

OPTIMASI *CLUSTER* DENGAN
ALGORITMA *K-MEANS++* BERBASIS TEKNIK REDUKSI
PRINCIPAL COMPONENT ANALYSIS

Skripsi

Disusun untuk melengkapi syarat-syarat
guna memperoleh gelar Sarjana Sains



NURLAELA

3125121995

PROGRAM STUDI MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI JAKARTA

2017

ABSTRACT

NURLAELA, 3125121995. Cluster Optimization Using K-Means++ Algorithm Based Principal Component Analysis. Thesis. Faculty of Mathematics and Natural Science Jakarta State University. 2014.

Clustering is the part of data mining function which aimed to group amount of data or objects which have similarity in character. K-Means is non hierarchy clustering method which using centers or centroids for grouping. There are some problem that appear in K-Means, such as k centroids which selected randomly. It causes the bad result of clustering and high dimensional data causes the process stuck by the curse of dimensionality. Principal Component Analysis (PCA) is one of solution for high dimensional data clustering by dimension reduction and K-Means++ algorithm for centroid initialization of clustering process. These methods can be applied to the case which evaluated by silhouette coefficient. The result shows that the usage of PCA and K-Means++ produce better cluster than K-Means++ and K-Means itself.

Keywords : *clustering, optimization, K-Means, K-Means++, dimension reduction, principal component analysis, silhouette coefficient.*

ABSTRAK

NURLAELA, 3125121995. Optimasi *Cluster* dengan Algoritma *K-Means++* Berbasis Teknik Reduksi *Principal Component Analysis*. Skripsi. Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta. 2017.

Clustering atau pengelompokan merupakan bagian dari fungsi *data mining* yang bertujuan untuk mengelompokkan sejumlah data atau objek yang memiliki kesamaan sifat. *K-Means* adalah suatu metode *clustering* tidak hierarki yang mengelompokkan objek menggunakan titik pusat atau *centroid*. Terdapat beberapa masalah mengenai metode *K-Means* ini yaitu k buah *centroid* yang dipilih secara acak mengakibatkan hasil *clustering* yang kurang optimal serta variabel atau atribut yang banyak (data berdimensi tinggi) mengakibatkan masalah yang disebut *curse of dimensionality*. Upaya dalam mencegah terjadinya masalah tersebut adalah dengan reduksi dimensi menggunakan *Principal Component Analysis* (PCA) serta menggunakan metode *K-Means++* untuk proses inisialisasi *centroid* pada *clustering*. Metode tersebut dapat diaplikasikan ke dalam studi kasus yang dievaluasi menggunakan koefisien *Silhouette* yang menunjukkan bahwa penggunaan metode reduksi dimensi PCA dan *K-Means++* menghasilkan *cluster* yang lebih optimal dibandingkan *K-Means++* atau *K-Means* itu sendiri.

Kata kunci : pengelompokan, optimasi, *K-Means*, *K-Means++*, reduksi dimensi, *Principal Component Analysis*, koefisien *silhouette*.

PERSEMBAHANKU...

"I'm not going to get somewhere and say 'OK, I'm done.' Success is never final; I'll just keep on going. The same way as failure never being fatal. Just keep going. I'm going to the stars and then past them "

-Connor McGregor

Skripsi ini kupersembahkan untuk Bapak, Umi, Enung, dan Naya.
"Terima kasih atas dukungan, do'a, serta kasih sayang kalian yang tiada henti menemani setiap langkah "

KATA PENGANTAR

Puji syukur kepada Tuhan YME atas pengetahuan dan kemampuan sehingga penulis dapat menyelesaikan skripsi yang berjudul ”Optimasi *Cluster* dengan Algoritma *K-Means++* Berbasis Teknik Reduksi *Principal component Analysis*” yang merupakan salah satu syarat dalam memperoleh gelar Sarjana Program Studi Matematika Universitas Negeri Jakarta.

Skripsi ini berhasil diselesaikan tidak terlepas dari adanya bantuan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis ingin menyampaikan terima kasih terutama kepada:

1. Allah SWT, yang alhamdulillah selalu mendengar doa penulis dan memberikan pelajaran berharga.
2. Ibu Ratna Widyati, S. Si, M. Kom., selaku Dosen Pembimbing I dan Bapak Med Irzal, M.Kom., selaku Dosen Pembimbing II, yang telah meluangkan waktunya dalam memberikan bimbingan, saran, nasehat serta arahan sehingga skripsi ini dapat menjadi lebih baik dan terarah.
3. Ibu Dr. Lukita Ambarwati, S. Pd, M. Si., selaku Koordinator Prodi Matematika FMIPA UNJ yang telah banyak membantu penulis.
4. Ibu Ratna Widyati, S. Si, M. Kom., selaku Pembimbing Akademik atas segala bimbingan dan kerja sama Ibu selama perkuliahan, dan seluruh Bapak/Ibu dosen atas pengajarannya yang telah diberikan, serta karyawan/karyawati FMIPA UNJ yang telah memberikan informasi yang penulis butuhkan dalam menyelesaikan skripsi.
5. Orang tua penulis, Umi Sanih dan Bapak Enjang yang selalu mendukung, memberi motivasi, dan setia membantu penulis dengan penuh cinta.

ta dan kasih sayang yang tulus.

6. Saudari perempuan penulis, Nurhasanah dan Nur Kanaya yang terus memberi semangat, mendoakan penulis, dan selalu menghibur ketika penulis mengalami kesulitan dalam penulisan skripsi ini.
7. Cubutaan alias sahabat selama perkuliahan, Zie Zie P.D., Siti Khotimah, Lusia Agustina, Irma Rahmayani, Rizka Annisa, Meila Nadya, Sharah Annisa, Leny Wiji dan Mella Apriliani yang selalu memberikan semangat dalam perkuliahan.
8. Seluruh mahasiswa Matematika angkatan 2012 dan 2011, selaku teman dan kakak tingkat penulis yang selalu memberikan masukan ketika penulis menyusun skripsi ini.
9. Teman-teman dan kakak senior di kosan Ibu Yuni, Rani Septiani, Ratna, kak Ade, kak Puji, kak Vegy, dan kak Uci yang telah menemani penulis dalam kehidupan perkuliahan selama ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Masukan dan kritikan akan sangat berarti. Semoga skripsi ini dapat bermanfaat bagi pembaca sekalian.

Jakarta, Agustus 2017

Nurlaela

DAFTAR ISI

ABSTRACT	i
ABSTRAK	ii
KATA PENGANTAR	iv
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	3
1.3 Pembatasan Masalah	3
1.4 Tujuan Penulisan	3
1.5 Manfaat Penulisan	4
1.6 Metode Penelitian	4
II LANDASAN TEORI	5
2.1 <i>Data Mining</i>	5
2.1.1 Fungsi <i>Data Mining</i>	7
2.2 <i>Clustering</i>	8
2.2.1 Algoritma <i>K-Means</i>	14
2.2.2 Algoritma <i>K-Means++</i>	26
2.3 Teknik Reduksi Dimensi	29
2.3.1 Algoritma <i>Principal Component Analysis</i>	30

III PEMBAHASAN	33
3.1 Langkah-Langkah Optimasi <i>Cluster</i>	33
3.2 Reduksi Dimensi Menggunakan PCA	34
3.2.1 Data Penelitian	35
3.2.2 Langkah-langkah Reduksi Dimensi PCA	35
3.3 <i>K-Means++ Clustering</i>	40
3.4 Validasi Hasil <i>Clustering</i> Menggunakan <i>Silhouette Coefficient</i> .	48
IV PENUTUP	51
4.1 Kesimpulan	51
4.2 Saran	52
DAFTAR PUSTAKA	53

DAFTAR TABEL

2.1	Data Umur dan Tinggi 8 Objek	19
2.2	<i>Centroid</i> awal (<i>initial centroid</i>)	19
2.3	Jarak <i>Squared Euclidean</i> dan anggota tiap <i>cluster</i> (Iterasi pertama)	21
2.4	Perhitungan <i>centroid</i> baru	21
2.5	<i>Centroid</i> baru ke-2	21
2.6	Jarak <i>Squared Euclidean</i> dan anggota tiap <i>cluster</i> (Iterasi kedua)	22
2.7	<i>Centroid</i> baru ke-3	22
2.8	Jarak <i>Squared Euclidean</i> dan anggota tiap <i>cluster</i> (Iterasi ketiga)	22
2.9	Data nilai 4 mata pelajaran dari 12 objek	23
2.10	<i>Centroid</i> awal (<i>initial centroid</i>)	24
2.11	Jarak <i>Squared Euclidean</i> dan anggota tiap <i>cluster</i> (iterasi pertama)	25
2.12	<i>Centroid</i> baru yang dihasilkan dari rata-rata cluster	25
2.13	Jarak <i>Squared Euclidean</i> dan anggota tiap <i>cluster</i> (iterasi kedua)	26
2.14	<i>Centroid</i> baru yang dihasilkan dari rata-rata cluster	26
2.15	Jarak <i>Squared Euclidean</i> (Iterasi pertama)	28
2.16	<i>Centroid</i> akhir (<i>final centroid</i>)	29
2.17	Jarak <i>Squared Euclidean</i> dan anggota tiap <i>cluster</i> (Iterasi keempat)	29
3.1	Rata-rata dataset	36
3.2	Langkah 3 dan langkah 4	42
3.3	Hasil <i>clustering K-Means</i>	43

3.4	Hasil <i>clustering K-Means++</i> dengan PC1	46
3.5	Hasil <i>clustering K-Means++</i>	47
3.6	Perbandingan hasil validasi koefisien Silhouette	50

DAFTAR GAMBAR

2.1	Proses KDD	6
2.2	<i>Flowchart</i> algoritma <i>K-Means</i>	18
2.3	Plot <i>K-Means cluster</i>	23
2.4	Plot <i>K-Means++ cluster</i>	29
3.1	Proses <i>clustering</i> menggunakan PCA	33
3.2	matriks vektor eigen	38
3.3	Algoritma <i>K-Means++ Clustering</i>	41
4.1	LAMPIRAN 7: Zero Means	70
4.2	LAMPIRAN 8: PC1	78
4.3	LAMPIRAN 9: Nilai koefisien <i>Silhouette K-Means++</i> dengan PC	80
4.4	LAMPIRAN 10 : Nilai koefisien <i>Silhouette K-Means</i>	82
4.5	LAMPIRAN 11 : Nilai koefisien <i>Silhouette K-Means++</i>	84

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan yang cepat dalam teknologi pengumpulan dan penyimpanan data telah memudahkan organisasi untuk mengumpulkan sejumlah data berukuran besar. Sebelum adanya perkembangan teknologi, seringkali alat dan teknik analisis data tradisional tidak dapat digunakan dalam mengekstrak informasi dari data berukuran besar. Penggalian data (*Data mining*) adalah teknologi yang merupakan campuran metode-metode analisis data dengan algoritma-algoritma untuk memproses data berukuran besar untuk dijadikan sebuah informasi. Peran utama *data mining* diantaranya untuk estimasi, prediksi, klasifikasi, termasuk proses pengelompokan (*clustering*).

Pada penulisan kali ini akan lebih fokus membahas mengenai *clustering*. Fungsi dari *clustering* ini sendiri adalah untuk mengelompokkan sejumlah data atau objek kedalam kelompok yang homogen. Artinya, dalam satu kelompok (*cluster*) terdapat data atau objek yang memiliki kesamaan sifat. Adapun permasalahan yang dihadapi saat melakukan *clustering* adalah keakuratan algoritma *clustering* dalam mengelompokkan objek ke dalam kelompok atau *cluster* yang benar.

K-Means merupakan salah satu algoritma *clustering*. Kelebihan dari algoritma ini ialah waktu yang dibutuhkan untuk menjalankan pembelajaran ini relatif cepat, mudah untuk diadaptasi untuk beberapa data dan banyak digunakan dalam teknik *clustering* (Bouveyron & Brunet-Saumard, 2012)(Jain,

2010), akan tetapi algoritma ini sangat sensitif terhadap penempatan awal titik pusat *cluster* (*centroid*) (Celebi, 2013). Hal ini disebabkan pemilihan calon *centroid* dilakukan secara acak. Jika hanya terdapat beberapa titik sampel data, maka cukup mudah untuk menghitung dan mencari titik terdekat dengan *centroid* yang diinisialisasi secara random. Namun jika terdapat banyak sekali titik data, maka perhitungan dan pencarian titik terdekat akan membutuhkan waktu yang lama. Salah satu kekurangan lainnya ialah terjebak dalam masalah tingginya jumlah dimensi. Fenomena permasalahan jumlah dimensi yang tinggi ini biasanya disebut *curse of dimensionality*. *Curse of dimensionality* adalah sebuah fenomena dimana suatu analisis data menjadi hal yang sulit disebabkan oleh jumlah dimensi data yang tinggi atau meningkat.

Beberapa penelitian telah dilakukan untuk memperbaiki kekurangan algoritma *K-Means*, salah satunya Arthur & Sergei Vassilvitskii(2007) dalam penelitiannya mengusulkan algoritma *K-Means++* untuk memilih *centroid* pada algoritma *K-Means*, bukan menghasilkan *centroid* secara acak. Sehingga, keakuratan hasil *cluster K-Means++* lebih baik dibandingkan *K-Means* itu sendiri. Namun masih terdapat beberapa masalah mengenai keakuratan hasil *clustering*, oleh karena itu diperlukan metode tambahan untuk menghasilkan hasil *cluster* yang optimal. Misal menghindari dari *curse of dimensionality* dengan mereduksi dimensi data menggunakan *Principal Component Analysis* (PCA)(Izzudin, 2015) yang dilanjutkan dengan validasi *cluster* menggunakan koefisien *Silhouette* (*Silhouette Coefficient*).

Berdasarkan permasalahan yang dikemukakan, maka dalam penulisan skripsi ini penulis mengangkat judul **Optimasi *Cluster* dengan Algoritma *K-Means++* Berbasis Teknik Reduksi *Principal Component Analysis***

1.2 Perumusan Masalah

Perumusan masalah yang akan dikaji adalah sebagai berikut:

1. Bagaimana cara kerja algoritma *K-Means++* berbasis teknik reduksi PCA?
2. Bagaimana hasil evaluasi *cluster* algoritma *K-Means++* berbasis teknik reduksi PCA menggunakan koefisien *Silhouette*?

1.3 Pembatasan Masalah

Pembatasan masalah dalam penulisan ini adalah:

1. Metode yang digunakan hanya metode *Data Mining Clustering* .
2. Algoritma *clustering* yang digunakan adalah *K-Means++* dengan menggunakan jarak *Squared Euclidean*.
3. Pemilihan jumlah k sudah ditentukan, yaitu $k = 3$.
4. Data yang digunakan merupakan data simulasi.
5. Reduksi dimensi menggunakan *Principal Component Analysis* (PCA).
6. Penghitungan validasi hasil *clustering* hanya dilihat dari koefisien *Silhouette*.
7. Software yang digunakan untuk pembuatan program adalah Matlab R2014a.

1.4 Tujuan Penulisan

Tujuan yang ingin dicapai dalam skripsi ini adalah memperoleh hasil *clustering* yang optimal menggunakan algoritma *K-Means++* berbasis teknik reduksi

PCA .

1.5 Manfaat Penulisan

Manfaat yang diharapkan dari skripsi ini adalah penulis dapat mengaplikasikan teori-teori matematika yang telah dipelajari selama perkuliahan ke dalam bentuk yang lebih riil. Selain itu, skripsi ini diharapkan dapat menambah pengetahuan dan pemahaman penulis mengenai matematika diskrit dan statistik, dan juga diharapkan dapat digunakan sebagai bahan perbandingan bagi pihak lain yang ingin melakukan penelitian lebih lanjut serta dapat digunakan sebagai referensi untuk pengembangan penelitian berikutnya.

1.6 Metode Penelitian

Skripsi ini merupakan kajian teori dalam bidang matematika diskrit dan statistik yang didasarkan pada buku-buku dan jurnal-jurnal tentang *data mining* dan *clustering*. Referensi utama yang digunakan dari jurnal David Arthur & Sergei Vassilvitskii (2007).

BAB II

LANDASAN TEORI

Bab ini berisi dasar-dasar teori (*data mining*), *clustering*, proses *K-Means* menjadi *K-Means++* dan PCA.

2.1 *Data Mining*

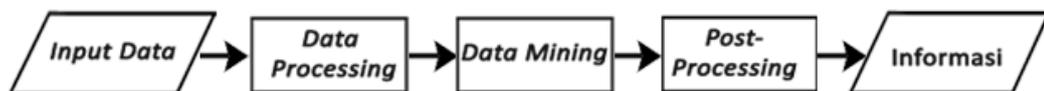
Data mining merupakan teknologi yang menggabungkan metode analisis tradisional dengan algoritma yang canggih untuk memproses data volume besar. Berry & Linoff(2004) dalam buku *Data Mining Technique for Marketing, Sales, and Customer Support* mendefinisikan *data mining* sebagai suatu proses eksplorasi dan analisis secara otomatis maupun semi otomatis terhadap data dalam jumlah besar dengan tujuan menemukan pola atau aturan yang berarti (Larose, 2006). Lalu menurut Connolly & Beg(2010), *Data mining* adalah suatu proses ekstraksi atau penggalian data yang belum diketahui sebelumnya, namun dapat dipahami dan berguna dari *database* yang besar serta digunakan untuk membuat suatu keputusan bisnis yang sangat penting.

Dari beberapa teori yang dijabarkan oleh para ahli di atas, bahwa *data mining* mempunyai peranan yang sangat penting dalam pengolahan data untuk dijadikan sebuah informasi. Peran utama *data mining* diantaranya untuk estimasi, prediksi, klasifikasi, termasuk proses *clustering*. Perkembangan yang cepat dalam teknologi pengumpulan dan penyimpanan data telah memudahkan organisasi untuk mengumpulkan sejumlah data berukuran besar, sehingga menghasilkan gunung data. *Data mining* juga merupakan proses pencari-

an secara otomatis informasi yang berguna dalam tempat penyimpanan data berukuran besar. Istilah lain yang sering digunakan diantaranya *Knowledge Discovery (Mining) in Databases* (KDD).

Istilah *data mining* dan *Knowledge Discovery in Database* (KDD) sering kali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain. Dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining*.

Hasil *data mining* sering diintegrasikan dengan *Decision Support System* (DSS). Sebagai contoh, informasi yang dihasilkan oleh *data mining* dapat diintegrasikan dengan Tool Manajemen Kampanye Produk sehingga promosi pemasaran yang efektif dilaksanakan dan dapat diuji. Integrasi tersebut memerlukan langkah *postprocessing* yang menjamin bahwa hanya hasil yang valid dan berguna yang akan digabungkan dengan DSS. Salah satu pekerjaan dan *postprocessing* adalah visualisasi yang memungkinkan *system analyst* untuk mengeksplor data dan hasil *data mining* dari berbagai sudut pandang. Ukuran-ukuran statistik dan metode pengujian hipotesis dapat digunakan selama *postprocessing* untuk membuang hasil *data mining* yang palsu.



Gambar 2.1: Proses KDD

Menurut Han & Kamber, (2011:24), secara garis besar *data mining* dapat dikelompokkan menjadi 2 kategori utama, yaitu:

1. *Predictive*

Predictive merupakan proses untuk menemukan pola dari data dengan

menggunakan beberapa variabel lain di masa depan. Salah satu teknik yang terdapat dalam *predictive mining* adalah klasifikasi. Tujuan dari tugas prediktif adalah untuk memprediksi nilai dari tertentu berdasarkan pada nilai atribut-atribut lain. Atribut yang diprediksi umumnya dikenal sebagai target atau variabel tak bebas, sedangkan atribut-atribut yang digunakan untuk membuat prediksi dikenal sebagai *explanatory* atau variabel bebas. Contohnya, perusahaan retail dapat menggunakan *data mining* untuk memprediksikan penjualan dari produk mereka di masa depan dengan menggunakan data-data yang telah didapatkan dari beberapa minggu sebelumnya.

2. *Descriptive*

Descriptive dalam *data mining* merupakan proses untuk menemukan karakteristik penting dari data dalam suatu basis data. Tujuan dari tugas *descriptive* adalah untuk menurunkan pola-pola (korelasi, *trend*, *cluster*, trayektori, dan anomali) yang meringkas hubungan yang pokok dalam data. Tugas *data mining* deskriptif sering merupakan penyelidikan dan seringkali memerlukan teknik postprocessing untuk validasi dan penjelasan hasil.

2.1.1 Fungsi *Data Mining*

Menurut Maclennan, et al (2009:6) berikut adalah fungsi *data mining* secara umum:

1. Fungsi *classification* adalah proses untuk mencari model atau fungsi yang menggambarkan dan membedakan kelas-kelas atau konsep data. Fungsi dari *classification* adalah untuk mengklasifikasikan suatu target class ke dalam kategori yang dipilih.

