# PROPOSAL

**PENERAPAN ALGORITMA K-MEDOIDS DALAM PENENTUAN FAKTOR TERBESAR SUMBER INFORMASI PEMILIHAN JURUSAN**

## DI FAKULTAS TEKNIK UNIVERSITAS HALU OLEO

Diajukan Untuk Memenuhi

Salah Satu Syarat Memperoleh Gelar Sarjana Teknik



**SASKIA RANDAWULA SILONDAE E1E1 16 034**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HALUOLEO**

## KENDARI 2020

## DAFTAR ISI

HALAMAN PENGESAHAN ................................................................................. ii

PERNYATAAN ..................................................................................................... iv

INTISARI ................................................................................................................ v

ABSTRACT ........................................................................................................... vi

HALAMAN PERSEMBAHAN ........................................................................... vii

KATA PENGANTAR ........................................................................................... ix

DAFTAR ISI .......................................................................................................... xi

DAFTAR GAMBAR ........................................................................................... xiv

DAFTAR TABEL ................................................................................................ xvi

BAB I ...................................................................................................................... 1

1.1 Latar Belakang .................................................................................................... 1

1.2 Rumusan Masalah ............................................................................................... 2

1.3 Batasan Masalah ................................................................................................. 2

1.4 Tujuan Penelitian ................................................................................................ 2

1.5 Manfaat Penelitan ............................................................................................... 3

1.6 Sistematika Penulisan ......................................................................................... 3

1.7 Tinjauan Pustaka ................................................................................................. 4 BAB II ..................................................................................................................... 6

2.1 Minat ................................................................................................................... 6

2.2 Data Mining ........................................................................................................ 6

2.3 Skala Likert ......................................................................................................... 7

2.4 K-Medoids atau Partitioning Around Medoids (PAM) ...................................... 7

2.5 Clustering .......................................................................................................... 10

2.6 Silhouette Coefficient ....................................................................................... 10

2.7 Principal Component Analysis (PCA) ............................................................. 11

2.8 Python ............................................................................................................... 11

2.9 Django Web Framework ................................................................................... 12

2.10 SQLite ............................................................................................................... 12

2.11 Database (Basis Data) ....................................................................................... 13

2.12 Unified Modeling Language (UML) ................................................................. 13

2.13 *Use Case Diagram* ..................................................................................... 14

2.15 Sequence Diagram ..................................................................................... 16

2.16 Class Diagram ............................................................................................ 17

2.17 Metode Pengembangan Sistem .................................................................. 18 BAB III ................................................................................................................. 20

3.1 Waktu Penelitian ............................................................................................... 20

3.2 Tempat Penelitian ......................................................................................... 20

3.3 Metode Pengumpulan Data ............................................................................... 21

3.4 Metode Pengembangan Sistem ......................................................................... 23

3.4.1 *Inception* / Permulaan ............................................................................... 23

3.4.2 *Elaboration* / Elaborasi ............................................................................. 23

3.4.3 *Construction* / konstruksi .......................................................................... 24

3.4.4 *Transition* / Transisi .................................................................................. 24 BAB IV ................................................................................................................. 25

4.1 Analisis Sistem .................................................................................................. 25

4.1.1 Analisis Kebutuhan Fungsional ................................................................ 25

4.1.2 Analisis Kebutuhan Nonfungsional .......................................................... 25

4.2 Analisis Sistem yang Direncanakan .................................................................. 26

4.3 Analisis Perancangan Sistem ............................................................................ 27

4.3.1 Perancangan Unified Modeling Language (UML) ................................... 27

4.4 Rancangan Antarmuka (Interface) .................................................................... 36 BAB V ................................................................................................................... 38

5.1 Implementasi Sistem ......................................................................................... 38 1.1.1 Tampilan Halaman *Login* ....................................................................... 38

5.1.2 Tampilan Halaman *Dashboard* ................................................................. 39

5.1.3 Tampilan Halaman Menu Data Mahasiswa .............................................. 39

5.1.4 Tampilan Halaman Menu Data Hasil Cluster ........................................... 40

5.1.5 Tampilan Halaman Visual ........................................................................ 40

5.2 Implementasi Algoritma ................................................................................... 42

5.2.1 Implementasi Algoritma *K-Medoids* ......................................................... 42

5.3 Pengujian Sistem ............................................................................................... 46

5.3.1 Pengujian *Black Box* ................................................................................. 46

5.4 Pengujian Algoritma ......................................................................................... 47

5.4.1 Poses Algoritma *K-Medoids* Pada System ................................................ 47 5.4.2 Perhitungan Manual dengan Metode *Clustering K-Medoids* .................... 47

5.5 *Silhoutette Coefficient* ................................................................................ 64

5.6 PLOT .......................................................................................................... 65 BAB VI ................................................................................................................. 67

6.1 Kesimpulan ....................................................................................................... 67

6.2 Saran ................................................................................................................. 67 DAFTAR PUSTAKA ........................................................................................... 68

## DAFTAR GAMBAR

Gambar 2.1 Proses *Clustering* Obyek Menggunakan metode *K-Means* ................. 7

Gambar 2.2 Ilustrasi *K-Medoids* ........................................................................... 11

Gambar 2.3 Proses Iteratif RUP ............................................................................ 19

Gambar 3.1 Pertanyaan Kuisioner ........................................................................ 22

Gambar 3.2 Pertanyaan Kuisioner ........................................................................ 23

Gambar 4.1 *Use Case Diagram* ............................................................................ 27

Gambar 4.2 *Activity Diagram Login* Admin ......................................................... 28

Gambar 4.3 *Activity Diagram* Mengelola Data Input Kuisioner .......................... 28

Gambar 4.4 *Activity Diagram* Mengelola Data Update Kuisioner ....................... 30

Gambar 4.5 Activity *Diagram* Mengelola Data Delete Kuisioner ....................... 31

Gambar 4.6 *Activity Diagram* Hasil *Cluster* ......................................................... 32

Gambar 4.7 *Sequence Diagram Login Admin* ...................................................... 33

Gambar 4.8 *Sequence Diagram* Input Data Kuisioner Mahasiswa ...................... 33

Gambar 4.9 *Sequence Diagram* Ubah Data Kuisioner Mahasiswa ...................... 34

Gambar 4.10 *Sequence Diagram* Hapus Data Kuisioner Mahasiswa ................... 35

Gambar 4.11 *Class Diagram* Faktor Terbesar Pemilihan Jurusan ........................ 35

Gambar 4.12 Halaman *Login* ................................................................................ 36

Gambar 4.13 Halaman Beranda ............................................................................ 36

Gambar 4.14 Halaman Form Input Data Kuisioner Mahasiswa ........................... 37

Gambar 4.15 Halaman Hasil Clustering ............................................................... 37

Gambar 5.1Tampilan Halaman *Login* ................................................................... 38

Gambar 5.2 Tampilan Halaman *Dashboard* ......................................................... 39

Gambar 5.3 Tampilan Halaman Menu Data Mahasiswa ...................................... 39

Gambar 5.4 Tampilan Halaman Menu Hasil Cluster ............................................ 40

Gambar 5.5 Tampilan Halaman *Visual* ................................................................. 41

Gambar 5.6 Proses Iterasi *K-Medoids* Pada *System* .............................................. 40

Gambar 5.7 Grafik pengaruh perbedaan jumlah klaster dengan 264 data terhadap

*silhouette coefficien.* .............................................................................................. 65 Gambar 5.8 Ilustrasi Plot....................................................................................... 65 Gambar 5.9 Ilustrasi Data Plot yang Sama ........................................................... 66

**DAFTAR TABEL**

Tabel 2.1 Sampel Data Perhitungan Bobot dari Kriteria ........................................ 8

Tabel 2.2 Sampel data yang dikonversi ................................................................ 10

Tabel 2.3 Simbol-simbol *Use Case Diagram* ....................................................... 19

Tabel 2.4 Simbol *Activity* Diagram ....................................................................... 16

Tabel 2.5 Simbol *Sequence* Diagram .................................................................... 16

Tabel 2.6 Simbol *Class* Diagram .......................................................................... 17

Table 2.7 Komponen-komponen *multiplicity* ....................................................... 17

Tabel 3.1*Gantt Chart* Waktu Penelitian................................................................ 23

Tabel 3.2 Spesifikasi perangkat keras ................................................................... 25

Tabel 3.3 Spesifikasi perangkat lunak .................................................................. 25

Tabel 5.1 Tabel Pengujian Aplikasi ...................................................................... 46

Tabel 5.2 Data Mahasiswa Berdasarkan Pilihan Terbanyak ................................. 47

Tabel 5.3 Hasil Normalisasi Data ......................................................................... 48

Tabel 5.4 Hasil Perhitungan Jarak ke Setiap *Medoids* Iterasi Pertama ................. 55

Tabel 5.5 Hasil Perhitungan Jarak ke Setiap Medoids Iterasi Kedua ................... 61

Tabel 5.6 Hasil Pengclusteran Mahasiswa Baru Dengan *K-Medoids* ................... 62

Tabel 5.7 Hasil *Silhouette Coefficient* dengan Nilai Klaster Berbeda-beda ......... 64

## BAB I

**PENDAHULUAN**

### 1.1 Latar Belakang

Dalam perkembangannya, lembaga perguruan tinggi atau universitas merupakan salah satu tujuan masyarakat dalam meningkatkan pendidikan yang fokus pada satu bidang konsentrasi yang dibutuhkan untuk memasuki dunia kerja. Meningkatnya keinginan masyarakat untuk memiliki karier yang baik dalam ketatnya dunia persaingan, mendorong meningkatnya jumlah universitas yang tersebar di seluruh Indonesia. Saat ini banyak universitas baik negeri maupun swasta yang menawarkan berbagai jurusan atau program studi. Oleh karena itu, suatu universitas harus mampu bersaing dengan universitas lainnya, selain untuk mendapatkan keuntungan yang maksimal, juga agar nama dari universitas tersebut tidak tertinggal.(Sri, Suharyati & Tati, 2013).

Menentukan perguruan tinggi dan jurusan yang tepat bukanlah persoalan yang mudah. Kurangnya informasi tentang fasilitas fakultas atau jurusan, lingkungan fakultas, jaminan kerja dari fakultas, biaya kuliah, dan promosi dapat menyebabkan calon mahasiswa salah dalam memilih jurusan sehingga mahasiswa tidak dapat mengikuti perkuliahan dengan baik karena tidak tertarik dibidang pilihannya, serta tidak dapat menyelesaikan perkuliahan dengan baik bahkan berujung pada *drop out (DO).*

Permasalahan diatas dapat diantisipasi oleh fakultas atau program studi dengan mengetahui faktor-faktor yang menentukan mahasiswa baru memilih jurusannya. Dengan mengetahui faktor-faktor tersebut, kedepannya fakultas dapat mengambil langkah-langkah antisipasi agar calon mahasiswa dapat mengambil jurusan yang sesuai.

Berdasarkan uraian permasalahan diatas, peneliti akan membuat sistem untuk menentukan faktor terbesar sumber informasi di Fakultas Teknik Universitas Halu Oleo, dengan mengelolah data tersebut menggunakan metode Data *Mining* dengan algoritma *K-Medoid* *clustering.* Dengan mengangkat judul tugas akhir “Penerapan

1

Algoritma K-Medoids dalam Penentuan Faktor Terbesar Sumber Informasi Pemilihan Jurusan di Fakultas Teknik Universitas Halu Oleo”.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah dalam penilitian ini adalah bagaimana menerapkan metode data *mining* dengan algoritma *K-Medoids* *Clustering* untuk menentukan faktor terbesar sumber informasi pemilihan jurusan di Fakultas Teknik Universitas Halu Oleo.

### 1.3 Batasan Masalah

Adapun hal-hal yang menjadi batasan masalah pada penelitian ini adalah sebagai berikut :

1. Penelitian ini dilakukan di Fakultas Teknik Universitas Halu Oleo pada angkatan 2018 sampai dengan angkatan 2019.
2. Algoritma yang digunakan adalah Algoritma *K-Medoids* *Clustering*.
3. Atribut atau parameter data yang digunakan berupa data kuantitatif di mana didapat dari hasil pengumpulan data kuisioner yang dibagikan kepada mahasiswa Fakultas Teknik Universitas Halu Oleo.

### 1.4 Tujuan Penelitian

Berdasarkan pada perumusan masalah yang telah dibahas, tujuan penyusunan tugas akhir ini adalah sebagai berikut:

1. Menerapkan metode algoritma *K-Medoids* *Clustering* ke dalam sistem untuk menentukan Faktor Terbesar Sumber Informasi Pemilihan Jurusan di Fakultas Teknik Universitas Halu Oleo.
2. Mengetahui hasil *Clustering*  dari kuisioner berdasarkan data mahasiswa.
3. Membantu pihak Prodi Fakultas Teknik Universitas Halu Oleo sebagai bahan untuk menentukan langkah selanjutnya dalam menerima mahasiswa baru.
4. Membantu calon mahasiswa khususnya Fakultas Teknik Universitas Halu Oleo untuk mendapatkan informasi agar memilih program studi sesuai minat.

### 1.5 Manfaat Penelitan

Adapun manfaat yang ingin dicapai dari tugas akhir ini adalah sebagai berikut:

1. Untuk menerapkan metode algoritma *K-Medoids* *Clustering* ke dalam sistem untuk menentukan Faktor Terbesar Sumber Informasi Pemilihan Jurusan di Fakultas Teknik Universitas Halu Oleo.
2. Untuk hasil *Clustering* dari kuisioner berdasarkan data mahasiswa.
3. Untuk membantu pihak Prodi Fakultas Teknik Universitas Halu Oleo sebagai bahan untuk menentukan langkah selanjutnya dalam menerima mahasiswa baru.
4. Untuk membantu calon mahasiswa khususnya Fakultas Teknik Universitas Halu Oleo untuk mendapatkan informasi agar memilih program studi sesuai minat.

### 1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir digunakan untuk menjelaskan penulisan perbab sebagai berikut :

## BAB I PENDAHULUAN

Merupakan bab pendahuluan yang menguraikan latar belakang masalah, rumusan nmasalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

## BAB II LANDASAN TEORI

Membahas mengenai dasar-dasar teori pendukung yang berhubungan dengan masalah yang diambil dan program aplikasi yang akan digunakan dalam pembangunan sistem.

## BAB III METODE PENELITIAN

Membahas mengenai waktu dan tempat penelitian, metode pengumpulan data, metode pembangunan sistem yang digunakan dalam penelitian tugas akhir ini.

## BAB IV ANALISIS DAN PERANCANGAN

Bab ini memberikan gambaran analisa dan perancangan sistem yang dibuat, dengan menggunakan desain UML (*Unified Modelling Language*).

## BAB V IMPLEMENTASI DAN PEMBAHASAN

Dalam bab ini akan dikaji mengenai implementasi hasil perancangan aplikasi yang dibuat serta melakukan pengujian terhadap sistem.

## BAB VI PENUTUP

Bab ini berisi kesimpulan dari program yang telah dibuat serta saran yang diperlukan untuk pengembangan program berikutnya.

### 1.7 Tinjauan Pustaka

Dalam penelitian yang dilakukan sebelumnya (Bagus, Alexius & Anggri, 2019), dengan judul Implementasi Metode *K-Medoids* *Clustering* Untuk Mengetahui Pola Pemilihan Program Studi Mahasiswa Baru Tahun 2018 Di Universitas Kanjuruhan Malang, penelitian ini menggunakan metode *partisional clustering* bertujuan untuk menemukan satu set *k-cluster* di antara data yang paling mencirikan objek dalam kumpulan suatu data. Hasil dari penelitian ini adalah pengelompokan mahasiswa baru yang berasal dari SMA/SMK dengan nilai ujian diatas 70 mengambil jurusan teknik informatika, sedangkan mahasiswa baru yang berasal dari SMK dengan nilai ujian dibawah 70 dan SMA dengan nilai ujian dibawah 50 mengambil jurusan sistem informasi, dan sisanya mengambil jurusan NON TI/SI.

Penelitian selanjutnya (Basilia Ria Irawati, 2008), dengan judul FaktorFaktor Yang Mempengaruhi Mahasiswa Dalam Memilih Pogram Studi Di Perguruan Tinggi, penelitian ini menggunakan metode *proporsional random sampling* bertujuan untuk mengetahui apakah minat, motivasi, status sosial ekonomi orang tua, pekerjaan yang diharapan dan lingkungan belajar mempengaruhi mahasiswa dalam memilih program studi di perguruan tinggi.

Penelitian lain yang dilakukan oleh (Sri, Suharyati & Tati, 2013), dengan judul Faktor-faktor Yang Berpengaruh Terhadap Keputusan Memilih Program Studi, penelitian ini menggunakan metode *Component Matrix* bertujuan untuk mengetahui faktor-faktor yang mempengaruhi siswa dalam memilih program studi yang berguna bagi universitas dan program studi untuk mengambil keputusan pengembangan institusi dalam menghadapi persaingan yang semakin ketat serta untuk menarik penelitian siswa baru dengan menggunakan analisis faktor.

Berdasarkan penelitian-penelitian yang telah dipaparkan di atas, penulis mecoba mengimplementasikan algoritma *K-Medoids* untuk penentuan faktor terbesar sumber informasi pemilihan jurusan. Data mahasiswa yang didapatkan dari kuisioner, dipilih atribut data sesuai dengan kebutuhan proses selanjutnya dilakukan kodefikasi dan disimpan dengan format *CSV*, kemudian di *cluster* menggunakan algoritma *K-Medoids.* Setelah pengelompokkan data mahasiswa, data tersebut akan divisualisaskian ke dalam bentuk sistem penentuan faktor terbesar sumber informasi pemilihan jurusan menggunakan bahasa pemrograman *Python*. Untuk data mahasiswa, penulis menggunakan data mahasiswa Fakultas Teknik Universitas Halu Oleo.

## BAB II

**LANDASAN TEORI**

### 2.1 Minat

Kata minat dapat diartikan sebagai ketertarikan seseorang akan sesuatu hal. Minat merupakan faktor psikologis yang dapat menentukan suatu pilihan yang tepat, selain itu minat merupakan salah satu faktor penentu yang sangat penting untuk suatu kemajuan dan keberhasilan seseorang. Minat seseorang dapat dilihat melalui kegiatan-kegiatan yang sering dilakukan yang didasari pernyataan senang dan tidak senang terhadap suatu obyek.

Widyastuti (2011) mendefinisikan minat adalah keinginan yang didorong oleh suatu keinginan setelah melihat, mengamati dan membandingkan serta mempertimbangkan dengan kebutuhan yang diinginkannya.

