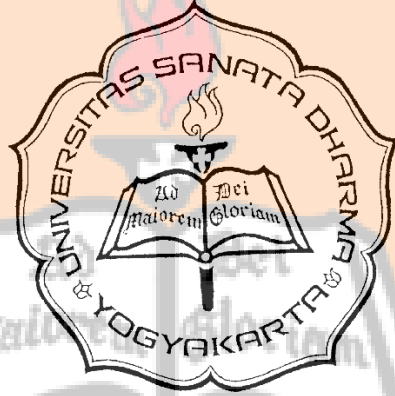


**IMPLEMENTASI ALGORITMA
AGGLOMERATIVE HIERARCHICAL CLUSTERING
UNTUK MENGELOMPOKKAN CAPAIAN BELAJAR SISWA SD
SKRIPSI**

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Program Studi Teknik Informatika



Disusun Oleh :

Aloysius Ari Kurniawan

125314084

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS SANATA DHARMA

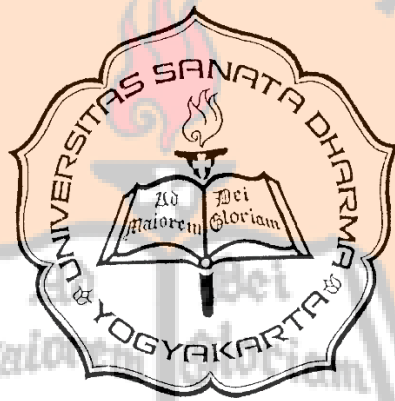
YOGYAKARTA

2017

**THE IMPLEMENTATION OF AN
AGGLOMERATIVE HIERARCHICAL CLUSTERING ALGORITHM
TO GROUP THESE LEARN ELEMENTARY SCHOOL STUDENTS**

THESIS

Presented as Partial Fulfillment of the Requirements
To Obtain *Sarjana Komputer* in Informatics Engineering



By :

Aloysius Ari Kurniawan

125314084

INFORMATICS ENGINEERING STUDY PROGRAM
DEPARTMENT OF INFORMATICS ENGINEERING
FACULTY OF SCIENCE AND TECHNOLOGY
SANATA DHARMA UNIVERSITY

YOGYAKARTA

2017

HALAMAN PERSETUJUAN

SKRIPSI
IMPLEMENTASI ALGORITMA
AGGLOMERATIVE HIERARCHICAL CLUSTERING
UNTUK MENGELOMPOKKAN CAPAIAN BELAJAR SISWA SD

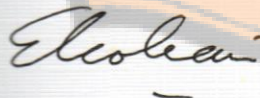
Disusun Oleh :

Nama : Aloysius Ari Kurniawan

NIM : 125314084

Telah disetujui oleh :

Dosen Pembimbing



Eko Hari Parmadi, S.Si, M.Kom

Tanggal, 15 November 2016

HALAMAN PENGESAHAN

IMPLEMENTASI ALGORITMA
AGGLOMERATIVE HIERARCHICAL CLUSTERING
UNTUK MENGELOMPOKKAN CAPAIAN BELAJAR SISWA SD

Dipersiapkan dan disusun oleh :

Nama : Aloysius Ari Kurniawan

NIM : 125314084

Telah dipertahan di depan penguji
pada tanggal 5 Desember 2016
dan dinyatakan memenuhi syarat

Susunan Penguji

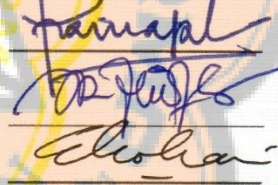
Nama Lengkap

Ketua : Paulina Heruningsih Prima Rosa M.Sc.

Sekretaris : Dr. Anastasia Rita Widiarti, M.Kom.

Anggota : Eko Hari Parmadi, S.Si, M.Kom

Tanda Tangan



Yogyakarta, 4 Januari 2017

Fakultas Sains dan Teknologi

Universitas Sanata Dharma

Dekan,



(Sudi Mungkasi, Ph.D.)

HALAMAN PERSEMBAHAN

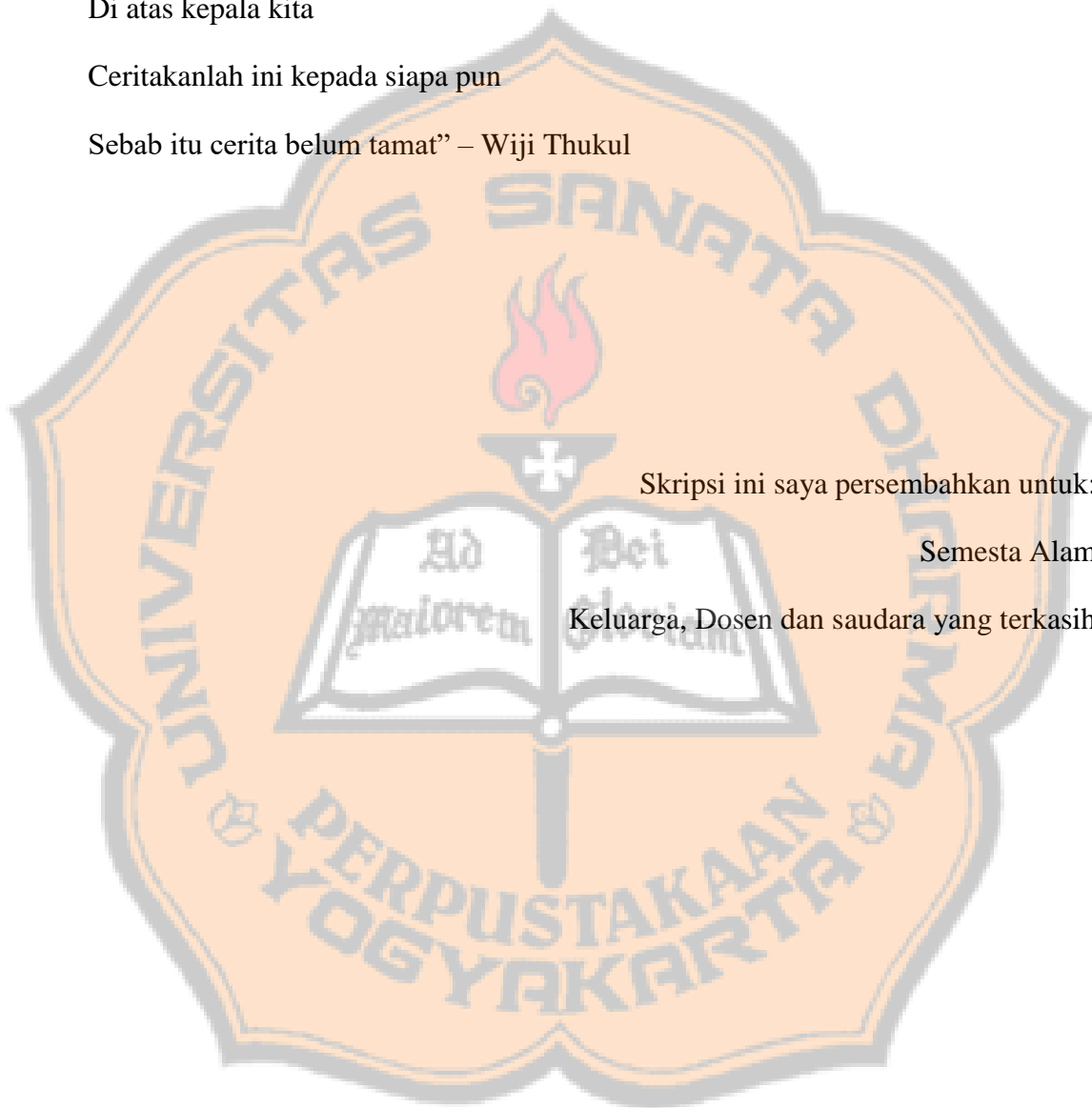
“.....

Di atas kepalaku

Di atas kepala kita

Ceritakanlah ini kepada siapa pun

Sebab itu cerita belum tamat” – Wiji Thukul



Skripsi ini saya persembahkan untuk:

Semesta Alam

Keluarga, Dosen dan saudara yang terkasih

PERNYATAAN KEASLIAN KARYA

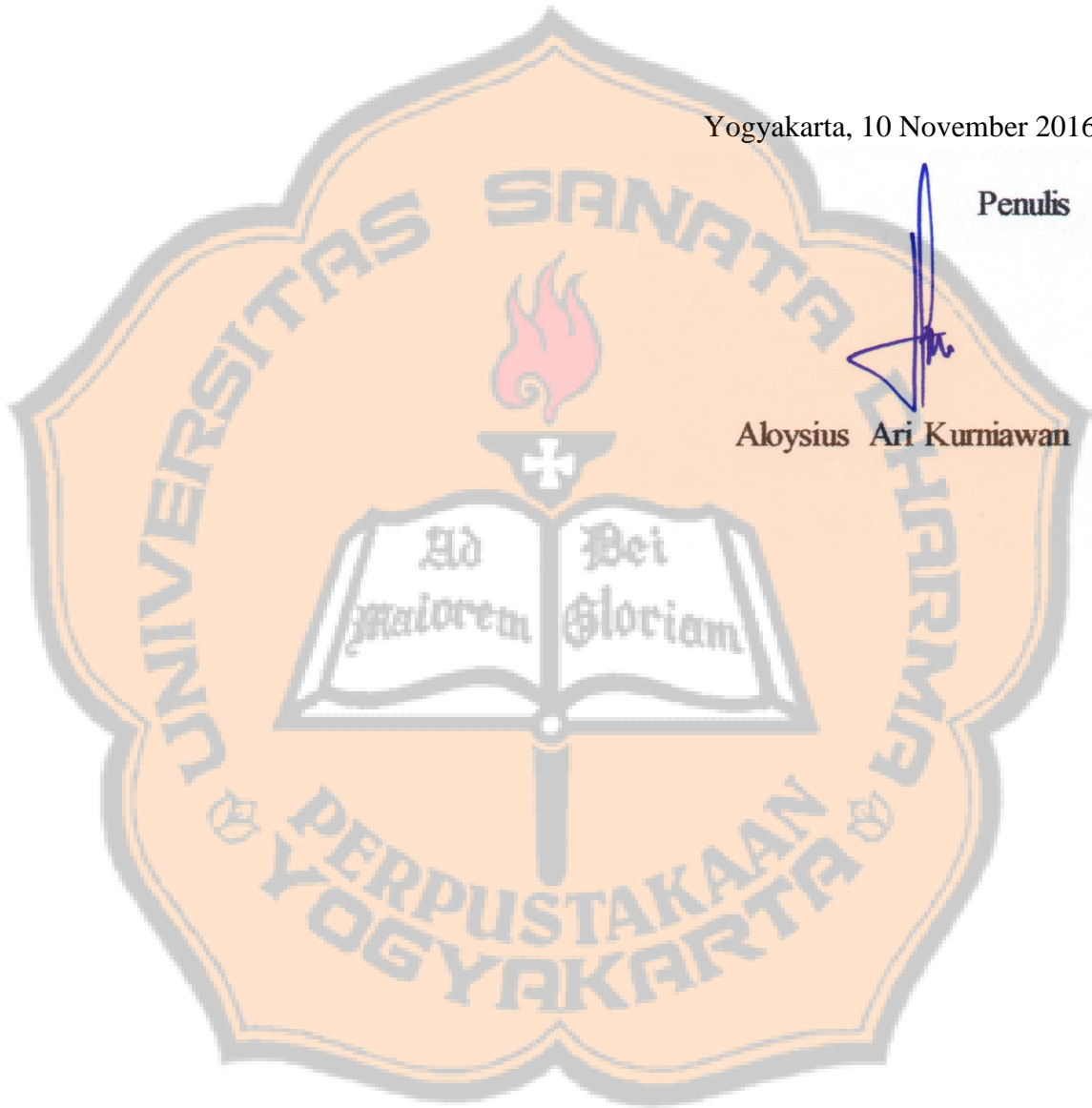
Saya menyatakan dengan sesungguhnya bahwa tugas akhir yang saya tulis tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka sebagaimana layaknya karya ilmiah.

Yogyakarta, 10 November 2016

Penulis



Aloysius Ari Kurniawan



LEMBAR PERNYATAAN PERSETUJUAN
PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS
HALAMAN PERSETUJUAN PUBLIKASI

Yang bertanda tangan dibawah ini, saya mahasiswa Universitas Sanata Dharma :

Nama : Aloysius Ari Kurniawan

NIM : 125314084

Demi pengembangan pengetahuan, saya memberikan kepada perpustakaan

Universitas Sanata Dhama karya ilmiah yang berjudul :

IMPLEMENTASI ALGORITMA
AGGLOMERATIVE HIERARCHICAL CLUSTERING
UNTUK MENGELOMPOKKAN CAPAIAN BELAJAR SISWA SD

Beserta perangkat yang diperlukan (bila ada) dengan demikian saya memberikan kepada perpustakaan Universitas Sanata Dharma hak untuk menyimpan, mengalihkan dalam bentuk media lain, mengelolanya dalam bentuk pangkalan data, mendistribusikan secara terbatas dan mempublikasikan di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya maupun memberikan royalti kepada saya selama tetap mencantumkan nama saya sebagai penulis. Demikian pernyataan ini saya buat dengan sebenarnya.

Yogyakarta, November 2016

Yang menyatakan,



Aloysius Ari Kurniawan

ABSTRAK

Pengelolaan penilaian berbasis kelas merupakan suatu kegiatan pengumpulan informasi tentang proses dan hasil belajar siswa yang dilakukan oleh guru yang bersangkutan sehingga penilaian tersebut akan persamaan kecerdasan dari setiap siswa. Pengelolaan penilaian kelas dilakukan untuk pengambilan keputusan tentang capaian belajar atau hasil belajar peserta didik. Dalam penelitian ini, SD Model melakukan pengelompokan siswa untuk kelas 6. Dalam pengelompokannya siswa dibagi dengan urutan dari kelas C untuk siswa dengan hasil nilai nilai tertinggi sampai kelas A untuk siswa dengan hasil nilai terendah. Pengelompokan tersebut dilakukan untuk mempermudah guru dalam proses belajar mengajar, karena siswa dalam satu kelas memiliki tingkat kecerdasan yang hampir sama.

Dalam pengelompokan ini digunakan algoritma agglomerative hierarchical clustering untuk mengelompokan siswa dengan menggunakan nilai UTS dan UAS sebagai data. Dengan menggunakan metode single linkage, average linkage, dan complete linkage. Masing-masing metode menggunakan euclidean distance untuk melakukan perhitungan jarak. Setiap hasil dari pengelompokan akan dihitung juga nilai sum of square error (SSE).

Dalam percobaan pengelompokan dengan algoritma agglomerative hierarchical clustering digunakan jumlah cluster tiga yang sesuai dengan kebutuhan dari SD Model Kabupaten Sleman. Dari hasil pengelompokan dengan menggunakan metode single linkage, average linkage, dan complete linkage didapatkan hasil pengelompokan yang paling seimbang pada metode average linkage dengan masing-masing jumlah data pada cluster 1, 2, dan 3 sebanyak 14, 22, dan 31. Dalam sistem kelas A merupakan cluster 3, kelas B merupakan cluster 1, dan kelas C merupakan cluster 2. Sesuai dengan kebutuhan dari SD Model Kabupaten Sleman, pemerataan jumlah data pada setiap cluster yang selanjutnya akan digunakan dalam pembagian kelas.

ABSTRACT

Assessment based management class is an activity the collection of information about the process and learning outcomes of students who performed by teachers concerned so that the appraisal will equation intelligence from each student. The management class assessment done to decision-making about achievement of learning or learning outcomes learners .In this research , do primary school model grouping students to class 6. In a clustering of students divided by the order of c class for students with the highest values to class a for students with the results of the lowest value. The clustering was to make teachers in the learning process, because students in a class having levels of intelligence almost the same.