2. Fungsi *association* adalah untuk mencari keterkaitan antara atribut atau item set, berdasarkan jumlah item yang muncul dan *rule association* yang ada.
3. Fungsi *regression* hampir sama dengan klasifikasi. Fungsi dari *regression* adalah bertujuan untuk mencari prediksi dari suatu pola yang ada.
4. Fungsi *forecasting* adalah untuk peramalan waktu yang akan datang berdasarkan trend yang telah terjadi di waktu sebelumnya.
5. Fungsi *sequence analysis* adalah untuk mencari pola urutan dari rangkaian kejadian.
6. Fungsi *deviation analysis* adalah untuk mencari kejadian langka yang sangat berbeda dari keadaan normal (kejadian abnormal).
7. Fungsi *clustering* adalah untuk mencari pengelompokan atribut ke dalam segmentasi-segmentasi berdasarkan similaritas.

2.2 *Clustering*

Menurut Han & Kamber (2011), *Clustering* adalah proses pengelompokan kumpulan data menjadi beberapa kelompok sehingga objek di dalam satu kelompok memiliki banyak kesamaan dan memiliki banyak perbedaan dengan objek dikelompok lain. *Clustering* (pengelompokan data) adalah proses membuat pengelompokan sehingga semua anggota dari setiap partisi mempunyai persamaan berdasarkan matriks tertentu. Sebuah *cluster* adalah sebuah sekumpulan objek yang digabung bersama karena persamaan atau kedekatannya. *Clustering analysis* bertujuan untuk membentuk kelompok dengan karakteristik yang sama (Sharma,1996).

Prinsip dari *clustering* ini adalah memaksimalkan kemiripan antar anggota satu *cluster* dan meminimumkan kemiripan antar *cluster*. Perbedaan dan persamaannya biasanya berdasarkan nilai atribut dari objek tersebut dan dapat juga berupa perhitungan jarak. *Clustering* sendiri juga disebut *Unsupervised Classification*, karena *clustering* lebih bersifat untuk dipelajari dan diperhatikan. Partisi tidak dilakukan dengan manual tetapi dengan algoritma *clustering*. Oleh karena itu, *Clustering* sangat berguna dan bisa menemukan *group* yang tidak dikenal dalam data.

Clustering mempertimbangkan sebuah pendekatan penting untuk mencari kemiripan dalam data dan menempatkan data yang sama kedalam kelompok-kelompok. Gagasan mengenai pengelompokan data, atau *clustering*, memiliki sifat yang sederhana dan dekat dengan cara berpikir manusia. Ketika dipresentasikan jumlah data yang besar, manusia biasanya cenderung merangkumkan jumlah data yang besar ini ke dalam sejumlah kelompok-kelompok kecil atau *cluster* untuk memfasilitasi analisisnya lebih lanjut.

Clustering juga dapat digunakan untuk *outlier detection* dimana jarak terluar lebih menarik dari kasus-kasus yang biasanya. Sebagai cabang dari statistika, analisis *cluster* telah lebih luas dipelajari dalam beberapa tahun, mengutamakan pada *distance-based cluster analysis* (Han, 2007: 384).

Berdasarkan uraian di atas diperoleh manfaat analisis *cluster* adalah sebagai berikut :

1. Mengelompokkan objek-objek (individu-individu) menjadi beberapa kelompok yang mempunyai sifat yang relatif sama (homogen),
2. Membedakan dengan jelas antara satu kelompok *cluster* dengan kelompok lainnya,
3. Menerapkan dasar-dasar pengelompokan dengan lebih konsisten,

4. Mengembangkan suatu metode generalisasi secara induktif, yaitu pengambilan kesimpulan secara umum dengan berdasarkan fakta-fakta khusus,
5. Menemukan tipologi yang cocok dengan karakter objek yang diteliti, dan
6. Mendiskripsikan sifat-sifat atau karakteristik dari masing-masing kelompok.

Berdasarkan manfaat dan tujuan dari analisis cluster adalah mengelompokkan objek yang mirip ke dalam cluster yang sama, sehingga beberapa ukuran diperlukan untuk mengetahui seberapa mirip objek-objek tersebut. Terdapat tiga metode yang diterapkan dalam mengukur kesamaan antar objek yaitu ukuran asosiasi, ukuran korelasi, dan ukuran jarak.

1. Ukuran asosiasi digunakan bila objek-objek yang diamati bertipe *non-metric* (tipe nominal atau ordinal). Misalnya, responden hanya menjawab ya atau tidak dalam sebuah pertanyaan.
2. Ukuran korelasi dapat diukur dengan *corelation coefficient* antara pasangan objek-objek yang diukur dalam beberapa variabel. Tingginya korelasi menunjukkan kesamaan dan rendahnya korelasi menunjukkan ketidaksamaan.
3. Ukuran jarak merupakan ukuran ketidakmiripan, dimana jarak yang besar menunjukkan sedikit kesamaan sebaliknya jarak kecil menunjukkan bahwa suatu objek semakin mirip dengan obyek lain. Perbedaan dengan ukuran korelasi adalah bahwa ukuran korelasi bisa saja tidak memiliki kesamaan nilai tetapi memiliki kesamaan pola, sedangkan ukuran jarak lebih memiliki kesamaan nilai meskipun memiliki pola yang berbeda.

Perhitungan jarak dapat dilakukan dengan menggunakan salah satu formula:

- Jarak *Euclidean*

$$d_{ij} = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (2.1)$$

- Jarak *Squared Euclidean*

$$(d_{ij})^2 = \sum_{k=1}^p (x_{ik} - x_{jk})^2 \quad (2.2)$$

- Jarak *Manhattan*

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}|$$

- Jarak *Pearson*

$$d_{ij} = \sqrt{\sum_{k=1}^p \frac{(x_{ik} - x_{jk})^2}{\text{var}(x_k)}}$$

Keterangan :

d_{ij}	:Jarak antara objek ke- i dengan ke- j
i	: $1, 2, \dots, n$
j	: $1, 2, \dots, n$
k	: $1, 2, \dots, p$
x_{ik}	:Nilai pengamatan objek ke- i peubah ke- k
x_{jk}	:Nilai pengamatan objek ke- j peubah ke- k
p	:Banyaknya peubah atau dimensi

Pengelompokkan jenis-jenis *clustering* terbagi 2 yaitu:

1. Berdasarkan pendekatan *clustering*

Menurut Gunadarma(2008), ada beberapa kategori pendekatan *clustering* yaitu:

- (a) Algoritma Partisi yang mempartisi objek-objek ke dalam k *cluster* dan realokasi objek-objek secara iteratif untuk memperbaiki *clustering*.
- (b) Algoritma Hierarki terdiri dari *Agglomerative* dan *Divisive*. *Agglomerative* yaitu dimana setiap objek merupakan *cluster*, gabungan dari *cluster-cluster* membentuk *cluster* yang besar dan *Divisive* dimana semua objek berada dalam suatu *cluster*, pembagian *cluster* tersebut membentuk *cluster-cluster* yang kecil.
- (c) Metode berbasis densitas menggunakan basis koneksitas dan fungsi densitas serta *noise* disaring, kemudian diperoleh *cluster-cluster* dalam bentuk sembarang.
- (d) Metode berbasis *grid* mengelompokkan dengan kuantitas ruang objek ke dalam struktur *grid*.
- (e) Berbasis model yaitu menggunakan model untuk menemukan keadaan data yang baik

2. Berdasarkan metode pengelompokan *clustering*

Jika ditinjau dari metode pengelompokannya, analisis *cluster* memiliki dua metode, yaitu :

(a) Metode Hierarki

Metode ini digunakan untuk mencari struktur pengelompokan dari objek-objek. Jadi, hasil pengelompokannya disajikan secara hierarki atau berjenjang. Metode hierarki ini terdiri dari dua cara, yaitu

penggabungan (*Agglomerative*) dan pemecahan (*Devise*). Salah satu *clustering* metode hierarki yaitu :

i. *Single Linkage*

Prosedur ini didasarkan pada jarak terkecil di antara dua data dari dua kelompok yang berbeda atau disebut juga nilai kemiripan termaksimal. Jika dua objek terpisah oleh jarak yang pendek maka kedua obyek tersebut akan digabung menjadi satu *cluster* dan demikian seterusnya. Misal, jika D menyatakan matriks kedekatan dan d disebut dengan nilai jarak antara dua data dari kelompok yang berbeda. Jarak kelompok U ke V diperoleh dengan membandingkan jarak data dalam kelompok U ke kelompok V (d_{UV}). Formulasi untuk *Single Linkage* adalah

$$d_{UV} = \min \{d_{UV}\}, d_{UV} \in D \quad (2.3)$$

ii. *Complete Linkage*

Berlawanan dengan *Single Linkage* prosedur ini pengelompokkannya berdasarkan jarak terjauh di antara pasangan dua data dari dua kelompok yang berbeda atau disebut juga dengan nilai kemiripan terminimal. Formulasi untuk *Complete Linkage* adalah

$$d_{UV} = \max \{d_{UV}\}, d_{UV} \in D \quad (2.4)$$

iii. *Average Linkage*

Prosedure ini hampir sama dengan *Single Linkage* maupun *Complete Linkage*, namun kriteria yang digunakan adalah rata-rata jarak seluruh individu dalam suatu *cluster* dengan jarak seluruh individu dalam *cluster* yang lain. Metode ini merupakan pendekatan yang mengambil pertengahan di antara *Single*

Linkage dan Complete Linkage.

$$d_{UV} = \frac{1}{|U| \times |V|} \sum_{U \in D} \sum_{V \in D} d_{UV} \quad (2.5)$$

dimana $|U|$ dan $|V|$ masing-masing menyatakan jumlah data yang ada dikelompok U dan V . Untuk semua jarak yang sudah didapat, jika kelompok U dan V adalah kandidat dua kelompok yang akan digabung sedangkan W adalah kelompok lain yang tidak menjadi kandidat. Sehingga dari semua jarak yang sudah dihitung dari persamaan 2.1, 2.2 atau 2.3 kemudian pilihlah nilai yang paling terkecil. Formulasinya adalah

$$U \cap (V, W) = \min(d_{UV}, d_{UV}), d_{UV}, d_{UV} \in D \quad (2.6)$$

(b) Metode Tidak Hierarki

Metode ini digunakan apabila jumlah kelompok yang diinginkan sudah diketahui dan biasanya dipakai untuk mengelompokkan data yang ukurannya besar. Salah satu contohnya adalah metode *K-Means clustering*.

2.2.1 Algoritma *K-Means*

Pengertian algoritma adalah suatu urutan dari beberapa yang logis guna menyelesaikan masalah. Algoritma merupakan prosedur dari beberapa langkah demi langkah untuk penghitungan menurut ilmu matematika dan komputer. Algoritma dipakai untuk penghitungan, penalaran otomatis, dan pemrosesan data, Sedangkan menurut statistik dan mesin pembelajaran, pengelompokan *K-Means* merupakan metode analisis kelompok yang mengarah pada partisi N objek pengamatan ke dalam kelompok *cluster*, dimana setiap objek pengamatan dimiliki oleh sebuah kelompok dengan *mean* (rata-rata) terdekat.

Keduanya mencoba untuk menemukan pusat dari kelompok dalam data sebanyak iterasi perbaikan yang dilakukan oleh kedua algoritma (Prasetyo, 2012).

K-Means memiliki properti : selalu ada k *cluster*, paling tidak memiliki satu data dalam tiap *cluster*, *cluster* ini merupakan *non-hierarki* dan tidak akan terjadi *overlap*.

Adapun kelebihan dan kekurangan *K-Means* yaitu:

1. Kelebihan *K-Means*

Beberapa kelebihan pada algoritma *K-Means*, yaitu:

- (a) Mudah untuk diimplementasikan dan dijalankan.
- (b) Waktu yang dibutuhkan untuk menjalankan pembelajaran ini relatif cepat.
- (c) Dengan jumlah variabel yang besar, *K-Means* dapat berjalan lebih cepat daripada hierarchical clustering (jika k kecil).
- (d) Mudah untuk diadaptasi.
- (e) Umum digunakan.

2. Kekurangan *K-Means*

Algoritma *K-Means* memiliki beberapa kelebihan, namun ada kekurangannya juga. Kekurangan dari algoritma tersebut yaitu :

- (a) Sebelum algoritma dijalankan, k buah titik diinisialisasi secara random sehingga pengelompokan data yang dihasilkan dapat berbeda-beda .
- (b) Jika nilai random untuk inisialisasi kurang baik, maka pengelompokan yang dihasilkan pun menjadi kurang optimal.

- (c) Dapat terjebak dalam masalah yang disebut *curse of dimensionality*. Hal ini dapat terjadi jika data memiliki dimensi yang sangat tinggi (Contoh jika data terdiri dari 2 atribut maka dimensinya adalah 2 dimensi. Namun jika ada 20 atribut, maka akan ada 20 dimensi).

Adapun algoritma *K-Means* itu sendiri adalah sebagai berikut :

1. Menentukan k sebagai jumlah *cluster* yang ingin dibentuk pada metode *K-Means*.
2. Menentukan *initial* pusat k *cluster* (*centroid*) secara acak

$$C = \{c_1, c_2, \dots, c_k\}$$

dimana C beranggotakan c_1 (*centroid* ke-1) hingga c_k (*centroid* ke- k) dimana k adalah jumlah *cluster* yang ingin dibentuk.

3. Menghitung jarak antara *centroid* dengan objek. Misalkan dua objek ke- i dan ke- j yang berada pada p dimensi. Formula jarak yang sering digunakan dalam perhitungan K-Means adalah jarak *Squared Euclidean*

$$(d_{ij})^2 = \sum_{k=1}^p (x_{ik} - x_{jk})^2$$

Beberapa penelitian menunjukkan bahwa jarak *Euclidean* adalah jarak yang paling sering digunakan, begitu juga dengan kuadrat jarak atau *Squared Euclidean*. Jarak *Euclidean* menjadi pilihan jika ingin memberikan jarak terpendek antara dua titik (jarak lurus), sedangkan *Manhattan* memberikan jarak terjauh pada dua data. *Manhattan* juga sering digunakan karena kemampuannya dalam mendeteksi keadaan khusus, seperti keberadaan outlier dengan lebih baik (Agusta, 2005). Namun *Squared Euclidean Distance* memiliki keunggulan dengan mengambil *square root* dimana kecepatan penghitungan ditekankan.

4. Alokasikan objek ke *centroid* terdekat dengan melihat jarak terdekat yang diperoleh.

$$\begin{aligned}\phi &= \sum_{x \in X} \min_{c \in C} (D) \\ &= \sum_{x \in X} \min_{c \in C} \|x - c\|^2\end{aligned}\tag{2.7}$$

Formula ini dapat menentukan hasil *clustering* dengan mengelompokkan objek berdasarkan masing-masing *cluster*, dimana ϕ dinotasikan sebagai *corresponding potential* atau jarak minimum x terhadap satu *centroid* terdekat.

5. Menentukan *centroid* baru dari rata-rata tiap *cluster* dengan rumus:

$$C_i^* = \frac{1}{|C_i|} \sum_{x \in C_i} x\tag{2.8}$$

Keterangan :

C_i^* : *Centroid* ke-i yang baru

C_i : *Centroid* ke-i yang lama

$|C_i|$: Banyaknya anggota *cluster* ke-i yang lama

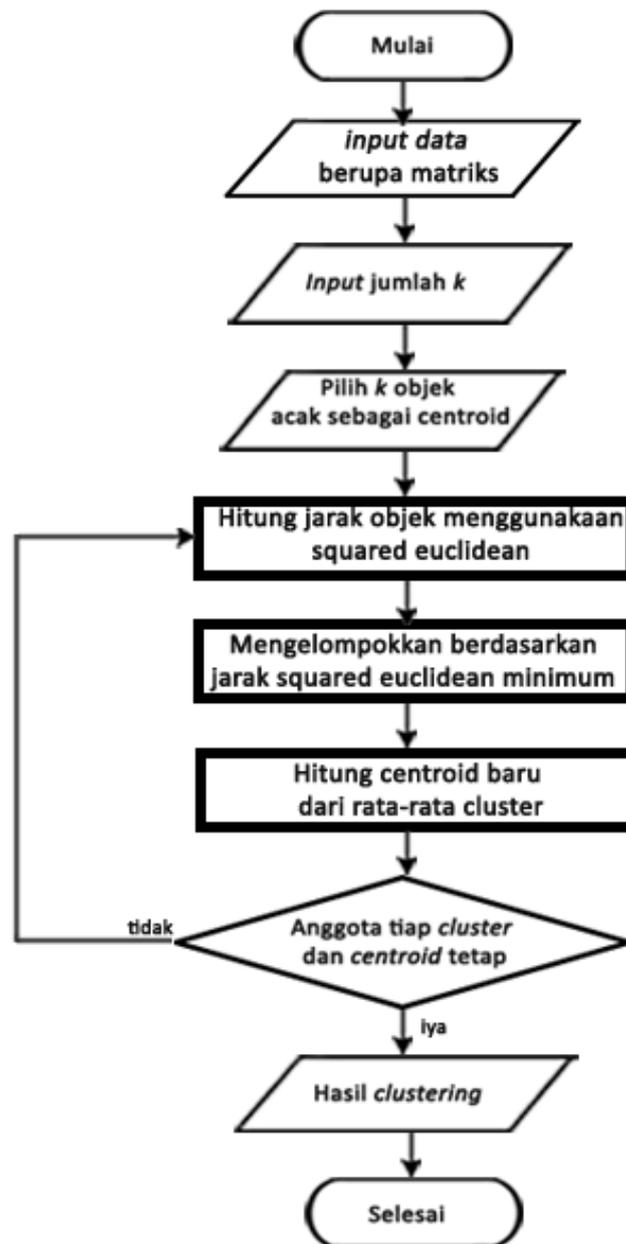
i : 1, 2, ..., k

Maka, dengan mengumpulkan semua objek yang termasuk ke dalam suatu *cluster* dan menghitung rata-rata dari tiap *cluster* untuk memperoleh *centroid* baru.

6. Kembali ke langkah 5 hingga objek dalam *cluster* dan *centroid* tidak berubah.

Setelah melakukan langkah-langkah tersebut akan dihasilkan suatu *cluster* yang memiliki jarak *Squared Euclidean* minimum yang menggambarkan kemiripan di tiap *cluster*nya. Dimana di tiap *cluster* memiliki kemiripan yang besar namun memiliki ketidakmiripan yang besar antar *cluster*.

Contoh perhitungan *K-Means*



Gambar 2.2: *Flowchart* algoritma *K-Means*

- *K-Means* untuk 2 variabel

Data yang akan dikelompokkan adalah data umur dan tinggi dari 8 objek dengan demikian data tersebut memiliki 2 dimensi.

Tabel 2.1: Data Umur dan Tinggi 8 Objek

Objek ke-	Umur	Tinggi
1	25	50
2	26	48
3	35	65
4	67	60
5	55	65
6	53	60
7	34	48
8	48	50

Sumber: Niswa Djupri (2010)

Berikut merupakan langkah-langkah perhitungan *K-Means clustering*:

1. Langkah pertama

Menentukan k sebagai jumlah *cluster* yang ingin dibentuk pada metode *K-Means*. Pada kasus ini, jumlah *cluster* yang ingin dibentuk adalah $k = 2$.

2. Langkah kedua

Menentukan *initial* pusat k *cluster* (*centroid*) secara acak yaitu

$$C = \{c_1 = x_2, c_2 = x_7\}.$$

Tabel 2.2: *Centroid* awal (*initial centroid*)

<i>centroid</i> ke-	umur	tinggi
1	26	48
2	34	48

3. Langkah ketiga

Menghitung jarak antara *centroid* dengan objek menggunakan per-

samaan 2.2 yaitu jarak *Squared Euclidean*. Misal, untuk mencari jarak objek ke- i ke *centroid* adalah sebagai berikut:

– Untuk objek ke-1

$$(\text{Jarak objek ke-1 ke } c_1) = (25 - 26)^2 + (50 - 48)^2 = 5$$

$$(\text{Jarak objek ke-1 ke } c_2) = (25 - 34)^2 + (50 - 48)^2 = 85$$

– Untuk objek ke-2

$$(\text{Jarak objek ke-2 ke } c_1) = (26 - 26)^2 + (48 - 48)^2 = 0$$

$$(\text{Jarak objek ke-2 ke } c_2) = (26 - 34)^2 + (48 - 48)^2 = 64$$

Seterusnya hingga perhitungan dilakukan pada semua objek terhadap *centroid*.

4. Langkah keempat

Alokasikan objek ke *centroid* terdekat dengan melihat jarak terdekat yang diperoleh, sehingga diperoleh jarak objek terhadap *centroid* dan anggota tiap *cluster*.

Diperoleh jarak objek ke-1 terhadap c_1 adalah 5 dan jarak objek ke-1 terhadap c_2 adalah 85, maka :

$$\phi = \sum_{x \in X} \min_{c \in C} (D)$$

$$\phi = \min(5, 85)$$

$$\phi = 5$$

Sehingga jarak minimum yang diperoleh adalah 5 yang merupakan jarak terdekat terhadap *centroid* ke-1 (c_1) maka objek ke-1 merupakan anggota *cluster* yang ke-1.

