IMPLEMENTASI METODE BALANCE K-MEANS UNTUK PEMETAAN MAHASISWA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :

SURYANTI INDAHSARI

NIM. 115060807111034

KEMENTRIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI

PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER

PROGRAM TERKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2015

LEMBAR PERSETUJUAN
IMPLEMENTASI METODE BALANCE K-MEANS UNTUK PEMETAAN MAHASISWA

Diajukan untuk memenuhi persyaratan

Memperoleh gelar Sarjana Komputer

Disusun oleh :

SURYANTI INDAHSARI

NIM : 115060807111034

Skripsi ini telah disetujui oleh dosen pembimbing pada

tanggal 4 Juli 2015

Dosen Pembimbing I	**Dosen Pembimbing II**
Rekyan Regasari MP, ST., MT.	**Drs. Achmad Ridok, M. Kom.**
NIK. 770414 06 1 2 0253	**NIP. 19680825 199403 1 002**

LEMBAR PENGESAHAN

IMPLEMENTASI METODE BALANCE K-MEANS UNTUK PEMETAAN MAHASISWA

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh:

Suryanti Indahsari

NIM. 115060807111034

Skripsi ini telah diuji dan dinyatakan lulus pada 4 Agustus 2015.

Dosen Penguji I	Dosen Penguji II
Budi Darma Setiawan, S.Kom., M.Cs	**Wayan Firdaus Mahmudy, S.Si.,MT.Ph.D**
NIP. 1984 1015 201 404 1002	**NIP. 1972 0919 199 702 1001**

Dosen Penguji III

Mochammad Hannats Hanafi I., S.ST, M.T

NIK. 201405 881229 1 1 001

Mengetahui

Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, MT.

NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang,
Mahasiswa,

Suryanti Indahsari
NIM. 115060807111034

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, karena atas limpahan rahmat, dan hidayah-Nya, sehingga penulis bisa menyelesaikan proposal skripsi ini dengan sebaik-baiknya. Penulis menyadari sepenuhnya bahwa dalam menyelesaikan proposal skripsi ini, penulis mendapatkan banyak bantuan dari semua pihak, untuk itu dalam kesempatan ini penulis mengucapkan terima kasih kepada:

1. Rekyan Regasari Mardi Putri, S.T., M.T. selaku Dosen Pembimbing I yang telah memberikan bimbingan, arahan dan motivasi hingga skripsi ini dapat terselesaikan.
2. Achmad Ridok. Drs., M.Kom. selaku Dosen Pembimbing II yang telah memberikan pengetahuan, bimbingan dan arahan untuk kesempurnaan penulisan skripsi ini.
3. Seluruh dosen pengajar dan karyawan Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Tarman dan Ibu Welas Rahayu selaku orangtua penulis yang selalu memberikan dukungan, nasihat dan doa.
5. Wahyu Krisnawati, Rezi Arista dan Agung Indu Antoro selaku saudara penulis yang selalu memberikan semangat.
6. Angga Iluk, Nanang A, Fadhila K merupakan sahabat terbaik yang selama ini telah mau susah senang bersama penulis selama menjalani kuliah .
7. Teman-teman Program Teknologi Informasi dan Ilmu Komputer angkatan 2011 yang selalu memberikan motivasinya.
8. Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu dalam terselesaikannya proposal skripsi ini.

Penulis menyadari bahwa skripsi ini tidak sempurna dan tidak luput dari kesalahan, sehingga penulis menerima apabila terdapat kritik dan saran. Semoga proposal skripsi ini dapat bermanfaat.

Malang, 4 Juli 2015

Penulis

ABSTRAK

Suryanti Indahsari, 2015 : Implementasi Metode Balance K-Means untuk Pemetaan Mahasiswa. Skripsi Program Studi Informatika /Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing : Rekyan Regasari MP, ST., MT.; Drs. Achmad Ridok, M. Kom.

Di dalam suatu fakultas pencarian informasi tentang keberhasilan studi mahasiswa sangat penting dilakukan untuk bahan pertimbangan dalam penentuan kebijakan untuk meningkatkan kualitas mahasiswa. Salah satu cara untuk mempermudah pencarian informasi tersebut dengan melakukan pemetaan mahasiswa. Pemetaan mahasiswa adalah suatu proses pengelompokkan mahasiswa untuk mendapatkan peta/gambaran kondisi dari mahasiswa berdasarkan beberapa parameter, sehingga dapat diketahui mahasiswa mana yang masuk ke dalam mahasiswa baik dan kurang. Proses pemetaan mahasiswa ini dapat diselesaikan dengan menggunakan metode clustering, yaitu metode Balance K-Means. Balance K-Means merupakan metode yang dikembangkan untuk mengatasi kelemahan K-Means yaitu menghiraukan data yang bernilai kecil sehingga dalam perhitungan Balance K-Means selalu dilakukan normalisasi pada perhitungan awal. Dalam penelitian ini menggunakan metode Balance K-Means, metode ini dimplementasikan untuk memetakkan 110 data mahasiswa dengan menggunakan 6 parameter yaitu : IP1, IP2, IP3, IP4, Prestasi serta Aktivitas. Berdasarkan pengujian yang dilakukan, didapatkan hasil pemetaan terbaik yaitu dengan menggunakan 5 cluster dan 5 parameter yaitu IP1, IP2, IP3, IP4 dan aktivitas. Hasil pemetaan tersebut memiliki nilai kualitas cluster yang dihitung dengan metode Silhoutte Coefficient sebesar 0.316188 (65.81%) dan 5 parameter yaitu 0.35079 (67.53%).

Kata kunci : Pemetaan Mahasiswa, *Balance K-Means*, *Silhoutte Coefficient*

ABSTRACT

Suryanti Indahsari, 2015 : ***Implementation of Balance K-Means Method for Mapping Students. Thesis Study Program Informatics / Computer Science Faculty, University of Brawijaya, Malang.***

Advisors* : *Rekyan Regasari MP, ST., MT.; Drs. Achmad Ridok, M. Kom.

In a faculty searching information about the success of a student's study is very important to do to material consideration in the determination of policies to improve the students quality. One way to simplify searching information by mapping the student. Student mapping is a process of grouping students to get a map / picture of the students condition based on several parameters, so that students can know where that goes into a good and less student. Student mapping process can be completed by using a clustering method, the method of Balance K-Means. Balance K-Means is a method that was developed to overcome the shortcoming of K-Means, ignoring small valued data , so that in Balance K-Means calculation always be normalized on preliminary calculation. In this study using the K-Means Balance method, this method is implemented to map 110 students data using six parameters, those are: IP1, IP2, IP3, IP4, Achievements and Activities. Based on testing performed, the best mapping results obtained by using 5 clusters and 5 parameters is IP1, IP2, IP3, IP4 and Activities. The mapping results have cluster strength values calculated by the method Silhoutte Coefficient of 0.316188 (65.81%) and 5 parameters is 0.35079 (67.53%).

Keyword :*Student Maping, Balance K-Means, Silhoutte Coefficient*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR PERSAMAAN

DAFTAR SOURCECODE

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam bagian pembelajaran di suatu universitas, fakultas adalah gardu terdepan yang berandil besar dalam menonjolkan sosok mahasiswa teladan. Evaluasi mahasiswa digunakan untuk mengetahui pencapaian hasil studi, evaluasi juga perlu diadakan guna melihat dan meningkatkan kinerja fakultas.

Untuk meningkatkan kinerja tersebut universitas telah memiliki sebuah sistem yang bernama "SIMPEL" yaitu Sistem Pelaporan. Di dalam sistem ini menyediakan data diri, status, jumlah sks dan nilai hasil studi yang telah ditempuh mahasiswa. Namun data yang terdapat di dalam SIMPEL masih sama seperti data yang ada di dalam SIAKAD. Untuk mensorting data di dalam SIMPEL hanya bisa menggunakan Nomor Induk Mahasiswa. Di dalam sistem ini juga belum menggambarkan kumpulan data mahasiswa secara keseluruhan. Sedangkan dalam menentukan mahasiswa yang masuk di dalam kelompok mahasiswa yang kurang atau baik, dilakukan dengan melihat data per individu. Oleh karena itu perlu adanya sistem yang berfokus pada pemetaan mahasiswa.

Pemetaan mahasiswa ini digunakan sebagai langkah awal untuk mengelompokkan mahasiswa FILKOM dan memudahkan *treatment* data mahasiswa yang nantinya data yang sudah terkelompok tersebut diolah lagi sehingga dapat mempermudah pekerjaan Pimpinan Program dalam penentuan kebijakan.

Salah satu ilmu yang dapat digunakan dalam pengelompokkan data adalah data mining, Data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai database yang besar [RSS-13]. Ilmu di dalam data mining yang di gunakan untuk pengelompokkan adalah *Cluster*ing, *Cluster*ing adalah tugas dari data mining untuk pengelompokkan obyek, sehingga obyek-obyek memiliki satu *cluster* yang mirip satu sama lain. Tujuan

pengelompokan adalah untuk menemukan kualitas tinggi *cluster* sehingga jarak antar *cluster* dimaksimalkan dan jarak intra-*cluster* diminimalkan [MAE-12].

Proses pemetaan mahasiswa ini dapat dilakukan dengan mengimplementasikan metode *cluster*ing, salah satu metode *cluster*ing yang cukup popular adalah *K-Means*. Metode ini banyak digunakan karena implementasinya yang sederhana, dan dapat menangani data dalam jumlah besar serta proses yang relatif singkat [WID-11]. Salah satu kekurangan metode K-Means adalah meghiraukan data yang bernilai kecil [WQZ-14]. Untuk mengatasi kelemahan dari metode *K-Means* telah banyak dikembangkan metode baru, salah satunya adalah Algoritma *Balance K-Means.*

Algoritma *Balance K-Means* adalah suatu metode yang menormalisasi semua nilai parameter yang ada di dalam dataset sebelum dilakukan *clustering*. Pada penelitian sebelumnya oleh Hongjun, Jianhuai, Weifan dan Mingwen (2014) dengan judul *"Balance K-means Algorithm"* menunjukkan bahwa dengan menggunakan *Balance K-means Algorithm* memiliki performa yang lebih unggul, lebih cepat konvergen serta waktu kompleksitas yang lebih baik dibanding dengan *K-Means* [WQZ-14].

Berdasarkan latar belakang yang telah dikemukakan di atas, maka topik skripsi yang akan diangkat di dalam penelitian ini adalah pemetaan mahasiswa dengan mengimplementasikan Algoritma *Balance K-Means* dengan judul **"Implementasi metode *Balance K-means* untuk Pemetaan Mahasiswa".**

1.2 Rumusan Masalah

Berdasarkan paparan latar belakang tersebut, maka dapat diambil rumusan masalah sebagai berikut :

1. Apakah Algoritma *Balance K-Means* dapat diimplementasikan untuk pemetaan mahasiswa.
2. Bagaimana kualitas *cluster* yang dihasilkan Algoritma *Balance K-Means* dalam pemetaan mahasiswa.

3. Bagaimana pengaruh parameter dalam Algoritma *Balance K-Means* untuk pemetaan mahasiswa.

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah diuraikan diatas, maka batasan masalah untuk menghindari melebarnya masalah yang akan di selesaikan adalah:

1. Objek studi kasus yang digunakan adalah mahasiswa Fakultas Ilmu Komputer (FILKOM).
2. Parameter yang akan digunakan dalam penelitian ini adalah nilai Indeks Prestasi (IP) mahasiswa dari semester 1 sampai semester 4, prestasi akademik mahasiswa, dan aktivitas mahasiswa.
3. Pemetaan mahasiswa hanya digunakan untuk mengelompokkan data mahasiswa.
4. Bahasa pemrograman yang digunakan dalam pembuatan aplikasi adalah Bahasa pemrograman HTML dan PHP.

1.4 Tujuan

Tujuan yang ingin dicapai di dalam penelitian ini adalah membuat aplikasi pemetaan mahasiswa dengan mengimplementasikan Algoritma *Balance K-Means.*

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah pemetaan mahasiswa merupakan tahapan awal yang dapat digunakan untuk mengelompokkan data mahasiswa, sehingga dari data yang telah terkelompok dapat dilakukan *treatment* yang lebih lanjut untuk dapat membantu Pimpinan Program dalam menentukan kebijakan yang akan di ambil dalam upaya meningkatkan mutu mahasiswa.

1.6 Sistematika Penulisan

Sistematika dalam penulisan laporan penelitian adalah sebagai berikut:

1. BAB I PENDAHULUAN

   Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

   Menguraikan kajian pustaka serta teori-teori yang berhubungan dengan pemetaan mahasiswa menggunakan Algoritma *Balance K-Means*.

3. BAB III METODOLOGI PENELITIAN DAN PERANCANGAN

   Menjelaskan tentang tahapan-tahapan yang dilakukan pada saat penelitian serta rancangan aplikasi yang akan dibuat dalam penelitian ini.

4. BAB IV IMPLEMENTASI

   Berisi tentang implementasi dan pembahasan aplikasi yang dibuat dalam penelitian ini.

5. BAB V PENGUJIAN DAN ANALISIS

   Berisi tentang pengujian serta analisis terhadap aplikasi yang telah dibuat dalam penelitian ini.

6. BAB VI PENUTUP

   Merupakan bab terakhir yang berisi kesimpulan dan saran.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini berisi kajian pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi. Kajian pustaka membahas penelitian sebelumnya yang telah ada. Dasar teori membahas teori yang diperlukan untuk menyusun penelitian yang diusulkan. Beberapa dasar teori yang dimaksud adalah pemetaan mahasiswa, data mining, *clustering* dan Algoritma *Balance K-Means.*

2.1 Kajian Pustaka

Kajian pustaka pada penelitian ini membahas tentang penelitian sebelumnya mengenai pengembangan Algoritma *Balance K-Means.*

Pada penelitian sebelumnya yang dilakukan oleh Hongjun, Jianhuai, Weifan dan Mingwen (2014) dengan judul *"Balance K-means Algorithm"* menunjukkan bahwa Algoritma *Balance K-Means* memiliki performa yang lebih baik dibandingkan dengan *K-Means* [WQZ-14], metode ini bertujuan untuk mengatasi kelemahan *K-means* yang menghiraukan data yang bernilai kecil. Metode ini di implementasikan pada beberapa dataset, yaitu data set iris, wdbc, ionosphere, bupa, pima, wine dan balance. Hasil penelitian dari beberapa dataset tersebut menunjukkan bahwa *Time-consumed* dari Algoritma *Balance K-Means* lebih rendah, nilai max dan nilai *average* yang lebih tinggi [WQZ-14].

2.2 Pemetaan Mahasiswa

Berdasarkan hasil wawancara dengan Ketua Gugus Jaminan Mutu (GJM) di Fakultas Ilmu Komputer Universitas Brawijaya seperti terlampir pada lampiran 1, pemetaan mahasiswa adalah proses pengelompokan mahasiswa untuk mendapatkan peta/gambaran mahasiswa sesuai dengan kelompok/*cluster* yang telah ditetapkan. Ide dasar pada pemetaan mahasiswa adalah untuk mengelompokkan data mahasiswa yang nantinya dari data tersebut dapat dilakukan *treatment* yang lebih lanjut untuk

membantu pimpinan program dalam menentukan kebijakan yang akan diambil dalam upaya peningkatan mutu mahasiswa.

Pada penelitian ini menggunakan 6 parameter yaitu : IP semester 1- 4, prestasi, dan aktivitas mahasiswa, evaluasi akan dilakasanakan pada mahasiswa semester 5 (angkatan 2012). Dari 6 parameter tersebut akan diberikan bobot yang berbeda sesuai dengan tingkatan parameter. Sedangkan evaluasi dilakukan pada angkatan 2012 dikarenakan, pada saat semester 5 merupakan semester yang tepat untuk dilakukan pemetaan, pada semester ini kinerja dari mahasiswa sudah dapat dilihat, seperti mengikuti organisasi, komunitas dan mengambil matakuliah pilihan.

Untuk penjabaran dari 6 parameter tersebut adalah :

1. IP (Indeks Prestasi)

Indeks Prestasi adalah nilai keberhasilan studi dari seorang mahasiswa yang dinyatakan dengan nilai kredit rata-rata yang merupakan satuan nilai akhir yang menggambarkan mutu penyelesaian suatu program studi.

Nilai Indeks Prestasi dapat dihitung dengan menggunakan persamaan 2.1 berikut [PTI-12] :

$$IP = \frac{\sum_{i=1}^{n} K_i Na_i}{\sum_{i=1}^{n} K_i} \qquad (2.1)$$

Keterangan :
IP = Indeks Prestasi
n = Banyaknya mata kuliah
K = Nilai kredit mata kuliah

2. Prestasi

Prestasi adalah hasil yang dicapai mahasiswa dalam upaya yang telah dilakukan melalui karya-karya positif yang telah diciptakan, baik secara individu maupun kelompok. Prestasi mahasiswa juga dapat diartikan sebagai hasil pencapaian mahasiswa dalam mengikuti berbagai ajang perlombaan, baik di tingkat internal, lokal, regional, nasional, maupun internasional. Keikutsertaan mahasiswa dalam ajang perlombaan tersebut disebut juga sebagai suatu prestasi.

3. Aktivitas

Aktivitas mahasiswa merupakan kegiatan mahasiswa untuk mengembangkan kemampuan *(soft skill)* dan mengembangkan potensi diri. Aktivitas mahasiswa dalam bidang akademik yaitu menjadi asisten praktikum atau menjadi asisten dosen. Sedangkan aktivitas mahasiswa dalam bidang non akademik dapat berupa partisipasi mahasiswa dalam kepanitian, komunitas, UKM ,mengikuti keorganisasian di dalam kampus maupun luar kampus.

2.3 Data Mining

Data mining adalah proses untuk menemukan hubungan yang berarti, pola dan kecenderungan dengan memeriksa sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika [LAR-05:21]. Definisi lain menyebutkan bahwa data mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi, mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* yang besar [RSS-13]. Berry dan Linoff dalam [LAR-05:4] menyebutkan bahwa data mining adalah proses *eksplorasi* dan *analisis*, dengan otomatis atau semi-otomatis yang berarti, data dalam jumlah besar digunakan untuk menemukan pola dan aturan yang berarti.

Dari definisi-definisi yang telah disampaikan diatas, terdapat beberapa poin yang terkait dengan data mining yaitu:

1. Data yang diproses di dalam data mining berupa data yang sangat besar.
2. Tujuan dari data mining adalah untuk menemukan hubungan atau pola yang dapat memberikan informasi yang bermanfaat.

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu :

1. Deskripsi : menggambarkan pola dan kecenderungan yang terdapat dalam suatu data. Deskripsi yang berkualitas tinggi sering dapat dicapai dengan menganalisis

data eksplorasi, metode grafis mengeksplorasi data yang bertujuan untuk mencari pola dan tren.

2. Perkiraan/ Estimasi : estimasi hampir sama dengan klasifikasi, perbedaannya pada variabel target. Variabel target estimasi lebih ke arah numerik, bukan kategori. Contoh dari penelitian yang menggunakan estimasi ini adalah memperkirakan nilai rata-rata (IPK) mahasiswa pascasarjana, berdasarkan IPK sarjana siswa.
3. Ramalan/Prediksi : prediksi hampir sama dengan klasifikasi, perbedaannya pada prediksi ini bertujuan untuk memperkirakan (*predicate*) nilai dari hasil yang akan datang. Contoh dari prediksi adalah prediksi presentase kenaikan kematian pengguna lalu lintas tahun depan jika batas kecepatan meningkat.
4. Klasifikasi : di dalam klasifikasi terdapat target variabel kategori. Contoh dari penelitian yang menggunakan klasifikasi adalah mengklasifikasikan orang yang dikatakan mampu dengan melihat karakteristik yang terdapat pada orang tersebut seperti usia, jenis kelamin dan pekerjaan.
5. Pengelompokan/*Clustering* : mengelompokkan *record* atau *dataset* ke dalam kelompok-kelompok yang memiliki kemiripan. Perbedaan mendasar dengan klasifikasi adalah tidak ada variabel target dalam pengelompokkan (*cluster*ing). Di dalam klasifikasi, variabel target telah ditentukan pada awal proses. Sedangkan pengelompokkan, mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), yang mana kemiripan *record* dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal. Contoh penelitian yang menggunakan *clustering* adalah pengelompokkan kode pos di Negara AS berdasarkan wilayah dengan jenis gaya hidup.
6. Asosiasi : tugas untuk mengungkap aturan untuk mengukur hubungan antara dua atau lebih attribut yang muncul dalam satu waktu. Di dalam dunia bisnis istilah ini lebih umum disebut dengan keranjang pasar. Contoh penelitian yang menggunakan asosiasi adalah supermarket tertentu mungkin menemukan bahwa dari 1000 pelanggan yang belanja hari kamis malam, 200 membeli popok dan 200 orang yang

membeli pokok 50 diantaranya membeli bir. Dengan demikian, aturan asosiasi akan menjadi "Jika membeli popok, kemudian membeli bir" dengan dukungan dari $\frac{200}{1000} = 20\%$ dan kepercayaan $\frac{50}{200} = 25\%$ [LAR-05:11-17].

2.4 *Cluster*ing

Dalam buku yang berjudul *DISCOVERING KNOWLEDGE IN DATA An Introduction to Data Mining, Dr. Daniel T. Larose (2005)* mendefinisikan *clustering* sebagai upaya mengelompokan *record*, observasi, atau mengelompokkan kelas yang memliki kesamaan objek [LAR-05:147].

Pengklasteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklasteran. Pengklasteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dan variabel target. Akan tetapi, algoritma pengklasteran mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan *(homogen)*, yang mana kemiripan *record* dalam suatu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal. Prinsip dasar untuk mendapatkan *homogen*y atau *heterogen* dapat menggunakan konsep ukuran jarak. Jarak yang dimaksud bisa berarti ukuran jarak kedekatan atau kemiripan *(similarity measure).*

Clustering sering dilakukan sebagai langkah awal dalam proses data mining, dengan *cluster* yang dihasilkan digunakan sebagai masukkan lebih lanjut ke teknik yang berbeda, seperti jaringan saraf [LAR-05:148].

Menurut Tan, dkk dalam Andayani (2007) membagi *cluster*ing dalam dua kelompok, yaitu *hierarchical* dan *partitional clustering*. *Partitional Clustering* disebutkan sebagai pembagian obyek-obyek data ke dalam kelompok yang tidak saling *overlap* sehingga setiap data berada tepat di satu *cluster*. *Hierarchical clustering* adalah sekelopok *cluster* yang bersarang seperti sebuah pohon berjenjang (*hirarki*) [AND-07].

William (2005) dalam Andayani (2007) membagi algoritma *cluster*ing ke dalam kelompok besar seperti berikut [AND-07]:

a. *Partitioning algorithms* : algoritma dalam kelompok ini membentuk bermacam partisi dan kemudian mengevaluasinya dengan berdasarkan beberapa kriteria.
b. *Hierarchy algorithms* : pembentukan dekomposisi hirarki dari sekumpulan data menggunakan beberapa kriteria.
c. *Density-based* : pembentukan *cluster* berdasarkan pada koneksi dan fungsi densitas.
d. *Grid-based* : pembentukan *cluster* berdasarkan pada koneksi dan fungsi densitas.
e. *Model-based* : sebuah model dianggap sebagai hipotesa untuk masing- masing *cluster* dan model yang baik dipilih diantara model hipotesa tersebut [AND-07]:.

2.5 K-Means

K-Means adalah suatu metode penganalisaan data atau metode data mining yang melakukan proses pemodelan tanpa supervisi (*unsupervised*) dan merupakan salah satu metode yang melakukan pengelompokan data dengan sistem partisi [AIF-10]. Metode *K-Means* berusaha mengelompokkan data yang ada ke dalam beberapa kelompok, dimana data dalam satu kelompok mempunyai karakteristik yang sama satu sama lainnya dan mempunyai karakteristik yang berbeda dengan data yang ada di dalam kelompok yang lain. Dengan kata lain, metode ini berusaha untuk meminimalkan variasi antar data yang ada di dalam suatu *cluster* dan memaksimalkan variasi dengan data yang ada di *cluster* lainnya.