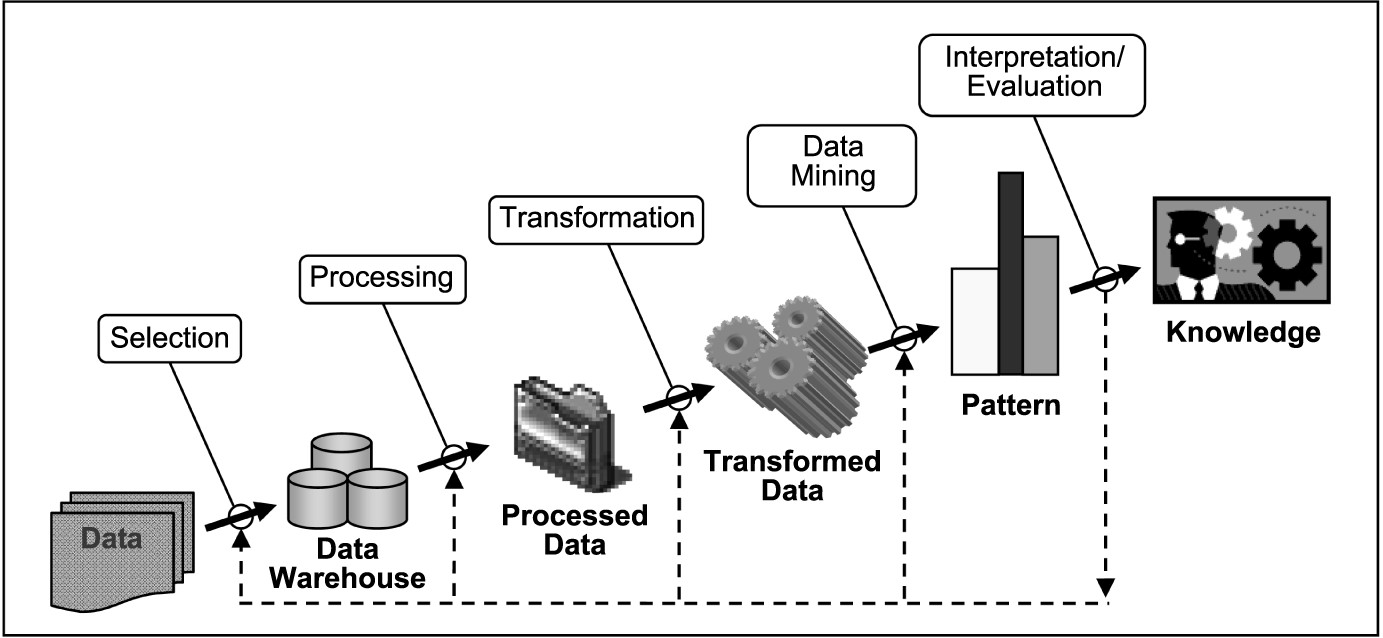
Berdasarkan pendapat diatas, maka minat memilih program studi atau jurusan pada mahasiswa, dapat diartikan sebagai kecendrungan yang mengarahkan mahasiswa untuk memilih program studi yang ditandai dengan adanya perasaan senang terhadap program studi tersebut, perasaan tertarik dan perasaan bahwa program studi tersebut bersangkut paut dengan kebutuhannya. Minat yang besar terhadap program studi yang telah mahasiswa pilih akan mempermudah terciptanya konsentrasi dalam studi. Jadi tanpa minat maka konsentrasi belajar juga akan sulit berkembang bahkan dipertahankan.

### 2.2 Data Mining

Data *Mining* adalah salah satu bidang yang berkembang pesat karena besarnya kebutuhan akan nilai tambah dari *database* dengan skala besar. Data *Mining* adalah serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data. Data *Mining* memiliki hubungan dari bidang ilmu seperti *artificial intelligent*, *machine learning*, statistik dan *database*. Beberapa teknik *data mining* antara lain: *clustering, classification, association rule mining, neural network, genetic algorithm* dan lain-lain (Witten, Frank, & Hall, 2011).

6

Adapun proses *data mining* dapat dibagi menjadi beberapa tahap yang diilustrasikan pada Gambar 2.1 :



## Gambar 2.1 Tahapan Data Mining(Witten et al., 2011)

### 2.3 Skala *Likert*

Skala likert adalah skala pengukuran yang dikembangkan oleh Likert (1932). Menurut Preston dan Colman, 2000 adalah skala likert mempunyai empat atau lebih butir-butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor atau nilai yang merepresentasikan sifat individu, misalkan pengetahuan, sikap, dan perilaku. Dalam proses analisis data, komposit skor, biasanya jumlah atau rataan, dari semua butir pertanyaan dapat digunakan. Jumlah titik 1 dan 2 menghasilkan indeks reliabilitas, validitas, dan kekuatan diskriminasi yang jelek. Hasil ini berbeda nyata dibandingkan jumlah titik 3, 4, dan 5 yang mempunyai indeks reliabilitas, validitas, dan kekua-tan diskriminasi lebih baik. Ukuran stabilitas (test-retest validitas) menun-jukkan jumlah titik 1 dan 2 mempu-nyai stabilitas yang paling buruk dan semakin bagus saat jumlah titik mening-kat menjadi 3, 4 atau 5.

### 2.4 *K-Medoids atau Partitioning Around Medoids (PAM)*

Algoritma Partitioning Around Medoids (PAM) atau dikenal juga dengan *K-*

*Medoids* adalah algoritma pengelompokan yang berkaitan dengan algoritma *KMeans* dan algoritma *medoidshift.* Algoritma *K-Medoids* ini diusulkan pada tahun

1987. Algoritma *K-Medoid* dikembangkan oleh Leonard Kaufman dan Peter J.

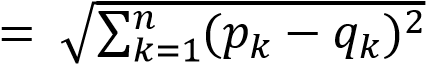
Rousseeuw. Algoritma ini sangat mirip dengan algoritma *K- Means*, terutama karena kedua algoritma ini partitional. Dengan kata lain, kedua algoritma ini memecah *dataset* menjadi kelompok-kelompok dan kedua algoritma ini berusaha untuk meminimalkan kesalahan. Tetapi algoritma *K-Medoids* bekerja dengan menggunakan *medoids*, yang merupakan entitas dari *dataset* yang mewakili kelompok dimana ia dimasukkan.*Medoids* adalah objek representatif dari kumpulan data atau *cluster* dengan kumpulan data yang ketidaksamaan rata-rata untuk semua objek dalam cluster minimal.

Algoritma *K-Medoids* menggunakan metode partisi *clustering* untuk mengelompokkan sekumpulan *n* (jumlah )obyek menjadi sejumlah *cluster*. Algoritma ini menggunakan obyek pada kumpulan obyek untuk mewakili sebuah cluster. Obyek yang terpilih untuk mewakili sebuah *cluster* disebut dengan *medoid*. *Cluster* dibangun dengan menghitung kedekatan yang dimiliki antara medoid dengan obyek *non-medoid*.

Algoritma dari *K-Medoids* adalah sebagai berikut(Mustakim, 2012):

1. Inisialisasi pusat *cluster* sebanyak *k* (jumlah *cluster*)
2. Alokasikan setiap data (objek) ke *cluster* terdekat menggunakan persamaan ukuran jarak *Euclidian Distance.*
3. Pilih secara acak pilih objekpada masing-masing *cluster* sebagai kandidat *medoids* baru
4. Hitung jarak setiap objek yang berada pada masing-masing *cluster* dengan kandidat *medoids* baru.
5. Hitung total biaya simpangan (*S)*, dengan menghitung nilai total distance baru – total *distance* lama.Jika *S <0*, maka tukar objek dengan data *cluster* untuk membentuk sekumpulan *k* obyek baru sebagai *medoid*.
6. Ulangi langkah 3 sampai 5 hingga tidak terjadi perubahan medoid, sehingga didapatkan *cluster* beserta anggota *cluster* masing-masing.

Dalam mengukur jarak dalam clustering dapat dilakukan dengan menggunakan *Euclidean Distance*. *Euclidean distance* merupakan pengukuran jarak obyek dan pusat cluster yang banyak digunakan secara luas dalam berbagai kasus pattern matching, termasuk *clustering*. *Eucludean distance* dinyatakan dengan persamaan :

𝑑𝑖𝑠𝑡 ……………………………….…………. (2.1)

Dimana :

*n* = jumlah fitur dalam suatu data.

*k* = indeks data. *pk* = nilai atribut (fitur) ke-*k* dari *p*.

*qk* = nilai atribut (fitur) ke-*k* dari *q*.

Nilai *S* dinyatakan dengan persamaan:

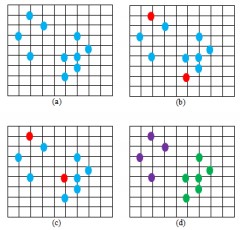
*S = Total* 𝑐𝑜𝑠𝑡 *baru − Total* 𝑐𝑜𝑠𝑡 *lama* ………………….....………. (2.2)

Dimana :

Total *cost* baru = jumlah biaya/*cost non-medoids*.

Total *cost* baru = jumlah biaya/*cost medoids*.

*K-medoids* sangat mirip dengan *K-means*, perbedaan utama diantara dua algoritma tersebut adalah jika pada *K-Means* cluster diwakili dengan pusat dari c*luster*, sedangkan pada *K-Medoids cluster* diwakili oleh obyek terdekat dari pusat *cluster*.



## Gambar 2.2 Ilustrasi *K-Medoids*

### 2.5 *Clustering*

*Clustering* atau klasterisasi adalah metode pengelompokan data. Menurut Tan, 2006 clustering adalah sebuah proses untuk mengelompokan data ke dalam beberapa cluster atau kelompok sehingga data dalam satu *cluster* memiliki tingkat kemiripan yang maksimum dan data antar *cluster* memiliki kemiripan yang minimum. *Clustering* merupakan proses partisi satu set objek data ke dalam himpunan bagian yang disebut dengan cluster. Objek yang di dalam cluster memiliki kemiripan karakteristik antar satu sama lainnya dan berbeda dengan cluster yang lain. Partisi tidak dilakukan secara manual melainkan dengan suatu algoritma *clustering*. Oleh karena itu, clustering sangat berguna dan bisa menemukan group atau kelompokyang tidak dikenal dalam data. *Clustering* banyak digunakan dalam berbagai aplikasi seperti misalnya pada *business inteligence,* pengenalan pola citra, *web search*, bidang ilmu biologi, dan untuk keamanan (*security*). Di dalam *business inteligence, clustering* bisa mengatur banyak customer ke dalam banyaknya kelompok. Contohnya mengelompokan pelanggan ke dalam beberapa *cluster* dengan kesamaan karakteristik yang kuat. *Clustering* juga dikenal sebagai data segmentasi karena *clustering* mempartisi banyak data set ke dalam banyak group berdasarkan kesamaannya. Selain itu *clustering* juga bisa sebagai *outlier detection*(Binus, 2017).

### 2.6 *Silhouette Coefficient*

*Silhouette Coefficient (SC)* merupakan metode evaluasi yang digunakan untuk melihat kualitas dan kekuatan cluster dan kualitas suatu objek dalam suatu *cluster* (Handoyo, Mangkudjaja, & Nasution,2014). Untuk menghitung *silhouette coefficient* terdapaat beberapa tahapan yaitu:

1. Menghitung rata-rata jarak setiap dokumen (disimbolkan dalam *i*) dengan setiap dokumen lain dalam *cluster* yang sama. Simpan nilai rata-rata pada suatu variable misalkan ai.
2. Menghitung rata-rata nilai jarak setiap dokumen suatu *cluster* (disimbolkan dalam *i*)dengan setiap dokumen pada *cluster* lain dan diambil nilai terkecil.

Simpan nilai tersebut pada suatu variable misalkan bi.

1. Hitung nilai koefisien silhouette *coefficient* dengan rumus berikut:

𝑏𝑖−𝑎𝑖

𝑆𝐶 = ...................……………………………….…………. (2.3) max (𝑎𝑖,𝑏𝑖)

### 2.7 *Principal Component Analysis* (PCA)

Menurut (Ifadah; 2011) metode PCA merupakan metode yang digunakan untuk menyederhanakan suatu variabel yang akan diteliti dengan mereduksi dimensinya. Reduksi dimensi dilakukan dengan cara menghilangkan korelasi yang dimiliki antar variabel bebas dengan transformasi varabel asal menjadi variabel baru yang tidak berkorelasi sama sekali.

Menurut (Margono; 2008) dalam Principal Component Analysis (PCA) memiliki beberapa kritera sebagai berikut:

1. Communalities sebagai faktor bersama. Communalities merupakan peranan faktor untuk satu kesatuan yang berasal dari masing-masing variabel atau subvariabel yang menyusun satu faktor secara bersama. Hasil yang ditunjukkan dalam communalities merupakan nilai nitial dan extraction. Nilai initial merupakan besarnya peranan variabel secara individu untuk membentuk faktor baru bersama. Dan nilai extraction merupakan besarnya peranan variabel yang berupa persentase masing-masing variabel untuk membentuk faktor baru.
2. Nilai akar karakteristik (eigen values) memiliki persamaan karakteristik |MvvλI|=0, dengan λ>1. Scree plot merupakan diagram yang menggambarkan kencenderungan penurunan eigen value. Sumbu Y pada diagram menunjukkan eigen value dan sumbu X menunjukkan jumlah faktor. Dalam menentukan banyaknya faktor yang terbentuk dapat dilihat pada grafik dengan kemiringan yang paling tajam yang terbentuk dari faktor satu ke faktor lainnya.

### 2.8 *Python*

*Python* adalah satu dari bahasa pemrograman tingkat tinggi yang bersifat interpreter, interaktif, *object-oriented* dan dapat beroperasi di hampir semua *platform*, seperti keluarga Linux, Windows, Mac, dan platform lainnya. *Python* adalah salah satu bahasa pemrograman tingkat tinggi yang mudah dipelajari karena sintaks yang jelas dan elegan, yang dikombinasikan dengan penggunaan modul-modul yang mempunyai struktur data tingkat tinggi, efisien, dan siap langsung digunakan (Rosmala & Gadya, 2012). *Source code* aplikasi dalam bahasa pemrograman *Python* biasanya akan dikompilasi menjadi format perantara yang dikenal sebagai *bytecod*e yang selanjutnya akan dieksekusi. Kelemahan dalam bahasa pemrograman ini terletak pada kecepatan eksekusi yang tidak secepat bahasa pemrograman yang dikompilasi dan bersifat lebih *low-level* seperti C dan C++.

### 2.9 *Django Web Framework*

Django adalah sebuah web *framework* yang menggunakan bahasa pemrograman *Python* yang mendukung pembuatan *website* dengan *konsep rapid development* (Django Software Foundation, 2017). Wiki-Python menyatakan bahwa *framework* yang sekarang telah berada di versi 2.1 ini sebagai *web framework* paling popular dikalangan *high-level frameworks*. Karena menggunakan bahasa pemrograman Python yang telah memiliki banyak *library* siap pakai, mengembangkan sistem dengan Django berarti dapat menggunakan *library* tersebut sesuai dengan keperluan pengembangan.

Django bertujuan untuk memudahkan pengembangan situs web dan basis data yang kompleks (Django Software Foundation, 2017). Salah satu keunggulan django adalah, *framework* ini telah merepresentasikan ORM (Object Relational Mapper) sehingga tidak perlu menyesuaikan *query* jika terjadi perubahan database yang digunakan. Berikut ini adalah beberapa keuntungan menggunakan Django: a. *Object-Relational Mapping (ORM) Support.*

1. *Automatic Admin Interface.*
2. *Elegant URL Design.*
3. *Template System.*
4. *Cache system.*
5. *Internationalization.*
6. *A light weight web server for development and test.*

### 2.10 *SQLite*

SQLite merupakan sebuah system manajemen basis data relasional yang bersifat ACID - compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C (Dimas R.S.N, 2011). Sqlite di embbeded database dapat digunakan dalam pembuatan database di aplikasi secara internal.

Kelebihankelebihan SQLite antara lain (Infolinux, 2005) : a. Secara umum sangat stabil

1. Lisensi *Public Domain.*
2. Mampu menampung data sampai 2 tera.
3. Banyak bahasa program yang mendukung.
4. Tabell dapat disimpan pada file terpisah, dan dapat di *attach* ke database utama.

### 2.11 Database (Basis Data)

Basis data atau *Database* merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya yang diaorganisasikan sesuai struktur tertentu dan disimpan dengan baik. Untuk mendapatkan informasi yang berguna dari kumpulan data maka diperlukan suatu perangkat lunak (*software*) untuk memanipulasi data sehingga mendapatkan informasi yang berguna.(Sukamto dan Shalahuddin, 2013). *Database* terbentuk dari beberapa komponen, yaitu (Bunafit Nugroho, 2008):

1. *Table*

*Table* atau Tabel adalah sekumpulan data dengan struktur yang sedemikian rupa, terbentuk dari *record* dan *field.* Istilah Tabel disini berbeda dengan istilah Tabel pada HTML, walaupun secara visual hampir sama.

1. *Record*

*Record* adalah sekumpulan *field* yang membentuk suatu objek tertentu.

1. *Field*

*Field* adalah atribut dari objek yang memiliki tipe data tertentu.

### 2.12 *Unified Modeling Language (UML)*

*Unified Modeling Language* (UML) merupakan bahasa dalam mendesain perangkat lunak secara visual. Dengan UML, desainer dapat melihat konsep global suatu desain. Desain kemudian dapat dijadikan panduan dalam proses pengembangan dan rekayasa perangkat lunak. Selain itu, UML dapat menjadi media komunikasi gagasan antara pengembang perangkat lunak dengan pengguna (Shalahudin, 2011). Ada beberapa jenis diagram dalam UML yaitu:

### 2.13 *Use Case Diagram*

*Use case diagram* menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat *user*) dan bukan bagaimana fungsi yang ada di dalam sistem.

(Shalahudin, 2011).

## Tabel 2.2 Simbol-simbol *Use Case Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1. |  | *Actor* | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case.* |
| 2. |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*dependent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| 3. |  | *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (*ancestor*). |
| 4. |  | *Include* | Menspesifikasikan bahwa *use case* sumber secara eksplisit. |
| 5. |  | *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| 6. |  | *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7. |  | *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8. |  | *Use case* | Deskripsi dari uraian aksi-aksi yang ditampilkan system yang menghasilkan suatu hasil yang terukur bagi suatu *actor.* |
| 9. |  | *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |

### 2.14 *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi (Sukamto & Shalahuddin, 2013).

## Tabel 2.3 Simbol *Activity* Diagram

|  |  |  |
| --- | --- | --- |
| No. | Simbol | Keterangan |
| 1. | Start | Kondisi awal menunjukkan  awal dari suatu diagram aktivitas |
| 2. | End | Kondisi akhir menunjukkan  akhir dari suatu diagram aktivitas |
| 3. |  | Kondisi transisi menunjukkan  kondisi transisi antar aktivitas |
| 4. |  | Swimlane menunjukkan  aktor dari diagram aktivitas yang dibuat |
| 5. |  | Aktivitas menunjukkan  aktivitas-aktivitas yang terdapat pada diagram aktivitas |
| 6. |  | Pengecekan kondisi  menunjukkan pengecekan terhadap suatu kondisi |

### 2.15 Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksiantara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi system (Sukamto & Shalahuddin, 2013)

## Tabel 2.4 Simbol *Sequence* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SIMBOL** | **NAMA SIMBOL** | **KETERANGAN** |
| 1. |  | *LifeLine* | Objek *entity*, antarmuka yang saling berinteraksi. |
| 2. |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |
| 3. |  | *Message* | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi. |

|  |  |  |  |
| --- | --- | --- | --- |
| 4. |  | *Message* | Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*. |

### 2.16 Class Diagram

*Class diagram* adalah kumpulan objek-objek yang mempunyai struktur umum, *behavior* umum, relasi umum, dan *semantic* atau kata yang umum. *Classclass* ditentukan atau ditemukan dengan cara memeriksa objek-objek dalam *sequence diagram* dan *collaboration diagram*. Sebuah *class* digambarkan seperti sebuah bujur sangkar dengan tiga bagian ruangan.