5. Langkah kelima

Tabel 2.3: Jarak *Squared Euclidean* dan anggota tiap *cluster* (Iterasi pertama)

Objek ke-	Jarak ke c_1	jarak ke c_2	Anggota <i>cluster</i> ke-
1	5	85	1
2	0	64	1
3	370	290	2
4	1825	1233	2
5	1130	730	2
6	873	505	2
7	64	0	2
8	488	200	2

Menentukan *centroid* baru dari rata-rata tiap *cluster* dan hitung kembali jarak antara objek dengan *centroid*. Dari Tabel 2.3 maka diperoleh rata-rata objek tiap *cluster* yaitu :

Tabel 2.4: Perhitungan *centroid* baru

Objek ke- i di c_1	Umur	Tinggi	Objek ke- i di c_2	Umur	Tinggi
1	25	50	3	35	65
2	26	48	4	67	60
-	-	-	5	55	65
-	-	-	6	53	60
-	-	-	7	34	48
-	-	-	8	48	50
Rata-rata	25,5	49	Rata-rata	48,67	58

Tabel 2.5: *Centroid* baru ke-2

<i>centroid</i> ke-	umur	tinggi
1	25,5	49
2	48,67	58

6. langkah keenam

Kembali ke langkah 5 hingga objek dalam *cluster* dan *centroid* tidak berubah yaitu menghitung jarak antara *centroid* dengan objek, mengalokasikan objek ke *centroid* terdekat serta menghitung *centroid* baru.

Jarak *Euclidean* dan anggota tiap *cluster* yang baru diperoleh sebagai berikut:

Tabel 2.6: Jarak *Squared Euclidean* dan anggota tiap *cluster* (Iterasi kedua)

Objek ke-	Jarak ke c_1	jarak ke c_2	Anggota <i>cluster</i> ke-
1	2,02	640	1
2	0,7	629	1
3	357,02	245	2
4	1850,7	328	2
5	1137	85	2
6	884	20	2
7	72,7	325	1
8	508,02	65	2

Centroid baru yang dihasilkan adalah :

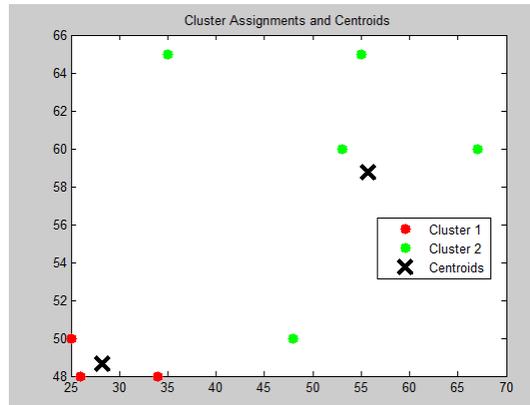
Tabel 2.7: *Centroid* baru ke-3

<i>centroid</i> ke-	umur	tinggi
1	28,3	48,67
2	58,75	52,75

Tabel 2.8: Jarak *Squared Euclidean* dan anggota tiap *cluster* (Iterasi ketiga)

Objek ke-	Jarak ke c_1	jarak ke c_2	Anggota <i>cluster</i> ke-
1	2,02	640	1
2	0,7	629	1
3	357,02	245	2
4	1850,7	328	2
5	1137	85	2
6	884,7	20	2
7	72,7	325	1
8	508,02	65	2

Dari Tabel 2.7 dan Tabel 2.8 anggota tiap *cluster* dan *centroid* tidak berubah, dengan demikian telah diperoleh hasil *clustering* menggunakan metode *K-Means*.



Gambar 2.3: Plot *K-Means cluster*

- *K-Means* untuk 4 variabel

Data yang akan dikelompokkan adalah data nilai mata pelajaran matematika, fisika, akuntansi dan sosiologi dari 12 objek dengan demikian data tersebut memiliki 4 dimensi.

Tabel 2.9: Data nilai 4 mata pelajaran dari 12 objek

Objek ke-	Matematika	Fisika	Akuntansi	Sosiologi
1	79	75	78	76
2	84	76	77	75
3	77	84	89	77
4	78	86	77	75
5	82	82	82	79
6	75	75	75	79
7	77	75	80	75
8	77	76	80	75
9	79	76	81	76
10	80	75	78	77
11	76	71	81	79
12	80	77	77	77

Sumber: Amanda,dkk(2013)

1. Langkah pertama

Menentukan k sebagai jumlah *cluster* yang ingin dibentuk pada me-

tode *K-Means*. Pada kasus ini, jumlah *cluster* yang ingin dibentuk adalah $k = 4$.

2. Langkah kedua

Menentukan *initial* pusat k *cluster* (*centroid*) secara acak yaitu

$$C = \{c_1 = x_2, c_2 = x_7, c_3 = x_9, c_4 = x_{12}\}.$$

Tabel 2.10: *Centroid* awal (*initial centroid*)

<i>centroid</i> ke-	Matematika	Fisika	Akuntansi	Sosiologi
1	84	76	77	75
2	77	75	80	75
3	79	76	81	76
4	80	77	77	77

- Langkah ketiga

Menghitung jarak antara *centroid* dengan objek menggunakan persamaan 2.2 yaitu jarak *Squared Euclidean*.

- Langkah keempat

Alokasikan objek ke *centroid* terdekat dengan melihat jarak terdekat yang diperoleh, sehingga diperoleh jarak objek terhadap *centroid* dan anggota tiap *cluster*.

- Langkah kelima

Menentukan *centroid* baru dari rata-rata tiap *cluster* dan hitung kembali jarak antara objek dengan *centroid*. Dari Tabel 2.11 maka diperoleh rata-rata objek tiap *cluster* pada Tabel 2.12.

- Langkah keenam

Kembali ke langkah 5 hingga objek dalam *cluster* dan *centroid* tidak berubah yaitu menghitung jarak antara *centroid* dengan objek, menga-

Tabel 2.11: Jarak *Squared Euclidean* dan anggota tiap *cluster* (iterasi pertama)

Objek ke-	Matematika	Fisika	Akuntansi	Sosiologi	Anggota <i>cluster</i> ke-i
1	28	9	10	7	4
2	0	59	42	21	1
3	261	166	133	202	3
4	136	131	118	89	4
5	81	94	55	58	3
6	102	45	62	37	4
7	59	0	7	26	2
8	58	1	6	23	2
9	42	7	0	19	3
10	22	17	12	5	4
11	121	34	43	72	2
12	21	26	19	0	4

Tabel 2.12: *Centroid* baru yang dihasilkan dari rata-rata *cluster*

<i>Cluster</i> ke-i	M	F	A	S
1	84	76	77	75
2	76,67	74	80,33	76,33
3	79,33	80,67	84	77,33
4	78,40	77,60	77	76,80

lokasikan objek ke *centroid* terdekat serta menghitung *centroid* baru.

Setelah melakukan tiga kali iterasi, telah diperoleh hasil dari pengelompokan *K-Means* dengan 4 *cluster* dari 12 objek dan 4 variabel. Dimana hasil pengelompokan pada iterasi ketiga sama persis dengan hasil iterasi kedua, dengan demikian iterasi pengelompokan karena anggota tiap *cluster* dan *centroid* tidak berubah. Hasil pengelompokan dengan tiga iterasi, sehingga diperoleh 4 *cluster* dengan *cluster* 1 beranggotakan 1 anggota, *cluster* 2 beranggotakan 5 anggota, *cluster* 3 beranggotakan 2 anggota, dan *cluster* 4 beranggotakan 5 anggota. Selengkapnya terdapat pada tabel berikut:

Tabel 2.13: Jarak *Squared Euclidean* dan anggota tiap *cluster* (iterasi kedua)

Objek ke-	Matematika	Fisika	Akuntansi	Sosiologi	Anggota <i>cluster</i> ke-i
1	28	9	10	7	4
1	28	12	70	8,76	4
2	0	70,67	98,00	37,16	1
3	261	175,67	41,67	186,96	3
4	136	158,67	84,67	73,96	4
5	81	102,33	15,67	62,16	3
6	102	39,33	134,67	27,16	4
7	59	3	59,00	20,96	2
8	58	6	48,67	16,76	2
9	42	10	32,67	19,56	2
10	22	18	68,67	10,36	4
11	121	17	116,33	70,16	2
12	21	31,67	63	2,96	2

Tabel 2.14: *Centroid* baru yang dihasilkan dari rata-rata *cluster*

<i>Cluster</i> ke-i	M	F	A	S
1	84	76	77	75
2	76,67	74	80,33	76,33
3	79,33	80,67	84	77,33
4	78,40	77,60	77	76,80

2.2.2 Algoritma *K-Means*++

Pada algoritma *K-Means*, *centroid* awal dipilih secara acak dari kumpulan data. Meskipun pendekatan ini sederhana dan cepat, akan tetapi terkadang menghasilkan hasil *cluster* yang jauh dari optimal, karena tidak ada jaminan akurasi. Menurut Arthur & Vassilvitskii,(2007), pada *K-Means*++ *centroid* awal dipilih dengan probabilitas yang dihasilkan dari hasil bagi jarak objek terhadap jumlah kumulatif semua jarak objek terhadap *centroid*, probabilitas pemilihan titik sebagai *centroid* sebanding dengan jarak *centroid* terdekat yang sudah dipilih (Karch ,2010). Berikut adalah algoritma *K-Means*++:

1. Pilih pusat awal c_1 seragam secara acak dari X . Hitung jarak objek

dengan jarak *Squared Distance* antara semua titik dalam dataset dan c_1 ,

$$D_i^2 = \|x_i - c_1\|^2 \quad (2.9)$$

2. Pilih pusat kedua c_2 dari X secara acak dari distribusi probabilitas :

$$\frac{D_i^2}{\sum_j D_j^2} \quad (2.10)$$

Formula tersebut disebut dengan "*D² weighting*".

3. Hitung ulang jarak objek, lalu pilih kembali *centroid* baru menggunakan *D² weighting*.

$$D_i^2 = \min (\|x_i - c_1\|^2, \|x_i - c_2\|^2) \quad (2.11)$$

4. Pilih pusat c_k berturut-turut dan menghitung ulang jarak objek

$$D_i^2 = \min (\|x_i - c_1\|^2, \dots, \|x_i - c_k\|^2) \quad (2.12)$$

5. Jika k center telah dipilih, langkah selanjutnya standar algoritma K-Means.

K-Means memperbaiki algoritma *K-Means* dalam pemilihan *initial centroid* sehingga pemilihan *initial centroid* tidak lagi dipilih secara acak melainkan mengikuti peluang dari "*D² weighting*". Langkah no (4) pun menunjukkan bahwa jarak minimum yang diperoleh, dihasilkan dari perbandingan jarak *Squared Euclidean* yang dihasilkan dari awal iterasi. Sehingga jarak minimum yang diperoleh lebih optimal hingga memperoleh k *centroid* yang ditentukan sebelumnya.

Misal berikut adalah perhitungan untuk pemilihan *initial cluster K-Means++*

1. Langkah pertama

Pilih pusat awal c_1 seragam secara acak dari X . Hitung jarak objek dengan jarak *Squared Distance* antara semua titik dalam dataset dan c_1 . Misal c_1 yang diambil adalah x_3 .

2. langkah kedua

Pilih pusat kedua c_2 dari X secara acak dari distribusi probabilitas.

Tabel 2.15: Jarak *Squared Euclidean* (Iterasi pertama)

Objek ke-	Jarak ke c_1	Kumulatif jarak	Peluang	Peluang kumulatif
1	325	325	$\frac{325}{3177} = 0,102$	0,102
2	370	695	$\frac{370}{3177} = 0,116$	0,218
3	0	695	$\frac{0}{3177} = 0$	0,218
4	1049	1744	$\frac{1049}{3177} = 0,33$	0,54
5	400	2144	$\frac{400}{3177} = 0,126$	0,67
6	349	2493	$\frac{349}{3177} = 0,109$	0,78
7	290	2783	$\frac{290}{3177} = 0,091$	0,87
8	394	3177	$\frac{394}{3177} = 0,124$	1

Besaran yang diambil secara acak berkisar dari nilai terkecil jarak hingga nilai terakhir kumulatif jarak. Namun hal itu sebanding dengan nilai kumulatif peluangnya, sehingga penentuan *centroid* berikutnya dapat dilakukan mengambil nilai peluang secara acak. Misal, nilai peluang acak yang diambil adalah 0,89. Maka pemilihan *centroid* selanjutnya harus memiliki peluang kumulatif yang lebih besar dari 0,89. Sehingga objek ke-8 diperoleh sebagai c_2 dimana peluang kumulatifnya lebih besar dari 0,89.

3. Langkah ketiga

k *centroid* telah terpilih, dimana $k = 2$, sehingga diperoleh $c_1 = x_3$ dan $c_2 = x_8$. Langkah selanjutnya dilanjutkan dengan proses K-Means. Sehingga hasilnya adalah sebagai berikut:

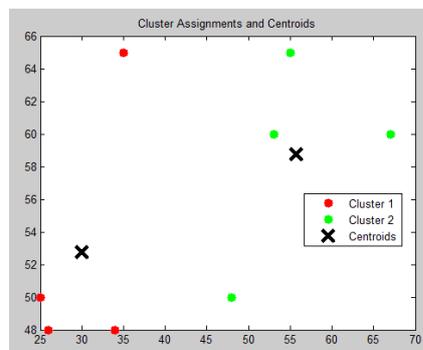
Tabel 2.16: *Centroid* akhir (*final centroid*)

<i>centroid</i> ke-	umur	tinggi
1	30	52,75
2	55,75	58,75

Tabel 2.17: Jarak *Squared Euclidean* dan anggota tiap *cluster* (Iterasi keempat)

Objek ke-	Jarak ke c_1	jarak ke c_2	Anggota <i>cluster</i> ke-
1	58,06	846	1
2	76,06	831,1	1
3	110,5	354,1	1
4	1387,06	204,6	2
5	710,5	44,12	2
6	547,06	1,625	2
7	76,06	467,125	1
8	357,06	99,12	2

Plot hasil *K-Means++* clustering :

Gambar 2.4: Plot *K-Means++* cluster

2.3 Teknik Reduksi Dimensi

Ada banyak metode *data mining* yang dapat bekerja dengan baik pada data berdimensi rendah, sedangkan set data dalam *data mining* umumnya

berdimensi tinggi yang memiliki banyak fitur sehingga reduksi dimensi data penting dilakukan. Jika dikurangi dimensionalitas data secara langsung maka kemungkinan ada karakteristik data yang hilang bisa saja fitur yang di buang adalah termasuk fitur yang berpengaruh. Oleh karena itu, diperlukan solusi dimana mereduksi dimensi tanpa harus kehilangan karakteristik data.

Menurut Maaten(2007), Reduksi dimensi adalah transformasi data yang memiliki dimensi tinggi menjadi sebuah representasi dengan data yang lebih sedikit dimensinya. Reduksi dimensi memiliki manfaat dalam pengolahan data. Reduksi dimensi dapat menghilangkan features yang tidak relevan dan mengurangi *noise* serta mengatasi *curse of dimensionality*. *Curse of dimensionality* adalah sebuah fenomena dimana analisis data menjadi hal yang sulit dikarenakan oleh jumlah dimensi data yang meningkat (Tan, 2007). Teknik Reduksi dimensi dapat dibagi menjadi 2 yaitu *feature extraction* dan *feature selection* (Fang et al., 2015). Pada penelitian ini menggunakan teknik reduksi-dimensi untuk *feature extraction*.

Feature extraction adalah salah satu kategori dari reduksi dimensi. *Feature extraction* bertujuan untuk membuat sebuah kumpulan fitur-fitur baru dengan memetakan fitur-fitur asli pada data. Salah satu teknik *unsupervised* pada *feature extraction* yang terkenal adalah *principal component analysis* (PCA)(Cunningham, 2008).

2.3.1 Algoritma *Principal Component Analysis*

Principal Component Analysis (PCA) adalah alat penemuan yang dirancang untuk mengurangi dimensi dan mengidentifikasi kecenderungan yang tidak diketahui dalam satu set data multidimensi. Langkah-langkah dalam teknik reduksi menggunakan PCA adalah sebagai berikut:

1. Hitung nilai vektor rata-rata dataset (μ).

2. Lakukan *Zero Mean* atau normalisasi data terhadap dataset dengan menggunakan rumus,

$$Y = \chi - \mu \quad (2.13)$$

3. Selanjutnya menghitung matriks kovarian dengan mensortir nilai dari besar ke kecil, dengan menggunakan rumus,

$$A = \left(\frac{1}{n-1} \right) Y^t Y \quad (2.14)$$

4. Hitung vektor eigen dan nilai eigen. Prosedur penyelesaiannya adalah sebagai berikut:

- Misal diketahui matrik A berukuran $m \times m$ (matriks simetris).
- Mencari nilai eigen (λ) dari matrik A . Dimana determinan dari matriks A dikurangi (λ) dikalikan dengan matrik identitas (I) sama dengan 0.

$$IAX = I\lambda X \quad (2.15)$$

$$(A - \lambda I)X = 0 \quad (2.16)$$

persamaan di atas terpenuhi jika dan hanya jika:

$$\det [A - \lambda I] = 0 \quad (2.17)$$

- Banyaknya nilai eigen (λ) yang akan diperoleh sama dengan ukuran matriks A yaitu sebanyak m .
- Setelah diketahui nilai-nilai λ , langkah selanjutnya adalah mencari vektor eigen untuk masing-masing (λ). Vektor eigen diperoleh melalui rumus:

$$(A - \lambda I)X = 0$$

Sehingga nanti akan diperoleh persamaan x dalam bentuk x_1, x_2 hingga x_m .

$$a_1x_1 + a_2x_2 + \dots + a_mx_m = 0$$

- Selajutnya dilakukan penormalan untuk masing-masing nilai eigen sehingga diperoleh $x_1^*, x_2^*,$ hingga x_m^* kemudian digabung menjadi matriks:

$$X = [x_1^* x_2^* \dots x_m^*]$$

atau

$$X = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{m1} \\ x_{21} & x_{22} & \dots & x_{x_{m2}} \\ \cdot & \cdot & \dots & \cdot \\ x_{m1} & x_{m2} & \dots & x_{mm} \end{bmatrix}$$

5. Nilai Komponen Utama atau PC dapat diperoleh dengan mengalikan matriks vektor eigen dengan matriks dataset yang sudah distandarisasi (matriks Y).

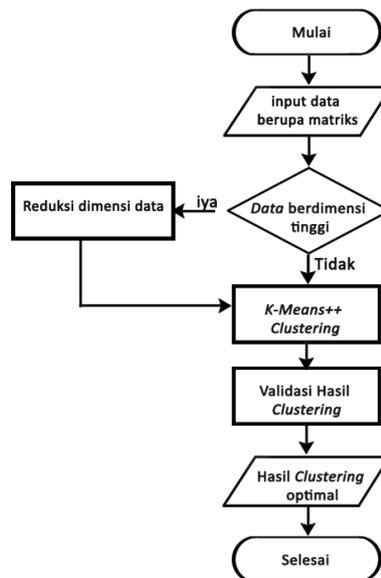
BAB III

PEMBAHASAN

Pada bab ini akan dibahas mengenai bagaimana langkah-langkah optimasi *cluster K-Means++* menggunakan teknik reduksi dimensi *Principal Component Analysis*.

3.1 Langkah-Langkah Optimasi *Cluster*

Hasil *cluster* yang optimal diperoleh setelah melakukan beberapa proses. Berikut adalah langkah-langkah yang dilakukan dalam penelitian yang harus dikerjakan:



Gambar 3.1: Proses *clustering* menggunakan PCA

Berdasarkan *flowchart* mengenai optimasi *cluster*, maka dapat dijelaskan sebagai berikut:

- Memulainya dengan memasukkan data berupa matriks
- Tahap selanjutnya terdapat dua proses perhitungan, yaitu:
 1. Perhitungan pertama, data berdimensi tinggi direduksi dimensinya menggunakan PCA untuk menghasilkan data yang lebih sedikit dimensi atau variabelnya.
 2. Perhitungan kedua, data berdimensi tinggi langsung di proses menggunakan *K-Means++* tanpa proses reduksi dimensi.
- Pada proses *K-Means++* terdiri dari dua proses utama yaitu inisialisasi *centroid* dilanjutkan dengan proses *K-Means*.
- Tahapan terakhir adalah validasi *cluster* dengan menggunakan koefisien *Silhouette*

3.2 Reduksi Dimensi Menggunakan PCA

Sebuah dataset merupakan sekumpulan dari objek-objek data yang terdiri dari beberapa atribut atau dimensi data. Tiap dimensi data memiliki tipe data berbeda antara satu dimensi dengan lainnya. Suatu dataset dikatakan berdimensi tinggi jika dataset tersebut memiliki atribut yang banyak (minimal 4). Data set ekspresi gen, data set teks dokumen, data set *image* adalah beberapa contoh dataset yang memiliki dimensi tinggi. Pada data ini, biasanya terdapat fenomena yang disebut *Curse of Dimensionality*. Oleh karena itu, diperlukan PCA yang bisa digunakan untuk mereduksi dimensi suatu data tanpa mengurangi karakteristik data tersebut secara signifikan.