Algoritma *K-Means* berfungsi untuk mengelompokkan suatu obyek yang memiliki kesamaan (proses pengelompokan biasa disebut *clustering*) dengan berdasar K *cluster*, dimana K adalah bilangan *integer* positif.

Langkah awal proses algoritma *K-Means* yaitu menentukan pusat dari tiap *cluster* yang hampir sejenis yang kemudian disebut *centroid*. *Centroid* bisa ditentukan secara acak. Lalu lakukan penghitungan jarak antara tiap *cluster* terhadap *centroid* yang ada, kemudian kelompokkan tiap *cluster* berdasar jarak terdekat dari tiap obyek terhadap *centroid*. Kemudian hitung kembali *centroid*, lakukan ini berulang-ulang sampai posisi

centroid tidak berpindah lagi. Berikut ini merupakan langkah-langkah perhitungan *K-Means* :

1. Tentukan nilai k sebagai jumlah *cluster* yang ingin dibentuk.
2. Bangkitkan k *centroid* (titik pusat *cluster*) awal secara *random*. Penentuan *centroid* awal dilakukan secara *random* dari data yang tersedia sebanyak k *cluster*, kemudian untuk menghitung *centroid cluster* ke-i berikutnya, dapat dihitung dengan persamaan 2.2 berikut ini :

$$v = \frac{\sum_{i=1}^{n} x_i}{n} \; ; i = 1,2,3, \dots n \quad (2.2)$$

Keterangan :

v : *centroid* pada *cluster*

x_i : data i

n : banyaknya data/jumlah data yang menjadi anggota *cluster* [ENM-13].

3. Hitung jarak setiap data ke masing-masing *centroid* menggunakan *Euclidean Distance* . Perhitungan jarak menggunakan *euclidean distance* karena *euclidean distance* sering digunakan pada perhitungan *k-means*. Persamaannya dapat dilihat pada 2.3 berikut ini:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_{i-} y_i)^2} \quad ; i = 1,2,3, \dots . n \quad (2.3)$$

Keterangan:

x_i : data x_i

y_i : pusat *cluster i*

n : banyaknya data [ENM-13].

4. Kelompokkan setiap data berdasarkan jarak terdekat *(Nearest Distance)* antara data dengan *centroid*-nya.

5. Tentukan posisi *centroid* baru dengan cara menghitung nilai rata-rata dari data-data yang ada pada *centroid* yang sama. Perhitungan untuk menentukan *centroid* baru dapat dilihat pada persamaan 2.2.
6. Kembali ke langkah 3 jika posisi *centroid* baru dengan *centroid* lama tidak sama.

K-Means merupakan metode *clustering* yang paling terkenal dan banyak digunakan di berbagai bidang karena sederhana, mudah di implementasikan, memiliki kemampuan untuk mengklaster data yang besar, mampu menangani data *outlier*. *K-Means* merupakan metode pengklasteran secara *partitioning* yang memisahkan data ke dalam kelompok yang berbeda. Dengan *partitioning* secara iterative, *k-means* mampu meminimalkan rata-rata jarak setiap data ke *cluster*-nya. Metode ini dikembangkan oleh Mac Queen pada tahun 1967 [AIF-10].

Menurut Wang Hongjun salah satu kekurangan metode *K-Means* adalah menghiraukan data yang bernilai kecil. Apabila menggunakan algoritma *K-Means* sebagai algoritma *custering*, nilai yang paling besar akan lebih besar berperan pada hasil *clustering*, dan nilai yang kecil akan dihiraukan.[WQZ-14].

2.6 Balance K-means Algorithm

Algoritma *Balance K-Means* merupakan salah satu pengembangan dari *K-means Algorithm*. Metode ini dikembangkan guna mengatasi kelemahan *K-means* yang menghiraukan data yang bernilai kecil. Metode ini memiliki performa yang lebih unggul, lebih cepat konvergen serta waktu kompleksitas yang lebih baik dibanding dengan *K-Means* [WQZ-14]. Perbedaan yang terdapat di dalam metode *Balance K-Means* dan *K-Means* adalah pada metode *Balance K-Means* pada data awal selalu dilakukan normalisasi untuk menyamakan nilai *range* yang terdapat pada data. Selain itu pada metode *Balance K-Means* terdapat perhitungan *error function*, perhitungan ini digunakan untuk mengetahui kualitas dari *cluster* yang telah terbentuk pada tiap iterasi.

Algoritma dari metode Algoritma *Balance K-Means* dapat dijabarkan sebagai berikut [WQZ-14]:

1. Perulangan i=1 sampai n.

Keterangan; *i* : Data yang terdapat di dalam dataset.

2. Lakukan normalisasi pada semua data menggunakan *min max normalization. Normalisasi* digunakan untuk menyamakan nilai *range* yang terdapat pada data. *Min-max normalization* merupakan metode normalisasi data yang bekerja dengan melihat seberapa besar nilai data (*max*) daripada nilai minimum data (*min*) dan melakukan skala dengan range yang berbeda [LAR-05:36]. Perhitungan *Min-max normalization* dapat dilihat pada persamaan 2.4 berikut ini :

$$x_i = \frac{x_i - MIN\,(f)}{MAX\,(f) - MIN\,(f)} \qquad (2.4)$$

Keterangan;

x_i : Data i

$MIN\,(f)$: Nilai minimum pada seluruh dataset

$MAX\,(f)$: Nilai maksimum pada seluruh dataset

3. Ulangi langkah 1 sampai semua data ternormalisasi.
4. Inisialisasi *k centroid* (wi,…., wk) secara *random* pada x_i,$j \in (1,k)$; $i \in (1,n)$
5. Masing-masing *cluster cj* diasosiasikan dengan *centroid*.
6. Ulangi
7. Perulangan pada vector xi, dimana $i \in 1,..n$
8. Lakukan perulangan
9. Hitung jarak data dengan *centroid* mengunakan jarak *manhattan.* Perhitungan ini menggunakan jarak *manhattan* karena kemampuannya dalam mendeteksi keadaan khusus seperti keadaan *outliers* dengan lebih baik. Perhitungan jarak *manhattan* dapat dilihat pada persamaan 2.5. Setelah mendapatkan jarak tetapkan x_i ke dalam *cluster cj* dengan *centroid* terdekat w_{j*}, dimana $w_{j*}, |x_i - w_{j*}| \leq |x_i - w_j|, j \in 1, \ldots k$

$$d_{(i,j)} = \sum_{k=1}^{n} |x_{ik} - y_{jk}| \qquad (2.5)$$

Keterangan:

x_{ik} : data x_i

y_{jk} : pusat *cluster*

n : banyaknya data

10. Untuk masing-masing *cluster* cj, dimana $j \in 1, \ldots k$ lakukan.
11. *Update Centroid* $= \sum_{xi \in cj} \frac{xl}{|cj|}$ (2.6)

    Keterangan:

    x_i : Data i

    cj : *Cluster* j
12. Hitung nilai *error function* menggunakan; $e = \sum_{xl \in cj} |x_i - w_j|^2$ (2.7)

    Keterangan:

    e : Error Function

    x_i : Data *i*

    c_j : *Cluster* j

    w_j : Pusat *cluster* j
13. Apabila tidak ada lagi perubahan pada tiap anggota *cluster* hentikan perhitungan.

2.7 Silhoutte Coefficient

Silhoutte Coefficient merupakan metode yang digunakan untuk melihat kualitas *cluster*, seberapa baik suatu objek ditempatkan dalam suatu *cluster*. Metode ini merupakan metode validasi *cluster* yang menggabungkan metode *cohesion* dan *separation*. Metode ini mengacu pada penafsiran dan validasi kelompok data. [HRM-14]

Tahapan perhitungan *Silhouette Coefficient* adalah sebagai berikut [HRM-14]:

1. Untuk setiap objek i, hitung rata-rata jarak dari data i dengan seluruh data yang berada dalam satu *cluster*. Akan didapatkan nilai rata-rata yang disebut $a(i)$.

$$a(i) = \frac{\sum d(i,j), j \in A, j \neq i}{|A|-1} \quad (2.8)$$

Keterangan:

$a(i)$ = rata-rata jarak antara data *i* dengan semua data lain dalam satu *cluster*

j = data lain dalam satu *cluster* A

$|A|$ = jumlah data dalam *cluster* A

2. Untuk setiap data i, hitung rata-rata jarak dari data i dengan data yang berada di *cluster* lainnya. Dari semua jarak rata-rata tersebut ambil nilai yang paling kecil. Nilai ini disebut $b(i)$.

$$d(i, C) = \frac{\sum d(i, j), j \epsilon C}{|C|}$$

dengan $d(i, C)$ adalah jarak rata-rata dokumen i dengan semua objek pada cluster lain C dimana A ≠ C.

$$b(i) = \min d(i, C),\ C \neq A \qquad (2.9)$$

3. Maka untuk data i memiliki nilai *silhoutte coefisien* :

$$S_i = (b_i - a_i) / \max(a_i, b_i) \qquad (2.10)$$

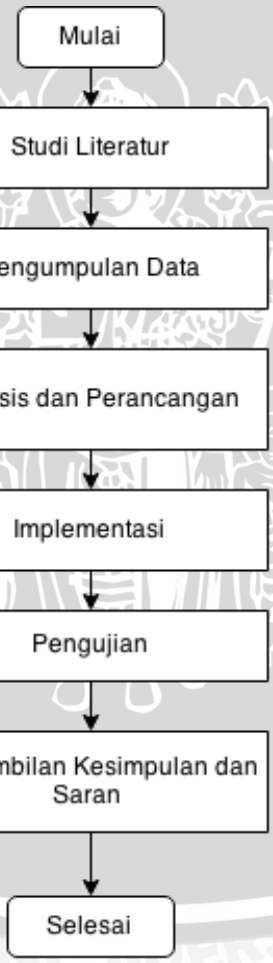
Hasil perhitungan nilai *silhoutte coeffisien* dapat bervariasi antara -1 hingga 1. Hasil clustering dikatakan baik jika nilai *silhoutte coeffisien* bernilai positif ($a_i < b_i$) dan a_i mendekati 0, sehingga akan menghasilkan nilai *silhoutte coeffisien* yang maksimum yaitu 1 saat a_i= 0. Maka dapat dikatakan, jika S_i = 1 berarti data i sudah berada dalam *cluster* yang tepat. Jika nilai S_i = 0 maka data i berada di antara dua *cluster* sehingga data tersebut tidak jelas harus dimasukan ke dalam *cluster* A atau *cluster* B. Akan tetapi, jika S_i = -1 artinya struktur *cluster* yang dihasilkan *overlapping*, sehingga data i lebih tepat dimasukan ke dalam *cluster* yang lain, sehingga pengelompokkan data dapat dikatakan buruk. Nilai rata-rata *silhoutte coeffisien* dari tiap data dalam suatu *cluster* adalah suatu ukuran yang menunjukan seberapa ketat data dikelompokan dalam *cluster* tersebut [MSC-11].

Berikut adalah nilai *silhoutte* berdasarkan Kaufman dan Rousseeuw [MSC-11]:

0.7 < SC <= 1	*Strong Stucture*
0.5 < SC <= 0.7	*Medium Structure*
0.25 < SC <= 0.5	*Weak Structure*
SC <= 0.25	*No structure*

BAB III
METODOLOGI PENELITIAN DAN PERANCANGAN

Pada bab ini akan dibahas metode-metode yang akan digunakan dalam penelitian yaitu studi literatur, metode pengambilan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis serta pengambilan kesimpulan. Tahapan dalam proses penelitian ini dapat dilihat secara jelas pada Gambar 3.1 yang menunjukkian rencana atau struktur penelitian yang akan digunakan untuk memecahkan permasalahan dalam penelitian ini.



Gambar 3.1 Diagram Alir Penelitian

Berdasarkan gambar 3.1 tahapan-tahapan yang dilakukan pada penelitian ini adalah:

1. Melakukan studi literatur untuk memahami teori – teori yang mendukung penelitian seperti teori tentang *Algoritma Balance K-means* dan pemetaan mahasiswa.
2. Mengumpulkan data yang akan digunakan dalam penelitian.
3. Menganalisa dan melakukan perancangan aplikasi pemetaan mahasiswa dengan menggunakan *Algoritma Balance K-means.*
4. Mengimplementasikan rancangan sistem yang dihasilkan menjadi suatu aplikasi pemetaan mahasiswa
5. Melakukan uji coba terhadap aplikasi pemetaan mahasiswa yang dihasilkan.
6. Melakukan evaluasi atau analisis hasil yang diperoleh dari uji coba aplikasi yang telah dilakukan.

3.1 Studi Literatur

Studi literatur adalah proses mempelajari dan memahami secara mendalam terkait teori-teori dan dasar keilmuan yang akan menjadi bahan untuk penelitian yang akan dilakukan. Teori-teori yang dipelajari meliputi data mining, *clustering*, *K-Means*, Algoritma *Balance K-means, Silhoutte Coefficient*. Sumber dari literatur tersebut berasal dari buku, jurnal, *e-book*, wawancara, penelitian sebelumnya, *browsing* dari internet, dan dari sumber pustaka lain yang terkait dengan penelitian.

3.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data mahasiswa semester 5. Data tersebut meliputi: data Indeks Prestasi (IP) mahasiswa dari semester 1 sampai semester 4, data prestasi serta data aktivitas. Data Indeks Prestasi di dapatkan dari data yang terdapat di akademik, sedangkan data prestasi dan aktivitas di dapatkan dari data yang terdapat di kemahasiswaan serta hasil dari pembagian kuisioner yang dibagikan kepada mahasiswa FILKOM angkatan 2012 dan laboratorium yang ada di Fakultas Ilmu Komputer Universitas Brawijaya (FILKOM UB). Laboratorium tersebut meliputi: laboratorium komputasi cerdas dan visualisasi, laboratorium komputer dasar, labolatorium jaringan komputer, laboratorium basis data, laboratorium rekayasa perangkat lunak, laboratorium sistem informasi,

laboratorium robotika. Jumlah data yang digunakan dalam penelitian ini adalah 110 data mahasiswa FILKOM angkatan 2012, data dapat dilihat pada lampiran 3.

3.3 Analisis dan Perancangan

Analisis dan perancangan sistem merupakan tahapan yang dilakukan untuk membuat suatu rancangan aplikasi pemetaan mahasiswa dengan mengimplementasikan metode *Balance K-Means*. Analisis dan perancangan ini terdiri dari formulasi permasalahan, solusi penyelesaian masalah, deskripsi umum sistem, perancangan sistem serta perancangan data.

3.3.1 Formulasi Permasalahan

Pemetaan mahasiswa ini dilakukan dengan menggunakan beberapa parameter, yaitu : Indeks Prestasi mahasiswa semester 1-4, Prestasi dan Aktivitas. Untuk mengelompokkan data dengan beberapa parameter ini tidak mudah dilakukan dengan hanya menggunakan *excel*. Karena untuk 1 data mahasiswa dengan IP bagus dan prestasi/ aktivitas tidak bagus belum tentu mahasiswa tersebut masuk di dalam kelompok yang dikatakan bagus. Contoh lain apabila mahasiswa memiliki IP jelek tetapi prestasi baik dan aktivitas kurang belum tentu juga mahasiswa tersebut dikatakan kurang.

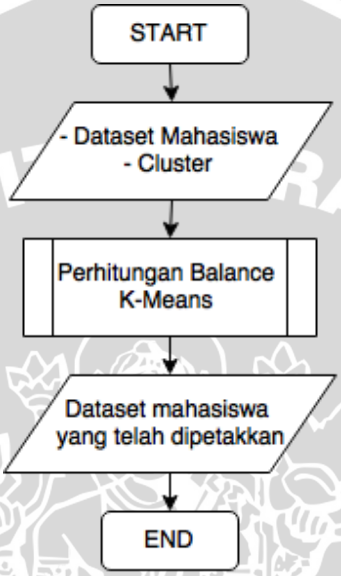
3.3.2 Solusi Penyelesaian Masalah

Solusi untuk masalah pemetaan ini adalah dengan menggunakan suatu metode yang khusus digunakan untuk pengelompokkan. Metode yang biasa di gunakan untuk pengelompokkan adalah *clustering*. Salah satu metode *clustering* yang cukup popular adalah *K-Means*. Metode ini banyak digunakan karena implementasinya yang sederhana, dan dapat menangani data dalam jumlah besar serta proses yang relatif singkat [WID-11].

Menurut Wang Hongjun salah satu kekurangan metode *K-Means* adalah menghiraukan data yang bernilai kecil. Apabila menggunakan algoritma *K-Means* sebagai algoritma *clustering*, nilai yang paling besar akan lebih besar berperan pada hasil *clustering*, dan nilai yang kecil akan dihiraukan.[WQZ-14]. Untuk mengatasi kelemahan dari metode *K-Means* telah banyak dikembangkan metode baru, salah satunya adalah Algoritma *Balance K-Means.*

3.3.3 Deskripsi Umum Sistem

Subbab ini menjelaskan tentang proses-proses yang dilakukan aplikasi untuk memetakan data mahasiswa ke dalam beberapa *cluster* dengan mengimplementasikan Algoritma *Balance K-means*. Secara umum, alur proses dari aplikasi ditunjukkan pada Gambar 3.2 :



Gambar 3.2 Diagram Alir Perancangan Sistem

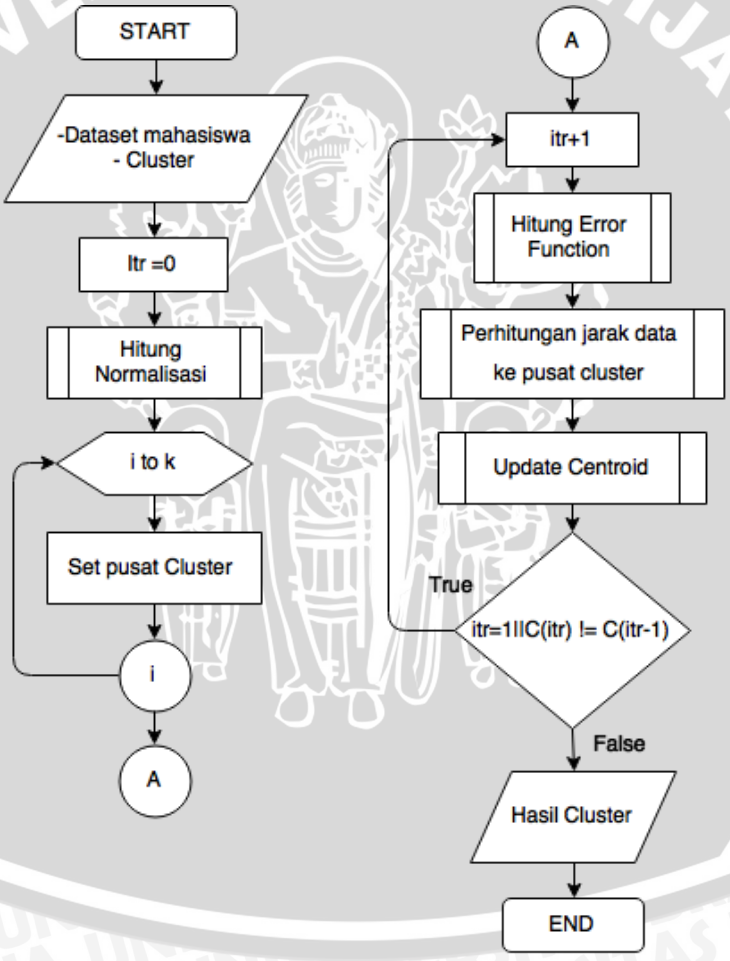
Berdasarkan Gambar 3.2 alur kerja dari sistem ini adalah sebagai berikut :

1. *User* memberikan masukkan ke sistem berupa file data mahasiswa yang memiliki format . *xls* dan juga jumlah *cluster* yang akan dibentuk.
2. Dari data mahasiswa yang telah di *input* akan dilakukan proses normalisasi, sehingga nilai data dapat disetarakan.
3. Tahap selanjutnya dari metode *Balance K-Means* adalah menentukan pusat awal *cluster* yang di tentukan secara *random*.
4. Setelah menentukan pusat awal *cluster* secara *random* dari proses *clustering* yang dilakukan aplikasi adalah penentuan *cluster* untuk setiap data. Penentuan *cluster* untuk setiap data dilakukan secara berulang sampai didapatkan nilai yang *konvergen*. Proses iteratif ini dilakukan untuk mendapatkan hasil *cluster* yang baik dimana data-data dalam satu *cluster* memiliki nilai kedekatan yang lebih tinggi dibandingkan dengan data pada *cluster* lain.

5. *Output* dari aplikasi ini adalah data mahasiswa yang telah dipetakkan. Data tersebut dapat dilihat, dicetak maupun disimpan oleh *user*.

3.3.4 Perancangan Proses

Subab ini menjelaskan tentang proses pemetaan mahasiswa dengan menggunakan metode *Balance K-Means* secara lebih rinci. Proses pemetaan terdiri dari beberapa proses yaitu proses normalisasi data, proses penentuan *cluster* untuk setiap data, proses *update cluster* dan proses penentuan nilai *Silhoutte Coefficent* data. Secara umum flowchart *Balance K-Means* ditunjukkan pada Gambar 3.3 berikut ini :



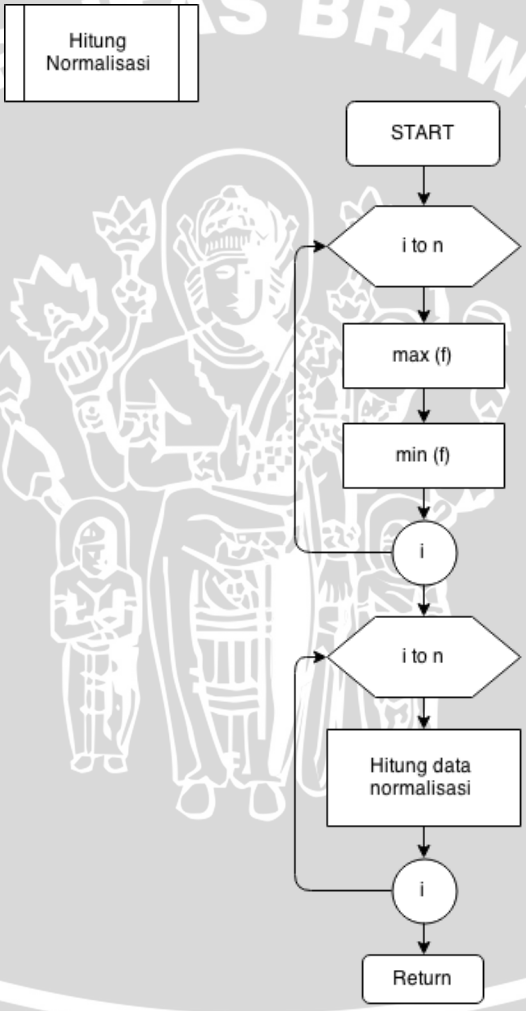
Gambar 3.3 Diagram Alir *Balance K-Means*

3.3.4.1 Normalisasi Data

Proses normalisasi data digunakan untuk menyamakan skala parameter data ke dalam sebuah *range* yang spesifik, misalkan 0 s/d 1. Proses normalisasi dilakukan dengan langkah sebagai berikut :

1. Untuk setiap parameter cari nilai minimun dan nilai maksimum parameter.
2. Untuk setiap data, hitung normalisasi dengan menggunakan rumus *Min-Max Normalization* yang ditunjukkan pada Persamaan (2.4).