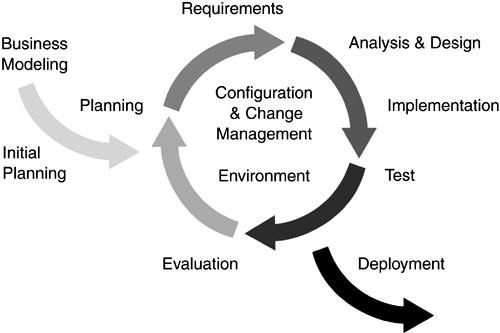
## Tabel 2.5 Simbol *Class* Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **GAMBAR** | **NAMA** | **KETERANGAN** |
| 1 |  | *Generalization* | Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| 2 |  | *Class* | Himpunan dari objek yang berbagi atribut serta operasi yang sama. |
| 3 |  | *Collaboration* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor |
| 4 |  | *Realization* | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 5 |  | *Dependency* | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)*. |

### 2.17 Metode Pengembangan Sistem

Model Pengembangan Sistem yang digunakan pada penelitian ini adalah RUP *(Rational Unified Process)* yang merupakan pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang *(iterative)*, fokus pada arsitektur *(architecturecentric)*, lebih diarahkan berdasarkan penggunaan kasus *(use case driven).* RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik *(well defined)* dan penstrukturan yang baik *(well structured).* Proses pengulangan/iterative pada RUP secara global dapat dilihat pada Gambar 2.4

(Rossa, AS dan M. Shalahuddi.2013)



## Gambar 2.4 Proses Iteratif RUP(Rossa dan Shalahuddin.2013)

RUP memiliki empat buah tahap atau fase yang dapat dilakukan pula secara iteratif.Berikut adalah penjelasan untuk setiap fase pada RUP.

1. *Inception*

Tahap ini yaitu tahap penentuan ruang lingkup pengembangan sistem dari hasil wawancara dan observasi yang dilakukan, meliputi hasil penelitian atau skripsi-skripsi terdahulu.

1. *Elaboration*

Tahap ini yaitu tahap melakukan identifikasi masalah pada sistem yang akan dibuat berdasarkan hasil observasi dan wawancara yang telah dilakukan.

Didalam *elaboration* terdapat dua tahapan yaitu: a. Analisis

Terdapat tiga fase dalam tahapan analisis sistem pada alur pengembangan sistem RUP, yaitu: analisis permasalahan, analisis persyaratan, dan analisis keputusan.

b. Perancangan

Tahap ini terdiri dari: perancangan aplikasi denganmenggunakan diagram UML meliputi *use case diagram*, *activity diagram* dan *sequence diagram,* dan perancangan tampilan.

1. *Construction*

Tahap ini menjelaskan bagaimana mengimplementasi dan melakukan uji coba terhadap aplikasi yang telah dibuat. Dalam tahapan implementasi dijelaskan perangkat keras dan perangkat lunak apa saja yang dibutuhkan untuk pengimplementasian sistem. Sedangkan pada tahapan uji coba dilakukan *testing*. Testing diperlukan untuk menjamin kualitas aplikasi yang telah dibuat apakah telah sesuai dengan yang diharapkan.

1. *Transition*

Pada tahap *transition* yaitu pembuatan panduan penggunaan dari sistem yang telah dibuat.

## BAB III

**METODOLOGI PENELITIAN**

### 3.1 Waktu Penelitian

Waktu pelaksanaan penelitian tugas akhir dilaksanakan mulai dari bulan Januari 2020 sampai dengan Juni 2020. Rincian kegiatan dapat dilihat pada tabel berikut :

## Tabel 3.1*Gantt Chart* Waktu Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Uraian |  | | | | |  | | Waktu (2020) | | | | | | | | | | | | | | |
| Juni | | | |  | Juli | | Agustus | | | | Oktober | | | | Novemb er | | | | Dese mber | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| 1. | *Incepti on* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2. | *Elabor ation* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3. | C*onstr uction* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4. | *Transiti on* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### 3.2 Tempat Penelitian

Adapun tempat pelaksanaan penelitian tugas akhir yang akan dilakukan di Fakultas Teknik Universitas Halu Oleo.

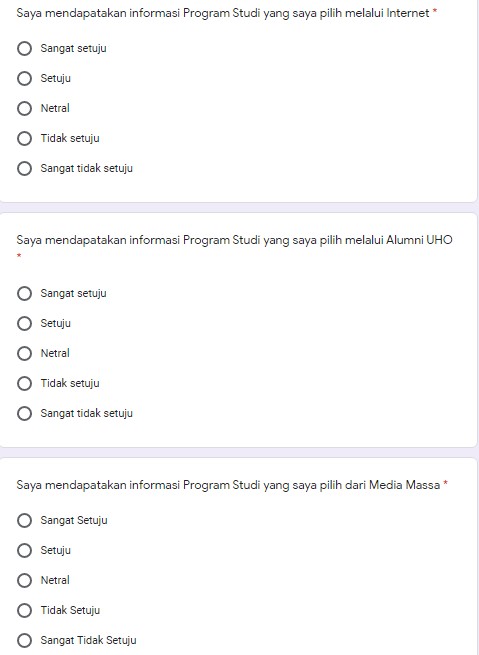
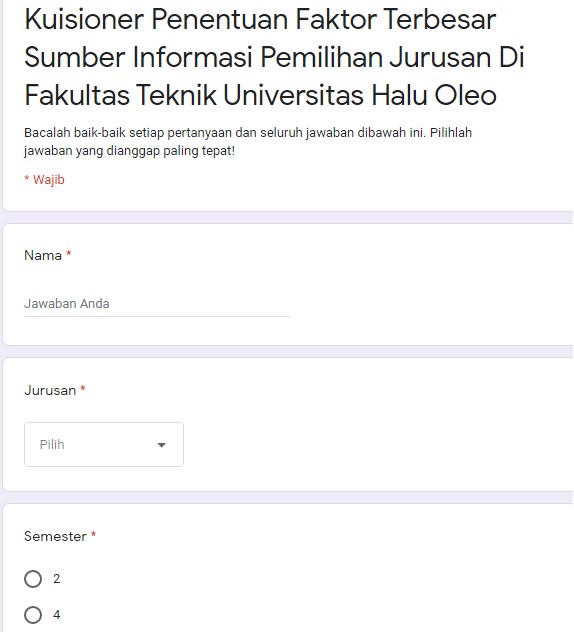
20

### 3.3 Metode Pengumpulan Data

Secara umum, system yang akan dibangun dalam penelitian ini adalah sebuah system dengan fungsi utama untuk melakukan pengolompokkan data mahasiswa Fakutas Teknik Universitas Halu Oleo. Adapun pengelompokkan data tersebut digunakan langkah – langkah sebagai berikut:

1. Sumber Data

1. Studi pustaka dilakukan dengan tujuan untuk mengetahui metode apa yang akan digunakan untuk menyelesaikan permasalahan yang akan diteliti, serta mendapatkan dasar-dasar referensi yang kuat dalam menerapkan suatu metode yang akan digunakan dalam penelitian, yaitu dengan mempelajari buku -buku*,*artikel-artikel dan jurnal-jurnal yang berhubungan dengan permasalahan yang akan dibahas.
2. Kuisioner suatu metode pengumpulan data yang diperoleh dengan membagikan link google formulir yang berisi instrumen pertanyaanpertanyaan berupa kuesioner kepada mahasiswa Fakutas Teknik Universitas Halu Oleo angkatan 2018 sampai dengan 2019. Adapun instrumen kuesioner ini terdiri dari 10 item pertanyaan, dapat dilihat pada gambar 3.1 dan 3.2 :

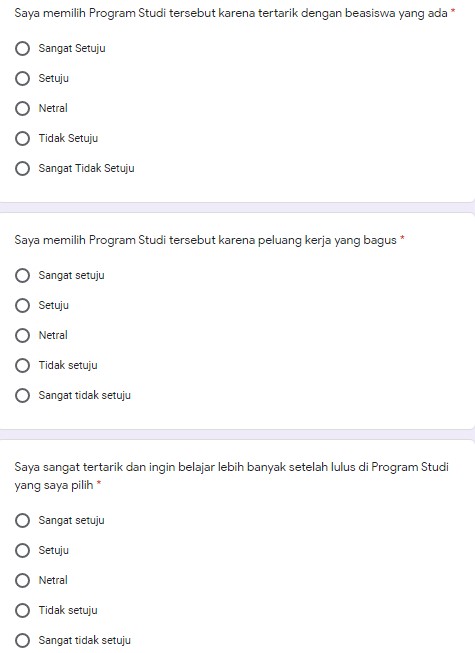


## Gambar 3.1 Pertanyaan Kuisioner

**Gambar 3**

**.2**

**Pertanyaan Kuisioner**



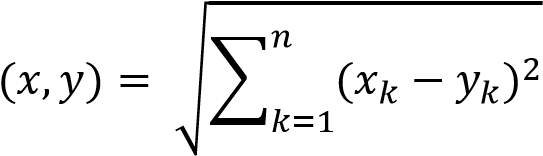
1. Normalisasi Data

Pada tahap ini dilakukan proses normalisasi data menggunakan pembobotan skala *likert* tujuannya adalah untuk menyemakan skala atribut data dalam range yang lebih spesifik, sehingga data dapat diolah dengan mudah menggunakan metode K-Medoids Clustering.

1. Pengolahan Data

Setelah proses transformasi langkah selanjutnya adalah proses pengolahan data menggunakan metode K-Medoids Clustering. Tahapan proses metode KMedoids Clustering adalah sebagai berikut :

1. Ditentukan *k* (jumlah *cluster*) dari *n* objek adalah 3
2. Menentukan cluster awal sebagai medoid
3. Menghitung jarak obyek ke masing-masing medoid yang telah dipilih. Untuk menentukan jarak obyek ke masing-masing medoid digunakan rumus Euclidian *Distance* :

𝑑

1. Hitung total simpangan (*S*) dengan menghitung total *distance* baru – total *distance* lama. Jika *S* < 0 maka tukar objek dengan data *cluster* untuk membentuk sekumpulan *k* objek baru sebagai medoid (Setyawati, 2017).

*S* = total *distance* baru – total *distance* lama

1. Karena nilai *S < 0* maka proses pengklasteran dihentikan. Sehingga objek non *medoids* dijadikan sebagai *medoid* baru.

### 3.4 Metode Pengembangan Sistem

Dalam merancang atau membuat perangkat lunak dibutuhkan suatu pemodelan dari keseluruhan proses-proses yang akan dilakukan selama pembuatan perangkat lunak tersebut. Model yang digunakan dalam perangkat lunak ini adalah proses *Rational Unified Proses* (RUP). Adapun proses dari metode/model RUP ini meliputi:

#### 3.4.1 *Inception* / Permulaan

Pada tahap ini, penulis memodelkan ruang lingkup sistem yang akan dibuat. Adapun ruang lingkup dari sistem ini adalah sebagai berikut:

1. Data yang akan diproses adalah data dari kuisioner, dipilih atribut data sesuai dengan kebutuhan proses selanjutnya dilakukan kodefikasi dan disimpan dalam file *csv*.
2. Dalam pembuatanya menggunakan bahasa pemrograman *Python* dan pengelolaan basis datanya menggunakan SQLite.
3. Data mahasiswa yang telah dihitung jarak antara objek data dengan algoritma *K-Medoids* baik di dalam maupun di luar cluster menggunakan teknik Euclidean *Distance,* serta uji mutu cluster akan divisualisaskian ke dalam bentuk sistem penentuan faktor terbesar sumber informasi pemilihan jurusan.

#### 3.4.2 *Elaboration* / Elaborasi

Pada tahap ini, setelah menentukan ruang lingkup sistem, selanjutnya penulis melakukan langkah-langkah berikut :

1. Melakukan analisis masalah menggambarkan alur sistem yang akan dibuat/diusulkan.
2. Membuat perancangan aplikasi menggunakan UML (*Unified Modelling Language).*
3. Membuat perancangan tampilan (*interface*) untuk setiap tampilan form / halaman yang ada dalam sistem yang akan dibuat.

#### 3.4.3 *Construction* / konstruksi

Pada tahap ini proses yang dilakukan yaitu membangun sistem mulai dari tampilan user *interface* sampai penerapan *coding* pada sistem.

#### 3.4.4 *Transition* / Transisi

Pada tahap ini difokuskan untuk melakukan proses pengujian terhadap aplikasi. Pengujian yang digunakan yaitu metode pengujian *black box.*

## BAB IV

**ANALISIS DAN PERANCANGAN SISTEM**

### 4.1 Analisis Sistem

Analisis sistem merupakan suatu tahapan yang bertujuan untuk mengetahui dan mengamati apa saja yang terlibat dalam suatu sistem. Pembahasan yang ada pada analisis sistem ini yaitu analisis masalah, analisis kebutuhan fungsional dan analisis kebutuhan nonfungsional.

#### 4.1.1 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah segala bentuk data yang dibutuhkan oleh sistem agar sistem dapat berjalan sesuai dengan prosedur yang dibangun. Setelah melalui tahapan analisis, maka telah ditetapkan kebutuhan-kebutuhan untuk membangun sistem meliputi *input*, proses dan *output*.

**4.4.4.1 Analisis Kebutuhan *Input***

*Input* dari sistem yang di bangun adalah data kusioner mahasiswa.

##### 4.4.4.2 Analisis Kebutuhan Proses

Kebutuhan proses dalam aplikasi ini antara lain:

1. Proses pembuatan sistem menggunakan *Django Framework* dengan bahasa pemograman *Python.*
2. Proses *clustering* akan menggunakan algoritma *K-Medoids.*

##### 4.4.4.3 Analisis Kebutuhan *Output*

*Output* yang diperoleh dalam sistem ini merupakan sebuah sistem berbasis *web* yang akan menampilkan faktor terbesar sumber informasi yang mempengaruhi mahasiswa memilih jurusan di Fakultas Teknik Universitas Halu Oleo dengan menggunakan algoritma *K-Medoids Clustering.*

#### 4.1.2 Analisis Kebutuhan Nonfungsional

Analisis kebutuhan nonfungsional adalah sebuah langkah dimana seorang pembangun sistem menganalisis sumber daya yang dibutuhkan untuk membangun sistem yang akan dibangun. Analisis kebutuhan nonfungsional yang dilakukan

25

dibagi dalam dua tahap, yaitu analisis kebutuhan perangkat keras dan analisis kebutuhan perangkat lunak.

##### 4.1.2.1 Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan pada pembangunan sistem ini, sebagai berikut:

## Tabel 3.2 Spesifikasi perangkat keras

|  |  |  |
| --- | --- | --- |
| No | Nama Perangkat | Spesifikasi |
| 1. | *Processor* | 2,5 *GHz Intel Core i*5 |
| 2. | *Monitor* | *Monitor 13 inch* |
| 3. | *Memori* | 4*GB* 1333 *MHz DDR*3 |
| 4. | *Harddisk* | *1TB HDD* |

**4.1.2.2 Analisis Kebutuhan Perangkat Lunak**

Perangkat lunak yang digunakan pada pembangunan ini, sebagai berikut:

## Tabel 3.3 Spesifikasi perangkat lunak

|  |  |  |
| --- | --- | --- |
| No | Nama Perangkat | Spesifikasi |
| 1. | *Operating System* | *Windows 10 64 bit* |
| 2. | *SQLite* | 3.8.2 |
| 3. | *Browser* | *Google Chrome* |
| 4. | *Text Editor* | *Django* |

### 4.2 Analisis Sistem yang Direncanakan

Sistem Implementasi algoritma *K-Medoids* menggunakan metode *clustering* untuk menentukan pemilihan jurusan merupakan sebuah sistem yang dapat membantu menentukan sumber informasi yang paling banyak digunakan calon mahasiswa Fakultas Teknik Universitas Halu Oleo dalam memilih jurusan. Dengan menggunakan data sebanyak 300 akan diterapkan proses clustering. Perhitungan jarak antara objek data dengan *medoids* baik di dalam maupun di luar cluster menggunakan teknik *Euclidean Distance.* Kemudian hasil dari sistem berbasis web ini menggambarkan sumber informasi terbesar bagi calon mahasiswa dalam bentuk *cluster*.

### 4.3 Analisis Perancangan Sistem

Perancangan sistem ini berbasis web dibangun dengan menggunakan

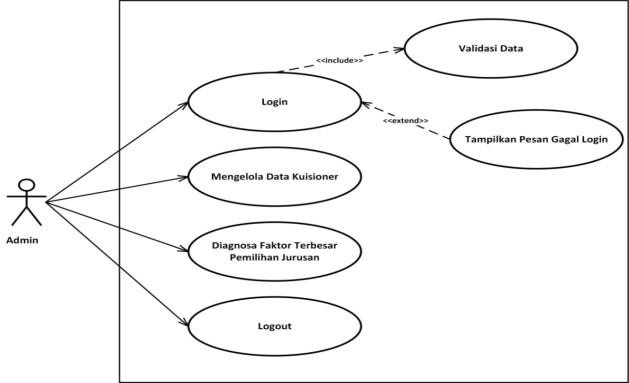
*Unified Modeling Language* (UML) yang terdiri dari *Use Case Diagram, Activty Diagram,* dan *Sequence Diagram, Class Diagram,* serta perancangan *user interface.*

#### 4.3.1 Perancangan Unified Modeling Language (UML)

Aplikasi dibangun dengan menggunakan *Unified Modeling Language* (UML). *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram yang terdiri dari *Use Case Diagram, Activity Diagram, Class Diagram*dan *Sequence Diagram.* **a. *Use Case Diagram***

*Use Case Diagram* adalah sebuah diagram yang dapat merepresentasikan interaksi yang terjadi antara user dengan sistem. Dalam *Use Case Diagram*, aktor yang dapat menggunakan sistem yaitu Administrator. Aktor Administrator yang setelah login akan masuk pada Halaman Utama. Setelah itu, Administrator bertugas untuk mengimput, *update* dan *delete* data dari kuisioner mahasiswa.

Proses yang terjadi pada *use case diagram* dijelaskan secara rinci pada gambar 4.1.