In the clustering of is used algorithms agglomerative hierarchical clustering to divide students with value uts and uas of file. By using the method single linkage, average linkage, and complete linkage.Each methods of using euclidean distance for performing calculations distance. Any result of clustering has been counted worth sum of square error (SSE)

In experiments clustering with algorithms agglomerative hierarchical clustering use the number of cluster three on demand from SD Model Kabupaten Sleman. From the clustering by using the method single linkage, average linkage, and complete linkage grouping get the most balanced on the average linkage every the amount of data on clusters of 1, 2, and 3 about 14, 22, and 31. In the system class A is clusters of 3, class B is clusters of 1, and class C is clusters of 2. In accordance with needs of SD Model Kabupaten Sleman, equity the amount of data on each cluster which will then used dalam pembagian class.

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Tuhan Yesus Kristus karena limpahan kebaikan dan penyertaan-Nya sehingga saya dapat menyelesaikan tugas akhir yang berjudul “Implementasi algoritma *agglomerative hierarchical clustering* untuk mengelompokkan capaian belajar siswa”. Pada proses penulisan tugas akhir ini, saya mengucapkan terima kasih yang sebesar-besarnya kepada

1. Eko Hari Parmadi, S.Si, M.Kom selaku dosen pembimbing, terima kasih atas segala bimbingan dan kesabarannya sehingga saya dapat menyelesaikan tugas akhir ini.
2. Yang terkasih kedua orang tua, Bapak M. Wagiyoto dan Ibu E. Sri Rahayu yang senantiasa mendukung saya dengan doa, kasih sayang dan perhatiannya sehingga saya mampu menyelesaikan studi saya.
3. Keluarga yang lainnya, yang juga telah memberikan dukungannya serta doa sehingga menjadi pemicu semangat dan dukungan moril.
4. Teman-teman dari Teknik Informatika 2012, terimakasih atas segala dukungan, bantuan, dan doa.

Dengan rendah hati penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu segala kritik dan saran untuk perbaikan tugas akhir ini sangat diperlukan. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi semua pihak. Sekian dan terima kasih.

Yogyakarta, 15 November 2016



Aloysius Ari Kurniawan

DAFTAR ISI

IMPLEMENTASI ALGORITMA	i
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERSEMBAHAN	iv
PERNYATAAN KEASLIAN KARYA	vi
LEMBAR PERNYATAAN PERSETUJUAN	vii
ABSTRAK	viii
ABSTRACT.....	ix
KATA PENGANTAR	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Metodologi Penelitian	3
1.6. Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	5
2.1. <i>Data Mining</i>	5
2.2. <i>Clustering</i>	5
2.3. <i>Agglomerative Hierarchical Clustering</i>	6
2.3.1. Single Linkage	6
2.3.2. Average Linkage	6
2.3.3. <i>Complete Linkage</i>	7
2.3.4. Menggunakan metode <i>single linkage</i>	9

2.3.5.	Menggunakan metode <i>average linkage</i>	11
2.3.6.	Menggunakan metode <i>complete linkage</i>	13
2.4.	Uji Akurasi	16
2.5.	Silhouette Index	17
BAB III	METODOLOGI PENELITIAN	19
3.1.	Metode Pengumpulan Data	19
3.1.1.	Wawancara	19
3.1.2.	Pengumpulan Data Dari Arsip Nilai Sekolah	19
3.2.	Analisa Data	19
3.3.	Desain <i>User Interface</i>	23
3.4.	Kebutuhan Software dan Hardware	24
BAB IV	IMPLEMENTASI SISTEM DAN ANALISA HASIL	25
4.1.	Implementasi	25
4.2.	Pengolahan Data	25
4.3.	Clustering	27
4.	28
4.3.1.	<i>Single Linkage</i>	28
4.3.2.	<i>Average Linkage</i>	30
4.3.3.	<i>Complete Linkage</i>	32
4.4.	<i>User Interface</i>	33
4.4.1.	<i>Input Data</i>	34
4.4.2.	Proses AHC dan SSE	35
4.5.	Analisa Hasil	36
4.5.1.	<i>Single Linkage</i>	38
4.5.2.	<i>Average Linkage</i>	38
4.5.3.	<i>Complete Linkage</i>	39
4.5.4.	<i>Silhouette Index</i>	40
4.5.5.	Perbandingan Hasil	49
BAB V	PENUTUP	51
5.1.	Kesimpulan	51
5.2.	Saran	52
DAFTAR PUSTAKA	53

LAMPIRAN 1..... 54
LAMPIRAN 2..... 55
LAMPIRAN 3..... 56
LAMPIRAN 4..... 57



DAFTAR GAMBAR

Gambar 2.1 Contoh Dendrogram.....	7
Gambar 2.2 Hasil Dendrogram <i>single linkage</i>	11
Gambar 2.3 Hasil Dendrogram <i>average linkage</i>	13
Gambar 2.4 Hasil Dendrogram <i>complete linkage</i>	16
Gambar 2.5 Contoh Perpotongan Dendrogram <i>complete linkage</i>	16
Gambar 3.1 Block Diagram proses program.....	19
Gambar 3.2 Dendrogram <i>Single Linkage</i>	21
Gambar 3.3 Dendrogram <i>Average Linkage</i>	21
Gambar 3.4 Dendrogram <i>Complete Linkage</i>	21
Gambar 3.5 <i>User Interface</i>	23
Gambar 3.1 <i>Flowchart</i> Proses Data.....	24
Gambar 4.1 Implementasi – Dendrogram <i>single linkage</i> dengan <i>cluster 3</i>	28
Gambar 4.2 Implementasi – Hasil SSE <i>single linkage</i>	28
Gambar 4.3 Implementasi – Dendrogram <i>average linkage</i> dengan <i>cluster 3</i>	30
Gambar 4.4 Implementasi – Hasil SSE <i>average linkage</i>	30
Gambar 4.5 Implementasi – Dendrogram <i>complete linkage</i> dengan <i>cluster 3</i>	32
Gambar 4.6 Implementasi – Hasil SSE <i>average linkage</i>	32
Gambar 4.7 Tampilan keseluruhan sistem.....	34
Gambar 4.8 Implementasi – Input data.....	35
Gambar 4.9 Implementasi – Hasil proses AHC dan SSE.....	36

DAFTAR TABEL

Tabel 2.1 Sampel data perhitungan jarak	8
Tabel 2.2 Hasil perhitungan <i>euclidean distance</i> dari sampel data	9
Tabel 2.3 Matriks jarak pertama untuk <i>single linkage</i>	9
Tabel 2.4 Matriks jarak kedua untuk <i>single linkage</i>	10
Tabel 2.5 Matriks jarak ketiga untuk <i>single linkage</i>	10
Tabel 2.6 Matriks jarak pertama untuk <i>average linkage</i>	12
Tabel 2.7 Matriks jarak kedua untuk <i>average linkage</i>	12
Tabel 2.8 Matriks jarak ketiga untuk <i>average linkage</i>	13
Tabel 2.9 Matriks jarak pertama untuk <i>complete linkage</i>	14
Tabel 2.10 Matriks jarak kedua untuk <i>complete linkage</i>	15
Tabel 2.11 Matriks jarak ketiga untuk <i>average linkage</i>	15
Tabel 3.1 Sample data nilai siswa	20
Tabel 3.2 Matriks Jarak dari Data Sampel	20
Tabel 3.3 Contoh <i>Cluster Single Linkage</i>	22
Tabel 3.4 Contoh <i>Cluster Average Linkage</i>	22
Tabel 3.5 Contoh <i>Cluster Complete Linkage</i>	22
Tabel 4.1 Data nilai siswa	25
Tabel 4.2 Jumlah data masing-masing <i>cluster</i>	29
Tabel 4.3 Jumlah data masing-masing <i>cluster</i>	31
Tabel 4.4 Jumlah data masing-masing <i>cluster</i>	33
Tabel 4.5 Analisa Keseimbangan Dendrogram	37
Tabel 4.6 Percobaan perhitungan standar deviasi <i>single linkage</i>	38
Tabel 4.7 Percobaan perhitungan standar deviasi <i>average linkage</i>	38
Tabel 4.8 Percobaan perhitungan standar deviasi <i>complete linkage</i>	39
Tabel 4.9 Pengujian <i>silhouette index average linkage cluster 2</i>	41
Tabel 4.10 Pengujian <i>silhouette index average linkage cluster 1</i>	42
Tabel 4.11 Pengujian <i>silhouette index average linkage cluster 3</i>	42
Tabel 4.12 Pengujian <i>silhouette index single linkage cluster 1</i>	44
Tabel 4.13 Pengujian <i>silhouette index single linkage cluster 2</i>	45
Tabel 4.14 Pengujian <i>silhouette index single linkage cluster 3</i>	45
Tabel 4.15 Pengujian <i>silhouette index complete linkage cluster 2</i>	46

Tabel 4.16 Pengujian *silhouette index complete linkage cluster 1* 47
Tabel 4.17 Pengujian *silhouette index complete linkage cluster 3* 48
Tabel 4.18 *Black Box Testing*.....50



BAB I PENDAHULUAN

1.1. Latar Belakang

Pengelolaan penilaian berbasis kelas merupakan suatu kegiatan pengumpulan informasi tentang proses dan hasil belajar siswa yang dilakukan oleh guru yang bersangkutan sehingga penilaian tersebut akan persamaan kecerdasan dari setiap siswa. Pengelolaan penilaian kelas merupakan suatu kegiatan guru yang berkaitan dengan pengambilan keputusan tentang pencapaian kompetensi atau hasil belajar peserta didik yang mengikuti proses pembelajaran. Pengelolaan penilaian kelas merupakan suatu proses yang dilakukan melalui langkah-langkah perencanaan, penyusunan alat penilaian, pengumpulan informasi melalui sejumlah bukti yang menunjukkan pencapaian hasil belajar peserta didik, pengolahan, dan penggunaan informasi tentang hasil belajar peserta didik. Penilaian kelas dilaksanakan melalui berbagai teknik/cara, seperti penilaian unjuk kerja (*performance*), penilaian sikap, penilaian tertulis (*paper and pencil test*), penilaian proyek, penilaian produk, penilaian melalui kumpulan hasil kerja/karya peserta didik (*portfolio*), dan penilaian diri (Suharsini, 1987, 78). Rumusan dalam capaian pembelajaran terdiri dari sikap, pengetahuan, dan keterampilan. Dalam tulisan ini akan membahas aspek pengetahuan, yang berhubungan dengan nilai siswa.

Dalam prosesnya SD Model juga melakukan pengelompokan siswa untuk kelas 6. Kelas 6 C merupakan kelas dengan siswa yang memiliki nilai rata-rata (perhitungan manual per kelas) untuk nilai Bahasa Indonesia, matematika, dan IPA tingkat kecerdasan tinggi, kelas 6 B diisi oleh siswa dengan tingkat kecerdasan cukup, dan kelas 6 A diisi oleh siswa yang memiliki tingkat kecerdasan kurang. Pengelompokan tersebut dilakukan untuk mempermudah guru dalam proses belajar mengajar, karena siswa dalam satu kelas memiliki tingkat kecerdasan yang hampir sama.

Selama ini, pengelompokan dilakukan secara manual. Dengan menghitung rata-rata tiga aspek nilai, yaitu Matematika, Bahasa Indonesia,

dan IPA. Setelah didapat rata-ratanya, nilai tersebut diranking dan dipecah menjadi tiga kelompok sebagai pembagian kelas.

Data mining atau penambangan data memiliki metode *clustering*. Metode *clustering* digunakan untuk mengelompokkan dan menganalisis sejumlah data. Pengelompokan tersebut akan dibagi menjadi beberapa cluster yang memiliki kemiripan antara data satu dengan yang lain. *Agglomerative hierarchical clustering* adalah metode *clustering* dalam penambangan data atau *data mining*.

Dan dalam pengelompokan capaian belajar siswa SD, akan digunakan algoritma *agglomerative hierarchical clustering* untuk mengelompokkan siswa dengan menggunakan nilai UTS dan UAS sebagai data.

1.2. Rumusan Masalah

1. Bagaimana metode *agglomerative hierarchical clustering* mampu dengan baik mengelompokkan capaian belajar siswa, sehingga sekolah bisa melakukan tindakan untuk mengefektifkan proses pendampingan dan pembelajaran siswa.

1.3. Batasan Masalah

1. Data yang diolah dibatasi hanya untuk siswa SD Model Kabupaten Sleman.
2. Pengelompokan dilakukan merupakan hasil dari pengolahan nilai Ujian Tengah Semester (UTS) dan Ujian Akhir Semester (UAS).
3. Pengelompokan menggunakan metode *agglomerative hierarchical clustering single linkage, average linkage, dan complete linkage*.
4. Nilai yang digunakan untuk proses *clustering* merupakan nilai tanpa bobot.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengelompokkan capaian belajar siswa dengan mengolah nilai UTS dan UAS menggunakan algoritma *agglomerative hierarchical clustering*.

1.5. Metodologi Penelitian

Metode yang digunakan dalam mengelompokkan capaian belajar siswa SD dengan menggunakan algoritma *agglomerative hierarchical clustering* adalah sebagai berikut:

1. Studi Pustaka
2. Pada tahap pengumpulan informasi dengan menggunakan data nilai sekolah dari SD Model Kabupaten Sleman.
3. Perancangan Sistem
4. Pada tahap ini dilakukan perancangan sistem yang akan dibuat.
5. Implementasi
6. Tahap untuk membuat aplikasi yang akan mempermudah dalam melakukan mengelompokkan capaian belajar siswa.
7. Pengujian Perangkat Lunak
8. Tahap ini dilakukan untuk mengetahui keakuratan dari algoritma *agglomerative hierarchical clustering* dalam mengelompokkan capaian belajar siswa.

1.6. Sistematika Penulisan

BAB I : PENDAHULUAN

Berisi latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penelitian.

BAB II : LANDASAN TEORI

Bab ini berisi pembahasan landasan teori yang digunakan dalam mengelompokkan capaian belajar dengan *data mining* algoritma *agglomerative hierarchical clustering*.

BAB III : METODOLOGI

Bab ini berisi metode pengumpulan data, teknik analisa data, rancangan *user interface*, dan kebutuhan *software* serta *hardware*.

BAB IV : IMPLEMENTASI SISTEM DAN ANALISA HASIL

Bab ini berisi pembahasan tentang implementasi program dan analisa sistem pengelompokan capaian belajar dengan algoritma *agglomerative hierarchical clustering*.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari keseluruhan proses pembuatan tugas akhir dan saran pengembangan lebih lanjut.



BAB II

LANDASAN TEORI

2.1. *Data Mining*

Data mining adalah proses yang mempekerjakan satu atau lebih teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) secara otomatis. Definisi lain diantaranya adalah pembelajaran berbasis induksi (*induction-based learning*) adalah proses pembentukan definisi konsep-konsep yang dilakukan dengan cara mengobservasi contoh-contoh spesifik dari konsep-konsep yang akan dipelajari. *Knowledge Discovery in Database* (KDD) adalah penerapan metode statistik pada *data mining*. Dalam konteks ini *data mining* merupakan satu langkah dalam KDD (Fajar Astuti H, 2013).

2.2. *Clustering*

Clustering merupakan proses mengelompokkan atau penggolongan objek berdasarkan informasi yang diperoleh dari data yang menjelaskan hubungan antar objek dengan prinsip untuk memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas/*cluster*. *Clustering* bertujuan menemukan kelompok (*cluster*) objek yang berguna, dimana gunanya tergantung dari tujuan analisa data.

Ada dua tipe dari *clustering*, yaitu *partitional clustering* dan *hierarchical clustering*. Dengan *partitional clustering*, objek data dibagi ke dalam sub-himpunan (*cluster*) yang tidak *overlap* sedemikian hingga tiap objek data berada dalam tepat satu sub-himpunan (Fajar Astuti H, 2013). *K-means* merupakan algoritma yang menggunakan *partitional clustering*. Seperti pada *partitional clustering*, yang menggabungkan semua objek atau membuat pengelompokan data. *Hierarchical clustering* merupakan sebuah *cluster* bersarang yang diatur sebagai pohon hirarki.