Diagram Alir proses normalisasi data ditunjukkan pada Gambar 3.4 berikut:

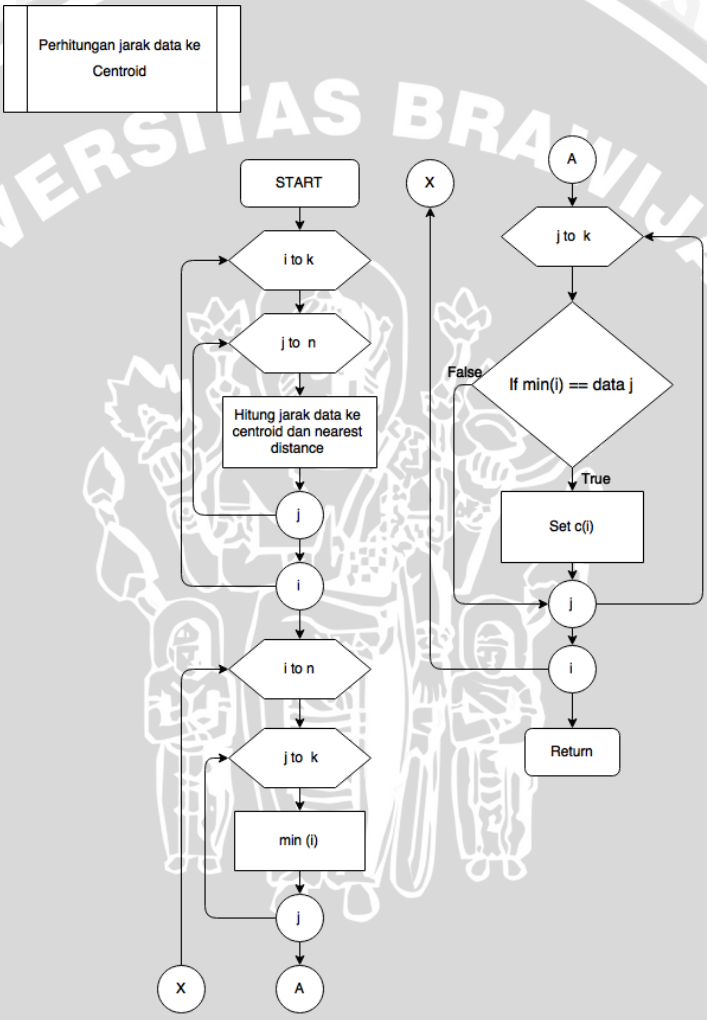


Gambar 3.4 Flowchart Normalisasi Data

3.3.4.2 Perhitungan Jarak ke Pusat Cluster

Setelah dilakukan normalisasi, proses selanjutnya adalah menghitung jarak ke pusat *cluster*/proses penentuan *cluster*, hal yang pertama dilakukan adalah

menentukan pusat awal *cluster*. Penentuan pusat awal *cluster* ini dilakukan secara *random*. Setelah menentukan pusat *cluster* secara *random*. Langkah selanjutnya adalah menghitung jarak data terhadap pusat *cluster* yang menggunakan jarak *Manhattan*. Untuk menentukkan *cluster* akhir persamaan matematikanya ditunjukkan pada persamaan (2.5). Flowchart proses penentuan *cluster* ditujukkan pada Gambar 3.5 berikut :

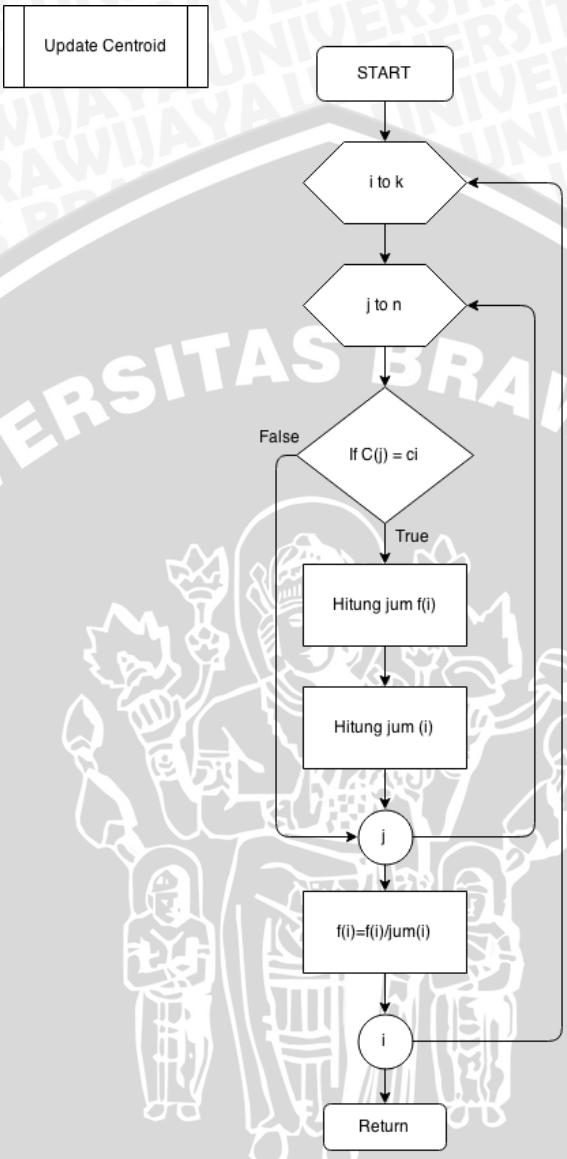


Gambar 3.5 Flowchart Penentuan *Cluster*

3.3.4.3 Update Pusat Cluster Baru

Penentuan pusat *cluster* baru ini dilakukan setelah mendapatkan *cluster*. Proses *update* pusat *cluster* ini dilakukan dengan menjumlahkan nilai data yang sama dengan *cluster* yang telah dibentuk dan dibagi dengan jumlah data pada tiap *cluster* yang terbentuk. Persamaan matematikanya ditunjukkan pada persamaan

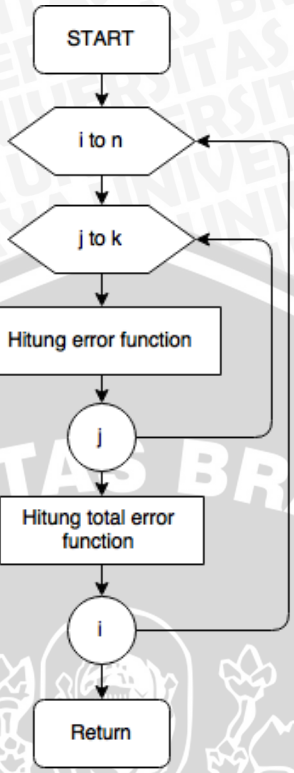
(2.6). Flowchart dari proses penentuan pusat *cluster* baru ditunjukkan pada Gambar 3.6 berikut :



Gambar 3.6 Flowchart Penentuan Pusat *Cluster* Baru

3.3.4.4 Perhitungan Error Function

Setelah mendapatkan *cluster* baru, langkah terakhir dari *Balance K-Means* adalah menghitung nilai *error function*. Nilai *error function* ini digunakan untuk mengetahui kualitas *cluster* setiap iterasi. Persamaan matematikanya ditunjukkan pada persamaan (2.7). Flowchart dari proses perhitungan *error function* ditunjukkan pada Gambar 3.7 berikut :

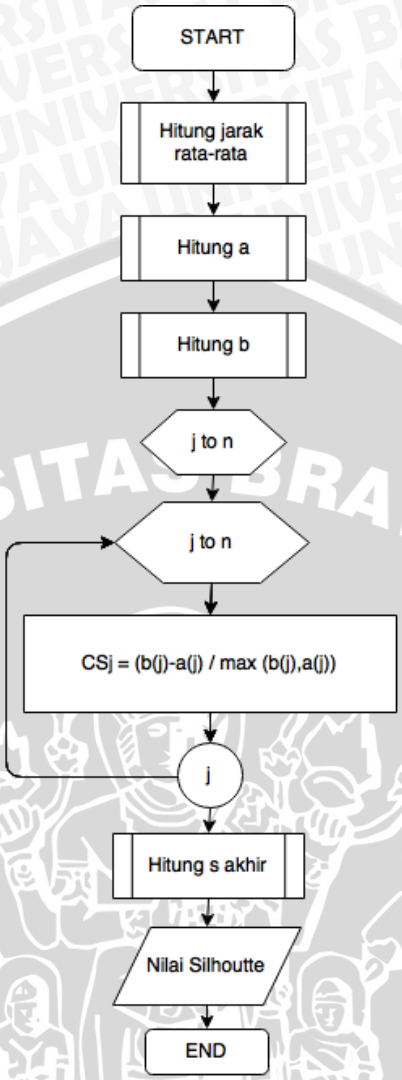


Gambar 3.7 Flowchart Perhitungan *Error Function*

3.3.4.5 Silhoutte Coefficient Data

Sillhouette Coefficient merupakan metode yang digunakan untuk melihat kualitas *cluster* dari hasil pemetaan mahasiswa, seberapa baik suatu objek ditempatkan dalam suatu *cluster*. Oleh karena itu dalam aplikasi pemetaan mahasiswa ini, ditambahkan satu proses untuk menghitung nilai *Sillhouette Coefficient*.

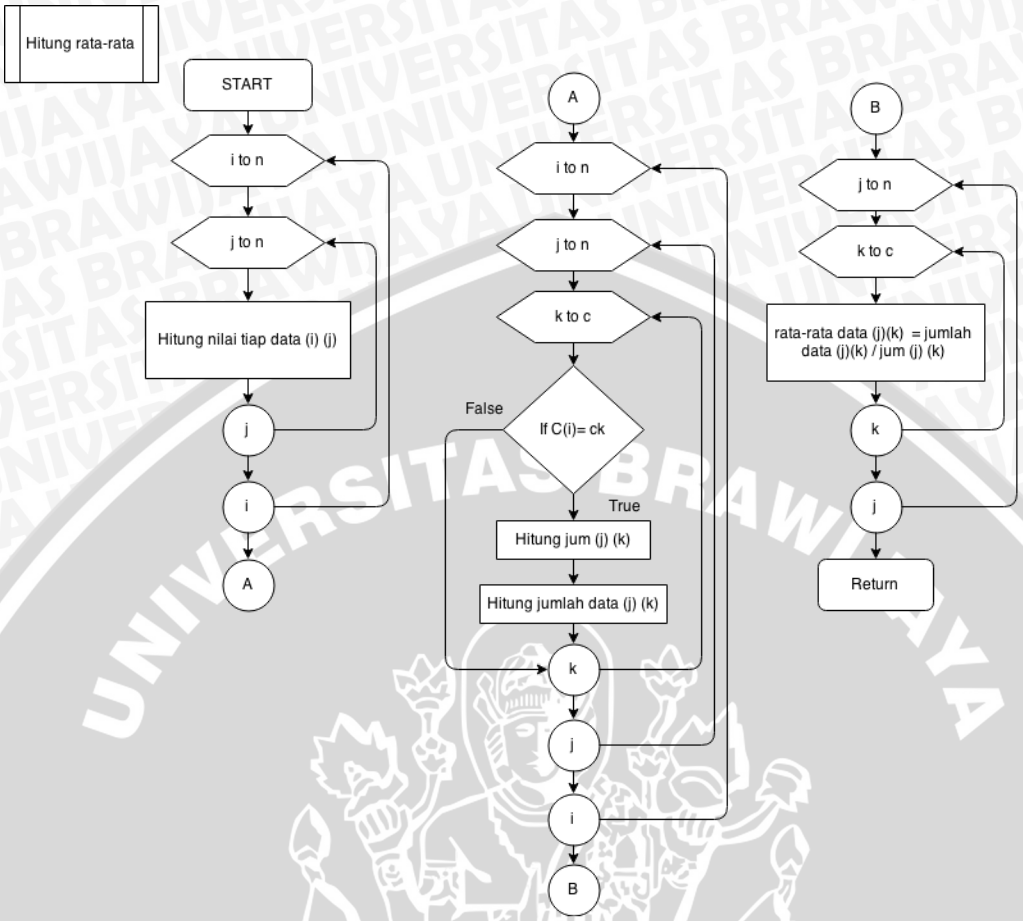
Flowchart dari proses penentuan nilai *Sillhouette Coefficient* data ditunjukkan pada Gambar 3.8 berikut :



Gambar 3.8 Flowchart *Silhoutte Coefficient* Data

3.3.4.6 Hitung rata-rata data

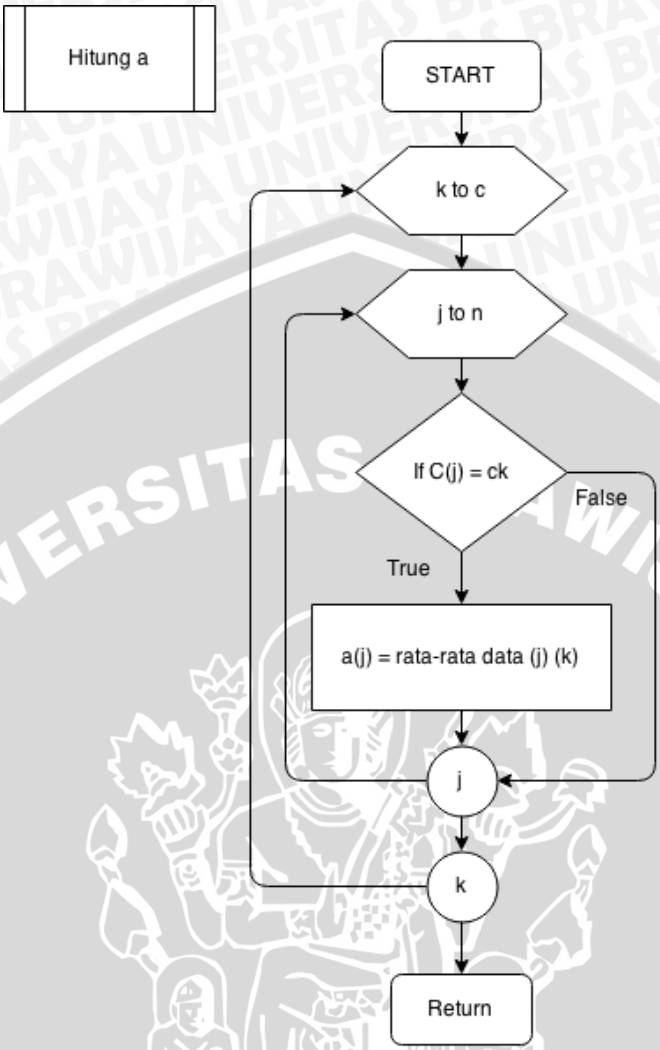
Proses perhitungan untuk mendapatkan nilai rata-rata data adalah proses awal dari *silhouette coefficient* untuk mendapatkan nilai $a(i)$ dan nilai $b(i)$. Proses ini untuk mendapatkan nilai rata-rata jarak suatu data (misalkan *i*) dengan semua data lain yang berada dalam satu *cluster*. Persamaan matematikanya ditunjukkan pada persamaan (2.8). *Flowchart* dari proses perhitungan rata-rata data ditunjukkan pada Gambar 3.9 berikut :



Gambar 3.9 Flowchart menentukan nilai rata-rata

3.3.4.7 Hitung nilai $a(i)$

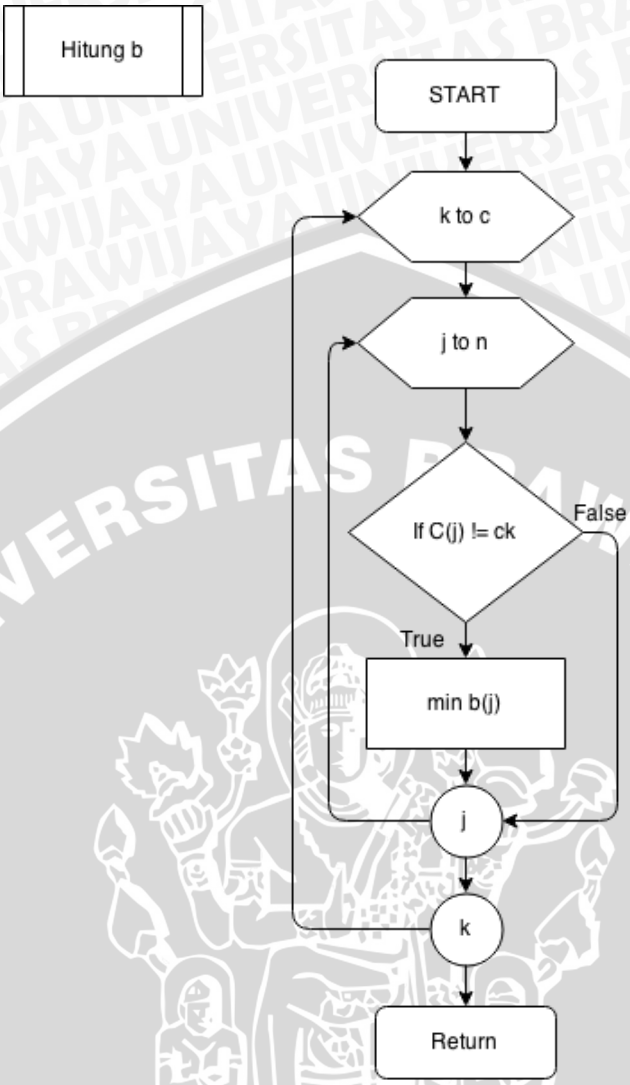
Setelah proses perhitungan untuk mendapatkan nilai rata-rata data, proses selanjutnya adalah proses untuk mendapatkan nilai $a(i)$. Nilai $a(i)$ adalah rata-rata jarak antara data i dengan semua data lain dalam satu *cluster*, persamaan matematikanya ditunjukkan pada persamaan (2.8). Flowchart dari proses perhitungan $a(i)$ ditunjukkan pada Gambar 3.10 berikut :



Gambar 3.10 Flowchart perhitungan $a(i)$

3.3.4.8 Hitung nilai $b(i)$

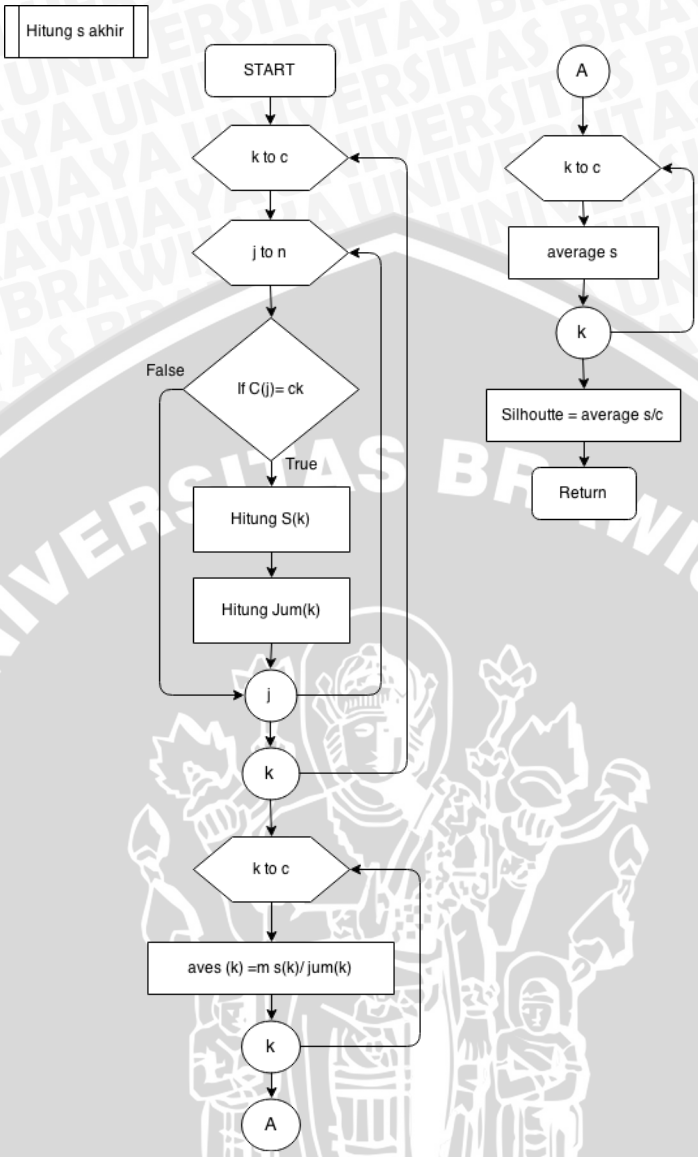
Setelah proses perhitungan untuk mendapatkan nilai $a(i)$, proses selanjutnya adalah proses untuk mendapatkan nilai $b(i)$. Nilai $b(i)$ adalah jarak rata-rata dokumen i dengan semua objek pada *cluster* lain C dimana A $\neq$ C. Persamaan matematikanya ditunjukkan pada persamaan (2.9). Flowchart dari proses perhitungan $b(i)$ ditunjukkan pada Gambar 3.11 berikut :



Gambar 3.11 Flowchart perhitungan $b(i)$

3.3.4.9 Hitung nilai S_i

Setelah proses perhitungan untuk mendapatkan nilai $a(i)$ dan $b(i)$, proses selanjutnya adalah proses untuk mendapatkan nilai *Silhoutte Coefficient*. Persamaan matematikanya ditunjukkan pada persamaan (2.10). Flowchart dari proses perhitungan *Silhoutte Coefficient* ditunjukkan pada Gambar 3.12 berikut :



Gambar 3.12 Flowchart perhitungan *Silhoutte Coefficient*

3.3.5 Perancangan Data

Pada subbab ini menjelaskan perancangan bobot parameter yang digunakan di dalam penelitian, dimana parameter yang digunakan adalah : Indeks Prestasi (IP) semester 1-4, data prestasi dan data aktivitas. Proses penentuan skor (*scoring*) data dilakukan dengan memberikan skor untuk setiap prestasi dan aktivitas sesuai dengan tingkatan-tingkatannya. Penentuan skor data disesuaikan dengan penilaian SKM (Satuan Kegiatan Kemahasiswaan) yang sudah dirancang oleh (Pembantu Rektor III) atau PR III beserta (Pembantu Dekan III) atau PD III berdasarkan hasil

wawancara dan data dari PD III FILKOM yang dapat dilihat pada lampiran 2. Adapun penentuan skor untuk data prestasi dan data aktivitas dapat dilihat pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Bobot Parameter untuk data Prestasi

No.	Tingkat Prestasi	Kategori Prestasi	Point
1	Internasional	Juara	1500
		Masuk Final (Finalis)	1000
		Lolos Babak Penyisihan	750
		Peserta	400
2	Nasional	Juara	1000
		Masuk Final (Finalis)	750
		Lolos Babak Penyisihan	600
		Peserta	300
3	Regional	Juara	750
		Masuk Final (Finalis)	500
		Lolos Babak Penyisihan	400
		Peserta	200
4	Universitas	Juara	500
		Masuk Final (Finalis)	300
		Lolos Babak Penyisihan	200
		Peserta	70
5	Fakultas	Juara	300
		Masuk Final (Finalis)	200
		Lolos Babak Penyisihan	150
		Peserta	50
6	Jurusan	Juara/ Terplih	100
		Peserta	30

Sumber : SKM

Tabel 3.2 Penentuan Bobot Parameter untuk data Aktivitas

No	Aktivitas	Jabatan	Point
1	UKM/Organisasi Kerohanian	Ketua	550
		Pengurus Inti	450
		Pengurus Non Inti	350
		Anggota	100

No	Aktivitas	Jabatan	Point
2	ORMAWA Tk. Universitas	Ketua	550
		Pengurus Inti	500
		Pengurus Non Inti	400
		Anggota	100
3	ORMAWA Tk. Fakultas	Ketua	500
		Pengurus Inti	450
		Pengurus Non Inti	350
		Anggota	100
4	ORMAWA Tk. Prodi/Jurusan	Ketua	450
		Pengurus Inti	400
		Pengurus Non Inti	300
		Anggota	100
5	Panitia Kegiatan Tk.Nasional	Panitia Inti	350
		Anggota	100
6	Panitia Kegiatan Tk.Universitas	Panitia Inti	300
		Anggota	50
7	Panitia Kegiatan Tk.Fakultas	Panitia Inti	250
		Anggota	50
8	Panitia Kegiatan Tk.Prodi/Jurusan	Panitia Inti	200
		Anggota	50
9	Asisten Praktikum dan Project Profit	Asisten	200
		Project Profit	150

Sumber : SKM

Dalam proses penentuan skor data, jika terdapat data yang memiliki lebih dari satu item prestasi dan aktivitas maka skor yang didapatkan adalah penjumlahan skor dari beberapa item yang dimiliki. Contoh proses penentuan skor data dapat dilihat pada Tabel 3.3 berikut :

