Meningkatkan Akurasi dan Efisiensi Algoritma *K-Means Clustering* dengan Mengubah Proses Inisialisasi *Centroid* dan Metode *Canopy Clustering*

Improving the Accuracy and Efficiency of K-means Clustering Algorithm using Modified Centroid Initialization Process and Canopy Clustering Method

Tugas Akhir

Diajukan untuk memenuhi sebagian dari syarat
untuk memperoleh gelar Sarjana
Program Studi Teknik Informatika
Fakultas Informatika
Universitas Telkom

Irfan Ardiansyah
1103091168

Program Studi Teknik Informatika
Fakultas Informatika
Universitas Telkom
Bandung
2016

Lembar Pernyataan

Dengan ini saya menyatakan bahwa Tugas Akhir dengan judul " **Meningkatkan Akurasi dan Efisiensi Algoritma K-Means Clustering dengan Mengubah Proses Inisialisasi Centroid dan Metode Canopy Clustering**" beserta seluruh isinya adalah benar-benar karya saya sendiri dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Bandung, 31 Agustus 2016
Yang membuat pernyataan,

(Irfan Ardiansyah)

Lembar Pengesahan

Meningkatkan Akurasi dan Efisiensi Algoritma K-Means Clustering dengan Mengubah Proses Inisialisasi Centroid dan Metode Canopy Clustering

Improving the Accuracy and Efficiency of K-means Clustering Algorithm using Modified Centroid Initialization Process and Canopy Clustering Method

Irfan Ardiansyah
1103091168

Tugas Akhir ini telah diterima dan disahkan untuk memenuhi sebagian dari syarat untuk memperoleh gelar Sarjana Program Studi Sarjana Teknik Informatika Fakultas Informatika
Universitas Telkom

Bandung, 31 Agustus 2016

Menyetujui

Pembimbing I	Pembimbing II
(Shaufiah, S.T., M.T.)	(Veronikha Effendy, S.T., M.T.)
NIP: 06820332-1	NIP: 14791532-1

Ketua Program Studi
Sarjana Teknik Informatika,

(Moch. Arif Bijaksana Ir., M.Tech., Ph.D)
NIP. 03650312-4

Abstrak

Clustering atau pengelompokan merupakan salah satu metode yang digunakan untuk memecahkan persoalan pada *data mining*. Terdapat beberapa algoritma yang digunakan untuk menyelasikan permasalahan pada *clustering*, salah satunya yaitu *k-means clustering*. Namun terdapat beberapa kelemahan pada algoritma *k-means* sehingga dapat mengurangi akurasi dan efisiensi. Salah satu kelemahan tersebut yaitu penentuan pusat *cluster* atau inisialisasi *centroid* secara acak dapat menyebabkan hasil *cluster* yang tidak akurat karena pusat *cluster* yang tidak tersebar ke seluruh *dataset*. Pada penelitian ini akan dibahas mengenai metode untuk meminimalisir kelemahan tersebut sehingga dapat menghasilkan akurasi dan efisiensi yang lebih baik. Penentuan pusat *cluster* atau inisialisasi *centroid* mengunakan algoritma *k-means++* menyebabkan persebaran *cluster* pada *data point* lebih merata jika dibandingkan dengan menggunakan *k-means* yang memilih secara acak pusat-pusat *cluster* yang dituju. Metode *canopy* atau *canopy clustering* ialah suatu metode yang biasa digunakan sebagai langkah awal dari metode *clustering* seperti *k-means* dan *expectation maximization* (EM). Metode *canopy* memiliki algoritma yang sederhana, cepat, dan akurat untuk mengelompokkan data ke dalam kelompok tertentu. Dengan mengkombinasikan metode *canopy* dengan metode *clustering* lainnya seperti *k-means* akan mengurangi penghitungan jarak antara *data point* yang biasanya memakan waktu dengan cara membentuk *canopy* untuk membatasi jumlah *data point* yang akan dihitung masing-masing jaraknya. Dengan menggunakan metode *canopy* dan *k-means++* tingkat akurasi yang dihasilkan yaitu 98.3% untuk *dataset* dim1024 dan 87.7% untuk dataset Iris. Sedangkan dari segi waktu *canopy* dan *k-means++* memperoleh hasil dalam waktu 7 detik pada *dataset* dim1024 dan 0.77 detik pada *dataset* iris.

Kata kunci: *Data mining, Clustering, K-means Clustering, Canopy Clustering, Centroid Initialization. K-Means++.*

Abstract

Clustering or grouping is one of the method used to solve problems in data mining. There are several algorithms available to solve such problems in clustering, one of which is k-means clustering. However, there are some weaknesses in the k-means algorithm so that it reduce the accuracy and efficiency of the result. One of these weaknesses is the determination of the cluster center or centroid initialization randomly can cause inaccurate results because the cluster center does not spread within the entire dataset. In this study some methods will be discussed to minimize these weaknesses hence improving accuracy and efficiency of the process. Determining cluster center or initial cluster centroid algorithm using K-Means++ led to distribution of clusters in the data point is more prevalent when compared to using K-Means which choosing random cluster centers are intended. Canopy method or canopy clustering is a method commonly used as an initial step of clustering methods such as K-Means and Expectation Maximization (EM). Canopy clustering has a simple, fast, and cheap process to classify the data into specific groups. By combining Canopy with other clustering methods such as K-Means will reduce the distance calculation between the data points that usually taken by forming a canopy to limit the number of data points to be calculated. By using canopy and k-means ++ generating the accuracy of 98.3% for dim1024 dataset and 87.7% for Iris dataset. In terms of running time, canopy and k-means ++ get the results within 7 seconds on dim1024 dataset and 0.77 seconds in the iris dataset.

Keywords: Data mining, Clustering, K-means Clustering, Canopy Clustering, Centroid Initialization, K-Means++

Lembar Persembahan

Pada lembar persembahan ini, penulis ingin mengucapkan banyak terimakasih kepada pihak-pihak yang senantiasa membantu penulis dalam pembuatan tugas akhir ini. Adapun pihak-pihak tersebut antara lain:

1. Allah SWT yang atas berkat dan rahmat-NYA lah saya masih bisa menyelesaikan perkuliahan ini.
2. Nabi Muhammad SAW yang atas ajaran dan panutannya lah saya terus berusaha untuk lebih baik.
3. Ayahanda Muchlis yang selalu menuntun dan mendorong penulis untuk terus maju kedepan, untuk menjadi orang yang sukses, untuk selalu ingin meningkatkan pribadi. Yang dengan sabarnya memberi kesempatan lagi dan lagi. Terima kasih atas segala support dan motivasinya, dan tidak ada kata lain selain maaf yang sedalam-dalamnya dan tiada henti-hentinya.
4. Ibunda Eniwati yang selalu memberi semangat, selalu memberikan doa dan pengorbanan bagi anak-anaknya. Seorang ibu tempat penulis berkeluh kesah, seorang ibu yang dengan sabarnya meladeni kekeras kepalaan kedua anaknya. Terima kasih mamah, hanya Allah yang dapat membalas semua yang mamah dan ayah berikan.
5. Adik tersayang Hasya Arsitarini yang selalu mendukung dan menghibur penulis dikala waktu sulit dan gundah. Semoga adek segera lulus cepat waktu.
6. Ibu Shaufiah selaku dosen pembimbing I yang tiada hentinya mendorong dan memberikan solusi bagi penulis di masa-masa kritis seperti ini. Tanpa dukungan dan motivasi ibu penulis tidak akan mampu menyelesaikan tugas akhir ini, terima kasih yang sebesar-besarnya kepada ibu.
7. Ibu Veronikha Effendy selaku dosen pembimbing II yang dengan sabar dan baik hati menerima penulis sebagai murid bimbingan. Terima kasih atas segala bimbingan dan panduannya.
8. Bapak Moch. Arif Bijaksana selaku Kaprodi Fakultas Informatika yang selalu membantu penulis dikala kesulitan dalam proses pengerjaan Tugas Akhir ini.
9. Bapak Endro Ariyanto selaku dosen wali yang selalu mengawasi dan mengarahkan penulis dalam proses pengerjan Tugas Akhir ini.
10. Seluruh dosen Telkom University khususnya dosen Teknik Informatika yang telah membagikan ilmunya kepada penulis agar penulis dapat mengerjakan Tugas Akhir ini. Semoga ilmu yang diajarkan akan dapat berguna bagi masa depan penulis.
11. Seluruh teman-teman BSYE yang telah rela meluangkan waktunya bersama-sama, melewati suka dan duka selama masa perkuliahan ini, kepada Bagoes Siswo, Prafajar Rizkyanno, Lukman Azis, M. Abdurrachman Irfandi, Micko

Lesmana, Putut Andre, Alfa Pahlawan, Yoga Prakoso, Lalu Fikri, dan Pocit yang telah berbagi banyak pengalaman bersama penulis.

12. Seluruh teman-teman B&G yang canda tawanya tiada henti, Alfa Pahlawan, Yuricho Billy, Delfi Kusumawardhani, Dian Fitrah, Dinda Larasati, Nadiya Trisnawati, dan tak lupa juga Rizka Utami.
13. Seluruh teman-teman Intrinsix.
14. Teman-teman kosan Drenaya.
15. Pihak-pihak lain yang tidak bisa penulis sebutkan satu per satu.

Kata Pengantar

Puji dan syukur penulis panjatkan kepada kehadirat Allah SWT atas rahmat dan berkah-NYA sehingga tugas akhir ini yang berjudul " *Meningkatkan Akurasi dan Efisiensi Algoritma K-Means Clustering dengan Mengubah Proses Inisialisasi Centroid dan Metode Canopy Clustering*" dapat terselesaikan. Tugas akhir ini disusun sebagai salah satu syarat sidang demi memperoleh gelar Sarjana di Fakultas Informatika Universitas Telkom.

Penulis menyadari bahwa dalam penulisan dan penyampaiannya terdapat banyak kekurangan. Oleh karena itu saran dan kritik sangatlah membantu penulis dalam membangun tugas akhir ini.

Akhir kata semoga ilmu yang terdapat pada buku tugas akhir ini dapat bermanfaat bagi kita semua.

Bandung, 31 Agustus 2016

Penulis

Daftar Isi

Daftar Gambar

Daftar Tabel

1. Pendahuluan

1.1 Latar belakang

Clustering atau pengelompokan merupakan salah satu metode yang paling sering digunakan untuk memecahkan suatu persoalan data mining. Meskipun *clustering* identik dengan *classification*, namun terdapat perbedaan di antara kedua metode, yaitu pada klasifikasi data yang diproses sudah diketahui sebelumnya kategori-kategori atau *label* yang dimiliki oleh data sehingga sistem dapat mempelajari pola masing-masing kategori. Metode seperti ini biasa disebut dengan *supervised learning*. Sedangkan pada metode *clustering*, tidak diketahui sebelumnya *label-label* tersebut atau *label* yang tersedia digunakan sebagai bahan uji coba sistem untuk mengukur tingkat akurasi dari hasil *cluster* yang diperoleh. Tujuan dari *clustering* sendiri ialah membentuk kelompok-kelompok dari sejumlah data di dalam data set sehingga data-data yang termasuk ke dalam suatu kelompok memiliki kemiripan tertentu, sedangkan data-data pada kelompok yang berbeda memiliki perberbedaan satu dengan yang lain.

Terdapat beberapa metode *clustering* yang telah dikembangkan pada saat ini, masing-masing metode memiliki kelebihan dan kekurangan sehingga untuk kasus tertentu terdapat metode yang lebih efisien dibandingkan dengan metode lainnya. Salah satu metode yang paling sering digunakan yaitu metode *k-means* karena mudah digunakan dan sederhana. Algoritma *k-means* mengelompokkan sejumlah *data point* ke dalam kelompok yang berjumlah dua atau lebih dengan meminimalkan fungsi objektif pada proses pengelompokan. Metode *canopy* atau *canopy clustering* ialah suatu metode yang biasa digunakan sebagai langkah awal dari metode *clustering* seperti *k-means*. Metode *canopy* memiliki algoritma yang sederhana, cepat, dan akurat untuk mengelompokkan data ke dalam kelompok tertentu. Tujuan dari penggunaan *canopy* ialah untuk mempercepat proses *clustering* dengan cara membentuk *canopy* untuk membatasi jumlah *data point* yang akan dihitung masing-masing jaraknya.

Metode *canopy clustering* digunakan sebagai proses awal dari algoritma *k-means* agar waktu yang dibutuhkan proses tersebut dapat diminimalisir. Namun meskipun metode *canopy clustering* digunakan, masih terdapat kekurangan pada kombinasi kedua metode tersebut. Kekurangan terdapat pada metode *k-means* yaitu sangat bergantung pada penentuan pusat *cluster* awal yang berpengaruh pada seringnya terjadi perbedaan hasil *cluster* jika pusat *cluster* awal berbeda. Algoritma *k-means* juga dipengaruhi oleh urutan data yang dibaca dan juga rentan terhadap pengaruh gangguan dan *data point* yang memiliki jarak yang jauh dengan *data point* lainnya (terpencil). Untuk menanggulangi kekurangan tersebut, diajukan metode *k-means* yang lebih akurat khususnya dalam pemilihan pusat *cluster* awal dengan menentukan pusat *cluster* awal secara sistematis tidak acak seperti pada algoritma *k-means* umumnya. Dengan menggunakan algoritma *k-means++*, inisialisasi pusat *cluster* ditentukan dengan menggunakan distribusi probabilitas dari jarak data poin terhadap pusat *cluster* yang sudah dipilih. Pada penelitian-penelitian sebelumnya [3], [4], [8], [10], [12] telah dibuktikan bahwa teknik *canopy clustering* dapat mempersingkat waktu yang dibutuhkan sistem

untuk mengelola suatu data sekaligus mengingkatkan akurasi hasil *clustering* yang dibentuk. Sedangkan metode *k-means++* memberikan hasil *cluster* yang lebih tersebar karena pada inisialisasi *centroid* tidak digunakan metode inisialisasi secara acak.

