IMPLEMENTASI METODE *IMPROVEMENT K-MEANS* DENGAN INISIALISASI *CENTROID* MENGGUNAKAN *WEIGHTED AVERAGE* PADA *DATASET* IRIS DENGAN PEMBOBOTAN OWA

SKRIPSI

Diajukan untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana Komputer

Disusun Oleh :

HERLAMBANG PRIYO UTOMO

NIM. 115060800111102

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2015

LEMBAR PERSETUJUAN

IMPLEMENTASI METODE *IMPROVEMENT K-MEANS* DENGAN INISIALISASI *CENTROID* MENGGUNAKAN *WEIGHTED AVERAGE* PADA *DATASET* IRIS DENGAN PEMBOBOTAN OWA

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI

Diajukan untuk Memenuhi Persyaratan Memperoleh Gelar Sarjana Komputer

Disusun Oleh :

HERLAMBANG PRIYO UTOMO

115060800111102

Skripsi ini telah disetujui oleh dosen pembimbing pada Tanggal 10 Juli 2015

Dosen Pembimbing I	Dosen Pembimbing II
Dian Eka Ratnawati, S.Si., M.Kom.	**Budi Darma Setiwan, S.Kom., M.Cs.**
NIP. 19730619 200212 2 001	**NIP. 19841015 201404 1 002**

LEMBAR PENGESAHAN

IMPLEMENTASI METODE *IMPROVEMENT K-MEANS* DENGAN INISIALISASI *CENTROID* MENGGUNAKAN *WEIGHTED AVERAGE* PADA *DATASET* IRIS DENGAN PEMBOBOTAN OWA

SKRIPSI

LABORATORIUM KOMPUTASI CERDAS DAN VISUALISASI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

HERLAMBANG PRIYO UTOMO

NIM. 115060800111102

Skripsi ini telah diuji dan dinyatakan lulus pada tanggal 7 Agustus 2015

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, 10 Juli 2015

Mahasiswa

Herlambang Priyo Utomo
NIM. 115060800111102

KATA PENGANTAR

Puji dan syukur di panjatkan kehadirat Allah SWT yang telah melimpahkan berkat, kemudahan dan karunia-Nya serta kelancaran sehingga penulis dapat menyelesaikan skripsi dengan judul "Implementasi Metode *Improvement K-Means* Dengan Inisialisasi *Centroid* Menggunakan *Weighted Average* Pada *Dataset* Iris Dengan Pembobotan OWA)".

Pada kesempatan ini, penulis mengucapkan banyak terima kasih karena dalam penyusunan skripsi ini penulis telah mendapat bantuan dan dorongan baik lahir maupun batin dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada :

1. Ibu Dian Eka Ratnawati, S.Si., M.Kom. dan bapak Budi Darma Setiawan, S.Kom., M.Cs selaku dosen pembimbing tugas akhir penulis.
2. Bapak Ir. Sutrisno, M.T, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Himawat Aryadita, S.T, M.Sc, dan Bapak Eddy Santoso, S.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2 dan Wakil Ketua 3 Fakultas Ilmu Komputer Universitas Brawijaya
3. Bapak Drs. Marji, MT dan Bapak Issa Arwani, S.Kom., M.Sc selaku Ketua dan Sekretaris Program Studi Informatika.
4. Bapak / Ibu Dosen, Staff Administrasi dan Perpustakaan Fakultas Ilmu Komputer Universitas Brawijaya.
5. Ayahku Kardi Prabowo, Ibuku Emi Suhartatik, Tante ku Wiwik Utami, Kakakku Rulani Indra Pratiwi, Adikku Kartika Anggraini beserta keluarga besarku yang tiada henti-hentinya memberikan doa, semangat dan kasih sayang demi terselesainya skripsi.
6. Angga Huda, Rinadewi dan Vina sebagai teman seperjuangan yang sudah mau menemani di keadan susah, senang dan berbagi permasalahan dan solusi dalam menyelesaikan berbagai permasalahan di tugas akhir ini.
7. Rachmawati, Fitarina, Vita, Rasita, Fenty, Luluk, Itto, Valerian, yang sudah memberi bantuan dan saran dalam menyelesaikan skripsi ini. Dan lutvi yang bisa memberi motivasi-motivasi yang mampu menambah semangat.

8. Tusi, Tya, dan Emak yang sudah berkenan meminjamkan laptop sehingga skripsi ini dapat terselesaikan.
9. Rizky Karisma, Putri, Niki, Weni, Agung, Falah, Ega, Dyang, Dina, Nancy PSDM Fams terima kasih atas doa dan dukungannya.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun, sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 10 Juli 2015

Penulis

ABSTRAK

Herlambang, 2015. Implementasi Metode *Improvement K-Means* Dengan Inisialisasi *Centroid* Menggunakan *Weighted Average* Pada *Dataset* Iris Dengan Pembobotan OWA

Dosen Pembimbing : Dian Eka Ratnawati, S.Si., M.Kom. dan Budi Darma Setiawan, S.Kom., M.Cs.

K-Means merupakan metode *clustering* yang populer namun memiliki kelemahan berupa inisial *centroid* ditentukan secara random, hal ini membuat hasil *cluster* menjadi tidak konsisten. *K-Means* dengan *weighted average* merupakan algoritma *clustering* yang menghindari penentuan inisial *centroid* yang *random* sehingga menghasilkan *cluster* yang konsisten. Namun, nilai bobot *attribute* cenderung ditentukan bebas oleh pengguna yang apabila tidak sesuai menghasilkan *cluster* yang buruk. Maka dari itu diperlukan bobot yang sesuai dengan distribusi data. Penelitian ini menggunakan *dataset* iris yang nilai bobotnya diperoleh dari perhitungan OWA. Dari pengujian bobot yang dilakukan, penggunaan bobot yang berbeda menghasilkan *cluster* yang berbeda. Dataset iris merupakan data yang sudah memiliki label kelas, sehingga dapat dilakukan pengujian *silhouette coefficient* dan pengujian akurasi dengan menggunakan dua metode perhitungan jarak, *euclidean* dan *manhattan*. Pada pengujian *silhouette coefficient improve K-Means* memiliki hasil yang lebih baik, selain itu nilai *silhouette coefficient euclidean* lebih baik dari *manhattan*. Pada pengujian akurasi *K-Means* konvensional memiliki akurasi yang lebih baik, di mana akurasi dengan *manhattan* lebih baik daripada *euclidean*. Sebaliknya, meskipun *improve K-Means* memiliki akurasi yang lebih rendah, namun akurasi dengan *euclidean* lebih baik daripada saat menggunakan *manhattan*. Berdasarkan pengujian didapati, *improve K-Means* memiliki hasil yang konsisten, namun tidak bisa mencapai hasil yang optimum. Berbeda dengan *K-Means* konvensional yang mampu memiliki hasil lebih baik tapi bisa saja hasilnya lebih buruk.

Kata kunci *: Data mining,* Klasterisasi, Algoritma *K-Means, Improved K-Means, Weighted Average*.

ABSTRACT

Herlambang, 2015. ***Implementation Of Improvement K-Means With Centroid Initialitation Using Weighted Average*** **on** ***Iris Dataset with OWA Weighting***
Advisor **: Dian Eka Ratnawati, S.Si., M.Kom. and Budi Darma Setiawan, S.Kom., M.Cs.**

K-Means is popular method on clustering but has the disadvantage, the initials centroid is determined randomly, it makes the results of the cluster inconsistent. K-Means with the weighted average is a clustering algorithm which avoids the random initial centroid determination, which produce a consistent cluster. However, the value of the attribute weights determined by user, so if the attribute weights is not correct the cluster result become worst. So, required weights value which corresponding to the data distribution. This research uses iris dataset which weight value obtained from the calculation of OWA. From weigth test, different weights values produce different clusters result. Iris dataset is the data that already has a class label, so it can be tested using silhouette coefficient and accuracy testing using different calculating distance methods, euclidean and manhattan. In coefficient silhouette testing, improve K-Means has better results, where Euclidean has better silhouette coefficient than manhattan. On the accuracy testing K-Means conventional has better accuracy, where the accuracy using manhattan is better than euclidean. Different from improve K-Means which has lower accuracy, but accuracy testing using euclidean better than manhattan. Based on the test found, improve K-Means have consistent results, but can not achieve optimum results. Different from K-Means conventional which able to have a better or bad result.

Keyword : Data mining, Clustering, K-Means, Improved K-Means, Weighted Average.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE IMPLEMENTASI

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Data mining merupakan serangkaian proses dalam menganalisa data dan kemudian mengolah data tersebut menjadi informasi dan pengetahuan yang berguna. Informasi tersebut dapat dimanfaatkan oleh pihak yang berkepentingan seperti sebuah perusahaan untuk melakukan analisa pasar, deteksi penipuan dan pelayanan pelanggan, sehingga dapat meningkatkan pendapatan atau menghemat pengeluaran pada perusahaan tersebut. Salah satu tugas dalam data mining adalah *clustering* yaitu pengelompokan data ke dalam kelompok yang memiliki karakteristik serupa [HAN-06]. Salah satu metode dalam *clustering* yang umum digunakan adalah metode *K-Means*.

Algoritma *K-Means* merupakan algoritma *clustering* yang cukup populer dalam mengolah dataset yang cukup besar karena memiliki algoritma yang sederhana. Algoritma K-Means adalah metode *clustering* non hirarki yang digunakan untuk mengelompokan data berdasarkan karakteristik *attribute* yang dimilikinya sehingga data yang memiliki karakteristik sama akan berada dalam *cluster* yang sama. Cara kerja algoritma ini adalah mengelompokkan data dengan proses iteratif dan kemudian melakukan penghitungan ulang parameter sebuah *cluster*. Proses K-Means ini akan berhenti pada saat tiap anggota di tiap *cluster* tidak mengalami perubahan saat penghitungan ulang parameter [YUA-04].

Di balik penggunaan algoritma *K-Means* yang cukup sederhana, terdapat beberapa kelemahan yang harus diperhatikan [SIN-11]. Pertama jumlah *cluster* harus diketahui dan ditentukan terlebih dahulu oleh pengguna. Kedua inisialisasi *centroid* atau titik tengah suatu *cluster* berpengaruh secara langsung terhadap hasil *clustering*. *Centroid* memiliki peran dalam merepresentasikan bagaimana penyebaran data dari setiap *cluster*. Seperti yang sudah diketahui algoritma *K-Means* menentukan inisial *centroid* dengan cara *random* atau acak. Berdasarkan

nilai *centroid* yang ditentukan secara acak, tentunya akan memberikan hasil yang berbeda. Hasil *clustering* bisa saja memberikan hasil yang baik atau bisa juga memberikan hasil tidak diinginkan [SIN-11].

Apabila inisialisasi *centroid* dihasilkan dengan cara random, maka *K-Means* tidak dapat menjamin hasil akhir *clustering* adalah hasil yang baik [KHA-04]. Untuk mengatasi masalah inisialisasi *centroid* secara random Md. Sohrab Mahmud, Md. Mostafizer Rahman, dan Md.Nasim Akhtar melakukan penelitian untuk menentukan nilai inisialisasi *centroid* berdasarkan nilai *weighted average* dari *attribute* yang dimiliki oleh masing-masing *data-point*. Dengan demikian akan menghasilkan nilai *centroid* yang sesuai dengan distribusi data dari dataset yang diproses [MAH-12].

Penelitian sebelumnya yang dilakuan oleh Md. Sohrab Mahmud, Md. Mostafizer Rahman, dan Md. Nasim Akhtar menggunakan *K-Means clustering* dengan inisialisasi *centroid* berdasar *weighted average* dengan data uji berupa dataset *diabetes, thyroid*, dan *blood pressure*. Hasil *clustering* menggunakan *K-Means* dengan *weighted average* pada *dataset diabetes* rata-rata waktu *clustering* 0,0781 mili detik, pada *dataset thyroid* rata-rata waktu yang diperoleh sebesar 0,1074 mili detik, dan pada *dataset blood pressure* diperoleh rata-rata waktu sebesar 0,0924 mili detik. Sedangkan *clustering* pada algoritma *K-Means* konvensional menghasilkan rata-rata waktu 0,0938 pada *dataset diabetes*, pada *dataset thyroid* rata-rata waktu *clusteringnya* sebesar 0,1293 mili detik, dan pada *dataset blood pressure* didapati rata-rata waktu *clustering* sebesar 0,1055. Dari perbandingan rata-rata waktu *clustering* yang didapat, *Improvement K-Means* dengan inisialisasi *centroid* berdasar *weighted average* memiliki rata-rata waktu *clustering* yang lebih cepat dibandingkan dengan algoritma *K-Means* konvensional [MAH-12].

Namun, dalam penelitian sebelumnya yang dilakuan oleh Md. Sohrab Mahmud, Md. Mostafizer Rahman, dan Md. Nasim Akhtar menggunakan *K-Means clustering* dengan inisialisasi *centroid* berdasar *weighted average* nilai bobot *attribute* dari suatu *dataset* ditentukan secara manual oleh pengguna sistem sehingga pengguna bebas memberikan nilai inputan bobot suatu *attribute* dari

dataset. Bobot sendiri merupakan variabel yang juga berpengaruh terhadap hasil inisial *centroid*, sehingga nilai inputan bobot yang salah juga akan memberikan hasil *clustering* yang kurang. Maka dari itu, pada penelitin ini bobot dari suatu *attribute* diambil dari penelitian Ching-Huse Cheng, Jing-Wei Liu, dan Ming-Chang Wu mengenai *Ordered Weighted Averaging* untuk menyelesaikan permasalah klasifikasi yang mana salah satu objek yang digunakan di dalam penelitian tersebut adalah *dataset* iris [CHE-07].

Dataset iris adalah dataset yang terdiri dari 150 data, yang mana memiliki empat *attribute sepal-length*, *sepal_width*, *petal_length*, dan *petal_width. Dataset* iris sendiri merupakan *well-known dataset* yaitu dataset yang sudah memiliki informasi label kelompok pada setiap datanya. *Dengan* penggunaan dataset *iris*, pengujian hasil *clustering* dapat dilakukan dengan menggunakan pengujian *external quality measure* dan *internal quality measure*. *external quality measure* merupakan pengujian hasil *clustering* dengan adanya referensi informasi label yang dimiliki oleh suatu *dataset*, dan pengujian *internal quality measure* yaitu pengujian dengan membandingkan *cluster-cluster* yang dihasilkan tanpa menggunakan referensi informasi dari suatu dataset [STE-00].

Pengujian yang dilakkan pada penelitian ini berbeda dengan penelitian yang dilakukan oleh Md. Sohrab Mahmud, Md. Mostafizer Rahman, dan Md. Nasim Akhtar, yaitu pengujian perbandingan estimasi waktu yang dibutuhkan dalam *clustering*. Pada penelitian ini nantinya akan dilakukan tiga macam pengujian yang pertama pengujian untuk mengetahui bagaimana pengaruh bobot yang diperoleh dari penelitian Ching-Huse Cheng, Jing-Wei Liu, dan Ming-Chang Wu terhadap hasil kualitas *clustering*, kedua pengujian kualitas *clustering* dengan *silhouette coefficient*, dan ketiga pengujian akurasi *clustering* data berdasarkan referensi informasi *dataset*. Selain itu, pada tahap proses *K-Means clustering* akan dilakukan dengan menggunakan dua metode perhitungan jarak yang berbeda, yaitu dengan *euclidean* dan *manhattan*, sehingga dapat diketahui perhitungan jarak yang menghasilkan hasil *clustering* yang lebih baik.

Berdasarkan latar belakang yang telah diuraikan maka skripsi ini diberi judul **"Implementasi Metode *Improvement K-Means* Dengan Inisialisasi**

***Centroid* Menggunakan *Weighted Average* Pada *Dataset* Iris Dengan Pembobotan OWA".**

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah yang diambil adalah :

1. Bagaimana menentukan inisial *centroid* algoritma *improvement K-Means* pada *clustering* data dengan menggunakan *weighted average*?
2. Bagaimana perbandingan kualitas *clustering improvement K-Means* dengan dengan inisialisasi *centroid* menggunakan *weighted average* saat menggunakan bobot yang dihasilkan oleh *Ordered Weighted Averaging*?
3. Bagaimana perbandingan kualitas *clustering* data antara algoritma *improvement K-Means* yang inisial *centroid*-nya telah ditentukan dengan metode *wighted average* dan algoritma *K-Means* konvensional?
4. Bagaimana perbandingan kualitas *clustering* antara *euclidean* dan *manhattan* saat diterapkan pada *improve K-Means* dan *K-Means* konvensional?
5. Bagaimana perbandingan akurasi *clustering* data antara algoritma *improvement K-Means* yang inisial *centroid*-nya telah ditentukan dengan metode *wighted average* dan algoritma *K-Means* konvensional?
6. Bagaimana perbandingan akurasi *clustering* antara *euclidean* dan *manhattan* saat diterapkan pada *improve K-Means* dan *K-Means* konvensional?

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah :

1. Membangun aplikasi *clustering* data dengan mengimplementasikan algoritma *K-Means* dengan inisial *centroid*-nya yang telah ditentukan dengan metode *wighted average*.
2. Mengetahui perbandingan kualitas *clustering* dari penggunaan bobot *ordered weighted averaging* terhadap *K-Means* dengan inisialisasi centroid menggunakan *weighted average*.

3. Mengetahui perbandingan kualitas *clustering* yang dihasilkan dengan menggunakan algoritma *improvement K-Means* dengan inisialisasi *centroid* metode *weighted average* dan algoritma *K-Means* konvensional.
4. Mengetahui metode perhitungan jarak antara *euclidean* dan *manhattan* yang dapat menghasilkan kualitas *clustering* yang lebih baik saat diterapkan pada *improvement K-Means* yang inisial *centroid*-nya telah ditentukan dengan metode *wighted average* dan algoritma *K-Means* konvensional.
5. Mengetahui perbandingan akurasi *clustering* data antara algoritma *improvement K-Means* yang inisial *centroid*-nya telah ditentukan dengan metode *wighted average* dan algoritma *K-Means* konvensional.
6. Mengetahui metode perhitungan jarak antara *euclidean* dan manhattan yang dapat menghasilkan kualitas *clustering* yang lebih baik saat diterapkan pada *improvement K-Means* yang inisial *centroid*-nya telah ditentukan dengan metode *wighted average* dan algoritma *K-Means* konvensional.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. *Dataset* yang dipergunakan adalah *iris* yang di ambil dari UCI *Machine Learning Repository.*
2. *Dataset* yang dipergunakan tidak memiliki data dengan nilai attribute yang tidak ada atau *missing value*.

1.5 Manfaat Penelitian

Manfaat dari hasil skripsi ini diharapkan akan menghasilkan perangkat lunak yang dapat menerapkan algoritma *improvement K-Means* yang inisial *centroid*-nya ditentukan oleh metode *weighted average*. Dari aplikasi tersebut dapat diketahui bagaimana hasil perbandingan yang terbaik antara algoritma *improvement K-Means* yang inisial *centroid*nya telah ditentukan dengan *weighted average* dan algoritma K-Means konvensional dalam *clustering* data. Serta dapat memberikan pengetahuan dan pemahaman dalam *clustering* data dengan menggunakan algoritma *K-Means*.

1.6 Sistematika Penulisan

Penulisan Skripsi ini disusun secara sistematis agar pembaca lebih mudah dalam mempelajari dan diambil manfaatnya. Sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II : LANDASAN TEORI

Berisi dasar teori dan referensi yang menunjang penelitian dalam pembuatan skripsi. Teori yang dibahas meliputi penertian dan sefinisi tentang *data minin*, *clusterin*, *algoritma K-Means*, dan algoritma *improvement K-Means* dengan perbaikan inisial *centroid* menggunakan *weighted average*.

BAB III : METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

Bab metodologi menjelaskan serangkaian langkah atau tahapan yang dilakukan di dalam penelitian ini. Setiap langkah akan diberi penjelasan yang bertujuan untuk membantu peneliti dalam menyelesaikan penelitian ini. Pada bagian perancangan sistem berisi penjelasan tentang rancangan sistem menggunakan algoritma *improvement K-Means* yang inisial *centroid* ditentukan menggunakan *weighted average*.

BAB IV : IMPLEMENTASI

Memberikan pembahasan mengenai implementasi perangkat lunak atau sistem yang dibuat berdasarkan pada tahap perancangan sistem.

BAB V : PENGUJIAN DAN ANALISIS

Dari perangkat lunak hasil implementasi yang telah dibuat kemudian dilakukan pengujian.

BAB VI : PENUTUP

Menjelaskan mengenai kesimpulan yang diperoleh dari penelitian dan kemudian saran untuk proses pengembangan lebih kanjut mengenai topik yang dibahas pada skripsi ini.

BAB II
LANDASAN TEORI

2.1 Kajian Pustaka

Penelitian sebelumnya mengenai *K-Means clustering* dengan inisialisasi *centroid* berdasar *weighted average* oleh Md. Sohrab Mahmud, Md. Mostafizer Rahman, dan Md. Nasim Akhtar. Nilai *weighted average* didapat dari *attribute* yang dimiliki oleh masing-masing *data-point*. Dengan demikian akan menghasilkan nilai *centroid* yang sesuai dengan distribusi data dari *dataset* yang diproses. Hasil nilai *wighted average* dari masing-masing *data-point* kemudian akan diurutkan dan membaginya menjadi sejumlah beberapa *subset* sesuai *cluster* yang dikehendaki. Selanjutnya nilai terdekat dari titk tengah dari tiap *subset* akan dijadikan sebagai nilai *centroid* awal. Dalam penggunaan *weighted average* sebagai penentu *centroid* awal terdapat kelebihan yakni mampu mengurangi jumlah iterasi yang diperlukan dalam *clustering* data dibandingkan dengan algoritma *K-Means* yang melakukan pemilihan nilai *centroid* secara *random* [MAH-12].

2.2 Data Mining

2.2.1 Pengertian Data Mining

Data mining merupakan disiplin ilmu yang dibangun dalam cakupan kecerdasan buatan dan rekayasa pengetahuan. Pada beberapa tahun terakhir teknologi pengumpulan dan penyimpan data berjalan cukup cepat yang menghasilkan tumpukan data yang besar. Pada kondisi tersebut, proses ekstraksi data menjadihal yang sangat dibutuhkan. Berdasarkan kondisi tersebut diperlukanlah sebuah teknologi yang disebut data mining [PUR-12].

Data mining merupakan salah satu tahapan yang ada pada *knowledge discovery process*. *Data mining* sendiri dapat dikatakan sebagai kegiatan mengekstraksi suatu informasi dari data yang berjumlah besar. Data berjumlah besar tersebut meliputi data yang tersimpan pada *databases*, *data warehouses*, atau

berbagai *repository* informasi. Informasi yang ditemukan dapat diolah dan digunakan oleh pihak yang berkepentingan [HAN-06].

Data mining adalah bidang gabungan dari beberapa bidang keilmuwan yang mana menyatukan teknik pembelajaran pada mesin, pengenalan pola, statistik, database, dan visualisasi, yang di dalamnya terdapat proses analisa data untuk menemukan pola dan hubungan, kemudian meringkasnya untuk dijadikan penanganan masalah pemilik data [LAR-05].

Sedangkan menurut Paul Beynon-Davies (2007) *data mining* digunakan untuk melakukan ekstraksi data *warehaouse* dan data pasar yang besar dan terus berkembang seiring berjalannya waktu. Berdasarkan hasil ekstraksi diperoleh informasi yang berguna untuk memperoleh keuntungan perusahaan.

Menurut Paul Beynon-Davies (2007) data mining memeiliki karakteristik antara lain :

1. *Data mining* fokus pada pola data yang tersembunyi dan bisa tidak dapat diprediksi.
2. *Data mining* digunakan untuk menggali informasi dari data yang berukuran besar, di mana semakin besar data akan semakin dapat dipercaya hasil akhir hasil ekstraksi datanya.
3. *Data mining* berguna bagi sebuah organisasi dalam menentukan keputusan di dalam kondisi yang kritis.