Tabel 3.3 Contoh Proses *Scoring* Data

Data	Prestasi	Skor	Aktivitas	Skor
1	Comfest (Peserta) + PKM (Peserta)	200+300 = **500**	Asisten + PK2 Maba Fakultas (Peserta)	200+50 = **250**

3.3.6 Perhitungan manual

Subbab ini merupakan penjelasan dari perhitungan manual dari Algoritma *Balance K-means* dan perhitungan manual pengujian *cluster*, dimana jumlah data yang digunakan dalam perhitungan manual ini adalah 20 data yang akan di jadikan 3 *cluster*. Dataset yang digunakan di dalam perhitungan manual ini berupa dataset mahasiswa yang telah diberikan pembobotan pada parameter prestasi, dan aktivitas. Dataset yang digunakan dalam perhitungan manual ini di gambarkan di dalam Tabel 3.4 berikut ini:

Tabel 3.4 Tabel Dataset Mahasiswa yang telah dipilih *random*

No	NIM	IP1	IP2	IP3	IP4	Prestasi	Aktivitas
1	125150100111018	3.32	4.00	3.21	3.63	100	900
2	125150101111001	3.39	3.62	2.83	3.10	0	0
3	125150200111021	3.26	3.37	3.74	3.61	0	700
4	125150200111041	3.21	3.75	3.52	3.20	1350	450
5	125150200111042	3.05	3.62	3.36	3.67	100	900
6	125150200111049	2.84	3.70	3.60	3.37	350	400
7	125150200111055	3.32	3.90	3.62	3.45	100	300
8	125150200111060	3.21	3.68	3.52	3.80	1050	300
9	125150200111062	3.00	3.65	3.21	3.78	200	1000
10	125150200111070	3.47	3.87	3.57	3.26	1050	150
11	125150200111078	2.84	3.48	3.62	3.65	0	0
12	125150200111156	3.16	2.95	3.29	3.00	0	700
13	125150201111009	3.32	3.20	3.67	3.48	0	300
14	125150201111025	3.32	3.65	3.74	3.70	0	350
15	125150201111027	3.00	2.87	3.14	2.89	0	400
16	125150201111030	3.39	3.73	3.67	3.67	0	300
17	125150201111041	3.68	3.77	3.48	3.43	0	0
18	125150207111086	2.58	2.35	2.06	2.05	0	0
19	125150301111031	2.53	3.10	2.81	3.05	450	1600
20	125150400111047	3.69	3.10	3.64	3.56	0	1650

3.3.6.1 Normalisasi Data

Proses normalisasi data digunakan untuk menyamakan skala parameter data ke dalam sebuah *range* yang spesifik, misalkan 0 s/d 1. Proses normalisasi dilakukan dengan langkah sebagai berikut :

1. Untuk setiap parameter cari nilai minimum dan nilai maksimum parameter.
2. Untuk setiap data, hitung normalisasi dengan menggunakan rumus *Min-Max Normalization* yang ditunjukkan pada Persamaan (2.4).

$$X_i = \frac{x_i - MIN\ (f)}{MAX\ (f) - MIN\ (f)}$$

Perhitungan parameter IP1

$$d_1P_1 = \frac{(3.32 - 2.53)}{(3.69 - 2.53)} = 0.681034$$

$$d_2P_1 = \frac{(3.39 - 2.53)}{(3.69 - 2.53)} = 0.741379$$

$$d_3P_1 = \frac{(3.26 - 2.53)}{(3.69 - 2.53)} = 0.162931$$

$$d_4P_1 = \frac{(3.21 - 2.53)}{(3.69 - 2.53)} = 0.586207$$

$$d_5P_1 = \frac{(3.05 - 2.53)}{(3.69 - 2.53)} = 0.448276$$

Perhitungan parameter IP2

$$d_1P_2 = \frac{(4.00 - 2.35)}{(4.00 - 2.35)} = 1$$

$$d_2P_2 = \frac{(3.62 - 2.35)}{(4.00 - 2.35)} = 0.769697$$

$$d_3P_2 = \frac{(3.37 - 2.35)}{(4.00 - 2.35)} = 0.618182$$

$$d_4P_2 = \frac{(3.75 - 2.35)}{(4.00 - 2.35)} = 0.848485$$

$$d_5P_2 = \frac{(3.62 - 2.35)}{(4.00 - 2.35)} = 0.769697$$

Perhitungan parameter IP3

$$d_1P_3 = \frac{(3.21 - 2.06)}{(3.74 - 2.06)} = 0.684524$$

$$d_2P_3 = \frac{(3.83 - 2.06)}{(3.74 - 2.06)} = 0.458333$$

$$d_3P_3 = \frac{(3.74 - 2.06)}{(3.74 - 2.06)} = 1$$

$$d_4P_3 = \frac{(3.52 - 2.06)}{(3.74 - 2.06)} = 0.869048$$

$$d_5P_3 = \frac{(3.36 - 2.06)}{(3.74 - 2.06)} = 0.77381$$

Perhitungan parameter IP4

$$d_1P_4 = \frac{(3.63 - 2.05)}{(3.80 - 2.05)} = 0.902857$$

$$d_2P_4 = \frac{(3.10 - 2.05)}{(3.80 - 2.05)} = 0.6$$

$$d_3P_4 = \frac{(3.61 - 2.05)}{(3.80 - 2.05)} = 0.891429$$

$$d_4P_4 = \frac{(3.20 - 2.05)}{(3.80 - 2.05)} = 0.657143$$

$$d_5P_4 = \frac{(3.67 - 2.05)}{(3.80 - 2.05)} = 0.925714$$

Perhitungan parameter Prestasi

$$d_1P_5 = \frac{(100 - 0)}{(1350 - 0)} = 0.074074$$

$$d_2P_5 = \frac{(0 - 0)}{(1350 - 0)} = 0$$

$$d_3P_5 = \frac{(0 - 0)}{(1350 - 0)} = 0$$

$$d_4P_5 = \frac{(1350 - 0)}{(1350 - 0)} = 1$$

$$d_5P_5 = \frac{(100 - 0)}{(1350 - 0)} = 0.074074$$

Perhitungan parameter Aktivitas

$$d_1P_6 = \frac{(900 - 0)}{(1650 - 0)} = 0.545455$$

$$d_2P_6 = \frac{(0 - 0)}{(1650 - 0)} = 0$$

$$d_3P_6 = \frac{(700 - 0)}{(1650 - 0)} = 0.424242$$

$$d_4P_6 = \frac{(450-0)}{(1650-0)} = 0.272727$$

$$d_5P_6 = \frac{(900-0)}{(1650-0)} = 0.545455$$

Untuk semua dataset yang telah dilakukan normalisasi dapat dilihat pada tabel 3.5 berikut ini :

Tabel 3.5 Tabel Dataset Mahasiswa yang telah dinormalisasi

No	NIM	IP1	IP2	IP3	IP4	Prestasi	Aktivitas
1	125150100111018	0.68103	1	0.68452	0.90286	0.07407	0.54545
2	125150101111001	0.74138	0.7697	0.45833	0.6	0	0
3	125150200111021	0.62931	0.61818	1	0.89143	0	0.42424
4	125150200111041	0.58621	0.84848	0.86905	0.65714	1	0.27273
5	125150200111042	0.44828	0.7697	0.77381	0.92571	0.07407	0.54545
6	125150200111049	0.26724	0.81818	0.91667	0.75429	0.25926	0.24242
7	125150200111055	0.68103	0.93939	0.92857	0.8	0.07407	0.18182
8	125150200111060	0.58621	0.80606	0.86905	1	0.77778	0.18182
9	125150200111062	0.40517	0.78788	0.68452	0.98857	0.14815	0.60606
10	125150200111070	0.81034	0.92121	0.89881	0.69143	0.77778	0.09091
11	125150200111078	0.26724	0.68485	0.92857	0.91429	0	0
12	125150200111156	0.5431	0.36364	0.73214	0.54286	0	0.42424
13	125150201111009	0.68103	0.51515	0.95833	0.81714	0	0.18182
14	125150201111025	0.68103	0.78788	1	0.94286	0	0.21212
15	125150201111027	0.40517	0.31515	0.64286	0.48	0	0.24242
16	125150201111030	0.74138	0.83636	0.95833	0.92571	0	0.18182
17	125150201111041	0.99138	0.86061	0.84524	0.78857	0	0
18	125150207111086	0.0431	0	0	0	0	0
19	125150301111031	0	0.45455	0.44643	0.57143	0.33333	0.9697
20	125150400111047	1	0.45455	0.94048	0.86286	0	1

3.3.6.2 Inisialisasi Centroid Awal

Setelah dilakukan normalisasi, proses selanjutnya adalah proses inisialisasi *centroid* awal. Penentuan pusat awal *cluster* ini dilakukan secara *random*. Setelah menentukan pusat *cluster* secara *random*. Pada perhitungan ini inisialisasi awal *centroid* ada pada tabel 3.6 berikut ini :

Tabel 3.6 Tabel inisialisasi *centroid* awal

No.	NIM	IP1	IP2	IP3	IP4	Prestasi	Aktivitas
5	125150200111042	0.44828	0.7697	0.77381	0.92571	0.07407	0.54545
10	125150200111070	0.81034	0.92121	0.89881	0.69143	0.77778	0.09091
14	125150201111025	0.68103	0.78788	1	0.94286	0	0.21212

3.3.6.3 Perhitungan Jarak data terhadap Pusat Cluster

Langkah selanjutnya adalah menghitung jarak data terhadap pusat *cluster* yang menggunakan jarak *Manhattan*. $d = \sum_{i=1}^{n} |x_i - y_i|$ Persamaan matematikanya ditunjukkan pada persamaan (2.5). Setelah didapatkan jarak maka di cari jarak yang paling dekat dengan pusat *cluster*:$w_{j*}|x_i - w_{j*}| \leq |x_i - w_j|, j \in 1, \ldots, k$

Contoh perhitungan pada C1 untuk parameter IP1:

X1W1
|(0.681034-0.448276)| + |(1-0.769697)| + |(0.684524-0.77381)| + |(0.902857-0.925714)| + |(0.074074-0.074074)| + |(0.545455-0. 545455)| = 0.575205

X2W1
|(0.741379-0.448276)| + |(0.769697-0.769697)| + |(0.458333-0.77381)| + |(0.6-0.925714)| + |(0-0.074074)| + |(0-0. 545455)| = 1.553823

X3W1
|(0.62931-0.448276)| + |(0.618182-0.769697)| + |(1-0.77381)| + |(0.891429-0.925714)| + |(0-0.074074)| + |(0.424242-0. 545455)| = 0.788312

X4W1
|(0.586207-0.448276)| + |(0.848485-0.769697)| + |(0.869048-0.77381)| + |(0.657143-0.925714)| + |(1-0.074074)| + |(0.272727-0. 545455)| = 1.779182

X5W1
|(0.448276-0.448276)| + |(0.769697-0.769697)| + |(0.77381-0.77381)| + |(0.925714-0.925714)| + |(0.074074-0.074074)| + |(0. 545455-0. 545455)| = 0

Untuk semua dataset pada C1 yang telah dilakukan perhitungan dapat dilihat pada tabel 3.7 berikut ini:

Tabel 3.7 Tabel hasil perhitungan dataset mahasiswa pada C1

C1	IP1	IP2	IP3	IP4	Prestasi	Aktivitas	Nearest Dist
x1.w1	0.232759	0.230303	0.089286	0.022857	0	0	0.575205
x2.w1	0.293103	0	0.315476	0.325714	0.074074	0.545455	1.553823
x3.w1	0.181034	0.151515	0.22619	0.034286	0.074074	0.121212	0.788312
x4.w1	0.137931	0.078788	0.095238	0.268571	0.925926	0.272727	1.779182
x5.w1	0	0	0	0	0	0	0
x6.w1	0.181034	0.048485	0.142857	0.171429	0.185185	0.30303	1.032021
x7.w1	0.232759	0.169697	0.154762	0.125714	0	0.363636	1.046568
x8.w1	0.137931	0.036364	0.095238	0.074286	0.703704	0.363636	1.411159
x9.w1	0.043103	0.018182	0.089286	0.062857	0.074074	0.060606	0.348108
x10.w1	0.362069	0.151515	0.125	0.234286	0.703704	0.454545	2.031119
x11.w1	0.181034	0.084848	0.154762	0.011429	0.074074	0.545455	1.051602
x12.w1	0.094828	0.406061	0.041667	0.382857	0.074074	0.121212	1.120698
x13.w1	0.232759	0.254545	0.184524	0.108571	0.074074	0.363636	1.21811
x14.w1	0.232759	0.018182	0.22619	0.017143	0.074074	0.333333	0.901681
x15.w1	0.043103	0.454545	0.130952	0.445714	0.074074	0.30303	1.45142
x16.w1	0.293103	0.066667	0.184524	0	0.074074	0.363636	0.982004
x17.w1	0.543103	0.090909	0.071429	0.137143	0.074074	0.545455	1.462113
x18.w1	0.405172	0.769697	0.77381	0.925714	0.074074	0.545455	3.493922
x19.w1	0.448276	0.315152	0.327381	0.354286	0.259259	0.424242	2.128596
x20.w1	0.551724	0.315152	0.166667	0.062857	0.074074	0.454545	1.625019

Contoh perhitungan pada C2 :

X1W2
|(0.681034-0.810345)| + |(1-0.921212)| + |(0.684524-0.89881)| + |(0.902857-0.691429)| + |(0.074074-0.777778)| + |(0.545455-0.090909)| = 1.792062

X2W2
|(0.741379-0.810345)| + |(0.769697-0.921212)| + |(0.458333-0.89881)| + |(0.6-0.691429)| + |(0-0.777778)| + |(0-0.090909)| = 1.621072

X3W2
|(0.62931-0.810345)| + |(0.618182-0.921212)| + |(1-0.89881)| + |(0.891429-0.691429)| + |(0-0.777778)| + |(0.424242-0.090909)| = 1.896366

X4W2
|(0.586207-0.810345)| + |(0.848485-0.921212)| + |(0.869048-0.89881)| + |(0.657143-0.691429)| + |(1-0.777778)| + |(0.272727-0.090909)| = 0.764953

X5W2
|(0.448276-0.810345)| + |(0.769697-0.921212)| + |(0.77381-0.89881)| + |(0.925714-0.691429)| + |(0.074074-0.777778)| + |(0. 545455-0.090909)| = 2.031119

Untuk semua dataset pada C2 yang telah dilakukan perhitungan dapat dilihat pada tabel 3.8 berikut ini:

Tabel 3.8 Tabel hasil perhitungan dataset mahasiswa pada C2

C2	IP1	IP2	IP3	IP4	Prestasi	Aktivitas	Nearest Dist
x1.w2	0.12931	0.078788	0.214286	0.211429	0.703704	0.454545	1.792062
x2.w2	0.068966	0.151515	0.440476	0.091429	0.777778	0.090909	1.621072
x3.w2	0.181034	0.30303	0.10119	0.2	0.777778	0.333333	1.896366
x4.w2	0.224138	0.072727	0.029762	0.034286	0.222222	0.181818	0.764953
x5.w2	0.362069	0.151515	0.125	0.234286	0.703704	0.454545	2.031119
x6.w2	0.543103	0.10303	0.017857	0.062857	0.518519	0.151515	1.396882
x7.w2	0.12931	0.018182	0.029762	0.108571	0.703704	0.090909	1.080438
x8.w2	0.224138	0.115152	0.029762	0.308571	0	0.090909	0.768532
x9.w2	0.405172	0.133333	0.214286	0.297143	0.62963	0.515152	2.194715
x10.w2	0	0	0	0	0	0	0
x11.w2	0.543103	0.236364	0.029762	0.222857	0.777778	0.090909	1.900773
x12.w2	0.267241	0.557576	0.166667	0.148571	0.777778	0.333333	2.251166
x13.w2	0.12931	0.406061	0.059524	0.125714	0.777778	0.090909	1.589296
x14.w2	0.12931	0.133333	0.10119	0.251429	0.777778	0.121212	1.514253
x15.w2	0.405172	0.606061	0.255952	0.211429	0.777778	0.151515	2.407907
x16.w2	0.068966	0.084848	0.059524	0.234286	0.777778	0.090909	1.31631
x17.w2	0.181034	0.060606	0.053571	0.097143	0.777778	0.090909	1.261042
x18.w2	0.767241	0.921212	0.89881	0.691429	0.777778	0.090909	4.147378
x19.w2	0.810345	0.466667	0.452381	0.12	0.444444	0.878788	3.172625
x20.w2	0.189655	0.466667	0.041667	0.171429	0.777778	0.909091	2.556286

Contoh perhitungan pada C3 :

X1W3
|(0.681034-0.681034)| + |(1-0.787879)| + |(0.684524-1)| + |(0.902857-0.942857)| + |(0.074074-0)| + |(0.545455-0.212121)| = 0.975005

X2W3
|(0.741379-0.681034)| + |(0.769697-0.787879)| + |(0.458333-1)| + |(0.6-0.942857)| + |(0-0)| + |(0-0.212121)| = 1.175172

X3W3
|(0.62931-0.681034)| + |(0.618182-0.787879)| + |(1-1)| + |(0.891429-0.942857)| + |(0-00)| + |(0.424242-0.212121)| = 0.484971

X4W3
|(0.586207-0.681034)| + |(0.848485-0.787879)| + |(0.869048-1)| + |(0.657143-0.942857)| + |(1-0)| + |(0.272727-0.212121)| = 1.632706

X5W3
|(0.448276-0.681034)| + |(0.769697-0.787879)| + |(0.77381-1)| + |(0.925714-0.942857)| + |(0.074074-0)| + |(0. 545455-0.212121)| = 0.901681

Untuk semua dataset pada C3 yang telah dilakukan perhitungan dapat dilihat pada tabel 3.9 berikut ini:

Tabel 3.9 Tabel hasil perhitungan dataset mahasiswa pada C3

C3	IP1	IP2	IP3	IP4	Prestasi	Aktivitas	Nearest Dist
x1.w3	0	0.212121	0.315476	0.04	0.074074	0.333333	0.975005
x2.w3	0.060345	0.018182	0.541667	0.342857	0	0.212121	1.175172
x3.w3	0.051724	0.169697	0	0.051429	0	0.212121	0.484971
x4.w3	0.094828	0.060606	0.130952	0.285714	1	0.060606	1.632706
x5.w3	0.232759	0.018182	0.22619	0.017143	0.074074	0.333333	0.901681
x6.w3	0.413793	0.030303	0.083333	0.188571	0.259259	0.030303	1.005563
x7.w3	0	0.151515	0.071429	0.142857	0.074074	0.030303	0.470178
x8.w3	0.094828	0.018182	0.130952	0.057143	0.777778	0.030303	1.109185
x9.w3	0.275862	0	0.315476	0.045714	0.148148	0.393939	1.17914
x10.w3	0.12931	0.133333	0.10119	0.251429	0.777778	0.121212	1.514253
x11.w3	0.413793	0.10303	0.071429	0.028571	0	0.212121	0.828945
x12.w3	0.137931	0.424242	0.267857	0.4	0	0.212121	1.442152
x13.w3	0	0.272727	0.041667	0.125714	0	0.030303	0.470411
x14.w3	0	0	0	0	0	0	0
x15.w3	0.275862	0.472727	0.357143	0.462857	0	0.030303	1.598892
x16.w3	0.060345	0.048485	0.041667	0.017143	0	0.030303	0.197942
x17.w3	0.310345	0.072727	0.154762	0.154286	0	0.212121	0.904241
x18.w3	0.637931	0.787879	1	0.942857	0	0.212121	3.580788
x19.w3	0.681034	0.333333	0.553571	0.371429	0.333333	0.757576	3.030277
x20.w3	0.318966	0.333333	0.059524	0.08	0	0.787879	1.579701

Setelah mendapatkan *nearest distance* data pada tiap *cluster*, langkah selanjutnya adalah membandingkan *nearest distance* yang terdapat di masing-

masing *cluster*, nilai yang mendekati pusat *centroid* dipilih sebagai *cluster* akhir yang telah di klasterisasi, proses ini dapat dilihat pada tabel 3.10 berikut ini :

Tabel 3.10 Hasil Pengklasteran

NO	NIM	C1	C2	C3	CLUSTER
1	125150100111018	0.575205	1.792062	0.975005	C1
2	125150101111001	1.553823	1.621072	1.175172	C3
3	125150200111021	0.788312	1.896366	0.484971	C3
4	125150200111041	1.779182	0.764953	1.632706	C2
5	125150200111042	0	2.031119	0.901681	C1
6	125150200111049	1.032021	1.396882	1.005563	C3
7	125150200111055	1.046568	1.080438	0.470178	C3
8	125150200111060	1.411159	0.768532	1.109185	C2
9	125150200111062	0.348108	2.194715	1.17914	C1
10	125150200111070	2.031119	0	1.514253	C2
11	125150200111078	1.051602	1.900773	0.828945	C3
12	125150200111156	1.120698	2.251166	1.442152	C1
13	125150201111009	1.21811	1.589296	0.470411	C3
14	125150201111025	0.901681	1.514253	0	C3
15	125150201111027	1.45142	2.407907	1.598892	C1
16	125150201111030	0.982004	1.31631	0.197942	C3
17	125150201111041	1.462113	1.261042	0.904241	C3
18	125150207111086	3.493922	4.147378	3.580788	C1
19	125150301111031	2.128596	3.172625	3.030277	C1
20	125150400111047	1.625019	2.556286	1.579701	C3

3.3.6.4 Update Centroid

Penentuan *centroid* baru ini dilakukan setelah mendapatkan *cluster*. Proses *update* pusat *cluster*/*centroid* ini dilakukan dengan menjumlahkan nilai data yang sama dengan *cluster* yang telah dibentuk dan dibagi dengan jumlah data pada tiap *cluster* yang terbentuk.

Untuk menghitung *update centroid* menggunakan pesamaan 2.6 yaitu :

$$j = \sum_{xl \in cj} \frac{xl}{|cj|}$$

Update centroid pada C1 untuk parameter IP1

(0.681034+0.448276+0.405172+0.543103+0.405172+0.043103+0)/7= 0.360837

Update centroid pada C2 untuk parameter IP1

(0.586207+0.586207+0.810345)/3 = 0.66092

Update centroid pada C3 untuk parameter IP1

(0.741379+0.62931+0.267241+0.681034+0.267241+0.681034+0.681034+0.741379+0.991379+1)/10 = 0.668103

Untuk hasil *update centroid* dapat dilihat pada tabel 3.11 berikut ini:

Tabel 3.11 Hasil *Update Centroid*

	IP1	IP2	IP3	IP4	PRESTASI	AKTVITAS
C1	0.360837	0.527273	0.566327	0.630204	0.089947	0.47619
C2	0.66092	0.858586	0.878968	0.782857	0.851852	0.181818
C3	0.668103	0.728485	0.893452	0.829714	0.033333	0.242424

3.3.6.5 Error Function

Perhitungan terakhir pada Algoritma *Balance K-Means* adalah menghitung nilai *error function*. *Error function* ini digunakan untuk melihat kualitas *cluster* pada tiap iterasi. Di dalam Algoritma *Balance K-Means* pemberhentian iterasi selain menggunakan *cluster* data yang sama juga dapat dilakukan dengan melihat nilai *error function* yang sama.