1.2 Perumusan masalah

Berdasarkan latar belakang yang telah dipaparkan di atas, berikut adalah perumusan masalah yang diangkat :

a. Bagaimana mengimplementasikan algoritma *canopy* sebagai inisialisasi dari metode *k-means* ?
b. Bagaimana mengimplementasikan metode *k-means++* untuk inisialisasi pusat *cluster* ?
c. Berapakah akurasi yang didapatkan dengan mengkombinasikan algoritma *canopy* dan algoritma *k-means* yang diajukan ?
d. Apakah akurasi yang didapatkan pada kombinasi kedua metode lebih baik jika dibandingkan dengan hanya menggunakan *k-means* ?
e. Apakah waktu yang dibutuhkan algoritma untuk memproses data pada kombinasi kedua metode lebih cepat dibandingkan dengan hanya menggunakan *k-means* ?

Adapun yang menjadi batasan masalah pada penelitian Tugas Akhir ini adalah :

a. Algoritma *canopy* digunakan sebagai metode *pre-clustering* sebelum melakukan *clustering* dengan menggunakan algoritma *k-means++* untuk inisialisasi *centroid.*
b. Analisis yang dilakukan yaitu dengan membandingkan algoritma *k-means* standar dan algoritma *k-means* yang telah dikombinasikan dengan *k-means++* dan *canopy clustering.*.
c. Analisis dilakukan dengan membandingkan akurasi dan waktu *running* yang dibutuhkan kedua metode.
d. Masing-masing data dapat termasuk kedalam lebih dari satu *cluster* sehingga dapat terjadi *overlapping.*
e. Dataset yang dipergunakan pada penelitian ini yaitu *dataset* dim1024 yang memiliki 1024 objek dan atribut, dan *dataset* iris dengan 150 objek dan 4 atribut.

1.3 Tujuan

Berdasarkan perumusan masalah diatas, tujuan yang ingin dicapai dari penelitian ini adalah sebagai berikut :

1. Mengimplementasikan algoritma *canopy* sebagai inisialisasi dari metode *k-means.*
2. Mengimplementasikan metode *k-means++* untuk mencari pusat *cluster* awal.
3. Menghitung akurasi yang didapatkan dengan mengkombinasikan algoritma *canopy* dan *k-means++*.

4. Menghitung akurasi yang didapatkan pada *k-means,* kemudian membandingkan dengan akurasi yang didapatkan pada kombinasi kedua metode.
5. Menghitung waktu algoritma untuk memproses data yang didapatkan pada *k-means* dan pada kombinasi kedua metode, kemudian membandingkan waktu proses pada *k-means* dengan waktu proses pada kombinasi kedua metode.

1.4 Metodologi penyelesaian masalah

Tahapan-tahapn yang akan diterapkan dalam menyelesaikan masalah pada penelitian ini adalah sebagai berikut :

1. Pengumpulan data dan studi literatur
   Pada tahap pengumpulan studi literatur akan dilakukan pencarian mengenai *data mining, clustering, k-means, k-means++, canopy clustering,* evaluasi hasil *clustering,* teknik kombinasi *canopy clustering* dan *k-means,* dan *data analysis.* Literatur didapatkan dari internet, perpustakaan Telkom University, jurnal yang dapat diunduh secara gratis, dan sumber-sumber lainnya. Terdapat dua jenis *dataset* yang akan dipergunakan untuk melakukan analisis penelitian, yaitu *Dim1024 Dataset* diunduh melalui [https://cs.joensuu.fi/sipu/datasets/] dan *Iris Dataset* melalui [https://archive.ics.uci.edu/ml/datasets/Iris]
2. Perancangan sistem
   Melakukan analisis dan perancangan sistem terhadap sistem yang akan dirancang. Menganalisis metode dan algoritma apa yang sesuai untuk menyelesaikan masalah, bahasa pemrograman yang akan digunakan, arsitektur, fungsionalitas dan antarmuka sistem yang akan dirancang tersebut.
3. Pengujian sistem
   Pengujian dilakukan dengan mengimplementasikan algoritma *k-means++* dan mengkombinasikan dengan *canopy clustering* pada dua jenis data yang berbeda, yaitu *Dim1024* dan Iris. Algoritma *k-means* tanpa menggunakan *canopy clustering* akan diimplementasikan pada data yang sama guna menghasilkan *cluster*. Hasil dari pengujian sistem akan digunakan sebagai bahan untuk melakukan analisis.
4. Analisis sistem
   Analisis sistem digunakan untuk mengetahui tingkat akurasi dari sistem yang telah dibuat dengan cara menghitung akurasi dari hasil *cluster* yang diperoleh melalui modifikasi algoritma *k-means* kombinasi dengan *canopy clustering* dan dibandingkan dengan akurasi hasil *cluster* pada algoritma *k-means* tanpa menggunakan *canopy clustering*. Waktu *running* kedua algoritma juga akan dibandingkan untuk mengetahui efisiensi dari kedua algoritma.
5. Pembuatan laporan
   Pembuatan laporan merupakan tahapan terakhir dari penelitian ini, didapat dari langkah-langkah penilitian yang telah dilakukan, hasil dari penelitian,

analisis, dan juga kesimpulan yang didapat dari penelitian sebagai dokumentasi terhadap seluruh tahapan penelitian yang dilakukan.

1.5 Sistematika Penulisan

Adapun sistematika penulisan penelitian Tugas Akhir ini adalah sebagai berikut:

1. BAB 1 Pendahuluan
   Pada Bab 1 diuraikan isi dan rencana pengerjaan penelitian Tugas Akhir secara keseluruhan yang meliputi latar belakang, masalah, rumusan masalah, tujuan, batasan masalah, dan metode penyelesaian masalah yang diterapkan.
2. BAB 2 Dasar Teori
   Bab 2 memaparkan dasar-dasar teori yang berkaitan dengan *Data Mining*, *Clustering*, *K-Means*, *Canopy Clustering*, *Implementasi algoritma K-means++ dengan canopy clustering*, dan *Validasi Cluster*.
3. BAB 3 Perancangan dan Implementasi
   Perancangan sistem, gambaran umum sistem, *dataset*, dan analisis kebutuhan sistem yang akan dibangun dipaparkan pada bab ini. Selanjutnya akan dilakukan proses implementasi.
4. BAB 4 Pengujian dan Analisis
   Pada bab ini akan dibahas skenario dan hasil pengujian yang dilakukan pada hasil implementasi sistem.
5. BAB 5 Kesimpulan dan Saran
   Bab ini berisi kesimpulan dan saran yang didapatkan dari hasil implementasi sistem secara keseluruhan

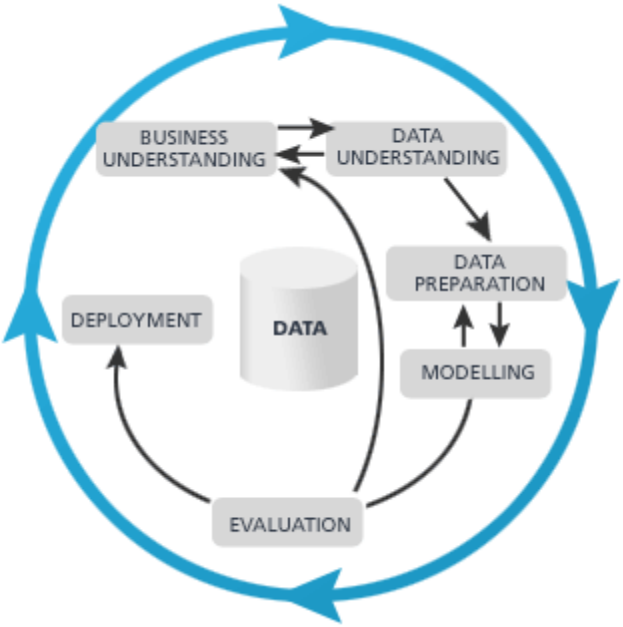
2. Tinjauan Pustaka

2.1 Data Mining

Data mining merupakan suatu kajian dalam bidang komputer yang khusus membahas mengenai proses penggalian informasi dari sekumpulan data yang tersedia. Biasanya teknik *data mining* digunakan untuk membantu perusahaan atau institusi dalam mengelola data yang jumlahnya sangat banyak. Penggunaan *data mining* pada suatu perusahaan bertujuan untuk mendapatkan suatu analisis dari data yang dimiliki perusahaan tersebut, analisis dapat digunakan untuk berbagai macam kepentingan seperti evaluasi kinerja, strategi pemasaran, pengembangan teknologi dan sumber daya manusia, pemeliharaan alat, untuk menunjang proses penilaian kerja karyawan, penghematan sumber daya, dan lain sebagainya.

Pada *data mining* dikenal istilah-istilah seperti data, *data set*, informasi, dan pengetahuan. Data merupakan kumpulan informasi teks, angka, ataupun fakta yang dapat diproses oleh komputer. Dalam komputasi, datamerupakan informasi yang telah diterjemahkan ke dalam bentuk yang lebih mudah untuk diproses. Sehubungan dengan media komputer dan transmisi saat ini, data digunaka dalam bentuk digital biner untuk mempermudah proses pada komputer. Sedangkan informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Informasi dapat dianalisis untuk mendapatkan pola-pola keterikatan pada data sehingga dapat digunakan sebagai pengetahuan.

Data set dapat diartikan sebagai seebuah informasi yang memiliki keterikatan yang terdiri dari unsur-unsur yang terpisah tetapi dapat dimanipulasi sebagai sebuah unit oleh komputer. Tidak seluruh *data set* direpresentasikan secara utuh, kadang kala terdapat *data set* yang memiliki nilai-nilai yang hilang baik disengaja untuk kepentingan pengujian sebuah penelitian maupun tidak sengaja atau natural, terdapat data-data yang salah, data-data yang memiliki kesamaan nilai, data yang tidak konsisten, dan sebagainya. Permasalahan-permasalahan pada *data set* tersebut dapat mempengaruhi mempengeruhi kualitas dari *data set* itu sendiri sehingga menyebabkan kualitas dari hasil proses *data mining* juga akan berpengaruh. Untuk menanggulangi persoalan tersebut maka perlu dilakukan *preprocessing* sebelum *data set* diproses pada proses utama *data mining* [11]. Tujuan dari *preprocessing* ialah untuk memperbaiki atau mengoreksi *data set* sehingga data-data yang bersifat merugikan bagi proses tidak perlu digunakan. Beberapa metode yang dapat digunakan untuk *preprocessing* antara lain agregasi, *sampling*, pengurangan dimensi, dan lain-lain.



Gambar 2.1.1 *Alur proses data mining* [http://www.statsoft.com]

Data mining bekerja secara otomatis untuk menggali relasi-relasi yang rumit pada sebuah data set yang rumit. Teknik yang digunakan pada data mining diadaptasi dari teknik-teknik lainnya seperti basis data, statistika, artificial intelegence, machine learning, information retrival, dan teknik-teknik lain. Data mining dapat digunakan untuk memecahkan permasalahan pada berbagai bidang. Selain digunakan untuk persoalan manajemen, data mining juga dapat digunakan pada bidang ilmu pengetahuan seperti penelitian mengenai gen dan DNA, penelitian mengenai tingkah laku manusia, dan lain-lain.

Pada umumnya *data mining* terbagi menjadi enam teknik yang masing-masing dapat menyelesaikan permasahan tertentu yaitu klasifikasi, deteksi anomali, asosiasi, sumarisasi, regresi, dan *clustering*. Teknik klasifikasi digunakan untuk mencari pola pada data yang memiliki label kelas sehingga didapatkan aturan tertentu pada data tersebut yang dapat diterapkan kepada data baru. Proses klasifikasi dikategorikan sebagai *supervised learning*, yaitu ketika data yang diolah sudah diketahui label-label kelasnya untuk digunakan sebagi aturan atau generalisasi atribut data terhadap suatu kelas tertentu. Aturan-aturan ini diterapkan kepada data baru untuk mencari label kelas dari masing-masing data tersebut.

Sedangkan teknik deteksi anomali digunakan untuk mencari atau mendeteksi pola yang unik pada *dataset* sehingga dapat ditemukan kejanggalan pada data. Teknik ini biasa digunakan pada kasus-kasus seperti audit data keuangan sehingga jika ditemukan data yang tidak biasa artinya ada indikasi terjadi korupsi atau penggelapan uang.

Teknik berikutnya yaitu asoiasi adalah mencari keterikatan yang terdapat pada atribut-atribut pada *dataset* yang berguna untuk mencari pengaruh suatu atribut tertentu terhadap atribut lainnya. Teknik ini dapat digunakan untuk pemasaran suatu produk, data transaksi pembeli dapat dianalisis untuk mencari keterikatan barang apa saja yang sering dibeli dalam waktu yang bersamaan.

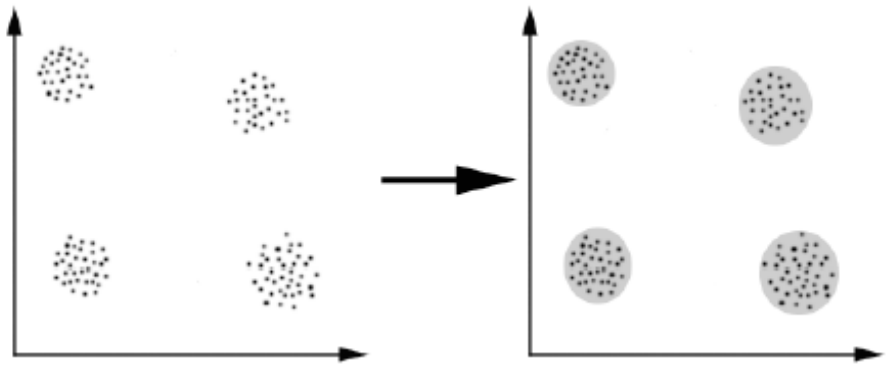
Informasi ini kemudian dapat digunakan untuk menawarkan produk tertentu seperti *make-up* kepada pembeli yang sering membeli produk-produk kecantikan. Sumarisasi merupakan teknik yang digunakan untuk merepresentasikan data sesuai dengan informasi yang berkaitan dengan permasalahan tertentu. Artinya informasi mana sajakah yang penting untuk digunakan pada *dataset* untuk kemudian informasi tersebut diolah kembali.

Selanjutnya regresi ialah mencari teknik tertentu yang paling optimal dalam menangani permasalahan pada data tertentu. Tujuannya adalah untuk mencari teknik yang lebih akurat ketika diterapkan pada data sehingga hasil informasi yang didapatkan dari data juga akurat.