Tiap simpul (*cluster*) dalam pohon (kecuali simpul daun) merupakan gabungan dari anaknya (*subcluster*) dan simpul akar berisi semua objek (Fajar Astuti H, 2013).

2.3. Agglomerative Hierarchical Clustering

Strategi pengelompokan *hierarchical clustering* umumnya ada dua jenis yaitu *Agglomerative (Bottom-Up)* dan *Devisve (Top-Down)* (Afif S, 2014). Namun untuk penelitian ini penulis menggunakan pendekatan *agglomerative hierarchical clustering*. Penulis memilih algoritma *agglomerative hierarchical clustering* dikarenakan; hasil dari pengelompokan data dapat dilihat dendrogram, tidak diperlukan penentuan jumlah *cluster* pada awal pengelompokan, dan *agglomerative hierarchical clustering* dengan pendekatan bawah-atas (*bottom-up*) dimana pengelompokan data dimulai dari kecil ke pengelompokan yang besar.

Agglomerative hierarchical clustering (AHC) dengan menggunakan *bottom-up*, dimulai dari masing-masing data sebagai sebuah *cluster*, kemudian secara rekursif mencar kelompok terdekat sebagai pasangan yang kemudian akan digabungkan menjadi kelompok yang lebih besar. Proses tersebut diulang terus sehingga tampak bergerak keatas membentuk hirarki (Prasetyo E, 2012).

Terdapat tiga teknik kedekatan dalam *hierarchical clustering*, yaitu; *single linkage* (jarak terdekat) atau tautan tunggal, *average linkage* (jarak rata-rata) atau tautan rata-rata, dan *complete linkage* (jarak terjauh) atau tautan lengkap.

2.3.1. Single Linkage

Single linkage (MIN) menentukan kedekatan diantara dua kelompok tersdekak (terkecil) antar dua data dari *cluster* yang berbeda.

Formulasi untuk *single linkage* adalah :

$$d_{UV} = \min\{d_{UV}\} \dots\dots\dots (2.1)$$

Keterangan :

$\{d_{UV}\}$ adalah jarak antara data U dan V dari masing-masing *cluster* U dan V.

2.3.2. Average Linkage

Average linkage (AVERAGE) menentukan kedekatan diantara dua kelompok dari jarak rata-rata antar dua data dari *cluster* yang berbeda.

Formulasi untuk *average linkage* adalah :

$$d_{UV} = \frac{1}{|U|*|V|} \sum_{U \in D} \sum_{V \in D} d_{UV} \dots\dots\dots (2.2)$$

Keterangan :

|U| dan |V| adalah jumlah data yang ada dalam *cluster* U dan V.

2.3.3. Complete Linkage

Complete linkage (MAX) menentukan kedekatan diantara dua kelompok dari jarak terjauh (terbesar) antara dua data dari *cluster* yang berbeda.

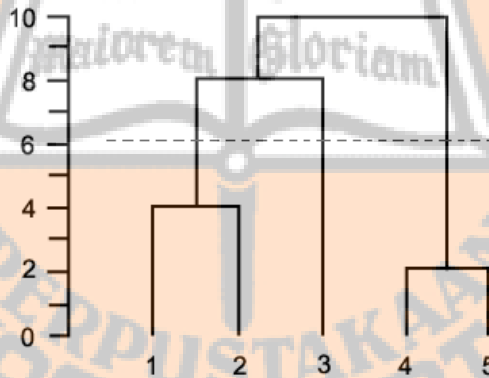
Formulasi untuk *complete linkage* adalah :

$$d_{UV} = \max\{d_{UV}\} \dots\dots\dots (2.3)$$

Keterangan :

{*d_{UV}*} adalah jarak antara data U dan V dari masing-masing *cluster* U dan V.

Dengan menggunakan formulasi *single linkage*, *average linkage*, dan *complete linkage* akan menghasilkan dendrogram. Penulis dalam penelitian ini akan menggunakan *single linkage*.



Gambar 2.1 Contoh Dendrogram

Algoritma *agglomerative hierarchical clustering* dapat dijabarkan sebagai berikut (Prasetyo E, 2012):

1. Hitung matriks kedekatan berdasarkan jenis jarak yang digunakan.
2. Ulangi langkah 3 dan 4, hingga hanya satu *cluster* yang tersisa.
3. Gabungkan dua *cluster* terdekat berdasarkan parameter kedekatan yang ditentukan.

- Perbarui matriks kedekatan untuk merefleksikan kedekatan di antara *cluster* baru dan *cluster* asli yang sudah digabung.

Sebagai contoh penerapan dari *agglomerative hierarchical clustering*, data sebanyak lima dan dengan jumlah dua dimensi. Pengelompokan dilakukan dengan metode *agglomerative hierarchical clustering* menggunakan jarak *euclidean distance* dan metode tautan *single linkage*, *average linkage*, dan *complete linkage*.

Tabel 2.1 Sampel data perhitungan jarak

Data	X	Y
1	1	1
2	4	1
3	1	2
4	3	4
5	5	4

Yang pertama dilakukan adalah menghitung jarak *euclidean distance* setiap pasangan data.

$$d(1,2) = \sqrt{(|4 - 1|^2 + |1 - 1|^2)} = 3$$

$$d(1,3) = \sqrt{(|1 - 1|^2 + |2 - 1|^2)} = 1$$

$$d(1,4) = \sqrt{(|3 - 1|^2 + |4 - 1|^2)} = 3,61$$

$$d(1,5) = \sqrt{(|5 - 1|^2 + |4 - 1|^2)} = 5$$

$$d(2,3) = \sqrt{(|1 - 4|^2 + |2 - 1|^2)} = 3,16$$

$$d(2,4) = \sqrt{(|3 - 4|^2 + |4 - 1|^2)} = 3,16$$

$$d(2,5) = \sqrt{(|5 - 4|^2 + |4 - 1|^2)} = 3,16$$

$$d(3,4) = \sqrt{(|3 - 1|^2 + |4 - 2|^2)} = 2,83$$

$$d(3,5) = \sqrt{(|5 - 1|^2 + |4 - 2|^2)} = 4,47$$

$$d(4,5) = \sqrt{(|5 - 3|^2 + |4 - 4|^2)} = 3$$

Dengan perhitungan tersebut maka akan didapatkan hasil *similarity matriks*:

Tabel 2.2 Hasil perhitungan *euclidean distance* dari sampel data

	1	2	3	4	5
1	0	3	1	3,61	5
2	3	0	3,16	3,16	3,16
3	1	3,16	0	2,83	4,47
4	3,61	3,16	2,83	0	2
5	5	3,16	4,47	2	0

2.3.4. Menggunakan metode *single linkage*

Dengan metode *single linkage*, selanjutnya dipilih dua jarak *cluster* yang paling terkecil.

$$\min(d_{UV}) = d_{13} = 1$$

Maka dipilih *cluster* 1 dan 3, sehingga *cluster* 1 dan 3 digabungkan.

Selanjutnya, hitung kembali jarak-jarak antara *cluster* (13) dengan *cluster* yang tersisa 2,4 dan 5.

$$d_{(13)2} = \min\{d_{12}, d_{32}\} = \min\{3, 3.16\} = 3$$

$$d_{(13)4} = \min\{d_{14}, d_{34}\} = \min\{3.61, 2.83\} = 2.83$$

$$d_{(13)5} = \min\{d_{15}, d_{35}\} = \min\{5, 4.47\} = 4.47$$

Kemudian baris-baris dan kolom-kolom pada matriks yang bersesuaian dengan *cluster* 1 dan 3, kemudian menambahkan baris dan kolom *cluster* (13).

Tabel 2.3 Matriks jarak pertama untuk *single linkage*

	13	2	4	5
13	0	3	2.83	4.47
2		0	3,16	3,16
4			0	2
5				0

Selanjutnya dipilih jarak dua *cluster* terkecil.

$$\min(d_{UV}) = d_{45} = 2$$

Selanjutnya, hitung kembali jarak-jarak antara *cluster* (45) dengan *cluster* yang tersisa (13) dan 2.

$$d_{(45)(13)} = \min\{d_{41}, d_{43}, d_{51}, d_{53}\} = \min\{3.61, 2.83, 5, 4.47\} = 2.83$$

$$d_{(45)2} = \min\{d_{42}, d_{52}\} = \min\{3.16, 3.16\} = 3.16$$

Kemudian baris-baris dan kolom-kolom pada matriks yang bersesuaian dengan *cluster* 4 dan 5, kemudian menambahkan baris dan kolom *cluster* (45).

Tabel 2.4 Matriks jarak kedua untuk *single linkage*

	13	2	45
13	0	3	2.83
2		0	3,16
45			0

Selanjutnya dipilih kembali jarak dua *cluster* terkecil

$$\min(d_{UV}) = d_{(1345)} = 2.83$$

Selanjutnya, hitung kembali jarak-jarak antara *cluster* (1345) dengan *cluster* yang tersisa, yaitu 2.

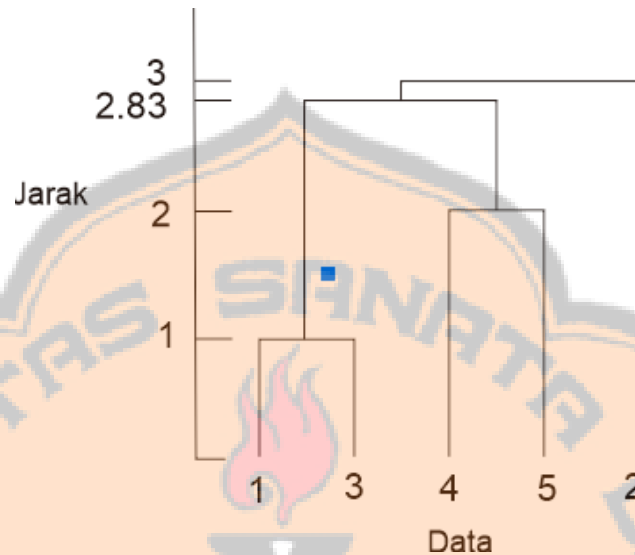
$$d_{(1345)2} = \min\{d_{12}, d_{32}, d_{42}, d_{52}\} = \min\{3, 3.16, 3.16, 3.16\} = 3$$

Kemudian baris-baris dan kolom-kolom pada matriks yang bersesuaian dengan *cluster* (13) dan (45), kemudian menambahkan baris dan kolom *cluster* (1345).

Tabel 2.5 Matriks jarak ketiga untuk *single linkage*

	1345	2
1345	0	3
2		0

Jadi, cluster (1345) dan 2 digabung untuk membentuk *cluster* tunggal dari lima data, yaitu *cluster* (12345) ketika jarak terdekat mencapai 3. Berikut ini adalah dendrogram hasil *agglomerative hierarchical clustering* dengan *single linkage*.



Gambar 2.2 Hasil Dendrogram *single linkage*

2.3.5. Menggunakan metode *average linkage*

Dengan metode *average linkage*, selanjutnya dipilih dua jarak *cluster* yang akan dihitung dengan nilai rata-rata.

$$\min(d_{UV}) = d_{13} = 1$$

Perhitungan awal *cluster* (13) tetap digunakan dikarenakan *cluster* (13) memiliki jarak paling dekat.

Maka dipilih *cluster* 1 dan 3, sehingga *cluster* 1 dan 3 digabungkan. Selanjutnya, akan dihitung jarak-jarak antara *cluster* (13) dengan *cluster* 2,4 dan 5.

$$d_{(13)2} = \text{average}\{d_{12}, d_{32}\} = \frac{3 + 3.16}{2} = 3.08$$

$$d_{(13)4} = \text{average}\{d_{14}, d_{34}\} = \frac{3.61 + 2.83}{2} = 3.22$$

$$d_{(13)5} = \text{average}\{d_{15}, d_{35}\} = \frac{5 + 4.47}{2} = 4.73$$

Kemudian baris-baris dan kolom-kolom pada matriks yang bersesuaian dengan *cluster* 1 dan 3, kemudian menambahkan baris dan kolom *cluster* (13).

Tabel 2.6 Matriks jarak pertama untuk *average linkage*

	13	2	4	5
13	0	3.08	3.22	4.73
2		0	3,16	3,16
4			0	2
5				0

Selanjutnya dipilih kembali jarak dua *cluster* terkecil.

$$\min(d_{UV}) = d_{(45)} = 2$$

Selanjutnya, hitung kembali jarak-jarak antara *cluster* (45) dengan *cluster* yang tersisa (13) dan 2.

$$d_{(45)(13)} = \text{average}\{d_{41}, d_{43}, d_{51}, d_{53}\} = \frac{3.61 + 2.83 + 5 + 4.47}{4} = 3.97$$

$$d_{(45)2} = \text{average}\{d_{42}, d_{52}\} = \frac{3.61 + 2.83 + 5 + 4.47}{4} = 3.$$

Kemudian baris-baris dan kolom-kolom pada matriks yang bersesuaian dengan *cluster* 4 dan 5, kemudian menambahkan baris dan kolom *cluster* (45).

Tabel 2.7 Matriks jarak kedua untuk *average linkage*

	13	2	45
13	0	3.08	3.97
2		0	3,16
45			0

Selanjutnya dipilih kembali jarak dua *cluster* terkecil.

$$\min(d_{UV}) = d_{(132)} = 3.08$$

Selanjutnya, hitung kembali jarak-jarak antara *cluster* (132) dengan *cluster* yang tersisa, yaitu (45).

$$d_{(132)(45)} = average\{d_{14}, d_{15}, d_{34}, d_{35}, d_{24}, d_{25}\}$$

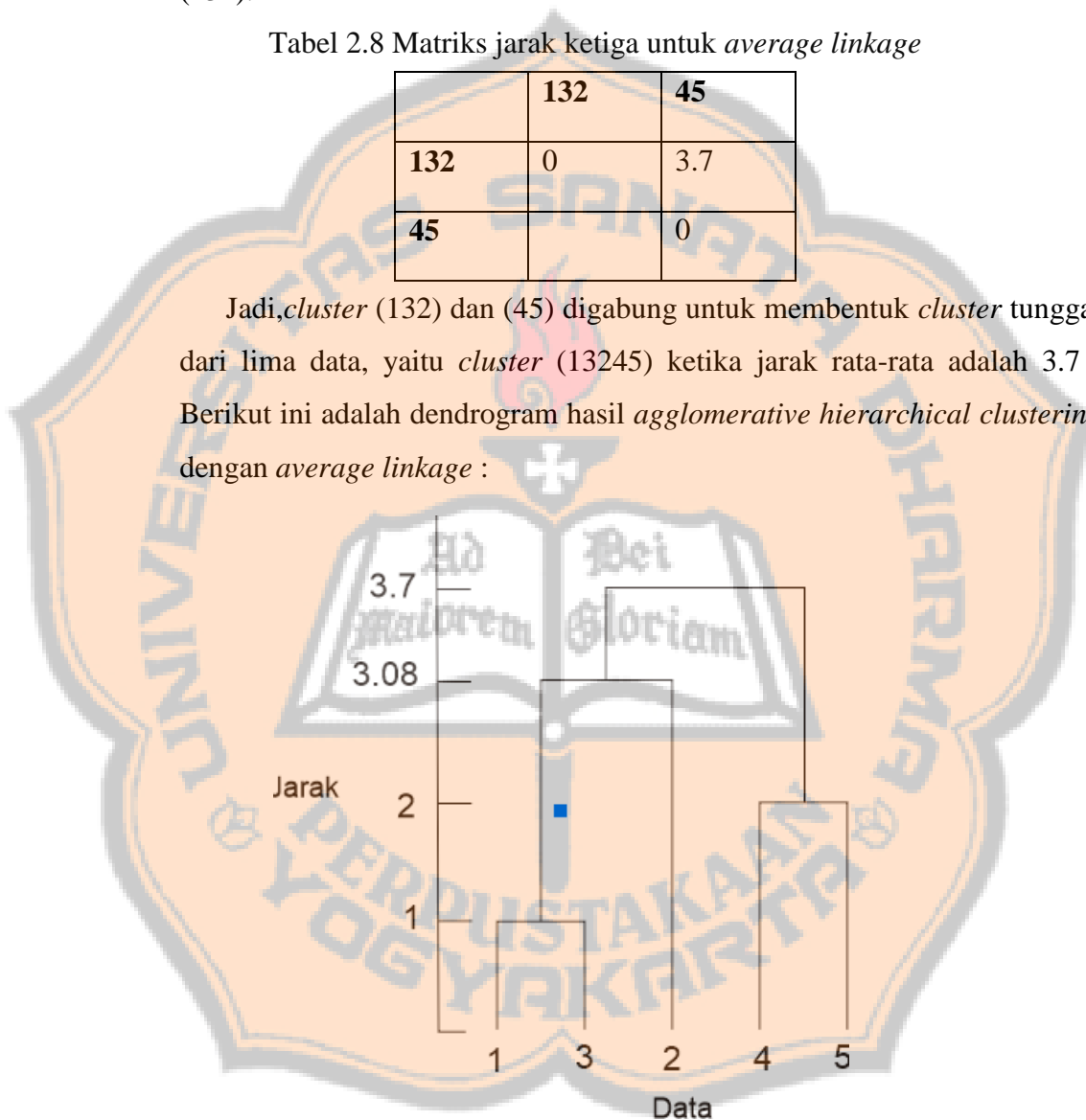
$$= \frac{3.61 + 5 + 2.83 + 4.47 + 3.16 + 3.16}{6} = 3.7$$

Kemudian baris-baris dan kolom-kolom pada matriks yang bersesuaian dengan *cluster* (13) dan 2, kemudian menambahkan baris dan kolom *cluster* (132).