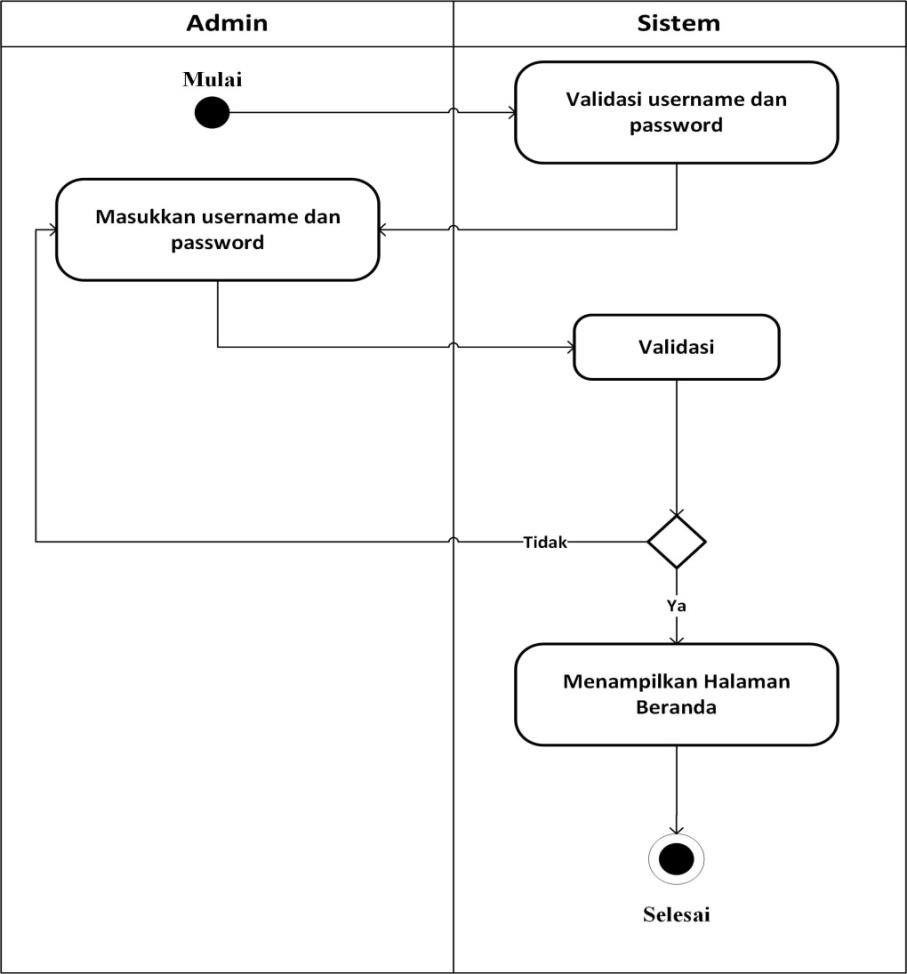


**Gambar 4.1 *Use Case Diagram***

### b. *Activity Diagram*

#### 1. *Activity Diagram Login*

Pada Gambar 4.2 menunjukkan *activity diagram login* untuk admin. Proses *login* dimulai dengan mengisi *username* dan *password* kemudian sistem melakukan pengecekan *username* dan *password*, apabila sesuai maka admin dapat menjalankan sistem, namun apabila data yang dimasukkan tidak *valid* maka sistem akan menampilkan halaman *login.*

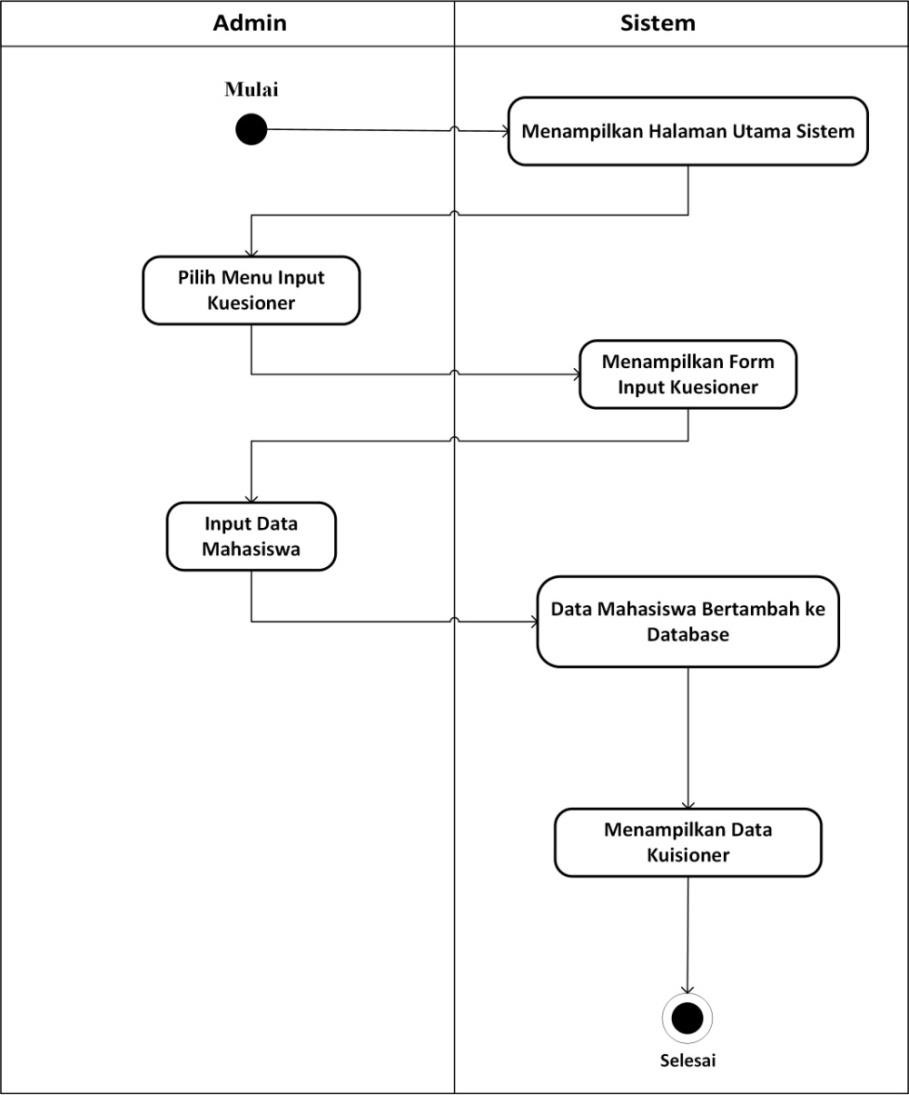


**Gambar 4.2 *Activity Diagram Login* Admin**

#### 2. *Activity Diagram* Mengelola Data Input Kuisioner

*Activity Diagram* Mengelola Data Kuisioner meta menjelaskan bagaimana user melakukan pengolahan data terhadap data-data mahasiswa. Admin yang sudah berhasil masuk ke dalam sistem setelah *login* dengan benar, memilih menu Mengelola Data Kuisioner. Sistem akan menampilkan halaman Mengelola Data Kuisioner dan data-data mahasiswa yang sudah tersimpan. Untuk menambah data mahasiswa maka pilih menu *Input* kemudian akan tampil form *Input* Data Mahasiswa, dan isi data siswa yang sesuai dengan data-data yang telah diberikan. Jika pengisian sudah lengkap, dilakukan penyimpanan terhadap data-data tersebut.Admin juga dapat mengubah serta menghapus data siswa yang sudah tersimpan jika terjadi kesalahan dalam memasukkan data-data sebelumnya. Jika telah selesai, data disimpan kembali.

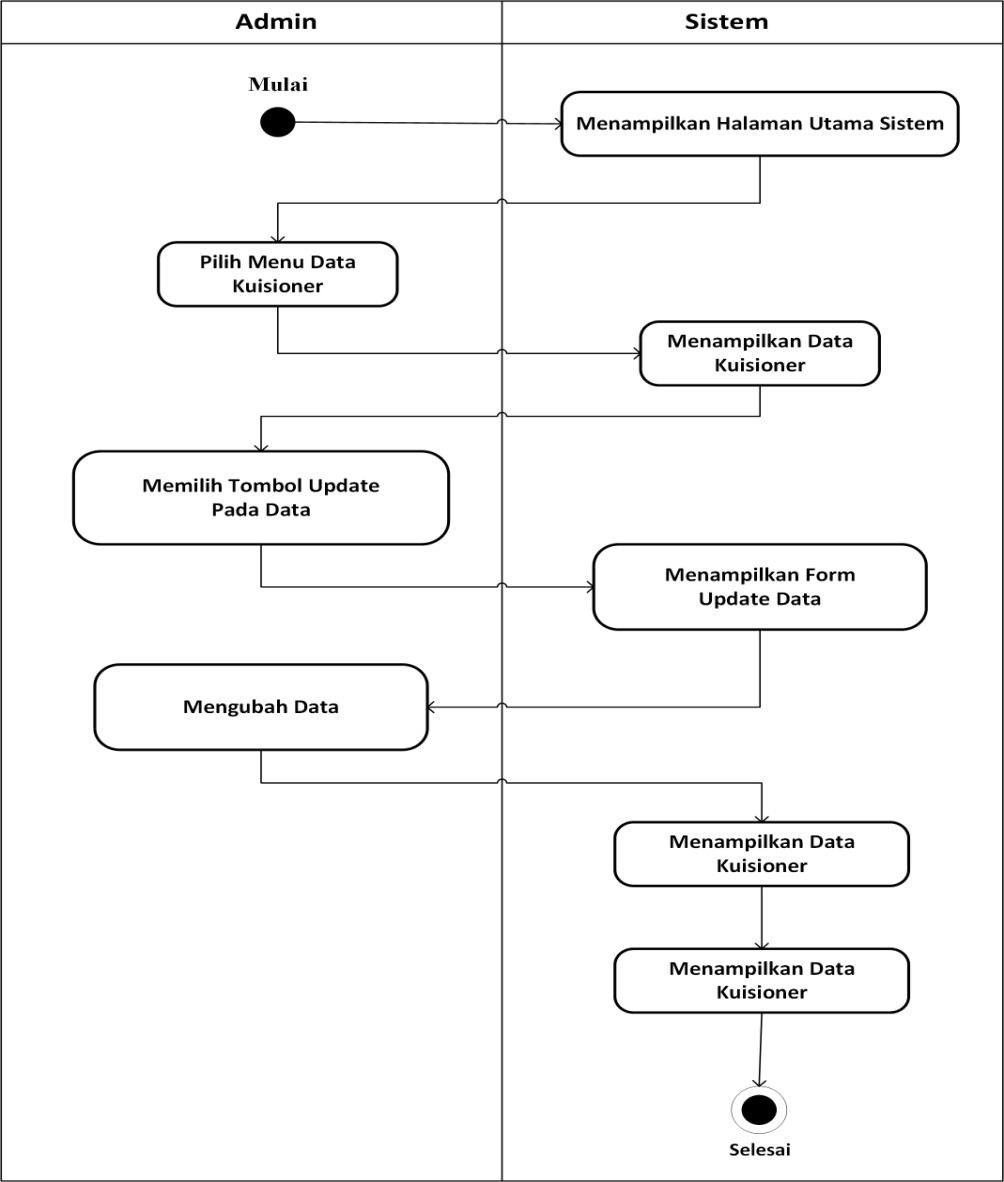
Untuk lebih jelasnya, dapat dilihat pada Gambar 4.3.



## Gambar 4.3 *Activity Diagram* Mengelola Data Input Kuisioner

### 3. *Activity Diagram* Mengelola Data Update Kuisioner

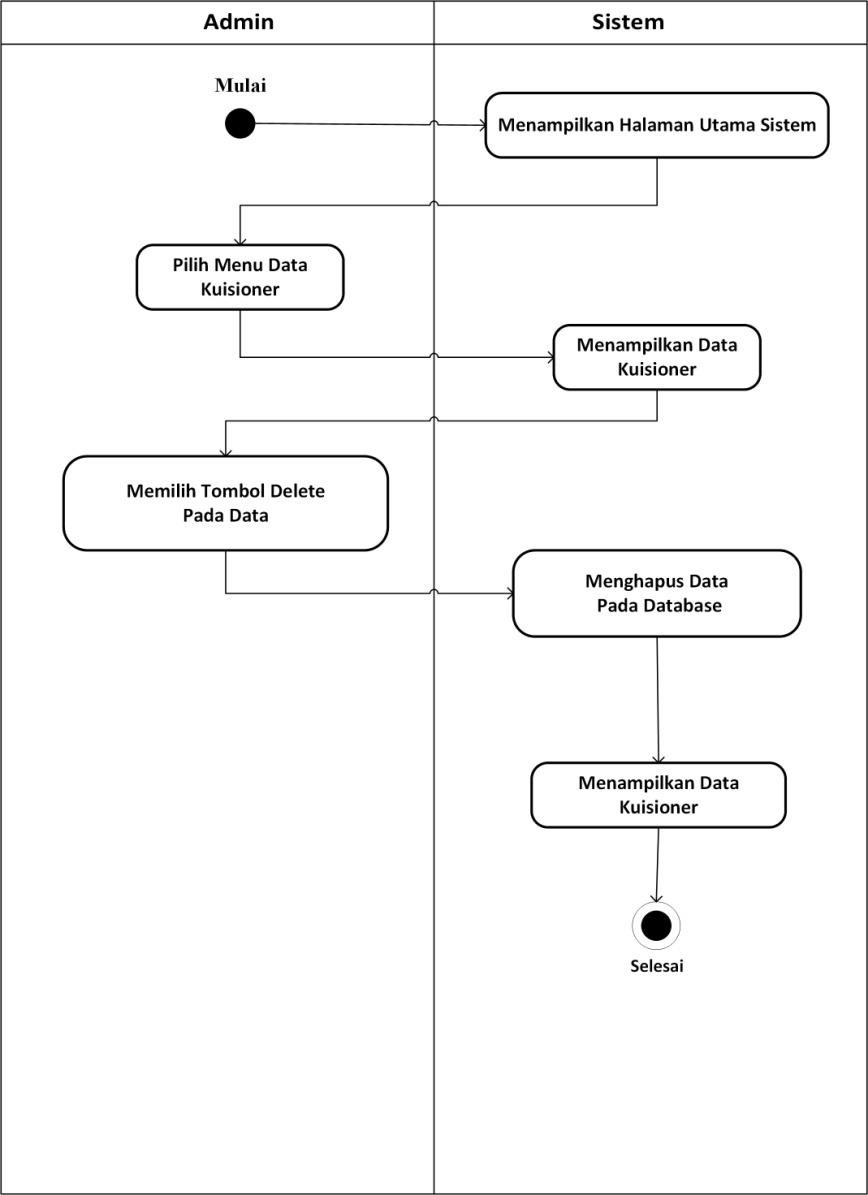
Gambar 4.4 menjelaskan proses apabila admin ingin mengubah data, maka terlebih dahulu sistemmenampilkan data kuisioner, lalu adminmemilih data yang akan diubah, sistem menampilkan *form* ubah data, adminmeng-*input* data yang akan diubah. Setelah divalidasi data yang baru akan ter-*update* dan tersimpan ke *database.*



## Gambar 4.4 *Activity Diagram* Mengelola Data Update Kuisioner

### 4. *Activity Diagram* Mengelola Data Delete Kuisioner

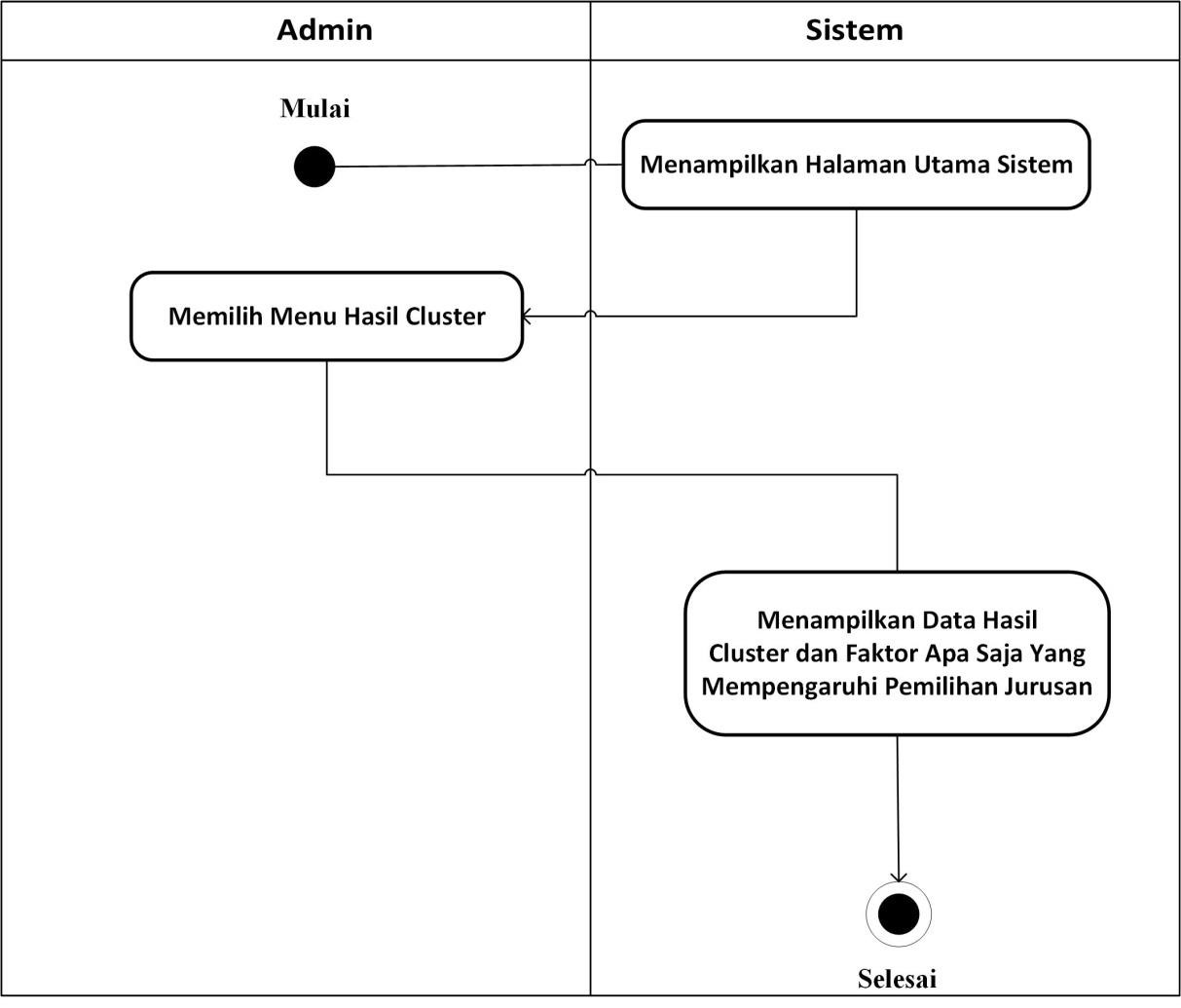
Gambar 4.5 menjelaskan bahwa ketika adminmenghapus data maka terlebih dahulu sistem menampilkan data, lalu adminmemilih data yang akan dihapus. Setelah divalidasi, data yang terpilih akan terhapus dari *database*.



## Gambar 4.5 *Activity Diagram* Mengelola Data Delete Kuisioner

### 5. *Activity Diagram* Analisa Hasil *Cluster*

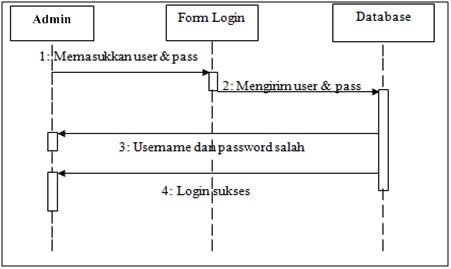
*Activity diagram* Hasil *Cluster* Faktor Terbesar Pemilihan Jurusan menjelaskan bagaimana admin melakukan akses terhadap Hasil *Clustering* Faktor Terbesar Pemilihan Jurusan. Admin yang sudah berhasil masuk ke dalam sistem, memilih menu Hasil *Cluster*. Untuk melakukan akses hasil *Cluster* dan Faktor Apa Terbesar Pemilihan Jurusan, admin memilih menu Hasil *Cluster* dan sistem akan menampilkan halaman hasil *Cluster* Faktor Terbesar Pemilihan Jurusan dan data statistik mahasiswa. Untuk lebih jelasnya, dapat dilihat pada Gambar 4.6



**Gambar 4.6 *Activity Diagram* Hasil *Cluster***

### c. *Sequence Diagram* 1. *Sequence Diagram Login*

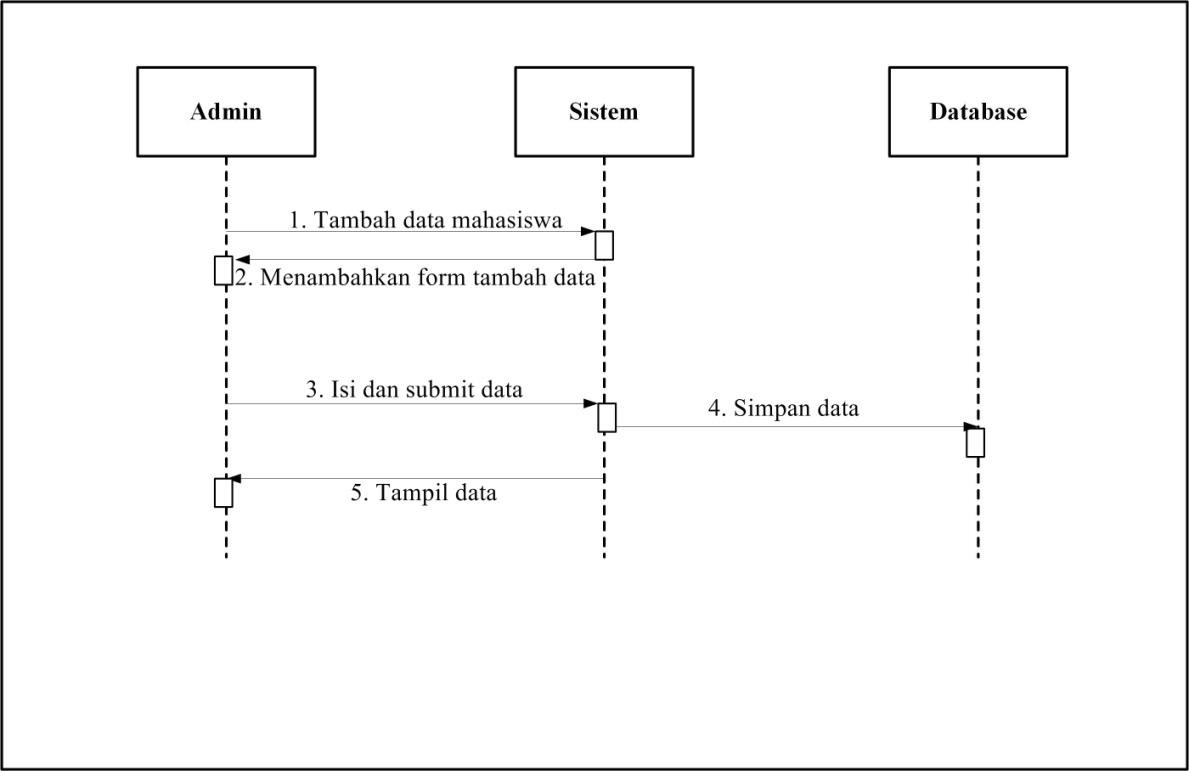
Pada gambar 4.7 menunjukkan alur dari *sequence diagram login*. Pengguna memasukkan *username* dan *password* sehingga masuk di *form login*, jika *username* dan *password* salah maka akan kembali ke *login* jika masuk, *username* akan di kirim ke *database*.