Teknik terakhir yaitu *clustering* bertujuan untuk mengelompokkan data kedalam kelompok-kelompok tertentu sehingga kelompok yang terbentuk memiliki kemiripan yang tinggi pada masing-masing data dan memiliki perbedaan yang tinggi pada data yang berada pada kelompok lain. *Clustering* meiliki kemiripan dengan teknik klasifikasi yaitu bertujuan untuk mengelompokkan data sesuai dengan atributnya, perbedaannya ialah pada *clustering* data label kelas tidak diketahui sebelumnya sehingga pengelompokan tidak dilakukan berdasarkan aturan tertentu. Jika data yang digunakan pada *clustering* memiliki label kelas, maka label kelas tersebut akan digunakan untuk melakukan validasi terhadap kelompok yang terbentuk. Teknik *clustering* akan dijelaskan lebih lanjut pada bagian berikutnya.

2.2 Clustering

Clustering merupakan metode pengelompokan data berdasarkan kelas-kelas tertentu sehingga obyek-obyek yang berada pada suatu kelas yang sama memiliki tingkat kemiripan yang tinggi sedangkan obyek-obyek-obyek yang berada pada kelas yang berbeda memiliki tingkat perbedaan yang tinggi (Yang dan Wang, 2001). *Clustering* jugadapat diartikan sebagai bentuk dari pengkompresian data sehingga data yang berjumlah besar diubah menjadi *prototype* ataupun *cluster* data dengan jumlah yang kecil (Giles dan Draeseke, 2001). Pengelompokan dilakukan berdasarkan jenis data dan aplikasi untuk menentukan kelas-kelas yang mungkin terbentuk dengan jenis kemiripan data berbeda dapat digunakan. *Clustering* bertujuan untuk menentukan pengelompokan dari data-data yang tidak berlabel sehingga data dapat mudah dikenali berdasarkan karakteristiknya [5].



Gambar 2.2.1 *Contoh grafik clustering* [http://home.deib.polimi.it]

Metode *clustering* dapat digunakan pada beberapa bidang keahlian seperti bidang *marketing* untuk menemukan pengelompokan dari pelanggan yang memiliki tingkah laku yang serupa dengan data pelanggan yang terdiri dari sifat dan catatan mengenai transaksi yang dilakukan oleh pelanggan tersebut. Dalam dunia medis, *clustering* dapat digunakan untuk mengelompokan penyakit-penyakit berdasarkan kemiripan dalam hal gejala yang ditimbulkan serta metode penyembuhan. Tujuan utama *clustering* adalah menentukan struktur data dengan cara meletakkan observasi yang mirip dalam satu kelompok. Pengelompokan hasil observasi yang mirip ke dalam satu kelompok didasarkan pada korelasi antar objek atau dapat juga dengan mengukur proximity pada ruang dua dimensi sehingga jarak antara dua observasi menunjukkan kesamaan. Langkah berikutnya adalah menentukan bagaimana membentuk cluster dan berapa jumlah cluster yang akan dibentuk (Imam Ghozali, 2009 : 312).

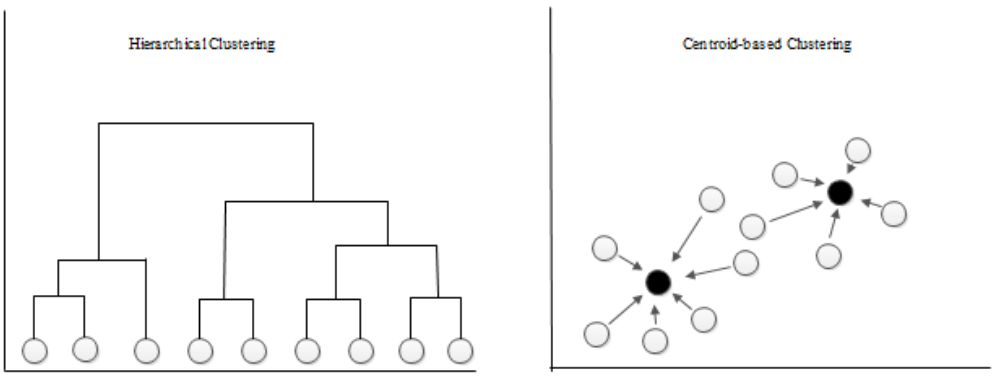
Tujuan dari *clustering* terbagi menjadi dua, yaitu *clustering* untuk kepentingan pemahaman dan *clustering* untuk kepentingan penggunaan. Untuk kepentingan pemahaman, kelompok yang terbentuk harus menangkap struktur alami data. Contoh *clustering* untuk kepentingan pemahaman ialah mengelompokkan gen-gen dengan fungsi yang sama, menemukan pola tekanan udara untuk memprediksi cuaca, pengelompokan pelanggan untuk kepentingan pemasaran. Sedangkan untuk kepentingan penggunaan bertujuan untuk mencari kelompok yang paling representatif pada data yang tersedia. Adapun contoh *clustering* untuk kepentingan penggunaan yaitu peringkasan, kompresi, pencarian tetangga terdekat, dan sebagainya. Terdapat berbagai jenis metode *clustering*, diantaranya yaitu *hierarchical clustering* atau dapat disebut *connectivity-based clustering* dan *centroid-based clustering*. *Hierarchical clustering* ialah teknik *clustering* yang memisahkan masing-masing data sebagai *cluster* untuk kemudian ditentukan persamaan masing-masing data terhadap data yang lainnya. Sedangkan teknik *centroid-based clustering* yaitu dengan menggunakan titik-titik pusat akan ditentukan jarak masing-masing data kepada titik-titik tertentu untuk kemudian data akan digabungkan terhadap titik-titik pusat yang jaraknya paling dekat.

2.2.1 Hierarchical Clustering

Hierarchical clustering memisahkan data sehingga masing-masing data direpresentasikan sebagai satu *cluster* untuk kemudian dicari persamaan suatu data dengan data lainnya. Jika dua buah data memiliki persaman yang tinggi maka kedua data tersebut akan disatukan menjadi sebuah *cluster*. Kemudian *cluster* tersebut akan dibandingkan dengan c*luster* data lainnya. Langkah ini akan berulang hingga hanya terdapat satu buah *cluster* yang berisi seluruh data sehingga didapatkan *hierarchical tree* yang merepresentasikan langkahlangkah yang diambil. *Hierarchical tree* ini digunakan untuk mencari kombinasi pengelompokan yang optimal sehingga didapatkan hasil berupa *cluster* yang optimal. Contoh algoritma *hierarchical clustering* yaitu *single-linkage clustering, complete-linkage clustering, average-linkage clustering,* dan sebagainya.

2.2.2 Centroid-based Clustering

Centroid-based clustering merupakan teknik *clustering* yang membagi data kedalam kelompok-kelompok tertentu sesuai dengan *centroid* atau pusat *cluster* terdekat yang terhadap masing-masing data tersebut. Untuk mengelompokkan data kedalam *cluster* masing-masing, akan dihitung jarak antara masing-masing data terhadap pusat *cluster* yang telah ditentukan. Jika suatu data memiliki jarak dengan suatu pusat *cluster* yang paling dekat diantara pusat *cluster* lainnya, maka data tersebut akan diasosiasikan sebagai anggota *cluster* dari pusat *cluster* yang dipilih tersebut. Langkah ini akan diulang hingga seluruh data sudah berada pada *cluster* dengan jarak menuju pusat *cluster* yang terdekat. Kemudian akan ditentukan pusat *cluster* baru dengan menghitung rata-rata masing-masing *cluster* yang terbentuk dan proses akan diulang kembali dengan menggunakan pusat *cluster* yang baru terbentuk hingga tidak ada lagi perubahan pusat *cluster*. Metode-metode yang termasuk kedalam *centroid-based clustering* antara lain *k-means, Expectation-maximization (EM),* dan lain sebagainya



Gambar 2.2.2 Perbedaan *Hierarchical* dan *Centroid-based Clustering*

2.3 K-Means Clustering

K-means merupakan teknik yang sangat sering digunakan untuk melakukan proses *clustering.* Selain mudahnya diimplementasikan, metode ini juga banyak dimodifikasi untuk keperluan tertentu sehingga terdapat banyak macam modifikasi algoritma ini dan masih banyak digunakan hingga saat ini meskipun sudah berusia hampir empat puluh tahun. Langkah dari algoritma ini dapat diidentifikasi menjadi 4 bagian yaitu inisialisasi *centroid,* klasifikasi data, penghitungan pusat *cluster* baru, dan meningkatkan hasil *cluster* hingga konvergen. Pada tahap inisialisasi *centroid,* sistem meilih secara acak poin-poin sejumlah *cluster* yang diinginkan dari sejumlah objek pada *dataset* yang akan dilakukan pembagian. Kemudian pada tahapan kedua, setelah sejumlah *centroid* dipilih akan dicari jarak poin-poin lain yang berada pada *dataset* untuk dimasukkan kedalam anggota *cluster* tersebut. Jarak yang digunakan pada umumnya yaitu *Euclidean Distance* atau *Manhattan Distance* meskipun terdapat varian-varian lain dalam menghitug jarak antar poin. Setelah seluruh poin didapatkan jarak terdekatnya dengan masing-masing *centroid,* proses *clustering* dimulai dengan memilah poin-poin berdasarkan jarak centroid terdekatnya. Kemudian tahapan terakhir yaitu akan ditentukan *centroid* baru dengan mencari nilai rata-rata dari masing-masing kelas atau kelompok yang terbentuk pada tahapan sebelumnya. Proses akan mengulang kembali dengan nilai rata-rata tersebut sebagai pusat *cluster* baru. Proses pengulangan berlangsung hingga memenuhi keadaan konvergen terpenuhi, yaitu tidak terjadi perubahan pusat *cluster* setelah dicari nilai rata-rata.

Karena langkah-langkahnya yang sederhana, algoritma *k-means* memiliki beberapa kelemahan yang berdampak pada buruknya hasil *cluster* yang didapat. Diantara kelemahan-kelemahan tersebut yaitu : 1) Proses inisialisasi *centroid* atau pemilihan pusat *custer* menggunakan metode acak kerap menimbulkan permasalahan yaitu ketika pusat *cluster* yang dipilih tidak secara merata sehingga menyebabkan proses berlangsung lama atau kondisi konvergen didapatkan setelah melakukan pengulangan dalam jumlah banyak. 2) Pada data yang memiliki dimensi tinggi atau terdapat banyak atribut, algoritma *k-means* membutuhkan waktu yang lama untuk menyelesaikan proses. 3) Jumlah *cluster* yang dibentuk harus ditentukan oleh *user* sehingga untuk mendapatkan hasil yang optimal algoritma harus diulang berkali-kali dengan beragam kombinasi jumlah *cluster* pada setiap percobaan.

Untuk menanggulangi kelemahan-kelemahan pada metode *k-means* diberikan banyak modifikasi seperti untuk menanggulangi permasalahan 1 dapat digunakan proses inisialisasi *centroid* yang lebih sistematis dengan menggunakan distribusi khusus, untuk permasalahan 2 dapat digunakan metode-metode reduksi dimensi yaitu mengurangi dimensi agar perhitungan lebih cepat tanpa mengurangi tingkat akurasi seperti metode PCA (*Principal Component Analysis*) maupun *Canopy Clustering*. Sedangkan untuk permasalahan 3 dapat digunakan modifikasi-modifikasi yang akan mencari jumlah *cluster* optimal secara otomatis misalnya menggabungkan metode *hierarchical clustering* dengan algoritma *k-means.* Pada penelitian ini akan dibahas mengenai metode-metode yang dapat digunakan untuk meminimalisir kelemahan 1 dan kelemahan 2 yaitu dengan menggunakan algoritma *k-means++* untuk inisialisasi pusat *cluster* dan *canopy*

clustering untuk mempercepat proses pada data dengan dimensi tinggi. Metode-metode tersebut akan dijelaskan pada bagian selanjutnya.

2.3.1 **Algoritma K-Means++**

Pada tahap pemilihan *centroid* atau inisialisasi *centroid* metode *k-means* memilik sejumlah titik pusat secara acak tanpa adanya aturan khusus ataupun distribusi khusus posisi titik pusat yang lebih baik dari kumpulan data. Hal tersebut memungkinkan adanya sekumpulan titik pusat yang terletak pada daerah tertentu sehingga tidak seluruh area pada data poin memiliki titik pusat. Meskipun pada iterasi-iterasi selanjutnya titik pusat tersebut akan menyebar sehingga mendekati daerah-daerah lain pada data poin, algoritma *k-means* sendiri bersifat memenuhi lokal optimum yaitu ketika konvergensi sudah tercapai dengan kata lain tidak ada titik pusat *cluster* yang berpindah maka algoritma menganggap kondisi tersebut sudah memeberikan hasil *cluster* yang optimal. Sedangkan masih ada kemungkinan kondisi yang menghasilkan *cluster* yang lebih baik. *K-means* akan memilih suatu solusi ketika ditemukan tanpa mempertimbangkan solusi lainnya. Untuk menanggulangi terjadinya lokal optimum pada hasil *cluster* maka diperlukan suatu metode yang mampu memberika solusi lebih baik dengan pemilihan inisialisasi yang lebih baik.

Salah satu metode yang dapat menanggulangi hal tersebut yaitu metode *k-means++* dengan memilih titik pusat kedua dan seterusnya dengan menggunakan probabilitas :

$$\frac{D(x)^2}{\sum_{x \in X} D(x)^2} \qquad (2.1)$$

Dimana x merupakan data poin yang berada pada himpunan X dan D(x) merupakan jarak terdekat antara data poin x terhadap seluruh pusat *cluster* yang sudah dipilih sebelumnya. Algoritma ini memiliki langkah sebagai berikut :

1. Pilih secara acak suatu titik pada data poin X sebagai pusat *cluster* pertama c_1
2. Untuk seluruh data poin x, hitung D(x), jarak antara data poin x dengan pusat *cluster* terdekat yang sudah dipilih sebelumnya.
3. Pilih pusat *cluster* baru secara acak menggunakan distribusi probabilitas pada persamaan (2.1)_ di atas.
4. Ulangi langkah 2 dan 3 hingga seluruh pusat *cluster* telah terpilih.