2.2.2 Proses Data Mining

Data mining adalah proses pencarian dari berbagai model, ringkasan, dan dapat diperoleh nilainya dari sekumpulan data. Di dalam proses *data mining* terdapat serangkaian frase atau langkah yang harus dilakukan [KAN-11] :

1. Menentukan masalah dan merumuskan hipotesis.

   Untuk menentukan sebuah masalah dan merumuskan hipotesis diperlukan sebuah pakar keilmuwan khusus yang dapat menunjang aktivitas *data mining*. Pada tahap ini diperlukan adanya kombinasi keahlian antara pakar aplikasi dan pakar dalam *data mining*.

2. Mengumpulkan data.

   Pada tahap ini akan memahami tentang bagaimana dihasilkan dan bagaimana mengumpulkannya untuk proses data mining. Perlu digaris bawahi, terdapat dua kemungkinan pertama data yang dihasilkan masih berada dalam kendali ahli, ke dua saat data yang dihasilkan di luar kendali ahli.

3. *Preprocessing* data.

   Saat data telah diperoleh, dilakukan *preprocessing* data yaitu untuk mengolah data agar menjadi format yang sesuai untuk tahapan atau proses selanjutnya. Pada tahap ini terbagi menjadi dua sub fase

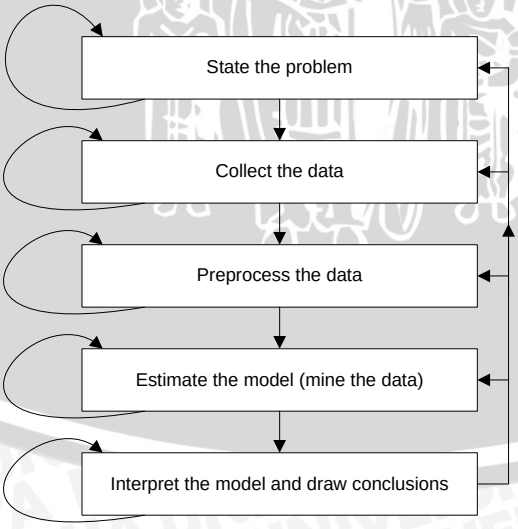
   a. Deteksi dan penghapusan outlier atau data yang tidak konsisten atau menyimpangan dengan data lainnya.
   b. Menentukan skala, *encoding*, dan pemilihan fitur yang mana akan membuat data menjadi lebih sederhana atau ringkas.

4. Estimasi model.

   Pada tahap ini akan dilakukan analisa untuk memilih dan implementasi teknik data mining yang sesuai dengan masalah yag dihadapi.

5. Menafsirkan model dan kemudian menarik kesimpulan.

Secara sederhana, proses-proses dalam *data mining* digambarkan pada Gambar 2.1



Gambar 2. 1 Proses-Proses dalam Data Mining

2.2.3 Metode Data Mining

Data mining memiliki fungsi untuk mengetahui dan menetapkan bentuk dari pola yang ditemukan pada kegiatan penggalian data. Pada umumnya terdapat dua pendekatan di dalam *data mining*, yaitu *descriptive mining* dan *predictive mining* [HAN-06].

Menurut Pang-ning Tan (2005) dibagi menjadi dua *tasks*, yaitu *Descriptive* dan *Predictive*, berikut penjelasannya :

1. *Descriptive mining* bertujuan untuk menemukan korelasi, kecenderungan, dan alur yang meringkas keterkaitan antar data. Metode *data mining* yang termasuk dalam *descriptive mining* adalah *clustering*, *association rule discovery*, dan *sequential patern discovery*.
2. *Predictive mining* bertujuan untuk memprediksi nilai dari atribut yang spesifik, berdasarkan atribut yang lain. Metode *data mining* yang termasuk dalam metode *predictive mining* adalah klasifikasi, *regression*, dan *deviation detection*.

2.3 *Clustering* Data

Clustering data merupakan pegelompokkan data yang belum diketahui labelnya atau kelasnya atau yang disebut *unsupervised-learning*. Pada metode *clustering* akan mengelompokkan data berdasarkan atribut dan nilai kesamaan antar data dalam sebuah *dataset*, yang kemudian dikelompokkan menjadi kelas atau *cluster* [HAN-06].

Pada metode *clustering* akan dilakukan sebuah observasi berupa pada sebuah dataset. Observasi bertujuan untuk mencari infomasi mengenai kelas dari sebuah data.*Clustering* atau pengelompokkan data berdasarkan pada prinsip "*maximizing intraclass similarity and minimizing interclass similarity"*. Sehingga, akan didapati objek yang memiliki tingkat kesamaan yang tinggi dengan antar objek dalam satu *cluster*, dan akan akan memiliki tingkat ketidaksamaan yang tinggi pula jika dibandingkan dengan objek yang berada di luar *cluster*nya [HAN-06]. Pada dasarnya *clustering* data dibagi menjadi dua metode, antara lain [TAN-11] :

1. *Hierarchical method* melakukan pengelompokannya dimulai dari mengelompokkan data dari dua objek atau lebih berdasarkannya kesamaanya. Kemudian dilanjutkan pada objek lain yang mempunyai kedekatan kedua. Sehingga akan menghasilkan tingkatan hirarki yang berbentuk seperti pohon.
   Contoh algoritma *Hierarchical method Tree*.
2. *Non-Hierarchical method* dimulai dengan menentukan jumlah *cluster* yang diinginkan terlebih dahulu, kemudian dilakukan proses *clustering* data tanpa memerlukan proses hirarki.
   Contoh algoritma *Non-Hierarchical method* adalah *K-Modes*, *K-Means*.

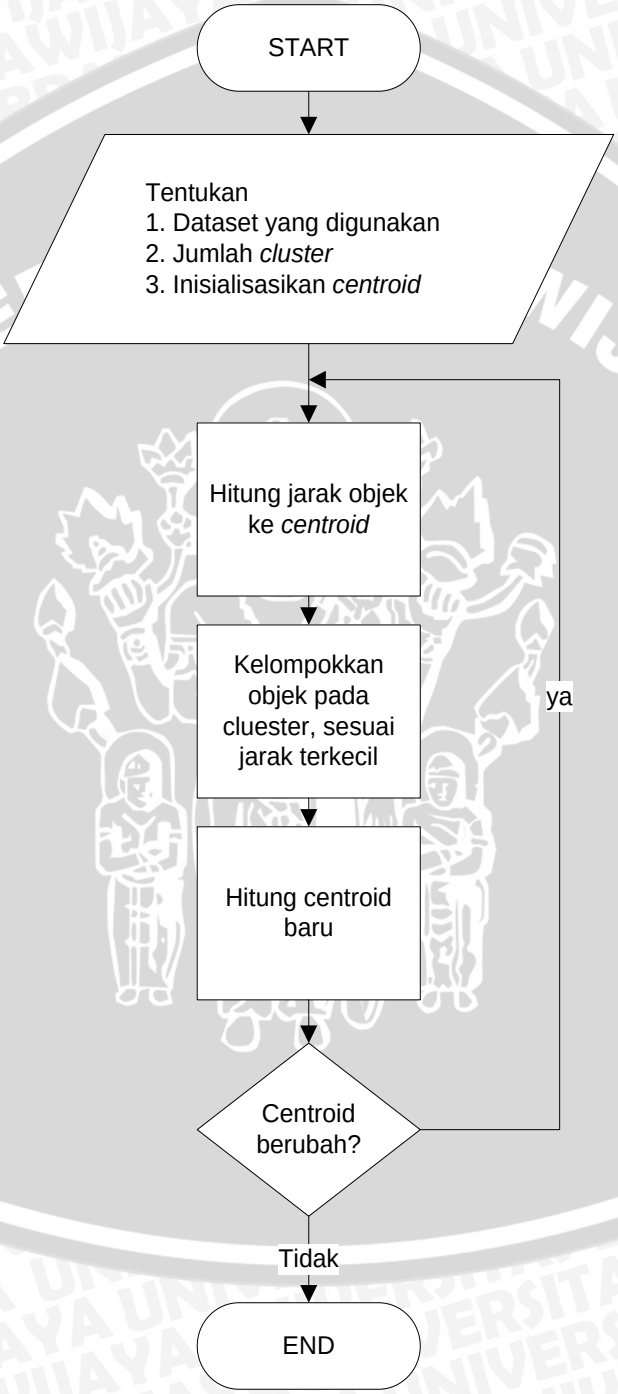
2.4 Algoritma *K-Means*

Algoritma *K-Means* adalah metode unsupervised hierarchical method di dalam *data mining* yang berguna unruk melakkukan *clustering* data. Pada algortima K-Means dapat dikatakan sebagai model *centroid*. *Centroid* atau yang disebut titik tengah untuk membuat *cluster*, yang pada implementasinya digunakan untuk menghitung jarak suatu objek data terhadap *centroid* [BEN-14].

Algoritma *K*-Means merupakan salah ssatu algoritma *clustering* yang popular. Alasan kenapa algoritma *K-Means* banyak digunakan karena mudah dan sederhana dalam mengimplementasikannya. Selain itu, algoritma K-Means juga memiliki skalabilitas, cepat, dan dapat diadaptasikan pada berbagai macam data. Dari berbagai aktifitas *clustering* yang telah dilakukan selama ini, algoritma *K-Mean juga* cukup efektif dalam melakukan *clustering* untuk berbagai bidang. [FAH-06].

Pada implemantasi algoritma *K-Means*, terdiri dari beberapa langkah. Pertama menentukan Dataset yang digunakan, lalu tentukan banyaknya *cluster* yang ingin digunakan beserta masing-masing inisial *centroid*nya. Kedua hitung jarak antara tiap objek dan dan *centroid*. Ketiga kelompokkan data objek, pada *cluster* yang memiliki jarak *centroid* terdekat. Keempat hitung nilai *centroid* baru berdasarkan hasil pengelompokkan pada tahap ketiga. Kelima apabila nilai *centroid* berubah kembali ke tahap kedua, apabila nilai *centroid* sama *clustering* selesai [YED-10].

Seperti yang sudah dijelaskan di atas, algortima K-Means implementasinya cukup sederhana dalam *clustering*, berikut ini adalah langkah-langkah *clustering* dengan metode K-Means dijelaskan pada flowchart Gambar 2.2 :



Gambar 2. 2 Flowchart Algoritma *K-Means*

2.5 Weighted Average

Seperti yang sudah diketahui metode *K-Means* menentukan inisial *centroid* dengan random, sedangkan hasil *clustering* menggunakan K-Means sangat sensitif dengan inisialisasi *centroid*. Berdasarkan inisialisasi *centroid* yang random, bisa saja memberikan hasil *clustering* yang baik, atau bisa saja memberikan hasil yang buruk. Untuk mengatasi hal tersebut, Md. Sohrab Mahmud, Md. Mostafizer Rahman, dan Md. Nasim Akhtar mengusulkan penggunaan *weighted average* dalam menentukan inisial *centroid* [MAH-12].

Metode *weighted average* adalah metode menentukan inisialisasi *centroid* berdasarkan nilai *weighted average* dari *attribute* yang dimiliki oleh masing-masing baris data pada *dataset*. Hasil nilai weighted average dari masing-masing *data-point* kemudian akan dilakukan proses sorting dan membaginya menjadi sejumlah *subset* sesuai *cluster* yang dikehendaki. Selanjutnya nilai terdekat dari titk tengah dari tiap *subset* akan dijadikan sebagai nilai *centroid* awal [MAH-12].

Dalam penggunaan *weighted average* sebagai penentu *centroid* awal, terdapat kelebihan yakni apabila ingin meningkatkan prioritas dari sebuah *dataset*, dapat dilakukan dengan meningkatkan nilai bobot dari *attribute* dari objek data yang dikehendaki, sehingga diharapkan akan menghasilkan nilai *centroid* yang sesuai dengan distribusi data dari *dataset* yang diproses [MAH-12].

Berikut ini adalah penjelasan mengenai langkah-langkah metode *weighted average* dalam menentukan inisial *centroid* [MAH-12] :

1. Tentukan dataset yang akan digunakan beserta jumlah *cluster* yang diiginkan.
2. Hitung nilai di(avg) atau *weighted average* setiap *data point* dengan menggunakan persamaan 2.1 :

   $d_i(avg) = (w1*x1 + w2*x2 + w3*x3 + \ldots. + wm*xn);\quad i= 1,2,3,\ldots n \qquad (2.1)$

   dimana; d_i : adalah baris pada baris data atau *data point*,
   w : merupakan bobot yang dimiliki suatu *attribute*, dan
   x : merupakan nilai dari *attribute* yang dimiliki suatu *data point*.
3. Lakukan *sorting dataset* secara *ascending* berdasarkan nilai di(avg) yang diperoleh pada langkah sebelumnya.

4. Bagi dataset yang telah diurutkan menjadi sejumlah *subset* sesuai dengan jumlah *cluster* yang ditentukan sebelumya.
5. Hitung nilai rata-rata di(avg) setiap subset pada masing-masing subset.
6. Gunakan *datapoint* yang memiliki di(avg) terdekat dengan rata-rata di(avg) sebagai inisial centroid.

2.6 Perhitungan Jarak

Pada penelitian ini akan digunakan dua metode perhitungan jarak, yaitu *euclidean distance* dan *manhattan distance*. Penggunaan perhitungan jarak yang berbeda diharapkan dapat membantu menentukan penggunaan metode perhitungan jarak yang tepat untuk diimplementasikan pada sistem [HER-05].

2.6.1 *Euclidean Distance*

Euclidean distance meghitung akar pangkat dua dari perbedaan koordinat dari sepasang objek. Formula perhitungan *euclidean* sebagai berikut.

$$\|X_i - C_j\| = \sqrt{\sum_{j=1}^{c}(X_i - C_j)^2} \quad (2.2)$$

Dimana:

D(i,j) = Jarak data *i* ke pusat *cluster j*

X_{ki} = Data ke *i* pada atribut data ke *k*

X_{kj} = Data titik pusat ke *j* pada atribut ke *k*

2.6.2 *Manhattan Distance*

Manhattan distance merepresentasikan jarak antara dua objek secara absolut. Formula perhitungan *manhattan* sebagai berikut.

$$d = \sum_{i=1}^{m}\sum_{j=1}^{n}\left|X_{i,j} - Y_{i,j}\right| \quad (2.3)$$

Dimana:

D(i,j) = d merupakan jarak antara dua titik.

Xi dan Yi = nilai-nilai dari variabel i, pada titik titik X dan Y masing-masing.

2.7 *Silhouette Coefficient*

Dalam pengujian hasil clustering terdapat metode dalam menguji kualitas hasil clustering, salah satu metode tersebut adalah silhouette coefficient. Silhouette coefficient bekerja dengan melakukan evaluasi clustering dengan memeriksa bagaimana cluster dipisahkan dan bagaimana kekompakan suatu cluster [HAN-06].

Langkah-langkah yang dilakukan dalam perhitungan *silhouette coefficient* adalah sebagai berikut :

1. Untuk data ke-i, hitung hitung nilai ai yaitu average *dissimilarity* (ketidaksamaan rata-rata). Semakin kecil nilai ai maka semakin baik i dikelompokkan dalam suatu klaster.

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

   Dimana :

   a(i) = rata-rata jarak antara data i dengan semua data dalam klaster yang sama

   j = data lain dalam satu klaster A

   |A| = jumlah data dalam klaster A

2. Kemudian hitung nilai bi yang merupakan rata-rata *dissimilarity* terendah dari I terhadap semua klaster dimana i bukan anggota *cluster*. *Cluster* dengan nilai rata-rata *dissimilarity* paling rendah disebut dengan *neighbouring cluster* (kelompok tetangga) dari i karena menjadi klaster tetangga terbaik untuk i.

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

   d(i,C) = rata-rata jarak data i dengan semua data dalam klaster C dimana C ≠ A. Sehingga didapatkan nilai b(i) sebagai berikut :

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

3. Untuk data ke-i, nilai *silhouette coefficient* didapatkan dengan persamaan :

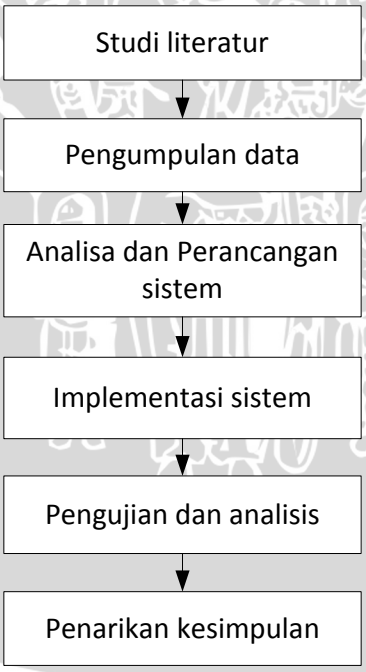
   $$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

   Rata-rata s_i dari seluruh data dalam suatu *klaster* menunjukkan kedekatan kemiripan data suatu *klaster*. Sehingga, rata-rata s_i seluruh

data menunjukkan ketepatan data yang telah terkelompokkan. Nilai s_i yang mendekati nilai 1 dikatakan semakin baik, namun apabila jika nilai s_i mendekati -1 dikatakan semakin buruk [RPJ-87].

BAB III
METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

Pada bab metodologi penelitian berisi penjelasan tentang metode yang digunakan dalam menentukan inisial *centroid* pada algoritma *K-Means* menggunakan metode *weighted average*. Metodologi penelitian berfungsi sebagai acuan penelitian, agar penelitian dapat berjalan sistematis dan terarah. Di dalam metode penelitian terdapat penjelasan umum mengenai penelitian yang dilakukan beserta tahapan yang akan dilakukan pada penelitian ini. Tahapan di dalam metode penelitian ini antara lain, studi literatur, pengumpulan data, analisa kebutuhan, perancangan sistem, analisa sistem, pengujian beserta analisa sistem, dan diikuti dengan penerikan kesimpulan. Tahapan dalam metode penelitian akan dijelaskan pada Gambar 3.1.



Gambar 3. 1 Tahapan Metode Penelitian

3.1 Studi Literatur

Pada tahap studi literatur, akan dilakukan proses mempelajari dasar teori yang digunakan untuk menunjang penelitian tentang penentuan inisial *centroid* pada

metode algoritma *K-Means* menggunakan metode *weighted average*. Di dalam penelitian ini studi literatur diperoleh dari jurnal, buku, internet, serta bimbingan dari dosen pembimbing. Pada tahap studi literatur terdapat beberapa teori yang dipelajari, antara lain :

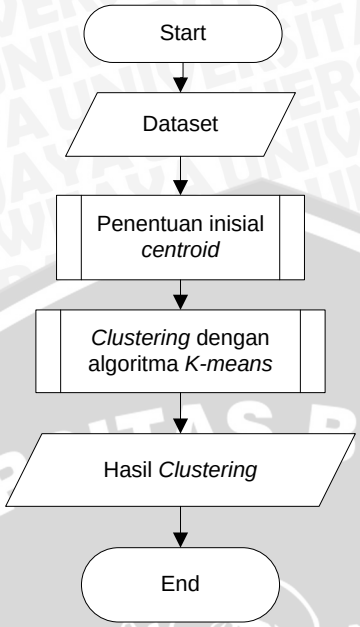
1 Pengertian tentang *data mining* meliputi proses dan metode di dalam *data mining*.
2 Pengertian mengenai *clustering* data beserta contohnya.
3 Pengertian tentang algoritma *K-Means* untuk *clustering* data.
4 Pengertian mengenai metode *weighted average* dalam menentukan inisial *centroid* algoritma *K-Means*.

3.2 Data Penelitian

Pada penelitian ini, data yang digunakan adalah *dataset iris* yang didapat dari *website* UCI *Machine Learning Repository* (https://archive.ics.uci.edu). *Dataset* iris merupakan *dataset* yang sudah memiliki informasi label atau disebut dengan *well-known dataset*, yang mana dengan menggunakan data tersebut dapat dilakukan dua macam pengujian *clustering*, yakni pengujian kualitas *cluster* yang dihasilkan dengan *silhouette coefficient* dan pengujian akurasi untuk mengukur tingkat kemiripan data pada suatu cluster. Total data yang terdapat pada dataset iris adalah sebanyak 150 data yang terbagi menjadi tiga kelas meliputi *iris-setosa, iris-versicolor*, dan *iris-virginica*. *Dataset* iris memiliki empat *attribute* yaitu, *sepal length, sepal width,, petal length*, dan *petal width*.

3.3 Perancangan Sistem

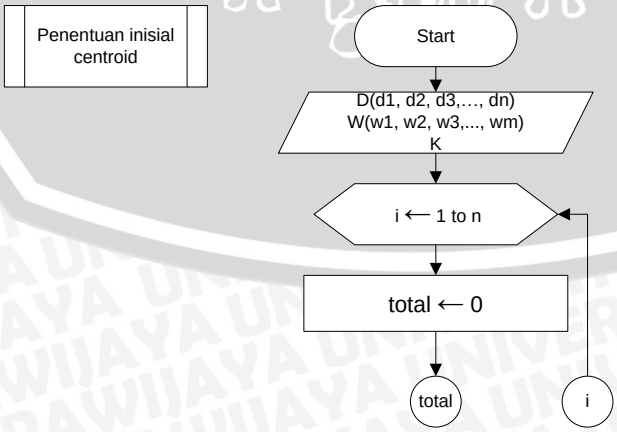
Pada tahap perancangan akan dilakukan proses analisa sistem yang kemudian dilanjutkan dengan merancang arsitektur sistem berdasarkan hasil analisa. Arsitektur yang diperoleh akan menggambarkan kerangka sistem secara umum. Secara umum, sistem terbagi menjadi dua bagian *utama* yaitu penentuan inisial *centroid* dengan menggunakan *weighted average* dan kemudian dilanjutkan dengan proses *clustering* dengan algoritma *K-means*. Gambaran mengenai gambaran sistem digambarkan oleh Gambar 3.2.



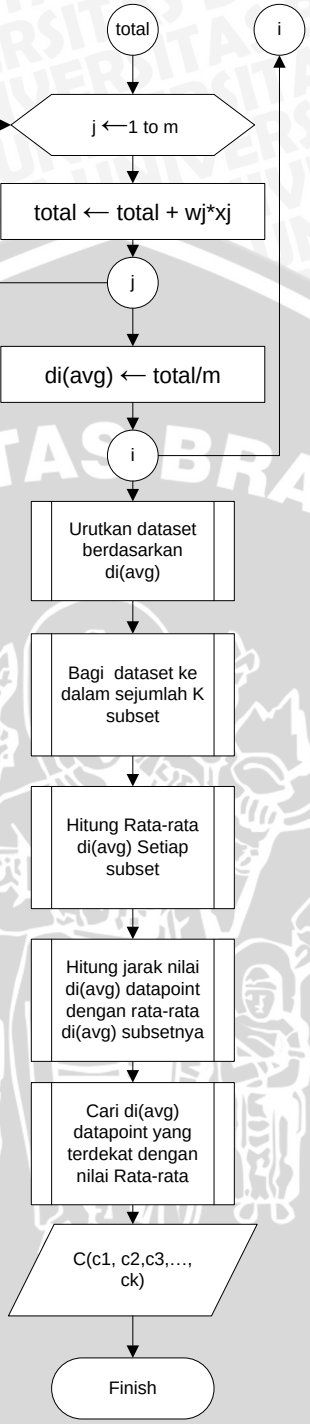
Gambar 3. 2 Gambaran Umum Sistem

3.3.1 Penentuan Inisial Centroid

Penentuan inisial *centroid* diawali dengan menginputkan dataset yang akan digunakan dan kemudian menentukan jumlah *cluster* yang diinginkan. *Dataset* D yang diinputkan berupa data sebanyak n-data, *cluster* yang ingin dibentuk diinputkan senjumlah k-*cluster*. Hasil akhir penentuan insial *centroid* berupa nilai *centroid* sejumlah nilai k. Berikut ini adalah flowchart dalam menentukan inisial *centroid* metode weighted average ditampilkan pada Gambar 3.3.