Tabel 2.8 Matriks jarak ketiga untuk *average linkage*

	132	45
132	0	3.7
45		0

Jadi, *cluster* (132) dan (45) digabung untuk membentuk *cluster* tunggal dari lima data, yaitu *cluster* (13245) ketika jarak rata-rata adalah 3.7 . Berikut ini adalah dendrogram hasil *agglomerative hierarchical clustering* dengan *average linkage* :



Gambar 2.3 Hasil Dendrogram *average linkage*

2.3.6. Menggunakan metode *complete linkage*

Dengan metode *complete linkage*, selanjutnya dipilih dua jarak *cluster* yang akan dihitung dengan nilai terjauh.

$$\min(d_{UV}) = d_{13} = 1$$

Perhitungan awal *cluster* (13) tetap digunakan dikarenakan *cluster* (13) memiliki jarak paling dekat.

Maka dipilih *cluster* 1 dan 3, sehingga *cluster* 1 dan 3 digabungkan. Selanjutnya, akan dihitung jarak-jarak antara *cluster* (13) dengan *cluster* 2,4 dan 5.

$$d_{(13)2} = \max\{d_{12}, d_{32}\} = \min\{3, 3.16\} = 3.16$$

$$d_{(13)4} = \max\{d_{14}, d_{34}\} = \min\{3.61, 2.83\} = 3.61$$

$$d_{(13)5} = \max\{d_{15}, d_{35}\} = \min\{5, 4.47\} = 5$$

Kemudian baris-baris dan kolom-kolom pada matriks yang bersesuaian dengan *cluster* 1 dan 3, kemudian menambahkan baris dan kolom *cluster* (13).

Tabel 2.9 Matriks jarak pertama untuk *complete linkage*

	13	2	4	5
13	0	3.16	3.61	5
2		0	3,16	3,16
4			0	2
5				0

Selanjutnya dipilih kembali jarak dua *cluster* terkecil.

$$\min(d_{UV}) = d_{(45)} = 2$$

Selanjutnya, hitung kembali jarak-jarak antara *cluster* (45) dengan *cluster* yang tersisa (13) dan 2.

$$d_{(45)(13)} = \max\{d_{41}, d_{43}, d_{51}, d_{53}\} = \max\{3.61, 2.83, 5, 4.47\} = 5$$

$$d_{(45)2} = \min\{d_{42}, d_{52}\} = \min\{3.16, 3.16\} = 3.16$$

Kemudian baris-baris dan kolom-kolom pada matriks yang bersesuaian dengan *cluster* 4 dan 5, kemudian menambahkan baris dan kolom *cluster* (45).

Tabel 2.10 Matriks jarak kedua untuk *complete linkage*

	13	2	45
13	0	3.16	5
2		0	3,16
45			0

Selanjutnya dipilih kembali jarak dua *cluster* terkecil.

$$\min(d_{UV}) = d_{(452)} = 2$$

Dan

$$\min(d_{UV}) = d_{(132)} = 2$$

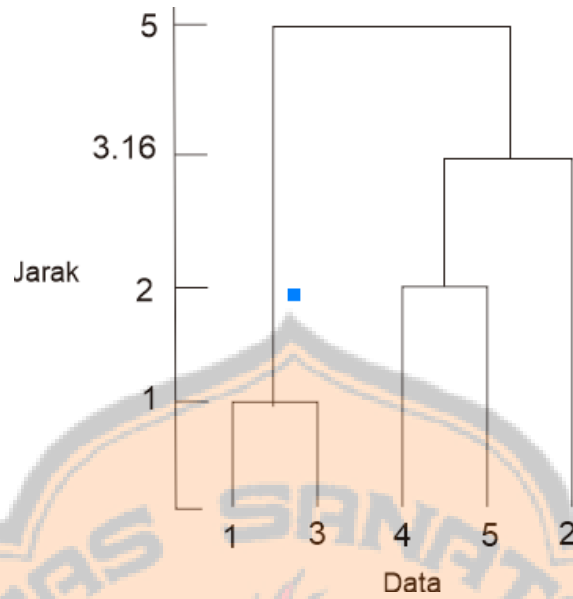
Dikarenakan kedua *cluster* berada pada obyek yang sama, maka dapat dipilih salah satunya. Penulis memilih *cluster* (132), kemudian jarak dihitung dengan *cluster* yang tersisa, yaitu (45).

$$\begin{aligned} d_{(132)(45)} &= \max\{d_{14}, d_{15}, d_{34}, d_{35}, d_{24}, d_{25}\} \\ &= \max\{3.61, 5, 2.83, 4.47, 3.16, 3.16\} = 5 \end{aligned}$$

Kemudian baris-baris dan kolom-kolom pada matriks yang bersesuaian dengan *cluster* (132) dan (45), kemudian menambahkan baris dan kolom *cluster* (13245).

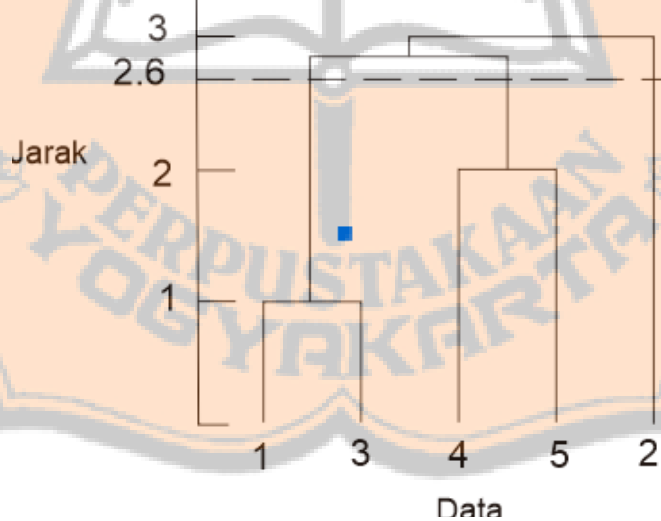
Tabel 2.11 Matriks jarak ketiga untuk *average linkage*

	132	45
132	0	5
45		0



Gambar 2.4 Hasil Dendrogram *complete linkage*

Hasil akhir dari *agglomerative hierarchical clustering* adalah dendrogram dan jumlah *cluster* ditentukan pada potongan (*cut-off*) pada dendrogram (perpotongan pada jarak. Dan penulis memotong dendrogram *single linkage* pada jarak 2.6 maka akan dihasilkan tiga *cluster*, yaitu : (13), (45) dan 2.



Gambar 2.5 Contoh perpotongan dendrogram *complete linkage*

2.4. Uji Akurasi

Hasil *clustering* yang telah didapatkan perlu diuji akurasi agar dapat diketahui validasi dari data tersebut. Karena proses yang digunakan adalah metode *agglomerative hierarchical clustering* maka untuk mengukur

validasi data dapat menggunakan Validasi Internal dan Validasi Eksternal (Prasetyo E, 2012).

1. Validasi Internal

Validasi Internal atau *unsupervised* merupakan uji akurasi data *cluster* tanpa membutuhkan informasi eksternal. Dan contoh paling umum adalah SSE (*sum of square error*).

2. Validasi Eksternal

Validasi Eksternal dilakukan dengan mengukur tingkat kedekatan antara label *cluster* dengan label kelas. Validasi Eksternal dapat dilakukan dengan *entropy, purity, precision, recall, dan F-measure*.

Dalam tulisan ini, penulis menggunakan Validasi Internal yaitu SSE (*sum of squsred error*) untuk melakukan uji akurasi. Setiap percobaan *cut-off* akan diuji akurasi menggunakan SSE. Jadi setiap *cut-off* akan didapatkan nilai SSE. Dan dalam SSE, semakin kecil nilai SSE maka akan semakin baik pula dalam pembentukan *cluster*-nya. Rumus yang digunakan sebagai berikut :

$$SSE = \sum_{k=1}^K \sum_{x_i \in C_k} ||X_i - \mu_k||^2 \dots\dots\dots (2.4)$$

Keterangan :

x_i adalah jarak data x di indeks i

μ_k adalah rata – rata semua jarak data x_i di cluster k

2.5. Silhouette Index

Silhouette index (SI) metode untuk mengukur validasi baik sebuah data, *cluster* tunggal atau keseluruhan *cluster*. Untuk menghitung nilai SI dari data ke- i , terdapat dua komponen yaitu a_i dan b_i . a_i adalah jarak dari data ke- i terhadap semua data dalam *cluster* yang sama. Sedangkan b_i adalah hasil rata-rata jarak data ke- i terhadap semua data dari *cluster* lain (Prasetyo E, 2012).

Berikut meruapak rumus yang digunakan untuk menghitung a_i :

$$a_i^j = \frac{1}{m_j - 1} \sum_{\substack{r=1 \\ r \neq i}}^{m_j} d(x_i^j, x_r^j) \dots\dots\dots (2.5)$$

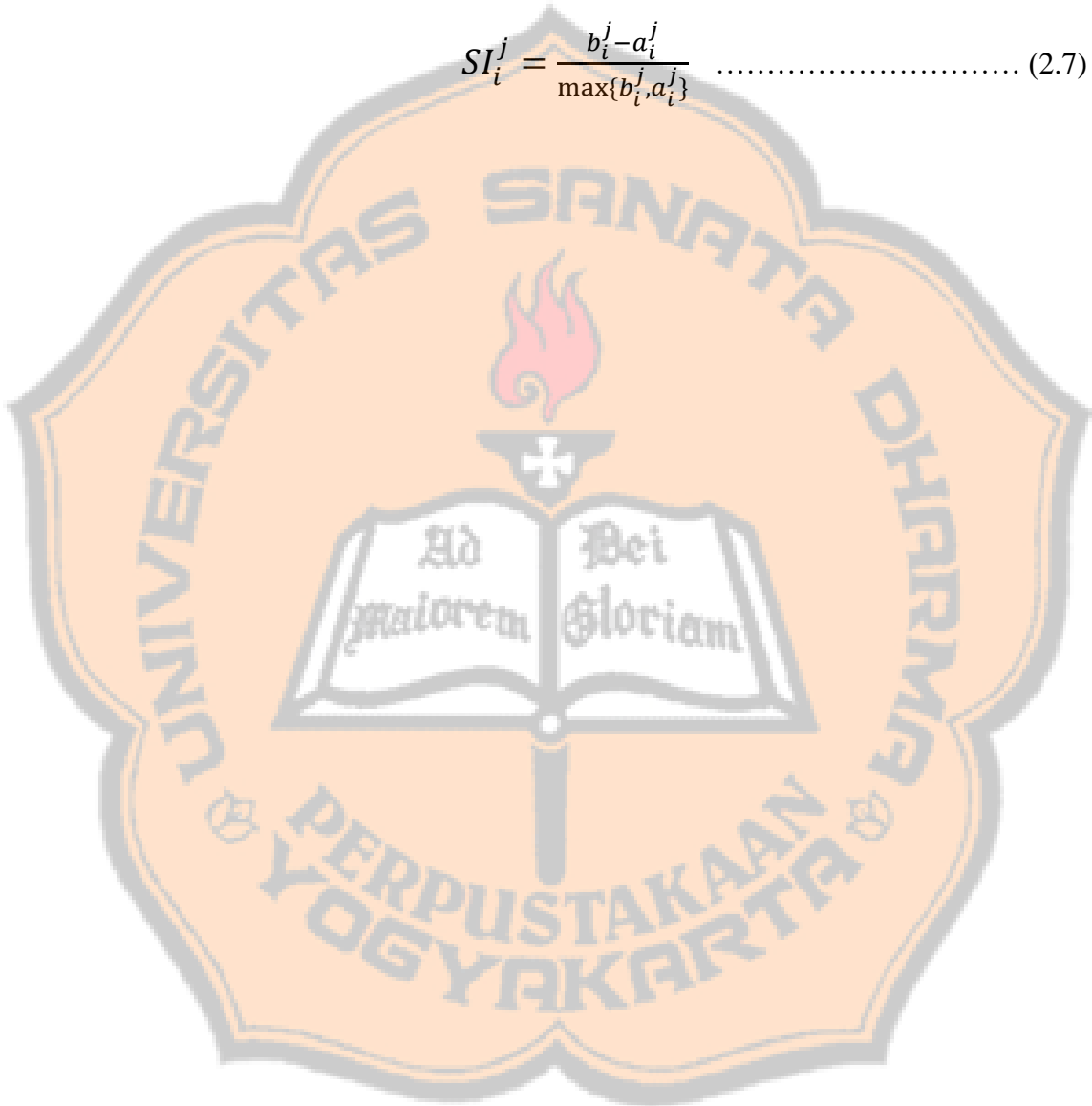
$d(x_i^j, x_r^j)$ merupakan jarak data ke- i dengan data ke- r dalam satu *cluster*, sedangkan m_j adalah jumlah data dalam satu *cluster*.

Berikut rumus b_i :

$$b_i^j = \min\left\{\frac{1}{m_n} \sum_{\substack{r=1 \\ r \neq i}}^{m_j} d(x_i^j, x_r^j)\right\} \dots\dots\dots (2.6)$$

Untuk menghitung SI data ke- i menggunakan persamaan :

$$SI_i^j = \frac{b_i^j - a_i^j}{\max\{b_i^j, a_i^j\}} \dots\dots\dots (2.7)$$



BAB III METODOLOGI PENELITIAN

3.1. Metode Pengumpulan Data

Dalam penelitian ini, pengumpulan data dilaksanakan dengan wawancara, dan pengumpulan data dari arsip nilai sekolah. Penjelasan lebih lanjut akan akan dijabarkan seperti berikut :

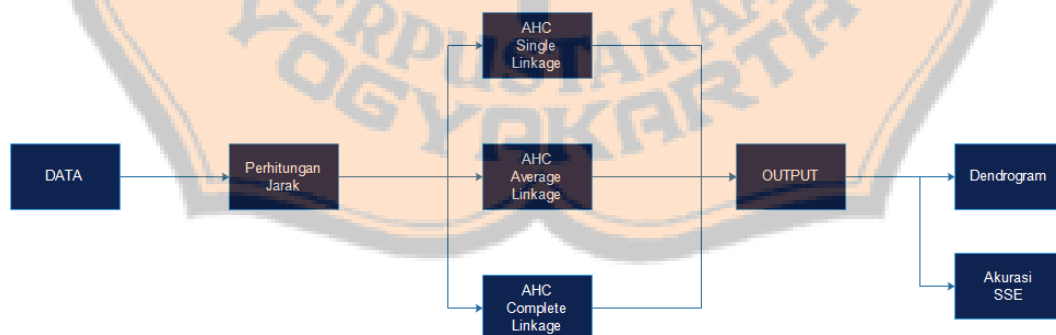
3.1.1. Wawancara

Wawancara dilakukan dengan Guru SD Model Kabupaten Sleman, wawancara dengan guru dilakukan untuk mendapatkan gambaran nilai pada setiap tingkatan kelas yang digunakan komposisi untuk menentukan capaian belajar siswa dan pembagian kelas 6. Hal tersebut dibutuhkan untuk mengetahui struktur dan komposisi nilai yang nantinya akan diproses menggunakan *agglomerative hierarchical clustering*. Wawancara juga sebagai penunjang keakuratan data dan informasi yang dikumpulkan oleh penulis.