Perhitungan nilai *error function* menggunakan persaman 2.7 di bawah ini :

$$e = \sum_{xl \in cj} |x_i - w_j|^2$$

Nilai *error function* untuk data 1 *cluster* 1

e = ($|0.68103-0.41379|^2 + |1-0.61515|^2 + |0.68452-0.66071|^2 + |0.90286-0.73524|^2 + |0.07407-0.10494|^2 + |0.54545-0.55556|^2$) = 0.54545

Nilai *error function* untuk data 1 *cluster* 2

e = ($|0.68103-0.66092|^2 + |1-85859|^2 + |0.68452-0.87897|^2 + |0.90286-0.78286|^2 + |0.07407-0.85185|^2 + |0.54545-181812|^2$) = 0.80978

Nilai *error function* untuk data 1 *cluster* 3

e = (|0.68103-0.61129|^2 + |1-0.66226|^2 + |0.68452-0.81223|^2 + |0.90286-0.75429|^2 + |0.07407-0.0303|^2 + |0.54545-0.22039|^2) = 0.2649

Total *Error function* untuk data 1

(0.54545+0.80978+0.2649) = 1.323927

Untuk hasil *error function* dapat dilihat pada tabel 3.12 berikut ini:

Tabel 3.12 Hasil E*rror Function*

|x-w1|^2	|x-w2|^2	|x-w3|^2	Error Function
0.249244	0.809781	0.264902	1.323927
0.510098	0.983455	0.227002	1.720555
0.214222	0.869651	0.098809	1.182683
1.0148	0.051799	0.991026	2.057625
0.0752	0.821755	0.176565	1.073519
0.250439	0.513692	0.206492	0.970623
0.393125	0.614627	0.100697	1.108449
0.772067	0.061077	0.645118	1.478262
0.099072	0.825714	0.292094	1.216879
0.978144	0.048754	0.693611	1.72051
0.449795	0.963611	0.2075	1.620905
0.150349	1.122434	0.187416	1.460199
0.327395	0.851478	0.054209	1.233081
0.388432	0.772223	0.092449	1.253104
0.264603	1.237523	0.26829	1.770416
0.431797	0.759326	0.100377	1.2915
0.750401	0.869088	0.235566	1.855055
1.812595	3.263026	2.039569	7.11519
0.493448	1.72147	1.237356	3.452275
0.672529	1.68348	0.831196	3.187205

3.3.6.6 Silhoutte Coefficient

Evaluasi *cluster* ini dilakukan untuk menguji kualitas *cluster* yang terbentuk. Apabila nilai *Silhoutte Coefficient s*(i) semakin mendekati nilai 1 maka semakin cocok *cluster* yang terbentuk sehingga hasil pengelompokkan datanya juga semakin baik. Akan tetapi jika nilai semakin mendekati nilai -1 maka semakin tinggi nilai

ketidakcocokan data terhadap *cluster*-nya saat ini sehingga hasil pengelompokkan datanya semakin buruk.

1. Menghitung nilai rata-rata jarak data *i* dengan semua data lain yang berada dalam satu *cluster* menggunakan Persamaan (2.8). Untuk contoh data dapat dilihat pada tabel 3.13 berikut ini :

Tabel 3.13 Contoh data perhitungan *Silhoutte Coefficient*

Cluster	Data ke-	IP1	IP2	IP3	IP4	Prestasi	Aktivitas
C1	1	0.681	1	0.684	0.902	0.074	0.545
C1	5	0.448	0.769	0.773	0.925	0.074	0.545
C1	9	0.405	0.787	0.684	0.988	0.148	0.606
C1	12	0.543	0.363	0.732	0.542	0	0.424
C1	15	0.405	0.315	0.642	0.48	0	0.242
C1	19	0	0.454	0.446	0.571	0.333	0.969

Misalkan data yang akan diuji adalah Data 1.

- $D_{(1,1)}$

$$\sqrt{(0.681-0.681)^2+(1-1)^2+(0.684-0.684)^2+(0.902-0.902)^2+(0.074-0.074)^2+(0.545-0.545)^2}$$

$D_{(1,1)} = 0$

- $D_{(1,5)}$

$$\sqrt{(0.681-0.448)^2+(1-0.769)^2+(0.684-0.773)^2+(0.902-0.925)^2+(0.074-0.074)^2+(0.545-0.545)^2}$$

$D_{(1,5)} = 0.340$

- $D_{(1,9)}$

$$\sqrt{(0.681-0.405)^2+(1-0.787)^2+(0.684-0.684)^2+(0.902-0.988)^2+(0.074-0.148)^2+(0.545-0.606)^2}$$

$D_{(1,9)} = 0.370$

- $D_{(1,12)}$

$$\sqrt{(0.681-0.543)^2+(1-0.363)^2+(0.684-0.732)^2+(0.902-0.542)^2+(0.074-0)^2+(0.545-0.424)^2}$$

$D_{(1,12)} = 0.758$

- $D_{(1,15)}$

$$\sqrt{(0.681-0.405)^2+(1-0.315)^2+(0.684-0.642)^2+(0{,}902-0{,}48)^2+(0.074-0)^2+(0.545-0.242)^2}$$

$D_{(1,15)} = 0.907$

- $D_{(1,19)}$

$$\sqrt{(0.681-0)^2+(1-0.454)^2+(0{,}684-0{,}446)^2+(0.902-0{,}571)^2+(0.074-0.333)^2+(0.545-0.969)^2}$$

$D_{(1,19)} = 1.084$

Nilai $a_{(1)} = \frac{\sum D(i,j)}{|A|-1} = \frac{0+0.340+0.370+0.758+0.907+1.084}{5} = 0{,}6922$

2. Menghitung nilai rata-rata jarak data *i* tersebut dengan semua data di *cluster* lain, kemudian diambil nilai minimunmya. Perhitungan dilakukan dengan menggunakan Persamaan (2.9). Untuk contoh data perhitungan jarak data dengan data yang berada di *cluster* lain dapat dilihat pada tabel 3.14

Tabel 3.14 Contoh data perhitungan jarak data dengan data *cluster* lain

ClusterId	Data Ke-	IP1	IP2	IP3	IP4	Prestasi	Aktivitas
C1	Data 1	0.681	1	0.684	0.902	0.074	0.545
C2	Data 4	0.586	0.848	0.869	0.657	1	0.272

- $D_{(1,4)}$

$$\sqrt{(0.681-0.586)^2+(1-0.848)^2+(0.684-0.869)^2+(0.902-0.657)^2+(0.074-1)^2+(0.545-0.272)^2}$$

$D_{(1,4)}$ = 1.028

Hasil perhitungan selengkapnya dapat dilihat pada Tabel 3.15

Tabel 3.15 Hasil Perhitungan Jarak *d(i)* ke Seluruh Data di *Cluster* lain

Data Uji	Data di Cluster Lain	Cluster	D(i,j)	Rata-Rata Jarak D(i,C)
Data 1	Data 4	C2	1.028636	0.926225
	Data 8	C2	0.847066	
	Data 10	C2	0.902973	
	Data 2	C3	0.708924	0.729567
	Data 3	C3	0.517973	
	Data 6	C3	0.637469	
	Data 7	C3	0.45392	
	Data 11	C3	0.795765	
	Data 13	C3	0.674622	
	Data 14	C3	0.512561	
	Data 16	C3	0.49359	
	Data 17	C3	0.676493	
	Data 18	C3	1.730228	
	Data 20	C3	0.823693	

Nilai $b_{(5)}$ = min (D(i,C))

= min (0.926225; 0.729567) = 0.729567

3. Menghitung nilai *Silhouette Coeffient* dengan menggunakan Persamaan (2.10).

$$s_{(1)} = \frac{b_{(1)} - a_{(1)}}{\max(a_{(1)}, b_{(1)})}$$

$$s_{(1)} = \frac{0.729567 - 0.6922}{\max(0.6922; 0.729567)} = 0.05115$$

Jadi nilai *Silhouette Coefficient* untuk Data 1 adalah 0.05115

Penentuan kualitas *cluster* yang dihasilkan aplikasi dilakukan dengan menghitung nilai *Silhouette Coefficient* seluruh data kemudian dihitung nilai rata-ratanya. Hasilnya dapat dilihat pada Tabel 3.16

Tabel 3.16 Nilai *Silhoette Coefficient*

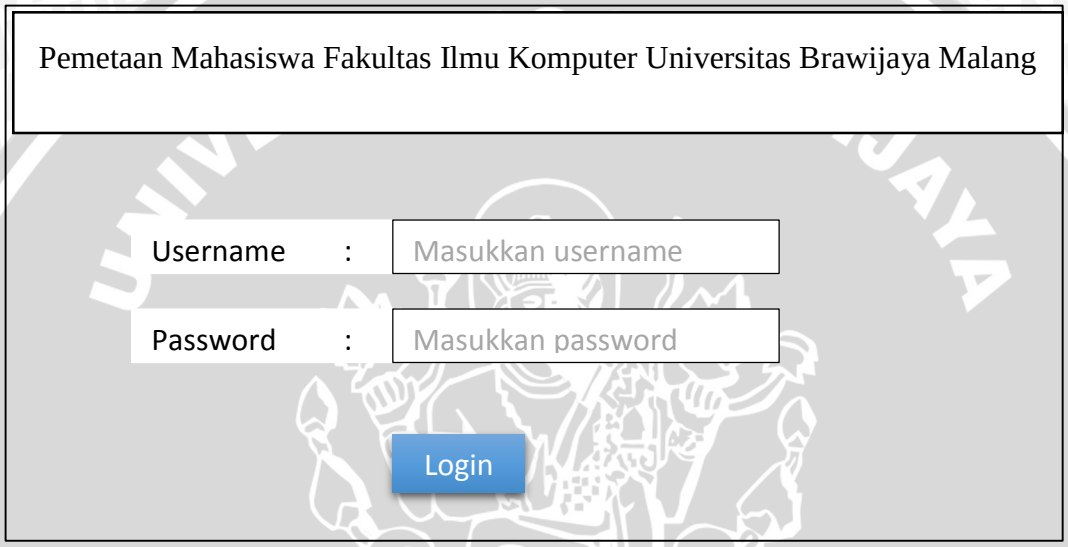
Data ke-	Cluster	*s(i)*	Rata-rata *s(i)*	Rata-rata akhir
1	**C1**	0.0511471	0.1689787	0.281302611
5	**C1**	0.2041509		
9	**C1**	0.2643354		
12	**C1**	0.1360069		
15	**C1**	0.0872662		
19	**C1**	0.2709657		
4	C2	0.6412243	0.6051468	
8	C2	0.5694929		
10	C2	0.6047233		
2	**C3**	0.0565321	0.0697823	
3	**C3**	-0.0108801		
6	**C3**	-0.0938534		
7	**C3**	0.188877		
11	**C3**	0.0830406		
13	**C3**	0.1435215		
14	**C3**	0.2203135		
16	**C3**	0.2401397		
17	**C3**	0.2359065		
18	**C3**	-0.1365947		
20	**C3**	-0.1593977		

3.3.5 Perancangan Antarmuka

Pada perancangan antarmuka ini bertujuan untuk mewakili keadaan sebenarnya dari implementasi aplikasi yang akan dibangun. Implementasi aplikasi ini dibagi menjadi 3 halaman, yaitu : halaman *log*-in, halaman input data, halaman *clustering*, dan halaman hasil *clustering*.

3.3.5.1 Perancangan Antarmuka Halaman Login

Gambar 3.13 merupakan halaman *login* dari aplikasi ini

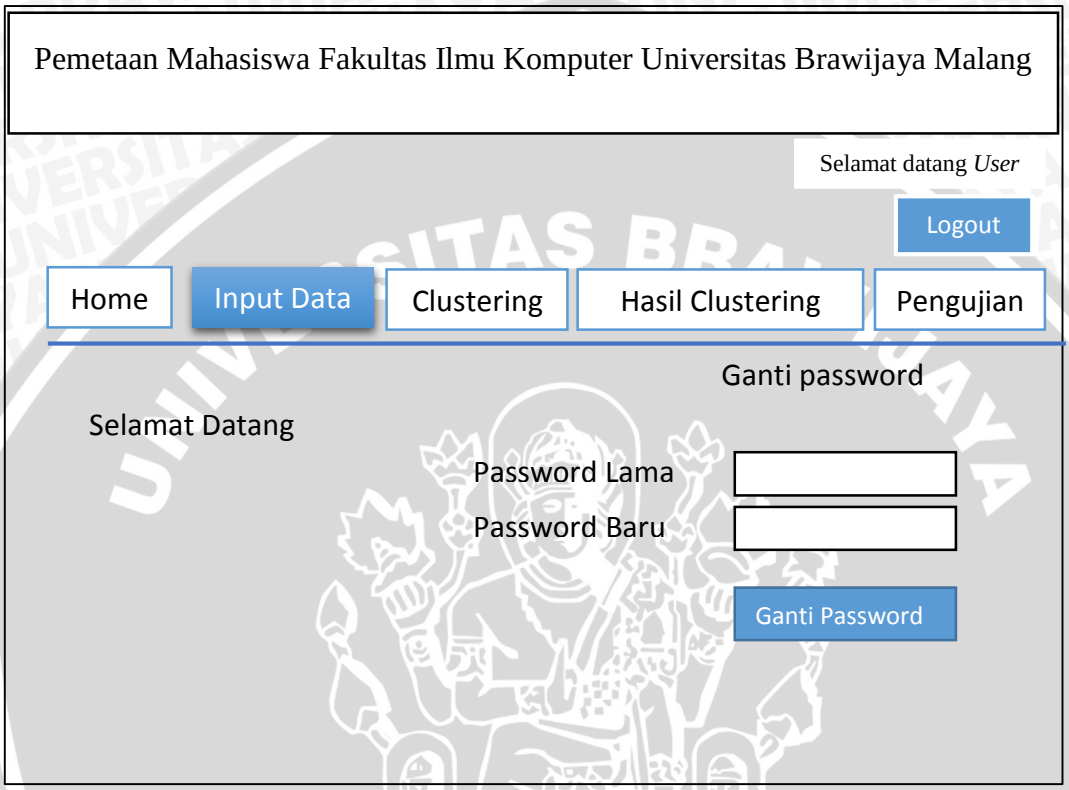


Gambar 3.13 Desain Antar Muka Halaman *login*

Dalam Gambar 3.13 menjelaskan tentang halaman *login*, dimana pengguna harus memasukkan *username* dan *password* terlebih dahulu untuk bisa masuk ke dalam aplikasi.

3.3.5.2 Perancangan Antar Muka Halaman Home

Antarmuka halaman *home* merupakan antarmuka untuk menampilkan ucapan selamat datang aplikasi dan juga proses ganti *password*. Rancangan antar muka *home* ditujukkan pada Gambar 3.14

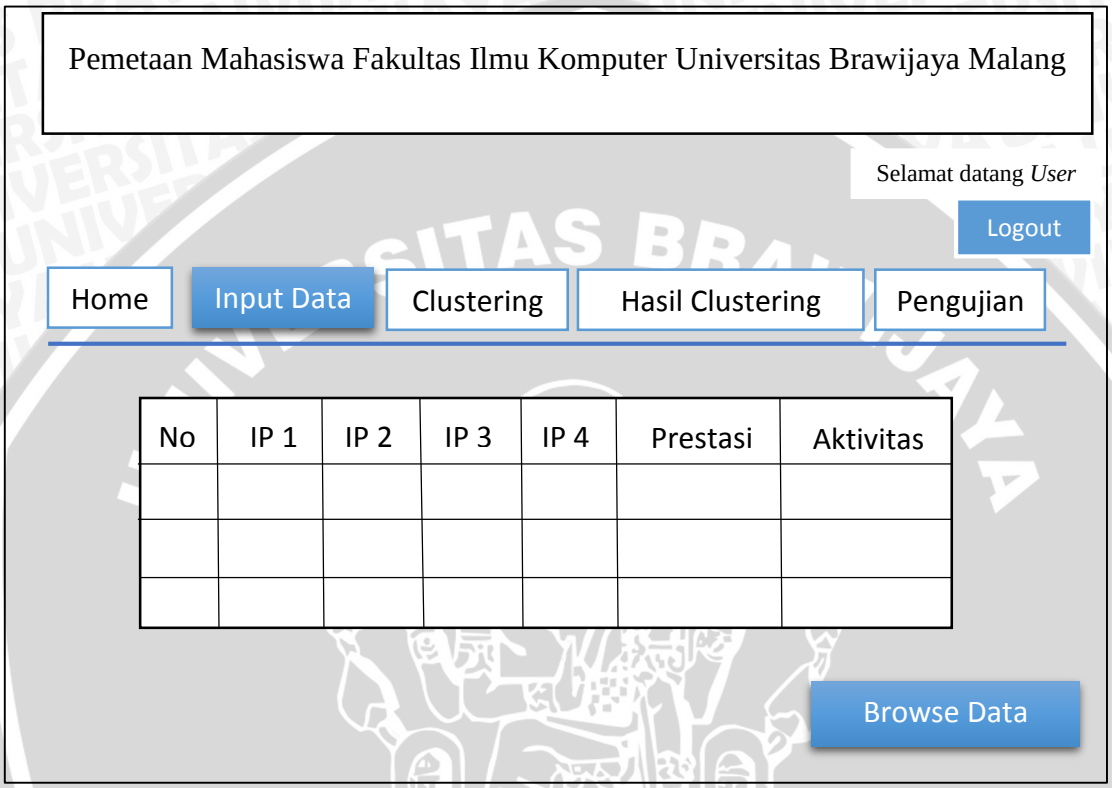


Gambar 3.14 Rancangan Antar Muka Halaman *Home*

Dalam Gambar 3.14 menjelaskan tentang halaman *Home*, dimana terdapat ucapan selamat datang kepada *user* serta form ganti *password*, yang digunakan apabila *user* ingin mengganti *password*.

3.3.5.3 Perancangan Antar Muka Halaman Input Data.

Antarmuka halaman *input* merupakan antarmuka untuk menampilkan data yang di masukkan untuk *cluster*. Rancangan antar muka *input* data ditujukkan pada Gambar 3.15



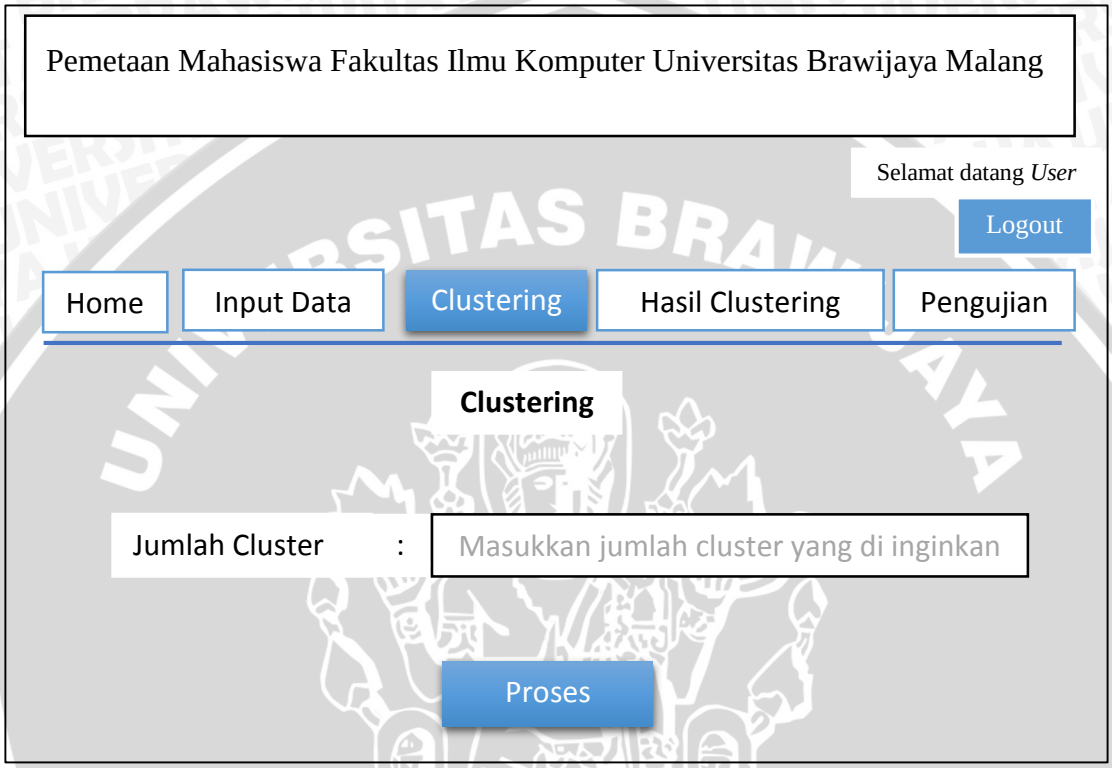
Gambar 3.15 Rancangan antarmuka *Input* data

Berikut ini merupakan keterangan rancangan antarmuka halaman *input* data pada Gambar 3.15 :

1. *Input* data digunakan untuk memasukkan data yang akan digunakan untuk perhitungan.
2. *Input* data terdiri dari 7 parameter yaitu IP1, IP2, IP3, IP4, prestasi dan aktivitas.

3.3.5.4 Perancangan Antarmuka Clustering

Antarmuka *clustering* adalah antarmuka untuk menampilkan masukkan jumlah *cluster* dan maksimum *iterasi* yang ingin dibentuk. Rancangan antarmuka *clustering* ditujukkan pada Gambar 3.16.