Gambar 3. 3 Flowchart Penentuan Inisial *Centroid*



Gambar 3. 3 Flowchart Penentuan Inisial *Centroid* (lanjutan)

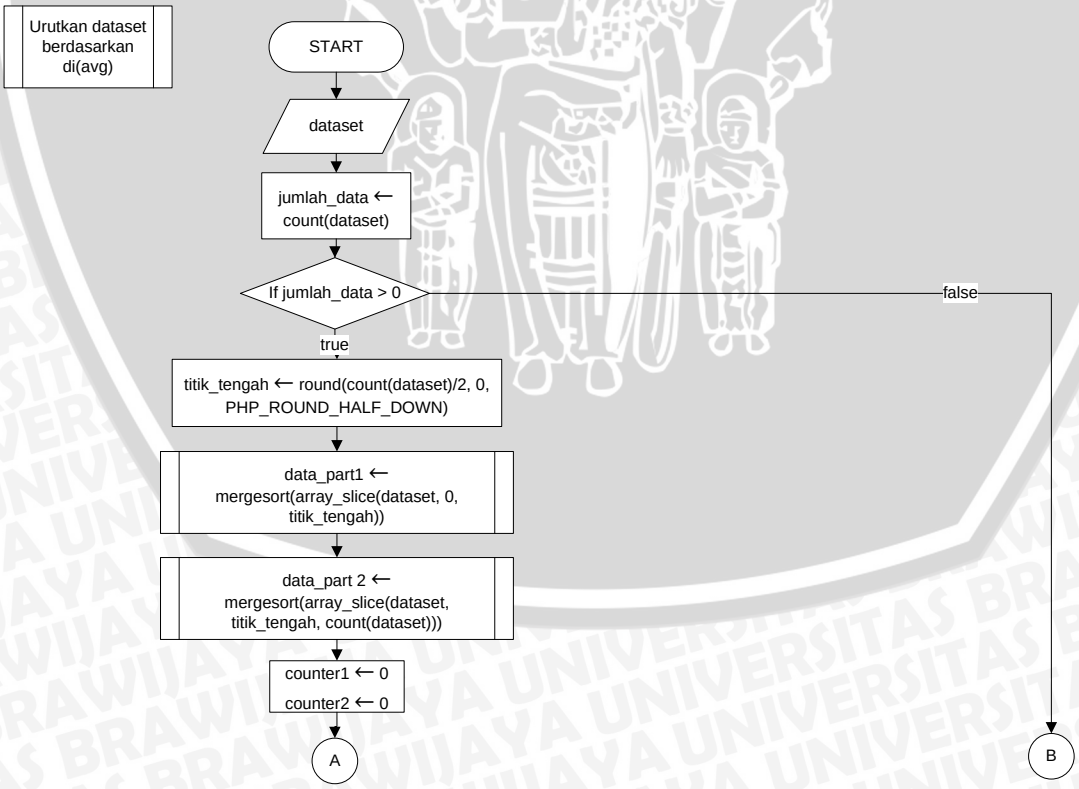
Penjelasan flowchart penentuan inisial *centroid* :

1. Penentuan inisial centroid dimulai dari menginputkan dataset yang akan digunakan dan juga menginputkan jumlah *cluster* sejumlah K.
2. Kemudian dilakukan perhitungan di(avg) pada setiap *datapoint*.

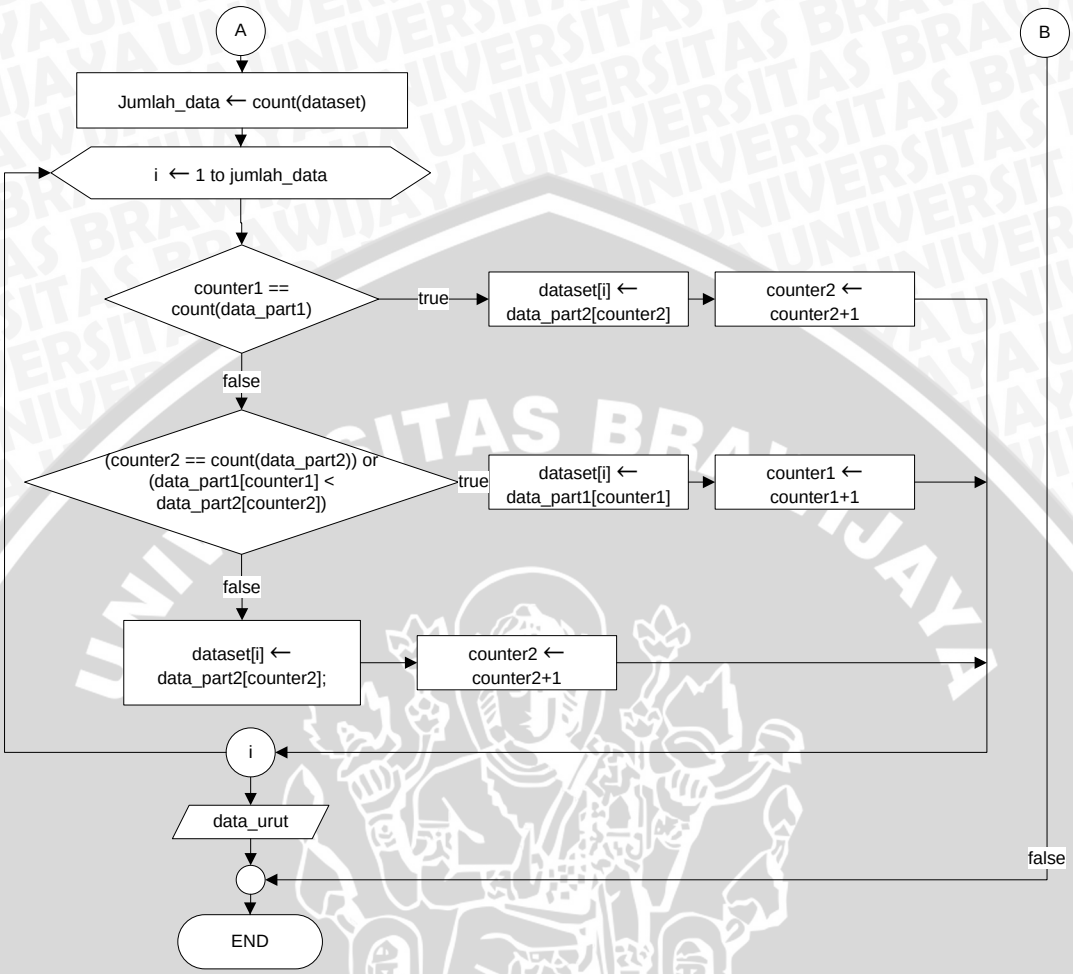
3. Urutkan dataset secara *ascending* berdasarkan nilai di(avg).
4. Bagi dataset menjadi sejumlah K subset.
5. Hitung rata-rata di(avg) dari sebuah subset.
6. Hitung jarak di(avg) datapoint dengan rata-rata di(avg) *subset*-nya
7. Cari *datapoint* yang memiliki nilai di(avg) yang terdekat dengan rata-rata, dan jadikan *datapoint* tersebut sebagai inisial *centroid*.
8. Hasil akhir dari penentuan inisial *centroid* adalah satu set *centroid* sejumlah K *cluster* yang telah ditentukan.

3.3.1.1 Sorting Dataset Berdasarkan nilai di(avg)

Setelah dilakukan perhitungan nilai di(avg), akan dilakukan proses *sorting dataset* dengan menggunakan algoritma *merge-sort*. Proses *sorting* dimulai dengan menginputkan dataset yang sudah mengalami proses perhitungan di(avg), dan hasil dari tahap ini berupa *dataset* yang telah diurutkan berdasarkan nilai di(avg). Flowchart *sorting dataset* berdasarkan nilai di(avg) ditampilkan pada Gambar 3.4.



Gambar 3. 4 *Sorting Dataset* Dengan Berdasarkan Nilai di(avg)



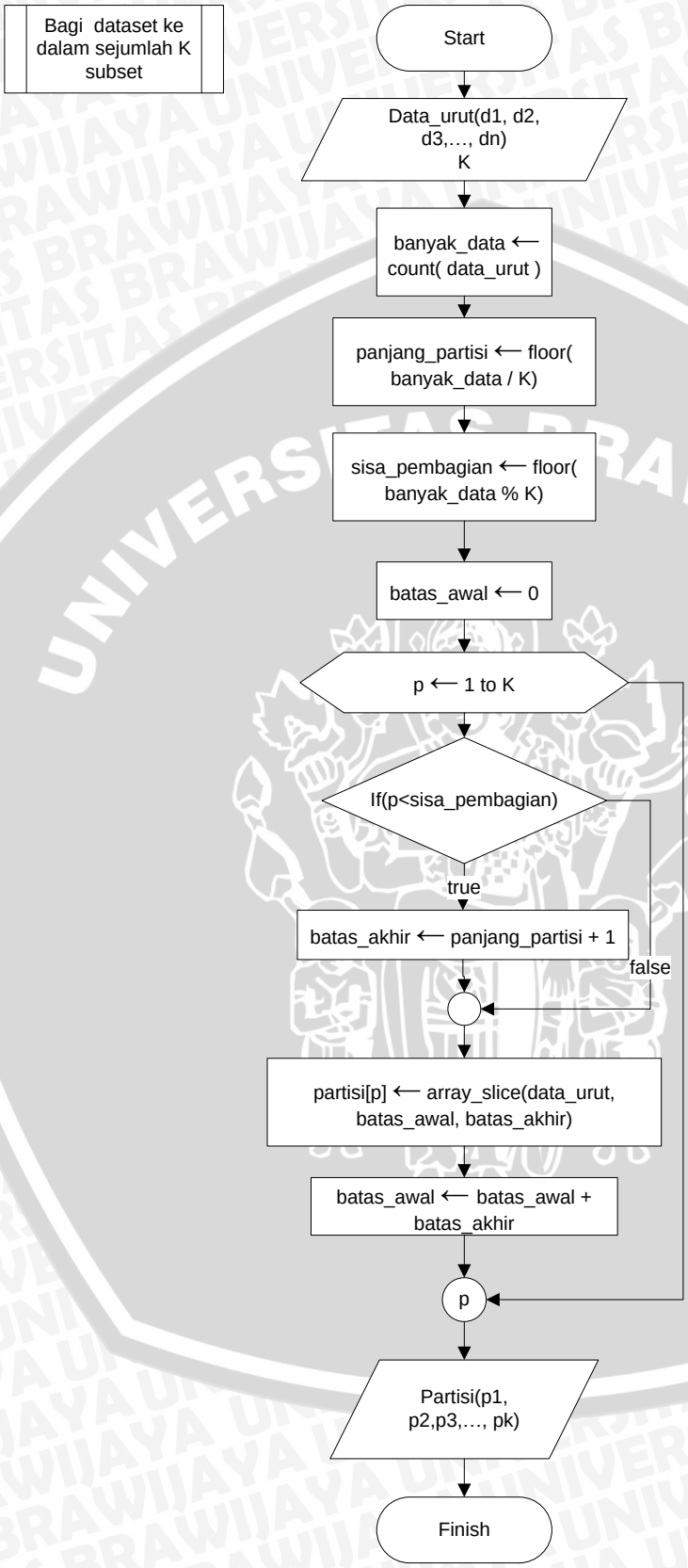
Gambar 3.4 *Sorting Dataset* Dengan Berdasarkan Nilai di(avg)(lanjutan)

Penjelasan flowchart *sorting dataset* berdasarkan nilai di(avg) :

1. Inputan berupa *dataset* yang sudah dihitung nilai di(avg)-nya.
2. Apabila array *dataset* memiliki hanya satu elemen, proses dihentikan.
3. Menghitung nilai tengah *dataset*, kemudian *dataset* dibagi menjadi dua bagian.
4. Setelah data dibagi menjadi dua, data akan dirutkan secara *ascending* dan kemudian digabungkan menjadi *dataset* yang sudah terurut.

3.3.1.2 Membagi Dataset Menjadi Sejumlah K Subset

Setelah didapati *dataset* yang terurut berdasarkan nilai di(avg), kemudian *dataset* akan dibagi menjadi beberapa bagian yang disebut *subset* atau *cluster* sementara. Flowchart perancangan membagi *dataset* menjadi beberapa bagian sejumlah nilai K digambarkan pada Gambar 3.5.



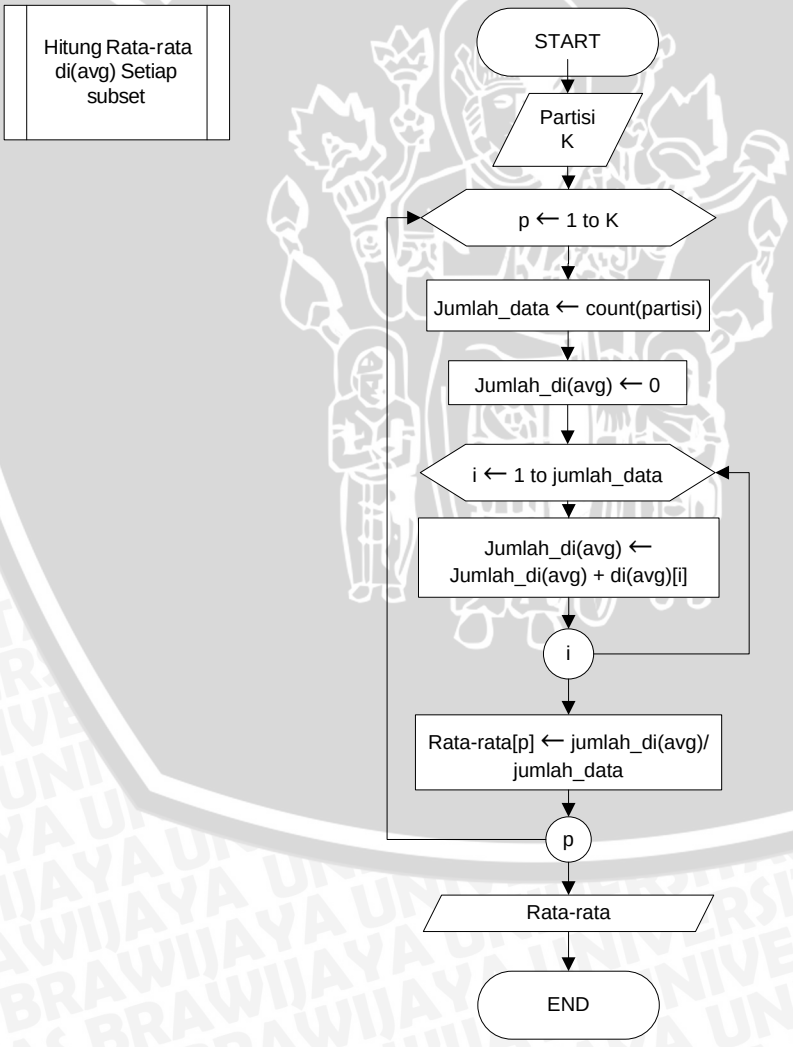
Gambar 3. 5 Membagi Dataset Ke Sejumlah K *Subset*

Penjelasan flowchart pembagian *dataset* menjadi sejumlah K *subset* :

1. Inputan berupa *dataset* yang telah diurutkan.
2. Selanjutnya dilakukan estimasi panjang *subset*.
3. Kemudian akan dilakukan perulangan yang membagi dataset menjadi beberapa bagian sesuai inputan nilai K.

3.3.1.3 Hitung Rata-rata di(avg) Setiap Subset

Setiap *subset* yang telah dihasilkan pada tahap sebelumnya, pada tahap ini akan dihitung nilai rata-rata di(avg)-nya. Sehingga didapati nilai rata-rata sebanyak nilai tiga elemen. Flowchart perancangan menghitung rata-rata nilai di(avg) setiap subset ditampilkan pada Gambar 3.6



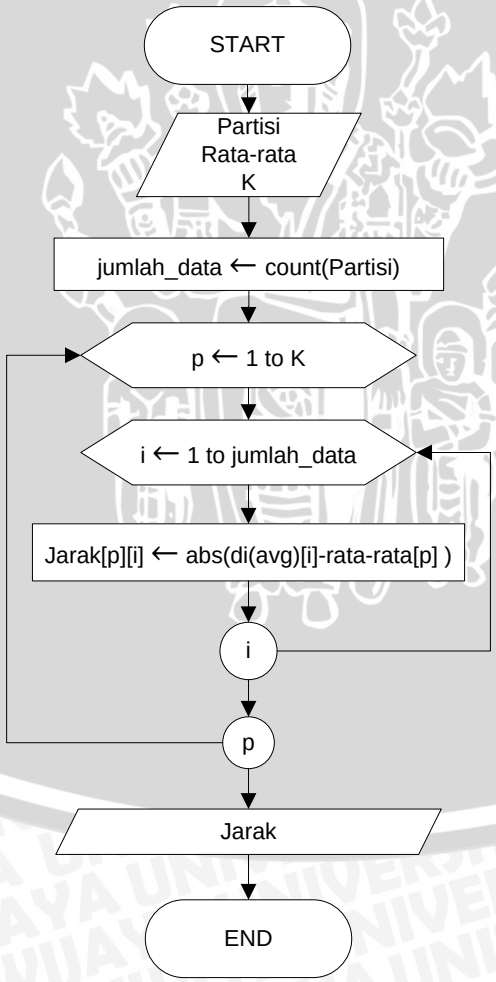
Gambar 3. 6 Hitung Rata-rata di(avg) Setiap *Subset*

Penjelasan flowchart menghitung rata-rata di(avg) setiap *subset* :

1. Dimulai dengan inputan berupa *subset* atau partisi dan jumlah *cluster*.
2. Menghitung nilai jumlah di(avg) dari *datapoint* anggota partisi.
3. Hitung rata-rata di(avg) dengan membagi total jumlah dengan jumlah anggota *subset* atau partisi.

3.3.1.4 Hitung Jarak di(avg) Setiap Datapoint Dengan Rata-rata di(avg) *Subset*-nya

Setelah didapati nilai rata-rata di(avg), tahap selanjutnya adalah menghitung jarak di(avg) setiap *datapoint* dengan rata-rata di(avg) subsetnya. Flowchart menghitung jarak di(avg) datapoint dengan rata-rata di(avg) subset ditampilkan pada Gambar 3.7.



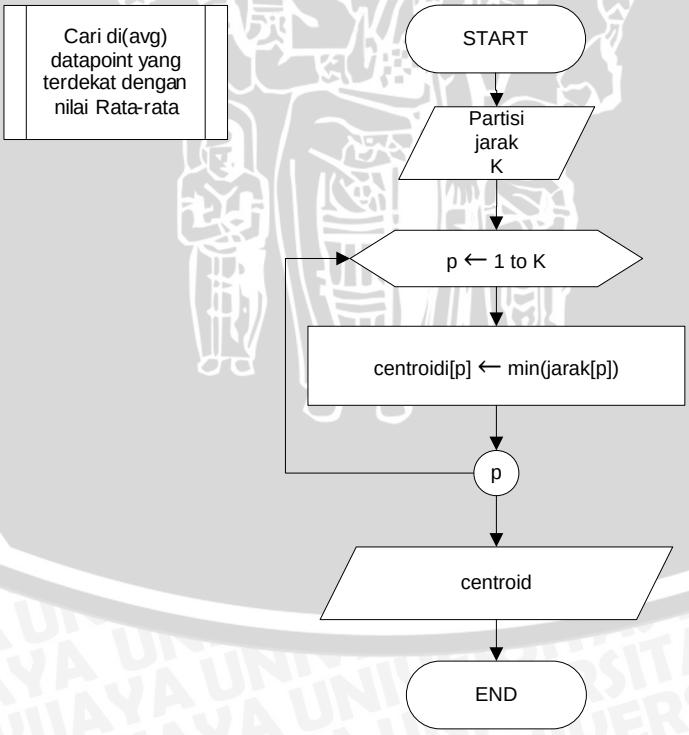
Gambar 3. 7 Hitung Jarak Datapoint Dengan Rata-Rata di(avg) *Subset*

Penjelasan flowchart menghitung jarak datapoint dengan rata-rata di(avg) setiap *subset* :

1. Dimulai dengan inputan berupa *subset* atau partisi beserta rata-ratanya dan jumlah *cluster*.
2. Dari nilai rata-rata yang diperoleh, akan dihiung jarak di(avg) setiap datapoint di(avg) dengan rata-rata *subset*-nya.
3. Pada tahap ini akan dihasilkan jarak setiap data point dengan rata-rata di(avg) *subset* atau partisinya.

3.3.1.5 Mencari Datapoint Yang Memiliki di(avg) Terdekat Dengan Rata-rata di(avg) *Subset*-nya Untuk Dijadikan Inisial *Centroid*

Berdasarkan nilai jarak yang diperoleh sebelumnya, selanjutnya akan dilakukan pencarian *datapoint* yang memiliki jarak dengan rata-rata di(avg) *subset*-nya yang paling dekat atau kecil untuk dijadikan inisial centroid. Flowchart pencarian nilai jarak di(avg) minimum akan ditampilkan pada Gambar 3.8.



Gambar 3. 8 Memilih Inisial Centroid

Penjelasan flowchart menghitung jarak datapoint dengan rata-rata di(avg) setiap *subset* :

1. Dimulai dengan inputan berupa *subset* atau partisi beserta rata-ratanya dan jumlah *cluster*.
2. Dari nilai rata-rata yang diperoleh, akan dihiung jarak di(avg) setiap datapoint di(avg) dengan rata-rata *subset*-nya.
3. Pada tahap ini akan dihasilkan jarak setiap data point dengan rata-rata di(avg) *subset* atau partisinya.

3.3.2 Pembobotan *Attribute*

Pada *improvement K-Means* yang inisialisasi *centroid*-nya ditentukan dengan *weighted average*, nilai bobot dari sebuah attribute memiliki peran penting dalam menentukan inisial *centroid* yang dihasilkan sistem, yang mana nilai inisial centroid tersebut berpengaruh terhadap hasil *cluster*ing. Pada tahap pengujian nilai bobot, nilai bobot diperoleh dari penelitian yang dilakukan oleh Ching-Huse Cheng yang berjudul "*OWA Based Information Fusion Technique For Classification Problem*". Bobot dataset iris yang diperoleh dari penelitian "*OWA Based Information Fusion Technique For Classification Problem*" disajikan oleh Tabel 3.1.