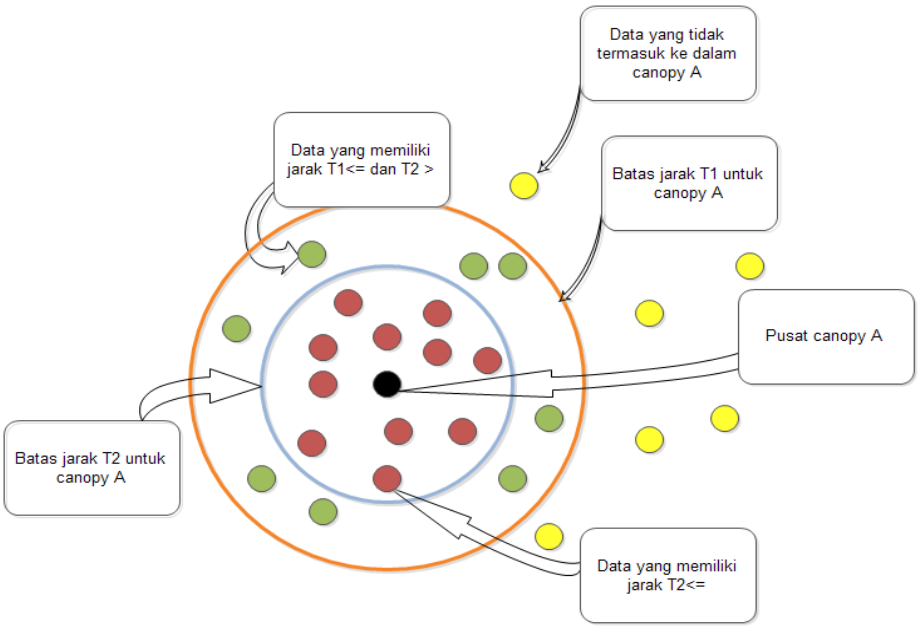
Dengan menggunakan distribusi probabilitas, pusat *cluster* yang terpilih berada pada posisi yang lebih merata atau lebih tersebar sehingga memungkinkan *cluster* yang tersebar sejak inisialisasi proses.

2.4 Canopy Clustering

Algoritma *canopy clustering* adalah algoritma *pre-clustering* yang bersifat tanpa pengawasan atau *unsupervised learning method*, sering digunakan sebagai langkah *preprocessing* untuk algoritma k-means atau algoritma *hierarcial clustering*. Metode ini pertama kali muncul dalam sebuah *paper* oleh Andrew McCallum, Kamal Nigam dan Lyle Ungar. Pada dasarnya *canopy clustering* digunakan untuk mempercepat proses pengelompokan dalam kasus *data set* yang besar, di mana implementasi langsung dari algoritma utama mungkin tidak praktis karena ukuran dari *data set*. Proses pengelompokkan dengan *canopy clustering* terbagi menjadi dua tahap, pada tahap pertama *cheap distance measure* digunakan untuk membentuk beberapa himpunan data yang disebut dengan *canopy* sedangkan pada tahap selanjutnya *actual* atau *expensive distance* digunakan pada data yang telah terhimpun dalam subset Secara singkat, *canopy* merupakan himpunan yang berisi data-data dengan jarak dari titik pusat *canopy* masih di dalam batas yang ditentukan (threshold). Data-data yang diproses memiliki kemungkin termasuk ke dalam lebih dari satu *canopy*, namun paling tidak harus termasuk ke dalam satu *canopy*.

Tahap penting dalam algoritma *canopy clustering* ialah menentukan batas dari masing-masing *canopy* yang terbentuk agar mampu menghasilkan *output* yang baik. Pada tahap kedua metode *clustering* utama seperti *k-means, expectation-maximization,* digunakan dengan data yang sudah ditentukan pada masing-masing *canopy*. Pada tahap ini digunakan *expensive distance measure* untuk mengukur jarak data pada masing-masing pusat *cluster. Cheap distance measure* ialah penghitungan jarak dari data ke pusat atau *centroid* tenpa menggunakan seluruh atribut yang tersedia pada data tersebut. Sedangkan pada *expensive distance measure* seluruh atribut yang tersedia akan digunakan untuk mengukur jarak.

Algoritma *canopy* menggunakan dua batas jarak (*threshold*) T1> T2 untuk diproses. Algoritma dasar dimulai dengan satu set poin dan menghapus satu secara acak. Langkah selanjutnya membuat *canopy* mengandung titik ini dan melakukan pengulangan melalui sisa dari set point. Pada setiap titik, jika jarak dari titik pertama adalah <T1, kemudian tambahkan titik ke cluster. Jika, di samping itu, jarak <T2, kemudian menghapus titik dari himpunan. Dengan cara ini poin yang sangat dekat dengan aslinya akan menghindari semua proses lebih lanjut. Algoritma melakukan perulangan hingga set awal kosong, mengumpulkan satu set *canopy*, masing-masing berisi satu atau lebih poin. Kelebihan penggunaan metode *canopy clustering* ini ialah jumlah data yang akan dibandingkan pada algoritma *clustering* utama berkurang dan *clustering* yang dihasilkan melalui proses ini lebih baik dibandingkan tanpa metode ini [10].



Gambar 2.4.1 Pembentukan *canopy* dengan dua nilai threshold T1 dan T2

2.5 Implementasi Algoritma K-Means++ dengan Canopy Clustering

Seperti dijelaskan sebelumnya, algoritma *canopy* terlebih dahulu membagi data menjadi beberapa bagian yang disebut dengan *canopy* dengan menggunakan dua nilai *threshold* T1 dan T2 dan juga menggunakan *cheap distance measurement.* Kedua kondisi diatas dintentukan oleh *user* dengan menilai aspek-aspek tertentu pada data yang digunakan. *Cheap distance measure* dapat digunakan melalui berbagai cara yaitu dengan mengurangi dimensi data ketika jumlah dimensi data banyak, menghitung tidak menghitung jarak antar data melainkan menghitung kesamaan antar dua buah data (*similarity measurement*), ataupun dengan menggunakan penghitungan jarak biasa. Meskipun cara ketiga dapat dikategorikan sebagai *actual distance* namun dengan menggunakan kedua nilai *threshold* maka data-data pada jarak < T2 terhadap pusat *canopy* tidak perlu dihitung terhadap jarak dengan pusat *canopy* lainnya. Setelah *canopy* terbentuk dan masing-masing data sudah menjadi anggota suatu *canopy*, jarak sebenarnya masing-masing data akan dihitung dengan data lain hanya jika kedua data tersebut termasuk kedalam *canopy* yang sama. Pada tahap ini metode *k-means++* akan diterapkan untuk mencari pusat-pusat *cluster* yang kan digunakan pada proses selanjutnya. Kemudian akan dihitung jarak masing-masing data dengan titik-titik pusat *cluster* tersebut jika berada dalam satu *canopy.*

Pada proses inisialisasi pusat *cluster*, algoritma *k-means++* hanya melakukan satu kali pencarian acak dan kemudian pencarian-pencarian selanjutkan ditentukan dengan menggunakan distribusi probabilitas seperti yang sudah dijelaskan pada bagian sebelumnya. Hal ini dapat menguntungkan proses karena mampu menghindari alokasi pusat *cluster* yang menumpuk pada beberapa *canopy* tertentu. Karena algoritma menggunakan distribusi untuk mengalokasikan pusat *cluster* maka pusat-pusat *cluster* akan terpilih merata pada *dataset.*

2.6 Validasi Cluster

Untuk menghitung nilai akurasi dari hasil *cluster* yang diperoleh sistem, dapat dipergunakan beberapa metode validasi agar dapat ditentukan seberapa dekat hasil yang dikeluarkan dengan hasil yang sebenarnya. Validasi pada *clustering* berguna untuk mengetahui seberapa banyak data yang berada pada *cluster* yang tepat. Validasi dilakukan dengan membandingkan kelompok tempat beradanya masing-masing data yang dihasilkan oleh sistem dengan data label kelas tempat data tersebut berada pada kelompok yang sebenarnya. Adapun rumus yang dipergunakan untuk menghitung akurasi yaitu :

$$\boldsymbol{ACC_i} = \frac{\boldsymbol{TP_i + TN_i}}{\boldsymbol{N}} \qquad (2.2)$$

N merupakan jumlah seluruh data yang ada pada *dataset*, sedangkan *TP* dan *TN* merupakan *True Positive* dan *True Negative. True Positive* yaitu sejumlah data observasi yang memiliki label kelas yang dan terhimpun kedalam suatu *cluster* tertentu. Sedangkan *True Negative* yaitu data-data selain data observasi yang label kelasnya tidak sama dengan label kelas data observasi dan berada pada *cluster* yang berbeda.

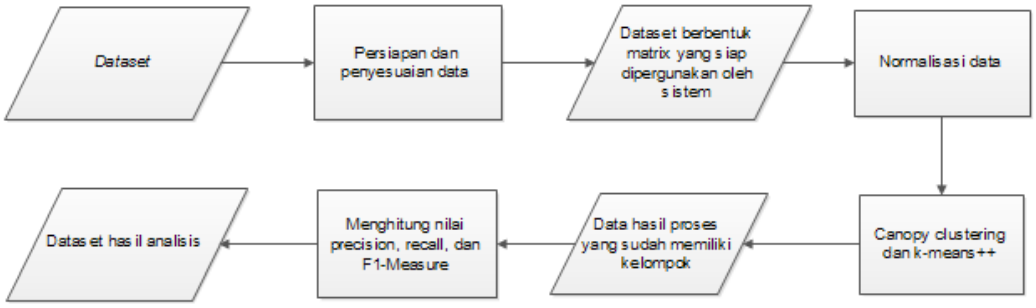
3. Perancangan dan Implementasi Sistem

3.1 Perancangan Sistem

Sistem ini dibangun untuk mengimplementasikan proses algoritma *canopy clustering* yang dikombinasikan dengan metode *k-means++* untuk proses pemilihan pusat *cluster* awal. Sistem ini dapat mengelompokkan data-data yang terdapat pada *dataset* kedalam kelompok tertentu setelah sebelumnya ditentukan keanggotaan data terhadap masing-masing *canopy* berdasarkan *similarity measure* maupun penghitungan jarak masing-masing data kepada pusat *canopy* yang terbentuk. Hasil keluaran sistem yaitu berupa data yang sudah dikelompokkan sesuai dengan *cluster* masing-masing. Hasil ini dipergunakan untuk melakukan analisis pengaruh parameter tertentu terhadap akurasi dan efisiensi yang diberikan sistem. Sebelum implementasi dilakukan, terdapat beberapa kebutuhan dari rancangan sistem ini, yaitu

a. Proses pemilihan *dataset* yang dapat dipergunakan oleh sistem secara optimal melihat dari keterbatasan perangkat keras maupun perangkat lunak dimana sistem ini dibangun.
b. *Dataset* yang dipergunakan diunduh secara gratis melalui situs-situs penyedia masing-masing *dataset.*
c. Melakukan persiapan dan penyesuaian terhadap *dataset* yang akan digunakan. Adapun persiapan dan penyesuaian *dataset* tersebut berbeda untuk masing-masing *dataset* dan akan dijelaskan pada bagian selanjutnya.
d. Melakukan proses normalisasi pada data dengan menggunakan proses *feature scaling* yaitu setiap atribut akan dikonversi sesuai dengan nilai minimun dan maksimum atribut tersebut sehingga jangkauan atribut yang dihasilkan berupa nilai 0 hingga 1.
e. Menentukan nilai-nilai parameter jumlah *cluster* (k), nilai *threshold* (T1 dan T2), serta menentukan kriteria *cheap distance measurement* yang dipergunakan.
f. Melakukan pembentukan kelompok dengan menggunakan *canopy clustering* dan *k-means++*.
g. Menghitung nilai akurasi untuk melakukan evaluasi hasil *cluster* yang didapatkan oleh sistem.

Berdasarkan perancangan sistem diatas, maka sistem dapat digambarkan melalui diagram sebagai berikut :

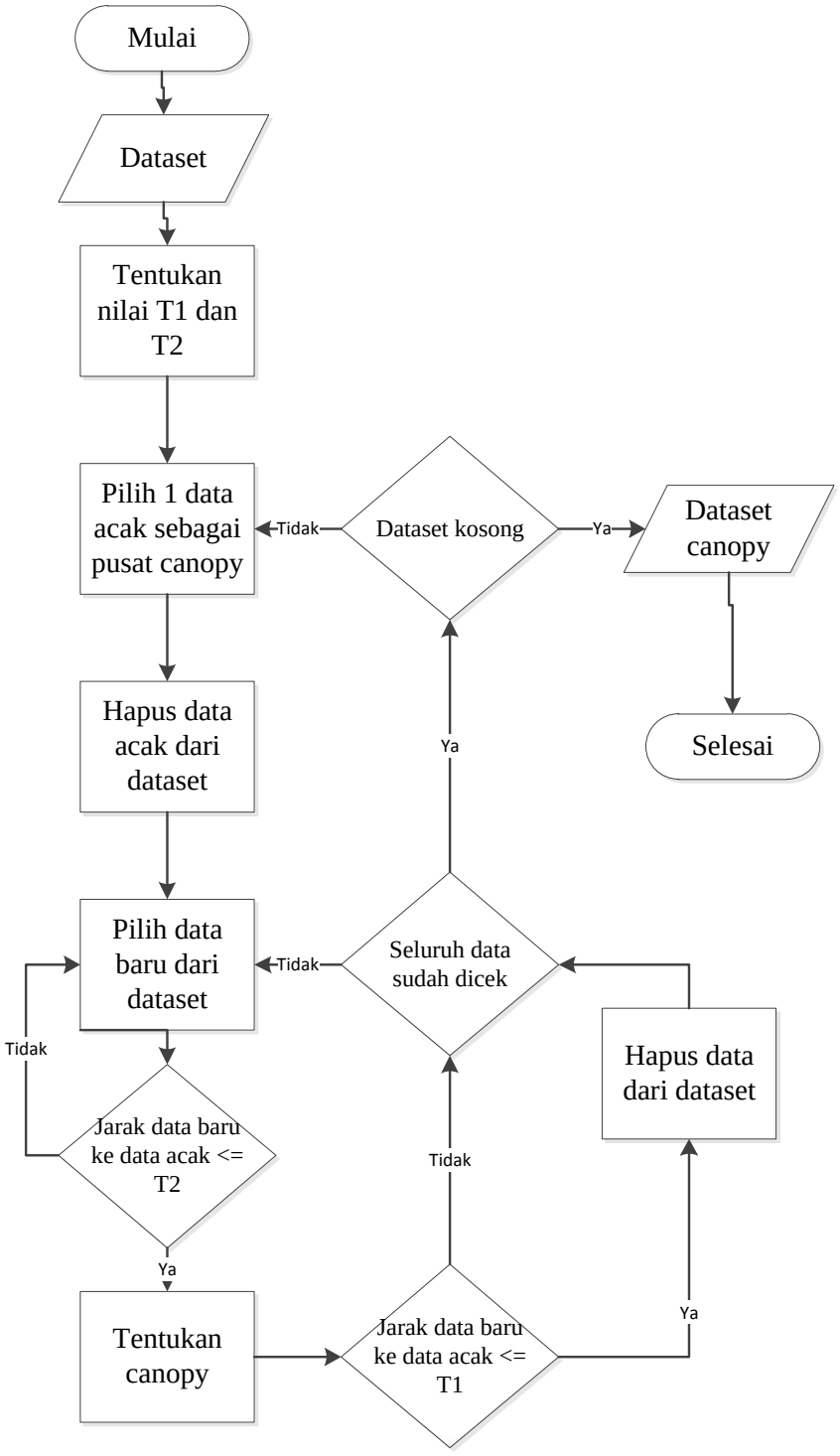


Gambar 3.1.1 Diagram diatas menunjukkan alur kerja sistem

Agar data dapat diproses oleh sistem sebelumnya akan dilakukan tahap persiapan dan penyesuaian *dataset* sehingga format *dataset* sesuai dengan inputan yang dibutuhkan sistem.