3.2.1 Data Penelitian

Data yang digunakan pada penelitian ini adalah data simulasi dataset *ratings*. Dataset ini merupakan data sekunder yang didapatkan dari UCI (University of California) *Machine Learning Repository*. Data set *ratings* memiliki 329 objek kota dan 9 variabel atau kategori yang merujuk pada kualitas kehidupan kota di Amerika Serikat yang terdiri dari variabel *climate*, *housing*, *health*, *crime*, *transportation*, *education*, *arts*, *recreation* dan *economics* yang terdapat pada Lampiran 6. Data set *ratings* terdapat juga pada aplikasi Matlab. Berikut cara memanggil data *ratings* pada aplikasi Matlab dengan menggunakan perintah "load":

```
<<load cities
```

Setelah menggunakan perintah seperti diatas pada command window Matlab, maka akan terdapat tiga *output* yang terdiri dari *categories*, *names* dan dataset *ratings*. Dimana *categories* berisi 9 variabel yaitu:

No	variabel	No	variabel
1	<i>Climate</i>	6	<i>Education</i>
2	<i>Housing</i>	7	<i>Arts</i>
3	<i>Health</i>	8	<i>Recreation</i>
4	<i>Crime</i>	9	<i>Economics</i>
5	<i>Transportation</i>		

3.2.2 Langkah-langkah Reduksi Dimensi PCA

Langkah-langkah yang dilakukan untuk reduksi dimensi PCA menggunakan matlab adalah sebagai berikut:

1. Nilai rata-rata dataset berupa vektor berukuran 1×9 .

Tabel 3.1: Rata-rata dataset

variabel	rata-rata
<i>Climate</i>	538,7325
<i>Housing</i>	8346,559
<i>Health</i>	1185,739
<i>Crime</i>	961,0547
<i>Transportation</i>	4210,082
<i>Education</i>	2814,888
<i>Arts</i>	3150,884
<i>Recreation</i>	1845,957
<i>Economics</i>	5525,365

2. *Zero Mean* (Y) atau normalisasi data terhadap dataset menggunakan rumus $Y = \chi - \mu$ sehingga diperoleh matriks berukuran 329×9 yang terdapat pada Lampiran 7.

3. Matriks kovarians yang berukuran 9×9 diperoleh dari :

$$A_{9 \times 9} = \left(\frac{1}{329 - 1} \right) Y^t_{9 \times 329} * (Y)_{329 \times 9}$$

$$A = 10^3 \times \begin{bmatrix} 15 & 111 & 26 & 8 & 14 & 3 & 127 & 21 & -13 \\ 111 & 5689 & 1084 & 114 & 941 & 151 & 4967 & 814 & 697 \\ 26 & 1084 & 1006 & 109 & 685 & 158 & 4031 & 264 & 75 \\ 8 & 114 & 109 & 128 & 149 & 9 & 646 & 99 & 101 \\ 14 & 941 & 685 & 149 & 2106 & 156 & 3131 & 428 & 93 \\ 3 & 151 & 158 & 9 & 156 & 103 & 556 & 20 & 42 \\ 127 & 4967 & 4031 & 646 & 3131 & 556 & 21551 & 1420 & 381 \\ 21 & 814 & 264 & 99 & 428 & 20 & 1420 & 653 & 152 \\ -13 & 697 & 75 & 101 & 93 & 42 & 381 & 152 & 1176 \end{bmatrix}$$

4. *Eigen value* dan *eigen vector* dari matriks varians kovarians.

$$D = 10^4 \times \begin{bmatrix} 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 67 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 93 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 241 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 478 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1076 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1638 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4408 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24414 \end{bmatrix}$$

Sebelumnya pada Bab 2 matriks kovarian mensortir nilai dari besar ke kecil. Karena elemen-elemen pada matriks kovarian dan matriks nilai eigen bersesuaian maka untuk melihat nilai kovarians yang berurutan dapat dilihat juga dari nilai matriks nilai eigen yang berurutan.

$$D = 10^4 \times \begin{bmatrix} 24414 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4408 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1638 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1076 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 478 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 241 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 93 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 67 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 \end{bmatrix}$$

Berikut merupakan matriks vektor eigen yang dihasilkan dari matriks nilai eigen. Matriks vektor eigen ini yang akan dikalikan dengan matriks yang distandarisasi untuk dijadikan PC.

$$\begin{bmatrix} -0,995 & 0,042 & -0,081 & -0,001 & 0,016 & -0,026 & 0,007 & -0,015 & 0,006 \\ 0,023 & 0,012 & -0,027 & -0,049 & -0,084 & -0,178 & 0,083 & -0,937 & 0,269 \\ -0,001 & -0,241 & -0,137 & 0,929 & -0,159 & -0,027 & -0,028 & 0,021 & 0,178 \\ 0,088 & 0,267 & -0,945 & -0,054 & 0,116 & 0,099 & -0,038 & 0,011 & 0,028 \\ -0,009 & -0,042 & 0,014 & -0,092 & -0,147 & -0,038 & -0,972 & -0,019 & 0,149 \\ 0,017 & 0,929 & 0,241 & 0,253 & -0,106 & 0,022 & -0,042 & 0,001 & 0,025 \\ -0,001 & 0,016 & 0,043 & -0,168 & 0,009 & 0,028 & 0,151 & 0,282 & 0,931 \\ 0,005 & 0,019 & 0,127 & 0,173 & 0,954 & 0,069 & -0,150 & -0,104 & 0,070 \\ -0,033 & -0,054 & 0,070 & 0,005 & -0,102 & 0,975 & -0,013 & -0,173 & 0,025 \end{bmatrix}$$

Gambar 3.2: matriks vektor eigen

Nilai-nilai eigen atau nilai *latent* pada output Matlab berisi nilai varians yang menjelaskan terhadap *principal component* (PC) atau komponen utama. Setiap kolom pada PC mempunyai sampel varians yang sama terhadap baris *latent* yang berkoresponding.

$$latent = 10^4 \times \begin{bmatrix} 24414 \\ 4408 \\ 1638 \\ 1076 \\ 478 \\ 241 \\ 93 \\ 67 \\ 11 \end{bmatrix}$$

Berdasarkan nilai latent diperoleh nilai *explained* pada output matlab. Nilai ini merupakan nilai nilai eigen yang dibagi dengan total nilai nilai eigen atau berisi nilai varians yang menjelaskan dalam persen yang berkoresponding terhadap PC.

$$explained = \begin{bmatrix} 75,2903 \\ 13,5940 \\ 5,0516 \\ 3,3194 \\ 1,4752 \\ 0,7428 \\ 0,2862 \\ 0,2066 \\ 0,0338 \end{bmatrix}$$

Berdasarkan nilai *explained* pada baris pertama berkoresponding dengan kolom pertama PC yang dapat menjelaskan 75% varians keseluruhan dataset, begitu juga pada baris kedua varians menjelaskan sebesar 13% terhadap dataset begitu seterusnya. Sehingga berdasarkan nilai di atas, bisa diambil satu kolom PC (*PC1*) saja karena varians yang sudah menjelaskan adalah 75% atau mengambil *PC2* dengan 88.9% atau *PC4* dengan 97.5% varians yang menjelaskan.

5. Nilai Komponen Utama (*Principal Component*(PC))

Nilai Komponen Utama atau PC dapat diperoleh dengan mengalikan nilai vektor eigen dengan matriks dataset yang sudah distandarisasi (matriks Y). Misal untuk *PC1* maka kalikan kolom pertama matriks Y dengan kolom pertama matriks D atau $PC1 = B_{329 \times 1} \cdot Y_{329 \times 1}$. Jika menggunakan hingga *PC3* maka perhitungan menggunakan $PC3 = B_{329 \times 3} \cdot Y_{329 \times 3}$ sehingga dihasilkan matriks $PC3_{329 \times 3}$. Selanjutnya data yang akan digunakan untuk *clustering* adalah *PC1* yang terdapat pada Lampiran 8 mempunyai varians yang menjelaskan sebanyak 75% yang dianggap sudah mewakili informasi yang terkandung pada data.

3.3 *K-Means++ Clustering*

K-Means++ adalah salah satu yang paling populer dalam teknik *seeding* karena membuat proses *K-Means* berkumpul lebih cepat (dengan pengurangan jumlah dari iterasi). Mengenai *K-Means*, diberikan k anggota bilangan bulat dan himpunan n dataset $\chi \subset R^d$. Selanjutnya akan dipilih $k - centers$ (*centroid*) atau pusat *cluster* C sehingga dapat meminimalkan fungsi potensial,

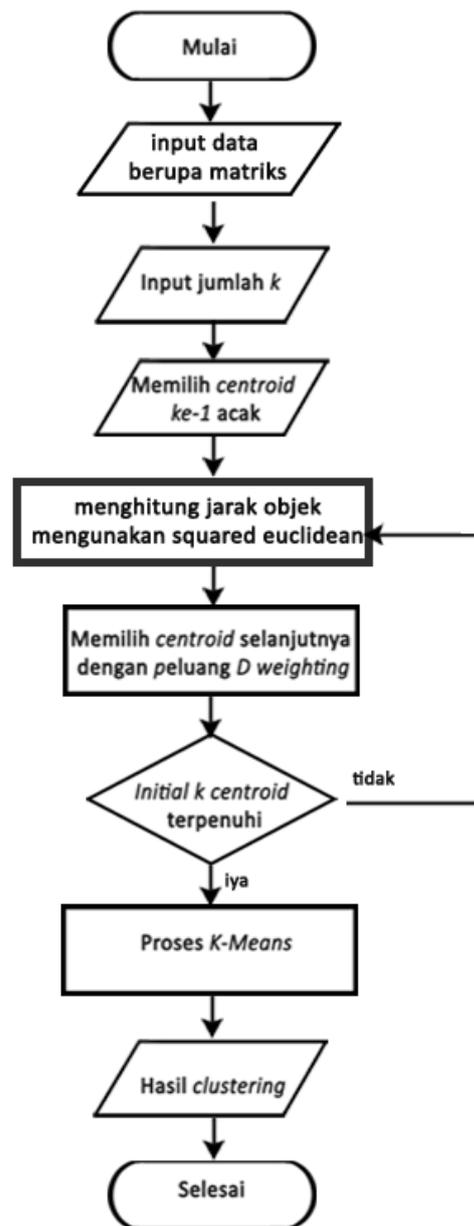
$$\phi = \sum_{x \in \chi} \min_{c \in C} \|x - c\|^2 \quad (3.1)$$

Berdasarkan pusat *cluster* ini dapat ditentukan *clustering* dengan mengelompokkan titik data berdasarkan *centroid*nya. Misal C_{OPT} adalah notasi dari *clustering* yang optimal dan ϕ_{OPT} adalah potensial yang berkoresponding. Diberikan *clustering* C dengan potensial ϕ maka dimisalkan juga $\phi(A)$ adalah notasi dari kontribusi dari $A \subset \chi$ hingga optimal.

Algoritma *K-Means++* telah terbukti $O(\log(k))$ -*competitive* dengan pengelompokan yang optimal (Arthur & Vassilvitski, 2007). Hal ini tidak menunjukkan bahwa hasil akan menjangkau global optimal, namun memberitahukan bahwa nantinya akan mendapatkan hasil cukup dekat dengan global optimal. Khususnya, jika ϕ adalah nilai yang diperoleh oleh *K-Means++* maka hasil tersebut tidak akan jauh dari optimal ϕ_{OPT} .

Berikut merupakan penjelasan dari langkah-langkah dalam menginisialisasi *centroid* awal menggunakan algoritma *K-Means++*. Dataset yang dipakai adalah dataset *ratings* dimana matriks berukuran 329×1 dan perhitungan kedua menggunakan dataset *PC1* dari dataset *ratings* dengan variansi yang menjelaskan sebesar 75% Dimana dataset merupakan matriks yang berukuran 329×1 . Dataset PC1 terlampir pada Lampiran 8.

Berikut merupakan *flowchart* dari algoritma *K-Means++*:

Gambar 3.3: Algoritma *K-Means++ Clustering*

Sebelum membahas perhitungan *K-Means++*, berikut merupakan perhitungan *clustering* menggunakan *K-Means* dimana tidak ada inisialisasi secara khusus melainkan pengambilan centroid secara acak.

1. Menentukan k sebagai jumlah *cluster* yang ingin dibentuk pada metode *K-Means*, dimana k yang akan dipilih adalah 3.
2. Menentukan *initial* pusat 3 *cluster* (*centroid*) secara acak

$$C = \{c_1, c_2, c_3\}$$

$$centroid = \begin{bmatrix} 618 & 9531 & 1348 & 756 & 6041 & 3489 & 1422 & 1704 & 6055 \\ 546 & 11014 & 2508 & 1067 & 3433 & 3346 & 7559 & 2288 & 4579 \\ 516 & 10509 & 1245 & 903 & 4900 & 3128 & 1360 & 1217 & 6470 \end{bmatrix}$$

3. Menghitung jarak antara *centroid* dengan objek *Squared Euclidean*

$$(d_{ij})^2 = \sum_{k=1}^p (x_{ik} - x_{jk})^2 \quad (3.2)$$

4. Alokasikan objek ke *centroid* terdekat dengan melihat jarak minimum yang diperoleh.

Langkah 3: jarak objek ke-i terhadap				Langkah 4	
i	c1	c2	c3	min(d)	anggota
1	19704065	82237497	21997181	19704065	1
2	25424000	16108648	30440962	16108648	2
3	21355476	74628584	19261300	19261300	3
...
328	20537854	42206534	23276974	20537854	1
329	20298911	73702325	17960411	17960411	3

Tabel 3.2: Langkah 3 dan langkah 4

5. Menentukan *centroid* baru dari rata-rata anggota tiap *cluster*, sehingga diperoleh:

$$C = \begin{bmatrix} 516,2 & 7322,6 & 926,3 & 936,1 & 4396,2 & 2801,3 & 1971,9 & 1762,5 & 5204,6 \\ 618,0 & 11192,1 & 2845,0 & 1176,9 & 5268,4 & 3084,7 & 10317,8 & 2535,6 & 5766,2 \\ 539,1 & 8820,3 & 740,7 & 884,5 & 3175,1 & 2681,5 & 1353,2 & 1609,0 & 6065,6 \end{bmatrix}$$

6. Kembali ke langkah 3 hingga objek dalam *cluster* dan *centroid* tidak berubah. Pada dataset ini, iterasi berhenti pada iterasi ke-18. Perolehan hasil *K-Means* sebagai berikut:

Tabel 3.3: Hasil *clustering K-Means*

<i>final centroid</i>		
Rata anggota c1	Rata anggota c2	Rata anggota c3
519,4722222	603,5555556	592,8115942
7543,460317	11462,22222	10752,26087
820,0555556	4602,333333	2058,449275
904,1230159	1351,444444	1104,130435
3882,960317	5940,555556	5118,057971
2753,93254	3032,555556	2968,318841
1551,035714	21513,44444	6553,028986
1649,119048	2602,666667	2439,391304
5423,452381	5076,111111	5876,086957
Jumlah anggota c1	Jumlah anggota c2	Jumlah anggota c3
252	28	69

Selanjutnya adalah perhitungan menggunakan metode *K-Means++*:

- Perhitungan pertama menggunakan matriks yang sudah tereduksi dimensinya yaitu dataset *PC1*.

1. Pilih pusat awal c_1 seragam secara acak dari X . Hitung jarak objek dengan jarak *Squared Euclidean* antara semua titik dalam dataset dan c_1 ,

Misal c_1 yang diambil adalah $c_1 = [-3446]$

2. Pilih pusat kedua c_2 dari X secara acak dari distribusi probabilitas $\frac{D_i^2}{\sum_j D_j^2}$. Hal tersebut diperoleh dengan langkah-langkah sebagai berikut:

- Pertama, hitung jarak semua titik dataset terhadap c_1 menggunakan jarak *squared euclidean* sehingga diperoleh vektor jarak

$$D = \begin{bmatrix} 0,0005 \\ 0,034 \\ 0 \\ 0,0267 \\ \dots \\ \dots \\ 0,0065 \\ 0 \end{bmatrix} \rightarrow \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ \dots \\ \dots \\ d_{328} \\ d_{329} \end{matrix}$$

Untuk setiap titik data, jika terdapat n elemen maka terdapat n jarak terhadap masing-masing objek terhadap *centroid*.

Secara keseluruhan tujuan dari *K-Means++* adalah untuk memilih *centroid* baru dari data yang jauh dari *centroid* yang terpilih. Sehingga meningkatkan peluang akan terpilihnya titik-titik data yang jauh dari pusat atau *centroid*.

- Kedua, jumlahkan n jarak untuk mendapatkan $\sum_j D_j^2$ atau S_{tot} . Untuk setiap titik data hitunglah jarak (d_i) dibagi dengan S_{tot} hingga memperoleh vektor yang berisi probabilitas terpilihnya titik data untuk dijadikan *centroid* baru.

$$P_{kumulatif} = \begin{bmatrix} 0,0001 \\ 0,0029 \\ 0,0029 \\ 0,0051 \\ \dots \\ \dots \\ 0,99 \\ 1,000 \end{bmatrix}$$

Catat bahwa total dari n probabilitas akan bertambah sampai ke angka 1. Setelah diperoleh $P_{kumulatif}$ pilih titik data yang proporsional terhadap p_i . Karena, sebagaimana menghitung p_i , titik-titik data yang jauh terhadap dari *centroid* manapun juga akan bisa terpilih dibandingkan titik-titik data yang mempunyai nilai p_i yang kecil. Hal tersebut dapat diimplementasikan dengan pemilihan proporsional peluang dengan membagi unit interval ($0 \rightarrow 1$) sepanjang p_i dan memilih secara seragam nilai antara 0 dan 1. Misal nilai peluang yang terambil secara seragam adalah 0,6999, maka dengan melihat nilai dari $P_{kumulatif}$ nilai tersebut jatuh pada data ke-213. Dengan demikian, *centroid* baru yang terpilih adalah data ke-131 atau x_{213} yaitu

$$c_2 = x_{213} = [53249]$$

3. Selanjutnya hitung ulang jarak objek dan pilih pusat c_l berturut-turut dan menghitung ulang jarak objek hingga k center telah dipilih. Dengan mengulang langkah ke-2 sehingga diperoleh *centroid* ke-3.

$$c_2 = x_{213} = [12336]$$

4. Langkah selanjutnya adalah standar algoritma *K-Means*. Pada proses ini, titik-titik data atau objek akan mengalami iterasi yang berulang sebanyak 13 kali hingga memperoleh anggota tiap *cluster* yang tidak berubah. Perolehan hasil *K-Means++* adalah sebagai berikut:

Tabel 3.4: Hasil *clustering K-Means++* dengan PC1

<i>final centroid</i>		
Rata anggota c1	Rata anggota c2	Rata anggota c3
-3446	53249	12336
Jumlah anggota c1	Jumlah anggota c2	Jumlah anggota c3
284	1	4

- Perhitungan kedua menggunakan dataset *ratings* tanpa melakukan reduksi dimensi.
 1. Pilih pusat awal c_1 seragam secara acak dari X . Hitung jarak objek dengan jarak *Squared Euclidean* antara semua titik dalam dataset dan c_1 , Misal c_1 yang diambil adalah :

$$\left[\begin{array}{ccccccccc} 564 & 6858 & 1099 & 1423 & 2904 & 2876 & 1077 & 2668 & 5390 \end{array} \right]$$

2. Pilih pusat kedua c_2 dari X secara acak dari distribusi probabilitas $\frac{D_i^2}{\sum_j D_j^2}$. Sebelumnya hitung jarak semua titik dataset terhadap c_1 menggunakan jarak *squared euclidean* sehingga diperoleh vektor jarak d_i yang berukuran 329×1 .