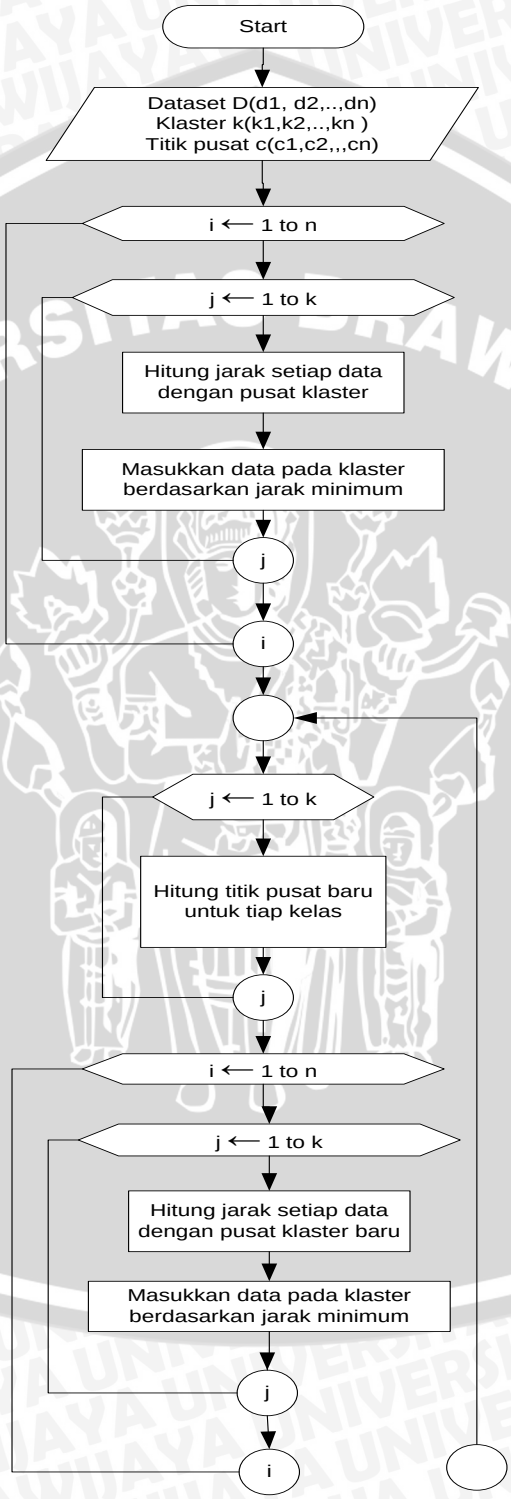
Tabel 3. 1 Tabel Bobot *Attribute* Dataset Iris

Nomor Bobot	*Sepal Length*	*Sepal Width*	*Petal Length*	*Petal Width*
1	0.25	0.25	0.25	0.25
2	0.3475	0.2722	0.2133	0.1671
3	0.4609	0.2754	0.1646	0.0987
4	0.5695	0.2521	0.1065	0.045
5	0.7641	1.822	0.0434	0.0104
6	1	0	0	0

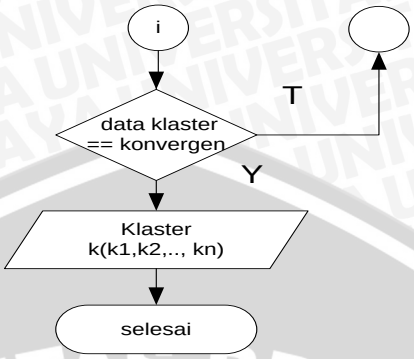
3.3.3 *K-Means Clustering*

Setelah diperolah inisial *centroid* pada tahap sebelumnya, pada tahap ini akan dilakukan tahap *clustering dataset* ke sejumlah k *dataset.* Tahap ini dimulai

dari menginputkan dataset dengan inisial *centroid* yang telah ditentukan pada tahap sebelumnya. Hasil keluaran dari tahap ini adalah berupa *clustering* dataset sejumlah k *cluster*. Flowchart *K-Means* clustering disajikan pada Gambar 3.9.



Gambar 3. 9 Flowchart Implementasi *K-Means clustering*



Gambar 3. 9 Flowchart Implementasi *K-Means clustering* (lanjutan)

3.4 Perhitungan Manual

Perhitungan manual digunakan untuk melakukan implementasi sistem secara bertahap melalui perhitungan matematis pada *dataset*. Berikut ini merupakan contoh dari penerapan perhitungan manual pada *dataset iris*.

Jumlah data yang digunakan pada perhitungan manual ini adalah sebanyak 12 data yang diperoleh dari UCI *Machine Learning Repository*. Terdapat empat *attribute* yang ada pada dataset *iris* antara lain, *petal length, petal width, sepal length,* dan *sepal width*. Perhitungan manual ini bertujuan untuk melakukan pengklasteran *dataset* menjadi tiga *cluster* sama seperti *cluster* sebenarnya dari *dataset iris*, yakni *iris-setosa, iris-versicolor*, dan *iris-virginica*. *Dataset iris* yang digunakan sebagai dataset pada proses *clustering* disajikan pada Tabel 3.2.

Tabel 3. 2 Data *Iris*

No	*sepal length*	*sepal width*	*petal length*	*petal width*	Class
1	5.1	3.5	1.4	0.2	*Iris-setosa*
2	4.9	3	1.4	0.2	*Iris-setosa*
3	4.7	3.2	1.3	0.2	*Iris-setosa*
4	4.6	3.1	1.5	0.2	*Iris-setosa*
5	5.9	3	4.2	1.5	*Iris-versicolor*
6	6.4	3.2	4.5	1.5	*Iris-versicolor*

Tabel 3. 2 Data *Iris* (lanjutan)

No	*sepal length*	*sepal width*	*petal length*	*petal width*	Class
7	6.9	3.1	4.9	1.5	*Iris-versicolor*
8	5.5	2.3	4	1.3	*Iris-versicolor*
9	7.1	3	5.9	2.1	*Iris-virginica*
10	7.3	2.9	6.3	1.8	*Iris-virginica*
11	7.7	2.8	6.7	2	*Iris-virginica*
12	7.2	3.6	6.1	2.5	*Iris-virginica*

Langkah 1

Pada tahap ini akan dilakukan penghitungan untuk menentukan inisial *centroid*. Pertama hitungan nilai *weighted average* pada masing-masing data. Nilai *weighted average* diperoleh dari rata-rata attribute tiap data point dikalikan dengan masing-masing bobotnya. Bobot dari masing-masing attribute ditampilkan pada Tabel 3.3 [CHE-07].

Tabel 3. 3 Bobot Attribute Dataset *Iris*

attribute	bobot
sepal length	0.7641
sepal width	0.1822
petal length	0.0434
petal width	0.0104

Perhitungan *weighted average* dilakukan pada setiap titik data di, dimana i merupakan urutan tiap baris data. Perhitungan diambil dari Tabel 3.1 untuk nomer 1, dan hasil perhitungannya ditampilkan pada Tabel 3.3. nomer 1. Berikut ini adalah penjelasan perhitungannya :

$$d1(avg) = \frac{(5.1 * 0.7641 + 3.5 * 0.1822 + 1.4 * 0.0434 + 0.2 * 0.0104)}{4} =$$

= 1.1493625

Hasil perhitungan *weighted average* masing-masing data ditampilkan pada Tabel 3.4.

Tabel 3. 4 Hasil Perhitungan *Weighted Average*

No	*sepal length*	*sepal width*	*petal length*	*petal width*	di(avg)
1	5.1*0.7641	3.5*0.1822	1.4*0.0434	0.2*0.0104	1.1493625
2	4.9*0.7641	3*0.1822	1.4*0.0434	0.2*0.0104	1.0883825
3	4.7*0.7641	3.3*0.1822	1.3*0.0434	0.2*0.0104	1.0582025
4	4.6*0.7641	3.1*0.1822	1.5*0.0434	0.2*0.0104	1.036715
5	5.9*0.7641	3*0.1822	4.2*0.0434	1.5*0.0104	1.3131675
6	6.4*0.7641	3.2*0.1822	4.5*0.0434	1.5*0.0104	1.421045
7	6.9*0.7641	3.1*0.1822	4.9*0.0434	1.5*0.0104	1.5163425
8	5.5*0.7641	2.3*0.1822	4*0.0434	1.3*0.0104	1.2021825
9	7.1*0.7641	3*0.1822	5.9*0.0434	2.1*0.0104	1.5624025
10	7.3*0.7641	2.9*0.1822	6.3*0.0434	1.8*0.0104	1.5996125
11	7.7*0.7641	2.8*0.1822	6.7*0.0434	2*0.0104	1.6763275
12	7.2*0.7641	3.6*0.1822	6.1*0.0434	2.5*0.0104	1.612045

Langkah 2

Setelah didapati nilai *weighted average* pada masing-masing baris data, tahap selanjutnya adalah mengurutkan baris data berdasarkan nilai *weighted average*-nya. Hasil pengurutan dataset ditampilkan pada Tabel 3.5.

Tabel 3. 5 Pengurutan Berdasarkan Nilai *Weighted Average*

No	*sepal length*	*sepal width*	*petal length*	*petal width*	di(avg)
1	4.6	3.1	1.5	0.2	1.036715
2	4.7	3.2	1.3	0.2	1.0582025
3	4.9	3	1.4	0.2	1.0883825

Tabel 3. 5 Pengurutan Berdasarkan Nilai *Weighted Average*(lanjutan)

No	*sepal length*	*sepal width*	*petal length*	*petal width*	di(avg)
4	5.1	3.5	1.4	0.2	1.1493625
5	5.5	2.3	4	1.3	1.2021825
6	5.9	3	4.2	1.5	1.3131675
7	6.4	3.2	4.5	1.5	1.421045
8	6.9	3.1	4.9	1.5	1.5163425
9	7.1	3	5.9	2.1	1.5624025
10	7.3	2.9	6.3	1.8	1.5996125
11	7.2	3.6	6.1	2.5	1.612045
12	7.7	2.8	6.7	2	1.6763275

Langkah 3

Setelah didapati data sudah terurutkan, maka dilanjutkan dengan membagi data menjadi sejumlah K subset, dengan K bernilai sesuai dengan jumlah kelas *dataset iris* yaitu tiga kelas. Hasil pembagian dataset menjadi tiga subset ditampilkan pada Tabel 3.6.

Tabel 3. 6 Dataset Dibagi Menjadi Tiga Subset

No	*sepal length*	*sepal width*	*petal length*	*petal width*	di(avg)
Subset 1					
1	4.6	3.1	1.5	0.2	1.036715
2	4.7	3.2	1.3	0.2	1.058203
3	4.9	3	1.4	0.2	1.088383
4	5.1	3.5	1.4	0.2	1.149363

Tabel 3. 6 Dataset Dibagi Menjadi Tiga Subset (lanjutan)

No	*sepal length*	*sepal width*	*petal length*	*petal width*	di(avg)
Subset 2					
5	5.5	2.3	4	1.3	1.202183
6	5.9	3	4.2	1.5	1.313168
7	6.4	3.2	4.5	1.5	1.421045
8	6.9	3.1	4.9	1.5	1.516343
Subset 3					
9	7.1	3	5.9	2.1	1.562403
10	7.3	2.9	6.3	1.8	1.599613
11	7.2	3.6	6.1	2.5	1.612045
12	7.7	2.8	6.7	2	1.676328

Langkah 4

Pada tahap ini akan dilakukan perhitungan nillai rata-rata di(avg) dari nilai *weighted average* masing-masing subset. Nilai rata-rata di(avg) dari masing-masing subset ditampilkan oleh Tabel 3.7.

Tabel 3. 7 Rata-Rata di(Avg) Masing-Masing *Subset*

No	*sepal length*	*sepal width*	*petal length*	*petal width*	di(avg)
Subset 1					
1	4.6	3.1	1.5	0.2	1.036715
2	4.7	3.2	1.3	0.2	1.058203
3	4.9	3	1.4	0.2	1.088383
4	5.1	3.5	1.4	0.2	1.149363
				Rata-rata	1.083166
Subset 2					
5	5.5	2.3	4	1.3	1.202183

Tabel 3. 7 Rata-Rata di(Avg) Masing-Masing *Subset* (lanjutan)

No	*sepal length*	*sepal width*	*petal length*	*petal width*	di(avg)
6	5.9	3	4.2	1.5	1.313168
7	6.4	3.2	4.5	1.5	1.421045
8	6.9	3.1	4.9	1.5	1.516343
				Rata-rat	1.363184
Subset 3					
9	7.1	3	5.9	2.1	1.562403
10	7.3	2.9	6.3	1.8	1.599613
11	7.2	3.6	6.1	2.5	1.612045
12	7.7	2.8	6.7	2	1.676328
				Rata-rata	1.612597

Langkah 5

Langkah selanjutnya yaitu menghitung selisih nilai rata-rata di(avg) dengan *weighted average* setiap data pada masing-masing subset. Berdasarkan nilai selisih yang diperoleh, gunakan data yang memiliki selisih terkecil untuk dijadikan sebagai inisial centroid. Pemilihan inisial centroid dijelaskan pada Tabel 3.8.

Tabel 3. 8 Selisih Rata-Rata di(Avg) dan Nilai *Weighted Average*

No	*sepal length*	*sepal width*	*petal length*	*petal width*	di(avg)	Selisih	Selisih terkecil
1	4.6	3.1	1.5	0.2	1.036715	-0.0464506	0.00521688
2	4.7	3.2	1.3	0.2	1.0582025	-0.0249631	
3	4.9	3	1.4	0.2	1.0883825	0.00521688	
4	5.1	3.5	1.4	0.2	1.1493625	0.06619687	
					1.08316563		

Tabel 3. 8 Hasil Perhitungan Selisih Rata-Rata di(Avg) dan Nilai *Weighted Average* (lanjutan)

No	*sepal length*	*sepal width*	*petal length*	*petal width*	di(avg)	Selisih	Selisih terkecil
5	5.5	2.3	4	1.3	1.2021825	0.16100188	0.05001687
6	5.9	3	4.2	1.5	1.3131675	0.05001687	
7	6.4	3.2	4.5	1.5	1.421045	-0.0578606	
8	6.9	3.1	4.9	1.5	1.5163425	-0.1531581	
					1.36318438		
9	7.1	3	5.9	2.1	1.5624025	0.05019438	0.00055188
10	7.3	2.9	6.3	1.8	1.5996125	0.01298438	
11	7.2	3.6	6.1	2.5	1.612045	0.00055188	
12	7.7	2.8	6.7	2	1.6763275	-0.0637306	
					1.61259688		

Berdasarkan penghitungan selisih terkecil, diperoleh inisial centroid untuk masing-masing *cluster*. Inisial centroid tersebut antara lain dijelaskan pada Tabel 3.9.

Tabel 3. 9 Hasil Inisial Centroid

Centroid	*sepal length*	*sepal width*	*petal length*	*petal width*	rata-rata	Selisih
c1	4.9	3	1.4	0.2	1.0883825	0.00521688
c2	5.9	3	4.2	1.5	1.3131675	0.05001687
c3	7.2	3.6	6.1	2.5	1.612045	0.00055188

Langkah 6

Setelah didapati inisial *centroid* untuk masing-masing *cluster* tahap selanjutnya adalah melakukan proses *clustering dataset* menggunakan algoritma *K*-

Means. Berdasarkan dataset dan jumlah *cluster* yang telah ditentukan sebelumnya, yakni 12 data dari *dataset iris* dan tiga *cluster*, akan dilakukan *clustering* data dengan algoritma *K-Means* yang dimulai dari proses perhitungan jarak setiap baris data dengan setiap *centroid*. Perhitungan berikut ini merupakan perhitungan jarak dataset iris nomer 1 pada Tabel 3.2 dengan inisial *centroid* pada Tabel 3.9 dengan menggunakan metode *Euclidean distance*.

$$d1 = \sqrt{(5.1 - 4.9)^2 + (3.5 - 3)^2 + (1.4 - 1.4)^2 + (0.2 - 0.2)^2}$$

$$= 0.538516481$$

$$d2 = \sqrt{(5.1 - 5.9)^2 + (3.5 - 3)^2 + (1.4 - 4.2)^2 + (0.2 - 1.5)^2}$$

$$= 3.228002478$$

$$d3 = \sqrt{(5.1 - 7.2)^2 + (3.5 - 3.6)^2 + (1.4 - 6.1)^2 + (0.2 - 2.5)^2}$$

$$= 5.639148872$$

Setelah dilakukan penghitungan jarak dengan metode *Euclidean distance*, akan didapati jarak masing-masing baris data dengan masing-masing centroid. Hasil penghitungan jarak tersebut dijabarkan pada Tabel 3.10.

Tabel 3. 10 Hitung Jarak *Datapoint* Dengan Masing-Masing *Centroid*

No	***sepal length***	***sepal width***	***petal length***	***petal width***	Jarak *datapoint* dengan		
					Pusat *cluster* 1	Pusat *cluster* 2	Pusat *cluster* 3
1	5.1	3.5	1.4	0.2	0.538516481	3.228002478	5.639148872
2	4.9	3	1.4	0.2	0	3.244996148	5.747173218
3	4.7	3.2	1.3	0.2	0.3	3.402939905	5.894064811

Tabel 3. 10 Hitung Jarak *Datapoint* Dengan Masing-Masing *Centroid* (lanjutan)

No	***sepal length***	***sepal width***	***petal length***	***petal width***	Jarak *datapoint* dengan		
					Pusat *cluster* 1	Pusat *cluster* 2	Pusat *cluster* 3
4	4.6	3.1	1.5	0.2	0.331662479	3.268026928	5.784461946
5	5.9	3	4.2	1.5	3.244996148	0	2.58069758
6	6.4	3.2	4.5	1.5	3.686461718	0.6164414	2.088061302
7	6.9	3.1	4.9	1.5	4.236744033	1.224744871	1.6673332
8	5.5	2.3	4	1.3	2.969848481	0.854400375	3.229551052
9	7.1	3	5.9	2.1	5.357238094	2.165640783	0.754983444
10	7.3	2.9	6.3	1.8	5.686826883	2.543619468	1.014889157
11	7.7	2.8	6.7	2	6.261788882	3.127299154	1.224744871
12	7.2	3.6	6.1	2.5	5.747173218	2.58069758	0

Langkah 7

Setelah dilakukan penghitungan jarak, tahap selanjutnya adalah menempatkan data ke dalam *cluster* yang memiliki jarak *centroid* terkecil. Tahap ini menghasilkan penempatan dataset menuju *cluster* yang memiliki jarak terdekat. Hasil penempatan ditunjukkan oleh Tabel 3.11.

Tabel 3. 11 Penempatan Data Ke Dalam *Cluster* Terdekat

No	*sepal length*	*sepal width*	*petal length*	*petal width*	*Cluster*		
					k1	k2	k3
1	5.1	3.5	1.4	0.2	1	FALSE	FALSE
2	4.9	3	1.4	0.2	1	FALSE	FALSE
3	4.7	3.2	1.3	0.2	1	FALSE	FALSE
4	4.6	3.1	1.5	0.2	1	FALSE	FALSE
5	5.9	3	4.2	1.5	FALSE	1	FALSE
6	6.4	3.2	4.5	1.5	FALSE	1	FALSE
7	6.9	3.1	4.9	1.5	FALSE	1	FALSE
8	5.5	2.3	4	1.3	FALSE	1	FALSE
9	7.1	3	5.9	2.1	FALSE	FALSE	1
10	7.3	2.9	6.3	1.8	FALSE	FALSE	1
11	7.7	2.8	6.7	2	FALSE	FALSE	1
12	7.2	3.6	6.1	2.5	FALSE	FALSE	1

Langkah 8

Pada tahap ini, akan dilakukan penghitungan *centroid* baru yang akan digunakan untuk proses *K-Means* pada iterasi selanjutnya. Penghitungan dilakukan dengan menghitung rata-rata niliai *attribute* yang terdapat pada suatu *cluster*. Berikut ini adalah perhitungan *centroid* baru pada *cluster* 1 *attribute* sepal-length Tabel 3.11. Hasil keseluruhan perhitungan centroid baru akan ditampilkan pada Tabel 3.12 baris centroid baru.

$$\text{Centroid 1 } sepal\ length = \frac{5.1 + 4.9 + 4.7 + 4.6}{4}$$

$$= 4.825$$

Tabel 3. 12 Perhitungan Centroid Baru

Centroid lama	c1				c2				c3			
	4.9	3	1.4	0.2	5.9	3	4.2	1.5	7.2	3.6	6.1	2.5
Centroid baru	4.83	3.2	1.4	0.2	6.18	2.9	4.4	1.45	7.33	3.08	6.25	2.1

Langkah 9

Langkah selanjutnya merupakan iterasi selanjutnya, tahap ini juga akan dilakukan proses yang sama dengan proses *clustering* sebelumnya. Meliputi perhitungan jarak data dengan masing-masing centroid, penempatan data ke dalam *cluster* terdekat, dan perhitungan centroid baru untuk proses iterasi selanjutnya. Proses iterasi akan tetap diteruskan sampai konvergen atau tidak terjadi perubahan nilai centroid baru dengan iterasi yang sebelumnya. Pada Tabel 3.13 ditampilkan hasil perhitungan jarak *datapoint* dengan *centroid* atau pusat *cluster* yang baru.

Tabel 3. 13 Hitung Jarak *Datapoint* Dengan *Centroid* Baru Iterasi 2

No	***sepal length***	***sepal width***	***petal length***	***petal width***	Jarak *datapoint* dengan		
					Centroid baru 1	Centroid baru 2	Centroid baru 3
1	5.1	3.5	1.4	0.2	0.406970515	3.475359694	5.680118837
2	4.9	3	1.4	0.2	0.213600094	3.492581424	5.746194393
3	4.7	3.2	1.3	0.2	0.160078106	3.665804823	5.917664235

Tabel 3. 13 Tabel Hasil Penghitungan Jarak *Datapoint* Dengan Masing-Masing *Centroid* Baru Iterasi 2 (lanjutan)

No	*sepal length*	*sepal width*	*petal length*	*petal width*	Jarak *datapoint* dengan		
					Centroid baru 1	Centroid baru 2	Centroid baru 3
4	4.6	3.1	1.5	0.2	0.265753645	3.5345615	5.796442875
5	5.9	3	4.2	1.5	3.275	0.357945527	2.568803223
6	6.4	3.2	4.5	1.5	3.712226421	0.391311896	2.072136579
7	6.9	3.1	4.9	1.5	4.272660178	0.904502626	1.537449186
8	5.5	2.3	4	1.3	3.039017111	0.999062 06	3.103828281
9	7.1	3	5.9	2.1	5.39218184	1.880990 431	0.422788363
10	7.3	2.9	6.3	1.8	5.725873296	2.235648 675	0.351781182
11	7.7	2.8	6.7	2	6.305206182	2.815692 632	0.654790043
12	7.2	3.6	6.1	2.5	5.760262581	2.352259 552	0.688294995

Tahap selanjutnya adalah menempatkan datapoint ke dalam *cluster* terdekat. Penempatan *datapoint* ke dalam *cluster* terdekat disajikan pada Tabel 3.14.

Tabel 3. 14 Penempatan *Datapoint* Dalam *Cluster* Iterasi 2

No	*sepal length*	*sepal width*	*petal length*	*petal width*	*Cluster*		
					k1	k2	k3
1	5.1	3.5	1.4	0.2	1	FALSE	FALSE
2	4.9	3	1.4	0.2	1	FALSE	FALSE
3	4.7	3.2	1.3	0.2	1	FALSE	FALSE
4	4.6	3.1	1.5	0.2	1	FALSE	FALSE
5	5.9	3	4.2	1.5	FALSE	1	FALSE
6	6.4	3.2	4.5	1.5	FALSE	1	FALSE
7	6.9	3.1	4.9	1.5	FALSE	1	FALSE
8	5.5	2.3	4	1.3	FALSE	1	FALSE
9	7.1	3	5.9	2.1	FALSE	FALSE	1
10	7.3	2.9	6.3	1.8	FALSE	FALSE	1
11	7.7	2.8	6.7	2	FALSE	FALSE	1
12	7.2	3.6	6.1	2.5	FALSE	FALSE	1

Setelah dilakukan penempatan *cluster*, selanjutnya akan dilakukan perhitungan centroid baru menggunakan cara yang sama dengan iterasi selanjutnya, yaitu dengan menghitung rata-rata niliai *attribute* yang terdapat pada suatu *cluster*. Hasil perhitungan centroid baru ditampilkan pada Tabel 3.15.

Tabel 3. 15 Hasil Centroid Baru Iterasi 2

Centroid lama	c1				c2				c3			
	4.83	3.2	1.4	0.2	6.18	2.9	4.4	1.45	7.33	3.08	6.25	2.1
Centroid baru	4.83	3.2	1.4	0.2	6.18	2.9	4.4	1.45	7.33	3.08	6.25	2.1

Pada perhitungan centroid baru iterasi kedua didapati tidak terjadi perubahan nilai *centroid* atau dapat dikatakan sudah konvergen. Sehingga proses iterasi

dihentikan sampai pada iterasi ini. Hasil akhir proses *clustering* disajikan pada Tabel 3.16.