**Gambar 4.7 *Sequence Diagram Login Admin***

### 2. *Sequnce Diagram* Input Data Kuisioner Mahasiswa

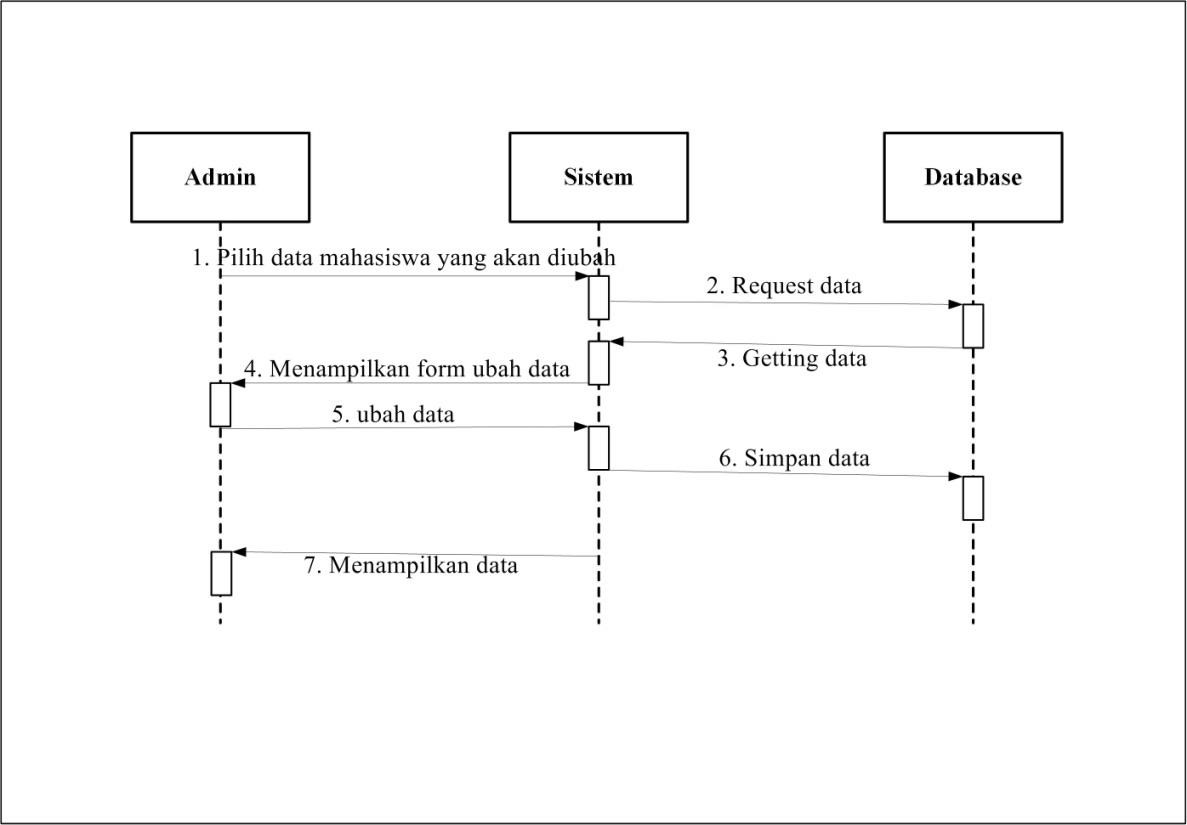
Gambar 4.8 admin melakukan *input* data kuisioner mahasiswa ke sistem. Sistem menampilkan *form* tambah data yang kemudian admin dapat mengisi dan *submit* data yang akan ditambah. Kemudian sistem menyimpan data ke *database*, dan menampilkan ke layar.



## Gambar 4.8 *Sequence Diagram* Input Data Kuisioner Mahasiswa

### 3. *Sequence Diagram* Ubah Data Kuisioner Mahasiswa

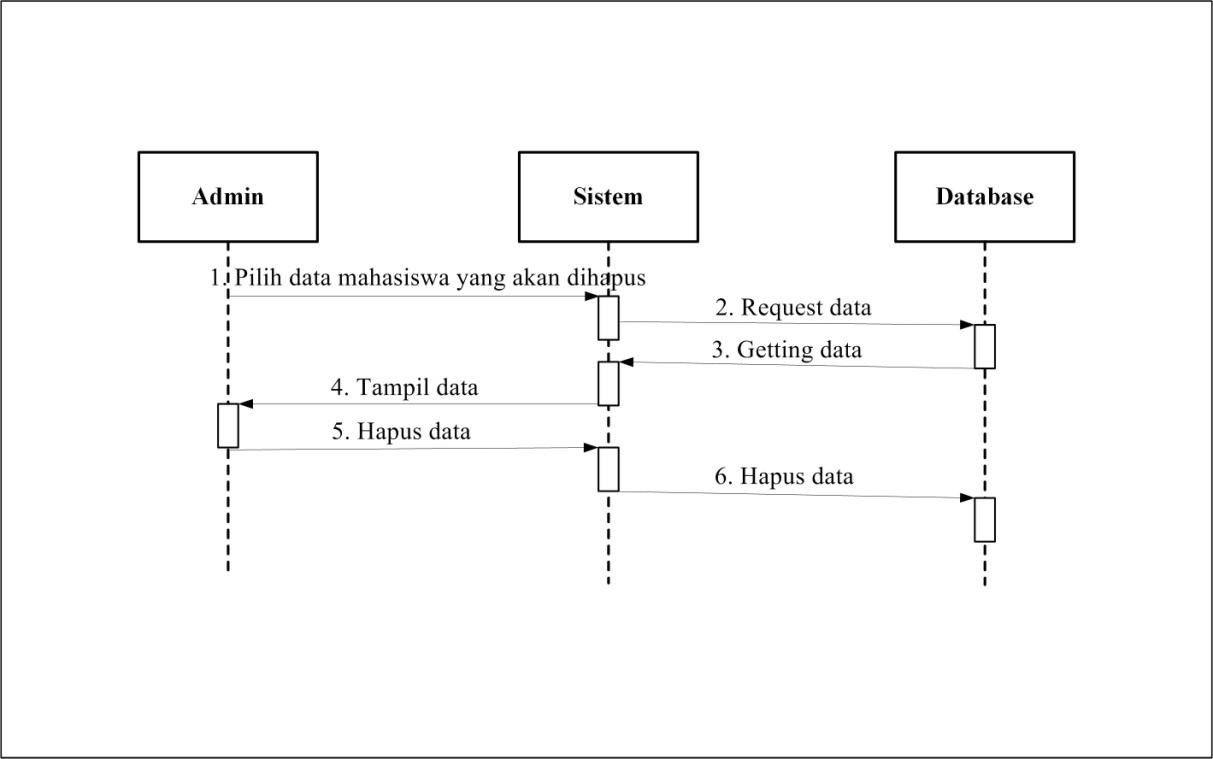
Gambar 4.9 adminmelakukan ubahdata ke sistem. Kemudian sistem mengambil data dari *database*. Kemudian database mengirim kembali data yang dipilih ke sistem. Selanjutnya admin mengubah data. Data yang di ubah akan di simpan ke *database*.



## Gambar 4.9 *Sequence Diagram* Ubah Data Kuisioner Mahasiswa

### 4. *Sequence Diagram* Hapus Data Kuisioner Mahasiswa

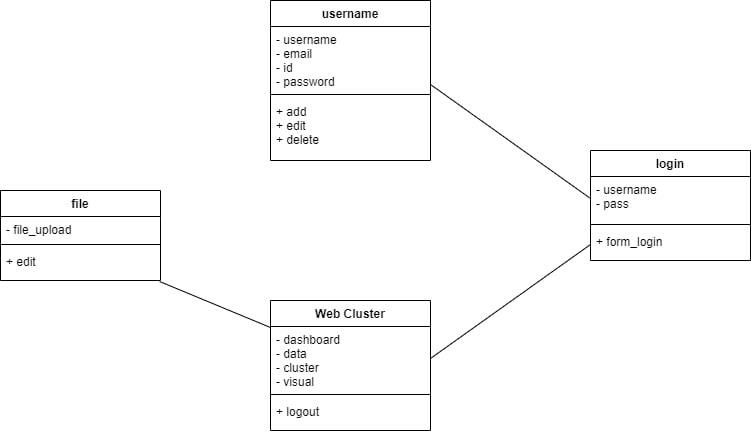
Gambar 4.10 adminmemilih data mahasiswa yang akan dihapus pada sistem, kemudian sistem mengambil data yang dipilih untuk dicocokkan ke *database* dan dari database mengirim kembali data yang telah dipilih. Sistem menampilkan data ke admin. Selanjutnya adminmenghapus data pada sistem dan selanjutnya data tersebut dihapus dari *database.*



## Gambar 4.10 *Sequence Diagram* Hapus Data Kuisioner Mahasiswa

### d. Class Diagram

Pada Gambar 4.11 menunjukkan *Class Diagram* dari Aplikasi Faktor Terbesar Sumber Informasi Pemilihan Jurusan.

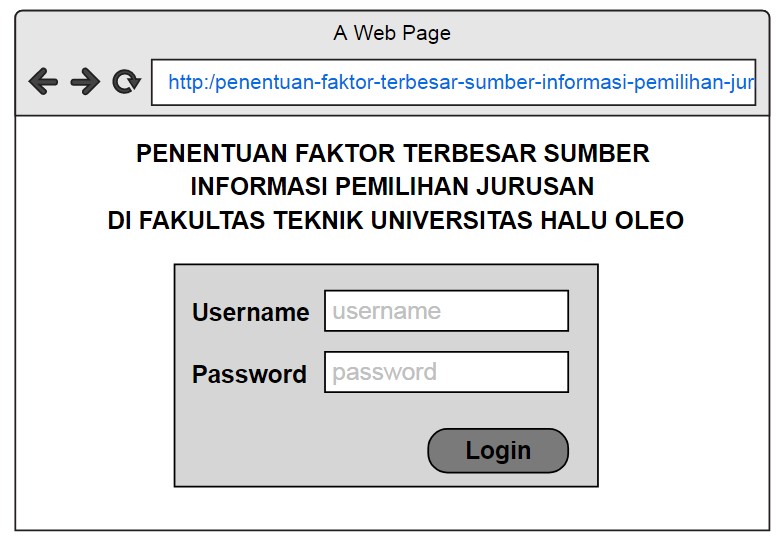


## Gambar 4.11 *Class Diagram* Faktor Terbesar Pemilihan Jurusan

**4.4 Rancangan Antarmuka *(Interface)***

### 1. Halaman *Login*

Pada Gambar 4.12 menampilkan halaman *login* sebelum masuk ke sistem. Admin melakukan proses *login* terlebih dahulu pada halaman ini sebelum masuk ke sistem*,* dimulai dengan mengisi *username* dan *password* kemudian terdapat tombol *login* untuk msuk ke sistem.



## Gambar 4.12 Halaman *Login*

### 2. Halaman Beranda

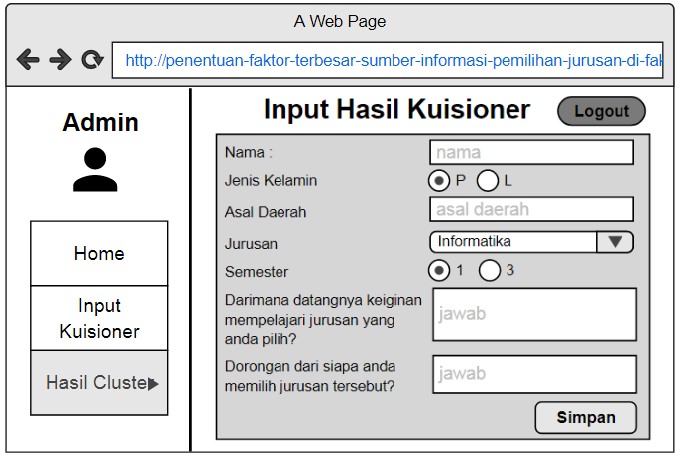
Pada Gambar 4.13 menampilkan halaman beranda atau halaman awal bagian karyawan toko. Terdapat 3 menu pada halaman ini yaitu menu beranda seperti terdapat pada gambar, menu imput data kuisioner, dan menu hasil analisa.



## Gambar 4.13 Halaman Beranda

**3. Halaman *Form Input* Data Kuisioner Mahasiswa**

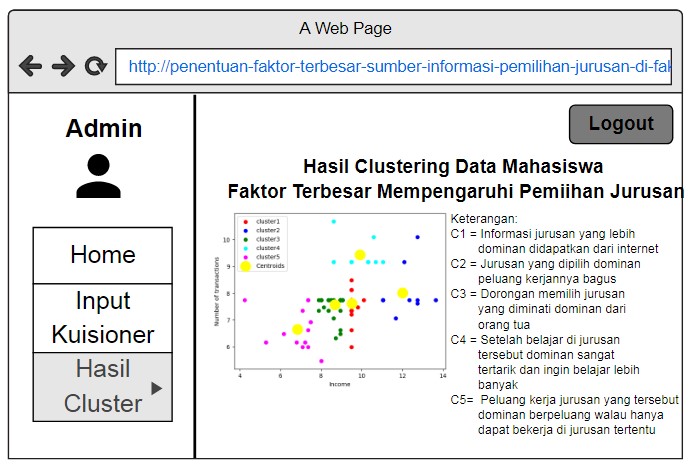
Pada Gambar 4.14 menampilkan halaman *form input* kuisioner. Di halaman ini terdapat *columns* yang harus diisi oleh *user* bagian karyawan toko apabila ingin melakukan *input* data barang masuk, dan terdapat *button “submit”* untuk selanjutnya akan *input/*tambah data barang masuk ke sistem*.*



## Gambar 4.14 Halaman Form Input Data Kuisioner Mahasiswa

**4. Halaman Hasil Analisa**

Pada Gambar 4.15 Menampilkan halaman hasil proses analisa data yang telah di clustering dan faktor apa saja yang mepengaruhi.



**Gambar 4.15 Halaman Hasil Analisa**

## BAB V

**IMPLEMENTASI DAN PENGUJIAN SISTEM**

### 5.1 Implementasi Sistem

Pada tahap implementasi merupakan tahap dimana hasil analisis dan perancangan diimplementasikan dalam bentuk Penentuan Faktor Terbesar Sumber Informasi Pemilihan Jurusan di Fakultas Teknik Universitas Halu Oleo menggunakan bahasa pemrograman *Python* yang siap untuk dioperasikan. Sistem ini menggunakan metode *Clustering K-Medoids.*

#### 1.1.1 Tampilan Halaman *Login*

Halaman *login* merupakan halaman yang pertama kali muncul pada saat sistem dijalankan. Halaman *login* ini menerima masukan *username* dan *password.* Hak akses hanya dapat dilakukan oleh admin. Tampilan halaman *login* dapat dilihat pada Gambar 5.1.