Selanjutnya jumlahkan n jarak untuk mendapatkan $\sum_j D_j^2$ atau S_{tot} untuk menghitung vektor P yang berisi nilai probabilitas hasil dari pembagian d_i dengan S_{tot} yang akan dijadikan *centroid* baru. Kemudian menghitung kumulatif dari P untuk memilih *centroid* baru

berdasarkan nilai peluangnya. Terpilih nilai 0,18 sehingga diperoleh c_2 pada objek ke-65 dengan nilai peluang 0,19104849 yaitu:

$$\left[\begin{array}{cccccccccc} 514 & 10913 & 5766 & 1034 & 7742 & 3486 & 24846 & 2856 & 5205 \end{array} \right]$$

3. Selanjutnya hitung ulang jarak objek dan pilih pusat c_i berturut-turut dan menghitung ulang jarak objek hingga k center telah dipilih. Dengan mengulang langkah ke-3 terpilih nilai 0,101 sehingga diperoleh c_3 pada objek ke-43 dengan nilai peluang 0,1179 yaitu:

$$\left[\begin{array}{cccccccccc} 623 & 11609 & 5301 & 1215 & 6801 & 3479 & 21042 & 3066 & 6363 \end{array} \right]$$

4. Langkah selanjutnya adalah standar algoritma *K-Means*. Pada proses ini, titik-titik data atau objek akan mengalami iterasi yang berulang sebanyak 8 kali hingga memperoleh anggota tiap *cluster* yang tidak berubah lagi seperti pada perhitungan pertama. Berikut merupakan hasil *clustering K-Means++*:

Tabel 3.5: Hasil *clustering K-Means++*

<i>final centroid</i>		
Rata anggota c1	Rata anggota c2	Rata anggota c3
526,8419244	603,5555556	629,7894737
7859,5189	11462,22222	12076,26316
918,0652921	4602,333333	3235,552632
934,7250859	1351,444444	1162,684211
4046,828179	5940,555556	5460,263158
2769,817869	3032,555556	3160,026316
2000,821306	21513,44444	11957,94737
1745,793814	2602,666667	2613
5471,725086	5076,111111	5936,131579
Jumlah anggota c1	Jumlah anggota c2	Jumlah anggota c3
291	6	32

3.4 Validasi Hasil *Clustering* Menggunakan *Silhouette Coefficient*

Koefisien *silhouette* didasarkan terhadap pertimbangan geometri mengenai pemisahan dan kohesi *cluster* oleh Rousseeuw dan Kaufman. Kohesi ini digunakan untuk mengukur kedekatan data yang terdapat pada satu *cluster*, sedangkan pemisahan digunakan untuk mengukur kedekatan antar *cluster* yang berbeda.

Kualitas hasil *cluster* (anggota *cluster*) dapat diketahui pada nilai *Silhouette Coefficient* (Carlo Vercellis 2009), untuk menghitungnya dapat menggunakan rumus jarak *Squared Euclidean*. Berikut langkah untuk menghitung nilai Koefisien *Silhouette*:

1. Untuk setiap titik i , hitung nilai tengah jarak objek i terhadap seluruh titik yang terdapat dalam satu *cluster*. Maka diperoleh nilai tengah jarak antar titik dalam satu *cluster* yang sama (a_i).
2. Untuk setiap titik i , hitung nilai tengah jarak titik i dengan seluruh titik yang terdapat di *cluster* lain. Dari semua jarak nilai tengah tersebut ambil nilai yang tekecil (b_i)
3. Setelah itu titik objek akan memiliki nilai koefisien *silhouette*. Koefisien *silhouette* dari tiap objek didefinisikan sebagai:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \quad (3.3)$$

untuk s_i yang memiliki nilai besar atau tinggi maka semakin tepat penempatan objek data terhadap *cluster*. Nilai koefisien *silhouette* biasanya di antara rentang -1 hingga 1. Nilai koefisien *silhouette* yang mendekati 1 adalah yang

lebih baik dibandingkan nilai koefisien *silhouette* yang mendekati -1. Nilai *Silhouette* didefinisikan sebagai rata-rata s_i dirumuskan sebagai berikut:

$$\frac{1}{N} \sum_{i=1}^N s_i$$

Hasil perhitungan *Silhouette Coefficient* menggunakan matlab adalah sebagai berikut:

1. Hasil *Silhouette Coefficient* dataset *PC1* dengan *K-Means++* :

$$Sillhoutte = \begin{bmatrix} 0,962949 \\ 0,662583 \\ 0,951709 \\ 0,783333 \\ \dots \\ \dots \\ 0,96534 \\ 0,951566 \end{bmatrix} \rightarrow \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_{i+1} \\ \dots \\ s_{328} \\ s_{329} \end{matrix}$$

Rata-rata yang diperoleh dari *Silhouette Coefficient* diatas adalah 0,863466015.

2. Hasil *Silhouette Coefficient* dataset *ratings* dengan *K-Means* :

$$Sillhoutte = \begin{bmatrix} 0,822388 \\ -0,12716 \\ 0,866015 \\ -0,27643 \\ \dots \\ \dots \\ 0,799302 \\ 0,856081 \end{bmatrix} \rightarrow \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_{i+1} \\ \dots \\ s_{328} \\ s_{329} \end{matrix}$$

Rata-rata yang diperoleh dari *Silhouette Coefficient* diatas adalah 0,6473.

3. Hasil *Silhouette Coefficient* dataset *ratings* dengan *K-Means++* :

$$\text{Silhoutte} = \begin{bmatrix} 0,93027209 \\ 0,8374394 \\ 0,9411364 \\ 0,85237667 \\ \dots \\ \dots \\ 0,93623827 \\ 0,93924946 \end{bmatrix} \rightarrow \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_{i+1} \\ \dots \\ s_{328} \\ s_{329} \end{matrix}$$

Rata-rata yang diperoleh dari *Silhouette Coefficient* diatas adalah 0,79229.

Metode	Rata-rata dari koefisien <i>Silhouette</i>
<i>K-Means</i>	0,647373
<i>K-Means++</i>	0,79229
<i>PC dengan K-Means++</i>	0,86346

Tabel 3.6: Perbandingan hasil validasi koefisien Silhouette

Berdasarkan hasil validasi koefisien *Silhouette* bisa disimpulkan bahwa untuk data *ratings* memiliki hasil *clustering* yang lebih optimal menggunakan algoritma *K-Means++* berbasis teknik dimensi reduksi PCA .

BAB IV

PENUTUP

4.1 Kesimpulan

1. Cara kerja dari PCA adalah mereduksi dimensi data dan merangkumnya ke dalam dimensi yang lebih sedikit namun tidak kehilangan karakteristiknya, sedangkan cara kerja algoritma *K-Means++* adalah tidak memilih k *centroid* awal secara acak, hanya memilih satu *centroid* secara acak dilanjutkan dengan memilih *centroid* yang lain hingga c_k dengan menggunakan nilai peluang,

$$P = \frac{D_i^2}{\sum_j D_j^2}$$

sehingga semua titik data yang jauh dari *centroid* sama-sama memiliki peluang terpilihnya sebagai *centroid* yang baru. Hal itu menyebabkan algoritma ini memberikan hasil yang lebih baik dibandingkan algoritma *K-Means*.

2. Evaluasi hasil *clustering* menunjukkan bahwa *K-Means++* berbasis PCA memiliki hasil yang lebih baik dibandingkan *K-Means++* dan *K-Means* itu sendiri. Hal ini berarti reduksi dimensi menggunakan PCA berpengaruh untuk mereduksi variabel yang tidak terlalu berpengaruh dan mengoptimalkan hasil *cluster*. Hal itu ditunjukkan pada Tabel 3.6 dengan nilai rata-rata *Silhouette Coeficient* yang lebih besar yaitu 0,863466015. Sedangkan *K-Means++* menunjukkan hasil *clustering* yang lebih baik dibandingkan dengan algoritma *K-means* sendiri. Hal itu karena adanya

D^2 weighting dan ditunjukkan dengan nilai rata-rata *Silhouette Coefficient* yang lebih besar yaitu 0,792295313 dimana nilai rata-rata *Silhouette Coefficient K-Means* hanya 0,647373642.

4.2 Saran

Untuk penelitian selanjutnya diharapkan:

- Menambahkan metode *clustering* untuk perbandingan dengan metode yang lain misal *Fuzzy clustering*.
- Menggunakan dataset berdimensi tinggi yang lain, misal dataset *image*.

DAFTAR PUSTAKA

- Afrin, dkk. 2014. "Comparative Performance of Using PCA with K-Means and Fuzzy C Means Clustering for customer Segmentation", *International Journal of Scientific and Technology* Vol. 4 No. 10. Hal. 397-400.
- Agusta, Yudi. 2005. "Mixture Modelling Menggunakan Prinsip Minimum Message Length", *Jurnal Sistem dan Informatika* Vol. 1. Hal.1 - 16
- Arthur, David dan Sergei Vassilvitskii. 2007. k-means++: The Advantages of Careful Seeding. *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, Hal.111.
- Berry. Linoff. 2004. *Data Mining Techniques for Marketing Sales and CRM, Second Edition*. chapter 17. Hal. 539-595.
- Celebi. Hasan dan Patricio. 2013. "A comparative study of efficient initialization methods for the k-means clustering algorithm", *Expert Systems with Applications* Vol. 40. Hal. 200-210.
- Connolly, Thomas M dan Carolyn Begg.2010. *Database Systems A Practical Approach to Design, Implementation, and Management, Fifth Edition*. America: Pearson Education.
- Han, Jiawei, dkk. 2006. *Data Mining: Concepts and Techniques, Second Edition*. United States of America: The Morgan Kaufmann, Inc.
- Han, Jiawei, dkk. 2011. *Data Mining: Concepts and Techniques, Third Edition*. United States of America: The Morgan Kaufmann, Inc.
- Irfan, Abbas. 2012." Integrasi Algoritma Singular Value Decomposition (SVD) dan Principal Component Analisis (PCA) untuk Pengurangan Dimensi pada Data Rekam Medis". *tugas akhir*. Universitas Ichsan Gorontalo.

- Jain, Anil k. 2010. "Model-Based Clustering of High-Dimensional Data: A review ", *Pattern Recognition Letters Journals* vol.31. hal.651666.
- Maaten, Laurens van der.2009."Feature Extraction from Visual Data".*thesis*, Tilburg University, The Netherlands
- Janecek, Mag. Andreas . 2009." Efficient Feature Reduction and Classification Methods". *disertasi*. University of Wien.
- Karch, Grzegorz. 2010." GPU-based acceleration of selected clustering techniques ". *thesis*. Silesian University of Technology in Gliwice.
- Larose, Daniel T.2006.*Data Mining Methods and Models*. United States of America: John Wiley & Sons, Inc.
- Prasetyo, Eko. 2012. Data Mining: Konsep dan Aplikasi menggunakan MATLAB. Yogyakarta: ANDI Yogyakarta. Bab 5. Hal. 213-214.
- Sharma, Subhash. 1996. *Applied Multivariate Techniques*. United States of America: John Wiley & Sons, Inc.

LAMPIRAN 1: KODE MATLAB ALGORITMA K-MEANS++

```

function[centroid,centroids,idx,SSE,koef,Perubahanjarakminimum,
Perubahantotaljarak] = runKMeans(X,K)
% Inisialisasi centroid
fprintf('Initializing centroids')
centroids=initiCentroids(X,K);
centroid=centroids
fprintf('...done\n\n')
idx=ones(size(X,1),1)
last=zeros(size(X,1),1)
num=0
while any(idx~=last)
    num=num+1 %menghitung banyaknya iterasi
    fprintf('iteration number %d\n\n',num)
    last=idx;
    [idx,SquaredEuclid,minSquaredEuclid,SSE]=findClosestCentroids(X, centroids);
    % menentukan centroid terdekat, Jarak, jarak minimum,total jarak %minimum
    Perubahanjarakminimum(:,num)=minSquaredEuclid
    Perubahantotaljarak(:,num)=SSE
    PerubahanAnggotacluster(:,num)=idx
    % u/ menghitung centroid baru
    [centroids,members]=computeNewCentroids(X, idx, K);
    PerubahantotalAnggota(num,:)=members
    any(last==idx)
end
%untuk menghitung final cost dari final cluster

```

```

silhouette(X,idx)
s = silhouette(X,idx,'sqeuclidean')
koef=mean(s)

```

LAMPIRAN 2: KODE MATLAB MENENTUKAN CENTROID BARU

```

function [centroids,members] = computeNewCentroids(X, idx, K)
%menghitung centroid baru u/ setiap cluster berdasarkan rata2 nilai
% Initialize variables
[m n] = size(X);
centroids = zeros(K, n);
for i=1:K
    temp = find(idx==i)
    Xtemp = X(temp,:);
    % u/ mendapatkan semua titik yg termasuk dlm cluster tsb
    centroids(i,:) = (sum(Xtemp,1))./length(Xtemp);
    % menghitung centroid baru dari mean
    members(:,i)=length(Xtemp)
end

```

LAMPIRAN 3: KODE MATLAB MENGALOKASIKAN OBJEK KE CENTROID TERDEKAT

```

function [idx]=findClosestCentroids(X, centroids)
%menentukan centroid terdekat untuk setiap titik berdasarkan jarak
%euclidean
K = size(centroids, 1);

```

```

idx = zeros(size(X,1), 1); %mengembalikan index dari centroid terdekat
for i=1:size(X,1)
    temp = X(i,:);
    SquaredEuclid(i,:)=sum(((bsxfun(@minus,temp,centroids)).^2),2);
    minSquaredEuclid(i,:)=min( SquaredEuclid(i,:));
    [~,idx(i,1)] = min( SquaredEuclid(i,:));
    SSE=sum( minSquaredEuclid);
end

```

LAMPIRAN 4: KODE MATLAB INISIALISASI CENTROID K-MEANS++

```

function InitialCentroids =initalCentroids(X,K)
M=X';
[b,k]=size(M);
L = [];
L1 = 0;
while length(unique(L)) ~= K
    % The k-means++ initialization.
    C = M(:,1+round(rand*(size(M,2)-1)));
    L = ones(1,size(M,2));
    for i = 2:K
        f = M-repmat(C(:,i-1),1,k);
        Ds(i-1,:)=dot(f,f,1);%pengurangan tiap titik ke centroid
        [ba,ko]=size(Ds);
        for ii=1:ko;
            Dst=Ds';
            Dsmin(ii,:)=min(Dst(ii,:));

```

```

end

JarakSqEucl=Dsmin
Dsminim= Dsmin';
D = cumsum(Dsminim);%nilai kumulatif dari jarak euclidean
Dkum=D';
Prob=D/D(end);
Peluang=(Dsminim/D(end))'
Peluangkum=Prob'
r=rand
choosen=find(r < Prob ,1)
if D(end) == 0, C(:,i:K) = M(:,ones(1,K-i+1)); return; end
C(:,i) = M(:,choosen) %untuk menentukan centroid secara random bergantung
[~,L] = max(bsxfun(@minus,2*real(C'*M),dot(C,C,1).'));
end
InitialCentroids=C'
X=M';
end

```

LAMPIRAN 5: KODE MATLAB REDUKSI DIMENSI PCA

```

function [hasilleft,penjelas] = pcared(A)
[n m] = size(A);
AMean = mean(A);
AStd = std(A);
B =A - repmat(AMean,[n 1]);
covB=cov(B);

```

```

[V,D,W] = eig(covB);
[V D] = eig(cov(B));
D2=diag(sort(diag(D),'descend'))
[c, ind]=sort(diag(D),'descend');
V2=V(:,ind)
W2=W(:,ind)
hasilleft=B*(W2*(-1))
hasil= B * V2;
jumlahvarpca= cumsum(diag(D2)) / sum(diag(D2))*100;
[a,b]=size(jumlahvarpca);
    penjelas= jumlahvarpca;
for ii=2:a
    penjelas(ii,:)= (jumlahvarpca(ii,:)-jumlahvarpca(ii-1,:));
end
disp(penjelas)
    penjelas1=sum(penjelas);
% [coeff,score,latent,tsquared,explained,mu] = pca(A); cara langsung
end

```

LAMPIRAN 6: *Data Ratings*

Keterangan:

1. var1 : *Climate*
2. var2 : *Housing*
3. var3 : *Health*
4. var4 : *Crime*
6. var5 : *Transportation*
7. var6 : *Education*
8. var7 : *Arts*
9. var8 : *Recreation*
10. var9 : *Economics*

NO	var1	var2	var3	var4	var5	var6	var7	var8	var9
1	6200	237	923	4031	2757	996	1405	7633	0
2	575	8138	1656	886	4883	2438	5564	2632	4350
3	468	7339	618	970	2531	2560	237	859	5250
4	476	7908	1431	610	6883	3399	4655	1617	5864
5	659	8393	1853	1483	6558	3026	4496	2612	5727
6	520	5819	640	727	2444	2972	334	1018	5254
7	559	8288	621	514	2881	3144	2333	1117	5097
8	537	6487	965	706	4975	2945	1487	1280	5795
9	561	6191	432	399	4246	2778	256	1210	4230
10	609	6546	669	1073	4902	2852	1235	1109	6241
11	885	16047	2025	983	3954	2843	5632	3156	6220
12	195	12175	601	1223	5091	2414	2346	3000	7668
13	530	5704	580	878	2865	2469	430	838	3370
14	591	5725	820	975	2707	2772	169	613	4262
15	546	11014	2508	1067	3433	3346	7559	2288	4579
16	560	5530	598	1125	3051	2189	268	1165	4730
17	396	7877	833	525	3298	2844	1166	2315	5275
18	694	6722	1204	566	5086	2990	1391	1542	5196
19	601	6691	605	933	1866	2646	3546	1001	5193
20	696	8316	3195	1308	8409	3057	7559	1362	6315
21	615	11074	637	1878	3556	2929	621	2711	8107
22	534	6292	1798	872	2523	2915	1047	913	5431
23	474	10384	1203	821	3943	2208	3857	1800	5097
24	435	8831	782	1049	3670	3063	5355	1063	7439
25	560	8068	420	1561	3725	2564	1222	1568	6056
26	567	9148	3562	1730	7405	3471	9788	2925	5503
27	451	7277	780	651	5613	2934	1995	2148	5172
28	427	8083	342	1565	3329	2635	4237	1413	6308
29	527	6342	900	1031	4652	2483	354	1648	4008
30	423	6288	616	1313	2782	2745	1795	1813	6019
31	586	7866	861	310	2960	2535	1284	1480	3119
32	772	8329	240	825	3776	2778	1302	3200	4247
33	566	6761	570	1190	2989	2545	79	1477	3635
34	559	14607	2661	857	3511	3653	9304	1918	6016
35	452	8315	479	810	6285	3008	778	2046	5913
36	584	6458	441	810	2516	2592	679	2106	5801
37	550	8257	1007	415	4529	3052	1599	1722	5614
38	612	6811	1692	1123	5177	2851	3958	1234	4843
39	149	8365	804	413	4303	2686	1211	1630	6019
40	558	7056	731	657	1746	2873	2152	1990	4829

NO	var1	var2	var3	var4	var5	var6	var7	var8	var9
41	487	8654	815	673	5889	2854	1470	1605	5863
42	592	8221	453	880	6575	2391	2385	1672	4633
43	623	11609	5301	1215	6801	3479	21042	3066	6363
44	459	11914	962	1088	7108	2587	3663	4012	7127
45	440	8242	333	1093	3805	2712	154	1349	7437
46	423	8394	438	768	2391	2718	1506	1512	6020
47	808	9060	310	651	1670	2544	382	1973	5671
48	648	13429	2550	943	3197	3029	8368	1913	7197
49	516	10041	975	545	4495	2628	514	777	6527
50	575	8263	916	1336	3810	2729	2001	1217	6900
51	440	5376	91	974	3119	2413	162	3000	4968
52	383	8228	640	1016	2530	2973	2002	1413	8040
53	571	8064	2465	971	5384	3121	8567	2441	5047
54	637	6179	994	707	1910	2519	131	701	5680
55	383	9673	1809	494	7146	3323	1741	3357	6726
56	575	7332	443	650	4279	2754	989	1157	4847
57	401	9839	345	989	4410	2453	303	1435	6303
58	434	7774	837	714	5270	2619	904	1501	5009
59	525	8627	672	1022	7447	3147	2203	1700	5485
60	569	7402	1463	1495	4207	3164	2993	2561	5153
61	627	7789	708	721	5470	2894	2605	844	5257
62	644	7169	999	1273	6099	3031	4313	1236	5671
63	618	9531	1348	756	6041	3489	1422	1704	6055
64	576	6189	564	946	3401	2415	2483	1238	4487
65	514	10913	5766	1034	7742	3486	24846	2856	5205
66	603	8587	243	947	4067	3126	1647	1543	5307
67	584	8143	2138	978	5748	2918	9688	2451	5270
68	544	6007	446	736	2226	2654	111	2219	4880
69	579	9168	3167	1138	7333	2972	12679	3300	4879
70	526	8509	721	1086	3389	2754	1749	2375	7699
71	541	7702	1951	1065	3893	2377	2882	1331	5147
72	526	7519	1421	1524	5859	2908	2489	1484	5279
73	517	5817	833	820	2995	2665	1861	1214	4812
74	558	8093	1837	1092	4364	2928	6648	2020	5165
75	362	6929	458	1335	3626	2840	1992	2037	6690
76	591	6054	760	337	3709	3363	373	1036	4741
77	544	9318	2825	1529	6213	3269	10438	2310	7710
78	569	14420	2350	548	2715	3029	7415	1572	7060
79	545	5709	593	379	3161	2943	85	501	4491
80	440	8083	1113	834	3907	2901	1017	1920	4997