3.2 Gambaran Umum Sistem

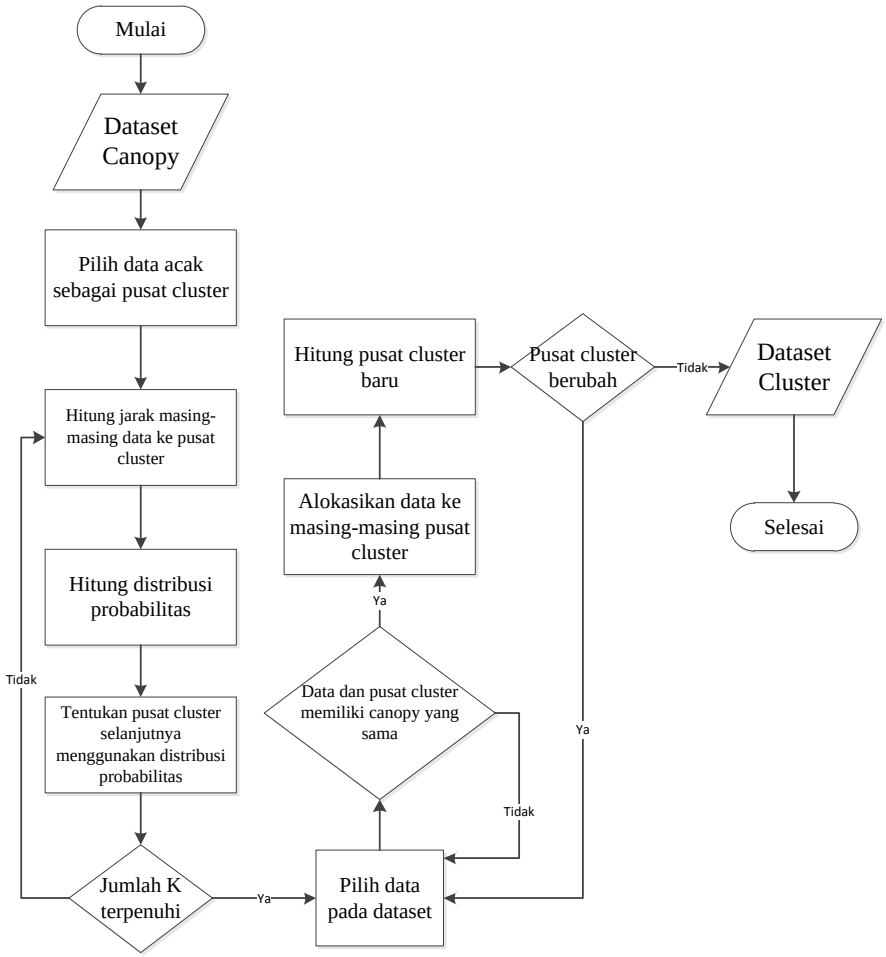
Tujuan dari penelitian ini adalah untuk menganalisa tingkat akurasi dan efisiensi yang dihasilkan pada algoritma *k-means++* dan mengkombinasikannya dengan menggunakan metode *canopy clustering.* Untuk menentukan akurasi dlakukan melalui perbandingan dari akurasi yang dihasilkan pada *k-means++* dan menggunakan metode *canopy clustering* dengan akurasi yang dihasilkan pada metode *k-means* tanpa menggunakan *canopy clustering.* Selain akurasi, analisis juga dilakukan dengan membandingkan waktu yang dibutuhkan sistem dalam memproses data menggunakan modifikasi algoritma *k-means* dan menggunakan metode *canopy clustering* dengan waktu yang dibutuhkan sistem dalam memproses data tanpa menggunakan metode *canopy clustering*. Sistem ini akan menjalani dua tahapan, yaitu proses pembuatan *canopy* dan proses *clustering* dengan menggunakan *k-means++*.



Gambar 3.2.1 Flowchart proses canopy clustering

Proses pembuatan *canopy* pada algoritma *canopy clustering* terdiri atas beberapa langkah, mula-mula akan ditentukan dua nilai batas jarak *canopy* T1 dan T2. Batas jarak T1 meliputi area luar dari sebuah *canopy*, sementara batas jarak T2 meliputi area dalam dari sebuah *canopy* oleh sebab itu maka penetapan batas jarak memiliki syarat T1 > T2. Untuk menentukan nilai T1 dan T2 akan dilakukan

analisis manual dengan melakukan beberapa percobaan untuk mendapatkan nilai T1 dan T2 yang optimum. Kemudian untuk mengisi sebuah *canopy* dipilih secara acak sebuah data yang berada dalam *data set* untuk dijadikan pusat dari *canopy* tersebut. Data-data yang memiliki jarak dengan pusat *canopy* kurang dari atau sama dengan T2 akan ditetapkan sebagai anggota dari *canopy* tersebut dan data-data ini tidak boleh menjadi anggota *canopy* berikutnya. Untuk data-data yang berjarak T2> dan T1<= akan ditetapkan sebagai anggota *canopy* tersebut, data-data ini masih dapat ditetapkan sebagai anggota dari *canopy* lainnya. Untuk pemilihan *canopy* berikutnya akan dipilih data lain secara random yang tidak termasuk ke dalam anggota *canopy* yang memiliki jarak dengan pusat masing-masing *canopy* T2<= kemudian data tersebut ditetapkan sebagai pusat *canopy* tersebut, proses ini akan diulang hingga seluruh data merupakan anggota dari satu atau lebih *canopy*.



Gambar 3.2.2 Flowchart proses *k-means++*

Setelah seluruh data sudah termasuk ke dalam *canopy*, langkah selanjutnya adalah penerapan algoritma *k-means++* dimulai dengan penentuan satu titik pusat awal secara acak, kemudian akan dicari jarak terdekat masing-masing data poin terhadap pusat *cluster* yang telah dipilih sebelumnya. Untuk mendapatkan titik pusat awal 2 hingga k ditentukan dengan menggunakan distribusi probabilitas terhadap jarak terdekat yang sebelumnya sudah didapatkan. Pusat *cluster*

digunakan untuk mengalokasikan data-data yang ada pada *data set* ke pusat *cluster* terdekat sehingga *cluster* awal terbentuk.

Tahap selanjutnya adalah menghitung nilai rata-rata dari masing-masing *cluster* yang terbentuk dan mengalokasikan kembali data-data pada rata-rata *cluster* yang baru. Algoritma akan terus berulang hingga tidak ada data yang berpindah *cluster,* atau hingga perubahan nilai rata-rata *cluster* telah mencapai batas yang ditentukan, atau hingga fungsi objektif sudah mencapai batas yang ditentukan. Setelah algoritma *canopy clustering* diimplementasikan dengan *k-means*++, Untuk melakukan evaluasi, hasil *cluster* dihitung nilai akurasinya dengan menggunakan rumus akurasi yang telah dijelaskan pada bagian sebelumnya. Nilai akurasi inilah yang menjadi acuan untuk menentukan tingkat akurasi yang dihasilkan *cluster.* Hasil akurasi algoritma *canopy clustering* dan *k-means*++ akan dibandingkan dengan hasil akurasi dari algortima *k-means* tanpa *canopy clustering.* Pengujian akan dilakukan dengan jumlah data yang bervariasi untuk mengetahui performa sistem pada jumlah data tertentu. Jika algoritma *k-means*++ dan *canopy clustering* dikombinasikan maka akan menghasilkan langkah sebagai berikut :

1. Tentukan batas jarak T1 dan T2, dan jumlah k cluster yang diinginkan.
2. Pilih salah satu data secara acak dari himpunan *data set* D sebagai pusat *canopy* Ch, alokasikan data pada himpunan Ch, dan hapus data dari himpunan *data set* D.
3. Hitung jarak seluruh data pada himpunan *data set* D terhadap data pada himpunan Ch jika (jarak <=T2) maka alokasikan data pada himpunan Ch dan hapus data dari himpunan *data set* D. Jika (T2<jarak<=T1) maka alokasikan data pada himpunan Ch tapi tidak menghapus data dari himpunan *data set* D.
4. Jika himpunan *data set* D belum kosong maka ulangi langkah 2
5. Tentukan nilai pusat *cluster* awal secara acak
6. Untuk seluruh data poin x, hitung D(x), jarak antara data poin x dengan pusat *cluster* terdekat yang sudah dipilih sebelumnya.
7. Pilih pusat *cluster* baru secara acak menggunakan distribusi probabilitas pada fungsi (2.1)
8. Ulangi langkah 6 dan 7 hingga seluruh pusat *cluster* telah terpilih.
9. Hitung jarak setiap data kepada seluruh pusat *cluster* kemudian alokasikan seluruh data ke dalam *cluster* dengan jarak terdekat terhadap pusat *cluster* kemudian ke langkah
10. Hitung pusat *cluster* baru dari data yang ada pada masing-masing *cluster.* Ulangi langkah 9 hinga terjadi konvergen.
11. Alokasikan data yang tidak memiliki pusat *cluster* di masing-masing *canopy* terhadap *cluster* yang sudah terbentuk.

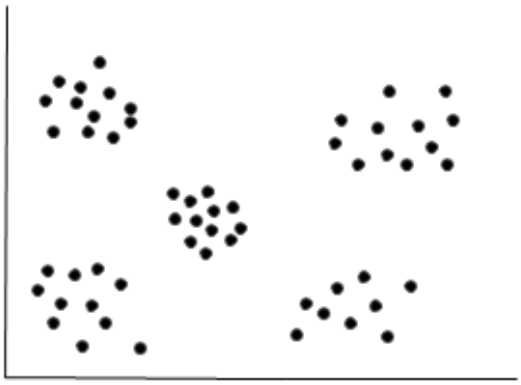
3.3 Dataset

Untuk melakukan analisis implementasi sistem digunakan dua jenis *dataset* yang berbeda dengan tujuan untuk mengetahui pengaruh fitur-fitur *dataset*

seperti jumlah data, dimensi data, dan juga struktur data tertentu terhadap kinerja pemrosesan sistem. Adapun kedua jenis *dataset* tersebut yaitu :

a. Dim1024 Dataset

Dim1024 merupakan *dataset* yang dikeluarkan oleh University of Eastern Finland, School of Computing dan dapat diakses melalui [https://cs.joensuu.fi/sipu/datasets/] secara gratis. *Dataset* ini merupakan salah satu dari 6 jenis *dataset* dengan kategori dimensi tinggi yang tersedia pada halaman tersebut. Fitur dari *dataset* ini yaitu memiliki 1024 atribut/dimensi, terdiri dari 1024 jumlah data, dan memiliki 16 *gaussian cluster* dengan perbedaan jarak masing-masing *cluster* yang tinggi sehingga memudahkan sistem dalam menemukan solusi optimal.