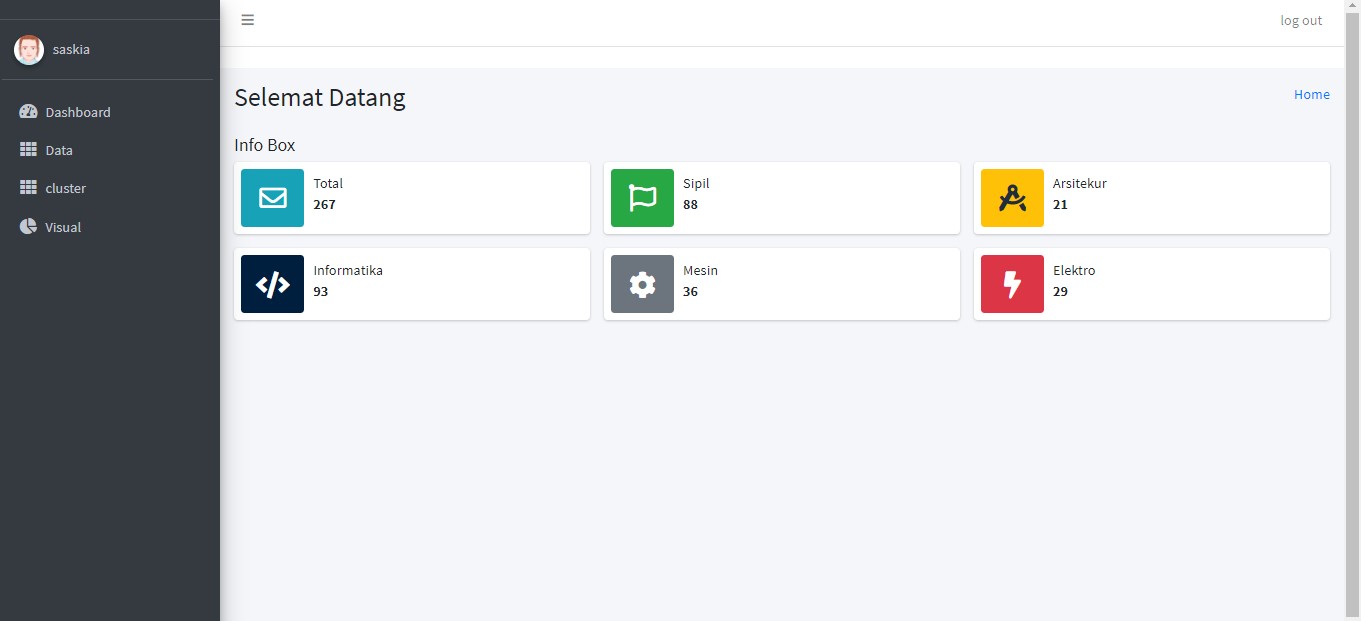


## Gambar 5.1 Tampilan Halaman *Login*

38

### 5.1.2 Tampilan Halaman *Dashboard*

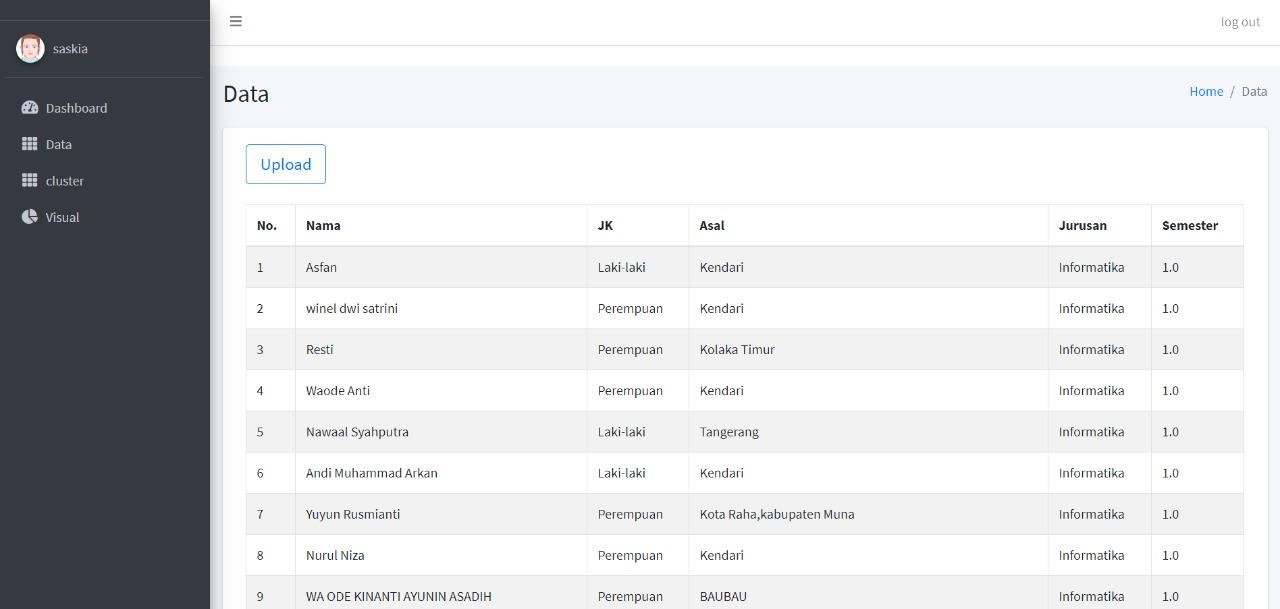
Pada halaman dashboard terdapat empat menu yaitu Menu Data Mahasiswa, Menu Hasil Cluster, dan Menu Visual. Tampilan halaman menu utama dapat dilihat pada Gambar 5.2.



## Gambar 5.2 Tampilan Halaman Dashboard

### 5.1.3 Tampilan Halaman Menu Data Mahasiswa

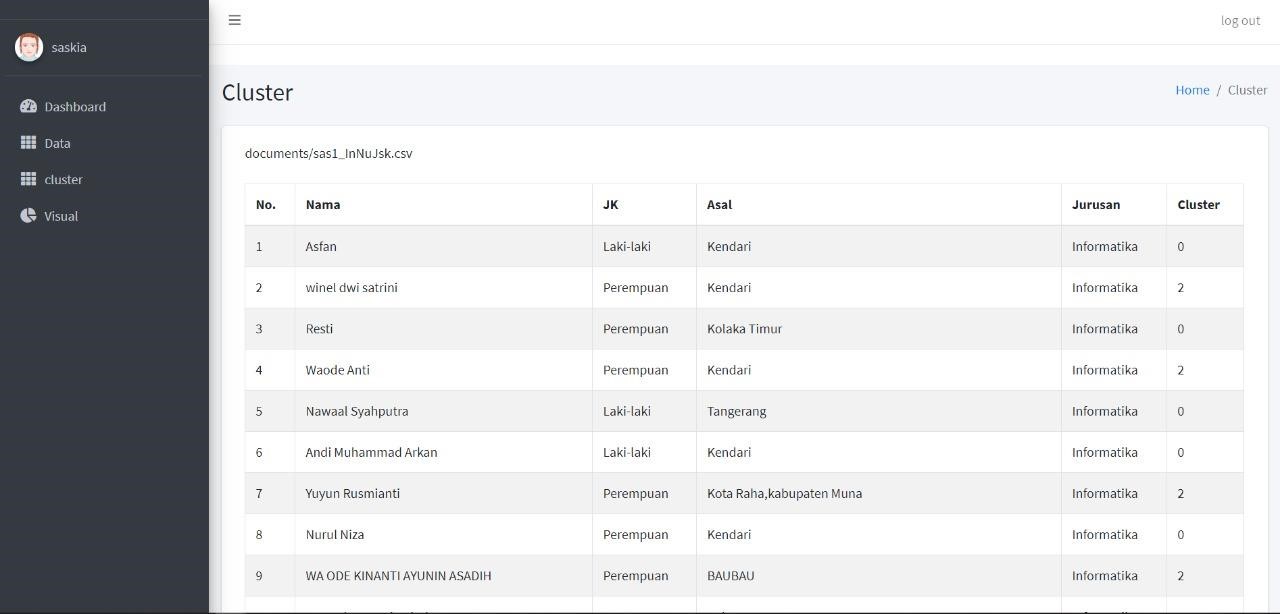
Halaman ini merupakan halaman yang menampilkan data mahasiswa yang dapat diolah oleh admin dari kuisioner diupload dalam bentuk *csv.*



## Gambar 5.3 Tampilan Halaman Menu Data Mahasiswa

### 5.1.4 Tampilan Halaman Menu Data Hasil Cluster

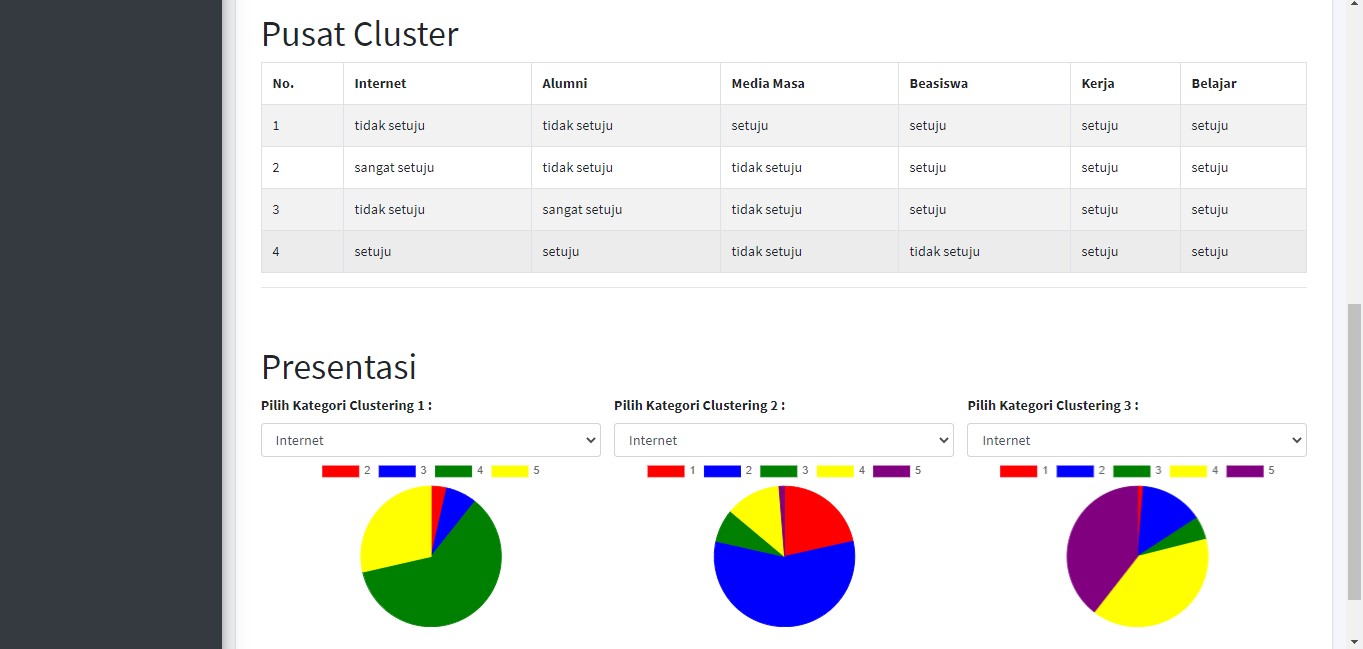
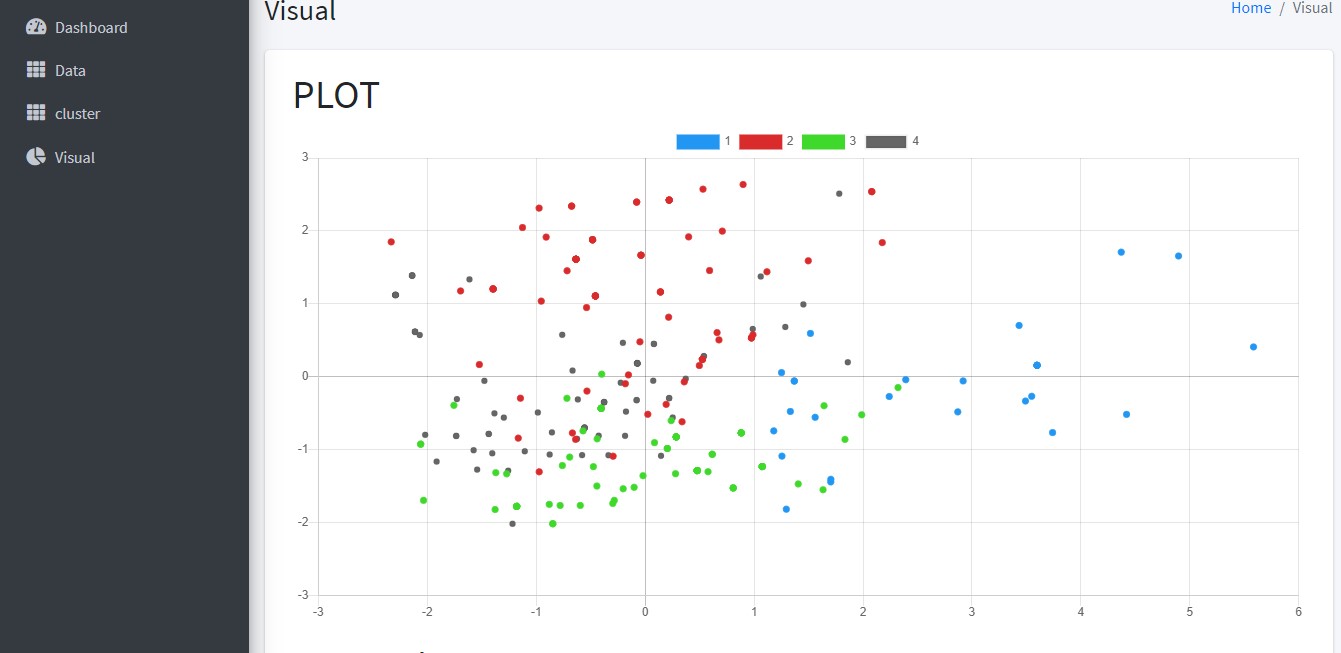
Halaman ini merupakan halaman yang menampilkan proses analisa pengelompokan data yang telah di clustering dan faktor dominan yang memprngaruhi pemilihan jurusan



## Gambar 5.4 Tampilan Halaman Menu Hasil Cluster

### 5.1.5 Tampilan Halaman Visual

Halaman ini merupakan halaman yang menampilkan pengelompokan data menggunakan algoritma *K-Medoids.*



## Gambar 5.5 Tampilan Halaman Visual

### 5.2 Implementasi Algoritma

#### 5.2.1 Implementasi Algoritma *K-Medoids*

|  |
| --- |
| import numpy as np # import keras as keras  # import matplotlib.pyplot as plt  # import pandas as pd  # from sklearn.preprocessing import LabelEncoder  def euclideanDistance(x, y):  '''  Euclidean distance between x, y  -------- Return d: float ''' squared\_d = 0 for i in range(len(x)): squared\_d += (x[i] - y[i]) \*\* 2 d = np.sqrt(squared\_d) return d  class k\_medoids:  def \_\_init\_\_(self, k=2, max\_iter=300, has\_converged=False):  '''  Class constructor  Parameters  ----------   * k: number of clusters. * max\_iter: number of times centroids will move * has\_converged: to check if the algorithm stop or not ''' self.k = k   self.max\_iter = max\_iter self.has\_converged = has\_converged self.medoids\_cost = []  def initMedoids(self, X):  '''  Parameters  ----------  X: input data.  '''  self.medoids = [] |

|  |
| --- |
| # Starting medoids will be random members from data set X  indexes = np.random.randint(0, len(X) - 1, self.k)  self.medoids = X[indexes]  for i in range(0, self.k):  self.medoids\_cost.append(0)  def isConverged(self, new\_medoids):  '''  Parameters  ---------- new\_medoids: the recently calculated medoids to be compared with the current medoids stored in the class  '''  return set([tuple(x) for x in self.medoids])  == set([tuple(x) for x in new\_medoids])  def updateMedoids(self, X, labels):  '''  Parameters ----------  labels: a list contains labels of data points '''  self.has\_converged = True    # Store data points to the current cluster they belong to clusters = [] for i in range(0, self.k):  cluster = [] for j in range(len(X)): if (labels[j] == i): cluster.append(X[j]) clusters.append(cluster)    # Calculate the new medoids new\_medoids = [] for i in range(0, self.k):  new\_medoid = self.medoids[i]  old\_medoids\_cost = self.medoids\_cost[i] for j in range(len(clusters[i])): |

|  |
| --- |
| # Cost of the current data points to be compared with the current optimal cost cur\_medoids\_cost = 0 for dpoint\_index in range(len(clusters[i])):  cur\_medoids\_cost += euclideanDistance(clusters[i][j], clusters[i][dpoint\_index])    # If current cost is less than current optimal cost,  # make the current data point new medoid of the cluster  if cur\_medoids\_cost < old\_medoids\_cost:  new\_medoid = clusters[i][j] old\_medoids\_cost = cur\_medoids\_cost    # Now we have the optimal medoid of the current cluster  new\_medoids.append(new\_medoid)    # If not converged yet, accept the new medoids if not self.isConverged(new\_medoids): self.medoids = new\_medoids self.has\_converged = False  def fit(self, X):  '''  FIT function, used to find clusters  Parameters  ----------  X: input data.  '''  self.initMedoids(X)  for i in range(self.max\_iter): # Labels for this iteration cur\_labels = [] for medoid in range(0, self.k):  # Dissimilarity cost of the current cluster  self.medoids\_cost[medoid] = 0 for k in range(len(X)): |

|  |
| --- |
| # Distances from a data point to each of the medoids d\_list = [] for j in range(0, self.k):    d\_list.append(euclideanDistance(self.medoids[j],  X[k]))  # Data points' label is the medoid which has minimal distance to it    cur\_labels.append(d\_list.index(min(d\_list)))    self.medoids\_cost[medoid] += min(d\_list)    self.updateMedoids(X, cur\_labels)  if self.has\_converged:  break    return np.array(self.medoids)  def predict(self, data):  '''  Parameters ---------- data: input data.    Returns: ----------  pred: list cluster indexes of input data  '''  pred = [] for i in range(len(data)):  # Distances from a data point to each of the medoids  d\_list = [] for j in range(len(self.medoids)):    d\_list.append(euclideanDistance(self.medoids[j], data[i]))  pred.append(d\_list.index(min(d\_list)))    return np.array(pred) |

### 5.3 Pengujian Sistem

Pengujian merupakan tahap yang utama dalam pembuatan suatu aplikasi. Pengujian berfungsi untuk untuk mengetahui bahwa sistem sudah berjalan sesuai dengan kasus dan metode yang digunakan atau belum. Sistem dikatakan benar dan berhasil jika sudah bisa menghasilkan output yang benar. Pengujian ini dilakukan dengan cara menginputkan istilah keliru yang ingin dicari untuk menguji pengoreksian kata berjalan dengan baik dalam aplikasi.

#### 5.3.1 Pengujian *Black Box*

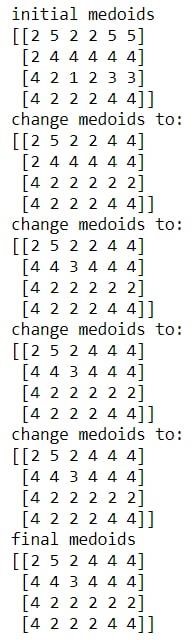
Pengujian *black box* dilakukan dengan menguji perangkat lunak dari segi fungsionalitas perangkat lunak. serta mengevaluasi berdasarkan tampilan dari aplikasi , tanpa mengetahui proses detail yang terjadi. Fungsionalitas perangkat lunak yang diuji sesuai dengan use case pada tahap perancangan. Peneliti membagi pengujian menjadi enam bagian.

## Tabel 5.1 Tabel Pengujian Aplikasi

|  |  |  |  |
| --- | --- | --- | --- |
| **Detail Uji** | **Proses** | **Output** | **Hasil Uji** |
| Membuka  Aplikasi Halaman Login | Menampilkan Halaman Login | Login aplikasi ditampilkan | Sesuai |
| Membuka  Aplikasi Halaman Beranda | Menampilkan  beranda aplikasi | Beranda aplikasi ditambikan | Sesuai |
| Memilih Halaman  Menu Data  Mahasiswa | Menampilkan data mahasiswa yang  diolah oleh admin dari kuisioner dalam bentuk csv | Menu data mahasiswa ditampilkan | Sesuai |
| Memilih Halaman  Menu Data Hasil Cluster | Menampilkan ptoses analisa  pengelompokan  data yang telah di clustering | Menu data hasil cluster ditampilkan | Sesuai |
| Memilih Halaman Menu Visual | Menampilkan pengelompokkan  data menggunakan algoritma *K-*  *Medoids* | Menu Visual ditampilkan | Sesuai |

### 5.4 Pengujian Algoritma

#### 1.4.1 Poses Algoritma *K-Medoids* Pada System



## Gambar 5.6 Proses Iterasi *K-Medoids* Pada *System*

Seperti pada contoh gambar 5.6 system melakukan proses *k-medoids* sebanyak lima literasi sehingga diperoleh final medoids yang akan ditampikan pada web.

### 1.4.2 Perhitungan Manual dengan Metode *Clustering K-Medoids*

Tahapan perhitungan manual dengan menggunakan metode *Clustering KMedoids* untuk mengelompokkan Data Masiswa adalah sebagai berikut:

1. Admin memilih menu data mahasiswa.
2. Admin memilih menu visual.
3. Data yang digunakan sebagai sampel sebanyak 10 data mahasiswa diambil secara acak dari data kuisioner mahasiswa. Data tersebut akan dikelompokan berdasarkan pilihan terbanyak yang didapatkan dari data kuisioner.