NO	var1	var2	var3	var4	var5	var6	var7	var8	var9
81	544	7635	2253	1151	4775	2772	6935	1122	4532
82	561	7203	723	1347	4117	2612	809	3967	6592
83	480	7395	732	897	3867	2683	298	1222	4274
84	521	10789	2533	1365	8145	3145	8477	2324	7164
85	444	8028	1256	1044	5521	2613	1857	1802	5346
86	536	8525	4142	1587	4808	3064	10389	2483	3904
87	336	5708	593	930	2232	2230	117	714	5453
88	419	7993	640	571	3668	2701	340	1587	3949
89	193	6040	1159	488	5205	2619	2377	3107	3922
90	537	6501	444	1096	6539	2630	904	1610	6113
91	257	7078	798	433	3197	2960	1807	1397	5348
92	592	7343	528	1323	3705	2479	3800	1101	5080
93	521	6573	596	524	4168	2537	353	1023	4214
94	467	7078	562	582	3324	3000	1048	1600	4813
95	461	6829	626	845	2312	2764	215	1200	8268
96	605	7715	529	635	5754	2641	2032	1340	4299
97	741	9370	539	874	5293	3118	2631	3400	3045
98	550	6743	783	864	3496	2797	1876	1622	5206
99	643	9017	900	861	4602	2439	749	2005	4884
100	148	8168	920	503	6325	2506	2111	1414	5594
101	561	6274	872	1150	4402	3051	844	709	5255
102	549	6686	594	545	3581	2334	1915	1695	4631
103	507	8252	655	655	2244	2799	270	790	5098
104	540	7204	724	1671	4912	2511	2163	1355	3724
105	546	5962	607	516	1454	2427	1021	994	4492
106	552	6508	818	1334	4963	3109	628	800	4842
107	490	9951	731	744	2637	2413	1609	4200	6631
108	572	10810	1252	1536	4186	2734	2027	2455	7136
109	342	9298	546	787	4583	2729	380	4005	7166
110	602	8842	527	1422	2143	3154	368	2058	7973
111	482	5784	466	663	3092	2927	145	1736	4849
112	536	7554	484	544	2886	2809	87	2092	6342
113	509	6733	1060	710	5416	2772	2846	1711	4195
114	528	7956	1038	1348	4472	2627	6466	2366	6862
115	559	9291	369	1483	4388	2407	3596	2984	5746
116	526	5382	622	749	2174	2299	153	300	4220
117	402	7388	1731	1658	3527	3094	3335	1237	5739
118	727	7767	1437	1213	3423	2809	1756	3000	6026
119	483	7641	1364	996	5855	2526	4115	1940	3826
120	476	7120	43	568	2241	2674	603	1883	5166

NO	var1	var2	var3	var4	var5	var6	var7	var8	var9
121	105	7898	1109	401	5587	2721	1921	1304	5646
122	513	7780	1274	952	3454	2705	3255	1909	4848
123	410	7143	667	792	3747	2737	401	2176	4697
124	490	8218	706	994	1641	2854	1254	739	5443
125	367	8401	916	583	3793	2622	2547	1925	5650
126	626	7064	1694	967	4453	3090	4188	1651	5204
127	655	6336	1260	1185	3950	3236	2569	1410	5012
128	568	7763	818	627	3431	2990	825	1491	4477
129	542	8227	1135	892	3338	2747	2316	1604	4618
130	556	7891	2087	629	6164	3224	3083	1532	5322
131	516	11652	2521	1279	7120	3628	3616	1790	6307
132	623	6760	1006	765	2703	2726	188	797	4728
133	717	17021	1298	891	5911	2502	7168	3703	5187
134	427	7094	583	400	1145	1995	725	2700	6662
135	424	9760	2467	1499	4626	3271	11073	1825	7464
136	636	6632	875	665	4001	2525	2195	840	4383
137	600	6283	685	924	2661	2257	1921	1075	6412
138	557	7012	2243	1000	5804	2690	6348	1906	5082
139	434	9429	2437	830	2770	2842	2255	1506	5165
140	518	6794	679	1021	4800	2654	323	1933	3822
141	412	7245	1792	1091	4917	3130	3209	1427	5186
142	457	6626	1181	1211	5611	3048	2162	2884	6139
143	564	6111	740	967	1780	2646	567	1177	6386
144	466	7447	700	858	3092	2532	1092	1615	4371
145	601	8810	1759	1434	4982	2574	7420	1001	4889
146	663	6119	1152	424	2532	2925	1925	2155	4903
147	547	6524	731	353	4343	2691	666	903	4181
148	479	9327	1058	837	4645	2868	3177	1636	4631
149	580	5159	500	628	2335	2871	80	801	5324
150	527	7919	1043	1120	5419	2896	2071	2163	4794
151	483	7230	609	976	3444	2855	75	1119	5579
152	549	8126	1711	1142	5006	3028	1167	1045	6166
153	549	7076	1939	1468	5869	2949	5553	2043	4865
154	496	8516	1067	911	4473	2918	3844	2224	5176
155	365	6463	398	733	2862	3167	920	931	6331
156	670	6692	960	622	4273	2761	3309	2514	5537
157	512	6616	596	413	1817	2904	285	1000	3429
158	352	8310	686	676	6096	3027	1466	1953	5648
159	494	7778	655	465	4956	2945	2235	1814	4333
160	429	8572	548	1030	5268	2305	1772	1734	9702

NO	var1	var2	var3	var4	var5	var6	var7	var8	var9
161	469	6921	314	1093	3549	2336	1456	1855	5872
162	514	13282	1237	822	3422	2607	3746	2435	5755
163	307	6680	323	1373	3412	2998	309	2513	5594
164	559	8631	1111	413	3908	3097	1015	1147	5120
165	480	7907	1371	894	5557	2891	4206	1609	4747
166	424	6152	465	1050	3322	2827	150	702	5264
167	552	6962	588	1457	2989	2736	2804	1609	5341
168	556	9906	412	1913	5900	2241	1586	3996	6035
169	513	7497	621	1018	2931	2700	3150	1752	4573
170	548	10414	1202	909	3575	2479	2111	1879	6527
171	479	5850	477	1156	2366	2375	1280	1757	6105
172	490	6876	759	764	2941	2694	736	1853	4444
173	635	8340	1860	1055	4080	2861	3596	1403	6245
174	522	6986	741	855	4084	2629	1352	1428	4313
175	398	8256	775	789	5618	2878	4523	1804	4908
176	497	7270	1861	1328	4186	2581	2180	1462	5273
177	500	6608	509	976	2680	2816	334	834	6898
178	579	8309	1105	609	3629	2582	2565	1602	3301
179	885	13868	5153	1960	4345	3195	23567	3948	5316
180	616	6812	2111	937	5420	3028	4916	1942	5402
181	526	9640	1083	819	3820	2479	3057	1129	6651
182	604	6990	900	1608	4158	2545	3402	1702	5923
183	642	6934	732	643	4909	2803	1079	1439	4926
184	447	6235	593	783	3144	2651	1435	1204	5659
185	378	9897	2168	779	6084	3047	5123	1944	5448
186	404	9860	737	633	4595	2728	475	837	7101
187	558	6881	303	1072	2876	2871	554	1133	4386
188	238	5345	372	836	2117	2644	1231	1059	5739
189	611	9008	256	728	3512	2797	1856	1416	3692
190	582	8721	517	1039	2560	2814	437	3800	7089
191	514	7015	2043	1488	6247	2804	4486	1994	5160
192	634	10267	2314	2459	5202	2879	4837	4300	5840
193	559	12135	2589	691	4198	3539	8058	1596	6324
194	593	11652	884	646	4636	3128	730	1682	6307
195	603	8672	97	1166	5310	2416	438	1502	9980
196	460	10176	3053	826	4945	3044	8766	2902	4982
197	293	9559	3934	906	5606	3013	11714	2158	5843
198	442	6704	1469	1511	3345	2779	1764	2164	4565
199	639	8630	347	1154	2000	2616	631	833	5107
200	615	11660	2482	819	2690	2787	7563	3544	6154

NO	var1	var2	var3	var4	var5	var6	var7	var8	var9
201	455	6190	331	957	3606	2453	1528	1541	5537
202	483	6754	832	815	3509	2388	1374	1112	4892
203	530	5800	949	783	4325	2965	2498	1428	3980
204	580	6391	699	1537	3353	2630	529	2666	3708
205	538	10757	853	452	3320	2728	2122	1523	6962
206	600	7800	1850	984	5030	2763	4342	1849	5938
207	656	11138	3919	566	2119	3234	8640	3705	7371
208	643	8087	519	1012	3219	2439	766	1450	4937
209	516	10509	1245	903	4900	3128	1360	1217	6470
210	583	11460	2068	893	5938	3495	7852	1604	5478
211	583	10218	556	633	4505	3244	1164	2281	6672
212	498	8515	2586	1604	4579	2995	7978	3500	6453
213	638	13358	7850	2498	8625	2984	56745	3579	5338
214	601	14220	4106	1461	3514	3362	14224	1818	5690
215	554	7686	507	775	5561	2538	966	1873	4463
216	632	8568	1932	997	3215	3014	7087	2964	5866
217	648	20151	2530	625	3536	3029	7273	2268	6432
218	910	13135	2362	1533	6430	2646	6162	2394	5457
219	333	6750	489	1327	3798	2864	266	3095	7060
220	603	6689	384	1698	1944	2721	259	900	7565
221	554	7186	1623	1297	4459	2908	4843	1742	8119
222	726	8263	338	752	4083	2625	708	1451	4912
223	440	7128	2559	1008	5806	3069	3787	1977	5853
224	509	10173	1574	804	3299	2754	6987	1866	5822
225	457	8196	765	1671	5887	2976	2681	2881	7413
226	524	6760	812	568	2273	2491	804	852	6062
227	890	14000	1106	791	2238	2155	2769	2135	5514
228	536	6373	201	1344	2778	2500	755	2089	6083
229	617	6657	665	488	4399	2503	91	1148	5187
230	584	6248	593	591	1750	1701	155	1956	4491
231	586	9462	1117	744	4738	3058	480	1513	5154
232	536	6479	563	1472	2918	2914	1954	2160	6029
233	491	8388	1184	921	2967	2452	2294	1688	4073
234	630	8310	5158	1059	5903	3781	17270	1979	5638
235	536	8921	1584	1268	4729	2942	4573	2472	6415
236	463	5674	617	1169	1671	2554	373	793	4247
237	586	8099	3413	687	5616	3544	11069	2145	5261
238	482	7807	694	638	3759	3264	228	1420	5483
239	483	8100	834	823	5185	2973	2351	3366	6186
240	768	9912	1590	1504	5947	3343	5160	2532	4535

NO	var1	var2	var3	var4	var5	var6	var7	var8	var9
241	469	9966	596	475	2321	3026	165	1390	8367
242	488	9981	355	633	4166	2898	785	1670	6746
243	586	9274	2467	998	5474	3558	6152	2263	5154
244	500	9321	198	485	4546	2618	1985	3300	3459
245	497	6637	468	1181	3501	2653	1307	1619	4646
246	496	8943	931	1055	3558	2732	1171	2016	4415
247	647	8230	3476	981	6544	3455	5730	1606	6405
248	614	7614	1154	522	3120	3028	1108	1549	5587
249	664	8584	274	892	5727	2471	845	2424	4459
250	535	12449	615	1116	6767	2529	2210	2386	5677
251	664	8461	300	779	4714	1728	529	1204	5326
252	585	8343	2448	1076	6680	2940	5697	1943	5870
253	615	9754	2201	1475	3141	2596	5327	1918	4923
254	652	7476	1036	784	3872	2723	1263	2036	5287
255	308	9193	2966	437	4399	2134	769	1503	6099
256	536	8609	1969	894	5165	3582	6956	2659	6304
257	466	7584	969	1156	2987	2680	1026	933	4592
258	576	9855	1027	1363	5097	2793	4483	2306	5309
259	515	7368	1022	1068	3186	2772	1708	2059	3709
260	195	7235	603	343	4565	2502	1871	1572	5488
261	475	5589	223	969	2689	2927	879	1265	5991
262	537	7605	2850	1306	7119	3530	8896	2243	5800
263	716	8378	749	1014	4732	3278	691	1873	3835
264	644	11622	1232	490	3459	2729	3276	2234	6309
265	843	13838	352	1107	4160	2439	1004	3179	5656
266	541	9466	1631	969	6228	2340	5528	3900	4942
267	488	6321	236	1032	2938	2707	1301	1136	7720
268	398	6898	1337	1197	5387	2938	4295	1509	6873
269	903	14465	2416	1099	5489	2794	8818	3347	5489
270	910	17158	3726	1619	8299	3371	14226	4600	6063
271	850	16048	2117	1065	5224	2709	6446	1964	7270
272	855	15547	532	1026	5662	2719	2684	3300	5821
273	843	14303	1035	964	5010	2611	3748	1703	5335
274	732	12931	1052	912	3313	2722	3457	2255	5703
275	391	9560	801	939	3742	2626	817	2535	7715
276	542	6896	1084	1614	5958	2456	2262	2237	5591
277	575	6697	1219	372	3683	3230	1832	1386	4907
278	808	10183	2715	1170	6634	2710	9577	4800	5901
279	570	6697	700	384	2017	3022	52	1100	4055
280	442	8121	593	450	3458	2557	268	1316	4765

NO	var1	var2	var3	var4	var5	var6	var7	var8	var9
281	524	5722	394	1035	1922	2652	68	937	6213
282	508	6534	1445	1197	4401	2858	2826	1389	6585
283	385	6528	846	759	4316	2673	1393	1359	4648
284	276	7983	1041	556	6271	2651	465	1324	5204
285	545	5938	830	1038	5634	2874	2672	1819	5056
286	574	6927	497	869	5534	2774	2988	1517	4722
287	524	7882	1877	1225	6172	3078	1983	1536	5384
288	453	8039	710	1212	6159	3525	3466	1514	5289
289	544	6343	577	892	3828	2709	1634	1737	5932
290	648	23640	2610	835	3110	3029	7865	1729	6158
291	575	8405	612	540	2740	3169	1271	1200	4677
292	542	6578	505	418	1532	2672	147	1460	3744
293	625	8474	342	1395	4427	2155	1579	1630	5672
294	548	7670	1040	689	6951	3144	5080	2851	4474
295	808	7770	539	1162	4730	2546	4297	4000	4887
296	404	8029	370	1161	5530	2790	2181	1936	6021
297	440	7442	1189	1493	5588	3044	5040	2943	7256
298	557	5527	453	630	3550	3012	1226	1401	4353
299	467	5717	343	822	2537	2899	63	669	4772
300	518	7767	1738	998	5323	2852	4389	1952	4534
301	501	7110	1148	999	5348	2795	1632	1141	5464
302	636	10616	1372	1181	4786	3311	5029	1646	5772
303	589	8548	1259	1400	4397	2685	4889	3131	6147
304	530	7498	1581	1080	3758	2628	4248	2024	7115
305	470	6464	674	1014	4723	2390	1432	1090	4900
306	500	7298	672	955	3460	3283	404	631	7327
307	548	6744	391	400	4592	2970	858	1750	5226
308	821	10503	1079	964	4153	2498	2962	1559	5819
309	768	9015	517	752	3817	2332	1557	1464	4571
310	336	7143	260	1092	2407	2696	87	1410	7599
311	615	7295	807	1135	4133	2747	2097	1474	5023
312	543	7778	210	1132	3094	2128	511	2800	5563
313	412	6106	538	1166	3018	2867	1141	1248	6259
314	631	13724	4361	1317	8236	3635	21701	1578	6072
315	569	10024	1218	789	2434	2995	318	946	5656
316	347	7881	925	700	3351	2889	3000	1900	4407
317	308	7642	818	442	3496	2749	761	1654	4300
318	509	10512	375	1783	5201	3224	2888	3772	7992
319	542	6576	791	308	2450	3002	422	1271	4740
320	494	7061	806	1164	3933	2981	2987	1508	6036

NO	var1	var2	var3	var4	var5	var6	var7	var8	var9
321	456	6404	549	1179	2793	2747	599	1126	6805
322	558	7284	860	464	3097	2906	196	726	3288
323	597	7927	1445	1115	4532	3112	4545	1923	6174
324	564	6858	1099	1423	2904	2876	1077	2668	5390
325	562	8715	1805	680	3643	3299	1784	910	5040
326	535	6440	317	1106	3731	2491	996	2140	4986
327	540	8371	713	440	2267	2903	1022	842	4946
328	570	7021	1097	938	3374	2920	2797	1327	3894
329	608	7875	212	1179	2768	2387	122	918	4694

82	22,3	-1143,6	-462,7	385,9	-93,1	-202,9	-2341,9	2121,0	1066,6
83	-58,7	-951,6	-453,7	-64,1	-343,1	-131,9	-2852,9	-624,0	-1251,4
84	-17,7	2442,4	1347,3	403,9	3934,9	330,1	5326,1	478,0	1638,6
85	-94,7	-318,6	70,3	82,9	1310,9	-201,9	-1293,9	-44,0	-179,4
86	-2,7	178,4	2956,3	625,9	597,9	249,1	7238,1	637,0	-1621,4
87	-202,7	-2638,6	-592,7	-31,1	-1978,1	-584,9	-3033,9	-1132,0	-72,4
88	-119,7	-353,6	-545,7	-390,1	-542,1	-113,9	-2810,9	-259,0	-1576,4
89	-345,7	-2306,6	-26,7	-473,1	994,9	-195,9	-773,9	1261,0	-1603,4
90	-1,7	-1845,6	-741,7	134,9	2328,9	-184,9	-2246,9	-236,0	587,6
91	-281,7	-1268,6	-387,7	-528,1	-1013,1	145,1	-1343,9	-449,0	-177,4
92	53,3	-1003,6	-657,7	361,9	-505,1	-335,9	649,1	-745,0	-445,4
93	-17,7	-1773,6	-589,7	-437,1	-42,1	-277,9	-2797,9	-823,0	-1311,4
94	-71,7	-1268,6	-623,7	-379,1	-886,1	185,1	-2102,9	-246,0	-712,4
95	-77,7	-1517,6	-559,7	-116,1	-1898,1	-50,9	-2935,9	-646,0	2742,6
96	66,3	-631,6	-656,7	-326,1	1543,9	-173,9	-1118,9	-506,0	-1226,4
97	202,3	1023,4	-646,7	-87,1	1082,9	303,1	-519,9	1554,0	-2480,4
98	11,3	-1603,6	-402,7	-97,1	-714,1	-17,9	-1274,9	-224,0	-319,4
99	104,3	670,4	-285,7	-100,1	391,9	-375,9	-2401,9	159,0	-641,4
100	-390,7	-178,6	-265,7	-458,1	2114,9	-308,9	-1039,9	-432,0	68,6
101	22,3	-2072,6	-313,7	188,9	191,9	236,1	-2306,9	-1137,0	-270,4
102	10,3	-1660,6	-591,7	-416,1	-629,1	-480,9	-1235,9	-151,0	-894,4
103	-31,7	-94,6	-530,7	-306,1	-1966,1	-15,9	-2880,9	-1056,0	-427,4
104	1,3	-1142,6	-461,7	709,9	701,9	-303,9	-987,9	-491,0	-1801,4
105	7,3	-2384,6	-578,7	-445,1	-2756,1	-387,9	-2129,9	-852,0	-1033,4
106	13,3	-1838,6	-367,7	372,9	752,9	294,1	-2522,9	-1046,0	-683,4
107	-48,7	1604,4	-454,7	-217,1	-1573,1	-401,9	-1541,9	2354,0	1105,6
108	33,3	2463,4	66,3	574,9	-24,1	-80,9	-1123,9	609,0	1610,6
109	-196,7	951,4	-639,7	-174,1	372,9	-85,9	-2770,9	2159,0	1640,6
110	63,3	495,4	-658,7	460,9	-2067,1	339,1	-2782,9	212,0	2447,6
111	-56,7	-2562,6	-719,7	-298,1	-1118,1	112,1	-3005,9	-110,0	-676,4
112	-2,7	-792,6	-701,7	-417,1	-1324,1	-5,9	-3063,9	246,0	816,6
113	-29,7	-1613,6	-125,7	-251,1	1205,9	-42,9	-304,9	-135,0	-1330,4
114	-10,7	-390,6	-147,7	386,9	261,9	-187,9	3315,1	520,0	1336,6
115	20,3	944,4	-816,7	521,9	177,9	-407,9	445,1	1138,0	220,6
116	-12,7	-2964,6	-563,7	-212,1	-2036,1	-515,9	-2997,9	-1546,0	-1305,4
117	-136,7	-958,6	545,3	696,9	-683,1	279,1	184,1	-609,0	213,6
118	188,3	-579,6	251,3	251,9	-787,1	-5,9	-1394,9	1154,0	500,6
119	-55,7	-705,6	178,3	34,9	1644,9	-288,9	964,1	94,0	-1699,4
120	-62,7	-1226,6	-1142,7	-393,1	-1969,1	-140,9	-2547,9	37,0	-359,4
121	-433,7	-448,6	-76,7	-560,1	1376,9	-93,9	-1229,9	-542,0	120,6
122	-25,7	-566,6	88,3	-9,1	-756,1	-109,9	104,1	63,0	-677,4

