai diimplementasikan, tahapan selanjutnya adalah dilakukan pengujian aplikasi. Tahapan ini dilakukan agar aplikasi yang dibangun berfungsi secara penuh tanpa adanya error saat dijalankan. Dalam penelitian ini, pengujian aplikasi dilakukan dengan mengukur kinerja dan fungsionalitas dari aplikasi yang dibuat. *Detail* hasil pengujian aplikasi dapat dilihat pada Tabel 4.3 berikut ini.

**Tabel 4.3 Hasil Pengujian Aplikasi**

| No | Halaman                 | Pengujian   | Hasil   |       |
|----|-------------------------|---|---|-------|
|    |                         |   | Berhasil  | Gagal |
| 1. | <i>Data Raw</i>         | Menampilkan 10 data asli  | Aplikasi berhasil menampilkan seluruh data dalam bentuk <i>pagination</i> dengan 10 data pada setiap <i>page</i>                                | -     |
| 2. | <i>Data Cleansing</i>   | Menampilkan 10 data setelah dilakukan penghapusan kolom yang tidak terpakai | Aplikasi berhasil menampilkan hasil penghapusan kolom yang tidak terpakai dalam bentuk <i>pagination</i> dengan 10 data pada setiap <i>page</i> | -     |
| 3. |                         | Menampilkan 10 data setelah dilakukan proses <i>encoding data</i>           | Aplikasi berhasil menampilkan hasil <i>encoding data</i> dalam bentuk <i>pagination</i> dengan 10 data pada setiap <i>page</i>                  | -     |
| 4. | <i>Data Preparation</i> | Menampilkan plot data yang terdeteksi sebagai <i>outlier</i>                | Aplikasi berhasil menampilkan plot data yang terdeteksi sebagai <i>outlier</i>  | -     |
| 5. |                         | Menampilkan 10 data setelah dilakukan pendekripsi <i>outlier</i>            | Aplikasi berhasil menampilkan hasil pendekripsi <i>outlier</i> dalam bentuk <i>pagination</i> dengan 10 data pada setiap <i>page</i>            | -     |

**Tabel 4.4 Hasil Pengujian Aplikasi Lanjutan**

|     |                               |   |  |   |
|-----|-------------------------------|---|--|---|
| 6.  | <i>Modelling</i>              | Menampilkan plot dari <i>Elbow Method</i>   | Aplikasi berhasil menampilkan plot dari <i>Elbow Method</i>  | - |
| 7.  |                               | Menampilkan tabel data setelah proses <i>clustering</i>                             | Aplikasi berhasil menampilkan hasil proses <i>clustering</i> dalam bentuk <i>pagination</i> dengan 10 data pada setiap <i>page</i> | - |
| 8.  |                               | Menampilkan plot dari <i>Silhouette Score</i>                                       | Aplikasi berhasil menampilkan plot dari <i>Silhouette Score</i>  | - |
| 9.  | <i>Clustering Comparation</i> | Menampilkan plot dari <i>Elbow Method</i> sebelum dan setelah optimasi              | Aplikasi berhasil menampilkan plot dari <i>Elbow Method</i> sebelum dan setelah optimasi   | - |
| 10. |                               | Menampilkan plot dari <i>Silhouette Score</i> sebelum dan setelah optimasi          | Aplikasi berhasil menampilkan plot dari <i>Silhouette Score</i> sebelum dan setelah optimasi                                       | - |
| 11. |                               | Menampilkan plot data setelah proses <i>clustering</i> sebelum dan setelah optimasi | Aplikasi berhasil menampilkan plot data setelah proses <i>clustering</i> sebelum dan setelah optimasi                              | - |
| 12. | <i>Cluster Analysis</i>       | Menampilkan hasil interpretasi <i>cluster</i>                                       | Aplikasi berhasil menampilkan hasil interpretasi <i>cluster</i>  | - |
| 13. |                               | Menampilkan plot <i>Silhouette Visualization</i> sesuai dengan nilai K Terbaik      | Aplikasi berhasil menampilkan <i>Silhouette Visualization</i> yang sesuai dengan nilai K terbaik                                   | - |

## 4.2 Pembahasan

Pada bagian pembahasan ini akan dijelaskan hasil yang diperoleh untuk masalah penelitian yang diangkat. Masalah pada penelitian ini yaitu *K-Means* yang *sensitive* terhadap anomali *outlier* yang akan dioptimasi dengan bantuan *Local Outlier Factor*. Selain itu, penelitian ini juga akan membandingkan kinerja *K-Means* sebelum dan setelah dilakukan optimasi.