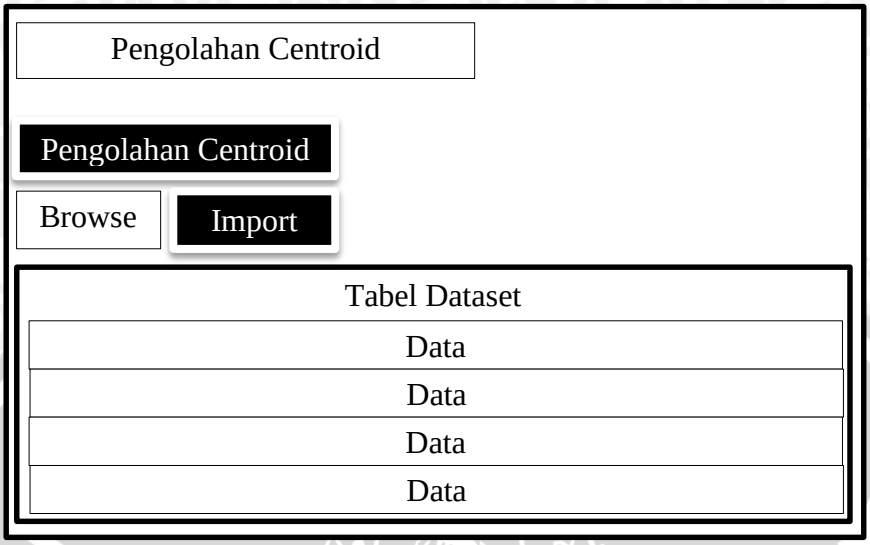
Tabel 3. 16 Hasil Akhir *Cluster*ing Data

	sepal length	***sepal width***	***petal length***	***petal width***	*cluster*
1	5.1	3.5	1.4	0.2	1
2	4.9	3	1.4	0.2	1
3	4.7	3.2	1.3	0.2	1
4	4.6	3.1	1.5	0.2	1
5	5.9	3	4.2	1.5	2
6	6.4	3.2	4.5	1.5	2
7	6.9	3.1	4.9	1.5	2
8	5.5	2.3	4	1.3	2
9	7.1	3	5.9	2.1	3
10	7.3	2.9	6.3	1.8	3
11	7.7	2.8	6.7	2	3
12	7.2	3.6	6.1	2.5	3

3.5 Perancangan Antarmuka

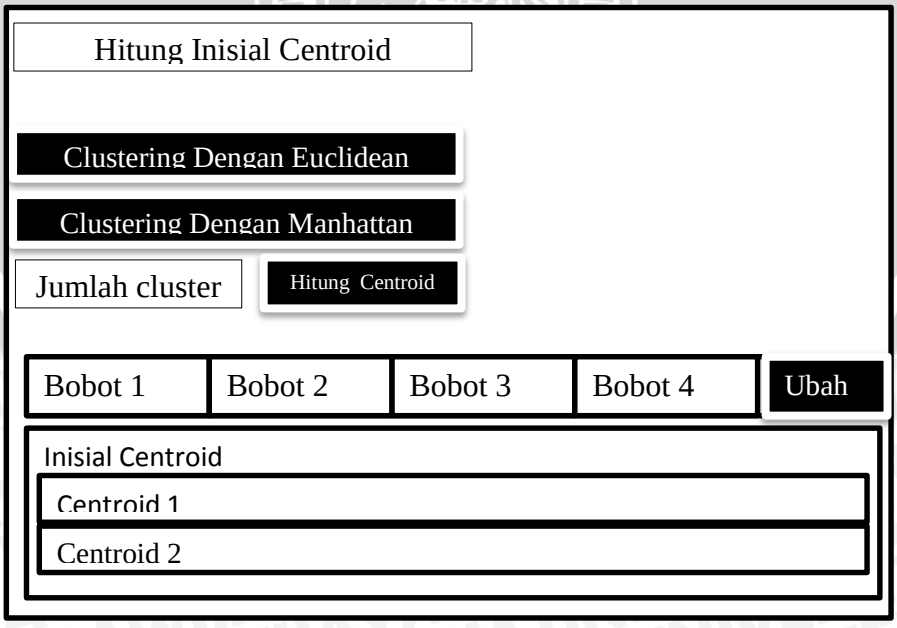
Perancangan antarmuka merupakan tahap yang menggambarkan sebuah antarmuka yang digunakan oleh pengguna agar dapat berkomunikasi dengan sistem. Pada antarmuka terbagi menjadi tiga bagian utama yaitu :

1. Antarmuka input berfungsi untuk melakukan input data berupa *dataset* yangakan dipergunakan. Terdapat tombol browse untuk memilih file *dataset* yang akan digunakan. Saat tombol import ditekan, dataset akan dimasukan database. Tombol pengolaha centroid digunakan untuk menuju halaman olah centroid. Tampilan Antarmuka input data latih digambarkan pada Gambar 3.10.



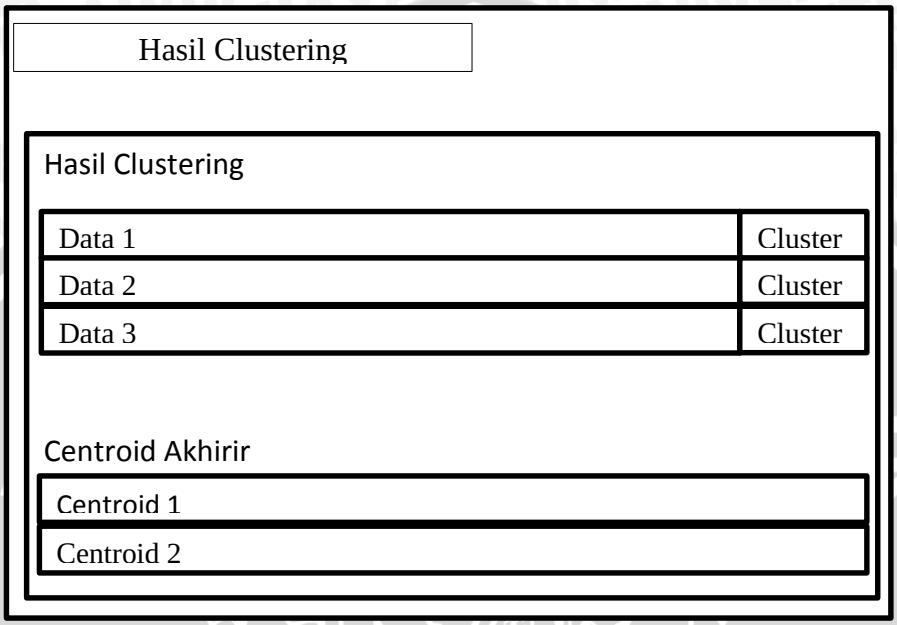
Gambar 3. 10 Antarmuka Input Data Latih

2. Antarmuka lihat centroid berfungsi untuk melakukan proses penentuan inisial centroid yang digunakan oleh sistem. Terdapat form ubah bobot, form jumlah cluster. Tombol hitung *centroid* untuk memulai olah centroid. digunakan untuk Tampilan Antarmuka lihat centroid data latih digambarkan pada Gambar 3.11.



Gambar 3. 11 Antarmuka Lihat *Centroid* Data Latih

3. Antarmuka hasil *cluster*ing berfungsi untuk menampilkan hasil akhir clustering. Pada halaman antarmuka akan ditampilkan data beserta *cluster*-nya dan *centroid* akhir proses clustering. Tampilan Antarmuka hasil *cluster*ing data latih digambarkan pada Gambar 3.12.



Gambar 3. 12 Antarmuka Hasil *Clustering* Data Latih

3.6 Perancangan Pengujian dan Analisis

Tahapan perancangan pengujian bertujuan untuk menguji sistem yang telah dibangun, kemudian dilanjutkan dengan melakukan analisa terhadap hasil pengujian sistem. Pengujian dilakukan dengan membandingkan metode *cluster*ing data algoritma *K-Means* konvensional dengan *improvement K-Means* yang inisial *centroid*-nya telah ditentukan dengan metode *weighted average.*

Hasil pengujian yang dilakukan adalah dengan membandingkan kualitas *clustering* dan keakurasian hasil *clustering* masing-masing algoritma, *K-Means* konvensional dan *improvement K-Means* yang inisial *centroid*-nya telah ditentukan dengan metode *weighted average*.

3.6.1 Bahan Pengujian

Bahan pengujian yang digunakan pada penelitian ini adalah *dataset iris* yang diperoleh dari UCI *Machine Learning Repository*. Dataset *iris* memiliki empat

attribute, yaitu *sepal length*, *sepal width*, *petal length*, dan *petal width*. Jumlah data *iris* pada UCI *Machine Learning Repository* data *iris* sebanyak 150 data.

3.6.2 Skenario Pengujian

Pada penelitian ini akan dilakukan pengujian sistem yang terdiri dari tiga pengujian, yaitu :

- Pengujian nilai bobot pada *improve K-Means*, pengujian nilai bobot dilakukan untuk menganalisa seberapa besar bobot berpengaruh terhadap hasil kualitas *clustering*, serta untuk menemukan bobot yang terbaik untuk *clustering* dataset iris. Pada pengujian bobot akan digunakan data iris sebanyak 150 data.
- Pengujian *silhoette coefficient improve K-Means* dengan *K-Means* konvensional. Pada pengujian ini diharapkan dapat diketahui perbandingan kualitas antara *improve K-Means* dengan *K-Means* konvensional. Pada pengujian nilai *silhouette coefficient*, pengujian dilakukan dengan menggunakan jumlah data yang bervariasi, yaitu 75 data, 105 data, 120 data, dan 150 data. Pengujian dengan menggunakan jumlah data yang bervariasi bertujuan untuk mengetahui jumlah data yang memiliki kualitas *clustering* paling dominan. Saat diperoleh jumlah data yang menghasilkan kualitas paling dominan, jumlah data tersebut akan digunakan untuk menguji perbandingan kualitas *clustering* antara *improve K-Means* dengan *K-Means* konvensional.
- Pengujian akurasi, pengujian akurasi dilakukan untuk mengetahui perbandingan kemiripan data pada suatu *cluster* antara *improve K-Means* dengan *K-Means* konvensional berdasarkan informasi label atau kelas sebenarnya yang dimiliki oleh suatu *dataset*. Pengujian akurasi dilakukan dengan menggunakan 105 data latih dan 45 data uji.

3.7 Kesimpulan dan Saran

Penarikan Kesimpulan dilakukan setelah semua tahapan metode penelitian selesai dilakukan. Kesimpulan diperoleh berdasarkan hasil pengujian dan hasil

analisa terhadap implementasi program yang telah dibangun dari tahap pengujian dan analisis. Setelah didapati kesimpulan penelitian selanjutnya pada tahap ini dilakukan penulisan saran. Penulisan saran bertujuan untuk memperbaiki kekurangan yang terjadi pada penelitian ini, serta memberikan pertimbangan untuk pengembangan atau penelitian selanjutnya.

BAB IV
IMPLEMENTASI

Implementasi merupakan tahapan untuk menerapkan atau mengimplementasikan perancangan yang telah dilakukan sebelumnya. Pada bab ini akan dilakukan pembahasan yang meliputi lingkungan implementasi, implementasi kode program, dan implementasi tampilan antarmuka.

4.1 Lingkungan Implementasi

Lingkungan implementasi membahas tentang berbagai komponen yang digunakan dalam pengembangan sistem. Lingkungan implementasi dalam implementasi *K-Means* dengan inisialisasi *centroid* menggunakan *weighted average* meliputi, lingkungan perangkat keras dan lingkungan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Lingkungan perangkat keras yang digunakan dalam mendukung penelitian ini antara lain :

1. Laptop Prosesor Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz
2. Memori RAM 4 GB DDR3
3. Hardisk 600GB

4.1.2 Lingkungan Perangkat Lunak

Lingkungan perangkat lunak meliputi :

1. Sistem Operasi Windows 8 64bit.
2. XAMPP versi 1.8.3 yang termasuk Apache Web Server sebagai server untuk implementasi PHP dan MySQL sebagai *database management system* (DBMS).
3. Notepad++ untuk implementasi bahasa pemrograman web.
4. Microsoft Visio 2008 guna pembuatan berbagai macam diagram saat proses perancangan aplikasi.
5. Browser Google Chrome versi 43.

4.2 Batasan Implementasi

Dalam mengimplementasikan metode *K-Means* dengan inisialisasi centroid menggunakan *weighted average* terdapat bebrapa batasan implementasi, antara lain :

- Perangkat yang dirancang dan dibangun menggunakan bahasa pemrograman PHP.
- Input dataset yang diinputkan berupa *dataset iris* yang berbentuk .xls, atau xlsx.
- Output yang dihasilkan berupa hasil *clustering* dataset iris.
- Aplikasi berbasis web dengan menggunakan media penyimpanan database MySQL.

4.3 Implementasi Program

Implementasi program dilakukan berdasarkan tahap perancangan yang telah ditentukan sebelumnya. Secara garis besar implementasi program meliputi tahap penentuan inisial *centroid* dan kemudian dilanjutkan dengan K-Means *Cluster*ing.

4.3.1 Penentuan Inisial centroid

Tahap penentuan inisial centroid terdiri dari beberapa tahap, antara lain :

1. Membagi dataset menjadi sejumlah K *subset*.
2. Manghitung rata-rata di(avg) dari masing-masing subset.
3. Menentukan *centroid* dari masing-masing subset yang memiliki nilai terdekat dengan di(avg)-nya untuk dijadikan sebagai inisial *centroid.*

4.3.1.1 Perhitungan di(avg)

Nilai di(avg) diperoleh melalui perhitungan rata-rata dari *attribute* datapoint yang dikalikan dengan bobot *attribute*. Kode Implementasi 4.1 menunjukkan proses perhitungan di(avg).

```
1  $no_datapoint = 0;
2  foreach($iris->result_array() as $row){
3        $row['sepal_length'];
4        $row['sepal_width'];
5        $row['petal_length'];
```

```
6       $avg[$no_datapoint] =
7   ($row['sepal_length']*$w_sepal_length+$row['sepal_width']*$
8   w_sepal_width+$row['petal_length']*$w_petal_length+$row['pe
9   tal_width']*$w_petal_width)/4;
10      $dataset_avg[$no_datapoint] =
11  array("avg"=>$avg[$no_datapoint],
12      "sepal_length"=>$row['sepal_length'],
13      "sepal_width"=>$row['sepal_width'],
14      "petal_length"=>$row['petal_length'],
15      "petal_width"=>$row['petal_width']);
16      $no_datapoint++;
```

Kode Implementasi 4. 1 Perhitungan Nilai Di(Avg)

Penjelasan *sources code*, baris :

1. Baris 2 sampai 6 untuk inisialisasi array row yang berisi *dataset*.
2. Baris 7 sampai 10 untuk perhitungan nilai di(avg).
3. Baris 11 sampai 18 mengiputkan nilai di(avg) ke dalam array $dataset_avg.

4.3.1.2 Pengurutan *Datapoint*

Kode Implementasi 4.2 menunjukkan proses pengurutan datapoint secara *ascending* dengan menggunakan *merg- sort*.

```
1   if(count($dataset)>1) {
2           $data_middle = round(count($dataset)/2, 0,
3   PHP_ROUND_HALF_DOWN);
4           $data_part1 = mergesort(array_slice($dataset, 0,
5   $data_middle));
6           $data_part2 = mergesort(array_slice($dataset,
7   $data_middle, count($dataset)));
8           $counter1 = $counter2 = 0; //inisialisasi counter
9           for ($i=0; $i<count($dataset); $i++) {
10              if($counter1 == count($data_part1)) {
11                  $dataset[$i] = $data_part2[$counter2];
12                  ++$counter2;
13              } elseif (($counter2 == count($data_part2)) or
14  ($data_part1[$counter1] < $data_part2[$counter2])) {
15                  $dataset[$i] = $data_part1[$counter1];
16                  ++$counter1;
17              } else {
18                  $dataset[$i] = $data_part2[$counter2];
19                  ++$counter2;}
20          }
21      }
```

Kode Implementasi 4. 2 Pengurutan Datapoint

Penjelasan *sources code*, baris

1. Baris 1 memastikan jika array yang akan diurutkan memiliki data lebih dari satu.

2. Baris 2 dan 3 untuk mencari titik tengah array.
3. Baris 4 sampai 7 untuk membagi array menjadi dua bagian, yaitu $data_part1 dan $data_part2, baris 4 dan 5 untuk bagian $data_part1, dan baris 4 dan 7 bagian $data_part2.
4. Baris 9 sampai 21 untuk melakukan penggabungan dan pengurutan data secara iteratif.

4.3.1.3 Membagi Dataset Ke Sejumlah K Subset

Dataset yang telah diurutkan kemudian dibagi menjadi sejumlah K *subset*, dengan nilai K merupakan jumlah *cluster* yang digunakan untuk *clustering*. Pembagian datset menjadi sejumlah K subset ditunjukkan oleh Kode Implementasi 4.3.

```
1       $listlen = count( $dataset );
2       $partlen = floor( $listlen / $jml_partisi );
3       $partrem = $listlen % $jml_partisi;
4       $partition = array();
5       $mark = 0;
6       for ($px = 0; $px < $jml_partisi; $px++) {
7           $incr = ($px < $partrem) ? $partlen + 1 : $partlen;
8           $partition[$px] = array_slice( $dataset, $mark,
9   $incr );
10          $mark += $incr;
11      }
```

Kode Implementasi 4. 3 Membagi Dataset Menjadi Sejumlah K Subset

Penjelasan *sources code*:

1. Baris 1 menghitung data pada array $dataset.
2. Baris 2 untuk mengitung perkiraan jumlah data dalam partisi.
3. Baris 3 untuk menghitung nilai sisa pembagian dari baris 2.
4. Baris 6 dan 11 untuk mengulangi proses sebanyak jumlah cluster.
5. Baris 7 menentukan jumlah data dalam partisi, apabila terdapat nilai sisa pada baris 3 akan dialokasikan disini.
6. Baris 8 melakukan pemotongan array sesuai nilai $mark dan $incr.

4.3.1.4 Menghitug Rata-rata Nilai di(avg) Setiap Subset

Setelah dataset dibagi menjadi sejumlah K subset, selanjutnya yang dilakukan adalah menghitung rata-rata nilai di(avg) masing-masing subset.

Perhitungan rata-rata nilai di(avg) masing-masing subset ditunjukkan oleh Kode Implementasi 4.4.

```
1   for($no_cluster=0; $no_cluster<=$jumlah_kelas_index;
2   $no_cluster++ ){
3         $average = array('avg' => 0); //inisialisasi nilai
4   'avg' dengan nilai 0
5         $i = count($partition[$no_cluster]);  //hitung jumlah
6   data
7         //menjumlah nilai avg
8         foreach($partition[$no_cluster] as $value){
9            $average['avg'] += $value['avg'];
10        }
11        //menghitung rata-rata
12        $averagefix[$no_cluster] = $average['avg']/$i ;
13  }
```

Kode Implementasi 4. 4 Menghitung Rata-Rata di(avg) Setiap Subset

Penjelasan *sources code*, baris :

1. Baris 1 dan 3 untuk melakukan pengulangan *query* di dalam for sebanyak jumlah *cluster*
2. Baris 5 menghitung jumlah data pada suatu *cluster*.
3. Baris 7 sampai 10 untuk menjumlahkan nilai di(avg) suatu *subset*.
4. Baris 12 menghitung nilai rata-rata di(avg) suatu *subset*.

4.3.1.5 Menghitung Jarak di(avg) *Datapoint* Dengan Rata-rata di(avg)

Langkah selanjutnya adalah membandingkan rata-rata di(avg) dengan nilai di(avg)setiap *datapoint*. Setelah ditemukan nilai di(avg) suatu *datapoint* yang terdekat dengan rata-rata di(avg), kemudian jadikan sebagai inisial *centroid*. Kode Implementasi 4.5 menunjukkan pemilihan *centroid* berdasarkan di(avg) yang terdekat dengan rata-rata di(avg).

```
1   $partition_distance = array();
2   for($no_cluster=0; $no_cluster<=$jumlah_kelas_index;
3   $no_cluster++ ){
4         $index_distance = 0;
5         foreach($partition[$no_cluster] as $row){
6               $distance[$index_distance] =
7   array("min"=>abs($row['avg']-$averagefix[$no_cluster]),
8         "avg"=>$row['avg'],
9         "sepal_length"=>$row['sepal_length'],
10        "sepal_width"=>$row['sepal_width'],
11        "petal_length"=>$row['petal_length'],
12        "petal_width"=>$row['petal_width']);
13              $index_distance++;
```

```
14          }
15          $partition_distance[$no_cluster] = $distance;
16  }
```

Kode Implementasi 4. 5 Hitung Jarak *Datapoint* Dengan Rata-rata di(avg)

Penjelasan *sources code* :

1. Baris 1 untuk deklarasi array $partition_distance.
2. Baris 2 dan 16 untuk pengulangan query di dalam perulangan sebanyak jumlah *cluster*.
3. Baris 5 dan 14 untuk melakukan perulangan *query* sebanyak jumlah data pada suatu *cluster*.
4. Baris 6 hingga 13 untuk deklarasi array $distance;
5. Pada baris terjadi perhitungan jarak antara nilai di(avg) *datapoint* dengan rata-rata di(avg) subset *datapoint* tersebut.

4.3.1.6 Menjadikan Datapoint Yang Memiliki Nilai di(avg) Terdekat Dengan Rata-rata di(avg) Sebagai Inisial Centroid

Dari nilai jarak yang telah dihitung pada tahap sebelumnya, selanjutnya akan dilakukan pencarian data yang memiliki jarak terdekat atau terkecil.

```
1   for($no_cluster=0; $no_cluster<=$jumlah_kelas_index;
2   $no_cluster++ ){
3         $partition_minimum[$no_cluster] =
4   min($partition_distance[$no_cluster]);
5         extract($partition_minimum[$no_cluster]);
6         $q = "INSERT INTO centroid_awal values
7   ('$sepal_length','$sepal_width','$petal_length','$petal_wid
8   th')";
9         $this->db->query($q);
10  }
```

Kode Implementasi 4. 6 Memilih Datapoint Sebagai Inisial Centroid

1. Baris 1, 2 dan 11 untuk perulangan proses sebanyak jumlah *subset* atau *cluster*.
2. Baris 3 dan 4 untuk mencari *datapoint* yang memiliki di(avg) terdekat dengan rata-rata di(avg).
3. Baris 5 untuk ekstrak array $partition_minimum agar dapat dimasukkan pada *database*.
4. Baris 6 sampai 9 untuk memasukkan nilai *datapoint* yang memiliki di(avg) terdekat dengan rata-rata di(avg) pada tabel centroid_awal.

4.3.2 *K-Means Clustering*

Pada tahap *K-Means clustering* terbagi menjadi tiga tahap utama yang terdiri dari :

4.3.2.1 Menghitung Jarak Datapoint Dengan Centroid

Pada penelitian ini terdapat dua cara dalam menghitung jarak *datapoint* dengan *centroid,* yaitu dengan *euclidean* dan *manhattan*. Pada Kode Implementasi 4.7 menujukkan perhitungan jarang menggunakan *euclidean*, sedangkan perhitungan *manhattan* ditunjukkan Kode Implementasi 4.8.

```
1  for($no_cluster=1;$no_cluster<$counter; $no_cluster++){
2            $jarak[$no_cluster] =
3  sqrt(pow(($s['sepal_length']-
4  ${"c".$no_cluster."a"}),2)+pow(($s['sepal_width']-
5  ${"c".$no_cluster."b"}),2)+pow(($s['petal_length']-
6  ${"c".$no_cluster."c"}),2)+pow(($s['petal_width']-
7  ${"c".$no_cluster."d"}),2));
8        }
```

Kode Implementasi 4. 7 Hitung Jarak Datapoint Menggunakan Euclidean

Penjelasan *sources code* :

1. Baris 1 dan 8 untuk melakukan perulangans sebanyak jumlah *cluster* yang diinputkan.
2. Baris 3 sampai 8 menghitung jarak *datapoint* dengan *centroid* menggunakan *euclidean*.

```
1  for($no_cluster=1;$no_cluster<$counter; $no_cluster++){
2        $jarak[$no_cluster] = abs($s['sepal_length']-
3  ${"c".$no_cluster."a"})+abs($s['sepal_width']-
4  ${"c".$no_cluster."b"})+abs($s['petal_length']-
5  ${"c".$no_cluster."c"})+abs($s['petal_width']-
6  ${"c".$no_cluster."d"});
7        }
```

Kode Implementasi 4. 8 Hitung Jarak Datapoint Menggunakan *Manhattan*

Penjelasan *sources code* :

1. Baris 1 dan 7 untuk melakukan perulangan sebanyak jumlah *cluster*.
2. Baris 2 sampai 6 menghitung jarak *datapoint* dengan *centroid* menggunakan *manhattan*.