123	-128,7	-1203,6	-518,7	-169,1	-463,1	-77,9	-2749,9	330,0	-828,4
124	-48,7	-128,6	-479,7	32,9	-2569,1	39,1	-1896,9	-1107,0	-82,4
125	-171,7	54,4	-269,7	-378,1	-417,1	-192,9	-603,9	79,0	124,6
126	87,3	-1282,6	508,3	5,9	242,9	275,1	1037,1	-195,0	-321,4
127	116,3	-2010,6	74,3	223,9	-260,1	421,1	-581,9	-436,0	-513,4
128	29,3	-583,6	-367,7	-334,1	-779,1	175,1	-2325,9	-355,0	-1048,4
129	3,3	-119,6	-50,7	-69,1	-872,1	-67,9	-834,9	-242,0	-907,4
130	17,3	-455,6	901,3	-332,1	1953,9	409,1	-67,9	-314,0	-203,4
131	-22,7	3305,4	1335,3	317,9	2909,9	813,1	465,1	-56,0	781,6
132	84,3	-1586,6	-179,7	-196,1	-1507,1	-88,9	-2962,9	-1049,0	-797,4
133	178,3	8674,4	112,3	-70,1	1700,9	-312,9	4017,1	1857,0	-338,4
134	-111,7	-1252,6	-602,7	-561,1	-3065,1	-819,9	-2425,9	854,0	1136,6
135	-114,7	1413,4	1281,3	537,9	415,9	456,1	7922,1	-21,0	1938,6
136	97,3	-1714,6	-310,7	-296,1	-209,1	-289,9	-955,9	-1006,0	-1142,4
137	61,3	-2063,6	-500,7	-37,1	-1549,1	-557,9	-1229,9	-771,0	886,6
138	18,3	-1334,6	1057,3	38,9	1593,9	-124,9	3197,1	60,0	-443,4
139	-104,7	1082,4	1251,3	-131,1	-1440,1	27,1	-895,9	-340,0	-360,4
140	-20,7	-1552,6	-506,7	59,9	589,9	-160,9	-2827,9	87,0	-1703,4
141	-126,7	-1101,6	606,3	129,9	706,9	315,1	58,1	-419,0	-339,4
142	-81,7	-1720,6	-4,7	249,9	1400,9	233,1	-988,9	1038,0	613,6
143	25,3	-2235,6	-445,7	5,9	-2430,1	-168,9	-2583,9	-669,0	860,6
144	-72,7	-899,6	-485,7	-103,1	-1118,1	-282,9	-2058,9	-231,0	-1154,4
145	62,3	463,4	573,3	472,9	771,9	-240,9	4269,1	-845,0	-636,4
146	124,3	-2227,6	-33,7	-537,1	-1678,1	110,1	-1225,9	309,0	-622,4
147	8,3	-1822,6	-454,7	-608,1	132,9	-123,9	-2484,9	-943,0	-1344,4
148	-59,7	980,4	-127,7	-124,1	434,9	53,1	26,1	-210,0	-894,4
149	41,3	-3187,6	-685,7	-333,1	-1875,1	56,1	-3070,9	-1045,0	-201,4
150	-11,7	-427,6	-142,7	158,9	1208,9	81,1	-1079,9	317,0	-731,4
151	-55,7	-1116,6	-576,7	14,9	-766,1	40,1	-3075,9	-727,0	53,6
152	10,3	-220,6	525,3	180,9	795,9	213,1	-1983,9	-801,0	640,6
153	10,3	-1270,6	753,3	506,9	1658,9	134,1	2402,1	197,0	-660,4
154	-42,7	169,4	-118,7	-50,1	262,9	103,1	693,1	378,0	-349,4
155	-173,7	-1883,6	-787,7	-228,1	-1348,1	352,1	-2230,9	-915,0	805,6
156	131,3	-1654,6	-225,7	-339,1	62,9	-53,9	158,1	668,0	11,6
157	-26,7	-1730,6	-589,7	-548,1	-2393,1	89,1	-2865,9	-846,0	-2096,4
158	-186,7	-36,6	-499,7	-285,1	1885,9	212,1	-1684,9	107,0	122,6
159	-44,7	-568,6	-530,7	-496,1	745,9	130,1	-915,9	-32,0	-1192,4
160	-109,7	225,4	-637,7	68,9	1057,9	-509,9	-1378,9	-112,0	4176,6
161	-69,7	-1425,6	-871,7	131,9	-661,1	-478,9	-1694,9	9,0	346,6
162	-24,7	4935,4	51,3	-139,1	-788,1	-207,9	595,1	589,0	229,6
163	-231,7	-1666,6	-862,7	411,9	-798,1	183,1	-2841,9	667,0	68,6

164	20,3	284,4	-74,7	-548,1	-302,1	282,1	-2135,9	-699,0	-405,4
165	-58,7	-439,6	185,3	-67,1	1346,9	76,1	1055,1	-237,0	-778,4
166	-114,7	-2194,6	-720,7	88,9	-888,1	12,1	-3000,9	-1144,0	-261,4
167	13,3	-1384,6	-597,7	495,9	-1221,1	-78,9	-346,9	-237,0	-184,4
168	17,3	1559,4	-773,7	951,9	1689,9	-573,9	-1564,9	2150,0	509,6
169	-25,7	-849,6	-564,7	56,9	-1279,1	-114,9	-0,9	-94,0	-952,4
170	9,3	2067,4	16,3	-52,1	-635,1	-335,9	-1039,9	33,0	1001,6
171	-59,7	-2496,6	-708,7	194,9	-1844,1	-439,9	-1870,9	-89,0	579,6
172	-48,7	-1470,6	-426,7	-197,1	-1269,1	-120,9	-2414,9	7,0	-1081,4
173	96,3	-6,6	674,3	93,9	-130,1	46,1	445,1	-443,0	719,6
174	-16,7	-1360,6	-444,7	-106,1	-126,1	-185,9	-1798,9	-418,0	-1212,4
175	-140,7	-90,6	-410,7	-172,1	1407,9	63,1	1372,1	-42,0	-617,4
176	-41,7	-1076,6	675,3	366,9	-24,1	-233,9	-970,9	-384,0	-252,4
177	-38,7	-1738,6	-676,7	14,9	-1530,1	1,1	-2816,9	-1012,0	1372,6
178	40,3	-37,6	-80,7	-352,1	-581,1	-232,9	-585,9	-244,0	-2224,4
179	346,3	5521,4	3967,3	998,9	134,9	380,1	20416,1	2102,0	-209,4
180	77,3	-1534,6	925,3	-24,1	1209,9	213,1	1765,1	96,0	-123,4
181	-12,7	1293,4	-102,7	-142,1	-390,1	-335,9	-93,9	-717,0	1125,6
182	65,3	-1356,6	-285,7	646,9	-52,1	-269,9	251,1	-144,0	397,6
183	103,3	-1412,6	-453,7	-318,1	698,9	-11,9	-2071,9	-407,0	-599,4
184	-91,7	-2111,6	-592,7	-178,1	-1066,1	-163,9	-1715,9	-642,0	133,6
185	-160,7	1550,4	982,3	-182,1	1873,9	232,1	1972,1	98,0	-77,4
186	-134,7	1513,4	-448,7	-328,1	384,9	-86,9	-2675,9	-1009,0	1575,6
187	19,3	-1465,6	-882,7	110,9	-1334,1	56,1	-2596,9	-713,0	-1139,4
188	-300,7	-3001,6	-813,7	-125,1	-2093,1	-170,9	-1919,9	-787,0	213,6
189	72,3	661,4	-929,7	-233,1	-698,1	-17,9	-1294,9	-430,0	-1833,4
190	43,3	374,4	-668,7	77,9	-1650,1	-0,9	-2713,9	1954,0	1563,6
191	-24,7	-1331,6	857,3	526,9	2036,9	-10,9	1335,1	148,0	-365,4
192	95,3	1920,4	1128,3	1497,9	991,9	64,1	1686,1	2454,0	314,6
193	20,3	3788,4	1403,3	-270,1	-12,1	724,1	4907,1	-250,0	798,6
194	54,3	3305,4	-301,7	-315,1	425,9	313,1	-2420,9	-164,0	781,6
195	64,3	325,4	-1088,7	204,9	1099,9	-398,9	-2712,9	-344,0	4454,6
196	-78,7	1829,4	1867,3	-135,1	734,9	229,1	5615,1	1056,0	-543,4
197	-245,7	1212,4	2748,3	-55,1	1395,9	198,1	8563,1	312,0	317,6
198	-96,7	-1642,6	283,3	549,9	-865,1	-35,9	-1386,9	318,0	-960,4
199	100,3	283,4	-838,7	192,9	-2210,1	-198,9	-2519,9	-1013,0	-418,4
200	76,3	3313,4	1296,3	-142,1	-1520,1	-27,9	4412,1	1698,0	628,6
201	-83,7	-2156,6	-854,7	-4,1	-604,1	-361,9	-1622,9	-305,0	11,6
202	-55,7	-1592,6	-353,7	-146,1	-701,1	-426,9	-1776,9	-734,0	-633,4
203	-8,7	-2546,6	-236,7	-178,1	114,9	150,1	-652,9	-418,0	-1545,4
204	41,3	-1955,6	-486,7	575,9	-857,1	-184,9	-2621,9	820,0	-1817,4

205	-0,7	2410,4	-332,7	-509,1	-890,1	-86,9	-1028,9	-323,0	1436,6
206	61,3	-546,6	664,3	22,9	819,9	-51,9	1191,1	3,0	412,6
207	117,3	2791,4	2733,3	-395,1	-2091,1	419,1	5489,1	1859,0	1845,6
208	104,3	-259,6	-666,7	50,9	-991,1	-375,9	-2384,9	-396,0	-588,4
209	-22,7	2162,4	59,3	-58,1	689,9	313,1	-1790,9	-629,0	944,6
210	44,3	3113,4	882,3	-68,1	1727,9	680,1	4701,1	-242,0	-47,4
211	44,3	1871,4	-629,7	-328,1	294,9	429,1	-1986,9	435,0	1146,6
212	-40,7	168,4	1400,3	642,9	368,9	180,1	4827,1	1654,0	927,6
213	99,3	5011,4	6664,3	1536,9	4414,9	169,1	53594,1	1733,0	-187,4
214	62,3	5873,4	2920,3	499,9	-696,1	547,1	11073,1	-28,0	164,6
215	15,3	-660,6	-678,7	-186,1	1350,9	-276,9	-2184,9	27,0	-1062,4
216	93,3	221,4	746,3	35,9	-995,1	199,1	3936,1	1118,0	340,6



246	-42,7	596,4	-254,7	93,9	-652,1	-82,9	-1979,9	170,0	-1110,4
247	108,3	-116,6	2290,3	19,9	2333,9	640,1	2579,1	-240,0	879,6
248	75,3	-732,6	-31,7	-439,1	-1090,1	213,1	-2042,9	-297,0	61,6
249	125,3	237,4	-911,7	-69,1	1516,9	-343,9	-2305,9	578,0	-1066,4
250	-3,7	4102,4	-570,7	154,9	2556,9	-285,9	-940,9	540,0	151,6
251	125,3	114,4	-885,7	-182,1	503,9	-1086,9	-2621,9	-642,0	-199,4
252	46,3	-3,6	1262,3	114,9	2469,9	125,1	2546,1	97,0	344,6
253	76,3	1407,4	1015,3	513,9	-1069,1	-218,9	2176,1	72,0	-602,4
254	113,3	-870,6	-149,7	-177,1	-338,1	-91,9	-1887,9	190,0	-238,4
255	-230,7	846,4	1780,3	-524,1	188,9	-680,9	-2381,9	-343,0	573,6
256	-2,7	262,4	783,3	-67,1	954,9	767,1	3805,1	813,0	778,6
257	-72,7	-762,6	-216,7	194,9	-1223,1	-134,9	-2124,9	-913,0	-933,4
258	37,3	1508,4	-158,7	401,9	886,9	-21,9	1332,1	460,0	-216,4
259	-23,7	-978,6	-163,7	106,9	-1024,1	-42,9	-1442,9	213,0	-1816,4
260	-343,7	-1111,6	-582,7	-618,1	354,9	-312,9	-1279,9	-274,0	-37,4
261	-63,7	-2757,6	-962,7	7,9	-1521,1	112,1	-2271,9	-581,0	465,6
262	-1,7	-741,6	1664,3	344,9	2908,9	715,1	5745,1	397,0	274,6
263	177,3	31,4	-436,7	52,9	521,9	463,1	-2459,9	27,0	-1690,4
264	105,3	3275,4	46,3	-471,1	-751,1	-85,9	125,1	388,0	783,6
265	304,3	5491,4	-833,7	145,9	-50,1	-375,9	-2146,9	1333,0	130,6
266	2,3	1119,4	445,3	7,9	2017,9	-474,9	2377,1	2054,0	-583,4
267	-50,7	-2025,6	-949,7	70,9	-1272,1	-107,9	-1849,9	-710,0	2194,6
268	-140,7	-1448,6	151,3	235,9	1176,9	123,1	1144,1	-337,0	1347,6
269	364,3	6118,4	1230,3	137,9	1278,9	-20,9	5667,1	1501,0	-36,4
270	371,3	8811,4	2540,3	657,9	4088,9	556,1	11075,1	2754,0	537,6
271	311,3	7701,4	931,3	103,9	1013,9	-105,9	3295,1	118,0	1744,6
272	316,3	7200,4	-653,7	64,9	1451,9	-95,9	-466,9	1454,0	295,6
273	304,3	5956,4	-150,7	2,9	799,9	-203,9	597,1	-143,0	-190,4
274	193,3	4584,4	-133,7	-49,1	-897,1	-92,9	306,1	409,0	177,6
275	-147,7	1213,4	-384,7	-22,1	-468,1	-188,9	-2333,9	689,0	2189,6
276	3,3	-1450,6	-101,7	652,9	1747,9	-358,9	-888,9	391,0	65,6
277	36,3	-1649,6	33,3	-589,1	-527,1	415,1	-1318,9	-460,0	-618,4
278	269,3	1836,4	1529,3	208,9	2423,9	-104,9	6426,1	2954,0	375,6
279	31,3	-1649,6	-485,7	-577,1	-2193,1	207,1	-3098,9	-746,0	-1470,4
280	-96,7	-225,6	-592,7	-511,1	-752,1	-257,9	-2882,9	-530,0	-760,4
281	-14,7	-2624,6	-791,7	73,9	-2288,1	-162,9	-3082,9	-909,0	687,6
282	-30,7	-1812,6	259,3	235,9	190,9	43,1	-324,9	-457,0	1059,6
283	-153,7	-1818,6	-339,7	-202,1	105,9	-141,9	-1757,9	-487,0	-877,4
284	-262,7	-363,6	-144,7	-405,1	2060,9	-163,9	-2685,9	-522,0	-321,4
285	6,3	-2408,6	-355,7	76,9	1423,9	59,1	-478,9	-27,0	-469,4
286	35,3	-1419,6	-688,7	-92,1	1323,9	-40,9	-162,9	-329,0	-803,4

287	-14,7	-464,6	691,3	263,9	1961,9	263,1	-1167,9	-310,0	-141,4
288	-85,7	-307,6	-475,7	250,9	1948,9	710,1	315,1	-332,0	-236,4
289	5,3	-2003,6	-608,7	-69,1	-382,1	-105,9	-1516,9	-109,0	406,6
290	109,3	15293,4	1424,3	-126,1	-1100,1	214,1	4714,1	-117,0	632,6
291	36,3	58,4	-573,7	-421,1	-1470,1	354,1	-1879,9	-646,0	-848,4
292	3,3	-1768,6	-680,7	-543,1	-2678,1	-142,9	-3003,9	-386,0	-1781,4
293	86,3	127,4	-843,7	433,9	216,9	-659,9	-1571,9	-216,0	146,6
294	9,3	-676,6	-145,7	-272,1	2740,9	329,1	1929,1	1005,0	-1051,4
295	269,3	-576,6	-646,7	200,9	519,9	-268,9	1146,1	2154,0	-638,4
296	-134,7	-317,6	-815,7	199,9	1319,9	-24,9	-969,9	90,0	495,6
297	-98,7	-904,6	3,3	531,9	1377,9	229,1	1889,1	1097,0	1730,6
298	18,3	-2819,6	-732,7	-331,1	-660,1	197,1	-1924,9	-445,0	-1172,4
299	-71,7	-2629,6	-842,7	-139,1	-1673,1	84,1	-3087,9	-1177,0	-753,4
300	-20,7	-579,6	552,3	36,9	1112,9	37,1	1238,1	106,0	-991,4
301	-37,7	-1236,6	-37,7	37,9	1137,9	-19,9	-1518,9	-705,0	-61,4
302	97,3	2269,4	186,3	219,9	575,9	496,1	1878,1	-200,0	246,6
303	50,3	201,4	73,3	438,9	186,9	-129,9	1738,1	1285,0	621,6
304	-8,7	-848,6	395,3	118,9	-452,1	-186,9	1097,1	178,0	1589,6
305	-68,7	-1882,6	-511,7	52,9	512,9	-424,9	-1718,9	-756,0	-625,4
306	-38,7	-1048,6	-513,7	-6,1	-750,1	468,1	-2746,9	-1215,0	1801,6
307	9,3	-1602,6	-794,7	-561,1	381,9	155,1	-2292,9	-96,0	-299,4
308	282,3	2156,4	-106,7	2,9	-57,1	-316,9	-188,9	-287,0	293,6
309	229,3	668,4	-668,7	-209,1	-393,1	-482,9	-1593,9	-382,0	-954,4
310	-202,7	-1203,6	-925,7	130,9	-1803,1	-118,9	-3063,9	-436,0	2073,6
311	76,3	-1051,6	-378,7	173,9	-77,1	-67,9	-1053,9	-372,0	-502,4
312	4,3	-568,6	-975,7	170,9	-1116,1	-686,9	-2639,9	954,0	37,6
313	-126,7	-2240,6	-647,7	204,9	-1192,1	52,1	-2009,9	-598,0	733,6
314	92,3	5377,4	3175,3	355,9	4025,9	820,1	18550,1	-268,0	546,6
315	30,3	1677,4	32,3	-172,1	-1776,1	180,1	-2832,9	-900,0	130,6
316	-191,7	-465,6	-260,7	-261,1	-859,1	74,1	-150,9	54,0	-1118,4
317	-230,7	-704,6	-367,7	-519,1	-714,1	-65,9	-2389,9	-192,0	-1225,4
318	-29,7	2165,4	-810,7	821,9	990,9	409,1	-262,9	1926,0	2466,6
319	3,3	-1770,6	-394,7	-653,1	-1760,1	187,1	-2728,9	-575,0	-785,4
320	-44,7	-1285,6	-379,7	202,9	-277,1	166,1	-163,9	-338,0	510,6
321	-82,7	-1942,6	-636,7	217,9	-1417,1	-67,9	-2551,9	-720,0	1279,6
322	19,3	-1062,6	-325,7	-497,1	-1113,1	91,1	-2954,9	-1120,0	-2237,4
323	58,3	-419,6	259,3	153,9	321,9	297,1	1394,1	77,0	648,6
324	25,3	-1488,6	-86,7	461,9	-1306,1	61,1	-2073,9	822,0	-135,4
325	23,3	368,4	619,3	-281,1	-567,1	484,1	-1366,9	-936,0	-485,4
326	-3,7	-1906,6	-868,7	144,9	-479,1	-323,9	-2154,9	294,0	-539,4
327	1,3	24,4	-472,7	-521,1	-1943,1	88,1	-2128,9	-1004,0	-579,4
328	31,3	-1325,6	-88,7	-23,1	-836,1	105,1	-353,9	-519,0	-1631,4
329	69,3	-471,6	-973,7	217,9	-1442,1	-427,9	-3028,9	-928,0	-831,4

Gambar 4.2: LAMPIRAN 8: PC1

No	PC1	No	PC1	No	PC1	No	PC1
1	-2760,01	39	-1878,84	79	-3959,13	199	-2829,09
2	2388,438	40	-1740,56	80	-2125,65	200	5133,195
3	-3417,98	41	-1313,16	81	3534,409	201	-2364,47
4	1721,852	42	-571,486	82	-2403,36	202	-2332,79
5	1813,478	43	18783,49	83	-3124,39	203	-1387,49
6	-3730,77	44	2018,437	84	6537,09	204	-3158,29
7	-1142,06	45	-3016,53	85	-1092,85	205	-504,182
8	-2011,14	46	-1942,82	86	7429,725	206	1212,839
9	-3497,48	47	-2922,12	87	-4033,06	207	6212,462
10	-2285,8	48	6368,596	88	-2962,25	208	-2606,51
11	4605,982	49	-2059,59	89	-1170,1	209	-985,552
12	437,9054	50	-1201,46	90	-2375,38	210	5626,76
13	-3688,48	51	-3883,67	91	-1861,6	211	-1353,03
14	-3889,11	52	-1412,02	92	80,29332	212	5004,643
15	4964,51	53	5406,704	93	-3303,05	213	53249,34
16	-3798,03	54	-3862,1	94	-2583,97	214	12335,55
17	-2159,09	55	-271,584	95	-3505,76	215	-2167,86
18	-1976,62	56	-2482,83	96	-1177,38	216	3801,326
19	-603,313	57	-2387,74	97	-109,471	217	7202,142
20	5083,231	58	-2199,19	98	-1823,53	218	4683,653
21	-1661,96	59	-416,411	99	-2064,44	219	-3164,9
22	-2721,63	60	-287,628	100	-799,279	220	-3615,37
23	1135,24	61	-636,185	101	-2807,27	221	1447,719
24	2030,956	62	989,5226	102	-1853,57	222	-2519,07
25	-2074,84	63	-973,152	103	-3189,02	223	771,7934
26	7407,789	64	-1512,87	104	-1271,85	224	3998,375
27	-1220,8	65	22311,19	105	-3246,8	225	-159,359
28	659,2645	66	-1543,45	106	-2868,67	226	-3042,71
29	-3186,41	67	6468,964	107	-1143,8	227	858,1858
30	-2113,49	68	-3888,04	108	-277,589	228	-3117
31	-2222,78	69	10004,45	109	-2197,97	229	-3445,61
32	-1900,07	70	-1373,23	110	-2784,98	230	-3883,04
33	-3651,79	71	-388,209	111	-3813,62	231	-2151,7
34	7607,003	72	-564,076	112	-3362,38	232	-1869,29
35	-2009,78	73	-2195,76	113	-611,647	233	-1030,49
36	-3179,49	74	3335,945	114	3069,538	234	14133,96
37	-1468,69	75	-1624,63	115	638,9997	235	1704,779
38	518,7248	76	-3433,26	116	-4152,81	236	-3892,48
39	-1878,84	77	7750,843	117	-102,748	237	7936,291
40	-1740,56	78	5601,707	118	-1425,82	238	-3049,99