Dalam penelitian ini digunakan data performa akademik mahasiswa yang didapatkan dari proses penyebaran kuisioner yang berjumlah 210 data. Kemudian data tersebut akan dilakukan preprocessing agar data yang akan digunakan menjadi bersih dan siap digunakan. Setelah itu akan dilakukan proses pendekripsi outlier pada data dengan bantuan *Local Outlier Factor*. Pada penelitian ini, sebanyak 38 data telah terdeteksi sebagai outlier. Data-data tersebut akan dihapus dan tidak akan digunakan dalam pembuatan model clustering. Data yang telah bersih akan digunakan untuk membuat model dengan nilai parameter K = 3. Pengujian model diterapkan dengan menggunakan *Silhouette Score* dengan nilai 0.7027509 yang termasuk sangat baik dalam evaluasi model struktur sebuah cluster karena tergolong sebagai struktur yang kuat. Langkah terakhir yaitu pengembangan aplikasi dilakukan dengan menggunakan Python Dash berbasis website agar informasi dalam aplikasi dapat diambil dan digunakan dengan mudah.

Dari seluruh tahapan tersebut dibutuhkan beberapa skenario pengujian agar model *K-Means* yang dibangun dapat menghasilkan kinerja yang baik. Untuk hal itu, diterapkan *hyperparameter tuning* pada nilai parameter K seperti yang tertera pada Tabel 4.1. Optimasi tersebut memperoleh hasil terbaik pada nilai parameter K = 3 dengan nilai silhouette score

sebesar 0.7024898. Pengujian juga dilakukan pada model *K-Means* tanpa *Local Outlier Factor* dengan skenario pengujian yang sama yaitu nilai parameter K = 2 hingga K = 8. Pengujian tersebut menghasilkan nilai terbaik pada nilai parameter K = 5 dengan nilai silhouette score sebesar 0.6372808. Dari kedua jenis pengujian tersebut, terdapat peningkatan *silhouette score* sebesar 10.23% dan terbukti model *K-Means* yang dioptimasi dengan *Local Outlier Factor* memiliki kinerja dan struktur cluster yang lebih baik.

Berdasarkan pada Gambar 4.7, data yang digunakan pada penelitian ini akan dibagi menjadi 3 buah cluster yaitu *cluster 0*, *cluster 1*, dan *cluster 2*. *Cluster 0* ditandai dengan warna kuning dan memiliki nilai 0 pada parameter organisasi dan pekerjaan. Hal ini dapat diinterpretasikan dengan mahasiswa yang hanya berfokus pada kegiatan akademik tanpa mengikuti organisasi maupun pekerjaan. Untuk *cluster 1* ditandai dengan warna biru dan memiliki nilai 0 pada parameter pekerjaan dan nilai 1 pada parameter organisasi. Hal ini dapat diartikan dengan mahasiswa yang mengikuti organisasi disamping kegiatan akademik. Lain halnya dengan mahasiswa yang berada pada *cluster 2*, *cluster* ini ditandai dengan warna merah dengan nilai 1 pada parameter organisasi dan pekerjaan. Hal ini dapat diartikan dengan mahasiswa yang tidak hanya mengikuti organisasi, namun juga bekerja disamping kegiatan akademik.