Data pada Tabel 5.1 merupakan data yang digunakan untuk melakukan percobaan perhitungan manual :

## Tabel 5.2 Data Mahasiswa Berdasarkan Pilihan Terbanyak

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Nama | Internet | Alumni  UHO | Media  Massa  (Koran) | Tertarik  Dengan  Beasiswa | Peluang  Kerja yang  Bagus | Sangat tertarik  dan ingin  belajar lebih banyak |
| 1. | Gita | Setuju | Setuju | Setuju | Tidak Setuju | Setuju | Setuju |
| 2. | Faza | Sangat  Setuju | Netral | Sangat  Setuju | Netral | Sangat  Setuju | Sangat  Setuju |
| 3. | Asfan | Sangat  Setuju | Tidak Setuju | Tidak Setuju | Setuju | Sangat  Setuju | Sangat  Setuju |
| 4. | Anggun | Setuju | Tidak Setuju | Setuju | Setuju | Sangat  Setuju | Sangat  Setuju |
| 5. | Fauziah | Netral | Setuju | Tidak Setuju | Tidak Setuju | Setuju | Setuju |
| 6. | Arkam | Tidak Setuju | Setuju | Sangat  Setuju | Sangat  Tidak  Setuju | Netral | Setuju |
| 7. | Helda | Tidak Setuju | Tidak Setuju | Tidak Setuju | Setuju | Sangat  Setuju | Sangat  Setuju |
| 8. | Kadek | Tidak Setuju | Sangat  Setuju | Tidak Setuju | Setuju | Setuju | Setuju |
| 9. | Randi | Tidak Setuju | Netral | Setuju | Sangat  Tidak  Setuju | Setuju | Setuju |
| 10  . | Razul | Tidak Setuju | Tidak Setuju | Tidak Setuju | Setuju | Setuju | Setuju |

Jika dilihat dari Tabel 5.1 akan sulit untuk melakukan cluster dengan menggunakan metode clustering K–Medoids sehingga data terlebih dahulu dinormalisasikan dengan cara memasukkan nilai dari setiap kriteria.

## Tabel 5.3 Hasil Normalisasi Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Alternatif | Kriteria | | | |  |  |
| C1  Internet | C2  Alumni  UHO | C3  Media  Massa  (Koran) | C4  Tertarik  Dengan  Beasiswa | C5  Peluang  Kerja yang  Bagus | C6  Sangat tertarik  dan ingin  belajar lebih banyak |
| A1 | 4 | 4 | 4 | 2 | 4 | 4 |
| A2 | 5 | 3 | 5 | 3 | 5 | 5 |
| A3 | 5 | 2 | 2 | 4 | 5 | 5 |
| A4 | 4 | 2 | 4 | 4 | 5 | 5 |
| A5 | 3 | 4 | 2 | 2 | 4 | 4 |
| A6 | 2 | 4 | 5 | 1 | 3 | 4 |
| A7 | 2 | 2 | 2 | 4 | 5 | 5 |
| A8 | 2 | 5 | 2 | 4 | 4 | 4 |
| A9 | 2 | 3 | 4 | 1 | 4 | 4 |
| A10 | 2 | 2 | 2 | 4 | 4 | 4 |
| Langkah – langkah pengelompokan data yang menggunakan metode | | | | | | |

*clustering* K-Medoids yaitu:

1. Ditentukan *k* (jumlah *cluster*) dari *n* objek adalah 3
2. Menentukan cluster awal sebagai medoid

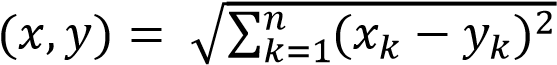
Penentuan nilai cluster awal ditentukan secara random, yaitu:

Medoid 1: A1

Medoid 2: A6

Medoid 3: A10

1. Menghitung jarak obyek ke masing-masing medoid yang telah dipilih. Untuk menentukan jarak obyek ke masing-masing medoid digunakan rumus Euclidian *Distance* :

𝑑……………………………….……….......... (2.1)

**Iterasi pertama:**

Jarak medoid 1

DA1 = √(4 − 4)2 + (4 − 4)2 + (4 − 4)2 + (2 − 2)2 + (4 − 4)2 + (4 − 4)2

= √(0)2 + (0)2 + (0)2 + (0)2 + (0)2 + (0)2

= √0 + 0 + 0 + 0 + 0 + 0



= 0

DA2 = √(5 − 4)2 + (3 − 4)2 + (5 − 4)2 + (3 − 2)2 + (5 − 4)2 + (5 − 4)2

= √(1)2 + (−1)2 + (1)2 + (1)2 + (1)2 + (1)2

= √1 + 1 + 1 + 1 + 1 + 1

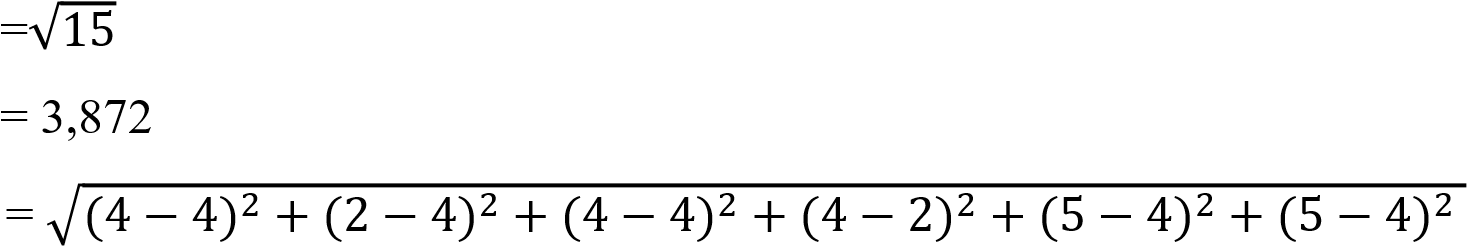


= 2,449

DA3 = √(5 − 4)2 + (2 − 4)2 + (2 − 4)2 + (4 − 2)2 + (5 − 4)2 + (5 − 4)2

= √(1)2 + (−2)2 + (−2)2 + (2)2 + (1)2 + (1)2

= √1 + 4 + 4 + 4 + 1 + 1

DA4 

= √(0)2 + (−2)2 + (0)2 + (2)2 + (1)2 + (1)2

= √0 + 4 + 0 + 4 + 1 + 1



= 3,162

DA5 = √(3 − 4)2 + (4 − 4)2 + (2 − 4)2 + (2 − 2)2 + (4 − 4)2 + (4 − 4)2

= √(−1)2 + (0)2 + (−2)2 + (0)2 + (0)2 + (0)2

= √1 + 0 + 4 + 0 + 0 + 0



= 2,236

DA6 = √(2 − 4)2 + (4 − 4)2 + (5 − 4)2 + (1 − 2)2 + (3 − 4)2 + (4 − 4)2

= √(−2)2 + (0)2 + (1)2 + (−1)2 + (−1)2 + (0)2

= √4 + 0 + 2 + 1 + 1 + 0



= 2,828

DA7 = √(2 − 4)2 + (2 − 4)2 + (2 − 4)2 + (4 − 2)2 + (5 − 4)2 + (5 − 4)2

= √(−2)2 + (−2)2 + (−2)2 + (2)2 + (1)2 + (1)2

= √4 + 4 + 4 + 4 + 1 + 1



= 4,242

DA8 = √(2 − 4)2 + (5 − 4)2 + (2 − 4)2 + (4 − 2)2 + (4 − 4)2 + (4 − 4)2

= √(2)2 + (1)2 + (−2)2 + (2)2 + (0)2 + (0)2

= √4 + 1 + 4 + 4 + 0 + 0



= 3,605

DA9 = √(2 − 4)2 + (3 − 4)2 + (4 − 4)2 + (1 − 2)2 + (4 − 4)2 + (4 − 4)2 = √(−2)2 + (1)2 + (0)2 + (1)2 + (0)2 + (0)2

= √4 + 1 + 0 + 1 + 0 + 0



= 2,236

DA10 = √(2 − 4)2 + (2 − 4)2 + (2 − 4)2 + (4 − 2)2 + (4 − 4)2 + (4 − 4)2

= √(−2)2 + (−2)2 + (−2)2 + (2)2 + (2)2 + (2)2

= √4 + 4 + 4 + 4 + 4 + 4



= 4,898

Jarak medoid 2

DA1 = √(4 − 2)2 + (4 − 4)2 + (4 − 5)2 + (2 − 1)2 + (4 − 3)2 + (4 − 4)2

= √(2)2 + (0)2 + (−1)2 + (1)2 + (1)2 + (0)2

= √4 + 0 + 1 + 1 + 1 + 0



= 2,236

DA2 = √(5 − 2)2 + (3 − 4)2 + (5 − 5)2 + (3 − 1)2 + (5 − 3)2 + (5 − 4)2

= √(1)2 + (−1)2 + (0)2 + (1)2 + (2)2 + (1)2

= √1 + 1 + 0 + 1 + 4 + 1



= 2,645

DA3 = √(5 − 2)2 + (2 − 4)2 + (2 − 5)2 + (4 − 1)2 + (5 − 3)2 + (5 − 4)2

= √(3)2 + (−2)2 + (−3)2 + (3)2 + (2)2 + (1)2

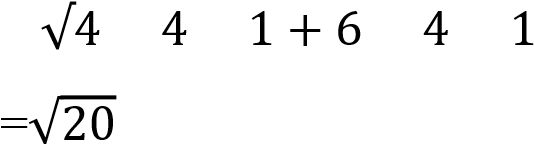
= √6 + 4 + 6 + 6 + 4 + 1



= 5,196

DA4 = √(4 − 2)2 + (2 − 4)2 + (4 − 5)2 + (4 − 1)2 + (5 − 3)2 + (5 − 4)2

= √(2)2 + (−2)2 + (−1)2 + (3)2 + (2)2 + (1)2



= 4,472

DA5 = √(3 − 2)2 + (4 − 4)2 + (2 − 5)2 + (2 − 1)2 + (4 − 3)2 + (4 − 4)2

= √(1)2 + (0)2 + (−3)2 + (1)2 + (1)2 + (0)2

= √1 + 0 + 6 + 1 + 1 + 0



=3

DA6 = √(2 − 2)2 + (4 − 4)2 + (5 − 5)2 + (1 − 1)2 + (3 − 3)2 + (4 − 4)2

= √(0)2 + (0)2 + (0)2 + (0)2 + (0)2 + (0)2

= √0 + 0 + 0 + 0 + 0 + 0



= 0

DA7 = √(2 − 2)2 + (2 − 4)2 + (2 − 5)2 + (4 − 1)2 + (5 − 3)2 + (5 − 4)2

= √(0)2 + (−2)2 + (−3)2 + (3)2 + (2)2 + (1)2

= √0 + 4 + 6 + 6 + 4 + 1



= 4,582

DA8 = √(2 − 2)2 + (5 − 4)2 + (2 − 5)2 + (4 − 1)2 + (4 − 3)2 + (4 − 4)2

= √(0)2 + (1)2 + (−3)2 + (3)2 + (1)2 + (0)2

= √0 + 1 + 6 + 6 + 1 + 0



= 3,741

DA9 = √(2 − 2)2 + (3 − 4)2 + (4 − 5)2 + (1 − 1)2 + (4 − 3)2 + (4 − 4)2

= √(0)2 + (−1)2 + (−1)2 + (0)2 + (1)2 + (0)2

= √0 + 1 1 0 1 0



= 1,732

DA10 = √(2 − 2)2 + (2 − 4)2 + (2 − 5)2 + (4 − 1)2 + (4 − 3)2 + (4 − 4)2

= √(0)2 + (−2)2 + (−3)2 + (3)2 + (1)2 + (0)2

= √0 + 4 + 6 + 6 + 1 + 0



= 4,123

Jarak Medoid 3

DA1 = √(4 − 2)2 + (4 − 2)2 + (4 − 2)2 + (2 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(0)2 + (0)2 + (0)2 + (0)2 + (0)2 + (0)2

= √0 + 0 + 0 + 0 + 0 + 0



= 0

DA2 = √(5 − 2)2 + (3 − 2)2 + (5 − 2)2 + (3 − 4)2 + (5 − 4)2 + (5 − 4)2

= √(3)2 + (1)2 + (3)2 + (−1)2 + (1)2 + (1)2

= √6 + 1 + 6 + 1 + 1 + 1



= 4

DA3 = √(5 − 2)2 + (2 − 2)2 + (2 − 2)2 + (4 − 4)2 + (5 − 4)2 + (5 − 4)2

= √(3)2 + (0)2 + (0)2 + (0)2 + (1)2 + (1)2

= √6 + 0 + 0 + 0 + 1 + 1



= 2,828

D4 = √(4 − 2)2 + (2 − 2)2 + (4 − 2)2 + (4 − 4)2 + (5 − 4)2 + (5 − 4)2

= √(2)2 + (0)2 + (2)2 + (0)2 + (1)2 + (1)2

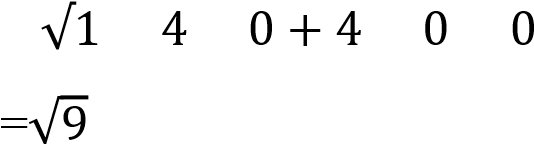
= √4 + 0 + 4 + 0 + 1 + 1



= 3,162

DA5 = √(3 − 2)2 + (4 − 2)2 + (2 − 2)2 + (2 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(1)2 + (2)2 + (0)2 + (−2)2 + (0)2 + (0)2



= 3

DA6 = √(2 − 2)2 + (4 − 2)2 + (5 − 2)2 + (1 − 4)2 + (3 − 4)2 + (4 − 4)2

= √(0)2 + (2)2 + (3)2 + (−3)2 + (−1)2 + (0)2

= √0 + 4 + 6 + 6 + 1 + 0



= 3

DA7 = √(2 − 2)2 + (2 − 2)2 + (2 − 2)2 + (4 − 4)2 + (5 − 4)2 + (5 − 4)2

= √(0)2 + (0)2 + (0)2 + (0)2 + (1)2 + (1)2

= √0 + 0 + 0 + 0 + 1 + 1



= 1,414

DA8 = √(2 − 2)2 + (5 − 2)2 + (2 − 2)2 + (2 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(0)2 + (3)2 + (0)2 + (0)2 + (0)2 + (0)2

= √0 + 6 + 0 + 0 + 0 + 0



= 1,565

DA9 = √(2 − 2)2 + (3 − 2)2 + (4 − 2)2 + (1 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(0)2 + (1)2 + (2)2 + (−3)2 + (0)2 + (0)2

= √0 + 1 + 4 + 6 + 0 + 0



= 3,316

DA0 = √(2 − 2)2 + (2 − 2)2 + (2 − 2)2 + (4 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(0)2 + (0)2 + (0)2 + (0)2 + (0)2 + (0)2

= √0 + 0 0 0 0 0



= 0

Hasil perhitungan jarak *Euclidien Distance iterasi* pertama dilihat pada Tabel 5.4

## Tabel 5.4 Hasil Perhitungan Jarak ke Setiap *Medoid*s Iterasi Pertama

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ALTERNATIF** | **C1** | **C2** | **C3** | ***Cluster*** |
| A1 | 0 | 2,236 | 0 | C1 |
| A2 | 2,449 | 2,645 | 4 | C1 |
| A3 | 3,872 | 5,196 | 2,828 | C3 |
| A4 | 3,162 | 4,472 | 3,162 | C1 |
| A5 | 2,236 | 3 | 3 | C1 |
| A6 | 2,828 | 0 | 3 | C2 |
| A7 | 4,242 | 4,582 | 1,414 | C3 |
| A8 | 3,605 | 3,741 | 1,565 | C3 |
| A9 | 2,23 | 1,73 | 3,74 | C1 |
| A10 | 4,898 | 4,123 | 0 | C3 |
| JUMLAH | 29,741 | 27,253 | 21,393 |  |
| Total |  | 78.387 | |  |

C1 = (A1, A2, A4, A5, A9)

C2 = (A6)

C3 = (A3, A7, A8, A10)

**Iterasi ke-2:**

Menentukan medoid yang akan berhenti jika anggota *cluster* yang terbentuk dari medoid tidak mengalami perubahan.

Medoid 1: A1

Medoid 2: A3

Medoid 3: A8

Jarak medoid 1

DA1 = √(4 − 4)2 + (4 − 4)2 + (4 − 4)2 + (2 − 2)2 + (4 − 4)2 + (4 − 4)2

= √(0)2 + (0)2 + (0)2 + (0)2 + (0)2 + (0)2

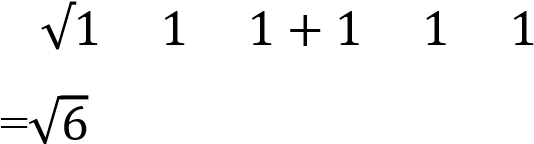
= √0 + 0 + 0 + 0 + 0 + 0



= 0

DA2 = √(5 − 4)2 + (3 − 4)2 + (5 − 4)2 + (3 − 2)2 + (5 − 4)2 + (5 − 4)2

= √(1)2 + (−1)2 + (1)2 + (1)2 + (1)2 + (1)2



= 2,449

DA3 = √(5 − 4)2 + (2 − 4)2 + (2 − 4)2 + (4 − 2)2 + (5 − 4)2 + (5 − 4)2

= √(1)2 + (−2)2 + (−2)2 + (2)2 + (1)2 + (1)2

= √1 + 4 + 4 + 4 + 1 + 1



= 3,872

DA4 = √(4 − 4)2 + (2 − 4)2 + (4 − 4)2 + (4 − 2)2 + (5 − 4)2 + (5 − 4)2

= √(0)2 + (−2)2 + (0)2 + (2)2 + (1)2 + (1)2

= √0 + 4 + 0 + 4 + 1 + 1



= 3,162

DA5 = √(3 − 4)2 + (4 − 4)2 + (2 − 4)2 + (2 − 2)2 + (4 − 4)2 + (4 − 4)2

= √(−1)2 + (0)2 + (−2)2 + (0)2 + (0)2 + (0)2

= √1 + 0 + 4 + 0 + 0 + 0



= 2,236

DA6 = √(2 − 4)2 + (4 − 4)2 + (5 − 4)2 + (1 − 2)2 + (3 − 4)2 + (4 − 4)2

= √(−2)2 + (0)2 + (1)2 + (−1)2 + (−1)2 + (0)2

= √4 + 0 + 2 + 1 + 1 + 0



= 2,828

DA7 = √(2 − 4)2 + (2 − 4)2 + (2 − 4)2 + (4 − 2)2 + (5 − 4)2 + (5 − 4)2

= √(−2)2 + (−2)2 + (−2)2 + (2)2 + (1)2 + (1)2

= √4 + 4 4 4 1 1



= 4,242

DA8 √(2 − 4)2 + (5 − 4)2 + (2 − 4)2 + (4 − 2)2 + (4 − 4)2 + (4 − 4)2

= √(2)2 + (1)2 + (−2)2 + (2)2 + (0)2 + (0)2

= √4 + 1 + 4 + 4 + 0 + 0



= 3,605

DA9 = √(2 − 4)2 + (3 − 4)2 + (4 − 4)2 + (1 − 2)2 + (4 − 4)2 + (4 − 4)2

= √(−2)2 + (0)2 + (0)2 + (1)2 + (0)2 + (0)2

= √4 + 1 + 0 + 1 + 0 + 0



= 2,449

DA10 = √(2 − 4)2 + (2 − 4)2 + (2 − 4)2 + (4 − 2)2 + (4 − 4)2 + (4 − 4)2

= √(−2)2 + (−2)2 + (−2)2 + (2)2 + (2)2 + (2)2

= √4 + 4 + 4 + 4 + 4 + 4



= 4,898p

Jarak medoid 2

DA1 = √(4 − 5)2 + (4 − 2)2 + (4 − 2)2 + (2 − 4)2 + (4 − 5)2 + (4 − 5)2

= √(−1)2 + (2)2 + (2)2 + (2)2 + (−1)2 + (−1)2

= √1 + 4 + 4 + 4 + 1 + 1



= 3,872

DA2 = √(5 − 5)2 + (3 − 2)2 + (5 − 2)2 + (3 − 4)2 + (5 − 5)2 + (5 − 5)2

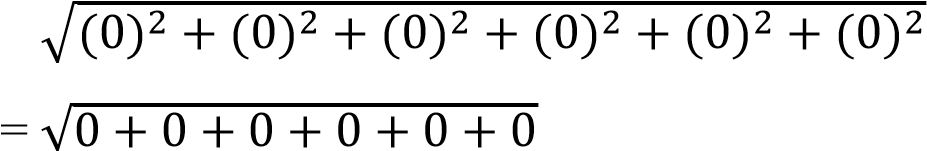
= √(0)2 + (2)2 + (2)2 + (−1)2 + (0)2 + (0)2