No	PCI	No	PCI	No	PCI
239	-605,624	279	-3842,5	319	-3423,44
240	2676,008	280	-3039,86	320	-608,793
241	-2700,01	281	-4107,21	321	-3237,61
242	-1906,2	282	-713,272	322	-3406,83
243	3500,259	283	-2236,86	323	1312,969
244	-917,995	284	-2377,87	324	-2472,9
245	-2446,47	285	-954,843	325	-1220,57
246	-1841	286	-505,389	326	-2742,6
247	3149,011	287	-807,221	327	-2446,61
248	-2292,9	288	411,976	328	-901,867
249	-2014,89	289	-2118,81	329	-3425,21
250	546,9415	290	8604,297		
251	-2574,05	291	-2125,09		
252	2985,087	292	-3883,99		
253	2425,21	293	-1562,23		
254	-2068,13	294	2041,333		
255	-1686,63	295	1008,964		
256	3988,662	296	-913,813		
257	-2490,05	297	1861,55		
258	1787,791	298	-2844,63		
259	-1817,59	299	-4085,55		
260	-1589,03	300	1245,499		
261	-3281,98	301	-1633,95		
262	5941,721	302	2489,81		
263	-2307,6	303	1827,878		
264	926,1823	304	846,8259		
265	-583,667	305	-2199,54		
266	3011,764	306	-3070,96		
267	-2621,89	307	-2676,41		
268	897,0078	308	358,2623		
269	7441,93	309	-1548,94		
270	13985,08	310	-3589,2		
271	5511,855	311	-1378,01		
272	1713,937	312	-2895,92		
273	2233,643	313	-2784,48		
274	1391,577	314	19908,43		
275	-1887,61	315	-2504,63		
276	-936,703	316	-471,551		
277	-1797,96	317	-2648,4		
278	7331,335	318	571,1922		

Gambar 4.3: LAMPIRAN 9: Nilai koefisien *Silhouette K-Means++* dengan PC

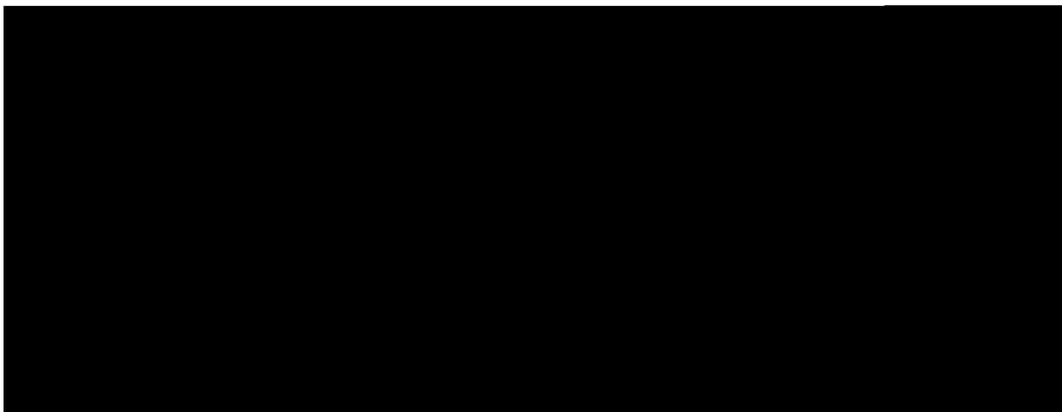
No	s	No	s	No	s	No	s
1	0,963	41	0,970	81	-0,343	121	0,968
2	0,663	42	0,958	82	0,967	122	0,946
3	0,952	43	0,649	83	0,957	123	0,958
4	0,783	44	0,735	84	0,630	124	0,968
5	0,769	45	0,959	85	0,968	125	0,961
6	0,945	46	0,971	86	0,719	126	0,894
7	0,969	47	0,961	87	0,938	127	0,968
8	0,970	48	0,607	88	0,960	128	0,966
9	0,950	49	0,970	89	0,969	129	0,967
10	0,968	50	0,969	90	0,968	130	0,927
11	0,145	51	0,942	91	0,971	131	0,731
12	0,915	52	0,971	92	0,935	132	0,949
13	0,946	53	0,417	93	0,954	133	0,620
14	0,942	54	0,942	94	0,965	134	0,957
15	0,284	55	0,949	95	0,950	135	0,758
16	0,944	56	0,966	96	0,969	136	0,971
17	0,970	57	0,968	97	0,943	137	0,970
18	0,971	58	0,969	98	0,971	138	0,496
19	0,959	59	0,954	99	0,970	139	0,958
20	0,323	60	0,950	100	0,964	140	0,958
21	0,971	61	0,960	101	0,962	141	0,941
22	0,963	62	0,871	102	0,971	142	0,968
23	0,857	63	0,966	103	0,956	143	0,951
24	0,733	64	0,971	104	0,970	144	0,967
25	0,970	65	0,583	105	0,955	145	-0,027
26	0,717	66	0,971	106	0,961	146	0,970
27	0,969	67	0,621	107	0,969	147	0,960
28	0,899	68	0,942	108	0,950	148	0,924
29	0,956	69	0,796	109	0,969	149	0,934
30	0,970	70	0,970	110	0,963	150	0,966
31	0,969	71	0,953	111	0,943	151	0,951
32	0,971	72	0,958	112	0,953	152	0,971
33	0,947	73	0,969	113	0,959	153	0,683
34	0,731	74	-0,414	114	0,485	154	0,894
35	0,970	75	0,971	115	0,901	155	0,960
36	0,956	76	0,951	116	0,936	156	0,950
37	0,971	77	0,740	117	0,943	157	0,945
38	0,909	78	0,465	118	0,971	158	0,970
39	0,971	79	0,940	119	0,876	159	0,967
40	0,971	80	0,970	120	0,955	160	0,968

No	s	No	s	No	s	No	s	No	s
161	0,969	201	0,968	241	0,964	281	0,937	321	0,955
162	0,770	202	0,968	242	0,971	282	0,962	322	0,952
163	0,954	203	0,971	243	-0,355	283	0,969	323	0,837
164	0,970	204	0,957	244	0,966	284	0,968	324	0,967
165	0,864	205	0,957	245	0,967	285	0,966	325	0,969
166	0,945	206	0,849	246	0,971	286	0,957	326	0,963
167	0,967	207	0,584	247	0,460	287	0,964	327	0,967
168	0,962	208	0,965	248	0,968	288	0,916	328	0,965
169	0,958	209	0,967	249	0,970	289	0,970	329	0,952
170	0,956	210	0,471	250	0,908	290	0,776		
171	0,962	211	0,970	251	0,965	291	0,970		
172	0,960	212	0,297	252	0,511	292	0,942		
173	0,910	213	1,000	253	0,655	293	0,971		
174	0,969	214	0,776	254	0,970	294	0,731		
175	0,831	215	0,969	255	0,971	295	0,870		
176	0,968	216	-0,236	256	-0,154	296	0,966		
177	0,951	217	0,701	257	0,966	297	0,762		
178	0,962	218	0,178	258	0,773	298	0,962		
179	0,599	219	0,956	259	0,971	299	0,937		
180	0,803	220	0,948	260	0,971	300	0,845		
181	0,931	221	0,821	261	0,954	301	0,971		
182	0,946	222	0,966	262	0,537	302	0,640		
183	0,968	223	0,891	263	0,968	303	0,767		
184	0,967	224	-0,149	264	0,877	304	0,884		
185	0,586	225	0,945	265	0,959	305	0,969		
186	0,970	226	0,959	266	0,503	306	0,958		
187	0,955	227	0,883	267	0,965	307	0,964		
188	0,957	228	0,957	268	0,880	308	0,920		
189	0,970	229	0,951	269	0,720	309	0,971		
190	0,965	230	0,942	270	0,747	310	0,948		
191	0,832	231	0,970	271	0,444	311	0,970		
192	0,599	232	0,971	272	0,785	312	0,961		
193	0,519	233	0,967	273	0,695	313	0,963		
194	0,970	234	0,744	274	0,828	314	0,627		
195	0,968	235	0,786	275	0,971	315	0,966		
196	0,585	236	0,942	276	0,966	316	0,956		
197	0,786	237	0,750	277	0,971	317	0,964		
198	0,971	238	0,958	278	0,711	318	0,906		
199	0,962	239	0,959	279	0,943	319	0,952		
200	0,339	240	0,595	280	0,959	320	0,959		

Gambar 4.4: LAMPIRAN 10 : Nilai koefisien *Silhouette K-Means*

No	s	No	s	No	s	No	s
1	0,974	41	0,981	81	0,760	121	0,980
2	0,865	42	0,977	82	0,978	122	0,971
3	0,967	43	0,613	83	0,970	123	0,971
4	0,908	44	0,890	84	-0,245	124	0,978
5	0,903	45	0,972	85	0,980	125	0,977
6	0,962	46	0,980	86	-0,014	126	0,950
7	0,981	47	0,973	87	0,957	127	0,981
8	0,980	48	-0,285	88	0,972	128	0,976

0,981	67	-0,262	107	0,981	147	0,972	27
0,952	68	0,960	108	0,973	148	0,962	28
0,970	69	0,429	109	0,979	149	0,954	29
0,979	70	0,981	110	0,974	150	0,980	30
0,979	71	0,974	111	0,961	151	0,966	31
0,980	72	0,976	112	0,967	152	0,981	32
0,963	73	0,979	113	0,977	153	0,872	33
0,035	74	0,781	114	0,808	154	0,950	34
0,980	75	0,981	115	0,953	155	0,972	35
0,970	76	0,966	116	0,955	156	0,973	36
0,981	77	0,071	117	0,970	157	0,962	37
0,956	78	-0,451	118	0,981	158	0,981	38
0,980	79	0,958	119	0,943	159	0,980	39
0,981	80	0,979	120	0,969	160	0,980	40



No	s	No	s	No	s	No	s	No	s
161	0,880	201	0,875	241	0,669	281	0,829	321	0,852
162	0,156	202	0,881	242	0,738	282	0,785	322	0,829
163	0,869	203	0,805	243	0,361	283	0,873	323	0,468
164	0,849	204	0,837	244	0,669	284	0,793	324	0,861
165	0,514	205	0,563	245	0,879	285	0,772	325	0,803
166	0,867	206	0,488	246	0,825	286	0,758	326	0,879
167	0,824	207	0,542	247	0,189	287	0,733	327	0,835
168	0,544	208	0,874	248	0,878	288	0,596	328	0,799
169	0,784	209	0,631	249	0,782	289	0,874	329	0,856
170	0,613	210	0,643	250	-0,149	290	0,369		
171	0,844	211	0,681	251	0,839	291	0,844		
172	0,873	212	0,485	252	0,161	292	0,815		
173	0,632	213	0,450	253	0,029	293	0,838		
174	0,874	214	-0,705	254	0,885	294	-0,197		
175	0,420	215	0,831	255	0,727	295	0,469		
176	0,837	216	0,304	256	0,395	296	0,781		
177	0,845	217	0,440	257	0,869	297	-0,258		
178	0,749	218	0,588	258	-0,149	298	0,850		
179	0,362	219	0,826	259	0,837	299	0,844		
180	0,408	220	0,799	260	0,863	300	0,475		
181	0,601	221	0,371	261	0,850	301	0,842		
182	0,758	222	0,869	262	0,465	302	0,200		
183	0,871	223	0,541	263	0,821	303	-0,251		
184	0,873	224	0,440	264	-0,222	304	0,553		
185	0,230	225	0,599	265	0,234	305	0,864		
186	0,728	226	0,855	266	0,265	306	0,833		
187	0,868	227	-0,043	267	0,811	307	0,877		
188	0,829	228	0,862	268	0,538	308	0,496		
189	0,778	229	0,873	269	0,647	309	0,817		
190	0,746	230	0,825	270	-0,602	310	0,816		
191	0,420	231	0,792	271	0,537	311	0,866		
192	0,251	232	0,849	272	0,205	312	0,854		
193	0,639	233	0,793	273	0,266	313	0,863		
194	0,546	234	-0,465	274	0,025	314	0,210		
195	0,606	235	-0,261	275	0,714	315	0,750		
196	0,624	236	0,817	276	0,764	316	0,780		
197	0,586	237	0,528	277	0,865	317	0,871		
198	0,851	238	0,877	278	0,615	318	-0,281		
199	0,821	239	0,723	279	0,835	319	0,861		
200	0,564	240	0,202	280	0,868	320	0,808		

Gambar 4.5: LAMPIRAN 11 : Nilai koefisien *Silhouette K-Means++*

No	s	No	s	No	s	No	s
1	0,822	41	0,768	81	0,144	121	0,798
2	-0,127	42	0,683	82	0,796	122	0,762
3	0,866	43	0,114	83	0,872	123	0,880
4	-0,276	44	0,197	84	0,609	124	0,814
5	-0,205	45	0,819	85	0,802	125	0,805
6	0,853	46	0,834	86	0,517	126	0,625
7	0,819	47	0,791	87	0,839	127	0,830
8	0,857	48	0,610	88	0,855	128	0,875
9	0,859	49	0,731	89	0,750	129	0,824
10	0,857	50	0,807	90	0,787	130	0,615
11	0,485	51	0,831	91	0,874	131	0,182
12	-0,192	52	0,734	92	0,725	132	0,863
13	0,823	53	0,474	93	0,865	133	0,532
14	0,840	54	0,838	94	0,885	134	0,785
15	0,558	55	0,442	95	0,785	135	0,587
16	0,851	56	0,882	96	0,785	136	0,844
17	0,872	57	0,767	97	0,540	137	0,838
18	0,856	58	0,852	98	0,877	138	0,000
19	0,734	59	0,585	99	0,821	139	0,686
20	0,420	60	0,762	100	0,726	140	0,847
21	0,546	61	0,772	101	0,868	141	0,734
22	0,849	62	0,516	102	0,866	142	0,767
23	-0,343	63	0,666	103	0,841	143	0,832
24	-0,183	64	0,834	104	0,800	144	0,873
25	0,865	65	0,441	105	0,817	145	0,384
26	0,579	66	0,844	106	0,856	146	0,837
27	0,811	67	0,522	107	0,608	147	0,860
28	0,597	68	0,846	108	0,497	148	0,622
29	0,852	69	0,552	109	0,694	149	0,831
30	0,854	70	0,754	110	0,741	150	0,794
31	0,810	71	0,779	111	0,861	151	0,881
32	0,811	72	0,748	112	0,860	152	0,820
33	0,847	73	0,850	113	0,759	153	-0,213
34	0,615	74	0,169	114	0,101	154	0,595
35	0,773	75	0,843	115	0,517	155	0,861
36	0,865	76	0,863	116	0,816	156	0,763
37	0,847	77	0,588	117	0,740	157	0,813
38	0,639	78	0,547	118	0,823	158	0,771
39	0,853	79	0,845	119	0,507	159	0,810
40	0,814	80	0,873	120	0,860	160	0,564

No	s	No	s	No	s	No	s	No	s
161	0,880	201	0,875	241	0,669	281	0,829	321	0,852
162	0,156	202	0,881	242	0,738	282	0,785	322	0,829
163	0,869	203	0,805	243	0,361	283	0,873	323	0,468
164	0,849	204	0,837	244	0,669	284	0,793	324	0,861
165	0,514	205	0,563	245	0,879	285	0,772	325	0,803
166	0,867	206	0,488	246	0,825	286	0,758	326	0,879
167	0,824	207	0,542	247	0,189	287	0,733	327	0,835
168	0,544	208	0,874	248	0,878	288	0,596	328	0,799
169	0,784	209	0,631	249	0,782	289	0,874	329	0,856
170	0,613	210	0,643	250	-0,149	290	0,369		
171	0,844	211	0,681	251	0,839	291	0,844		
172	0,873	212	0,485	252	0,161	292	0,815		
173	0,632	213	0,450	253	0,029	293	0,838		
174	0,874	214	-0,705	254	0,885	294	-0,197		
175	0,420	215	0,831	255	0,727	295	0,469		
176	0,837	216	0,304	256	0,395	296	0,781		
177	0,845	217	0,440	257	0,869	297	-0,258		
178	0,749	218	0,588	258	-0,149	298	0,850		
179	0,362	219	0,826	259	0,837	299	0,844		
180	0,408	220	0,799	260	0,863	300	0,475		
181	0,601	221	0,371	261	0,850	301	0,842		
182	0,758	222	0,869	262	0,465	302	0,200		
183	0,871	223	0,541	263	0,821	303	-0,251		
184	0,873	224	0,440	264	-0,222	304	0,553		
185	0,230	225	0,599	265	0,234	305	0,864		
186	0,728	226	0,855	266	0,265	306	0,833		
187	0,868	227	-0,043	267	0,811	307	0,877		
188	0,829	228	0,862	268	0,538	308	0,496		
189	0,778	229	0,873	269	0,647	309	0,817		
190	0,746	230	0,825	270	-0,602	310	0,816		
191	0,420	231	0,792	271	0,537	311	0,866		
192	0,251	232	0,849	272	0,205	312	0,854		
193	0,639	233	0,793	273	0,266	313	0,863		
194	0,546	234	-0,465	274	0,025	314	0,210		
195	0,606	235	-0,261	275	0,714	315	0,750		
196	0,624	236	0,817	276	0,764	316	0,780		
197	0,586	237	0,528	277	0,865	317	0,871		
198	0,851	238	0,877	278	0,615	318	-0,281		
199	0,821	239	0,723	279	0,835	319	0,861		
200	0,564	240	0,202	280	0,868	320	0,808		

SURAT PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya yang bertanda tangan di bawah ini, mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta:

Nama : Nurlaela
No. Registrasi : 3125121995
Program Studi : Matematika

Menyatakan bahwa skripsi ini yang saya buat dengan judul "**Optimasi Cluster dengan Algoritma *K-Means++* Berbasis Teknik Reduksi *Principal Component Analysis***" adalah :

1. Dibuat dan diselesaikan oleh saya sendiri.
2. Bukan merupakan duplikat skripsi yang pernah dibuat oleh orang lain atau jiplakan karya tulis orang lain.

Pernyataan ini dibuat dengan sesungguhnya dan saya bersedia menanggung segala akibat yang timbul jika pernyataan saya tidak benar.

Jakarta, 16 Agustus 2017

Yang membuat pernyataan

Nurlaela

DAFTAR RIWAYAT HIDUP



NURLAELA. Lahir di Bogor, 15 Agustus 1994. Anak pertama dari pasangan Bapak Enjang Mumin dan Ibu Sanih. Saat ini bertempat tinggal di Tarikolot Jalan Mujaer RT 002/ RW 07 No 34, Bogor 16912.

No. Ponsel : 08567764867

Email : lalaela22@gmail.com

Riwayat Pendidikan : Penulis mengawali pendidikan di SDN Tarikolot 2 pada tahun 2000 - 2006. Setelah itu, penulis melanjutkan ke SMPN 3 Cibinong. Kemudian kembali melanjutkan ke SMA Negeri 2 Cibinong dan lulus tahun 2012. Di Tahun yang sama penulis melanjutkan ke Universitas Negeri Jakarta (UNJ), prodi Matematika, melalui jalur SNMPTN tertulis. Di pertengahan tahun 2017 penulis telah memperoleh gelar Sarjana Sains untuk Program Studi Matematika, FMIPA, UNJ.

Riwayat Organisasi : Selama di bangku perkuliahan, penulis aktif di berbagai organisasi kemahasiswaan. Dalam dua tahun pertama, penulis mendapat kepercayaan sebagai staff Departemen Profeil BEMJ Matematika.

Riwayat Pekerjaan : Penulis mulai menjadi pengajar matematika sejak tahun 2013.