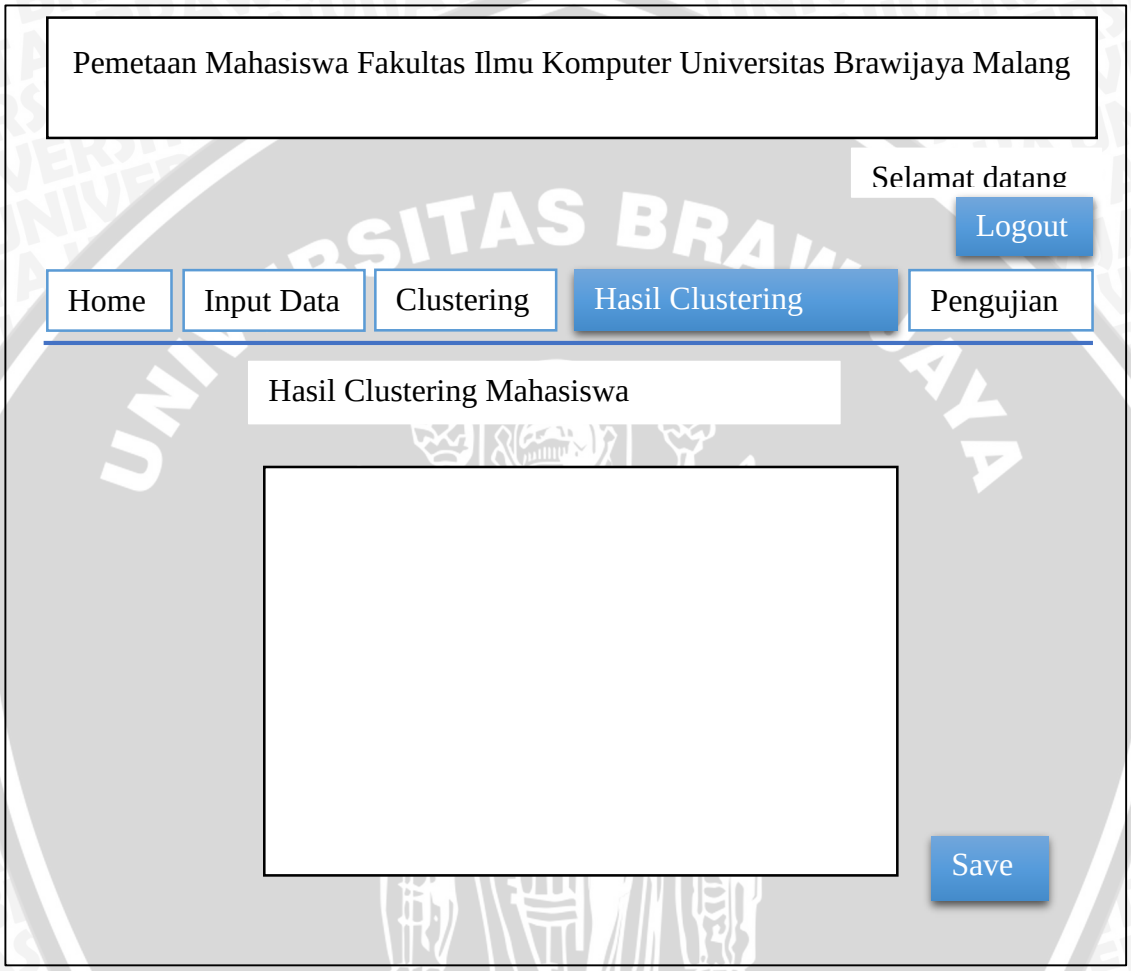
Gambar 3.16 Rancangan Antarmuka *clustering*

Berikut ini keterangan rancangan antarmuka *clustering* pada Gambar 3.16 :

1. Jumlah *cluster* digunakan untuk memasukkan banyaknya jumlah *cluster* yang akan dibentuk dari dataset mahasiswa.

3.3.5.5 Perancangan Antarmuka Hasil Clustering

Antarmuka hasil *clustering* adalah antarmuka untuk menampilkan hasil *clustering* yang terbentuk. Rancangan antarmuka hasil *clustering* ditujukkan pada Gambar 3.17



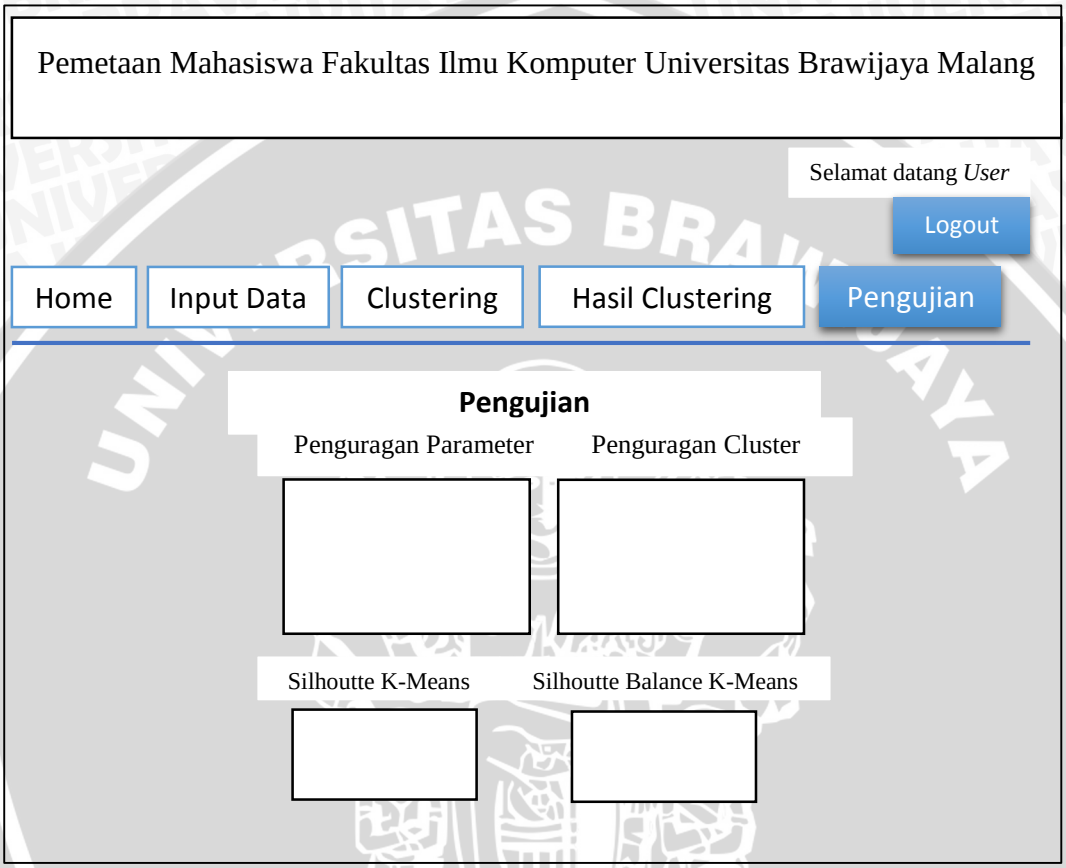
Gambar 3.17 Rancangan Antarmuka Hasil *clustering*

Berikut ini keterangan rancangan antarmuka clustering pada Gambar 3.17 :

1. Pada tabel hasil *clustering* ditampilkan hasil *cluster* akhir dari dataset mahasiswa berdasarkan data dan hasil clustering akhir data mahasiswa berdasarkan *cluster*.
2. Tombol *save* digunakan untuk menyimpan hasil *clustering* ke dalam dokumen berbentuk *.pdf*

3.3.5.6 Perancangan Antarmuka Pengujian

Antarmuka Pengujian adalah antarmuka untuk menampilkan hasil pengujian kualitas *cluster* menggunakan metode *Silhouette Coefficient*. Rancangan antarmuka evaluasi *clustering* ditujukkan pada Gambar 3.18



Gambar 3.18 Rancangan Antarmuka Pengujian

Berikut ini keterangan rancangan antarmuka *clustering* pada Gambar 3.18 :

1. Pada tabel pengujian ditampilkan hasil perhitungan kualitas *cluster Balance K-means* dan *K-Means* serta nilai kualitas *cluster* untuk pengurangan tiap parameter.

3.4 Implementasi

Implementasi aplikasi pemetaan dilakukan berdasarkan perancangan aplikasi yang dibuat. Implementasi aplikasi pemetaan ini menggunakan bahasa pemrograman berorientasi objek yaitu bahasa pemrograman PHP.

3.5 Pengujian

Pengujian ini bertujuan untuk melakukan pengujian terhadap aplikasi yang telah dibuat. Pengujian dimaksudkan untuk mengetahui apakah hasil *cluster* yang telah terbentuk memiliki kualitas *cluster* yang kuat atau lemah.

Kualitas hasil *clustering* berhubungan erat dengan tingkat validasi data yaitu bagaimana data-data dikelompokkan dalam suatu *cluster* (kelompok). Untuk mengetahui tingkat validasi data dalam suatu *cluster* digunakan metode *Silhouette Coefficient* seperti yang dijelaskan pada subbab 2.7. Adapun pengujian dan analisis yang akan dilakukan meliputi:

1. Pengaruh jumlah *cluster*

Pada analisis ini, jumlah *cluster* digunakan sebagai obyek analisis. Hal ini bertujuan untuk mengetahui berapa jumlah *cluster* yang menghasilkan tingkat validasi data tertinggi. Jumlah *cluster* yang digunakan untuk proses uji adalah 2, 3, 4, 5 dan 6.

2. Pengaruh parameter terhadap data yang diproses

Pada analisis ini, parameter data yang diproses digunakan sebagai obyek analisis. Hal ini bertujuan untuk mengetahui pengaruh parameter data terhadap validasi kelompok data. Terdapat 4 poin yang dilakukan dalam pengujian ini yaitu:

- Menggunakan semua parameter data 6 parameter data, yaitu IP1, IP2, IP3, IP4, data prestasi, serta data aktivitas.
- Menggunakan 5 parameter data dengan pengurangan parameter aktivitas, yang berupa data IP1, IP2, IP3, IP4, serta data prestasi.
- Menggunakan 4 parameter data dengan pengurangan parameter prestasi dan aktivitas, yang berupa data IP1, IP2, IP3, serta IP4.

3. Pengujian perbandingan K-Means

Pengujian ini dilakukan untuk melihat apakah metode *Balance K-Means* lebih baik dan lebih cocok digunakan untuk melakukan pemetaan mahasiswa dibandingkan dengan metode *K-Means* biasa. Dalam pengujian ini data mahasiswa yang ada akan diuji cobakan menggunakan metode *Balance K-Means* dengan 10 kali *run* program dan menggunakan metode *K-Means* dengan 10 kali *run* program.

4. Pengujian Akurasi

Pengujian ini dilakukan untuk melihat nilai akurasi dari metode *Balance K-Means*. Dalam pengujian ini data yang digunakan adalah data *iris* yang memiliki 150 data. Data iris digunakan karena data Iris merupakan data dengan rentang perbedaan nilai pada masing-masing parameter dataset iris cukup jelas, sehingga mudah untuk menemukan *cluster* yang tepat. Di dalam dataset iris juga sudah terdapat *cluster* yang telah terbentuk, sehingga memudahkan untuk mendapatkan nilai akurasi.

3.6 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah dilakukan proses pengujian aplikasi sehingga dapat diketahui efektifitas kinerja aplikasi. Tahap terakhir yaitu penulisan saran yang dapat membantu dalam pengembangan aplikasi pemetaan selanjutnya.

BAB IV
IMPLEMENTASI

Pada bab ini implementasi akan dibahas mengenai implementasi aplikasi yang dibuat baik itu perangkat keras maupun perangkat lunak, batasan-batasan implementasi: implementasi program dan implementasi *interface.*

4.1 Implementasi Sistem

Hasil kebutuhan dan perancangan perangkat lunak yang telah dibahas pada bab sebelumnya menjadi acuan untuk melakukan implementasi sebuah aplikasi yang dapat berfungsi sesuai kebutuhan. Spesifikasi aplikasi diimplementasikan pada perangkat keras dan perangkat lunak.

4.1.1 Implementasi Perangkat Keras

Pembuatan Pemetaan Mahasiswa menggunakan metode *Balance K-Means* menggunakan sebuah PC / Laptop dengan spesifikasi pada Tabel 4.1

Tabel 4.1 Spesifikasi Perangkat Keras PC / Laptop

Notebook TOSHIBA Satellite L745	
Nama *Hardware*	Spesifikasi
Processor	Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz (4 CPUs)
Memory (RAM)	4096MB
Graphic Card	NVIDIA GeForce GT 525M 2 GB

4.1.2 Implementasi Perangkat Lunak

Pembuatan Aplikasi Pemetaan Mahasiswa menggunakan metode *Balance K-Means* menggunakan sebuah aplikasi perangkat lunak dengan spesifikasi pada Tabel 4.2

Tabel 4.2 Spesifikasi Aplikasi Perangkat Lunak

Nama *Software*	Spesifikasi
Sistem Operasi	Windows 8 Pro 64-bit b(6,2 Build 9200)
Bahasa Pemrograman	PHP
Integrated Development Environment	phpDesigner 8
Database Management System	MySQL XAMMP
Browser	Google Chrome versi 42.0.2311.135 m

4.2 Implementasi Program

Hasil perancangan pemetaan mahasiswa yang telah diuraikan pada Bab 3 menjadi acuan untuk melakukan implementasi pada pemetaan mahasiswa. Bagian pemetaan mahasiswa yang diimplementasikan adalah implementasi algoritma dan implementasi antarmuka.

4.2.1 Implementasi Algoritma

Implementasi yang akan dibahas menggunakan bahasa pemrogaman php dan menggunakan *database* MySQL. Aplikasi pemetaan mahasiswa menggunakan metode *Balance K-Means* terdiri dari beberapa proses utama dan sub proses yang meliputi : proses normalisasi data, proses perhitungan jarak ke pusat *cluster*, proses penetuan pusat *cluster* baru, proses perhitungan *Silhouette Coefficient*.

4.2.1.1 Implementasi Normalisasi Data

Proses normalisasi data adalah proses awal dari metode *Balance K-Means*. Proses ini bertujuan untuk menyamakan rentang nilai pada seluruh parameter, yaitu diantara 0-1. Implementasi perhitungan normalisasi data ditunjukkan pada *Sourcecode* 4.1

***Sourcecode* 4.1** Implementasi Normalisasi Data

```
$query = "SELECT * FROM data_input ";
$exe = mysql_query($query);
for($a=1;$a<=$jumdata;$a++){
      $row = mysql_fetch_assoc($exe);
      if ($a==1){
        $maxip1= $row['ip1'];
        $minip1= $row['ip1'];
        $maxip2= $row['ip2'];
        $minip2= $row['ip2'];
        $maxip3= $row['ip3'];
        $minip3= $row['ip3'];
        $maxip4= $row['ip4'];
        $minip4= $row['ip4'];
        $maxpres= $row['prestasi'];
        $minpres= $row['prestasi'];
        $maxaktv= $row['aktivitas'];
        $minaktv= $row['aktivitas'];
       }
       $maxsip1= $row['ip1'];
       $minsip1= $row['ip1'];
       $maxsip2= $row['ip2'];
       $minsip2= $row['ip2'];
       $maxsip3= $row['ip3'];
       $minsip3= $row['ip3'];
       $maxsip4= $row['ip4'];
       $minsip4= $row['ip4'];
       $maxspres= $row['prestasi'];
       $minspres= $row['prestasi'];
       $maxsaktv= $row['aktivitas'];
       $minsaktv= $row['aktivitas'];

       $maxip1 = max($maxsip1,$maxip1);
       $minip1 = min($minsip1,$minip1);
       $maxip2 = max($maxsip2,$maxip2);
       $minip2 = min($minsip2,$minip2);
       $maxip3 = max($maxsip3,$maxip3);
       $minip3 = min($minsip3,$minip3);
       $maxip4 = max($maxsip4,$maxip4);
       $minip4 = min($minsip4,$minip4);
       $maxpres = max($maxspres,$maxpres);
       $minpres = min($minspres,$minpres);
       $maxaktv = max($maxsaktv,$maxaktv);
       $minaktv = min($minsaktv,$minaktv);
}
$maxminip1 = $maxip1 - $minip1;
$maxminip2 = $maxip2 - $minip2;
$maxminip3 = $maxip3 - $minip3;
$maxminip4 = $maxip4 - $minip4;
$maxminpres = $maxpres - $minpres;
$maxminaktv = $maxaktv - $minaktv;
for($a=1;$a<=$jumdata;$a++){
```

```
50      $rows = mysql_fetch_assoc($exe1);
51      $id[$a] = $rows['id'];
52      $nim[$a] = $rows['nim'];
53      $ip1[$a]= number_format (($rows['ip1']-$minip1)/
54      $maxminip1, 6);
55      $ip2[$a]= number_format (($rows['ip2']-$minip2)/
56      $maxminip2, 6);
57      $ip3[$a]= number_format (($rows['ip3']-$minip3)/
58      $maxminip3, 6);
59      $ip4[$a]= number_format (($rows['ip4']-$minip4)/
60      $maxminip4, 6);
61      if ($rows['prestasi'] == 0){ $prestasi[$a]= 0; } else
61  {
63          $prestasi[$a]= number_format (($rows['prestasi']-
64          $minpres)/$maxminpres, 6); }
65      if ($rows['aktivitas'] == 0){ $aktivitas[$a]= 0;} else
66  {
67          $aktivitas[$a]= number_format (($rows['aktivitas']-
68          $minaktv)/$maxminaktv, 6);}
69      mysql_query("INSERT INTO data_normalisasi (id, nim,
70      ip1, ip2, ip3, ip4, prestasi, aktivitas) VALUES
71      ('$id[$a]', '$nim[$a]','$ip1[$a]', '$ip2[$a]',
72      '$ip3[$a]', '$ip4[$a]', '$prestasi[$a]',
73      '$aktivitas[$a]')");
74  }
```

4.2.1.2 Implementasi Penentuan Pusat Awal *Cluster*

Proses penentuan pusat awal *cluster* merupakan proses setelah menghitung normalisasi. Proses ini bertujuan untuk mendapatkan suatu nilai yang digunakan sebagai pusat *cluster*. Proses penentuan pusat *cluster* ini dilakukan secara *random*. Pusat *cluster* inilah yang selanjutnya akan diproses dalam perhitungan jarak data ke pusat awal *cluster*. Implementasi dari proses penentuan pusat awal *cluster* ditunjukkan pada *Sourcecode* 4.2

***Sourcecode* 4.2** Implementasi Penentuan Pusat Awal *Cluster*

```
1   $query1 ="SELECT id,nim,ip1,ip2,ip3,ip4,prestasi,aktivitas
2   FROM data_normalisasi ORDER BY RAND() LIMIT $jum_cluster";
3   $jum = mysql_query($query1);
4   $num=mysql_num_rows($jum);
5   for($a=1;$a<=$num;$a++){
6       $ro = mysql_fetch_assoc($jum);
7       $idpc=$a;
8       $nimpc=$ro['nim'];
9       $ip1pc=$ro['ip1'];
10      $ip2pc=$ro['ip2'];
```

```
    $ip3pc=$ro['ip3'];
    $ip4pc=$ro['ip4'];
    $prestasipc=$ro['prestasi'];
    $aktivitaspc=$ro['aktivitas'];

    mysql_query("INSERT INTO pusat_cluster(id,nim,ip1,ip2,
    ip3,ip4,prestasi,aktivitas) VALUES('$idpc','$nimpc',
    '$ip1pc','$ip2pc','$ip3pc','$ip4pc','$prestasipc',
    '$aktivitaspc')");
    mysql_query("INSERT INTO pusat_awal_cluster(id,nim,
    ip1, ip2,ip3,ip4,prestasi,aktivitas)VALUES('$idpc',
    '$nimpc','$ip1pc','$ip2pc','$ip3pc','$ip4pc',
    '$prestasipc','$aktivitaspc')");
}
```

4.2.1.3 Implementasi Proses Perhitungan Jarak ke Pusat *Cluster*

Perhitungan jarak ke pusat *cluster* pada metode *Balance K-Means* dilakukan dengan menghitung data terhadap pusat *cluster* yang telah *dirandom* pada proses sebelumnya. Implementasi dari proses perhitungan jarak ke pusat *cluster* baru ditunjukkan pada *Sourcecode* 4.3

***Sourcecode* 4.3** Implementasi Perhitungan Jarak ke Pusat *Cluster*

```
$datapc[$iterasi] = mysql_query("SELECT * FROM
pusat_cluster ");
$j=0;
for($a=1;$a<=$num;$a++){
    $ro1 = mysql_fetch_assoc($datapc[$iterasi]);
    for($b=1;$b<=$jumdata;$b++){
        $j++;
        $id[$j] = $id[$b];
        $nimnorm[$j] = $nim[$b];
        $ip1norm[$j] = number_format (ABS($ip1[$b]-$ro1
        ['ip1']), 6);
        $ip2norm[$j] = number_format (ABS($ip2[$b]-$ro1
        ['ip2']), 6);
        $ip3norm[$j] = number_format (ABS($ip3[$b]-$ro1
        ['ip3']), 6);
        $ip4norm[$j] = number_format (ABS($ip4[$b]-$ro1
        ['ip4']), 6);
        $prestasinorm[$j] = number_format (ABS($prestasi
        [$b]-$ro1['prestasi']), 6);
        $aktivitasnorm[$j] = number_format (ABS($aktivitas
        [$b]-$ro1['aktivitas']), 6);
        $totnorm[$j] = number_format ($ip1norm[$j]+
        $ip2norm[$j]+$ip3norm[$j]+$ip4norm[$j]+
        $prestasinorm[$j]+$aktivitasnorm[$j], 6);
    }
```

```
}
$c=0;
for($b=1;$b<=$jumdata;$b++){
    $c+=1;
    $b

    $d=$c;
    for($a=1;$a<=$num;$a++){
        $totnormcluster[$a] = $totnorm[$d];
        $d+=$jumdata;
        if (!isset ($mincluster[$b])){
         $mincluster[$b] = $totnormcluster[$a];
         }
        $mincluster[$b] =
min($mincluster[$b],$totnormcluster[$a]);
    }
    for($a=1;$a<=$num;$a++){
        if($mincluster[$b]==$totnormcluster[$a]){
$mincluster[$b]= 'c'."".$a; }
    }
$minclusters[$iterasi][$b] = $mincluster[$b];

   mysql_query("CREATE TABLE $ite( id int(11) primary key,
   cluster varchar(3))");
   mysql_query("INSERT INTO $ite (id,cluster) VALUES ('$b'
   ,'$mincluster[$b]')");
}
```

4.2.1.4 Implementasi Penentuan Pusat *Cluster* Baru

Proses penentuan pusat *cluster* baru pada metode *Balance K-Means* dilakukan setelah melakukan perhitungan jarak ke pusat *cluster* dan mendapatkan nilai *cluster*. Nilai pusat *cluster* yang baru didapatkan dengan mencari rata-rata nilai dari seluruh data yang ada dalam satu *cluster*. Implementasi dari proses penentuan pusat *cluster* baru ditunjukkan pada *Sourcecode* 4.4

***Sourcecode* 4.4** Implementasi Penentuan Pusat *Cluster* Baru

```
for($b=1;$b<=$num;$b++){
    $centroid[$b]='c'."".$b;
    $ip1centroid[$b]='0';
    $ip2centroid[$b]='0';
    $ip3centroid[$b]='0';
    $ip4centroid[$b]='0';
    $prestasicentroid[$b]='0';
    $aktivitascentroid[$b]='0';
    $hitung[$b] = 0;
    for($a=1;$a<=$jumdata;$a++){
```

```
        if($mincluster[$a]=='c'."".$b){
            $hitung[$b]++;
            $ip1centroid[$b]=$ip1centroid[$b]+$ip1[$a];
            $ip2centroid[$b]=$ip2centroid[$b]+$ip2[$a];
            $ip3centroid[$b]=$ip3centroid[$b]+$ip3[$a];
            $ip4centroid[$b]=$ip4centroid[$b]+$ip4[$a];

$prestasicentroid[$b]=$prestasicentroid[$b]+$prestasi[$a];

$aktivitascentroid[$b]=$aktivitascentroid[$b]+$aktivitas[$a];
        }
    }

$ip1centroid[$b]=number_format($ip1centroid[$b]/$hitung[$b],6);

$ip2centroid[$b]=number_format($ip2centroid[$b]/$hitung[$b],6);

$ip3centroid[$b]=number_format($ip3centroid[$b]/$hitung[$b],6);

$ip4centroid[$b]=number_format($ip4centroid[$b]/$hitung[$b],6);
    if ($prestasicentroid[$b] == 0){
        $prestasicentroid[$b]= 0;
    }
    else {
        $prestasicentroid[$b]=number_format
($prestasicentroid[$b]/$hitung[$b], 6);
    }
    if ($aktivitascentroid[$b] == 0){
        $aktivitascentroid[$b]= 0;
    }
    else {
        $aktivitascentroid[$b]=number_format
($aktivitascentroid[$b]/$hitung[$b], 6);
    }

    mysql_query("UPDATE pusat_cluster SET ip1='$ip1centroid
    [$b]',ip2='$ip2centroid[$b]',ip3='$ip3centroid[$b]',ip4=
    '$ip4centroid[$b]',prestasi='$prestasicentroid[$b]',
    aktivitas='$aktivitascentroid[$b]'
     WHERE id='$b'");
}
```

4.2.1.5 Implementasi Perhitungan Error Function

Perhitungan *error function* adalah langkah terakhir di dalam Algoritma *Balance K-Means. Error function* ini digunakan untuk melihat kualitas tiap iterasi dari perhitungan *Balance K-Means*. Implementasi dari proses perhitungan *error function* ditunjukkan pada *Sourcecode* 4.5

***Sourcecode* 4.5** Implementasi *Error Function*

```
mysql_query("CREATE TABLE $ero( id int(11) primary key,
eror float)");
for($a=1;$a<=$jumdata;$a++)
{
    $pusat_cluster[$a] = mysql_query("select * from
pusat_cluster");
    $erorf[$a] = 0;
    for($i=1;$i<=$jum_cluster;$i++){
        $rro = mysql_fetch_assoc($pusat_cluster[$a]);
        $ip1pcef[$i] = $rro['ip1'];
        $ip2pcef[$i] = $rro['ip2'];
        $ip3pcef[$i] = $rro['ip3'];
        $ip4pcef[$i] = $rro['ip4'];
        $prestasipcef[$i] = $rro['prestasi'];
        $aktivitaspcef[$i] = $rro['aktivitas'];

        $xw[$i]=(pow(($ip1[$a]-
$ip1pcef[$i]),2))+(pow(($ip2[$a]-
$ip2pcef[$i]),2))+(pow(($ip3[$a]-
$ip3pcef[$i]),2))+(pow(($ip4[$a]-
$ip4pcef[$i]),2))+(pow(($prestasi[$a]-
$prestasipcef[$i]),2))+(pow(($aktivitas[$a]-
$aktivitaspcef[$i]),2));

        $erorf[$a] = $erorf[$a] + $xw[$i];
    }
    $erorf[$a] = number_format($erorf[$a],6);

    mysql_query("INSERT INTO $ero (id,eror) VALUES
('$a','$erorf[$a]')");
    }
```

4.2.1.6 Implementasi Fungsi menghitung jarak rata-rata

Pengujian terhadap hasil pemetaan menggunakan metode *Balance K-Means* dilakukan dengan menghitung nilai *Sillhouette Coefficient*. Proses awal untuk menghitung *Sillhouette Coefficient* adalah menghitung rata-rata jarak data *i* tersebut dengan semua data di *cluster* lain, kemudian diambil nilai terkecilnya Implementasi perhitungan jarak rata-rata ditunjukkan pada *Sourcecode* 4.6