= √0 + 4 + 4 + 1 + 0 + 0



= 3

DA3 = √(5 − 5)2 + (2 − 2)2 + (2 − 2)2 + (4 − 4)2 + (5 − 5)2 + (5 − 5)2





= 0

DA4 = √(4 − 5)2 + (2 − 2)2 + (4 − 2)2 + (4 − 4)2 + (5 − 5)2 + (5 − 5)2

= √(−1)2 + (0)2 + (2)2 + (0)2 + (0)2 + (0)2

= √1 + 0 + 4 + 0 + 0 + 0



= 2,236

DA5 = √(3 − 5)2 + (4 − 2)2 + (2 − 2)2 + (2 − 4)2 + (4 − 5)2 + (4 − 5)2

= √(−1)2 + (2)2 + (0)2 + (−2)2 + (−1)2 + (−1)2

= √1 + 4 + 0 + 4 + 1 + 1



= 3,316

DA6 = √(2 − 5)2 + (4 − 2)2 + (5 − 2)2 + (1 − 4)2 + (3 − 5)2 + (4 − 5)2

= √(−3)2 + (2)2 + (3)2 + (−3)2 + (−2)2 + (−1)2

= √6 + 4 + 6 + 6 + 4 + 1



= 5,196

DA7 = √(2 − 5)2 + (2 − 2)2 + (2 − 2)2 + (4 − 4)2 + (5 − 5)2 + (5 − 5)2

= √(−3)2 + (0)2 + (0)2 + (0)2 + (0)2 + (0)2

= √6 + 0 + 0 + 0 + 0 + 0



= 2,449

DA8 = √(2 − 5)2 + (5 − 2)2 + (2 − 2)2 + (4 − 4)2 + (4 − 5)2 + (4 − 5)2

= √(−3)2 + (3)2 + (0)2 + (0)2 + (−1)2 + (−1)2

= √6 + 6 0 0 1 1



= 3,741

DA9 = √(2 − 5)2 + (4 − 2)2 + (4 − 2)2 + (4 − 4)2 + (4 − 5)2 + (4 − 5)2

= √(−3)2 + (2)2 + (2)2 + (0)2 + (−1)2 + (−1)2

= √6 + 4 + 4 + 0 + 1 + 1



= 4

DA10 = √(2 − 5)2 + (2 − 2)2 + (2 − 2)2 + (4 − 4)2 + (4 − 5)2 + (4 − 5)2

= √(−3)2 + (0)2 + (0)2 + (0)2 + (−1)2 + (−1)2

= √6 + 0 + 0 + 0 + 1 + 1



= 2,828

Jarak medoid 3

DA1 = √(4 − 2)2 + (4 − 5)2 + (4 − 2)2 + (2 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(2)2 + (−1)2 + (2)2 + (−2)2 + (0)2 + (0)2

= √4 + 1 + 4 + 4 + 0 + 0



= 3,605

DA2 = √(5 − 2)2 + (3 − 5)2 + (5 − 2)2 + (3 − 4)2 + (5 − 4)2 + (5 − 4)2

= √(1)2 + (−1)2 + (1)2 + (1)2 + (1)2 + (1)2

= √1 + 1 + 1 + 1 + 1 + 1



= 2,449

DA3 = √(5 − 2)2 + (2 − 5)2 + (2 − 2)2 + (4 − 4)2 + (5 − 4)2 + (5 − 4)2

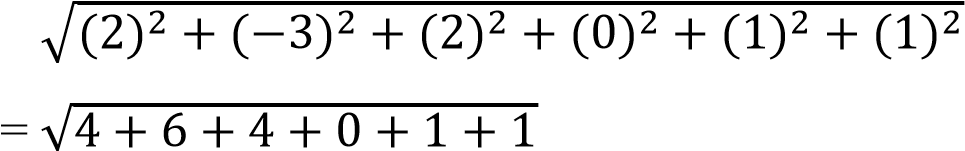
= √(3)2 + (−3)2 + (0)2 + (0)2 + (1)2 + (1)2

= √6 + 6 + 0 + 0 + 1 + 1



= 3,741

DA4 = √(4 − 2)2 + (2 − 5)2 + (4 − 2)2 + (4 − 4)2 + (5 − 4)2 + (5 − 4)2





= 4

DA5 = √(3 − 2)2 + (4 − 5)2 + (2 − 2)2 + (2 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(1)2 + (−1)2 + (0)2 + (−2)2 + (0)2 + (0)2

= √1 + 1 + 0 + 4 + 0 + 0



= 2,449

DA6 = √(2 − 2)2 + (4 − 5)2 + (5 − 2)2 + (1 − 4)2 + (3 − 4)2 + (4 − 4)2

= √(0)2 + (−1)2 + (3)2 + (−3)2 + (1)2 + (0)2

= √0 + 1 + 6 + 6 + 1 + 0



= 3,741

DA7 = √(2 − 2)2 + (2 − 5)2 + (2 − 2)2 + (4 − 4)2 + (5 − 4)2 + (5 − 4)2

= √(0)2 + (3)2 + (0)2 + (0)2 + (1)2 + (1)2

= √0 + 6 + 0 + 0 + 1 + 1



= 2,828

DA8 = √(2 − 2)2 + (5 − 5)2 + (2 − 2)2 + (4 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(0)2 + (0)2 + (0)2 + (0)2 + (0)2 + (0)2

= √0 + 0 + 0 + 0 + 0 + 0



=0

DA9 = √(2 − 2)2 + (3 − 5)2 + (4 − 2)2 + (4 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(−2)2 + (−2)2 + (0)2 + (−3)2 + (0)2 + (0)2

= √4 + 4 + 0 + 9 + 0 + 0



= 4,123

DA10 = √(2 − 2)2 + (2 − 5)2 + (2 − 2)2 + (4 − 4)2 + (4 − 4)2 + (4 − 4)2

= √(0)2 + (3)2 + (0)2 + (0)2 + (0)2 + (0)2

= √0 + 6 + 0 + 0 + 0 + 0



= 2,449

Hasil perhitungan iterasi ke-2 dilihat pada Tabel 5.5.

## Tabel 5.5 Hasil Perhitungan Jarak ke Setiap *Medoid*s Iterasi Kedua

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ALTERNATIF** | **C1** | **C2** | **C3** | ***Cluster*** |
| A1 | 0 | 3,872 | 3,605 | C1 |
| A2 | 2,449 | 3 | 2,449 | C1 |
| A3 | 3,872 | 0 | 3,741 | C2 |
| A4 | 3,162 | 2,236 | 4 | C1 |
| A5 | 2,236 | 3,316 | 2,449 | C1 |
| A6 | 2,828 | 5,196 | 3,741 | C1 |
| A7 | 4,242 | 2,449 | 2,828 | C2 |
| A8 | 3,605 | 3,741 | 0 | C3 |
| A9 | 2,449 | 4 | 2,123 | C1 |
| A10 | 4,898 | 2,828 | 2,449 | C3 |
| JUMLAH | 29,741 | 30,644 | 27,385 |  |
| Total |  | 87,77 | |  |

C1 = (A1, A2, A3, A4, A5, A6, A9)

C2 = (A7)

C3 = (A8, A10)

1. Hitung total simpangan (*S*) dengan menghitung total *distance* baru – total *distance* lama. Jika *S* < 0 maka tukar objek dengan data *cluster* untuk membentuk sekumpulan *k* objek baru sebagai medoid (Setyawati, 2017).

*S* = total *distance* baru – total *distance* lama

*S* = 87,77 - 78.387 = 9,383

1. Karena nilai *S < 0* maka proses pengklasteran dihentikan. Sehingga objek non *medoids* dijadikan sebagai *medoid* baru.
2. Sehingga *cluster* yang di dapatakan seperti pada table 5.5.

## Tabel 5.6 Hasil pengclusteran mahasiswa baru dengan k-medoids

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Nama | Internet | Alumni  UHO | Media Massa | Tertarik  Dengan | Peluang  Kerja | Sangat tertarik dan ingin | *Cluster* |
|  |  |  |  | (Koran) | Beasiswa | yang  Bagus | belajar lebih banyak |  |
| 1. | Gita | Setuju | Setuju | Setuju | Tidak Setuju | Setuju | Setuju | C1 |
| 2. | Faza | Sangat  Setuju | Netral | Sangat  Setuju | Netral | Sangat  Setuju | Sangat  Setuju | C1 |
| 3. | Asfan | Sangat  Setuju | Tidak Setuju | Tidak Setuju | Setuju | Sangat  Setuju | Sangat  Setuju | C3 |
| 4. | Anggun | Setuju | Tidak Setuju | Setuju | Setuju | Sangat  Setuju | Sangat  Setuju | C1 |
| 5. | Fauziah | Sangat  Setuju | Sangat  Setuju | Tidak Setuju | Tidak Setuju | Setuju | Setuju | C1 |
| 6. | Arkam | Tidak Setuju | Setuju | Sangat  Setuju | Sangat  Tidak  Setuju | Netral | Setuju | C2 |
| 7. | Helda | Tidak Setuju | Tidak Setuju | Tidak Setuju | Setuju | Sangat  Setuju | Sangat  Setuju | C3 |
| 8. | Kadek | Tidak Setuju | Sangat  Setuju | Tidak Setuju | Setuju | Setuju | Setuju | C3 |
| 9. | Randi | Tidak Setuju | Setuju | Setuju | Setuju | Setuju | Setuju | C1 |
| 10  . | Razul | Tidak Setuju | Tidak Setuju | Tidak Setuju | Setuju | Setuju | Setuju | C3 |

Dari 10 data yang telah diuji dengan 2 iterasi, pada iterasi ke-1 C1 memiliki jumlah mahasiswa sebanyak 5 (50%), lalu C2 memiliki jumlah mahasiswa sebanyak 1 (10%), dan C3 memiliki jumlah mahasiswa sebanyak 4 (40%).

Pada iterasi ke-2 C1 memiliki jumlah mahasiswa sebanyak 6 (60%), lalu C2 memiliki jumlah mahasiswa sebanyak 2 (20%), dan C3 memiliki jumlah mahasiswa sebanyak 2 (20%).

Karena hasil iterasi ke-2 posisi cluster berubah atau berbeda dengan posisi cluster pada iterasi pertama, disebekan oleh pergantian centroid utama yang mana dipertemukan dari calon medoids ke data mahasiswa yang telah dihitung dan ditentukan yang paling tepat jadi medoidsnya. Maka proses iterasi dihentikan karena sudah dapat diketahui hasil cluster yang membentuk suatu kelompok data.

### 1.5 *Silhoutette Coefficient*

Pengujian dilakukan dengan mengevaluasi kualitas hasil pengelompokan data algoritma *K-Medoids* menggunakan *Silhouette Coefficient* berdasarkan kombinasi masukan nilai klaster yang berbeda. Hasil pengujian ditunjukkan pada Tabel 5.6.

**Tabel 5.6 Hasil *Silhouette Coefficient* dengan Nilai Klaster Berbeda-beda.**

**No**

**Jumlah**

**Klaster**

**Nilai**

***Silhouette***

***Coefficient***

1.

2

0.14637431590665603

2.

3

0.11000736495160417

3.

4

0.23045321036704783

4.

5

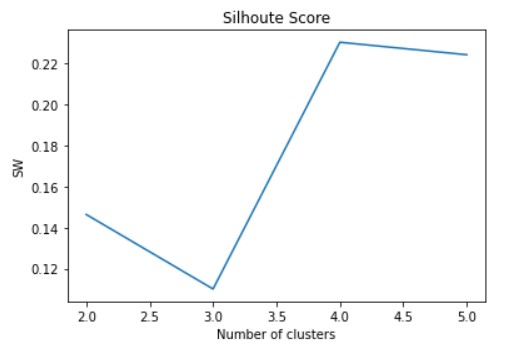
0.22435958179742807

Dari hasil pengujian pada tabel 5.

7

menunjukkan jumlah klaster yang

terbaik adalah klaster 4 dengan dengan nilai silhouette coefficien sebesar 0.23045321036704783 dan klaster yang paling buruk adalah klaster 3 dengan dengana nilai silhouette coefficien adalah 0.11000736495160417.



**Gambar 5.7 Grafik pengaruh perbedaan jumlah klaster dengan 264 data terhadap *silhouette coefficien.***

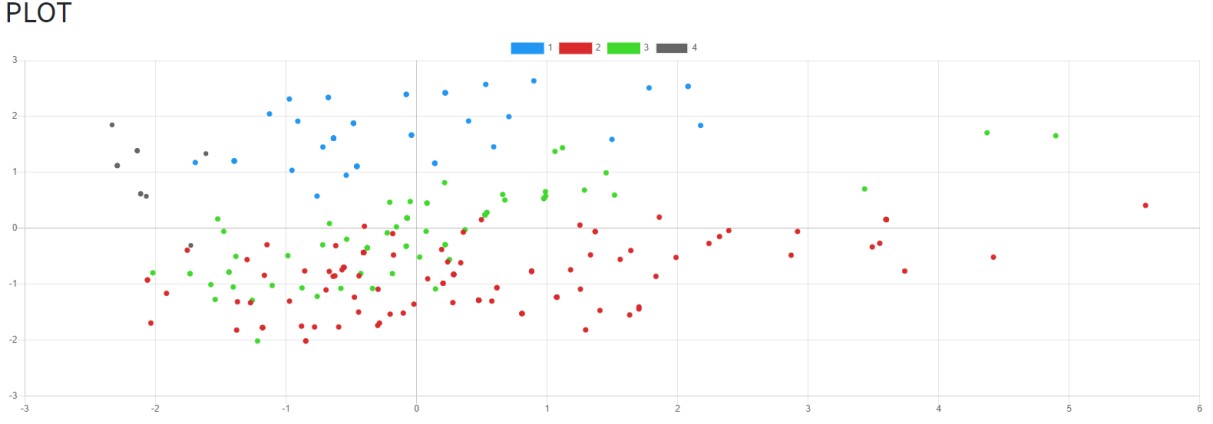
**1.6**

**PLOT**

**Gambar 5.**

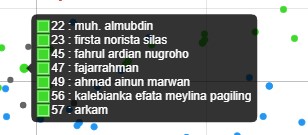
**8**

**Ilustrasi Plot**



Plot terbentuk dari Principal Component Analysis (PCA). Fungsi dari principal component analysis (PCA) adalah dengan merangkum banyak variable dari kuisioner yang memiliki indikasi saling berkorelasi ataupun saling mempengaruhi menjadi satu atau lebih variable baru yang memuat kombinasi dari variabel-variabel kuisioner sebelumnya. Dalam penelitian ini Principal Component Analysis (PCA) digunakan untuk membentuk dua variable yang berfungsi untuk membentuk plot.

Dari plot yang terbentuk terdapat beberapa orang yang mempunyai nilai yang sama. Hal ini terjadi karena mereka memiliki jawaban yang sama. Selain itu terdapat cluster yang mumpunyai nilai yang berdekatan sehinga menyebabkan dua cluster saling bercampur. Berikut diilustrasikan pada Gambar 5.9.



**Gambar 5.9 Ilustrasi Data Plot yang Sama**

## BAB VI

**PENUTUP**

### 6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai Penerapan Algoritma K-Medoids dalam Penentuan Faktor Terbesar Sumber Informasi Pemilihan Jurusan di Fakultas Teknik Universitas Halu Oleo memiliki beberapa kesimpulan antara lain:

1. Penerapan Algoritma K-Medoids dalam Penentuan Faktor Terbesar Sumber Informasi Pemilihan Jurusan di Fakultas Teknik Universitas Halu Oleo telah berhasil dibagun untuk menghasilkan cluster terbesar sumber informasi emilihan urusan di Fakultas Teknik Universitas Halu Oleo.
2. Berdasarkan hasil analisa dengan menerapkan metode K-Medoids menghasilkan kesimpulan bahwa sangat setuju ditempati oleh kriteria peluang kerja dan sangat tidak setuju ditempati oleh kriteria media.

Dari kesimpulan diatas memungkinkan dengan memasukkan informasi pada kerja lebih cenderung untuk di baca oleh calon mahasiswa baru dan untuk faktor media lebih ditingkatkan kembali.

### 6.2 Saran

Untuk pengembangan system selanjutnya diharapkan dapat dilakukan pengembangan system dengan metode pengelompokkan lainnya dan dapat menambahkan beberapa kriteria lainnya.

67

# DAFTAR PUSTAKA

Sri, M., Suharyati & Tati, H. (2013). *Faktor-Faktor Yang Berpengaruh Terhadap Keputusan Memilih Program Studi*. Jakarta: Informatika.

Basilia, R.I. (2008). *Faktor-faktor Yang Mempengaruhi Mahasiswa Dalam Memilih Program Studi di Perguruan Tinggi*. Yogyakarta: Ilmu Komputer.

Binus, S. (2017). Clustering. Retrieved November 1, 2019, from

https://socs.binus.ac.id/2017/03/09/clustering/ Ikbal. (2011). *Minat*. *53*(9), 1689–1699.

https://doi.org/10.1017/CBO9781107415324.004

Edy Irwansyah, M. F. (2015). *Advanced Clustering: Teori dan Aplikasi*.

Pahlevy. (2010). *Pengertian Flowchart dan definisi data*. Retrieved from http://www.landasanteori.com/2015/10/pengertian-flowchart-dan definisidata.html

Bunafit, N. ( 2004). *PHP dan MySQL dengan editor Dreamweaver MX. ANDI Yogyakarta, Yogyakarta.*

Hakim, L. (2010). Membangun. *Web Berbasis PHP dengan*

*Codeigniter.Yogyakarta : Lokomedia.*

A.S., Rosa dan Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak Terstruktur* *Dan Berorientasi Objek. Informatika*. Bandung.

Bagus, W., Alexius, E. B., & Anggri, S. W. (2019). *Implementasi Metode KMedoids Clustering Untuk Mengetahui Pola Pemilihan Program Studi Mahasiswa Baru Tahun 2018 Di Universitas Kanjuruhan Malang*.

Irwansyah, Edy, S.T., M.SI. 2017. Clustering, (Online),

[(http://socs.binus.ac.id/2017/03/09/clustering/),](http://socs.binus.ac.id/2017/03/09/clustering/)) diakses pada 7 Februari 2019.

Kumari, Archana, Bhagat, Hritu. *Compression* *Record Based Effecient kMedoid Algorithm* *Increase Scalability dan Effeciency*. International Journal of Advanced Researchin Computer Engineering and Technology(IJARCET), volume 2 issue 8, August 2013.

68

69

Boone Jr, Harry N, and Deborah A. Boone. 2012. Analyzing Likert Data. *Journal of Extension* 50 (2)

Maulana, D. (2011). Tutorial Android Penggunaan SQLite Sebagai Database Lokal.