Gambar 3.3.1 Contoh *dataset* dengan 5 *cluster* yang tersebar dengan baik

Karena jenis data yang tersebar seperti diatas, maka diharapkan pembentukan *cluster* dapat menghasilkan tingkat akurasi yang tinggi dengan waktu yang sangat singkat meskipun *dataset* memiliki dimensi yang tinggi. Masing-masing atribut direpresentasikan dengan bilangan integer yang memiliki nilai terkecil yaitu 30 dan nilai terbesar 226.

	dim_1	dim_2	dim_3	dim_4	dim_5	dim_6	dim_7	dim_8	dim_9	dim_10	dim_11	dim_12
1	126	151	205	39	163	36	136	187	113	196	113	70
2	126	151	205	39	162	36	136	187	113	195	113	70
3	126	151	205	39	163	36	137	188	113	195	114	70
4	126	151	204	39	162	37	136	187	113	194	114	70
5	127	151	206	39	162	36	135	188	112	195	114	70
6	126	151	205	39	163	37	136	187	113	195	113	69
7	125	149	205	40	163	36	138	188	114	194	114	70
8	126	151	206	38	162	36	137	187	113	195	113	70
9	127	150	205	39	163	36	136	187	113	195	114	70
10	126	151	205	39	162	37	136	187	114	195	113	69
11	126	151	205	39	163	36	136	187	113	195	113	70
12	126	151	205	39	163	37	136	187	113	195	113	70
13	127	150	204	39	162	37	136	187	113	195	113	69
14	126	150	206	39	162	36	136	187	113	195	113	70

Gambar 3.3.2 Dataset dim1024 sebelum dilakukan proses normalisasi

Sebelum *dataset* dipergunakan kedalam sistem, terlebih dahulu dilakukan proses normalisasi terhadap *dataset.* Proses normalisasi bertujuan untuk memberikan bobot yang sama pada masing-masing atribut dikarenakan

penghitungan menggunakan jarak sehingga untuk menghindari terdapatnya atribut yang memiliki nilai sangat tinggi yang akan mempengaruhi penghitungan jarak tersebut. Dalam penelitian [15] dibuktikan bahwa normalisasi dapat meningkatkan efektifitas dari hasil yang dikeluarkan oleh sistem, terutama pada sistem yang menggunakan jarak *euclidean* seperti *k-means.* Proses normalisasi pada seluruh *dataset* pada penelitian ini menggunakan *feature scalling* dengan rumus :

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2.3)$$

dengan X merupakan data atribut yang akan dinormalisasi, X_{max} nilai tertinggi atribut tersebut, dan X_{min} nilai terkecil dari atribut tersebut. Hasil proses normalisasi merupakan *dataset* yang memiliki nilai integer dari 0 hingga 1.

	dim_1	dim_2	dim_3	dim_4	dim_5	dim_6	dim_7	dim_8	dim_9	dim_10	dim_11	dim_12
1	0.48913	0.732283	0.982857	0.005618	0.694915	0.017341	0.57377	0.865169	0.425414	0.85443	0.416216	0.046667
2	0.48913	0.724409	0.982857	0.011236	0.700565	0.017341	0.562842	0.859551	0.425414	0.841772	0.405405	0.053333
3	0.478261	0.724409	0.988571	0.011236	0.700565	0.017341	0.579235	0.870787	0.430939	0.848101	0.410811	0.06
4	0.483696	0.740157	0.988571	0.005618	0.700565	0.017341	0.57377	0.870787	0.425414	0.85443	0.410811	0.06
5	0.478261	0.740157	0.994286	0.005618	0.700565	0.023121	0.57377	0.865169	0.430939	0.860759	0.410811	0.06
6	0.483696	0.740157	0.982857	0.005618	0.694915	0.023121	0.568306	0.865169	0.425414	0.848101	0.410811	0.06
7	0.483696	0.740157	0.988571	0.005618	0.694915	0.017341	0.568306	0.865169	0.425414	0.85443	0.405405	0.06
8	0.483696	0.740157	0.988571	0.005618	0.694915	0.011561	0.562842	0.859551	0.430939	0.860759	0.4	0.053333
9	0.483696	0.740157	0.988571	0.011236	0.700565	0.034682	0.579235	0.870787	0.436464	0.841772	0.405405	0.06
10	0.483696	0.740157	0.988571	0.005618	0.700565	0.023121	0.568306	0.865169	0.425414	0.85443	0.410811	0.06
11	0.483696	0.740157	0.988571	0.005618	0.700565	0.023121	0.568306	0.865169	0.425414	0.85443	0.410811	0.06
12	0.483696	0.732283	0.988571	0.005618	0.700565	0.00578	0.568306	0.870787	0.425414	0.860759	0.405405	0.06
13	0.483696	0.732283	0.988571	0.005618	0.700565	0.023121	0.568306	0.865169	0.430939	0.860759	0.405405	0.06
14	0.483696	0.732283	1	0.011236	0.694915	0.028902	0.568306	0.859551	0.430939	0.848101	0.410811	0.066667

Gambar 3.3.3 Dataset dim1024 hasil normalisasi

Pada *dataset* ini terdapat 16 *cluster* yang pada masing-masing *cluster* terdiri dari 64 objek data. Pemilihan *cheap distance measure* pada *dataset* ini yaitu menggunakan nilai jarak euclidean seperti pada umumnya.

b. Iris Dataset

Dataset ini merupakan salah satu *dataset* yang paling sering digunakan dan paling sering diunduh pada halaman [https://archive.ics.uci.edu/ml/datasets/Iris] dikarenakan struktur data yang sederhana dengan ukuran data yang relatif kecil dan juga jumlah kelas yang digunakan hanya terdiri dari 3 kelas. Berdasarkan fitur tersebut *dataset* iris cocok digunakan untuk menguji suatu algoritma. *Dataset* memiliki 150 buah objek observasi dengan 4 jumlah atribut dan masing-masing kelas terdiri dari 50 anggota. Untuk pemilihan *cheap distance* menggunakan *euclidean distance measure.*

	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
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	3.6	1.4	0.2	1
6	5.4	3.9	1.7	0.4	1
7	4.6	3.4	1.4	0.3	1
8	5	3.4	1.5	0.2	1
9	4.4	2.9	1.4	0.2	1
10	4.9	3.1	1.5	0.1	1
11	5.4	3.7	1.5	0.2	1
12	4.8	3.4	1.6	0.2	1

Gambar 3.3.4 Dataset Iris

3.4 Analisa Kebutuhan Sistem

Analisis kebutuhan sistem diperlukan untuk mendapatkan suatu hasil analisis yang akan menjadi acuan dalam proses perancangan sistem pada pembangunan sebuah aplikasi. Kebutuhan aplikasi yang akan dirancang memiliki beberapa kebutuhan fungsionalisat sebagai berikut:

1. Perangkat lunak untuk membangun program, meliputi :
   a. Sistem operasi Windows 7 Home Premium 64-bit
   b. Microsoft Excel 2010.
   c. Matlab Versi R2009a yang untuk implemtasi sistem.
2. Sedangkan perangkat keras yang digunakan untuk merancang serta membangun sistem pakar pada kasus diagnosis medis umum ini dengan spesifikasi :
   a. Processor Intel® Core™ i7-2670QM CPU @ 2.20 GHz.
   b. Memori DDR2 4.00 GB RAM.
   c. Harddisk ATA 7200RPM 750 GB.
   d. Kartu grafis NVIDIA GEFORCE® GT 549M Cuda™ 2 GB.

4. Analisis Hasil Pengujian

4.1 Tujuan Pengujian

Pengujian yang dilakukan pada tugas akhir ini bertujuan untuk mengimplementasikan metode *canopy clustering* dan algoritma *k-means++*, mengukur akurasi dari hasil *clustering* yang diperoleh pada metode tersebut, mengukurwaktu yang diperoleh sistem untuk dapat mengeluarkan hasil, dan meneliti kecenderungan sistem terhadap parameter-parameter yang tersedia yaitu threshold (T1 dan T2) dan jumlah cluster yang dipilih.

4.2 Skenario Pengujian

Dari tujuan pengujian yang telah diuraikan diatas, maka pengujian yang dilakukan yaitu dengan mengukur akurasi hasil clustering yang diperoleh pada metode tersebut pada kedua *dataset* dengan menghitung nilai akurasi algoritma *k-means* untuk dibandingkan dengan algoritma *canopy* dan *k-means++*, melihat perbedaan waktu yang dibutuhkan sistem dalam menyelesaikan proses *clustering* pada masing-masing *dataset* dan membandingkan waktu antara metode *k-means* dengan metode *canopy* dan *k-means++*. Meneliti perilaku sistem untuk masing-masing *dataset* terhadap perubahan yang terjadi pada parameter-parameter yang tersedia. Setiap pengujian akan dilakukan sebanyak 10 kali dengan parameter yang sama dan dihitung nilai rata-ratanya.

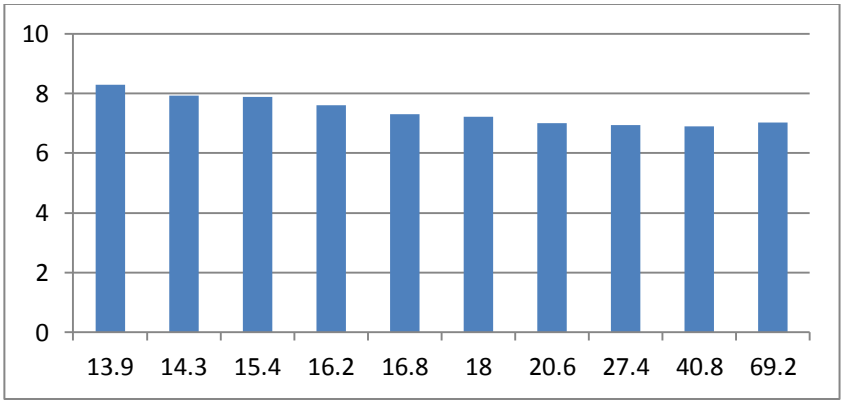
4.2.1 Pengujian Nilai Threshold (T1 dan T2) Pada *Dataset* dim1024

Tabel 4.2.1.1 Pengujian nilai *threshold* terhadap *dataset* dim1024

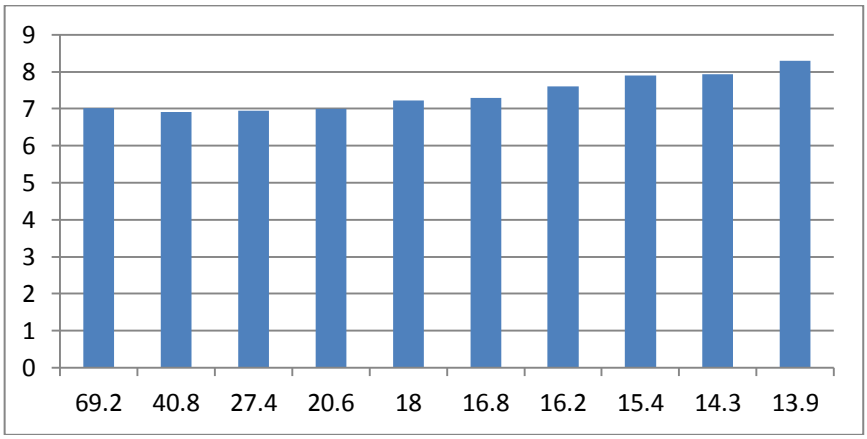
T1	T2	Canopy K-Means++			K-Means	
		Canopy	Akurasi	Waktu	Akurasi	Waktu
1	0.1	69.2	0.84765	7.022309	0.71386	11.4748
2	0.2	40.8	0.933773	6.906916	0.71386	11.4748
3	0.3	27.4	0.971491	6.950401	0.71386	11.4748
4	0.4	20.6	0.983818	7.003757	0.71386	11.4748
5	0.5	18	0.965298	7.219547	0.71386	11.4748
6	0.6	16.8	0.975255	7.293698	0.71386	11.4748
7	0.7	16.2	0.976958	7.609686	0.71386	11.4748
8	0.8	15.4	0.97106	7.888823	0.71386	11.4748
9	0.9	14.3	0.977339	7.935158	0.71386	11.4748
10	1	13.9	0.980657	8.290166	0.71386	11.4748

Tabel diatas merupakan hasil *running* algoritma *canopy* dan *k-means++* sebanyak 10 kali sesuai dengan nilai *threshold* yang tertera pada bagian kiri dan kemudian dihitung rata-rata setiap *running*. Dari tabel tersebut dapat diketahui pengaruh yang terjadi pada hasil *running* masing-masing kombinasi *threshold* terhadap hasil yang diperoleh. *Dataset* ini memiliki akurasi yang sangat tinggi bahkan ketika terdapat lebih dari 60 *canopy* yaitu pada *running* pertama dan akurasi pada *running-running* selanjutnya sangat konsisten yaitu antara 0.93 hingga 0.98. Hal tersebut dapat diartikan bahwa pengaruh *threshold* terhadap nilai akurasi sangat kecil pada *dataset* ini. Kemungkinan yang dapat terjadi yaitu karena *dataset* telah tersebar dengan baik pada masing-masing titik poin, penggunaan *k-means++* dengan distribusi probabilitas semakin mempengaruhi hasil *cluster* yang sangat baik. Sehingga pada saat inisialisasi *centroid*, titik-titik pusat yang terbentuk telah menyebar pada poin-poin dimana kemungkinan pusat-pusat *cluster* optimum berada. Sedangkan jika dibandingkan dengan akurasi algoritma *k-means*terdapat perbedaan yang sangt signifikan, yaitu lebih dari 20 persen ketika algoritma *canopy k-means++* mendapatkan akurasi maksimumnya.

Untuk waktu yang ditempuh sistem dalam menjalani program, terdapat perbedaan selama 3 hingga 4 detik mengindikasi bahwa performa *canopy k-means++* lebih baik dalam hal menemukan hasil *cluster* optimal dengan waktu yang lebih cepat. Terdapat juga indikasi bahwa nilai *threshold* mempengaruhi waktu yang ditempuh sistem untuk menyelesaikan proses. Jika dilihat dari waktu pemrosesan, semakin besar nilai *threshold* maka waktu yang dibutuhkan juga semakin besar. Hal ini dapat dikaitkan dengan prinsip utama *canopy clustering*, yaitu memisahkan objek-objek data kedalam himpunan-himpunan kecil yang disebut dengan *canopy* agar penghitungan jarak hanya dilakukan pada data poin yang berada pada *canopy* yang sama sehingga semakin kecil diameter *canopy* maka penghitungan jarak akan semakin sedikit dan waktu yang ditempuh sistem lebih sedikit. Sedangkan pada tabel akurasi *canopy k-means++* terdapat indikasi yang sebaliknya yaitu ketika *canopy* terbentuk semakin banyak, maka akurasi akan semakin menurun. Hal ini disebabkan oleh banyaknya *canopy-canopy* yang tidak memiliki pusat *cluster* sehingga objek-objek yang berada pada *canopy* tersebut baru akan diukur jaraknya pada pusat *cluster* ketika proses *canopy k-means* selesai dijalankan dan tidak dilakukan lagi penentuan pusat *cluster* baru.



Gambar 4.2.1.1 Pengaruh jumlah *canopy* terhadap akurasi



Gambar 4.2.1.2 Pengaruh jumlah *canopy* terhadap waktu

4.2.2 Pengujian Nilai Threshold (T1 dan T2) Pada *Dataset* Iris

Tabel 4.2.2.1 Pengujian nilai *threshold* terhadap *dataset* iris

T1	T2	Canopy K-Means++			K-Means	
		Canopy	Akurasi	Waktu	Akurasi	Waktu
1	0.1	61.6	0.64268	0.787437	0.8164	1.53163
10	1	10.7	0.770326	0.791172	0.8164	1.53163
20	2	6.1	0.768877	0.778332	0.8164	1.53163
30	3	4.8	0.822215	0.771484	0.8164	1.53163
40	4	4	0.860653	0.76089	0.8164	1.53163
50	5	3.1	0.877611	0.770263	0.8164	1.53163
60	6	3	0.827074	0.782849	0.8164	1.53163
70	7	2.8	0.828505	0.770608	0.8164	1.53163
80	8	2.5	0.869655	0.774157	0.8164	1.53163
90	9	2.7	0.868395	0.765987	0.8164	1.53163

Pada *dataset* ini algoritma *k-means* menghasilkan akurasi yang lebih tinggi dibandingkan dengan pada *dataset* sebelumnya. Salah satu faktor yang mempengaruhi yaitu jumlah objek pada *dataset* ini jauh lebih sedikit jika dibandingkan dengan *dataset* dim 1024 yang memiliki 1024 objrk dan 1024 atribut *dataset* iris ini hanya memiliki 150 objek dan 4 atribut sehingga algoritma dapat dengan mudah menemukan solusi dari suatu permasalahan. Pada *dataset* iris ini ditemukan bahwa waktu pemrosesan memiliki nilai yang stabil dengan artian bahwa hanya sedikit atau bahkan tidak ada faktor yang mempengaruhi nilai tersebut. Sedangkan ditemukan kembali bahwa jumlah *canopy* yang tinggi mengurangi jumlah akurasi yang dihasilkan. Meskipun hanya terdapat sedikit

perbedaan antara akurasi dan waktu yang dihasilkan kedua metode, algoritma *canopy k-means++* masih memberikan akurasi yang lebih tinggi dengan waktu proses yang lebih singkat.