3.1.2. Pengumpulan Data Dari Arsip Nilai Sekolah

Data diperoleh dari arsip nilai sekolah SD Model Kabupaten Sleman, nilai yang digunakan adalah nilai UTS dan UAS dari siswa. Arsip nilai yang telah didapatkan kemudian disesuaikan dengan format dan diolah untuk mendapatkan hasil *cluster*.

3.2. Analisa Data



Gambar 3.1 Block diagram proses program

Data yang digunakan penulis untuk penelitian merupakan data yang diperoleh dari arsip SD Model Kabupaten Sleman. Data yang digunakan merupakan data nilai UTS dan UAS. Berdasarkan nilai ini, akan menjadikan pengelompokan capaian belajar siswa. Sehingga nantinya sekolah dapat mengambil tindakan untuk melakukan penadmpingan dan pembelajaran siswa. Berikut ini adalah contoh data yang akan digunakan untuk perhitungan dengan *agglomerative hierarchical clustering* :

Tabel 3.1 Sample data nilai siswa

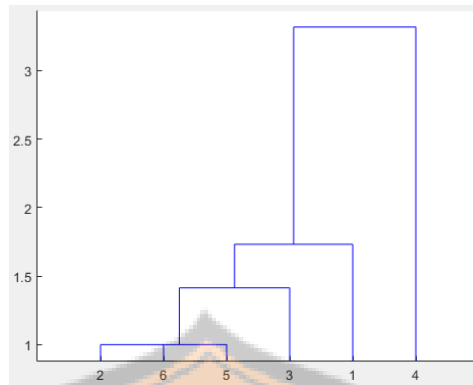
Siswa	MTK	B. Indonesia	IPA
1	7	8	6
2	6	9	8
3	8	7	9
4	9	8	8
5	6	9	7
6	7	8	7

Selanjutnya dilakukan perhitungan jarak dengan menggunakan *euclidean distance*, sehingga menghasilkan matriks jarak :

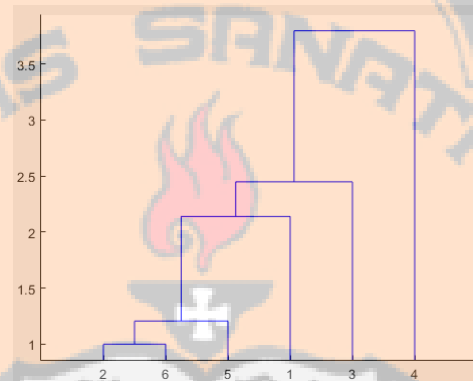
Tabel 3.2 Matriks Jarak dari Data Sampel

	1	2	3	4	5	6
1	0,00	2,45	3,32	2,83	1,73	1,00
2	2,45	0,00	3,00	3,16	1,00	1,73
3	3,32	3,00	0,00	1,73	3,46	2,45
4	2,83	3,16	1,73	0,00	3,32	2,24
5	1,73	1,00	3,46	3,32	0,00	1,41
6	1,00	1,73	2,45	2,24	1,41	0,00

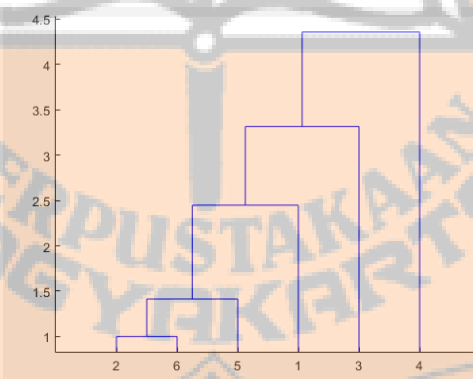
Dari matriks jarak, akan dilakukan pengolahan data dengan *agglomerative hierarchical clustering single linkage*, *complete linkage*, dan *average linkage*. Langkah-langkah perhitungan seperti pada bab kedua dalam tulisan ini. Pengolahan data dengan menggunakan matlab akan menghasilkan dendrogram *single linkage*, *complete linkage*, dan *average linkage*.



Gambar 3.2 Dendrogram *Single Linkage*



Gambar 3.3 Dendrogram *Average Linkage*



Gambar 3.4 Dendrogram *Complete Linkage*

Dari dendrogram *single linkage*, *average linkage* dan *complete linkage* dapat ditentukan *cluster* yang diinginkan dengan perpotongan atau *cut-off* pada ketinggian tertentu.

Tabel 3.3 Contoh *Cluster Single Linkage*

Cluster 1	Cluster 2	Cluster 3
2	1	4
6		
5		
3		

Tabel 3.4 Contoh *Cluster Average Linkage*

Cluster 1	Cluster 2	Cluster 3
2	3	4
5		
6		
1		

Tabel 3.5 Contoh *Cluster Complete Linkage*

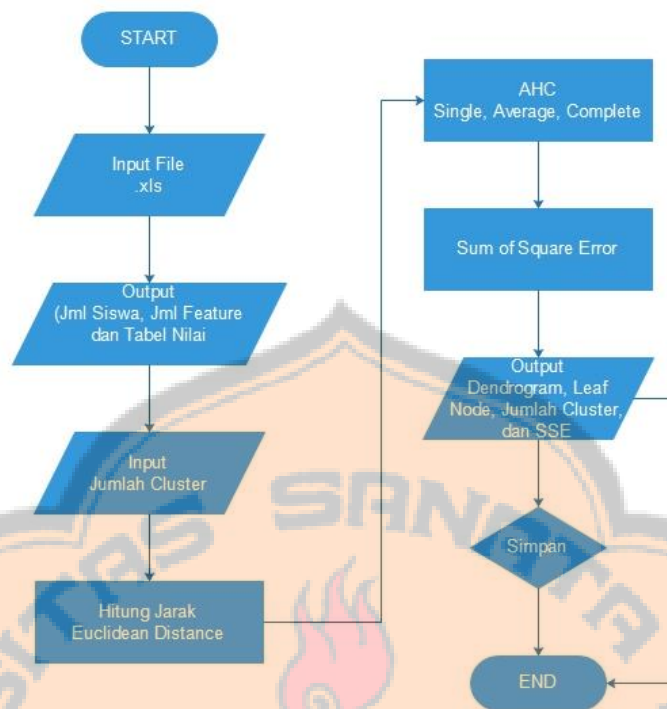
Cluster 1	Cluster 2	Cluster 3
2	3	4
5		
6		
1		

3.3. Desain *User Interface*



Gambar 3.5 User Interface

User interface pengelompokan capaian belajar siswa SD menggunakan *agglomerative hierarchical clustering* hanya memiliki satu tampilan. Dalam *user interface* terdapat *button* “input” untuk memasukkan data yang akan diolah. Setelah data dimasukkan, “Jumlah Siswa” akan menampilkan jumlah data pada file yang akan diolah, “Jumlah *Feature*” menampilkan jumlah mata pelajaran pada data. Dan data awal sebelum diolah akan ditampilkan dalam table “Data Awal”. Selanjutnya, *user* harus mengisi Jumlah *cluster* yang akan digunakan sebagai hasil perhitungan *clustering* pada “Jml Cluster”. Setelah tombol proses dipilih maka data akan diolah dengan algoritma *agglomerative hierarchical clustering*, sehingga akan mendapatkan hasil label *leaf note*, *Sum Square Error*, dendrogram, dan jumlah dari masing-masing metode yaitu *single linkage*, *average linkage* dan *complete linkage*. *Button* simpan digunakan untuk menyimpan tabel *leaf node* pada masing-masing metode *clustering* dalam file .xls. Proses pengolahan data pada sistem dapat dilihat pada gambar 3.6.



Gambar 3.6 Flowchart Proses Data

3.4. Kebutuhan Software dan Hardware

Spesifikasi *Software* dan *Hardware* yang digunakan adalah :

1. *Software*

- a) Matlab version 8.0 R2015a

2. *Hardware*

- a) *Processor* : Intel(R) Core(TM) i3-4030U
- b) *RAM* : 4,00 GB

BAB IV

IMPLEMENTASI SISTEM DAN ANALISA HASIL

Pada penelitian ini membahas mengenai hasil dari pengelompokan nilai siswa SD Model Kabupaten Sleman dengan menggunakan *agglomerative hierarchical clustering*.

4.1. Implementasi

Landasan teori dan metodologi yang telah disampaikan berkurang manfaatnya jika disertai dengan implementasi. Implementasi dibagi dalam bagian, yaitu berkaitan dengan pengolahan data dan *user interface* sistem.

4.2. Pengolahan Data

Data yang digunakan dalam tulisan ini berjumlah 67 data. Data tersebut merupakan data nilai siswa yang meliputi nilai ujian tengah semester dan nilai ujian akhir semester di SD Model Kabupaten Sleman, dapat dilihat dalam tabel berikut.

Tabel 4.1 Data nilai siswa

Ind	BID	MTK	IPA	BID	MTK	IPA
00184	93	100	94	86	100	97
00182	88	89	97	83	100	94
00180	94	75	99	91	92	98
00159	94	87	99	85	85	98
00197	95	72	99	89	85	98
00188	93	90	100	82	80	92
00211	91	71	99	84	90	94
00214	96	69	81	90	96	94
00220	89	80	90	83	95	87
00210	89	93	94	68	89	89
00217	91	63	97	88	80	100
00565	84	82	93	77	100	81
00194	93	73	94	78	76	94
00187	88	88	84	84	75	88
00185	91	70	94	77	89	80

00157	85	66	95	84	75	94
00199	91	65	93	68	94	88
00174	88	62	89	80	89	87
00221	80	72	96	80	78	88
00216	83	84	79	70	92	84
00225	89	58	96	80	82	85
00212	86	68	84	81	71	94
00309	85	59	89	89	71	89
00470	81	70	91	73	85	81
00172	86	60	83	77	96	76
00192	91	68	89	72	75	80
00173	85	62	82	85	75	85
00224	83	62	93	82	68	83
00228	86	55	81	83	80	85
00223	78	67	86	74	85	80
00226	83	60	86	73	83	84
00227	91	60	81	87	58	91
00204	79	60	89	76	75	87
00207	84	54	83	83	91	70
00209	81	72	79	77	83	71
00215	91	49	93	83	55	92
00219	86	57	96	75	61	88
00472	74	66	81	83	88	69
00310	83	62	84	86	71	74
00198	88	47	87	74	76	86
00179	85	38	89	83	68	91
00167	86	52	79	81	74	81
00163	86	54	88	83	55	84
00158	85	56	83	78	63	83
00200	81	70	78	56	79	80
00162	83	47	87	74	72	80

00170	86	45	79	76	74	79
00201	86	40	92	81	52	88
00169	81	54	89	65	65	81
00177	84	53	90	73	57	75
00203	83	40	87	76	68	78
00161	78	45	85	80	61	82
00570	81	35	91	71	61	87
00181	81	49	86	74	65	70
00213	86	48	86	71	53	80
00202	69	46	79	84	53	92
00164	77	54	71	74	72	72
00190	88	35	87	69	50	91
00475	80	49	77	76	61	71
00165	75	37	92	60	58	84
00196	66	46	79	69	58	88
00232	79	37	84	73	51	80
00189	79	34	83	68	55	85
00168	82	47	73	78	58	63
00156	83	52	60	73	58	63
00186	63	38	89	71	41	85
00218	74	40	65	69	42	78

Data sejumlah 67 dengan atribut berjumlah 6 ini yang akan diproses melalui *agglomerative hierarchical clustering* dan akhirnya perhitungan akurasi menggunakan *Sum Square Error*.

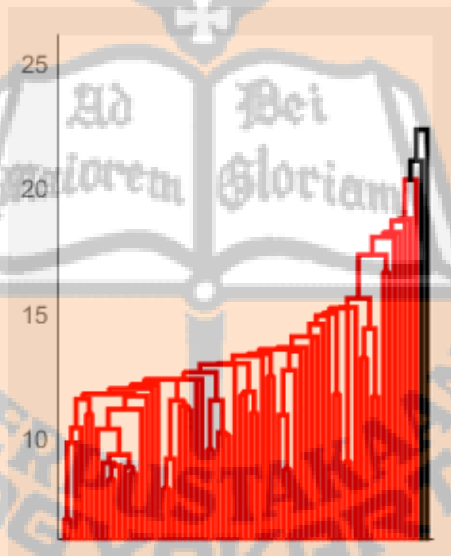
4.3. Clustering

Analisis kelompok merupakan pekerjaan mengelompokkan data (objek) yang ditemukan dalam data yang menggambarkan objek dan hubungan diantaranya (Prasetyo E, 2012). Hal ini dilakukan agar data dalam satu kelompok *clustering* merupakan objek yang memiliki kemiripan tinggi atau dengan kata lain memiliki similaritas yang tinggi.

Dalam tulisan ini, pengelompokan menggunakan *agglomerative hierarchical clustering* (AHC). Data input berupa excel selanjutnya diproses menggunakan AHC. Data nilai siswa pada SD Model Kabupaten Sleman sejumlah 67 dengan fitur 6 yang diproses dengan menggunakan AHC sehingga akan terbentuk kelompok-kelompok. Setiap pembentukan kelompok akan diuji menggunakan *Sum Of Square Error* (SSE). Proses pengelompokan dan pengujian dengan SSE dilakukan dengan menggunakan sistem yang telah dibuat.

Proses AHC pada sistem terdiri dari tiga metode, yaitu *single linkage*, *average linkage* dan *complete linkage*. Setiap metode akan menggunakan perhitungan jarak *Euclidean distance*. Hasil akhir dari pengelompokan berupa tabel kelompok, dendrogram dan hasil SSE. Berikut adalah hasil dari proses tersebut.

4.3.1. *Single Linkage*



Gambar 4.1 Implementasi – Dendrogram *single linkage* dengan cluster 3

Proses pengelompokan dengan menggunakan metode *single linkage* dengan tiga *cluster* menunjukkan bahwa hasil SSE bernilai 244331. Hasil dendrogram *single linkage* seperti pada gambar 4.1.

244331

Gambar 4.2 Implementasi – Hasil SSE *single linkage*

Dapat dilihat pada table 4.1 jumlah data pada masing-masing *cluster* tidak seimbang atau memiliki selisih jumlah data yang sangat jauh pada setiap *cluster*.