4.3.2.2 Menempatkan Datapoint Pada Suatu Cluster

Datapoint ditempatkan menuju *cluster* yang jarak *datapoint* terhadap *centroid cluster*-nya yang terkecil. Berdasar jarak yang diperoleh pada proses sebelumnya, selanjutnya semua jarak tersebut dibandingkan dan dicari mana yang memiliki nilai paling kecil. Kode Implementasi 4.9 menunjukkan penempatan suatu *datapoint* ke dalam suatu *cluster*.

```
1   $minimum=min($jarak);
2   $index=0;
3   for($no_cluster=1;$no_cluster<$counter;$no_cluster++){
4                       $cluster[$no_cluster][$no]=0;
5                       }
6                 foreach ($jarak as $key => $val) {
7                       if($val==$minimum){
8                         $cluster[$key][$no] = 1; //
9                         $class = $key;
10                      }
11                }
```

Kode Implementasi 4. 9 Penempatan Datapoint Dalam *Cluster*

Penjelasan *sources code*, baris :

1. Baris 1 untuk mencari nilai minimum dari array $jarak.
2. Baris 3 sampai 5 untuk inisialisasi *default* nilai 0 pada keterangan cluster yang berarti bukan anggota *cluster*.
3. Baris 6 sampai 11 memberikan nilai 1 pada suatu *cluster* apabila jarak suatu *datapoint* terhadap suatu *centroid cluster* merupakan jarak terkecil dibanding jarak dengan *centroid cluster* lainnya.

4.3.2.3 Menghitung Centroid Baru

Centroid baru digunakan untuk melakukan proses *clustering* ke iterasi selanjutnya. Centroid baru diperoleh melalui perhitungan rata-rata data yang masuk ke dalam suatu *cluster*. Perhitungan *centroid* baru ditunjukkan oleh Kode Implementasi 4.9.

```
1   for($no_cluster=1;$no_cluster<$counter;$no_cluster++{
2                       $jum = 0;
3                       $arr = array();
4         for($i=0;$i<count($cluster[$no_cluster]);$i++){
5         $arr[$i] = $attribute1[$i]*$cluster[$no_cluster][$i];
6               if($cluster[$no_cluster][$i]==1){
7                                       $jum++;
8                 }
```

```
9           }
10          $c_baru[$no_cluster][1] = array_sum($arr)/$jum;
11                  $jum = 0;
12                  $arr = array();
13          for($i=0;$i<count($cluster[$no_cluster]);$i++){
14          $arr[$i] = $attribute2[$i]*$cluster[$no_cluster][$i];
15                if($cluster[$no_cluster][$i]==1){
16                                  $jum++;
17                }
18          }
19          $c_baru[$no_cluster][2] = array_sum($arr)/$jum;
20                  $jum = 0;
21                  $arr = array();
22          for($i=0;$i<count($cluster[$no_cluster]);$i++){
23          $arr[$i] = $attribute3[$i]*$cluster[$no_cluster][$i];
24                if($cluster[$no_cluster][$i]==1){
25                  $jum++;
26              }
27          }
28          $c_baru[$no_cluster][3] = array_sum($arr)/$jum;
29                  $jum = 0;
30                  $arr = array();
31          for($i=0;$i<count($cluster[$no_cluster]);$i++){
32          $arr[$i] = $attribute4[$i]*$cluster[$no_cluster][$i];
33                if($cluster[$no_cluster][$i]==1){
34                  $jum++;
35                }
36          }
37          $c_baru[$no_cluster][4] = array_sum($arr)/$jum;
38                }
```

Kode Implementasi 4. 10 Menghitung Centroid Baru

Penjelasan *sources code*, baris :

1. Melakukan perhitungan centroid baru di setiap cluster.
2. Baris 2 sampai 10 untuk menghitung centroid baru attribute *sepal length*.
3. Baris 11 sampai 19 untuk menghitung centroid baru attribute *sepal width*.
4. Baris 20 sampai 28 untuk menghitung centroid baru attribute *petal length*.
5. Baris 29 sampai 37 untuk menghitung centroid baru attribute *petal width*.

4.3.2.4 Memeriksa Terjadinya Perubahan Cluster

Pada algoritma *K-Means* iterasi akan dihentikan saat tidak terjadi perubahan *cluster suatu datapoint*, sehingga diperlukan sebuah pemeriksaan dengan membandingkan hasil penempatan *cluster* terbaru dan penempatan *cluster* sebelumnya. Pemeriksaan Perubahan *cluster* atau pemeriksaan konvergen ditunjukkan Kode Implementasi 4.11.

```
1   foreach($it_sebelum->result() as $it_prev){
2         $cluster_sebelum[$no] = $it_prev->klaster;
3         $no++;
4   }
5   foreach($it_sesudah->result() as $it_next){
6         $cluster_sesudah[$no] = $it_next->klaster;
7         $no++;
8   }
9   if($cluster_sebelum==$cluster_sesudah){
10     alert("");
11     }else{
12     $this>load>view('kmeans/next_iterasi',$data);}
13  }
```

Kode Implementasi 4. 11 Pemeriksaan Perubahan *Cluster*

Penjelasan *sources code* :

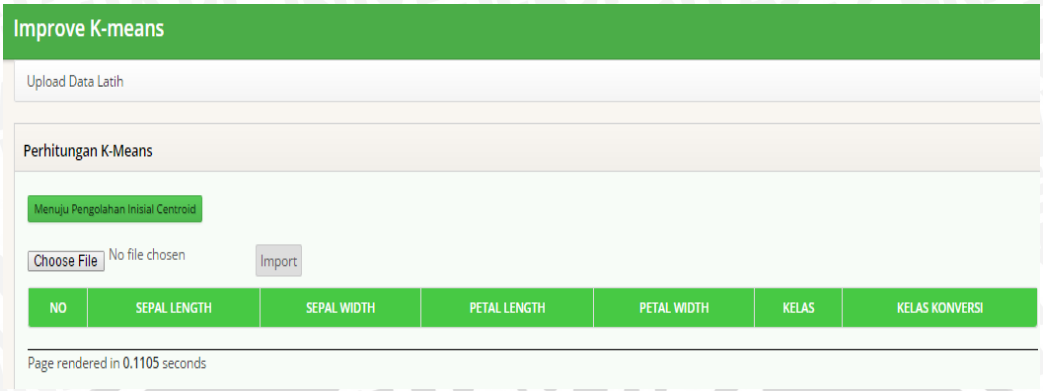
1. Baris 1 sampai 5 inisialisasi *cluster* sebelum iterasi yang baru dilakukan.
2. Baris 6 sampai 10 inisialisasi *cluster* iterasi yang baru dilakukan.
3. Baris 9 sampai 13 untuk membandingkan apakah terjadi perubahan anggota cluster. Jika sama proses iterasi dihentikan, dan jika berbeda proses iterasi akan dilanjutkan.

4.4 Implementasi Antarmuka

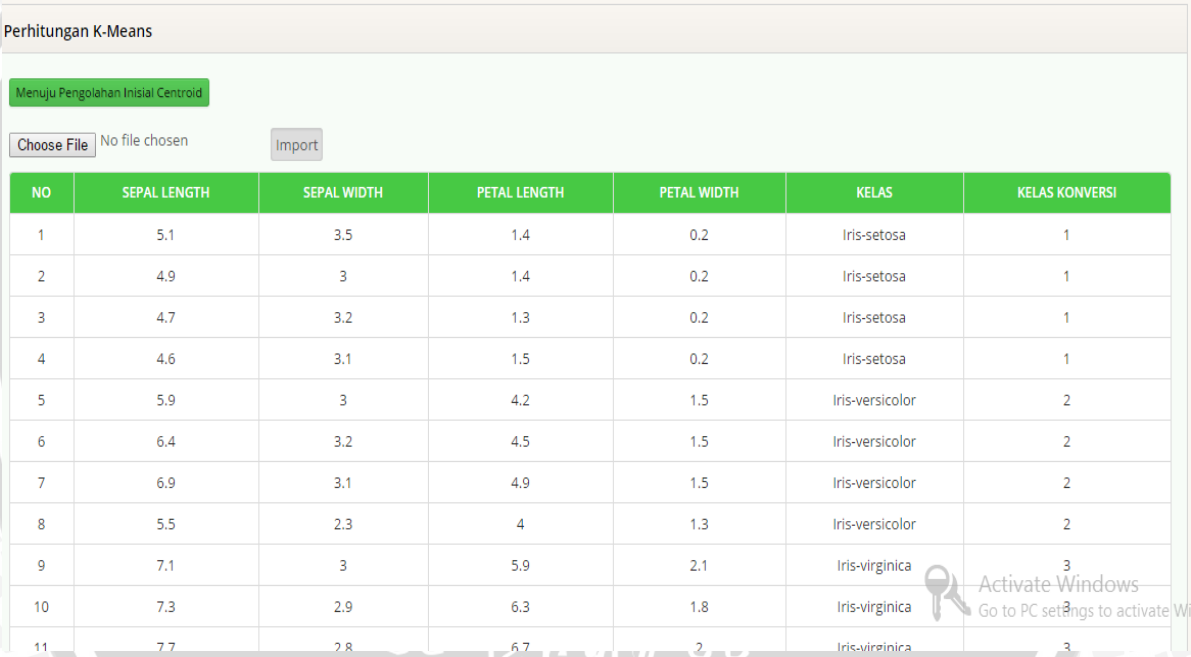
Pada sub bab implementasi antarmuka akan ditampilkan gambaran mengenai tahapan-tahapan berjalannya program Implementasi Metode *K-Means* dengan inisialisasi *centroid* menggunakan weighted average.

4.4.1 Tampilan Antarmuka Upload Dataset

Pada antarmuka dataset, di sini terdapat tombol untuk melakukan upload dataset berupa fila xls atau xlsx. Langkah pertama yang harus dilakukan adalah menekan tombol "*Choose File*" untuk memilih *file dataset* yang ingin digunakan. Lanhkah selanjutnya setelah memilih file yang digunakan adalah menekan tombol "*import*". Tampilan antarmuka upload dataset ditampilkan oleh Gambar 4.1. Apabila dataset telah di-*import* ke dalam sistem, selanjutnya halaman *upload dataset* akan dimuat ulang sehingga halaman *upload dataset* menampilkan dataset yang telah di-*import* sebelumnya. Tampilan antarmuka *upload dataset* setelah melakukan *import dataset* ditampilkan oleh Gambar 4.2.



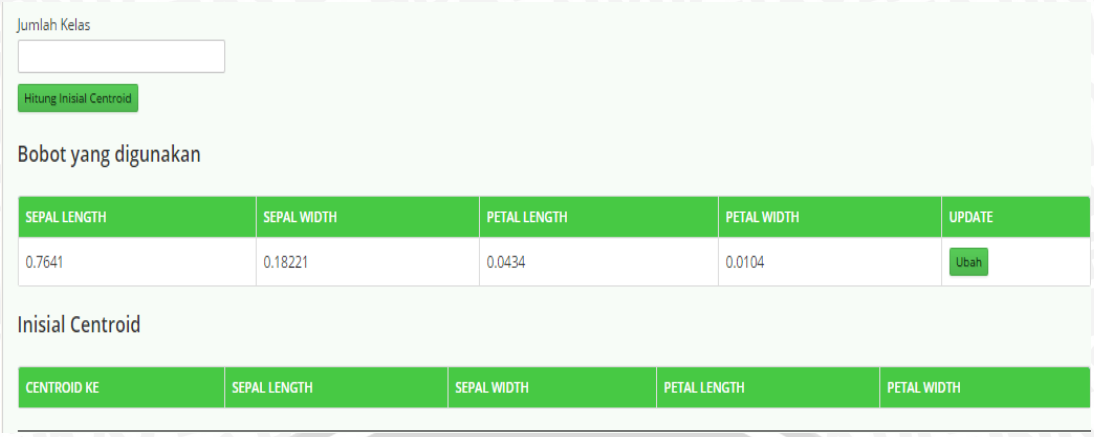
Gambar 4. 1 Antarmuka Upload Dataset



Gambar 4. 2 Antarmuka *Upload Dataset* Setelah *Import* Data

4.4.2 Tampilan Antarmuka Pengolahan Inisial Centroid

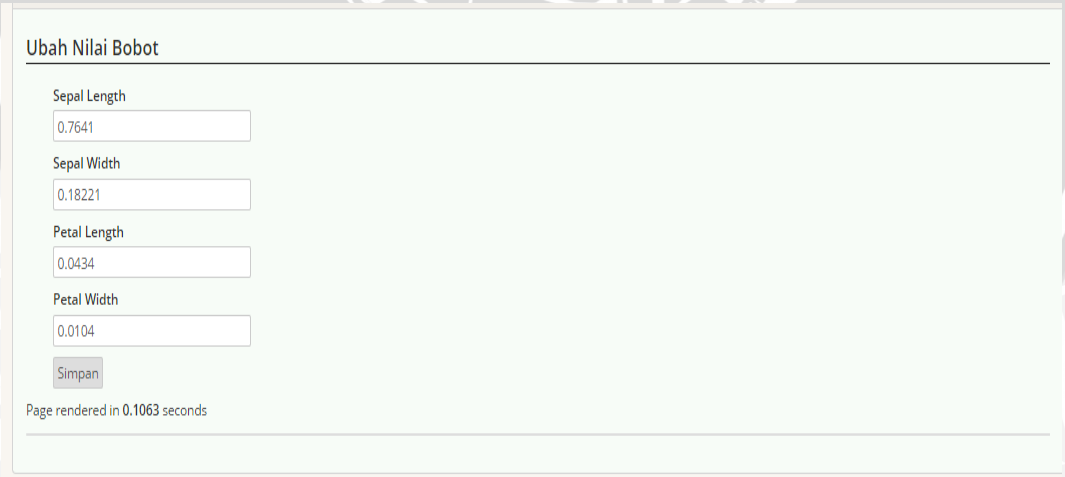
Setelah halaman selesai melakukan import databse, langkah selanjutnya adalah menekan tombol "Menuju Pengolahan Inisial *Centroid*". Pada halaman pengolahan inisial centroid terdapat dua tabel, yaitu tabel bobot dan tabel inisial *centroid*. Tampilan antarmukan halaman pengoahan inisial *centroid* disajikan pada Gambar 4.3.



Gambar 4. 3 Halaman Pengolahan Inisial *Centroid*

4.4.2.1 Tampilan Form Ubah Bobot yang Digunakan

Tabel bobot merupakan tabel yang menampilkan data berupa bobot *attribute* yang akan digunakan untuk pengolahan inisial centroid menggunakan weighted average. Bobot tersebut antara lain bobot untuk attribute *sepal length*, *sepal width*, *petal length*, dan *petal width*. Apabila ingin merubah nilai dari bobot yang akan digunakan dalam pengolahan inisial *centroid*, terdapat tombol "ubah" untuk mengarahkan pengguna menuju *form update* bobot. Untuk merubah nilai bobot, pengguna harus mengganti nilai yang sudah ada pada form tersebut menjadi nilai baru dan dilanjutan dengan menekan tombol simpan untuk merubah nilai bobot tersebut. Tampilan halaman *form update* bobot ditampilkan oleh Gambar 4.4.



Gambar 4. 4 Halaman *Form* Ubah Bobot

4.4.2.2 Tampilan Tabel Hasil Pengolahan Inisial Centroid

Pada tahap selanjutnya setelah melakukan perubahan nilai bobot, dapat dilakukan proses pengolahan inisial *centroid* yaitu dengan mengisi form "Jumlah Kelas" dan dilanjutkan dengan menekan tombol "Hitung Inisial Centroid". Selanjutnya sistem akan melakukan pengolahan inisial *centroid* dengan menggunakan *weighted average*, dan kemudian inisial *centroid* akan ditampilkan pada tabel inisial *centroid* pada halaman pengolahan inisial *centroid*. Halaman pengolahan inisial centroid yang menampilkan centroid yang akan digunakan oleh sistem disajikan pada Gambar 4.5.


Hitung Inisial Centroid

Bobot yang digunakan

SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	UPDATE
0.7641	0.18221	0.0434	0.0104	Ubah

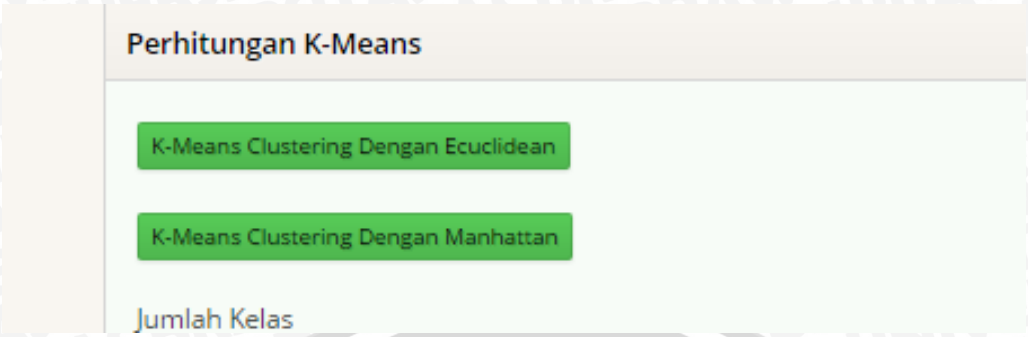
Inisial Centroid

CENTROID KE	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH
1	4.9	3	1.4	0.2
2	5.9	3	4.2	1.5
3	7.2	3.6	6.1	2.5

Gambar 4. 5 Pengolahan *Centroid* Menampilkan Hasil Inisial *Centroid*

4.4.3 Tampilan Antarmuka *Clustering* Data

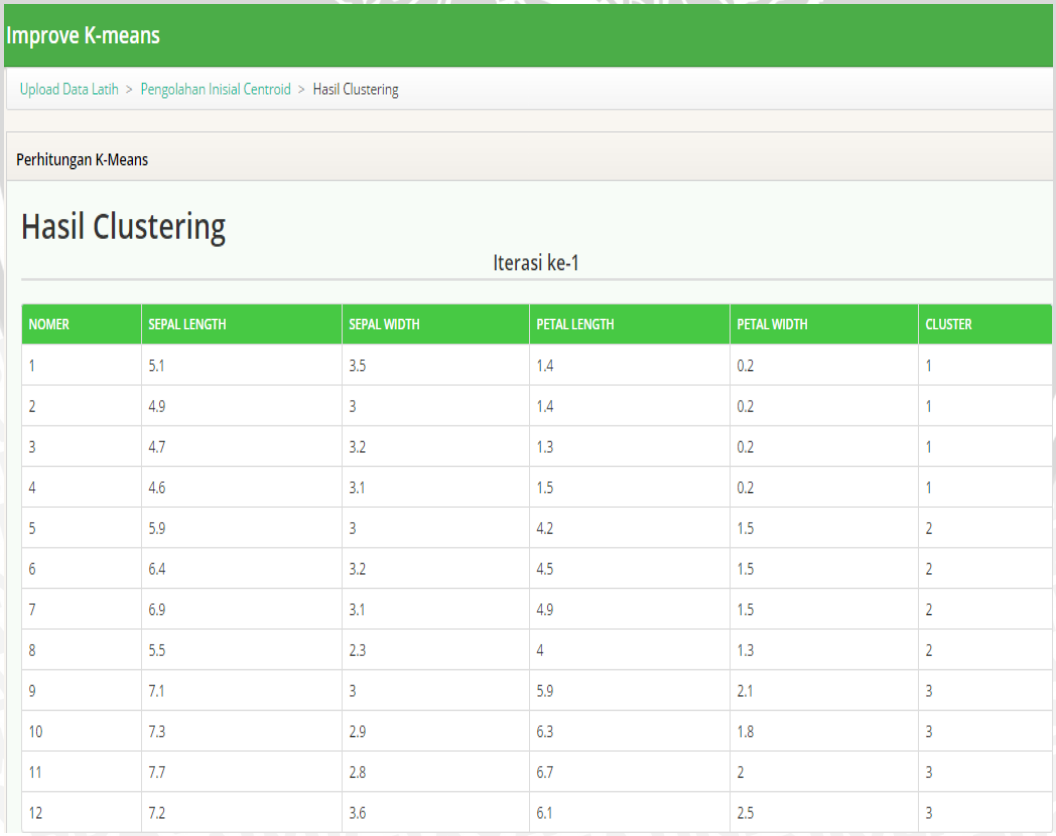
Pada halaman pengolahan inisial centroid terdapat dua tombol utama yang mengarahkan mengguna menuju proses *K-Means clustering*. Kedua tombol tersebut adalah tombol "*K-Means Clustering* Dengan *Euclidean*" untuk melakukan proses *K-Means clustering* dengan menggunakan perhitungan jarak *ecuclidean*, dan "*K-Means Clustering* Dengan *Manhattan*" untuk melakukan proses *K-Means clustering* dengan menggunakan perhitungan jarak *manhattan*. Tombol "*K-Means Clustering* Dengan *Euclidean*" dan tombol "*K-Means Clustering* Dengan *Manhattan*" pada halaman pengolahan inisial centroid ditampilkan pada Gambar 4.6.



Gambar 4. 6 Tampilan Antarmuka Tombol *Clustering* Data

4.4.3.1 Tampilan Antarmuka Hasil *Clustering* Data

Apabila tombol "*K-Means Cluster*ing Dengan *Euclidean*" atau "*K-Means Cluster*ing Dengan *Euclidean*" ditekan, sistem akan melakukan proses *K-Means clustering* data. Setelah proses *K-Means clustering* selesai, maka pengguna akan diarahkan menuju halaman hasil *cluster*ing untuk melihat catatan proses *cluster*ing, berupa hasil iterasi dan hasil centroid akhir *cluster*ing. Halaman antarmuka hasil *clustering* data disajikan pada Gambar 4.7.



NOMER	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLUSTER
1	5.1	3.5	1.4	0.2	1
2	4.9	3	1.4	0.2	1
3	4.7	3.2	1.3	0.2	1
4	4.6	3.1	1.5	0.2	1
5	5.9	3	4.2	1.5	2
6	6.4	3.2	4.5	1.5	2
7	6.9	3.1	4.9	1.5	2
8	5.5	2.3	4	1.3	2
9	7.1	3	5.9	2.1	3
10	7.3	2.9	6.3	1.8	3
11	7.7	2.8	6.7	2	3
12	7.2	3.6	6.1	2.5	3

Gambar 4. 7 Tampilan Antarmuka Halaman Hasil *Cluster*ing

4.4.3.1.1 Tampilan Antarmuka Hasil *Clustering* Bagian Hasil Iterasi

Pada halaman hasil *clustering* terdapat laporan berupa rekaman hasil *clustering* tiap iterasi. Pada rekaman ini, akan menampilkan hasil *clustering* setiap data pada setiap iterasi. Tampilan antarmuka hasil iterasi ditampilkan pada Gambar 4.8.