Dari hasil analisis kedua *dataset* terhadap nilai *threshold* didapatkan bahwa nilai *threshold* digunakan untuk menentukan ukuran dari suatu *canopy*, jika nilai *threshold* besar maka ukuran *canopy* akan mengikuti ukuran *threshold* tersebut. Jika pada suatu proses ditentukan ukuran *threshold* yang besar maka hanya akan ada sedikit *canopy* yang terbentuk. Jika hanya ada sedikit *canopy* yang terbentuk dari sebuah proses maka masing-masing *canopy* memiliki anggota yang banyak. Hal ini dapat menyebabkan lamanya waktu yang dipergunakan oleh sistem dalam menangani suatu masalah karena terdapat banyak perhitungan yang harus dicari. Namun sebaliknya ketika nilai *threshold* kecil maka akan ada banyak *canopy* yang terbentuk untuk memenuhi jarah di dalam *dataset.* Jika terdapat banyak *canopy* yang dihasilkan maka kemungkinan besar bahwa terdapat banyak *canopy* yang tidak memiliki *cluster* sehingga penghitungan jarak *cluster* tidak optimal dikarenakan banyaknya objek yang tidak disertakan dalam penghitungan. Jadi secara garis besar, pengaruh nilai *threshold* pada suatu proses yaitu dapat mengurangi akurasi yang dihasilkan oleh sistem dengan jumlah *threshold* yang tinggi, dan juga dapat mengurangi efisiensi kinerja sistem ketika *canopy* yang dibangun berjumlah banyak. Oleh sebab itu *threshold* yang dipilih harus mampu mengoptimalkan kinerja sistem. Pada kedua *dataset* diatas, *threshold* yang dipilih adalah yang menghasilkan akurasi tinggi tanpa memakan banyak waktu. Pada *dataset* dim1024 nilai *threshold* yang akan digunakan yaitu T1=4 dan T2=0.4 karena menghasilkan akurasi yang paling tinggi dengan waktu yang singkat. Sedangkan pada *dataset* iris nilai *threshold* yang digunakan yaitu T1= 50 dan T2=5 dengan kriteria pemilihan yang sama seperti *dataset* dim1024.

4.2.2 **Pengujian Jumlah Cluster (K) Pada *Dataset* dim1024 dan Iris**

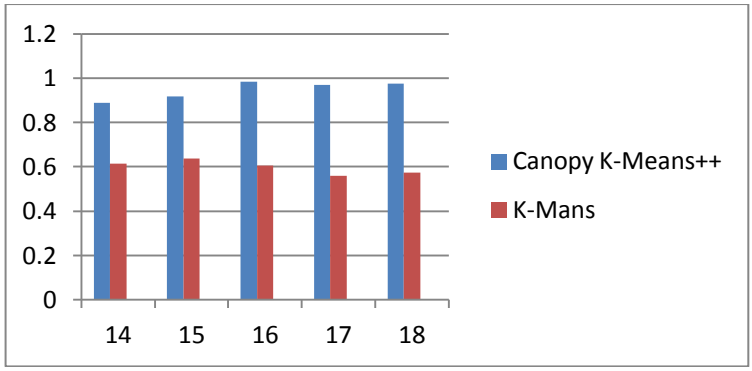
Tabel 4.2.3.1 Pengujian jumlah *cluster* terhadap *dataset* dim1024

K	Canopy K-Means++		K-Means	
	Akurasi	Waktu	Akurasi	Waktu
14	0.888895512	7.327577728	0.612983631	15.4531699
15	0.916316014	7.901012814	0.636802455	16.42304416
16	0.983817644	7.003756662	0.606884998	16.17675329
17	0.970235119	7.679918055	0.559640067	17.00039209
18	0.975245762	7.33045854	0.574125279	17.97500327

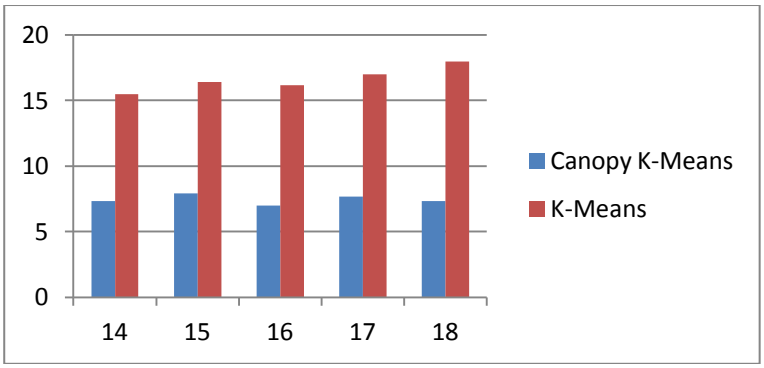
Tabel 4.2.3.2 Pengujian jumlah *cluster* terhadap *dataset* iris

K	Canopy K-Means++		K-Means	
	Akurasi	Waktu	Akurasi	Waktu
2	0.76650661	0.773499416	0.735393083	1.507431873
3	0.877611	0.770263	0.822935177	1.539745815
4	0.778759428	0.799065978	0.596625327	1.556788619
5	0.692877364	0.782610961	0.385649718	1.589451968
6	0.644529581	0.796310654	0.292181893	1.586913612

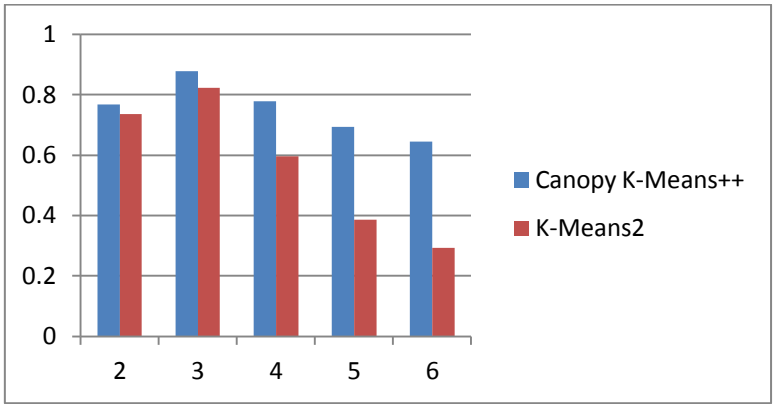
Kedua tabel diatas menunjukkan pengujian masing-masing *dataset* terhadap variasi nilai jumlah *cluster*. Untuk *dataset* dim1024, nilai K yang diuji yaitu K=14, K=15, K=16, K=17, dan K=18. Sedangkan untuk *dataset* iris nilai K yang diuji ialah K=2, K=3, K=4, K=5, dan K=6. Pada tabel tersebut dapat diketahui bahwa jumlah k atau jumlah *cluster* tidak terlalu mempengaruhi waktu yang dibutuhkan sistem untuk menjalankan proses. Nilai k berpengaruh terhadap akurasi yang dihasilkan sistem karena jika nilai k yang dipilih mendekati nilai k yang sebenarnya maka akurasi akan semakin tinggi. Akurasi tertinggi didapatkan ketika jumlah k yang dipilih sama dengan jumlah k yang sebenarnya.



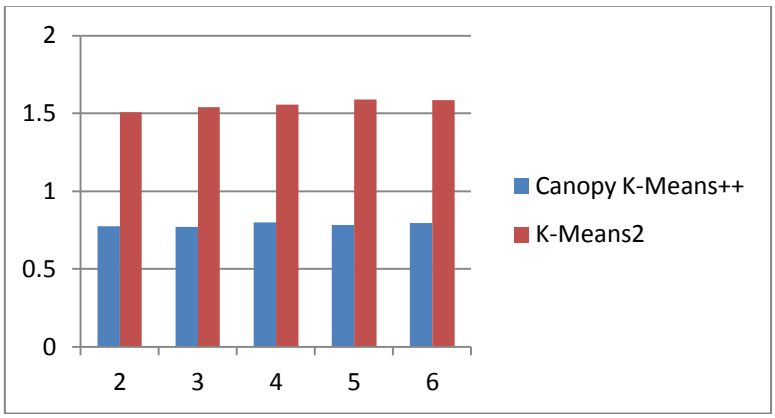
Gambar 4.2.3.1 Pengaruh jumlah k pada akurasi *dataset* dim1024



Gambar 4.2.3.2 Pengaruh jumlah k pada waktu *dataset* dim1024



Gambar 4.2.3.3 Pengaruh jumlah k pada akurasi *dataset* iris



Gambar 4.2.3.4 Pengaruh jumlah k pada waktu *dataset* iris

4.3 Hasil Analisis

Dari kedua pengujian yang telah dilakukan terhadap masing-masing *dataset* diatas, didapatkan bahwa :

1. Karena *dataset* dim1024 tersebar dengan baik, penggunaan *k-means++* mempengaruhi konsistensi nilai akurasi yang didapat.
2. Untuk *dataset* dim1024 akurasi yang dihasilkan dengan menggunakan metode *canopy k-means++* sangat akurat yaitu 0.9838 ketika dipilih T1 = 4, T2 = 0.4, dan K = 16. Jika dibandingkan dengan akurasi yang didapat pada metode *k-means* yaitu 0.7138 dengan jumlah K yang sama. Akurasi pada metode *canopy k-means++*memiliki perbedaan 0.27 ataupun 27% lebih akurat dibandingkan dengan menggunakan metode *k-means.*
3. Untuk jumlah waktu yang digunakan, *dataset* dim1024 dengan menggunakan *canopy k-means++* hanya membutuhkan waktu 6.9 hingga 8.2 detik. Sedangkan pada metode *k-means* waktu yang digunakan yaitu 11.4 detik. Pada saat akurasi tertinggi 0.9838 waktu yang digunakan yaitu 7 detik dibandingkan dengan waktu pada metode *k-means*, metode *canopy k-means++* 4.4 detik lebih cepat.

4. Untuk akurasi pada *dataset* iris, pada saat menggunakan metode *canopy k-means++*, akurasi tertinggi yang dihasilkan yaitu 0.8776 pada saat T1 = 50, T2 = 5, dan K = 3. Sedangkan akurasi yang dihasilkan pada metode *k-means* yaitu 0.8164. Artinya pada *dataset* iris ini metode *canopy k-means++* menghasilkan akurasi yang lebih baik dibandingkan dengan metode *k-means.*
5. Untuk jumlah waktu yang digunakan, karena *dataset* iris berukuran kecil, maka waktu yang dihasilkan kedua metode relatif kecil. Pada *canopy k-means ++* waktu yang diperoleh yaitu 0.7 jika dibandingkan dengan menggunakan metode *k-means* yang memakan waktu 1.5 detik. Artinya metode *canopy k-means++* lebih cepat 0.8 detik dibandingkan dengan metode *k-means.*
6. Pada analisis pengaruh nilai *threshold*, didapatkan bahwa nilai *threshold* digunakan untuk menentukan ukuran *canopy*. Sedangkan ukuran *canopy* menentukan banyaknya data yang terdapat dalam sebuah *canopy.* Jika *canopy* berukuran besar maka data yang terdapat di dalamnya lebih banyak sehingga penghitungan yang dilakukan akan lebih lama. Sebaliknya jika ukuran *canopy* kecil maka data yang ada di dalamnya hanya sedikit maka penghitungan akan lebih cepat, akan tetapi dapat terjadi *canopy-canopy* yang tidak memiliki *centroid* di dalamnya sehingga *canopy-canopy* tersebut baru akan ditentukan *cluster* masing-masing data ketika proses *k-means* selesai dan pusat *cluster* tidak lagi berubah. Hal ini berdampak kurangnya akurasi yang dihasilkan.
7. Dari hasil analisis kedua *dataset* terhadap nilai jumlah K atau *cluster*, nilai K hanya mempengaruhi hasil akurasi yang didapatkan. Semakin dekat nilai K yang dipilih dengan nilai K yang sebenarnya maka akurasi akan semakin tinggi.

5. Penutup

5.1 Kesimpulan

Dari hasil implementasi, pengujian, dan analisis yang dilakukan, maka dapat diperoleh kesimpulan-kesimpulan sebagai berikut :

1. Algoritma *K-means++* dapat dikombinasikan dengan metode *canopy clustering* dan mampu meningkatkan akurasi hasil *cluster* yang dibentuk pada *dataset* tertentu.
2. Pengaruh *threshold* baru dapat dirasakan ketika jumlah data yang digunakan banyak. Untuk data dengan ukuran kecil, karena *threshold* mempengaruhi waktu pemrosesan, maka dampaknya tidak terasa.
3. Untuk data yang ukurannya lebih besar, dampak nilai *threshold* yaitu semakin besar nilai *threshold* yang digunakan, akurasi yang dihasilkan semakin berkurang, semakin kecil nilai *threshold* yang digunakan, maka penghitungan akan semakin lama. Oleh karena itu penentuan nilai *threshold* yang optimal harus mampu memberikan akurasi tinggi dengan waktu yang singkat.
4. Dari kedua *dataset* yang digunakan, didapatkan bahwa penggunaan metode *canopy k-means++* mampu meningkatkan akurasi dan efisiensi dari algoritma *k-means* pada kedua *dataset.*

5.2 Saran

Adapun yang disarankan untuk pengembangan selanjutnya terhadap tugas akhir ini, ialah :

1. Melakukan pengujian dengan *dataset* dengan jumlah yang lebih besar agar dapat diukur kinerja dari kombinasi kedua metode.
2. Melakukan pengujian dengan *dataset* yang memiliki dimensi lebih tinggi
3. Penentuan nilai *threshold* yang tepat akan menghasilkan akurasi dan waktu yang lebih baik.