Untuk mendapatkan *cluster* yang memerlukan perhatian khusus atau dapat dikategorikan sebagai *cluster warning*, diperlukan analisis lebih lanjut pada setiap cluster. Jika dilihat pada *cluster 0*, data-data dinilai menyebar pada parameter IP dan memiliki kepadatan data yang tinggi. Selain itu, pada *cluster 1*, data-data juga memiliki kepadatan yang tinggi dan memiliki nilai IP diatas 3.40. Sedangkan pada *cluster 2*, data-data juga dinilai menyebar untuk parameter IP namun memiliki kerenggangan antar data. Jika dilihat pada analisis tersebut, *cluster 2* dapat dikategorikan sebagai *cluster warning* karena mahasiswa yang berada pada *cluster* tersebut dapat dinilai memiliki performa akademik yang kurang stabil. Selain dari *cluster warning*, mahasiswa yang memiliki IP dibawah 3 juga dapat dikategorikan sebagai mahasiswa yang diberi *warning* sehingga mahasiswa-mahasiswa tersebut tetap akan mendapatkan perhatian khusus dalam proses monitoring performa akademik.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan penelitian yang telah dilakukan, didapatkan hasil yang dapat dinyatakan dalam beberapa poin berikut ini:

- a. Klasterisasi performa akademik mahasiswa berhasil diterapkan dengan menggunakan metode *K-Means* yang dioptimasi dengan *Local Outlier Factor* pada 210 data. Penerapan metode ini dapat mendeteksi sebanyak 38 data atau sebesar 18.10% data *outlier*.
- b. Gabungan metode *K-Means* dan *Local Outlier Factor* terbukti dapat mempengaruhi nilai *silhouette score* pada proses *clustering* dengan nilai 0.7024898. Gabungan kedua metode tersebut menghasilkan struktur *cluster* yang kuat dengan bukti nilai *silhouette score* diatas 0.7.
- c. Penerapan metode *Local Outlier Factor* pada *K-Means* terbukti dapat membuat proses klasterisasi data menjadi lebih optimal. Hal tersebut dibuktikan dengan adanya peningkatan sebesar 10.23% setelah metode *Local Outlier Factor* diterapkan.
- d. Dengan mengelompokkan mahasiswa menjadi beberapa *cluster*, maka dapat diketahui bahwa pada *cluster* 0 lebih berfokus pada kegiatan akademik, *cluster* 1 mengikuti organisasi disamping kegiatan akademik, dan *cluster* 2 disamping kegiatan akademik juga mengikuti organisasi dan bekerja. Dengan analisis lanjutan, *cluster* 2 dapat dikategorikan sebagai *cluster warning* karena mahasiswa yang berada pada cluster tersebut dapat dinilai memiliki performa akademik yang kurang stabil.
- e. Dilakukan proses optimasi pada model dengan melakukan proses *hyperparameter tuning* dengan menerapkan gabungan *Elbow Method* dan *Silhouette Score* yang menghasilkan nilai parameter K terbaik yaitu nilai K = 3.

#### **5.2 Saran**

Dari penelitian yang telah dilakukan, terdapat kekurangan yang dapat dikembangkan untuk penelitian selanjutnya. Adapun saran yang dapat diberikan diantaranya:

- a. Penambahan kriteria pada parameter data performa akademik dari mahasiswa agar pada saat proses *clustering* data lebih tersebar dan terkelompok lebih baik.
- b. Jika parameter pada data yang digunakan merupakan data kategorikal, maka *value* dari parameter tidak hanya 2 jenis (“Ya” dan “Tidak”).
- c. Penambahan *database* yang dapat menyimpan data hasil pengelompokan sebelumnya. Selain itu juga ditambahkan proses *input* data pada aplikasi yang dibangun.