Hasil Clustering

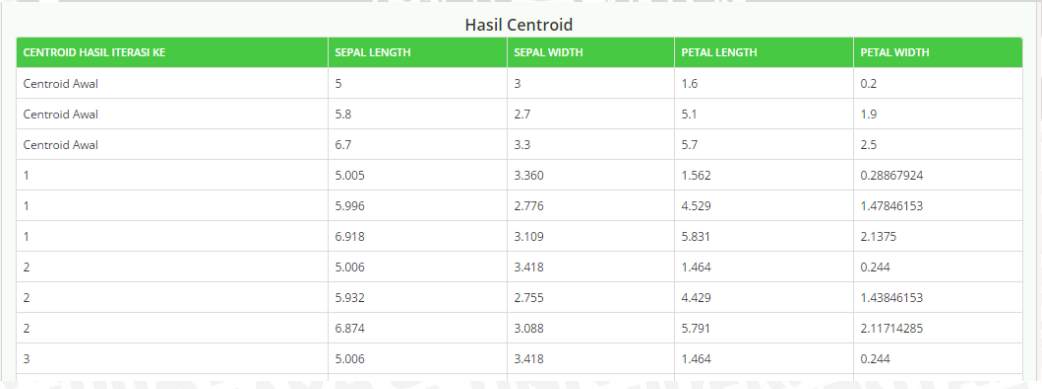
Iterasi ke-1

NOMER	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH	CLUSTER
1	5.1	3.5	1.4	0.2	1
2	4.9	3	1.4	0.2	1
3	4.7	3.2	1.3	0.2	1
4	4.6	3.1	1.5	0.2	1
5	5.9	3	4.2	1.5	2
6	6.4	3.2	4.5	1.5	2
7	6.9	3.1	4.9	1.5	2
8	5.5	2.3	4	1.3	2
9	7.1	3	5.9	2.1	3

Gambar 4. 8 Tampilan Antarmuka Halaman Hasil *Clustering* Bagian Iterasi.

4.4.3.1.2 Tampilan Antarmuka Hasil *Clustering* Bagian Hasil Centroid

Tampilan antarmuka halaman hasil *clustering* juga menampilkan hasil centroid yang diperoleh dari iterasi *K-Means clustering*. Setiap iterasi yang dilakukan centroid akan disimpan pada database. Pada halaman ini, setiap *centroid* akan ditampilkan.Tampilan antarmuka hasil centroid ditampilkan pada Gambar 4.9.



Hasil Centroid

CENTROID HASIL ITERASI KE	SEPAL LENGTH	SEPAL WIDTH	PETAL LENGTH	PETAL WIDTH
Centroid Awal	5	3	1.6	0.2
Centroid Awal	5.8	2.7	5.1	1.9
Centroid Awal	6.7	3.3	5.7	2.5
1	5.005	3.360	1.562	0.28867924
1	5.996	2.776	4.529	1.47846153
1	6.918	3.109	5.831	2.1375
2	5.006	3.418	1.464	0.244
2	5.932	2.755	4.429	1.43846153
2	6.874	3.088	5.791	2.11714285
3	5.006	3.418	1.464	0.244

Gambar 4. 9 Tampilan hasil *clustering* bagian hasil *centroid*.

BAB V
PENGUJIAN DAN ANALISIS

Pengujian yang dilakukan dalam bab ini meliputi pengujian nilai bobot untuk menentukan bobot terbaik bagi *improve K-Means* yang inisial *centroid*-nya ditentukan dengan *weighted average*, dan dilanjutkan dengan pengujian perbandingan antara *improve K-Means* yang inisial *centroid*-nya ditentukan dengan *weighted average* dan *K-Means* konvensional menggunakan pengujian kualitas *cluster* dengan menggunakan metode *silhouette coefficient*, serta pengujian tingkat akurasi *cluster*.

5.1 Pengujian Nilai Bobot Improve K-Means Dengan Weighted Average

Pada tahap pengujian bobot, nilai bobot diperoleh dari penelitian yang dilakukan oleh Ching-Huse Cheng yang berjudul "*OWA Based Information Fusion Technique For Classification Problem*". Pengujian bobot dilakukan guna mengetahui adakah pengaruh dari pembobotan yang dilakukan dengan OWA atau *ordered weighted averaging* terhadap hasil *clustering* yang diperoleh. Nilai bobot yang digunakan pada pengujian bobot disajikan pa da Tabel 5.1 [CHE-07].

Tabel 5. 1 Tabel Bobot *Attribute Dataset* Iris

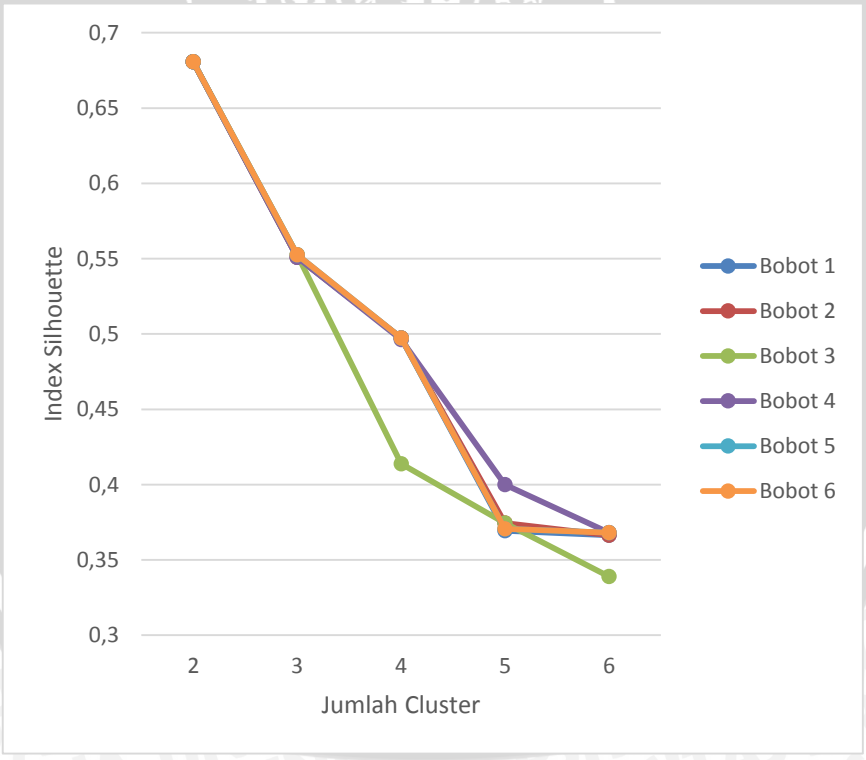
Nomor Bobot	*Sepal Length*	*Sepal Width*	*Petal Length*	*Petal Width*
1	0.25	0.25	0.25	0.25
2	0.3475	0.2722	0.2133	0.1671
3	0.4609	0.2754	0.1646	0.0987
4	0.5695	0.2521	0.1065	0.045
5	0.7641	1.822	0.0434	0.0104
6	1	0	0	0

Tahapan pengujian nilai bobot dilakukan dengan mengamati hasil *clustering* 150 data dari *dataset* iris pada jumlah *cluster* yang berbeda, yaitu 2, 3, 4, 5, dan 6. Hasil pengujian bobot disajikan pada Tabel 5.2.

Tabel 5. 2 Tabel hasil pengujian bobot

Jumlah *cluster*	Bobot 1	Bobot 2	Bobot 3	Bobot 4	Bobot 5	Bobot 6
2	0,68081	0,68081	0,68081	0,68081	0,68081	0,68081
3	0,55096	0,55096	0,55259	0,55096	0,55259	0,55259
4	0,49723	0,49723	0,41382	0,49629	0,49723	0,49723
5	0,36935	0,37444	0,37444	0,39994	0,37059	0,37059
6	0,3665	0,3665	0,33909	0,36821	0,36821	0,36821

Pada Gambar 5.1 menunjukkan grafik perbandingan hasil pengujian *silhouette coefficient* terhadap jumlah data dan jumlah *cluster* untuk masing-masing bobot.



Gambar 5. 1 Hasil *Silhouette Coefficient* Masing-Masing Bobot

Berdasarkan grafik perbandingan hasil penggunaan nilai bobot terhadap hasil *silhouette coefficient* dapat diamati jika nilai nilai bobot berpengaruh pada saat nilai *cluster* lebih dari dua. Hal ini menunjukkan jika nilai hasil clustering *Improvemt K-Means* dengan inisialisasi *centroid* menggunakan *weighted average* dipengaruhi oleh jumlah *cluster* dan bobot *attribute*, meskipun perbedaan hasil *silhouette coefficient* antar bobot tidak terlalu besar. Berdasarkan grafik perbandingan *silhouette coefficient* Gambar 5.1, bobot yang dapat menghasilkan kualitas terbaik adalah bobot ke empat.

5.2 Pengujian *Silhouette Coefficient*

Pengujian *silhouette coefficient* dilakukan untuk melakukan perbandingan kualitas *clustering* antara metode improve *K-Means* dengan inisiali *centroid* menggunakan *weighted average*, dengan metode *K-Means* konvensional. kualitas hasil *clustering* dihitung dengan menggunakan *silhouette coefficient* yang memiliki rentang -1 hingga 1.

Pada pengujian *silhouette coefficient* yang akan dilakukan dengan menggunakan dua metode perhitungan jarak yang berbeda, yaitu *euclidean* dan *manhattan*. Jumlah data yang digunakan dalam pengujian adalah 75, 105, 120, dan 150 data dari *dataset* iris, sedangkan jumlah *cluster* yang digunakan adalah 2, 3, 4, 5, dan 6. Pada pengujian *silhouette coefficient* pada metode *K-Means* konvensional, akan dilakukan pengujian sebanyak sepuluh kali percobaan yang kemudian akan dilakukan perhitungan rata-rata yang kemudian dijadikan sebagai hasil metode. Hasil perbandingan pengujian *silhouette coefficient* ditampilkan pada Tabel 5.3.

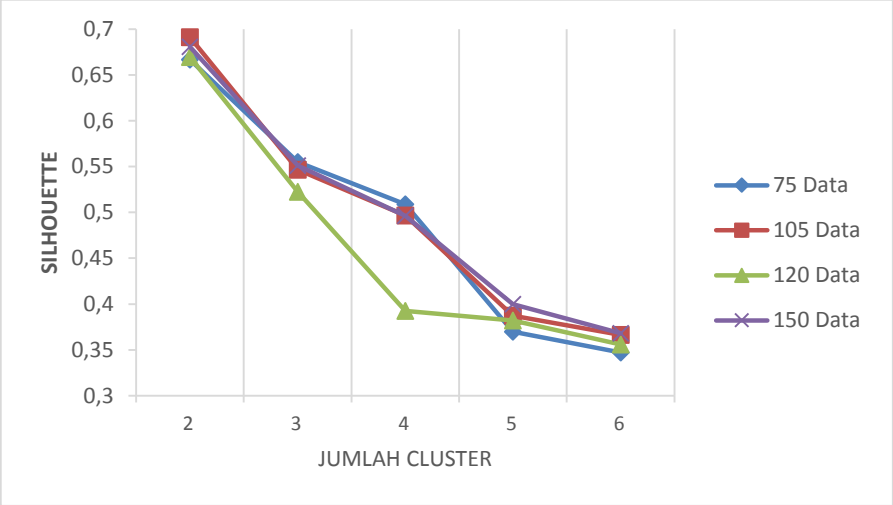
Tabel 5. 3 Hasil Pengujian *Silhouette Coefficient*

Jumlah		Improved K-Means		K-Means Konvensional	
Data	*Cluster*	Euclidean	Manhattan	Euclidean	Manhattan
75	2	0,6669	0,6669	0,6581	0,6669
	3	0,55493	0,55661	0,55541	0,55661
	4	0,50889	0,50777	0,47499	0,42916

Tabel 5.3 Hasil Pengujian *Silhouette Coefficient* (lanjutan)

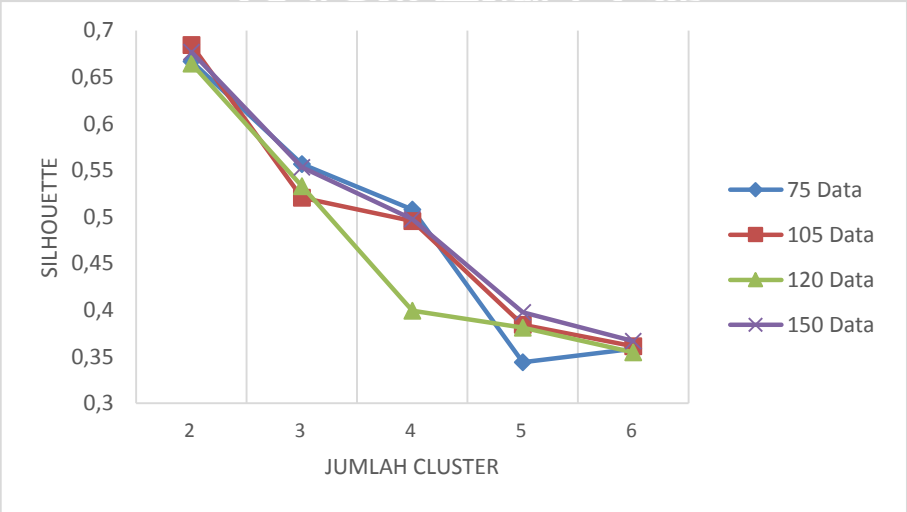
Jumlah		Improved K-Means		K-Means Konvensional	
Data	*Cluster*	Euclidean	Manhattan	Euclidean	Manhattan
75	5	0,36969	0,34388	0,40233	0,37918
	6	0,34698	0,35845	0,41377	0,36727
105	2	0,69131	0,68444	0,69131	0,68444
	3	0,54643	0,52071	0,53114	0,53303
	4	0,49664	0,49548	0,44889	0,45464
	5	0,38736	0,38402	0,4193	0,43227
	6	0,36651	0,36103	0,40617	0,36041
120	2	0,66956	0,66439	0,67032	0,65096
	3	0,52238	0,53292	0,52182	0,51621
	4	0,39246	0,39925	0,45524	0,46797
	5	0,38193	0,38106	0,42191	0,40983
	6	0,35579	0,3544	0,40519	0,38636
150	2	0,68081	0,67703	0,68081	0,67703
	3	0,55096	0,55312	0,54493	0,5453
	4	0,49629	0,4974	0,46954	0,44881
	5	0,39994	0,39749	0,40728	0,41869
	6	0,36821	0,36662	0,37748	0,38204

Pada Gambar 5.2 ditampilkan hasil perbandingan *silhouette coefficient* pada *improved K-Means* dengan menggunakan *euclidean* terhadap jumlah data dan jumlah *cluster*.



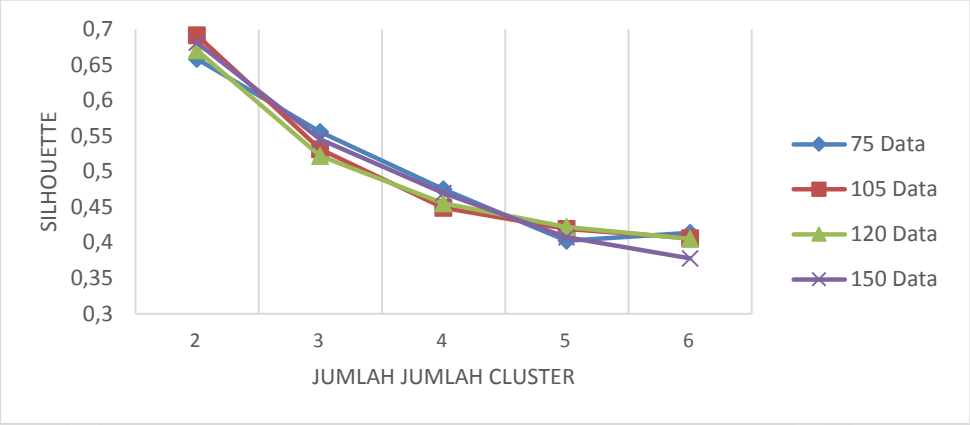
Gambar 5. 2 Hasil *Silhouette Improved K-Means* Dengan *Euclidean*

Pada Gambar 5.3 akan ditunjukkan hasil perbandingan *silhouette coefficient* pada *improved K-Means* dengan menggunakan *manhattan* terhadap jumlah data dan jumlah *cluster*.



Gambar 5. 3 Hasil *Silhouette Improved K-Means* Dengan *Manhattan*

Pada Gambar 5.4 akan ditunjukkan hasil perbandingan *silhouette coefficient* pada *K-Means* konvensional dengan menggunakan *euclidean* terhadap jumlah data dan jumlah *cluster*.



Gambar 5. 4 Hasil *Silhouette K-Means* Konvensional Dengan *Euclidean*

Pada Gambar 5.5 akan ditunjukkan hasil perbandingan *silhouette cosfficient* pada *K-Means* konvensional dengan menggunakan *manhattan* terhadap jumlah data dan jumlah *cluster*.

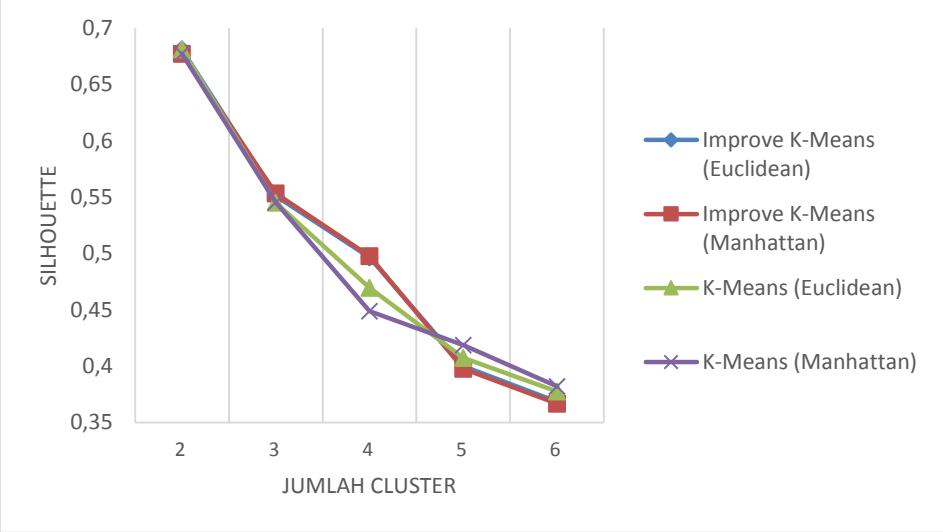
Gambar 5. 5 Hasil *Silhouette K-Means* Konvensional Dengan *Manhattan*

Berdasarkan grafik Gambar 5.2, Gambar 5.3, Gambar 5.4, dan Gambar 5.5 diketahui hasil *cluster*ing dari 150 data memiliki *silhouette coefficient* yang lebih dominan daripada hasil *clustering* dari 75, 105, dan 120 data iris saat diujikan pada jumlah *cluster* 2, 3, 4, 6. Pada Tabel 5.4 ditampilkan hasil perbandingan *clustering improvement K-Means* dengan inisialisasi *centroid* menggunakan *weighted average* dan *K-Means* konvensional dengan 150 data terhadap jumlah *cluster*.

Tabel 5. 4 Perbandingan *Silhouette Improve K-Means* dan *K-Means* Konvensional Dengan 150 Data.

Jumlah *Cluster*	Improve K-Means		K-Means Konvensional	
	Euclidean	Manhattan	Euclidean	Manhattan
2	0,68081	0,67703	0,68081	0,67703
3	0,55096	0,55312	0,54493	0,5453
4	0,49629	0,4974	0,46954	0,44881
5	0,39994	0,39749	0,40728	0,41869
6	0,36821	0,36662	0,37748	0,38204
Rata-rata	0,49924	0,49833	0,49601	0,49437

Pada Gambar 5.6 akan ditampilkan perbandingan *silhouette coefficient improve K-Means* dengan inisiali *centroid* menggunakan *weighted average*, dengan metode *K-Means* konvensional.



Gambar 5. 6 Hasil *Silhouette Improve K-Means* dan *K-Means* 150 Data.

Pada grafik Gambar 5.6 dapat diketahui jika metode *Improve K-Means* dan *K-Means* konvensional memiliki perbandingan *silhouette coefficient* yang tidak jauh berbeda. Pada pengujian jumlah *cluster* bernilai dua dan empat *Improve K-*

Means memiliki nilai *silhouette coefficient* yang lebih tinggi dibandingkan dengan *K-Means* konvensional. Pada pengujian jumlah *cluster* bernilai lima dan enam, *K-Means* konvensional memiliki nilai *silhouette coefficient* yang lebih tinggi dibandingkan dengan *improve K-Means*. Namun setelah dilakukan perhitungan rata-rata, *improve K-Means* memiliki rata-rata *silhouette coefficient* yang lebih baik daripada *K-Means* konvensional. Pada grafik Gambar 5.6 juga diketahui jika semakin banyak *cluster*, nilai *silhouette coefficient* semakin menurun. Hal ini menunjukkan jika tingkat kemiripan di dalam suatu *cluster* semakin rendah dan kemiripan dengan *cluster* lain semakin tinggi.

Berdasarkan pengujian di atas dapat diketahui jika pengujian dari *improve K-Means* dengan *weighted average* memiliki kualitas *clustering* yang lebih tinggi dibanding dengan *K-Means* konvensional, serta kualitas *clustering* pada *improve K-Means* dengan *weighted average* dan *K-Means* konvensional menggunaan *euclidean* menghasilkan kualitas *clustering* yang lebih baik, daripada dengan menggunakan *manhattan*.

5.3 Pengujian Akurasi

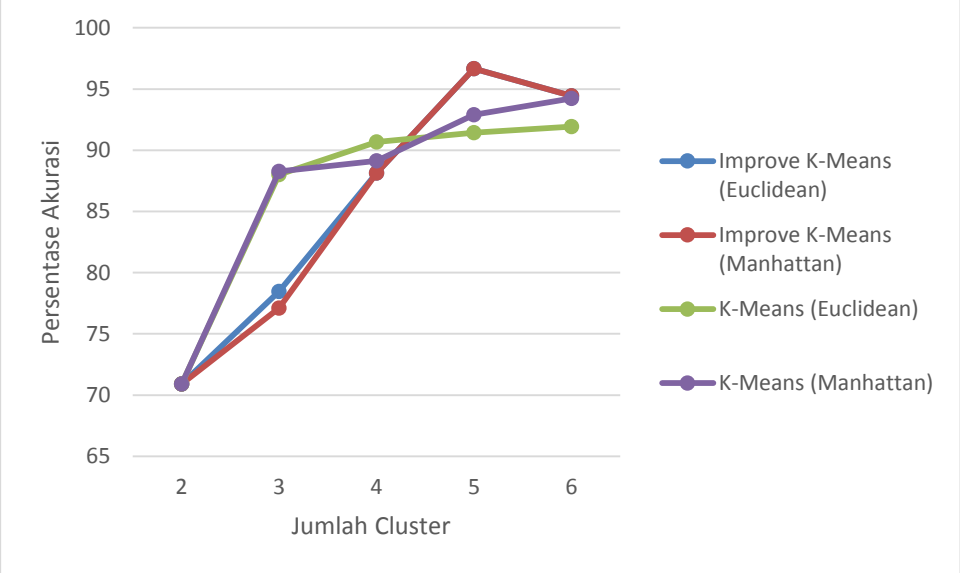
Pengujian akurasi dilakukan untuk mengetaui seberapa tinggi akurasi yang didapatkan oleh metode *improvement K-Means* dengan inisialisasi *centroid* menggunakan *weighted average*. Pada pengeujian ini dilakukan perbandingan akurasi antara *improvement K-Means* dengan inisialisasi *cetroid* menggunakan *weighted average*.