Berikut ini *source code* yang digunakan dalam perhitungan metode *single linkage* :

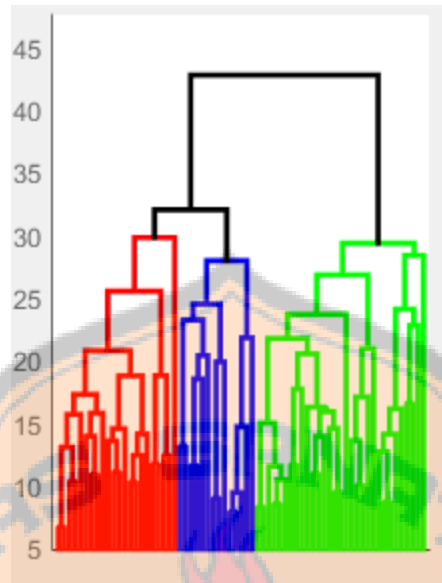
```
%SINGLE
elseif strcmp(linkage,'single')
while loop ~= kolom
[v,k] = min(dataEuclidian);
% Mencari dataHasil
[hasilData,i,j,I,J]=olahDataHasil...
(loop,baris,v,k,vektorIndeks);
dataHasil(loop,:)=hasilData(loop,:);
dataEuclidian(I) = ...
min(dataEuclidian(I),dataEuclidian(J));
[dataEuclidian~,baris,...
jmlPoinDiSetiapCluster,vektorIndeks]=updateMatriks...

(dataEuclidian,J,baris,jmlPoinDiSetiapCluster,..
.
kolom,loop,vektorIndeks,i,j);
loop=loop+1;
end
```

Tabel 4.2 Jumlah data masing-masing *cluster*

Cluster	Jumlah
1	65
2	1
3	1

4.3.2. Average Linkage



Gambar 4.3 Implementasi – Dendrogram *average linkage* dengan cluster 3

Dilihat dari gambar 4.3 dapat dilihat dendrogram *average linkage* lebih seimbang dibandingkan dengan dendrogram *single linkage*. Proses pengelompokan dengan menggunakan metode *average linkage* dengan tiga memiliki hasil SSE bernilai 192467. Hasil SSE pada *average linkage* lebih rendah dibandingkan dengan hasil SSE pada *single linkage*.

192467

Gambar 4.4 Implementasi – Hasil SSE *average linkage*

Dapat dilihat pada table 4.1 jumlah data pada setiap *cluster* lebih seimbang atau selisih jumlah pada setiap *cluster* tidak terlalu jauh jika dibandingkan dengan hasil pada *single linkage*. Jumlah data terkecil pada *cluster* satu dengan 14 dan tertinggi pada *cluster* 3 dengan 31.

Berikut ini *source code* yang digunakan dalam perhitungan metode *average linkage* :

```
%AVERAGE
    elseif strcmp(linkage,'average')
        while loop ~= kolom
            [v,k] = ...
            average(dataEuclidian,baris,...
            jmlPoinDiSetiapCluster,vektorIndeks);
        % Mencari dataHasil
```

```

[hasilData,i,j,I,J]=olahDataHasil...
(loop,baris,v,k,vektorIndeks);
    dataHasil(loop,:)=hasilData(loop,:);
    dataEuclidian(I) = dataEuclidian(I) +
...
    dataEuclidian(J);

[dataEuclidian,~,baris,jmlPoinDiSetiapCluster,vektorIndeks]=updateMatriks...

(dataEuclidian,J,baris,jmlPoinDiSetiapCluster,..
.
kolom,loop,vektorIndeks,i,j);
    loop=loop+1;
end
else
end
end
function [v,k]=average...
(dataEuclidian,baris,jmlPoinDiSetiapCluster,vektorIndeks)

    p = (baris-1):-1:2;
    %1x64
    I = zeros(baris*(baris-1)/2,1);
    %2145x1
    I(cumsum([1 p])) = 1;
    I = cumsum(I);
    J = ones(baris*(baris-1)/2,1);
    J(cumsum(p)+1) = 2-p;
    J(1)=2;
    J = cumsum(J);
    W =jmlPoinDiSetiapCluster(vektorIndeks(I)).
*jmlPoinDiSetiapCluster(vektorIndeks(J));
    [v, k]=min(dataEuclidian./W);

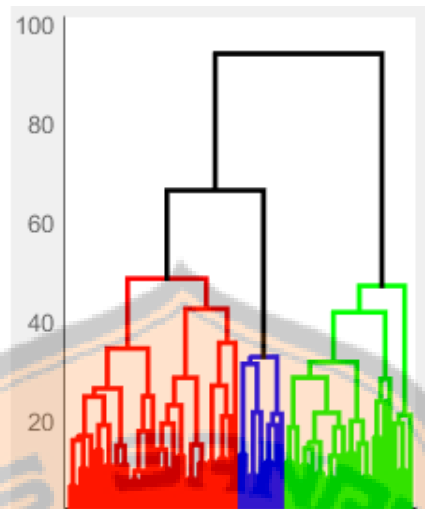
end

```

Tabel 4.3 Jumlah data masing-masing cluster

Cluster	Jumlah
1	14
2	22
3	31

4.3.3. Complete Linkage



Gambar 4.5 Implementasi – Dendrogram *complete linkage* dengan cluster 3

Hasil dendrogram pada pengelompokan menggunakan *complete linkage* dapat dilihat pada gambar 4.5. Dari *dendrogram complete linkage* dapat dilihat *cluster* dua yang ditandai dengan warna biru memiliki jumlah yang paling sedikit. Pada pengelompokan dengan *complete linkage* menghasilkan SSE dengan nilai 154457. Nilai SSE pada *complete linkage* memiliki nilai yang paling rendah jika dibandingkan dengan hasil SSE pada *single linkage* dan *average linkage*.

154457

Gambar 4.6 Implementasi – Hasil SSE *average linkage*

Namun jika dilihat dari jumlah data pada setiap *cluster*, metode *complete linkage* masih kurang seimbang dibandingkan dengan metode *average linkage*. Pada *cluster* satu metode *complete linkage* memiliki jumlah data 9 untuk *cluster* dengan jumlah data terendah dan *cluster* dua yang memiliki jumlah data 33.

Berikut ini *source code* yang digunakan dalam perhitungan metode *complete linkage* :

```
%COMPLETE
    if strcmp(linkage,'complete')
        while loop ~= kolom
            [v,k] = min(dataEuclidian);
% Mencari dataHasil
```



```

[hasilData,i,j,I,J]=olahDataHasil...
(loop,baris,v,k,vektorIndeks);
dataHasil(loop,:)=hasilData(loop,:);
dataEuclidian(I) = ...
max(dataEuclidian(I),dataEuclidian(J));

[dataEuclidian,~,baris,jmlPoinDiSetiapCluster,vektorIndeks]=updateMatriks...

(dataEuclidian,J,baris,jmlPoinDiSetiapCluster,...
.
kolom,loop,vektorIndeks,i,j);
loop=loop+1;
end

```

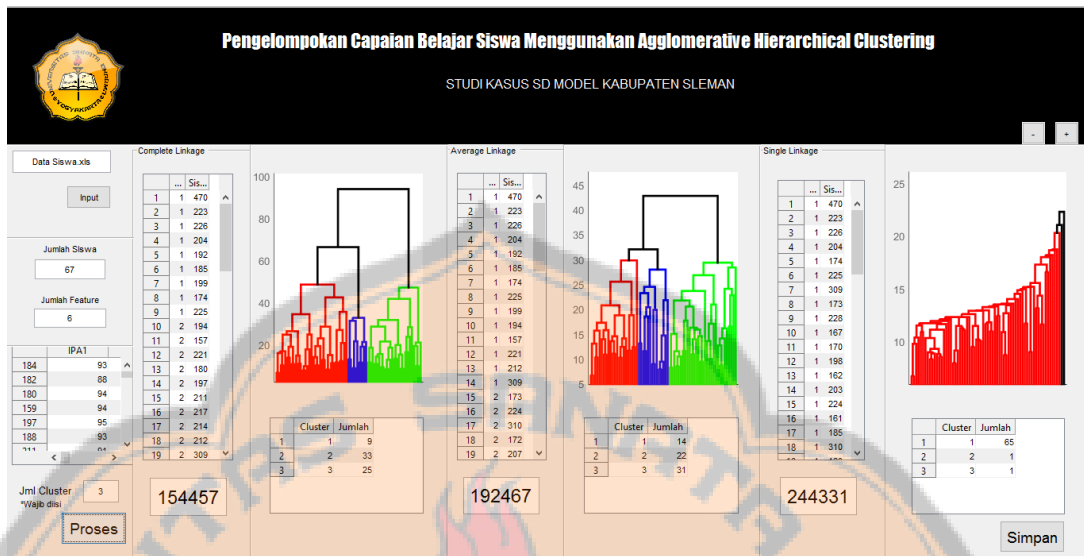
Tabel 4.4 Jumlah data masing-masing *cluster*

Cluster	Jumlah
1	9
2	33
3	25

4.4. User Interface

Dalam membuat sistem implementasi *agglomerative hierarchical clustering* untuk pengelompokan capaian belajar ini penulis menggunakan Matlab *version 8.0 R2015a*. *User interface* sistem telah dipaparkan dalam bab sebelumnya sebelum diimplementasikan dan digunakan untuk melakukan proses pengelompokan dengan AHC serta menghitung SSE. Sistem dibuat dengan menampilkan hasil pengelompokan dengan menampilkan tiga metode dalam satu *graphical user interface* (GUI).

Gambar 4.7 adalah contoh *user interface* dari keseluruhan sistem yang telah terbentuk.



Gambar 4.7 Tampilan keseluruhan sistem

Dalam proses pengelompokan data terdapat tiga langkah utama, yaitu *input* data dan proses AHC & SSE.

4.4.1. Input Data

Saat sistem dijalankan, langkah pertama yang harus dilakukan *user* adalah meng-*input*-kan. Tombol *input* digunakan untuk meng-*input*-kan data yang berekstensi *.xls. Setelah data di-*input*-kan, sistem akan menampilkan jumlah data pada *textfield* Jumlah Siswa, jumlah atribut yang akan ditampilkan pada *textfield* Jumlah Feature dan keseluruhan data yang telah di-*input*-kan pada tabel Data Awal. Selanjutnya yang harus dilakukan *user* adalah dengan menentukan jumlah *cluster* yang akan menentukan *cutoff* hasil clustering pada Jml Cluster. Gambar 4.8 adalah contoh proses input.

Data Siswa.xls

Input

Jumlah Siswa

67

Jumlah Feature

6

IPA1	
184	93
182	88
180	94
159	94
197	95
188	93
111	91

Jml Cluster

3

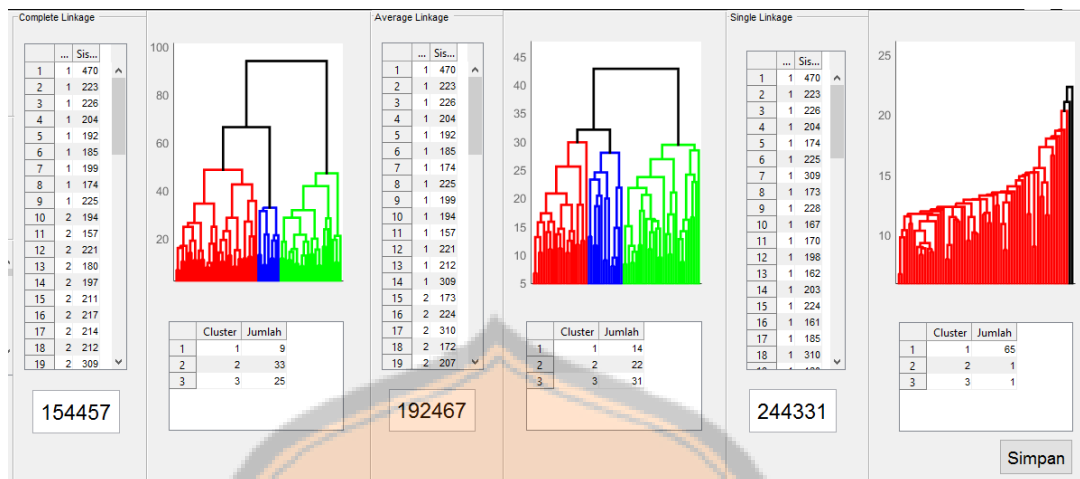
*Wajib diisi

Proses

Gambar 4.8 Implementasi – Input data

4.4.2. Proses AHC dan SSE

User interface dari implementasi hasil proses AHC dan SSE dapat dilihat pada gambar 4.9.



Gambar 4.9 Implementasi – Hasil proses AHC dan SSE

Ketika tombol "Proses" seperti pada gambar 4.9 diklik maka akan muncul hasil proses *clustering* metode *complete linkage*, *average linkage*, dan *single linkage*. Secara berurutan dari atas kebawah dan kanan ke kiri dari gambar 4.9 adalah :

1. Tabel label dari dendrogram yang merupakan No Induk Siswa yang disertai dengan label *cluster*.
2. Dendrogram hasil proses AHC.
3. Hasil SSE.
4. Tabel jumlah objek per *cluster*.

Dibagian kanan bawah terdapat tombol Simpan yang berguna untuk menyimpan Tabel label dari dendrogram dalam format *.xls.

4.5. Analisa Hasil

Implementasi yang telah dijelaskan di bab 4.1 membantu analisis terhadap pengelompokan capaian belajar siswa SD Model Kabupaten Sleman. Analisis dilakukan terhadap 67 siswa dalam satu tahun angkatan. Dengan demikian dapat diketahui kelompok siswa berdasarkan beberapa komposisi nilai tanpa menggunakan bobot.

Dalam pengelompokan ini, yang dijadikan sebagai atribut pengelompokan adalah nilai bahasa indonesia, matematika dan IPA yang meliputi nilai ujian tengah semester dan ujian akhir semester. Nilai tersebut

diolah dengan menggunakan *agglomerative hierarchical clustering* (AHC) yang meliputi metode *single linkage*, *average linkage*, dan *complete linkage*. Dan dalam setiap metode akan diuji dengan menggunakan *Sum Square Error* (SSE).

Berikut ini *source code* yang digunakan dalam perhitungan akurasi SSE:

```
%-----SSE
function [nilaiSSE]=hitungSSE(data)
rerata=mean(data,2);
[m,n]=size(data);
nilaiSSETemp=zeros(m,n);
for i=1:m
    for j=1:n
nilaiSSETemp1(i,j)=power(((data(i,j)-
rerata(i,1))),2);
    end
end
nilaiSSETemp2=sum(nilaiSSETemp1,2);
nilaiSSE=double(sum(nilaiSSETemp2));
end
```

Tabel 4.5 Analisa Keseimbangan Dendrogram

Metode	SSE	Dendrogram	Jumlah obyek per Cluster		
			1	2	3
Single Linkage	244331	Tidak Seimbang	65	1	1
Average Linkage	192467	Seimbang	14	22	31
Complete Linkage	154457	Tidak Seimbang	9	33	25

Dalam sistem, dengan menentukan 3 *cutoff* berurutan *cluster 1, 2, dan 3* memiliki sama dengan kelas B, C, dan A. Kelas A, B, dan C berdasarkan pada penetapan pengelompokan secara manual oleh SD Model Kabupaten Sleman.

Penetapan persamaan *cluster* dalam sistem dan pengelompokan secara manual berdasarkan percobaan perhitungan standar deviasi terhadap

metode *single linkage*, *average linkage* dan *complete linkage* yang digunakan dalam. Percobaan perhitungan standar deviasi dapat dilihat seperti berikut.