## DAFTAR PUSTAKA

- Agrawal, T. (2021). Hyperparameter Optimization in Machine Learning. In *APRESS*.  
<https://doi.org/10.1007/978-1-4842-6579-6>
- Alghushairy, O., Alsini, R., Soule, T., & Ma, X. (2020). *A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams*. <https://doi.org/10.3390/bdcc>
- Amalia, A. E., & Naf'an, M. Z. (2017). Implementasi Algoritma ID3 Untuk Klasifikasi Performansi Mahasiswa (Studi Kasus ST3 Telkom Purwokerto). *Seminar Nasional Teknologi Informasi Dan Multimedia*, 115–120.
- Ariawan, P. A. (2019). Optimasi Pengelompokan Data Pada Metode K-means dengan Analisis Outlier. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 5(2), 88–95.  
<https://doi.org/10.25077/teknosi.v5i2.2019.88-95>
- Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is machine learning? A primer for the epidemiologist. *American Journal of Epidemiology*, 188(12), 2222–2239. <https://doi.org/10.1093/aje/kwz189>
- Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. *ACM SIGMOD Record*, 29(2), 93–104.
- Budiarto, E. H., Permanasari, A. E., & Fauziati, S. (2019). Unsupervised Anomaly Detection Using K-Means, Local Outlier Factor and One Class SVM. *International Conference on Science and Technology (ICST)*.
- Cheng, Z., Zou, C., & Dong, J. (2019). Outlier detection using isolation forest and local outlier. *Research in Adaptive and Convergent Systems (RACS)*, 161–168.  
<https://doi.org/10.1145/3338840.3355641>
- el Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In *Machine Learning in Radiation Oncology* (Vol. 1, pp. 3–11). Springer International Publishing.  
[https://doi.org/10.1007/978-3-319-18305-3\\_1](https://doi.org/10.1007/978-3-319-18305-3_1)
- Firzada, F., & Yunus, Y. (2021). Klasterisasi Tingkat Masa Studi Tepat Waktu Mahasiswa Menggunakan Algoritma K-Medoids. *Jurnal Sistim Informasi Dan Teknologi*, 3(3), 162–168. <https://doi.org/10.37034/jsisfotek.v3i3.60>
- Hardani, Andriani, H., Untiawaty, J., Utami, E. F., Istiqomah, R. R., Fardani, R. A., Sukmana, D. J., & Auliya, N. H. (2020). *Buku Metode Penelitian Kualitatif & Kuantitatif* (H. Abadi, Ed.; Vol. 1). CV. Pustaka Ilmu Group Yogyakarta.
- Himawan, D. (2014). *Aplikasi Data Mining Menggunakan Algoritma ID3 Untuk Mengklasifikasi Kelulusan Mahasiswa Pada Universitas Dian Nuswantoro Semarang*.
- Irawan, E., Siregar, S. P., Damanik, I. S., & Saragih, I. S. (2020). Implementasi K-Medoids untuk Pengelompokan Sebaran Mahasiswa Baru. *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, 5(2), 275–281.  
<https://tunasbangsa.ac.id/ejurnal/index.php/jurasik>

- Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 3(31), 685–695. <https://doi.org/10.1007/s12525-021-00475-2>/Published
- Kurniadi, D., & Sugiyono, A. (2020). Pengelompokkan Data Akademik Menggunakan Algoritma K-Means Pada Data Akademik Unissula. *TRANSFORMTIKA*, 18(1), 93–101.
- Madhulatha, T. S. (2012). An overview of clustering methods. *IOSR Journal of Engineering*, 2(4), 719–725. <https://doi.org/10.3233/ida-2007-11602>
- Muttaqin, M. R., & Defriani, M. (2020). Algoritma K-Means untuk Pengelompokan Topik Skripsi Mahasiswa. *ILKOM Jurnal Ilmiah*, 12(2), 121–129. <https://doi.org/10.33096/ilkom.v12i2.542.121-129>
- Nur, F., Fauzan, R., Aziz, J., Darma Setiawan, B., & Arwani, I. (2018). *Implementasi Algoritma K-Means untuk Klasterisasi Kinerja Akademik Mahasiswa* (Vol. 2, Issue 6). <http://j-ptiik.ub.ac.id>
- Pradnyana, G. A., & Permana, A. A. J. (2018). Sistem Pembagian Kelas Kuliah Mahasiswa Dengan Metode K-Means Dan K-Nearest Neighbors Untuk Meningkatkan Kualitas Pembelajaran. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 16(1), 59–68.
- Qomariyah, & Siregar, U. M. (2022). Comparative Study of K-Means Clustering Algorithm and K-Medoids Clustering in Student Data Clustering. *Jurnal Informatika Sunan Kalijaga (JISKA)*, 7(2), 91–99.
- Rahmawati, E., Herry Chrisnanto, Y., & Maspupah, A. (2019). Identifikasi Kemampuan Akademik Mahasiswa Menggunakan K-Means Clustering. *Seminar Nasional Inovasi Teknologi UN PGRI Kediri*, 87–92.
- Ramadhan, F., Chrisnanto, Y. H., & Ningsih, A. K. (2021). Sistem Segmentasi Keluhan Pelanggan di Perumda Air Minum Tirta Raharja Cimahi Menggunakan Metode K-Medoids. *Informatics and Digital Expert (INDEX)*, 3(1), 6–09. <http://index.unper.ac.id>
- Ridwan, M., Suyono, H., & Sarosa, M. (2013). Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier. *Jurnal EECCIS*, 7(1), 59–64.
- Rosadi, R., Akmal, Hidayat, A., & Kharismawan, B. (2016). Aplikasi K-Means Clustering Untuk Mengelompokan Data Kinerja Akademik Mahasiswa. *SENTER*, 92–96. <http://unpad.ac.id>,
- Rustam, S., & Annur, H. (2019). Akademik Data Mining (Adm) K-Means Dan K-Means K-Nn Untuk Mengelompokan Kelas Mata Kuliah Kosentrasi Mahasiswa Semester Akhir. *ILKOM Jurnal Ilmiah*, 11(3), 260–268. <https://doi.org/10.33096/ilkom.v11i3.487.260-268>
- Saputra, D. M., Saputra, D., & Oswari, L. D. (2020). Effect of Distance Metrics in Determining K-Value in K-Means Clustering Using Elbow and Silhouette Method.

*Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)*, 172(Siconian 2019), 341–346.  
<https://doi.org/10.2991/aisr.k.200424.051>

- Saputra, H. K. (2018). Analisis Data Mining Untuk Pemetaan Mahasiswa Yang Membutuhkan Bimbingan dan Konseling Menggunakan Algoritma Naïve Bayes Classifier. *Jurnal Teknologi Informasi & Pendidikan*, 11(1), 14–26.
- Sari, B. N. (2016). Identification of Tuberculosis Patient Characteristics Using K-Means Clustering. *Scientific Journal of Informatics*, 3(2), 129–138.  
<https://doi.org/10.15294/sji.v3i2.7909>
- Sari, V. N., Yudianti, & Maharani, D. (2018). Penerapan Metode K-Means Clustering Dalam Menentukan Predikat Kelulusan Mahasiswa Untuk Menganalisa Kualitas Lulusan. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 4(2), 133–140.
- Shi, C., Wei, B., Wei, S., Wang, W., Liu, H., & Liu, J. (2021). A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. *Eurasip Journal on Wireless Communications and Networking*, 2021(31), 1–16.  
<https://doi.org/10.1186/s13638-021-01910-w>
- Sommerville, I. (2016). *Software Engineering* (Vol. 10).
- Sulistiyawati, A., & Supriyanto, E. (2021). Implementasi Algoritma K-means Clustering dalam Penetuan Siswa Kelas Unggulan. *TEKNO KOMPAK*, 15(2), 25–36.
- Vhallah, I., Sumijan, & Santony, J. (2018). Pengelompokan Mahasiswa Potensial Drop Out menggunakan Metode Clustering K-Means. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 2(2), 572–577. <http://jurnal.iaii.or.id>
- Yunita, F. (2018). Penerapan Data Mining Menggunakan Algoritma K-Means Clustering Pada Penerimaan Mahasiswa Baru (Studi Kasus : Universitas Islam Indragiri). In *Jurnal SISTEMASI* (Vol. 7).