***Sourcecode* 4.6** Implementasi Perhitungan jarak rata-rata

```
for($a=1;$a<=$jumdata;$a++){
    for($b=1;$b<=$jumdata;$b++){
        $datanshil[$a][$b] =
number_format(sqrt(pow(($ip1[$b]-
$ip1[$a]),2)+pow(($ip2[$b]-$ip2[$a]),2)+pow(($ip3[$b]-
$ip3[$a]),2)+pow(($ip4[$b]-$ip4[$a]),2)+pow(($prestasi[$b]-
$prestasi[$a]),2)+pow(($aktivitas[$b]-
$aktivitas[$a]),2)),6);
    }
   }
   $jum_iters = "iterasi_"."".$iteras;
   $cluster = mysql_query("select cluster from
$jum_iters");
   for($a=1;$a<=$jumdata;$a++){
    $rowss = mysql_fetch_assoc($cluster);
    $clu[$a] = $rowss['cluster'];
    for($b=1;$b<=$jumdata;$b++){
        for($c=1;$c<=$jum_cluster;$c++){
            if (!isset ($avec[$b][$c])){
            $avec[$b][$c] = 0;
            $hitungc[$b][$c] = 0;
            }
            if($clu[$a]=='c'."".$c){
            $hitungc[$b][$c]++;
            $avec[$b][$c] =
$avec[$b][$c]+$datanshil[$a][$b];
            }
        }
    }
   }
    for($b=1;$b<=$jumdata;$b++){
      for($c=1;$c<=$jum_cluster;$c++){
         $avec[$b][$c]=$avec[$b][$c]/$hitungc[$b][$c];
      }
    }

```

4.2.1.7 Implementasi Fungsi menghitung nilai $a(i)$

Proses menghitung nilai $a(i)$ pada pengujian *Sillhouette Coefficient* adalah proses untuk mendapatkan nilai terkecil di dalam satu *cluster*. Implementasi dari proses menghitung nilai $a(i)$ ditunjukkan pada *Sourcecode* 4.7

***Sourcecode* 4.7** Implementasi perhitungan nilai $a(i)$

```
for($c=1;$c<=$jum_cluster;$c++){
    for($b=1;$b<=$jumdata;$b++){
        if($clu[$b]=='c'."".$c){
          $ca[$b] = $avec[$b][$c];
        }
    }
   }
```

4.2.1.8 Implementasi Fungsi menghitung nilai $b(i)$

Proses menghitung $b(i)$pada pengujian *Sillhouette Coefficient* adalah proses untuk mendapatkan nilai rata-rata jarak data *i* dengan semua data pada *cluster* lain. Implementasi dari proses menghitung nilai $b(i)$ditunjukkan pada *Sourcecode* 4.8

***Sourcecode* 4.8** Implementasi perhitungan nilai $b(i)$

```
for($c=1;$c<=$jum_cluster;$c++){
    for($b=1;$b<=$jumdata;$b++){
        if (!isset ($cb[$b])){
            $cb[$b] = 100;
            }
        if($clu[$b]!='c'."".$c){
          $cb[$b] = min($cb[$b],$avec[$b][$c]);
        }
    }
   }
```

4.2.1.9 Implementasi Fungsi menghitung nilai *s(i)*

Proses menghitung nilai *s(i)* adalah proses untuk mendapatken nilai *Sillhouette Coefficient.* Implementasi dari proses menghitung nilai *s(i)* ditunjukkan pada *Sourcecode* 4.9

***Sourcecode* 4.9** Implementasi perhitungan nilai $s(i)$

```
for($b=1;$b<=$jumdata;$b++){
        $cs[$b]=($cb[$b]-$ca[$b])/max($cb[$b],$ca[$b]);
    }
   for($c=1;$c<=$jum_cluster;$c++){
    if (!isset ($csh[$c])){
            $csh[$c] = 0;
            $hitungs[$c] = 0;
            }
    for($b=1;$b<=$jumdata;$b++){
```

```
10          if($clu[$b]=='c'."".$c){
11              $hitungs[$c]++;
12            $csh[$c] = $csh[$c]+$cs[$b];
13          }
13       }
14     }
15     for($c=1;$c<=$jum_cluster;$c++){
16        $csh[$c]=$csh[$c]/$hitungs[$c];
17     }
18     for($c=1;$c<=$jum_cluster;$c++){
19       if ($c==1){
20              $csakhir = 0;
21              }
22        $csakhir=$csakhir+$csh[$c];
22     }
23     $csakhir=$csakhir/$jum_cluster;
24
25     mysql_query("UPDATE silhoutte SET silhoutte='$csakhir'
26
27 WHERE id='1'");
```

4.2.2 Implementasi Antarmuka Aplikasi

Antarmuka aplikasi pemetaan mahasiswa ini digunakan oleh *user* untuk berinteraksi dengan sistem perangkat lunak. Pada implementasi antarmuka perangkat lunak ini antara lain antarmuka halaman *login*, halaman *home*, halaman *input* data, halaman proses *clustering*, halaman hasil *clustering*.

4.2.2.1 Tampilan Halaman Login

Halaman *Login* merupakan halaman awal dari aplikasi pemetaan mahasiswa. Pada halaman utama hanya terdapat *form login* saja. *User* memasukkan *username* dan *password* dan menekan tombol *login* terlebih dulu untuk dapat masuk ke dalam halaman selanjutnya. Implementasi halaman *login* ditunjukkan pada Gambar 4.1

Gambar 4.1 Implementasi Halaman *Login*

4.2.2.2 Tampilan Halaman Home

Halaman *home* merupakan halaman pertama aplikasi. Pada halaman ini terdapat *form* untuk ganti *password.* Untuk dapat melakukan ganti *password user* harus memasukkan *password* lama dan *password* baru telebih dahulu dan menekan tombol ganti *password*. Implementasi halaman *home* ditunjukkan pada Gambar 4.2

Gambar 4.2 Implementasi Halaman *Home*

4.2.2.3 Tampilan Halaman Input Data

Halaman *input* data merupakan halaman untuk memasukkan data mahasiswa yang akan dipetakkan. Di dalam gambar dibawah ini menunjukkan data *inputan* pemetaan mahasiswa berupa NIM, IP1,IP2, IP3, IP4, Prestasi serta Aktivitas. Implementasi halaman *Input* data ditunjukkan pada Gambar 4.3.

Gambar 4.3 Implementasi Halaman *Input* Data

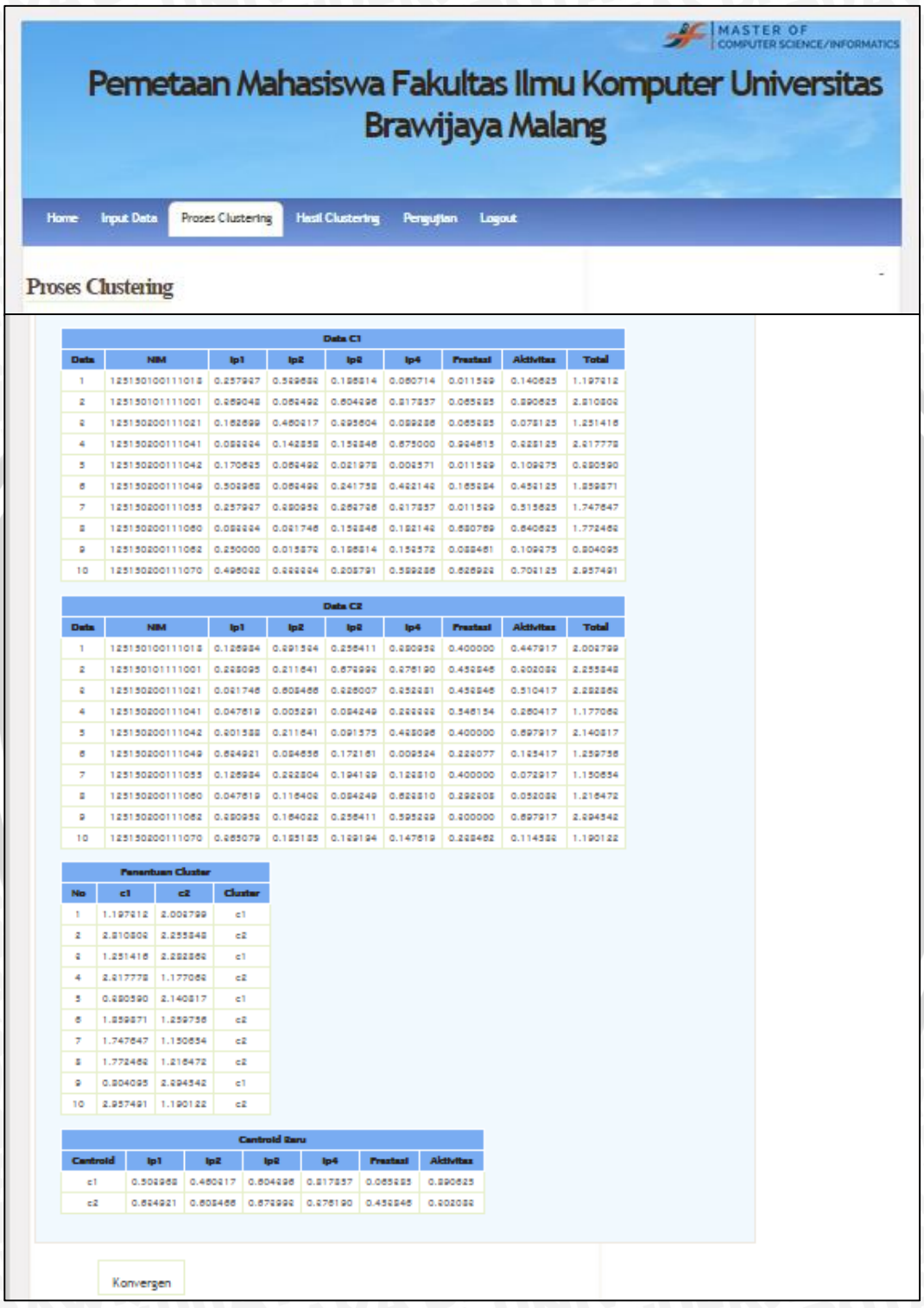
4.2.2.4 Tampilan Halaman Clustering

Halaman *clustering* merupakan halaman untuk memasukkan jumlah *cluster* yang di inginkan oleh *user*. Untuk dapat menggunakan halaman ini *user* harus menginputkan jumlah *cluster* yang diinginkan dan menekan tombol proses. Implementasi halaman *clustering* ditunjukkan pada Gambar 4.4

Gambar 4.4 Implementasi Halaman *Clustering*

4.2.2.5 Tampilan Halaman Proses Clustering

Halaman proses *clustering* merupakan halaman yang menampilkan proses perhitungan menggunakan metode *Balance K-Means*. Di dalam gambar di bawah ini menampilkan hasil perhitungan jarak data ke *centroid* yaitu C2 dan C3 selain itu menampilkan hasil proses *clustering* dan hasil perhitungan *centroid* baru. Implementasi halaman Proses *Clustering* ditunjukkan pada Gambar 4.5



Data C1

Data	NIM	Ip1	Ip2	Ip3	Ip4	Prestasi	Aktivitas	Total
1	125150100111018	0.257927	0.529688	0.196814	0.050714	0.011529	0.140625	1.197312
2	125150101111001	0.859048	0.058492	0.804496	0.217857	0.065885	0.890625	2.810809
3	125150200111021	0.182699	0.460317	0.295604	0.099286	0.065885	0.078125	1.251416
4	125150200111041	0.088334	0.142858	0.158846	0.675000	0.994615	0.928125	2.917778
5	125150200111042	0.170625	0.058492	0.021978	0.002571	0.011529	0.109375	0.380590
6	125150200111049	0.508968	0.058492	0.241758	0.482148	0.165884	0.453125	1.859871
7	125150200111055	0.257927	0.830952	0.258748	0.217857	0.011529	0.515625	1.747647
8	125150200111060	0.088334	0.021748	0.158846	0.182148	0.650769	0.640625	1.772468
9	125150200111062	0.250000	0.015872	0.196814	0.152572	0.088461	0.109375	0.804095
10	125150200111070	0.495032	0.333334	0.208791	0.599286	0.626923	0.703125	2.957491

Data C2

Data	NIM	Ip1	Ip2	Ip3	Ip4	Prestasi	Aktivitas	Total
1	125150100111018	0.126984	0.091534	0.256411	0.200952	0.400000	0.447917	2.002799
2	125150101111001	0.238095	0.211641	0.879999	0.276190	0.452845	0.902083	2.255848
3	125150200111021	0.021748	0.608466	0.226007	0.952381	0.452845	0.510417	2.282866
4	125150200111041	0.047619	0.005291	0.054249	0.333333	0.546154	0.260417	1.177063
5	125150200111042	0.021368	0.211641	0.091575	0.466096	0.400000	0.697917	2.140817
6	125150200111049	0.624921	0.084656	0.172161	0.009524	0.220077	0.125417	1.259756
7	125150200111055	0.126984	0.222304	0.194139	0.123810	0.400000	0.072917	1.150654
8	125150200111060	0.047619	0.116402	0.054249	0.622810	0.292308	0.052083	1.216472
9	125150200111062	0.200952	0.164022	0.256411	0.595229	0.200000	0.697917	2.094542
10	125150200111070	0.265079	0.185185	0.169194	0.147619	0.238462	0.114583	1.190122

Penentuan Cluster

No	c1	c2	Cluster
1	1.197312	2.002799	c1
2	2.810809	2.255848	c2
3	1.251416	2.282866	c1
4	2.917778	1.177063	c2
5	0.380590	2.140817	c1
6	1.859871	1.259756	c2
7	1.747647	1.150654	c2
8	1.772468	1.216472	c2
9	0.804095	2.094542	c1
10	2.957491	1.190122	c2

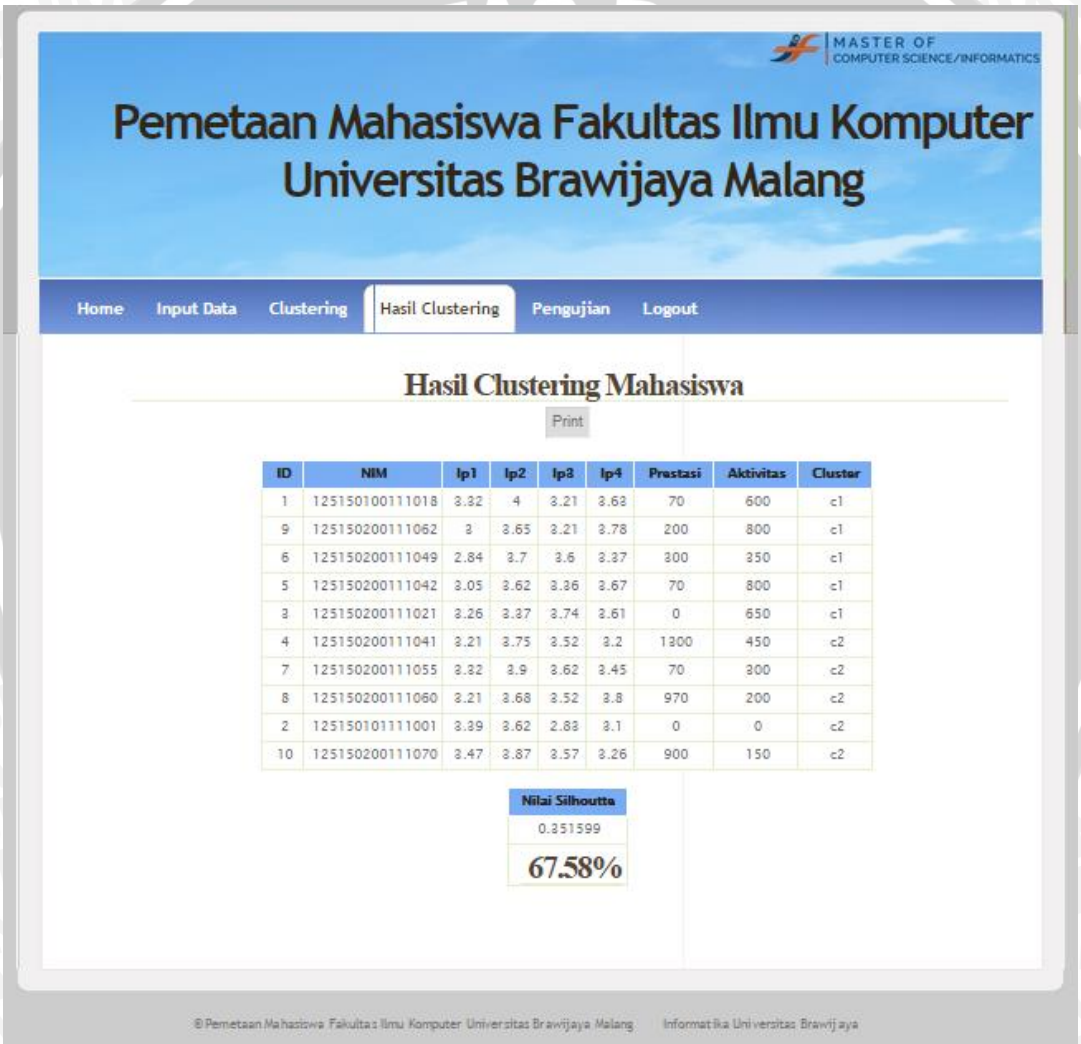
Centroid Baru

Centroid	Ip1	Ip2	Ip3	Ip4	Prestasi	Aktivitas
c1	0.508968	0.460317	0.804496	0.217857	0.065885	0.890625
c2	0.624921	0.608466	0.879999	0.276190	0.452845	0.902083

Gambar 4.5 Implementasi Halaman Proses *Clustering*

4.2.2.6 Tampilan Halaman Hasil *Clustering*

Halaman hasil *clustering* merupakan halaman yang menampilkan hasil perhitungan menggunakan metode *Balance K-Means*. Pada halaman ini juga ditampilkan nilai kualitas *cluster* yang dihitung menggunakan metode *Silhoutte Coefficient*. Gambar di bawah ini menunjukan hasil *clustering* yang mana data yang telah di *inputkan* di *cluster* menjadi 2 *cluster*. Implementasi halaman Hasil *Clustering* ditunjukkan pada Gambar 4.6.



Gambar 4.6 Implementasi Halaman Hasil *Clustering*

4.2.2.7 Tampilan Halaman Pengujian

Pada halaman ini akan ditampilkan hasil pengujian yang telah dilakukan berdasarkan pada skenario pengujian yang telah dibuat pada bab perancangan. Pada halaman pengujian menampilkan pengujian pengurangan parameter, pengujian jumlah *cluster* dan pengujian perbandingan metode *Balance K-Means* dan *K-Means*. Implementasi halaman pengujian ditunjukkan pada Gambar 4.7.

Gambar 4.7 Implementasi Halaman Pengujian

BAB V
PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis terhadap aplikasi pemetaan mahasiswa menggunakan metode *Balance K-Means*. Pengujian yang dilakukan dalam bab ini meliputi pengujian terhadap jumlah *cluster*, pengujian parameter, pengujian perbandingan dengan *K-Means* dan pengujian tingkat akurasi. Pengujian yang dilakukan bersifat *sequential*, artinya hasil terbaik satu pengujian digunakan untuk pengujian berikutnya.

5.1 Pengujian Jumlah Cluster Terbaik

Pengujian jumlah *cluster* terbaik digunakan untuk mengetahui berapa jumlah *cluster* yang menghasilkan nilai kualitas *cluster* tertinggi dalam proses pemetaan mahasiswa. Nilai kualitas *cluster* didapatkan dengan melakukan evaluasi *cluster* menggunakan metode *Silhoutte Coefficient.*

Pengujian jumlah *cluster* terbaik pada aplikasi pemetaan mahasiswa ini menggunakan data mahasiswa semester 5 sebanyak 110 data dengan *run* program sebanyak 10 kali. Jumlah *cluster* yang digunakan untuk pengujian ini adalah 2,3,4,5 dan 6. Tabel 5.1, 5.2, 5.3, 5.4, 5.5 dan 5.6 menunjukkan hasil pengujian pengaruh jumlah *cluster* yang digunakan menggunakan metode *Silhoutte Coefficient*. Metode *Silhoutte Coefficient* memiliki range nilai -1 sampai dengan 1 dimana, semakin mendekati -1 maka nilai kualitas *cluster*-nya semakin buruk dan semakin mendekati 1 maka nilai kualitas *cluster*-nya semakin baik.

Tabel 5.1 Hasil Pengujian jumlah *cluster* 2

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
1	8	0.284056
2	9	0.286676
3	4	0.286676
4	7	0.284056
5	4	0.284056

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
6	8	0.284056
7	5	0.284056
8	3	0.284056
9	7	0.284056
10	9	0.284056
Rata-rata		0.28458

Tabel 5.2 Hasil Pengujian jumlah *cluster* 3

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
1	7	0.335916
2	13	0.335916
3	4	0.335068
4	3	0.340799
5	4	0.319725
6	5	0.186263
7	4	0.273715
8	3	0.298143
9	4	0.327285
10	6	0.335068
Rata-rata		0.30879

Tabel 5.3 Hasil Pengujian jumlah *cluster* 4

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
1	7	0.200442
2	5	0.366836
3	8	0.239266
4	12	0.356688
5	8	0.230279
6	6	0.335554
7	5	0.364978
8	5	0.299075
9	9	0.363721
10	7	0.356785
Rata-rata		0.311362

Tabel 5.4 Hasil Pengujian jumlah *cluster* 5

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
1	10	0.275464
2	7	0.259624
3	9	0.313
4	10	0.374839
5	12	0.322387
6	14	0.423601
7	20	0.301884
8	7	0.336304
9	4	0.25289
10	11	0.301884
Rata-rata		0.316188

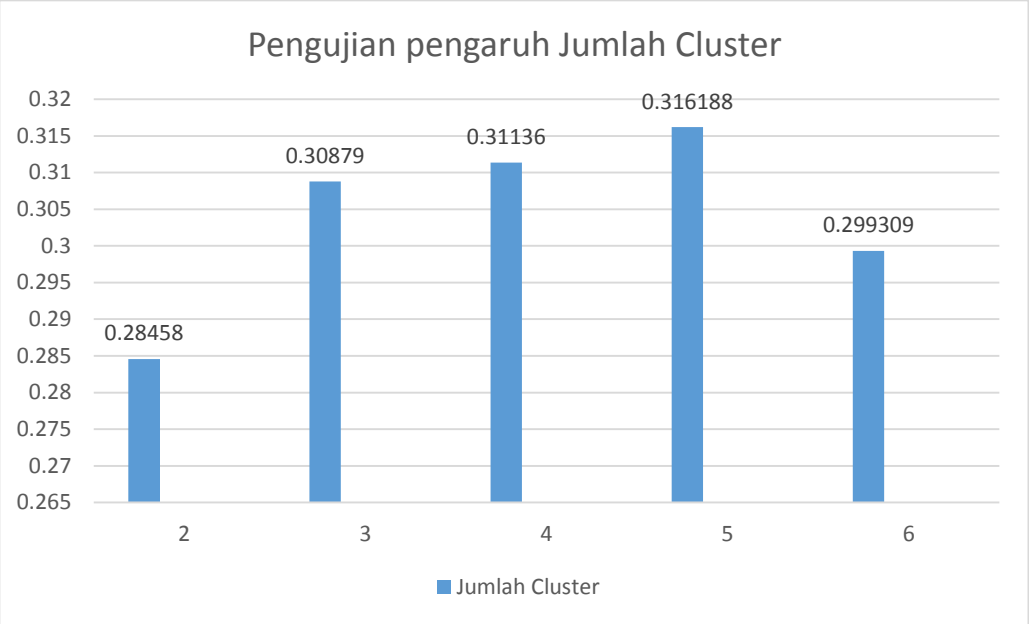
Tabel 5.5 Hasil Pengujian jumlah cluster 6

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
1	5	0.344716
2	8	0.307608
3	9	0.280777
4	11	0.258824
5	11	0.302355
6	11	0.330083
7	11	0.313725
8	8	0.326594
9	6	0.224842
10	12	0.303569
Rata-rata		0.299309

Tabel 5.6 Hasil Pengujian Jumlah *Cluster*

No.	Jumlah *Cluster*	Hasil (*Silhoutte Coefficient*)
1	2	0.28458
2	3	0.30879
3	4	0.311362
4	5	0.316188
5	6	0.299309

Sedangkan gambar 5.1 berikut ini merupakan grafik perbandingan hasil pengujian jumlah *cluster* terbaik.