Pengujian akurasi dilakukan dengan menggunakan 105 data latih dan 45 data uji dari dataset iris dengan menggunakan jumlah *cluster* yang berbeda. Sama seperti pengujian *silhouette coefficient*, pengujian akan dilakukan dengan mengguakan dua metode perhitungan jarak, yaitu *euclidean* dan *manhattan*. Pada metode *K-Means* konvensional akan dilakukan sebanyak 10 kali run dan kemudian akan dihitung rata-ratanya yang kemudian dijadikan sebagai hasil metode. Hasil perbandingan pengujian akurasi ditampilkan pada Tabel 5.5.

Tabel 5. 5 Hasil Pengujian Akurasi

Jumlah *Cluster*	Improve K-Means		K-Means Konvensional	
	Euclidean (%)	Manhattan (%)	Euclidean (%)	Manhattan (%)
2	70,9034	70,9034	70,9034	70,9034
3	78,4632	77,1164	88,0024	88,2738
4	88,1313	88,1313	90,6811	89,1421
5	96,6667	96,6667	91,4304	92,9063
6	94,4444	94,4444	91,944	94,2242
Rata-rata	85,7218	85,4524	86,5923	87,09

Pada Gambar 5.7 grafik perbandingan hasil pengujian akurasi yang dihasilkan antara improve K-Means dan K-Means konvensional.



Gambar 5. 7 Grafik Hasil Pengujian Akurasi

Berdasarkan grafik perbandingan hasil pengujian akurasi pada pada Gambar 5.7. Perbedaan akurasi terjadi disaat jumlah *cluster* lebih dari 2, di mana pada saat *cluster* bernilai dua dan tiga akurasi *K-Means* konvensional memiliki nilai lebih

tinggi daripada *Improvemt K-Means* dengan inisialisasi *centroid* menggunakan *weighted average*. Sedangkan pada *cluster* bernilai lima dan enam akurasi *Improvemt K-Means* dengan inisialisasi *centroid* menggunakan *weighted average* berada di atas K-Means konvensional. Saat dilakukan perhitungan rata-rata, didapati jika *K-Means* konvensional memiliki nilai rata-rata akurasi yang lebih baik dibandingkan *Improvemt K-Means* dengan inisialisasi *centroid* menggunakan *weighted average*.

Pada pengujian grafik Gambar 5.7, nilai akurasi dari *Improvemt K-Means* dengan inisialisasi *centroid* menggunakan *weighted average* mengalami penurunan nilai akurasi saat nilai cluster bernilai enam, berbeda dengan K-Means konvensional yang memiliki akurasi yang terus meningkat jika nilai *cluster* semakin tinggi. Hal ini terjadi dikarenakan nilai inisial *centroid* yang dihasilkan oleh *weighted average* bernilai tetap sehingga hasil akurasi juga tidak berubah-ubah, berbeda dengan *K-Means* konvensional yang inisial *centroid*-nya bernilai *random* sehingga dapat memperoleh hasil akurasi yang lebih tinggi atau lebih rendah.

BAB VI
PENUTUP

6.1 Kesimpulan

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

1. *Weighted average* menentukan inisial centroid dengan mencari *datapoint* yang memiliki nilai weighted average terdekat dengan rata-rata nilai weighted average atau di(avg) suatu *subset* atau *cluster* sementara.
2. Penggunaan bobot *Ordered Weighted Averaging* pada metode *weighted average* memiliki pengaruh terhadap hasil *clustering*, sehingga dengan menggunakan bobot yang berbeda dapat menghasilkan hasil *clustering* yang berbeda pula. Pada hasil penelitian yang dilakukan, dan bobot yang menghasilkan kualitas *clustering* terbaik diperoleh dengan menggunakan bobot ke empat, yang bernilai 0.5695 untuk attribute sepal length, 0.2521 untuk sepal width, 0.1065 untuk sepal length, dan 0.045 untuk petal length.
3. *Improvement K-Means* dengan inisialisasi *centroid* menggunakan *weighted average* menghasilkan kualitas *cluster*ing yang lebih tinggi dibandingkan dengan *K-Means* konvensional, namun hasil *clustering improvement K-Means* dengan inisialisasi *centroid* menggunakan *weighted average* belum tentu menghasilkan *clustering* yang optimum dari suatu dataset, hal ini dibuktikan pada proses *clustering K-Means* konvensional dapat menghasilkan kualitas *clustering* yang lebih baik pada saat tertentu.
4. Pada pengujian kualitas *cluster*ing yang dilakukan, penggunaan perhitungan jarak menggunakan *euclidean* memperoleh hasil yang lebih baik daripada menggunakan *manhattan*, baik saat diterapkan dengan menggunakan *Improvement K-Means* dengan inisialisasi *centroid* menggunakan *weighted average* maupun dengan *K-Means* konvensional.

5. Pada pengujian akurasi, *K-Means* konvensional memiliki tingkat akurasi yang lebih tinggi jika dibanding dengan *Improvement K-Means* dengan inisialisasi *centroid* menggunakan *weighted average*.
6. Pada pengujian akurasi, saat menggunakan *euclidean improvement K-Means* dengan inisialisasi *centroid* menggunakan *weighted average* memiliki akurasi yang lebih tinggi dibanding manhattan, namun berbeda dengan *K-Means* konvensional yang memiliki akurasi tertinggi saat menggunakan *manhattan* dibanding dengan *euclidean*.

6.2 Saran

Saran yang diberikan untuk pengembangan berikutnya antara lain :

1. Dalam menggunakan *weighted average* untuk menentukan inisial *centroid* sangat bergantung pada nilai bobot yang dimasukkan pengguna, mengingat pada penelitian ini kualitas *clustering* yang didapat belum mencapai nilai optimum, sehinga diperlukan metode lain dalam menentukan bobot dari suatu dataset.

DAFTAR PUSTAKA

[BEN-14] Ben-David, Shalev., Lev Reyzin., *Data stability in clustering: A closer look*, Theoretical Computer Science, 2014.

[CGP-12] Chen. Guang-ping, and Wang Wen-peng. "*An Improved K-Means Algorithm with Meliorated Initial Center*" in ICCSE, Australia, Melbourne, 2012, pp. 150-153.

[CHE-07] Cheng, Ching-Huse., et al., *OWA Based Information Fusion Techniques For Classification Problem*, Department of Information Management, National Yunlin University of Scienceand Technology, 123, section 3, University Road, Touliu, Yunlin 640, Taiwan, R.O.C., 2007.

[CHU-03] Cheung, Yiu-Ming., *k*-Means: A new generalized k-means clustering algorithm*, Department of Computer Science, Hong Kong Baptist University, Hong Kong, 2003.

[FAH-06] Fahim A.M, et al., *An efficient enhanced k-means clustering algorithm*, J Zhejiang Univ SCIENCE A, 2006.

[HAN-06] Han, Jiawei., Michelin Kamber., *Data Mining Concepts and Technique, Morgan Kaufman Publishers*, San Diego, 2006.

[HER-05] Heru, Darlis., Nanik Suciati., Daru Jani Nanjaya., *CLUSTERING* DATA *NON-NUMERIK* DENGAN PENDEKATAN ALGORITMA *K-MEANS* DAN *HAMMING DISTANCE* STUDI KASUS BIRO JODOH, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, 2005.

[KAN-11] Kantardzic, Mehmed., *Data Mining: Concepts, Models, Methods, and Algorithms, Second Edition*, John Wiley & Sons, 2011.

[KHA-04] Khan, Shehroz S., Amir Ahmad., *Cluster center Iitialization algorithm for K-means clustering*, Scientific Analysis Group, DRDO, Metcalfe House & Solid State Physics Laboratory, DRDO, Probyn Road, Delhi, India, 2014.

[LAR-05] Larose, T Danield., *DATA MINING METHODS AND MODELS*, Department of Mathematical Sciences, Central Connecticut State Universit, 2005.

[MAH-12] Mahmud, Sohrab., et al., *Improvement of K-means Clustering algorithm with better initial centroids based on weighted average*, Department of Computer Science & Engineering Dhaka University of Engineering & Technology, Gazipur Bangladesh, 2012.

[PJR-87] Rousseeuw, Peter J., *Silhouette : a graphical aid to the interpretation and validation of cluster analysis*, University of Fribourg, 1987.

[PUR-12] Purba, Ronsen., *DATA MINING* : MASA LALU, SEKARANG DAN MASA MENDATANG, STMIK Mikroskil, Medan, 2012.

[SIN-11] Sing, Ray Vinjay., M.P.S Bhatia., *Data Clustering With Modified K-Means Algorithm*, Department of Computer Science and Engineering, Netaji Subhash Institute of Technology, University Of Delhi, New Delhi, India, 2011.

[STE-00] Steinbach, Michael., George Karypis., Vipin Kumar., *A Comparison of Document Clustering Techniques*, Department of Computer Science and Egineering, University of Minnesota, 2000.

[TAN-11] Tan, S Chuan., et al., A general stochastic clustering method for automatic cluster discovery, ScienceDirect, 2011.

[TUR-05] Turban, E., dkk, 2005. *Decicion Support Systems and Intelligent Systems*. Andi Offset.

[YED-10] Yedla, Madhu., Srivinasa Rao Pathakota., T M Srivinasa., *Enghancing K-Means Clustering Algorithm With Improved Initial Centroid*, International Journal of Computer Science and Information Technologies, 2010.

[YUA-04] Yuan, Fang., et al., *A New Algorithm To Get The Initial Centroid*, Proc. of the 3rd International Conference On Machine Learning and Cybernetics, Sanghai, 2004.

LAMPIRAN 1. Hasil *Silhouete Coefficient K-Means* Konvensional

1. Pengujian jumlah data 75 dengan jumlah cluster bernilai 2.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,672	83,62	0,667	83,35
2	0,667	83,35	0,667	83,35
3	0,667	83,35	0,667	83,35
4	0,557	77,85	0,667	83,35
5	0,672	83,62	0,667	83,35
6	0,672	83,62	0,667	83,35
7	0,667	83,35	0,667	83,35
8	0,667	83,35	0,667	83,35
9	0,672	83,62	0,667	83,35
10	0,667	83,35	0,667	83,35
Rata-rata	0,658	82,9	0,667	83,35

2. Pengujian jumlah data 75 dengan jumlah cluster bernilai 3.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,557	77,85	0,557	77,83
2	0,555	77,75	0,557	77,83
3	0,555	77,75	0,557	77,83
4	0,557	77,85	0,557	77,83
5	0,555	77,75	0,557	77,83
6	0,555	77,75	0,557	77,83
7	0,555	77,75	0,557	77,83
8	0,555	77,75	0,557	77,83
9	0,555	77,75	0,557	77,83
10	0,555	77,75	0,557	77,83
Rata-rata	0,555	77,77	0,557	77,83

3. Pengujian jumlah data 75 dengan jumlah cluster bernilai 4.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,509	75,44	0,347	67,37
2	0,512	75,61	0,508	75,39
3	0,509	75,44	0,508	75,39
4	0,512	75,61	0,413	70,64
5	0,509	75,44	0,509	75,44
6	0,411	70,57	0,506	75,31
7	0,509	75,44	0,347	67,37
8	0,408	70,39	0,384	69,22
9	0,512	75,61	0,384	69,22
10	0,359	67,93	0,384	69,22
Rata-rata	0,475	73,75	0,429	71,46

4. Pengujian jumlah data 75 dengan jumlah cluster bernilai 5.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,37	68,48	0,321	66,07
2	0,434	71,71	0,339	66,97
3	0,377	68,86	0,346	67,3
4	0,496	74,8	0,376	68,82
5	0,322	66,09	0,309	65,44
6	0,491	74,53	0,442	72,12
7	0,365	68,23	0,423	71,15
8	0,375	68,73	0,447	72,34
9	0,361	68,07	0,479	73,94
10	0,433	71,66	0,309	65,44
Rata-rata	0,402	70,12	0,379	68,96

5. Pengujian jumlah data 75 dengan jumlah cluster bernilai 6.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,356	67,78	0,33	66,5
2	0,48	73,99	0,351	67,56

5. Pengujian jumlah data 75 dengan jumlah cluster bernilai 6 (lanjutan)

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
3	0,493	74,65	0,299	64,97
4	0,345	67,24	0,44	72
5	0,442	72,12	0,309	65,44
6	0,352	67,62	0,322	66,11
7	0,475	73,75	0,329	66,45
8	0,342	67,08	0,501	75,07
9	0,339	66,93	0,495	74,75
10	0,515	75,73	0,296	64,78
Rata-rata	0,414	70,69	0,367	68,36

6. Pengujian jumlah data 105 dengan jumlah cluster bernilai 2.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,691	84,57	0,684	84,22
2	0,691	84,57	0,684	84,22
3	0,691	84,57	0,684	84,22
4	0,691	84,57	0,684	84,22
5	0,691	84,57	0,684	84,22
6	0,691	84,57	0,684	84,22
7	0,691	84,57	0,684	84,22
8	0,691	84,57	0,684	84,22
9	0,691	84,57	0,684	84,22
10	0,691	84,57	0,684	84,22
Rata-rata	0,691	84,57	0,684	84,22

7. Pengujian jumlah data 105 dengan jumlah cluster bernilai 3.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,539	76,94	0,54	76,99
2	0,522	76,09	0,521	76,06
3	0,527	76,35	0,54	76,99
4	0,546	77,32	0,54	76,99

7. Pengujian jumlah data 105 dengan jumlah cluster bernilai 3. (lanjutan)

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
5	0,539	76,94	0,524	76,2
6	0,527	76,35	0,522	76,09
7	0,527	76,35	0,54	76,99
8	0,527	76,35	0,54	76,99
9	0,518	75,91	0,54	76,99
10	0,54	76,99	0,524	76,2
Rata-rata	0,531	76,56	0,533	76,65

8. Pengujian jumlah data 105 dengan jumlah cluster bernilai 4.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,395	69,76	0,408	70,42
2	0,412	70,62	0,4	70
3	0,495	74,74	0,496	74,79
4	0,488	74,42	0,4	70
5	0,404	70,18	0,488	74,38
6	0,495	74,74	0,389	69,45
7	0,489	74,46	0,496	74,79
8	0,406	70,29	0,484	74,18
9	0,408	70,39	0,49	74,5
10	0,497	74,83	0,496	74,79
Rata-rata	0,449	72,44	0,455	72,73

9. Pengujian jumlah data 105 dengan jumlah cluster bernilai 5.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,366	68,3	0,372	68,61
2	0,373	68,66	0,355	67,76
3	0,484	74,18	0,449	72,46
4	0,456	72,8	0,375	68,76
5	0,37	68,49	0,449	72,46
6	0,368	68,4	0,476	73,79

9. Pengujian jumlah data 105 dengan jumlah cluster bernilai 5. (lanjutan)

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
7	0,378	68,92	0,447	72,34
8	0,484	74,18	0,432	71,58
9	0,484	74,18	0,484	74,18
10	0,431	71,53	0,484	74,18
Rata-rata	0,419	70,96	0,432	71,61

10. Pengujian jumlah data 105 dengan jumlah cluster bernilai 6.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,341	67,05	0,342	67,12
2	0,426	71,31	0,367	68,33
3	0,441	72,04	0,334	66,68
4	0,33	66,5	0,326	66,31
5	0,367	68,33	0,444	72,2
6	0,487	74,33	0,353	67,64
7	0,487	74,33	0,342	67,11
8	0,485	74,26	0,342	67,11
9	0,372	68,59	0,343	67,17
10	0,327	66,33	0,411	70,55
Rata-rata	0,406	70,31	0,36	68,02

11. Pengujian jumlah data 120 dengan jumlah cluster bernilai 2.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,67	83,48	0,664	83,2
2	0,67	83,48	0,664	83,2
3	0,677	83,86	0,664	83,2
4	0,67	83,48	0,664	83,2
5	0,67	83,48	0,664	83,2
6	0,67	83,48	0,664	83,2
7	0,67	83,48	0,664	83,2
8	0,67	83,48	0,664	83,2

11. Pengujian jumlah data 120 dengan jumlah cluster bernilai 2. (lanjutan)

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
9	0,67	83,48	0,53	76,5
10	0,67	83,48	0,664	83,2
Rata-rata	0,67	83,52	0,651	82,5

12. Pengujian jumlah data 120 dengan jumlah cluster bernilai 3.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,532	76,6	0,53	76,5
2	0,532	76,6	0,533	76,6
3	0,532	76,6	0,533	76,6
4	0,509	75,46	0,53	76,5
5	0,53	76,49	0,484	74,2
6	0,522	76,12	0,502	75,1
7	0,522	76,12	0,533	76,6
8	0,522	76,12	0,53	76,5
9	0,522	76,12	0,502	75,1
10	0,494	74,69	0,484	74,2
Rata-rata	0,522	76,09	0,516	75,8

13. Pengujian jumlah data 120 dengan jumlah cluster bernilai 4.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,496	74,82	0,495	74,8
2	0,408	70,39	0,493	74,6
3	0,397	69,86	0,493	74,6
4	0,412	70,62	0,495	74,8
5	0,498	74,89	0,41	70,5
6	0,498	74,89	0,495	74,8
7	0,498	74,89	0,405	70,3
8	0,392	69,62	0,495	74,8
9	0.412	70,62	0,405	70,3
10	0,498	74,89	0,493	74,6

13. Pengujian jumlah data 120 dengan jumlah cluster bernilai 4. (lanjutan)

9	0.412	70,62	0,405	70,3
10	0,498	74,89	0,493	74,6
Rata-rata	0,455	72,55	0,468	73,4

14. Pengujian jumlah data 120 dengan jumlah cluster bernilai 5.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,476	73,8	0,479	73,9
2	0,382	69,12	0,379	69
3	0,468	73,42	0,428	71,4
4	0,422	71,12	0,472	73,6
5	0,377	68,87	0,454	72,7
6	0,481	74,03	0,374	68,7
7	0,371	68,57	0,381	69,1
8	0,382	69,12	0,381	69,1
9	0,378	68,89	0,374	68,7
10	0,481	74,03	0,376	68,8
Rata-rata	0,422	71,1	0,41	70,5

15. Pengujian jumlah data 120 dengan jumlah cluster bernilai 6.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,421	71,05	0,476	73,8
2	0,345	67,27	0,368	68,4
3	0,481	74,03	0,36	68
4	0,481	71,86	0,36	68
5	0,329	66,43	0,337	66,8
6	0,348	67,39	0,365	68,3
7	0,481	74,04	0,334	66,7
8	0,484	74,19	0,334	66,7
9	0,363	68,17	0,474	73,7
10	0,32	65,99	0,455	72,7
Rata-rata	0,405	70,04	0,386	69,3

16. Pengujian jumlah data 150 dengan jumlah cluster bernilai 2.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,681	84,04	0,677	83,852
2	0,681	84,04	0,677	83,852
3	0,681	84,04	0,677	83,852
4	0,681	84,04	0,677	83,852
5	0,681	84,04	0,677	83,852
6	0,681	84,04	0,677	83,852
7	0,681	84,04	0,677	83,852
8	0,681	84,04	0,677	83,852
9	0,681	84,04	0,677	83,852
10	0,681	84,04	0,677	83,852
Rata-rata	0,681	84,04	0,677	83,852

17. Pengujian jumlah data 150 dengan jumlah cluster bernilai 3.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,553	77,63	0,553	77,656
2	0,517	75,84	0,553	77,656
3	0,553	77,63	0,553	77,656
4	0,551	77,55	0,553	77,656
5	0,553	77,63	0,553	77,656
6	0,553	77,63	0,553	77,656
7	0,517	75,84	0,514	75,702
8	0,551	77,55	0,514	75,702
9	0,553	77,63	0,553	77,656
10	0,551	77,55	0,553	77,656
Rata-rata	0,545	77,25	0,545	77,265

18. Pengujian jumlah data 150 dengan jumlah cluster bernilai 4.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,412	70,62	0,417	70,851

18. Pengujian jumlah data 150 dengan jumlah cluster bernilai 4. (lanjutan)

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,412	70,62	0,417	70,851
2	0,415	70,77	0,497	74,87
3	0,551	77,55	0,496	74,82
4	0,498	74,89	0,414	70,7
5	0,415	70,77	0,419	70,936
6	0,495	74,76	0,417	70,851
7	0,497	74,87	0,414	70,7
8	0,495	74,76	0,419	70,936
9	0,419	70,93	0,497	74,87
10	0,497	74,86	0,497	74,87
Rata-rata	0,47	73,48	0,449	72,44

19. Pengujian jumlah data 150 dengan jumlah cluster bernilai 5.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,371	68,56	0,439	71,969
2	0,493	74,64	0,394	69,684
3	0,428	71,38	0,487	74,356
4	0,44	72,02	0,374	68,679
5	0,4	70	0,372	68,594
6	0,448	72,38	0,369	68,439
7	0,374	68,68	0,487	74,339
8	0,372	68,61	0,439	71,969
9	0,372	68,61	0,376	68,783
10	0,375	68,75	0,451	72,532
Rata-rata	0,407	70,36	0,419	70,934

20. Pengujian jumlah data 150 dengan jumlah cluster bernilai 6.

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
1	0,367	68,36	0,365	68,238
2	0,44	72	0,367	68,335

20. Pengujian jumlah data 150 dengan jumlah cluster bernilai 6. (lanjutan)

Run	Nilai Silhouette		Nilai Silhouette	
	Rentang -1 sampai 1	Persentase	Rentang -1 sampai 1	Persentase
3	0,367	68,33	0,462	73,094
4	0,358	67,88	0,348	67,377
5	0,374	68,71	0,368	68,42
6	0,347	67,35	0,328	66,384
7	0,349	67,45	0,467	73,349
8	0,358	67,88	0,318	65,908
9	0,479	73,97	0,432	71,59
10	0,336	66,81	0,366	68,323
Rata-rata	0,377	68,87	0,382	69,102

LAMPIRAN 2. Hasil Pengujian Akurasi *K-Means* Konvensional

1. Pengujian akurasi jumlah cluster bernilai 2.

Run	Akurasi (%)	
	Euclidean	*Manhattan*
1	70,9	70,9
2	70,9	70,9
3	70,9	70,9
4	70,9	70,9
5	70,9	70,9
6	70,9	70,9
7	70,9	70,9
8	70,9	70,9
9	70,9	70,9
10	70,9	70,9
Rata-rata	70,9	70,9

2. Pengujian akurasi jumlah cluster bernilai 3.

Run	Akurasi (%)	
	Euclidean	*Manhattan*
1	94,4	94,44
2	94,4	76,65
3	86,7	94,44
4	78,5	94,44
5	94,4	77,12
6	86,7	85,19
7	86,7	94,44
8	86,7	94,44
9	94,4	94,44
10	77,1	77,12
Rata-rata	88	88,27

3. Pengujian akurasi jumlah cluster bernilai 4.

Run	akurasi (%)	
	Euclidean	*Manhattan*
1	95,8	95,83
2	90	88,89
3	88,1	86,81

3. Pengujian akurasi jumlah cluster bernilai 4. (lanjutan)

Run	akurasi (%)	
	Euclidean	*Manhattan*
4	86,8	88,89
5	90	80,06
6	88,1	95,83
7	88,1	86,81
8	95,8	91,78
9	95,8	89,72
10	88,1	86,81
Rata-rata	90,7	89,14

4. Pengujian akurasi jumlah cluster bernilai 5.

Run	akurasi (%)	
	Euclidean	*Manhattan*
1	90,5	91,11
2	92	96,67
3	93,3	94,64
4	95,3	91,78
5	89,4	94,64
6	89,4	93,33
7	90,5	95
8	93,3	88,33
9	93,3	91,78
10	87,1	91,78
Rata-rata	91,4	92,91

5. Pengujian akurasi jumlah cluster bernilai 6.

Run	akurasi (%)	
	Euclidean	*Manhattan*
1	92,1	92,59
2	91	93,15
3	93,5	95,83
4	90,6	93,15
5	94,4	92,46
6	91,5	92,62
7	91,5	95,83
8	93,2	95,83

5. Pengujian akurasi jumlah cluster bernilai 6. (lanjutan)

Run	akurasi (%)	
	Euclidean	*Manhattan*
9	90,5	95,54
10	91,2	95,24
Rata-rata	91,9	94,22