4.5.1. Single Linkage

Tabel 4.6 Percobaan perhitungan standar deviasi *single linkage*

CLUSTER 2		CLUSTER 3		CLUSTER 1	
MIN	444	MIN	368	MIN	387
MAX	444	MAX	368	MAX	570
AVG	444.0	AVG	368.0	AVG	468.3
STD		STD		STD	44.1
UP SD	-	UP SD		UP SD	512.4
DO SD	-	DO SD		DO SD	424.2

Dilihat dari tabel 4.6, perhitungan standar deviasi menggunakan jumlah nilai dari masing-masing siswa yang dapat didapatkan dari tabel data nilai siswa. Untuk *cluster* dua memiliki jumlah nilai terendah (*MIN*), jumlah nilai tertinggi (*MAX*) dan rata-rata (*AVG*) sebanyak 444, hal ini disebabkan karena *cluster* dua hanya terdapat obyek didalamnya. Dengan demikian standar deviasi tidak perlu dihitung. Hal yang sama juga terjadi pada *cluster* tiga yang hanya memiliki satu obyek dengan nilai *MIN*, *MAX* dan *AVG* sebanyak 368. Dan pada *cluster* satu yang terdapat 65 obyek memiliki nilai *MIN* 387, nilai *MAX* 570 dan nilai *AVG* 468.3. Pehitungan standar deviasi (*STD*) menghasilkan nilai 44.1. Dengan nilai *STD* 44.1 maka menghasilkan batas atas (*UP SD*) pada nilai 512.4 dan batas bawah (*DO SD*) pada nilai 424.2. Dari data hasil *clustering* dengan menggunakan *single linkage* maka data jumlah nilai yang diatas *UP SD* ada sebanyak 12 siswa dengan dan dibawah *DO SD* ada sebanyak 12 siswa. (Lampiran 1)

4.5.2. Average Linkage

Tabel 4.7 Percobaan perhitungan standar deviasi *average linkage*

CLUSTER 2		CLUSTER 1		CLUSTER 3	
MIN	444	MIN	466	MIN	368
MAX	570	MAX	508	MAX	470
AVG	506.6	AVG	486.6	AVG	428.9
STD	36.7	STD	13.3	STD	25.5
UP SD	543.3	UP SD	500.0	UP SD	454.3
DO SD	469.9	DO SD	473.3	DO SD	403.4

Tabel 4.7 menunjukkan hasil percobaan perhitungan standar deviasi untuk metode *average linkage*. Metode *average linkage* menghasilkan

dendrogram yang paling seimbang, pada *cluster* dua memiliki nilai *MIN* 444, nilai *MAX* 570, dan nilai *AVG* 506.6. Dari nilai tersebut menghasilkan *STD* 36.7 dengan *UP SD* 543.3 dan *DO SD* 469.9. Dari hasil tersebut siswa yang diluar batas *UP SD* sebanyak 4 dan diluar batas *DO SD* sebanyak 5.

Dan pada *cluster* satu menghasilkan nilai *MIN* 466, nilai *MAX* 508 dan nilai *AVG* 486.6. Pada *cluster* satu yang memiliki jumlah obyek terendah dengan 14 siswa, memiliki nilai *STD* 13.3. Dengan nilai *UP SD* 500 dengan jumlah data melebihi sebanyak 2 siswa dan nilai *DO SD* 473.3 dengan banyak data dibawahnya sebanyak 3 siswa.

Sedangkan pada *cluster* tiga menghasilkan nilai *MIN* 368, *MAX* 470 dan *AVG* 428.9. Dengan *STD* 25.5, *UP SD* 454.3 dan *DO SD* 403.4. Menghasilkan jumlah data diatas *UP SD* sebanyak 5 siswa dan dibawah *DO SD* sebanyak 4 siswa.(Lampiran 2)

4.5.3. Complete Linkage

Tabel 4.8 Percobaan perhitungan standar deviasi *complete linkage*

CLUSTER 2		CLUSTER 1		CLUSTER 3	
MIN	439	MIN	466	MIN	368
MAX	570	MAX	501	MAX	463
AVG	495.8	AVG	482.9	AVG	421.8
STD	35.5	STD	13.7	STD	22.6
UP SD	531.3	UP SD	496.6	UP SD	444.4
DO SD	460.4	DO SD	469.2	DO SD	399.1

Dapat dilihat pada tabel 4.8, percobaan perhitungan standar deviasi pada metode *complete linkage* menghasilkan nilai *MIN* 439, *MAX* 570, *AVG* 495.8, *STD* 35.5, *UP SD* 531.3 dan *DO SD* 460.4 untuk *cluster* dua. Dari hasil tersebut jumlah data yang melebihi *UP SD* sebanyak 6 siswa dan dibawah *DO SD* sebanyak 5 siswa.

Sedangkan untuk *cluster* satu menghasilkan nilai *MIN* 466, *MAX* 501 dan *AVG* 482.9. Dari hasil tersebut perhitungan *STD* menghasilkan 13.7 dan memiliki *UP SD* 496.6 serta *DO SD* 469.2. Dengan demikian pada *cluster* satu memiliki data melebihi *UP SD* sebanyak 2 siswa dan dibawah *DO SD* sebanyak 2 siswa.

Dan *cluster* tiga menghasilkan nilai *MIN* 368, *MAX* 463 dan *AVG* 421.8. Dengan perhitungan standar deviasi menghasilkan *STD* 22.6, *UP SD* 444.4

dan DO SD 399.1. *Cluster* tiga memiliki jumlah data melebihi UP SD 4 siswa dan dibawah DO SD 3 siswa. (Lampiran 3)

Jika dilihat dari tabel 4.4, dapat disimpulkan nilai SSE terendah terdapat pada metode *complete linkage*. Namun dendrogram yang paling seimbang dari ketiga metode adalah *average linkage* dengan selisih jumlah obyek terbesar adalah 17. Berbeda jika dibandingkan dengan metode *single linkage* yang memiliki selisih jumlah obyek terbesar 64 dan metode *complete linkage* 24.

Dari proses *clustering* tersebut dapat dilihat dari dua sisi, yaitu:

1. Nilai SSE terendah dari ketiga metode, maka hasil dari metode *complete linkage* dipertimbangkan untuk menjadi pilihan. Namun jumlah obyek pada setiap *cluster* kurang begitu merata atau seimbang. Terdapat selisih yang cukup besar antara *cluster* 1 dan 2.
2. Jika yang dipilih adalah dendrogram yang paling seimbang, maka metode *average linkage* dapat dijadikan pilihan. Karena dibandingkan dengan metode *single linkage* dan *complete linkage*, metode *average linkage* memiliki jumlah obyek yang merata pada setiap *cluster* dengan selisih tertinggi antara *cluster* dengan jumlah obyek terbanyak dan terendah adalah 16. Jumlah tersebut menjadi yang terendah jika dibandingkan dengan metode *single linkage* dan *complete linkage* dengan masing-masing 64 dan 24.

4.5.4. *Silhouette Index*

Dalam kasus di SD Model Kabupaten Sleman yang dipergunakan adalah hasil dengan dendrogram yang paling seimbang. Hasil pengelompokan yang memiliki dendrogram yang paling seimbang adalah *average linkage*, Namun memiliki hasil SSE lebih tinggi daripada *complete linkage* yang memiliki hasil SSE terendah. Dari hal tersebut maka dilakukan pengujian dengan *silhouette index*.

Dalam pengujian *silhouette index* digunakan hasil *clustering* tanpa SSE yang telah dipilih yaitu hasil *clustering average linkage*. Proses pengujian seperti berikut :

4.5.4.1. Perhitungan Jarak

Perhitungan jarak yang digunakan adalah *euclidian dintance*. Perhitungan dilakukan dengan menggunakan semua data nilai yang dimiliki masing-masing siswa.

4.5.4.2. Perhitungan ai, bi, dan SI (Average Linkage)

Menghasilkan tabel seperti berikut :

Tabel 4.9 Pengujian silhouette index cluster 2

	A	b	SI
1	37.02507404	44.13653298	0.161124095
2	29.83710468	34.56169901	0.136700292
3	28.21504028	28.49948251	0.00998061
4	28.71598441	30.65159568	0.063148793
5	27.75076505	24.84438083	-0.104731679
6	29.52692616	30.34436693	0.026938798
7	25.32971007	21.35417459	-0.15695148
8	29.76790772	28.5143451	-0.04211121
9	23.79552076	24.45837678	0.027101391
10	31.39467278	32.78586549	0.042432697
11	13.11310475	22.31476148	0.412357387
12	27.76280998	28.42916317	0.023439072
13	29.03971088	27.78104351	-0.043342972
14	30.3695732	31.26541081	0.028652674
15	30.83039077	23.67472256	-0.232097876
16	29.1500608	17.89624768	-0.386064825
17	32.21336983	18.69303901	-0.419711781
18	34.97611495	27.22113817	-0.221722075
19	29.84562709	24.10685392	-0.192281876
20	33.00286539	27.48598279	-0.167163746
21	32.93610394	22.97048866	-0.3025742
22	38.00484764	29.04716295	-0.235698477

Tabel 4.10 Pengujian *silhouette index cluster 1*

	a	b	SI
23	19.27275845	26.13231402	0.262493232
24	18.32897734	24.36068005	0.247599932
25	15.76725724	27.38102826	0.42415394
26	21.75918849	28.74067223	0.242913029
27	16.38474578	25.07470361	0.346562734
28	17.00041465	25.55656583	0.334792681
29	16.7350265	28.22123411	0.407005858
30	19.46739487	28.76243001	0.32316585
31	21.00786919	28.4559827	0.261741567
32	16.41251108	25.17149302	0.347972285
33	17.85621695	27.90235946	0.360046344
34	16.02609523	26.43582438	0.393773578
35	16.02821054	27.87655349	0.425028975
36	16.75239246	26.97204093	0.378897855

Tabel 4.11 Pengujian *silhouette index cluster 3*

	a	b	SI
37	28.81186714	19.55519044	-0.321280001
38	28.74438455	29.73828356	0.033421532
39	26.07863444	33.74399277	0.227162161
40	25.08668538	25.00799267	-0.003136832
41	24.50089252	23.20547737	-0.052872162
42	24.96175449	33.15414369	0.247100009
43	23.91128287	23.37067529	-0.022608891
44	22.45824544	30.82490559	0.271425329
45	20.72087135	24.55310591	0.156079421
46	21.07713248	25.11424531	0.160749916
47	22.96970959	28.29298848	0.188148343
48	24.14488925	40.09780951	0.397850168

49	23.06601228	26.98396925	0.145195725
50	21.70027801	31.71706886	0.315817041
51	20.86314378	32.29159102	0.353914034
52	19.13567847	32.39756612	0.40934827
53	23.15214454	38.94669588	0.405542781
54	21.9708002	31.35538628	0.299297416
55	20.73528199	36.18454216	0.426957459
56	27.53813311	41.77984982	0.340875249
57	27.75183313	32.09176652	0.135235104
58	26.96775426	45.96735527	0.41332813
59	22.01970955	35.43101561	0.378518815
60	27.52913174	43.51888418	0.367421011
61	26.18952213	40.26673454	0.349599057
62	23.75213332	44.5077292	0.466336887
63	22.58069134	44.41792343	0.49163109
64	28.14292575	43.26275113	0.349488301
65	34.29538599	48.5530556	0.293651335
66	34.18562582	54.90150829	0.377328112
67	34.22289206	56.70029078	0.396424752

Dari hasil tersebut terdapat hasil minus pada *cluster* dua sebanyak 12 seperti pada tabel 4.9. Pada *cluster* 1 tidak terdapat hasil minus dan *cluster* 3 terdapat 4 hasil minus. Hasil pengujian menggunakan *silhouette index* ketika hasil semakin mendekati 1 maka hasil akan semakin baik, sebaliknya jika hasil mendekati 0 atau bahkan minus maka hasil dari penempatan *cluster* dari data ke-*i* kurang tepat.

Dari hasil tersebut, hasil pengujian tertinggi pada *cluster* 2 sebesar 0.412357 dan terendah 0.009981. Hasil tersebut dirasa tidak terlalu bagus dengan hanya hasil tertinggi masih dibawah 0.5. Hal yang hampir sama juga terjadi pada hasil pengujian *cluster* 1 dan *cluster* 3 dengan hasil tertinggi dibawah 0.5. Secara berurutan *cluster* 1 dan *cluster* 3 adalah 0.425029 dan 0.491631.

4.5.4.3. Perhitungan a_i , b_i , dan SI (*Single Linkage*)

Dari perhitungan *silhouette index* pada single linkage menghasilkan seperti berikut :

Tabel 4.12 Pengujian *silhouette index cluster 1*

	a	b	SI
1	54.63673217	54.12947441	-0.009284189
2	46.1065638	46.22769733	0.002620367
3	50.23930521	54.54356057	0.078914088
4	41.99999952	45.82575695	0.083484872
5	36.64530807	45.7165178	0.198423025
6	41.47012163	43	0.035578567
7	34.64670252	40.5339364	0.145242096
8	39.00414647	43.31281566	0.099477929
9	36.87076121	36.63331817	-0.006439874
10	44.46915533	34.2636834	-0.22949552
11	33.39697389	43.9886349	0.240781762
12	40.84047025	35.51056181	-0.13050556
13	30.99205877	33.13608305	0.064703613
14	28.74251588	30.95157508	0.071371463
15	31.18065183	29.94995826	-0.039469783
16	28.5088111	36.29049462	0.214427596
17	34.43535716	27.98213716	-0.187400989
18	28.68159768	30.96772513	0.073822906
19	29.28389395	31.144823	0.059750831
20	39.68195926	24.12467616	-0.392049268
21	27.58656428	33.7934905	0.183672243
22	28.2130488	30.82207001	0.084647826
23	27.34414934	38.62641583	0.292086808
24	29.2710966	22.24859546	-0.239912472
25	33.88537386	29.93325909	-0.116631877
26	27.78658013	22.29349681	-0.197688355

27	26.00572843	31.27299154	0.168428502
28	26.21438579	33.15116891	0.209247015
29	28.90789598	21.02379604	-0.272731711
30	26.84841833	31.84336666	0.156859932
31	26.6082072	22.11334439	-0.168927684
32	32.3410116	41.15823125	0.21422737
33	25.51739909	26.26785107	0.028569219
34	34.10888175	35.53871129	0.040233016
35	32.73387892	23.3023604	-0.288127128
36	34.09218443	41.41255848	0.176767008
37	29.21550275	35.59494346	0.179223229
38	35.10655697	31.70173497	-0.096985358
39	28.09230566	33.22649545	0.154520954
40	27.97521531	32.06243908	0.127477007
41	33.41612484	41.73727351	0.199369723
42	26.93381242	31.63858404	0.148703609
43	30.71826972	35.63705936	0.138024566
44	26.64660983	31.40063694	0.151399066
45	27.44251283	31.41655614	0.126495192
46	29.52135831	32.81767816	0.100443421
47	36.97522949	34.88552709	-0.05651628
48	29.62342003	25.59296778	-0.136056277
49	31.14920265	35.91656999	0.132734483
50	31.10788997	33.82306905	0.080275952
51	30.41556293	30.64310689	0.007425617
52	36.27540951	34.58323293	-0.046648035
53	30.79658	33.54101966	0.081823382
54	33.36602486	27.89265136	-0.164040323
55	39.18194254	28.26658805	-0.278581249
56	41.81344077	30.62678566	-0.267537301
56	33.06609037	27.5317998	-0.167370576

57	32.86046595	26.83281573	-0.183431672
58	40.91323666	33.346664	-0.184941923
59	38.16317022	25.53429067	-0.330917989
60	38.85619354	22.27105745	-0.426833783
61	39.619569	24.57641145	-0.379690086
62	39.53942078	27.18455444	-0.312469584
63	45.4784353	27.33130074	-0.399027241
64	49.8989183	27.47726333	-0.449341503
65	54.63673217	54.12947441	-0.009284189

Tabel 4.13 Pengujian *silhouette index* cluster 2

	a	b	SI
66	0	36.71079755	1

Tabel 4.14 Pengujian *silhouette index* cluster 3

	a	b	SI
67	0	36.71079755	1

Pada *single linkage* menghasilkan 29 hasil minus yang semuanya terdapat dalam *cluster 1* yang memiliki 65 data. Dan dalam *cluster 2* dan 3 tidak ada hasil minus, tetapi *cluster 2* dan 3 hanya memiliki masing-masing 1 data.