Gambar 5.1 Grafik Perbandingan Hasil Pengujian Jumlah *Cluster* Terbaik

Gambar 5.1 adalah grafik perbandingan hasil pengujian jumlah *cluster* dapat dilihat bahwa dengan menggunakan jumlah *cluster* sebanyak 5 *cluster* didapatkan nilai *Silhoutte Coefficient* yang tertinggi. Hal ini berarti tingkat kemiripan antar data di dalam satu *cluster* semakin baik dan tingkat kemiripan data antar *cluster* semakin buruk.

Apabila menggunakan jumlah *cluster* lainnya, yaitu : 2,3,4, dan 6 maka didapatkan nilai kualitas *cluster* yang cenderung lebih rendah. Hal ini memperlihatkan bahwa tingkat kemiripan data dalam satu *cluster* cenderung semakin buruk dan tingkat kemiripan data antar *cluster* cenderung semakin baik.

5.2 Pengujian Jumlah Parameter

Pengujian pengaruh parameter data yang diproses, digunakan untuk mengetahui pengaruh parameter terhadap nilai kualitas *cluster* dari hasil pemetaan menggunakan metode *Balance K-Means*. Kriteria pemilihan parameter yang diuji adalah :

1. Menggunakan semua parameter yang ada (6 parameter).
2. Menggunakan 5 parameter dengan pengurangan parameter aktivitas.
3. Menggunakan 5 parameter dengan pengurangan parameter prestasi.
4. Menggunakan 4 parameter dengan pengurangan parameter aktivitas dan prestasi.

Pengujian jumlah parameter pada aplikasi pemetaan mahasiswa ini dilakukan menggunakan data mahasiswa semester 5 sebanyak 110 data dengan *run* program sebanyak 10 kali. *Cluster* yang digunakan dalam pengujian ini adalah 5 *cluster* dimana *cluster* ini merupakan *cluster* yang terbaik.

Pada kondisi 2, dimana parameter aktivitas tidak diikutsertakan di dalam perhitungan dikarenakan sampai saat ini data pasti untuk parameter aktivitas belum tersedia di FILKOM, sehingga data yang terdapat di dalam penelitian ini didapatkan dari hasil *scoring* kuisioner yang telah disebarkan.

Pada kondisi 3, dimana parameter prestasi tidak diikutsertakan di dalam perhitungan dikarenakan sampai saat ini data pasti untuk parameter prestasi belum tersedia di FILKOM, sehingga data yang terdapat di dalam penelitian ini didapatkan dari hasil *scoring* kuisioner yang telah disebarkan. Selain itu dari seluruh data mahasiswa yang terkumpul hanya sedikit mahasiswa yang memiliki parameter prestasi. Sebagai contoh, dari 110 data mahasiswa yang terkumpul hanya 43 mahasiswa yang memiliki parameter prestasi.

Pada kondisi 4, dimana parameter aktivitas dan prestasi tidak diikutsertakan di dalam perhitungan, dan hanya menggunakan 4 parameter saja yaitu :IP1, IP2, IP3, dan IP4 dikarenakan sampai saat ini data pasti untuk parameter aktivitas dan prestasi belum tersedia di FILKOM, sehingga data yang terdapat di dalam penelitian ini didapatkan dari hasil scoring kuisioner yang telah disebarkan. Tabel 5.7, 5.8, 5.9, 5.10, dan 5.11 menunjukkan hasil dari pengujian jumlah parameter.

Tabel 5.7 Pengujian menggunakan 6 Parameter

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
1	10	0.30347
2	7	0.452919
3	9	0.305536
4	10	0.473009
5	12	0.275008
6	14	0.41433
7	20	0.286914
8	7	0.258564
9	4	0.279677
10	11	0.30347
Rata-rata		0.3352897

Tabel 5.8 Pengujian tanpa menggunakan Parameter Aktivitas

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
1	4	0.247753
2	14	0.338052
3	6	0.316702
4	13	0.352648
5	10	0.36986
6	7	0.352648
7	15	0.32374
8	4	0.285632
9	7	0.36986
10	6	0.316702
Rata-rata		0.3273597

Tabel 5.9 Pengujian tanpa menggunakan Parameter Prestasi

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
1	12	0.299309
2	6	0.341106
3	4	0.350276
4	12	0.502082
5	7	0.320747

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
6	11	0.375635
7	6	0.336569
8	9	0.331889
9	15	0.31654
10	6	0.33375
Rata-rata		0.3507903

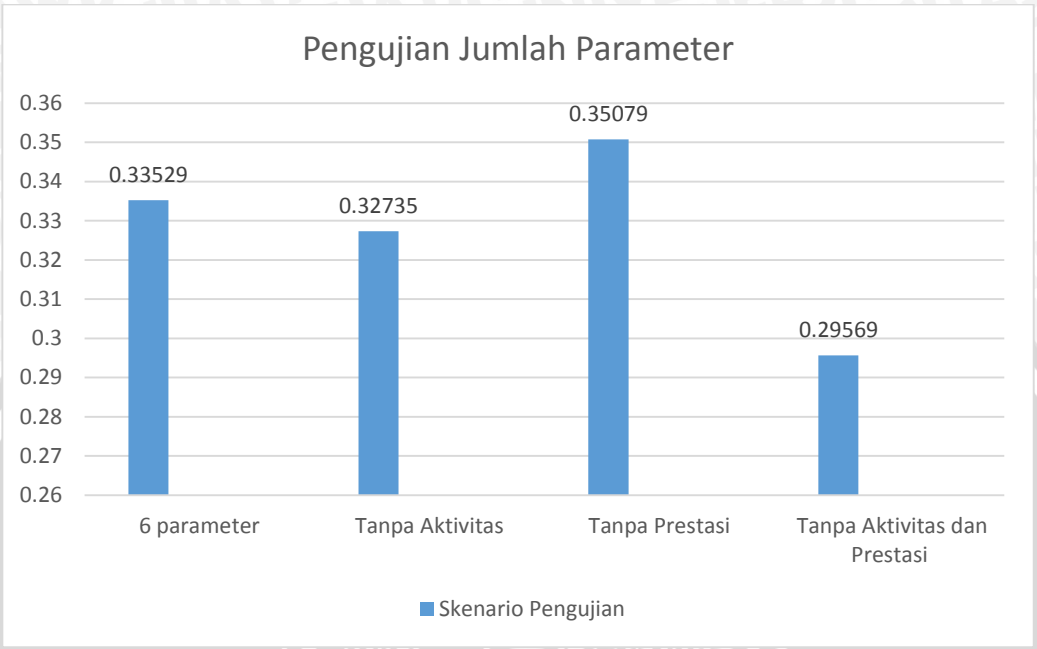
Tabel 5.10 Pengujian tanpa menggunakan Parameter Prestasi dan Aktivitas

Run ke-	Jumlah Iterasi	Hasil (*Silhoutte Coefficient*)
1	6	0.349091
2	9	0.258976
3	8	0.30404
4	10	0.24103
5	6	0.267219
6	6	0.349091
7	4	0.253528
8	7	0.320922
9	15	0.345424
10	7	0.267674
Rata-rata		0.2956995

Tabel 5.11 Hasil Pengujian Jumlah Parameter

No	Skenario Pengujian	Nilai (*Silhouette Coefficient*)
1	Menggunakan 6 parameter	0.33529
2	Menggunakan 5 parameter (aktivitas dihilangkan)	0.32735
3	Menggunakan 5 parameter (prestasi dihilangkan)	0.35079
4	Menggunakan 4 parameter (prestasi dan aktivitas dihilangkan)	0.29569

Sedangkan Gambar 5.2 berikut ini merupakan grafik perbandingan hasil pengujian jumlah parameter.



Gambar 5.2 Grafik Perbandingan Hasil Pengujian Jumlah Parameter

Berdasarkan pada grafik perbandingan hasil pengujian jumlah parameter yang ditunjukkan pada Gambar 5.2 diketahui bahwasannya pada pengujian kali ini nilai *silhouette coefficient* tanpa menggunakan parameter prestasi mendapatkan nilai yang tertinggi, hal ini dikarenakan pada parameter prestasi pada 110 data yang ada memiliki nilai rentang perbedaan yang jauh, seperti pada data 110 mahasiswa, yang memiliki prestasi hanya 43 mahasiswa.

Pengujian parameter ini dilakukan bukan untuk mencari nilai parameter yang terbaik, tetapi di dalam pengujian ini digunakan untuk mengetahui bahwa sistem ini dapat di pakai pada pengetahuan kedepannya seperti pencarian mahasiswa yang menerima beasiswa, asisten atau *student employe* dimana parameter yang ada bisa di tambahkan atau dikurangi sesuai dengan kebutuan pengambilan keputusan yang akan dilakukan.

5.3 Pengujian Perbandingan Metode

Pengujian perbandingan metode ini digunakan untuk mengetahui apakah metode *Balance K-Means* lebih baik untuk digunakan di dalam aplikasi Pemetaan Mahasiswa dibandingkan dengan metode *K-Means*. Pengujian ini dilakukan untuk membandingkan kualitas *cluster* metode *Balance K-Means* dengan *K-Means*. Pada pengujian ini menggunakan 110 data mahasiswa dengan jumlah *cluster* 5 (Jumlah *Cluster* Terbaik), jumlah parameter 5 (Jumlah Parameter Terbaik) dan dengan 10 kali *run* program. Hasil pengujian dengan metode *Balance K-Means* dan *K-Means* ditujukkan pada Tabel 5.12 dan 5.13.

Tabel 5.12 Pengujian dengan *Metode Balance K-Means*

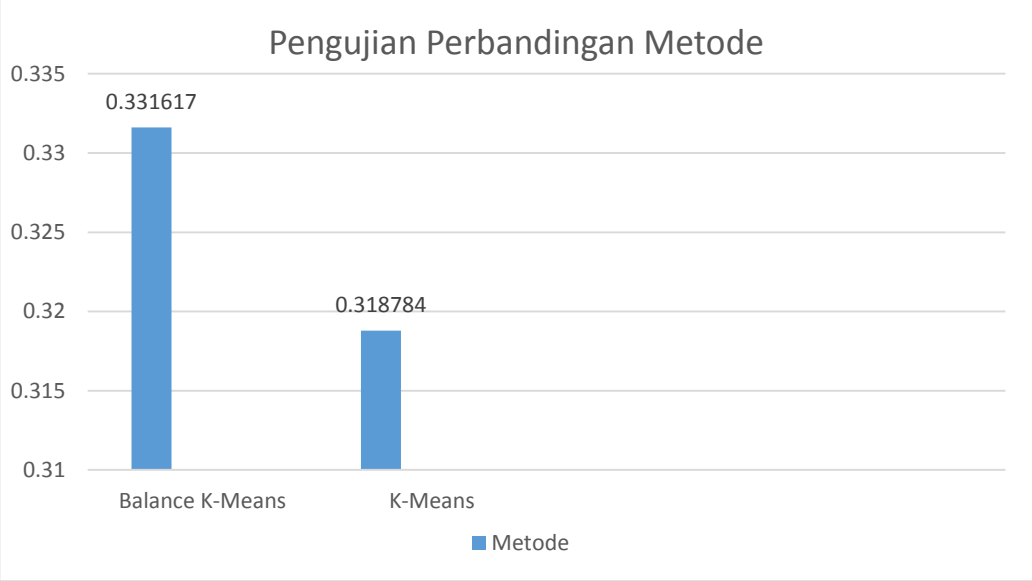
Run ke-	Jumlah Iterasi	Nilai Silhoutte Coefficient	
1	10	0.327505	66.38
2	9	0.327189	66.36
3	8	0.338293	66.91
4	14	0.330108	66.51
5	11	0.353053	67.65
6	9	0.318075	65.9
7	8	0.31933	65.97
8	7	0.321142	66.06
9	11	0.337801	66.89
10	6	0.34367	67.18
Rata-rata		0.331617	66.581

Tabel 5.13 Pengujian dengan *Metode K-Means*

Run ke-	Jumlah Iterasi	Nilai Silhoutte Coefficient	
1	5	0.295153	64.76
2	12	0.338414	66.92
3	4	0.29907	64.95
4	7	0.313211	65.66
5	17	0.338795	66.94
6	6	0.321724	66.09
7	10	0.32412	66.21

Run ke-	Jumlah Iterasi	Nilai Silhoutte	Nilai Silhoutte
8	9	0.315403	65.77
9	16	0.338795	66.94
10	11	0.30315	65.16
Rata-rata		0.318784	65.94

Sedangkan gambar 5.3 berikut ini merupakan grafik perbandingan Metode.



Gambar 5.3 Grafik Perbandingan Metode

Berdasarkan pada grafik perbandingan metode yang ditunjukkan pada Gambar 5.3 diketahui bahwa nilai kualitas *cluster* yang dihasilkan oleh metode *Balance K-Means* cenderung lebih baik dibandingkan dengan metode *K-Means*. Metode *Balance K-Means* menghasilkan nilai kualitas *cluster* sebesar 0.331617 atau jika dilihat pada rentang nilai -1 s/d 1 setara dengan 66.58%. Sedangkan metode *K-Means* menghasilkan nilai kualitas *cluster* sebesar 0.318784 atau jika dilihat pada rentang nilai -1 s/d 1 setara dengan 65.94%.

5.4 Pengujian Tingkat Akurasi

Pengujian tingkat akurasi ini digunakan untuk mengetahui berapa tingkat akurasi Metode *Balance K-Means*. Pada pengujian ini dilakukan dengan menggunakan dataset

iris yang terdapat pada UCI dataset dengan jumlah 150 data (3 *cluster*) dengan 10 kali *run* program. Dataset iris digunakan karena rentang perbedaan nilai pada masing-masing parameter dataset iris cukup jelas, sehingga mudah untuk menemukan *cluster* yang tepat. Hasil pengujiannya ditunjukkan pada Tabel 5.14 dan 5.15

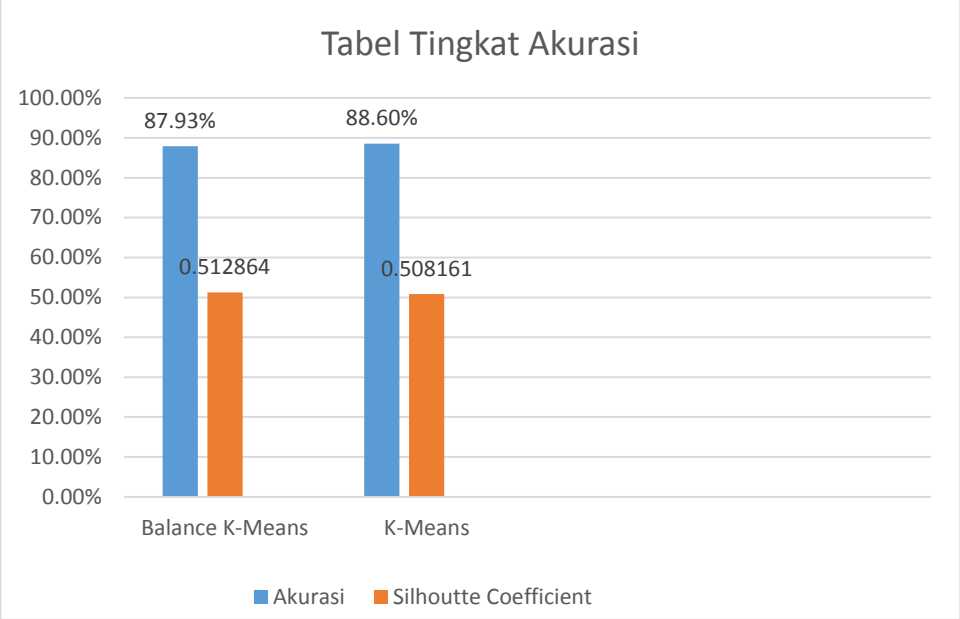
Tabel 5.14 Pengujian dengan Metode Balance K-Means pada Iris Dataset

Run ke-	Jumlah Iterasi	Akurasi	Nilai Silhoutte Coefficient
1	4	88.66%	0.515178
2	10	88.00%	0.511708
3	8	88.66%	0.511708
4	5	87.33%	0.511708
5	5	87.33%	0.511708
6	7	88.00%	0.511708
7	5	88%	0.515178
8	7	88%	0.511708
9	5	88%	0.515178
10	7	87.33%	0.511708
Rata-rata		87.93%	0.512864667

Tabel 5. 15 Pengujian dengan Metode K-Means pada Iris Dataset

Run ke-	Jumlah Iterasi	Akurasi	Nilai Silhoutte Coefficient
1	4	89.33%	0.51532
2	8	88.66%	0.515132
3	8	88.00%	0.494033
4	6	88.66%	0.515132
5	8	88.66%	0.515132
6	5	88.00%	0.494033
7	5	89%	0.515132
8	3	88.66%	0.494033
9	5	88.66%	0.515132
10	6	88.66%	0.494033
Rata-rata		88.60%	0.508161667

Sedangkan gambar 5.4 berikut ini merupakan grafik pengujian tingkat akurasi.



Gambar 5.4 Grafik Pengujian Tingkat Akurasi

Berdasarkan pada grafik pengujian tingkat akurasi yang ditujukkan pada gambar 5.4 dapat dilihat bahwa ketika menggunakan dataset Iris, metode *Balance K-Means* dapat menghasilkan nilai akurasi yang lebih rendah dibandingkan dengan metode *K-Means*, sedangkan nilai kualitas *cluster* yang dihasilkan lebih tinggi dari *K-Means*. Hal ini mungkin dikarenakan penentuan pusat awal *cluster* yang dilakukan secara *random*, sehingga dapat mempengaruhi pencarian *cluster* yang tepat.

Pada pengujian dengan dataset iris ini juga digunakan untuk membuktikan kebenaran dari algoritma yang telah diterapkan, nilai *silhouette* yang terbentuk dalam pengujian ini adalah 0.512864667 atau jika dilihat pada rentang nilai -1 s/d 1 setara dengan 75.64%. Hal ini membuktikan bahwa algoritma yang telah di terapkan sudah tepat.

Metode *Balance K-Means* kemungkinan juga lebih cocok digunakan pada data yang memiliki rentang nilai yang berbeda pada setiap parameternya, seperti pada data mahasiswa yang salah satu parameternya memiliki rentang nilai yang berbeda.

BAB VI
PENUTUP

3.2 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Metode *Balance K-Means* dapat diimplementasikan untuk melakukan pemetaan mahasiswa. Walaupun nilai *silhouette coefficient Balance K-Means* secara rata-rata lebih tinggi tetapi nilai akurasi dari *Balance K-Means* secara rata-rata lebih rendah dari pada *K-Means* yaitu 87.93% dan 88.60% nilai ini didapatkan setelah melakukan pengujian tingkat akurasi menggunakan dataset Iris.
2. Hasil pemetaan dengan metode *Balance K-Means* memiliki nilai kualitas *cluster* yang relatif lebih baik dibandingkan dengan metode *K-Means* dalam 10 kali *run* program. Hal ini dibuktikan metode *Balance K-Means* menghasilkan nilai kualitas *cluster* sebesar 0.331617 atau jika dilihat pada rentang nilai -1 s/d 1 setara dengan 66.581%. Sedangkan metode *K-Means* menghasilkan nilai kualitas *cluster* sebesar 0.318784 atau jika dilihat pada rentang nilai -1 s/d 1 setara dengan 65.94%.
3. Penambahan atau pengurangan jumlah *cluster* yang akan dibentuk berpengaruh terhadap nilai kualitas *cluster* yang dihasilkan. Hal ini dibuktikan berdasarkan pengujian yang telah dilakukan. Jumlah *cluster* 5 merupakan nilai yang tertinggi dengan nilai *silhouette coefficient* 0.3161, sedangkan jumlah *cluster* 2 merupakan jumlah *cluster* yang terendah dengan nilai *silhouette coefficient* 0.2845. Pengurangan jumlah parameter juga berpengaruh terhadap kualitas *cluster*. Dari hasil pengujian nilai *silhouette coefficient* terbaik di dapatkan pada data dengan parameter prestasi yang dihapus yaitu sebesar 0.3507.

3.3 Saran

Pemetaan mahasiswa menggunakan metode *Balance K-Means* ini masih memiliki beberapa kekurangan. Saran yang dapat diberikan untuk pengembangan penelitian selanjutnya, antara lain :

1. Untuk pengembangan lebih lanjut, aplikasi ini dapat dikembangkan dengan metode dari pengembangan *K-Means* yang lain, karena dalam *Balance K-Means* ini masih mempunyai kelemahan yang diakibatkan oleh penentuan pusat awal *cluster*. Hasil *cluster* yang terbentuk dari metode *Balance K-means* ini sangatlah tergantung pada inisiasi nilai pusat awal *cluster* yang diberikan. Hal ini menyebabkan hasil *clusternya* berupa solusi yang sifatnya *local optimal*. Metode lain yang bisa digunakan adalah metode mengkombinasikan antara Algoritma Lebah dan *K-Means*.
2. Penelitian ini bisa dikembangkan lebih lanjut, hasil pengelompokkan data dapat dilakukan *treatment* data mahasiswa agar bisa digunakan dalam pencarian data, untuk keperluan pemilihan asisten, peserta lomba, dan *student employed*.

DAFTAR PUSTAKA

[AIF-10] Albar., Ismail., & Fibriyanti. 2010. Identifikasi Dengan Menggunakan Algoritma K-Means Pada Plat Kendaraan. ISSN : 1858-3709 Volume 6, Nomor 1, Oktober 2010.

[AND-07] Andayani, Sri. 2007, "Pembentukan Cluster dalam Knowledge Discovery in Database dengan Algoritma K-Means". FMIPA UNY, Yogyakarta.

[ENM-13] Ediyanto. Novitasari Mara,.,Muhlasah. Satyahadew., Neva. 2013. Pengklasifikasian Karakteristik Dengan Metode K-Means Cluster Analysis. Buletin Ilmiah Mat. Stat. dan Terapannya (Bimaster) Volume 02 , No. 2

[HRM-14] Handoyo Rendy., M R.Rumani.,& Surya Michrandi N. 2014. Perbandingan Metode Clustering menggunakan Metode Single Linkage dan K-means pada Pengelompokan Dokumen ISSN.1412-0100 vol 15, no 2, Oktober 2014.

[LAR-05:147] Larose, Daniel T.2005. "Discovering Knowledge in Data: an Introduction to Data Mining ". Jhon &Wiley & Sons, Inc. New Jersey

[MAE-12] M Abu Tair,M., M.El-Haless,Alaa.2012.Mining Educational Data to Improve Students' Performance:A Case Studi.International Journal of Information and Communication Technology Research, Volume 2 No. 2, February 2012.

[MSC-11] Anonim. 2011. Metode Silhoutte Coeffisien. [terhubung berkala].https://lookmylife.wordpress.com/2011/10/03/metode-silhoutte-coeffisien/.[24 Maret 2015].

[PTI-12] PTIIK-UB. 2012. Pedoman Pendidikan Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Malang : Universitas Brawijaya.

[RSS-13] Ridwan, M., Suyono, H. &Sarosa, M. 2013.Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier. Jurnal EECCIS Vol.7, No. 1, Juni 2013.

[WID-11] Widiarta, I Made. 2011. Studi Komparasi Metode Klasterisasi Data K-Means dan K-Harmonik Means. Jurnal Ilmu Komputer Vol. 4, No. 1.

[WQZ-14] Wang Hongjun. Qi Jianhuai. Zheng Weifan., & Wang Mingwen. 2014. Balance K-means Algorithm. Jurnal IEEE 978-1-4244-4507-3/09.

LAMPIRAN