Daftar Pustaka

[1] De, T., et al. (2013). Clustering Large Number of Extragalactic Spectra of Galaxies and Quasars Through Canopies. Communications in Statistics. Theory and Methods, hal-00861873, version 1 - 13 Sep 2013.

[2] Dwi Nugraha, A. (2011). Teks Clustering Menggunakan Algoritma Canopy Clustering. Tugas Akhir, Telkom University, 2011. [online], (https://openlibrary.telkomuniversity.ac.id/home/catalog/id/95192/slug/pengelompokan-teks-menggunakan-algoritma-canopy-clustering.html, diakses tanggal 22 Agustus 2016)

[3] Kumar, A., et al. (2014). Canopy Clustering: A Review on Pre-Clustering Approach to K-Means Clustering. International Journal of Innovations & Advanvement in Computer Science, Vol. 3, Issue 5, June 2014.

[4] MacCallum, A., et al. (2005). Efficient Clustering of High Dimensional Data Sets with Application to Reference Matching

[5] Sabena, S., et al. (2011). Image Retrieval using Canopy and Improved K mean Clustering. International Conference on Emerging Technology Trends (ICETT) 2011, Proceedings published by International Journal of Computer Applications® (IJCA).

[6] Prasetyo, E. 2012. *Data Mining Konsep dan Aplikasi Menggunakan Matlab.* Yogyakarta : C. V. Andi Offset

[7] Fahim A. M., et al. (2006). An Efficient Enhanced K-Means Clustering Algorithm. *Journal of Zhejiang University Science A, ISSN 1009-3095 (Print); ISSN 1862-1775 (Online).*

[8] Virmani, D., et al. Normalization based K-Means Clustering Algorithm. *Departement of Computer Science, Baghwan Parshuram Institute of Thecnology, New Delhi.*

[9] Abdul Nazeer, K. A., & Sebastian P. M. (2009). Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. *Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009, London, U.K.*

[10] Christen, P. (2007). Towards Parameter-free Blocking for Scalable Record Linkage. *Joint Computer Science Technical Report Series, The Australian National University* .

[11] Geetha, S., et al. (2009). Improved K-Means Algorithm for Capacitated Clustering Problem. *PSG College of Technology, Tamil Nadu, India.*

[12] Kim, D. M., et al. (Unknown). Personalized Music Recommendation System Using Improved K-means Clustering Algorithm. *Department of Electrical and Electronic Engineering SungkyunKwan University Suwon, Korea.*

[13] Liu, G., et al. (2014). An improved K-Means Algorithm Based on Association Rules. *International Journal of Computer Theory and Engineering, Vol. 6, No. 2, April 2014.*

[14] Usman, G., et al. (2013). Improved K-Means Clustering Algorithm by Getting Initial Cenroids. *World Applied Sciences Journal 27 (4): 543-551, 2013, ISSN 1818-4952*

[15] Venkateswarlu, B., & Prasad Raju, G. S. V. (2013). Mine Blood Donors Information through Improved K-Means Clustering. *International Journal of Computational Science and Information Technology (IJCSITY) Vol.1, No.3, August 2013.*

Lampiran Data Pengujian

Data hasil pengujian 4.2.1

K-Means		
K	Akurasi	Waktu
16	0.627929688	16.7166258
16	0.693033854	15.35394088
16	0.689453125	15.31392329
16	0.574951172	16.42126774
16	0.652064732	14.95002034
16	0.639648438	14.95192738
16	0.535047743	15.57278768
16	0.585693359	17.08957694
16	0.705729167	15.68697985
16	0.638671875	15.43892329
Avg	0.634222315	15.74959732

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
1	0.1	16	0.917458	69	6.988688	2	0.2	16	0.970064344	38	6.633815412
1	0.1	16	0.894827	68	7.205525	2	0.2	16	0.951856351	41	6.696566855
1	0.1	16	0.83753	70	6.8084	2	0.2	16	0.937390024	42	6.64145385
1	0.1	16	0.916393	70	6.982789	2	0.2	16	0.946157446	42	6.841158837
1	0.1	16	0.794207	69	7.092945	2	0.2	16	0.942960182	43	6.798908115
1	0.1	16	0.828301	70	6.934842	2	0.2	16	0.909379392	40	6.987029957
1	0.1	16	0.819936	72	7.270753	2	0.2	16	0.903968811	42	7.012494215
1	0.1	16	0.863773	69	6.992302	2	0.2	16	0.924088589	40	7.542669171
1	0.1	16	0.840343	69	6.990022	2	0.2	16	0.921861765	41	6.815558361
1	0.1	16	0.763728	66	6.956829	2	0.2	16	0.929998405	39	7.099505419

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
3	0.3	16	0.993158	27	6.650719	4	0.4	16	0.99609542	21	6.678885648
3	0.3	16	0.957048	26	6.707722	4	0.4	16	0.996094346	22	7.548276477
3	0.3	16	0.971797	27	6.747315	4	0.4	16	0.966593492	21	6.991095953
3	0.3	16	0.990165	28	7.199063	4	0.4	16	0.998046994	21	6.86822365
3	0.3	16	0.977675	29	7.058365	4	0.4	16	0.976439995	18	6.798697258
3	0.3	16	0.948314	26	7.141535	4	0.4	16	0.978871922	20	6.94991631
3	0.3	16	0.952736	26	6.860612	4	0.4	16	0.995117602	21	7.037757414
3	0.3	16	0.937713	28	6.930888	4	0.4	16	0.976335463	21	7.076646551

3	0.3	16	0.991199	28	6.959688	4	0.4	16	0.998046994	20	7.041171712
3	0.3	16	0.995103	29	7.248104	4	0.4	16	0.956534215	21	7.046895643

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
5	0.5	16	0.960542	19	6.685174	6	0.6	16	0.999023497	16	6.935007035
5	0.5	16	0.996094	18	6.868791	6	0.6	16	1	16	7.097756051
5	0.5	16	0.947458	18	6.990102	6	0.6	16	0.937240615	16	7.432024713
5	0.5	16	0.991255	17	7.03724	6	0.6	16	0.954561015	17	7.306834382
5	0.5	16	0.998047	19	7.781864	6	0.6	16	0.994142772	17	7.348539302
5	0.5	16	0.958552	19	7.599956	6	0.6	16	0.988298448	17	7.22835496
5	0.5	16	0.95172	18	7.131487	6	0.6	16	0.954949377	16	7.177752991
5	0.5	16	0.949087	18	7.66687	6	0.6	16	0.987357383	19	7.545120151
5	0.5	16	0.990294	17	7.245698	6	0.6	16	0.951529719	17	7.434185532
5	0.5	16	0.909929	17	7.188285	6	0.6	16	0.985444253	17	7.431409402

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
7	0.7	16	0.935545	17	8.482146	8	0.8	16	0.999023497	15	7.196010789
7	0.7	16	0.996095	15	7.124589	8	0.8	16	0.999023497	14	7.327161099
7	0.7	16	0.990294	16	7.337372	8	0.8	16	0.966887031	15	8.274339532
7	0.7	16	0.97693	16	7.304325	8	0.8	16	0.922713956	15	7.682101685
7	0.7	16	0.990214	17	7.358352	8	0.8	16	0.992218137	15	7.486108623
7	0.7	16	0.958983	17	8.100155	8	0.8	16	1	15	7.806859106
7	0.7	16	0.993168	17	7.649658	8	0.8	16	0.940356972	16	9.155033607
7	0.7	16	0.954949	16	7.721496	8	0.8	16	0.981863352	17	8.183479254
7	0.7	16	0.99707	16	7.516766	8	0.8	16	0.999023497	15	7.888696847
7	0.7	16	0.976335	15	7.502001	8	0.8	16	0.90949277	17	7.888438407

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
9	0.9	16	1	14	7.938436	10	1	16	0.958767259	15	8.177140013
9	0.9	16	0.9818	14	7.622388	10	1	16	0.955404182	15	8.790348343
9	0.9	16	0.972251	14	7.609819	10	1	16	0.996094704	13	7.623446095
9	0.9	16	0.991237	14	7.628419	10	1	16	0.950861121	14	7.876446144
9	0.9	16	0.991242	15	7.863032	10	1	16	1	14	8.269799107
9	0.9	16	0.944024	15	8.731011	10	1	16	0.997056112	13	7.963676601
9	0.9	16	0.999023	14	7.821638	10	1	16	0.997070849	13	9.447211088
9	0.9	16	0.976797	15	7.844261	10	1	16	1	14	8.371176583
9	0.9	16	0.974726	14	8.215641	10	1	16	0.996095051	14	8.111785978

9	0.9	16	0.942293	14	8.07693	10	1	16	0.955217154	14	8.270625742

Data pengujian 4.2.2

K-Means		
K	Akurasi	Waktu
3	0.838123191	1.585257171
3	0.825236324	1.544895766
3	0.803648806	1.561417258
3	0.825236324	1.511516902
3	0.840967342	1.538339137
3	0.844789858	1.49744639
3	0.718377914	1.540340879
3	0.774153182	1.507257776
3	0.855317624	1.501398094
3	0.838123191	1.528445181
Avg	0.816397376	1.531631455

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
1	0.1	3	0.773001	59	0.861758	10	1	3	0.752417324	11	0.819894926
1	0.1	3	0.793742	60	0.831653	10	1	3	0.808828224	12	0.754900561
1	0.1	3	0.781736	61	0.782151	10	1	3	0.894020397	11	0.834009755
1	0.1	3	0.70671	61	0.76211	10	1	3	0.687011723	9	0.75885973
1	0.1	3	0.519731	62	0.769546	10	1	3	0.670837011	11	0.775738093
1	0.1	3	0.673652	63	0.790465	10	1	3	0.774652015	9	0.784630011
1	0.1	3	0.53245	63	0.78411	10	1	3	0.723689021	13	0.776353404
1	0.1	3	0.60052	63	0.754583	10	1	3	0.873977551	10	0.841086063
1	0.1	3	0.516417	62	0.764889	10	1	3	0.839250661	10	0.803753163
1	0.1	3	0.528834	62	0.773102	10	1	3	0.678579	11	0.762490483

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
20	2	3	0.668216	5	0.821156	30	3	3	0.78953353	4	0.832376546
20	2	3	0.863749	6	0.809202	30	3	3	0.88329756	5	0.762103757
20	2	3	0.869867	6	0.764997	30	3	3	0.869391921	5	0.761209013
20	2	3	0.777324	6	0.793382	30	3	3	0.671450576	6	0.765235824
20	2	3	0.784704	6	0.779518	30	3	3	0.875085441	4	0.774396445
20	2	3	0.788067	7	0.774225	30	3	3	0.763780344	5	0.748406443
20	2	3	0.630185	6	0.760351	30	3	3	0.855198284	4	0.775840256
20	2	3	0.836546	6	0.786092	30	3	3	0.795181985	4	0.75453436

20	2	3	0.75099	6	0.725589	30	3	3	0.894323633	5	0.764035991
20	2	3	0.719123	7	0.768812	30	3	3	0.824911106	6	0.776699545

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
40	4	3	0.894892	4	0.817064	50	5	3	0.888054796	3	0.771728543
40	4	3	0.816703	3	0.732666	50	5	3	0.90230179	3	0.760240564
40	4	3	0.888055	4	0.770718	50	5	3	0.861911486	3	0.754978933
40	4	3	0.92	4	0.748629	50	5	3	0.90230179	3	0.759679833
40	4	3	0.888615	4	0.771423	50	5	3	0.888054796	4	0.752284441
40	4	3	0.894411	5	0.742911	50	5	3	0.880192308	3	0.801854516
40	4	3	0.883922	4	0.776784	50	5	3	0.888054796	3	0.798909981
40	4	3	0.888615	4	0.749666	50	5	3	0.8084434	3	0.762282425
40	4	3	0.877051	4	0.743486	50	5	3	0.894891775	3	0.767963904
40	4	3	0.654264	4	0.755557	50	5	3	0.861899637	3	0.772711454

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
60	6	3	0.674843	3	0.803978	70	7	3	0.888054796	2	0.801401547
60	6	3	0.888055	3	0.72901	70	7	3	0.894891775	3	0.73937924
60	6	3	0.888055	3	0.749987	70	7	3	0.674842651	3	0.815029352
60	6	3	0.888615	3	0.797975	70	7	3	0.888054796	3	0.788369925
60	6	3	0.885727	3	0.798196	70	7	3	0.894891775	3	0.758273808
60	6	3	0.785538	3	0.750578	70	7	3	0.674842651	3	0.767561783
60	6	3	0.888055	3	0.771729	70	7	3	0.90230179	3	0.771623581
60	6	3	0.822748	3	0.788149	70	7	3	0.671055344	3	0.757940729
60	6	3	0.66105	3	0.799625	70	7	3	0.894891775	3	0.754098185
60	6	3	0.888055	3	0.839263	70	7	3	0.90122482	2	0.752405264

Canopy K-Means++						Canopy K-Means++					
T1	T2	K	Akurasi	Canopy	Waktu	T1	T2	K	Akurasi	Canopy	Waktu
80	8	3	0.894892	3	0.806993	90	9	3	0.888054796	3	0.823980049
80	8	3	0.888055	2	0.777553	90	9	3	0.888054796	3	0.761682975
80	8	3	0.888615	3	0.741722	90	9	3	0.894891775	2	0.757470499
80	8	3	0.888789	3	0.784274	90	9	3	0.888054796	3	0.742341503
80	8	3	0.902302	3	0.768137	90	9	3	0.888054796	3	0.764004269
80	8	3	0.888055	2	0.772796	90	9	3	0.670950465	2	0.776985509
80	8	3	0.888055	3	0.790805	90	9	3	0.894891775	2	0.762313681
80	8	3	0.894892	2	0.741687	90	9	3	0.894891775	3	0.754107515
80	8	3	0.674843	2	0.779777	90	9	3	0.888054796	3	0.750025657
80	8	3	0.888055	2	0.777832	90	9	3	0.888054796	3	0.766955336