4.5.4.4. Perhitungan a_i , b_i , dan SI (Complete Linkage)

Complete linkage menghasilkan uji *silhouette index* seperti berikut :

Tabel 4.15 Pengujian *silhouette index* cluster 2

	a	b	SI
1	42.31160363	44.57113766	0.050695005
2	34.64528591	34.29163372	-0.010207801
3	42.80471754	44.81367908	0.044829203
4	31.89086042	32.74114153	0.025969807
5	29.13554679	27.06053837	-0.071219134
6	32.18033433	32.32081768	0.004346528
7	27.1880384	21.7612871	-0.199600693

8	31.50545492	28.1906134	-0.105214844
9	27.40990725	23.35660177	-0.147877388
10	35.01409872	32.173811	-0.081118402
11	29.08149514	25.22770346	-0.132516972
12	31.78398909	25.85732365	-0.186467011
13	25.85732819	20.8573855	-0.193366563
14	23.91145674	20.30221277	-0.150942036
15	25.06623248	20.09748741	-0.198224646
16	24.89667563	17.66867729	-0.290319818
17	32.03990314	26.42088903	-0.175375502
18	25.68512428	21.73031854	-0.153972615
19	27.04688913	22.90231752	-0.153236536
20	30.7198124	19.46762832	-0.366284271
21	24.75991325	18.99729271	-0.232739933
22	27.19906758	20.46959394	-0.247415601
23	26.76804062	18.92182702	-0.293118712
24	32.1462475	29.62755117	-0.078351177
25	33.31795018	23.87330746	-0.283470102
26	29.69920995	21.52022032	-0.275394182
27	33.5772099	25.4390319	-0.242372074
28	32.28411761	24.77259976	-0.232669139
29	28.98425515	23.12895086	-0.202016725
30	32.03878551	22.03718585	-0.312171623
31	29.88249616	22.84553375	-0.235487771
32	36.61818544	26.2933772	-0.281958489
33	35.16594606	23.99637979	-0.317624507

Tabel 4.16 Pengujian *silhouette index* cluster 1

	a	b	SI
34	15.72323324	25.43572546	0.381844514
35	18.47290838	29.44226545	0.372571774

36	14.41422484	24.24633486	0.405509125
37	15.97163695	25.64676863	0.377245641
38	13.75960538	25.06820628	0.451113286
39	16.98419332	25.84675719	0.342888812
40	14.85872846	25.72189156	0.422331424
41	13.325829	25.31197116	0.473536497
42	17.10979545	25.59550245	0.331531175

Tabel 4.17 Pengujian *silhouette index* cluster 3

	a	b	SI
43	27.8722773	36.63948627	0.239283076
44	26.93823238	34.14364636	0.211032352
45	23.93451818	33.29301424	0.281094886
46	22.92127664	25.71956713	0.108800062
47	23.40718642	23.87877921	0.019749451
48	24.40348856	42.05557467	0.419732372
49	24.42410748	26.18193825	0.067139062
50	22.06614366	31.53757663	0.300322155
51	22.31175977	32.04791455	0.303799948
52	19.67789642	33.23322709	0.407884875
53	23.21861981	39.1472831	0.406890645
54	22.87579861	30.26685363	0.244196345
55	20.73074956	36.93896191	0.438783645
56	27.91462218	44.05912782	0.366428172
57	26.75359676	47.10196537	0.432006785
58	29.81706975	30.42002891	0.019821124
59	22.53528274	34.98641545	0.355884779
60	27.11484853	43.08241603	0.370628413
61	26.10071364	40.77117257	0.359824307
62	21.5502259	45.03184728	0.521444773
63	22.98007437	44.73446196	0.486300419

64	28.58427122	42.65620504	0.329891836
65	36.6186485	47.72631379	0.232736711
66	33.15295412	56.23783985	0.410486708
67	33.5138959	57.35094199	0.415634779

Dengan menguji dengan *silhouette index*, *cluster 2* menghasilkan 29 hasil minus dari 33 data. Dan tidak terdapat hasil minus pada *cluster 1* dan 3.

Dengan demikian hasil *silhouette index* terbaik terdapat pada *average linkage*. Walaupun dua dari 3 cluster terdapat data dengan hasil minus, namun hasil plus atau tidak minus lebih merata jika dibandingkan dengan *single linkage* dan *complete linkage*. Pada *average linkage* juga memiliki hasil minus paling sedikit dibandingkan dengan *single linkage* dan *complete linkage*.

4.5.5. Perbandingan Hasil

Perbandingan hasil yang dimaksud merupakan perbandingan antara hasil clustering dengan menggunakan sistem dan clustering secara manual. Perbandingan dalam bentuk tabel bisa dilihat dalam lampiran 2.

Dari tabel lampiran 2, *cluster 2* sebagai kelas C terdapat delapan data yang tidak sesuai antara perhitungan secara manual dan perhitungan dengan sistem. *Cluster 1* sebagai kelas B dan *cluster 3* sebagai kelas A memiliki data yang tidak sesuai sebanyak sembilan.

Dalam hal ini perbedaan penempatan data pada cluster yang tidak sesuai antara sistem dan manual dikarenakan menggunakan metode pengelompokan yang berbeda. Pengelompokan secara manual hanya menggunakan jumlah keseluruhan dari nilai yang didapatkan siswa, sedangkan sistem menggunakan *euclidean distance* yang kemudian dikelompokkan dengan menggunakan AHC.

Pada tabel 4.18 merupakan pengujian sistem dengan menggunakan *Black Box Testing*.

Tabel 4.18 *Black Box Testing*

No	Skenario	Test Case	Hasil yang didapat	Status
1	Input data berhasil	Menghasilkan input data dari file .xls	Input berhasil dan menampilkan jumlah data, jumlah <i>feature</i> dan keseluruhan data file .xls pada tabel	Berhasil
2	Proses berhasil	Mengolah data dengan <i>euclidean distance</i> , AHC dan Akurasi SSE	Menampilkan hasil dari AHC (<i>single linkage, average linkage</i> dan <i>complete linkage</i>) dengan <i>output</i> dendrogram, jumlah <i>cluster</i> , label <i>cluster</i> dan SSE	Berhasil
3	Simpan berhasil	Menyimpan data hasil <i>clustering</i>	Simpan berhasil dan data label <i>cluster</i> dari <i>single linkage, average linkage</i> dan <i>complete linkage</i> disimpan dalam satu file .xls	Berhasil

BAB V

PENUTUP

Bab terakhir pada tulisan ini menjelaskan mengenai kesimpulan dari penelitian terkait dengan pengelompokan capaian belajar siswa SD Model Kabupaten Sleman. Pada bab ini ini dijelaskan saran yang diberikan pada sistem yang sudah dibuat.

5.1. Kesimpulan

Pengelompokan capaian belajar siswa SD Model Kabupaten Sleman menggunakan *agglomerative hierarchical clustering* berdasarkan nilai ujian tengah semester dan ujian akhir semester.

Berdasarkan hasil percobaan diatas, kesimpulan yang dapat diambil sebagai berikut :

1. Dalam penelitian ini, metode *agglomerative hierarchical clustering* dapat dengan baik mengelompokkan data nilai capaian belajar siswa. Dalam penelitian ini, sistem mengelompokkan data nilai siswa dengan tiga metode dan mampu menghasilkan *clustering* dengan cukup seimbang yang akan digunakan dalam pengelompokkan kelas.
2. Dari tiga metode *agglomerative hierarchical clustering* yang digunakan, yang mampu menghasilkan dendrogram paling seimbang adalah metode *average linkage*. Metode *average linkage* memiliki obyek pada sateiap *cluster* yang paling merata jika dibandingkan dengan metode *single linkage* dan *complete linkage* dengan berurutan pada *cluster* 1, 2, dan 3 adalah 14, 22, dan 31.
3. Pengujian hasil dengan SSE, yang memiliki nilai terendah adalah metode *complete linkage* dengan nilai 154457 sedangkan pada metode *single linkage* dan *average linkage* menghasilkan nilai SSE 244331 dan 192467.

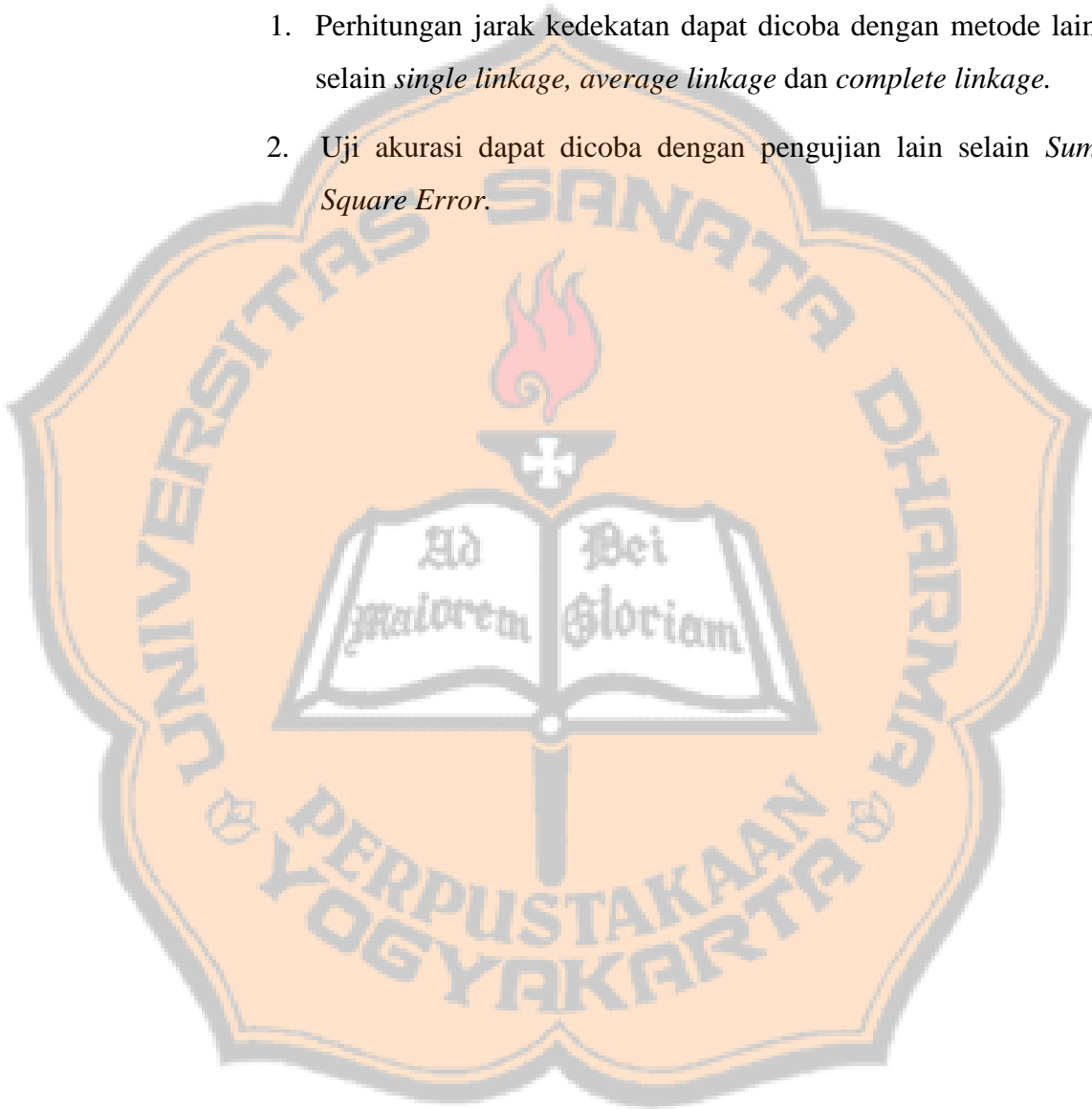
Hasil perhitungan *silhouette index* dari algoritma AHC yang dipilih mendapatkan hasil dalam cluster 1 ada sebanyak 14 data yang tepat berada di cluster 1 dari 14 data, cluster 2 ada sebanyak 10 data yang

tepat berada di cluster 2 dari 22 data, dan cluster 3 ada sebanyak 4 data yang tepat berada di cluster 3 dari 31 data.

5.2. Saran

Saran yang diperlukan untuk pengembangan sistem lebih lanjut sebagai berikut :

1. Perhitungan jarak kedekatan dapat dicoba dengan metode lain selain *single linkage*, *average linkage* dan *complete linkage*.
2. Uji akurasi dapat dicoba dengan pengujian lain selain *Sum Square Error*.



DAFTAR PUSTAKA

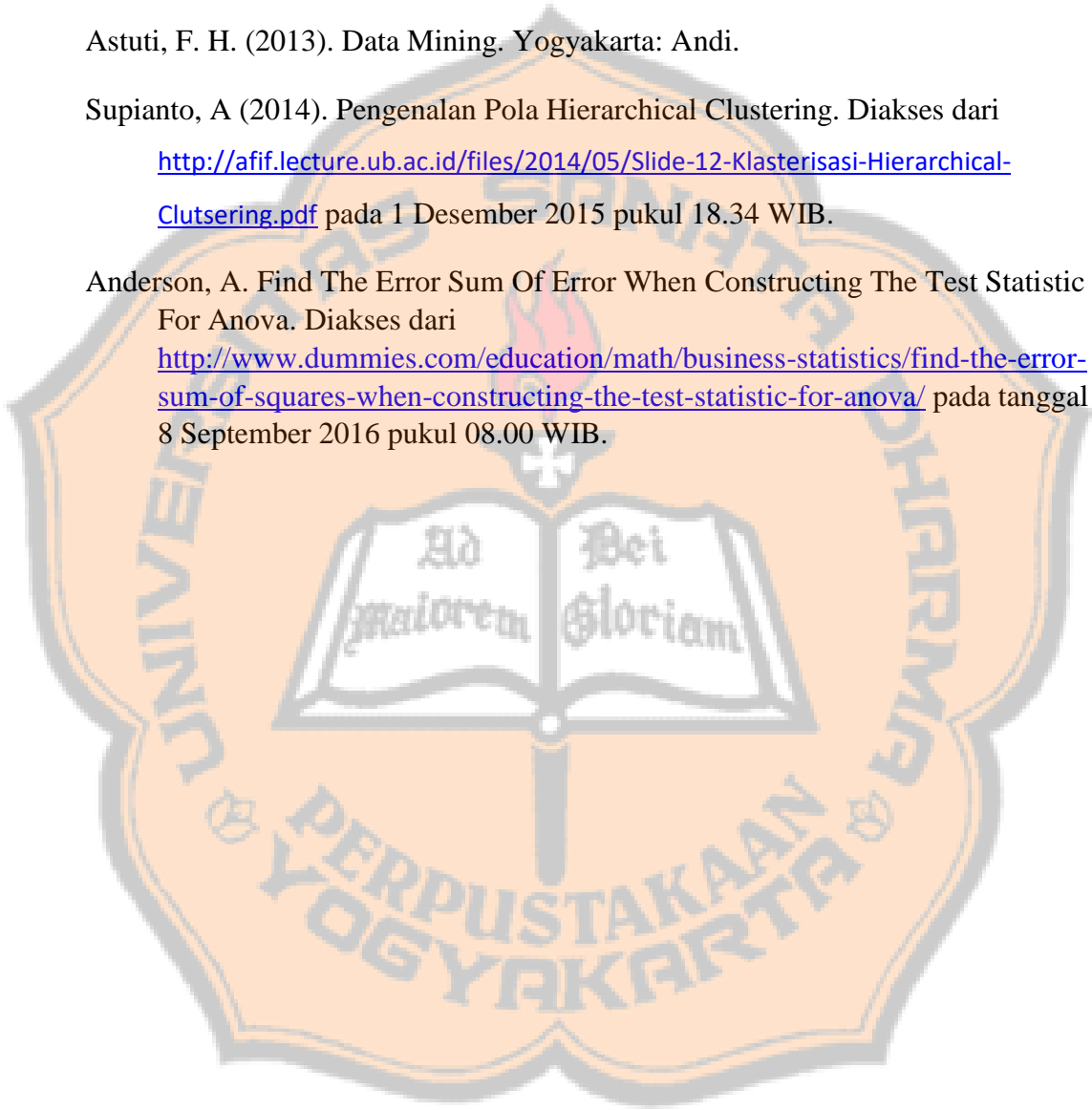
Prasetyo, E (2012). Data Mining : Konsep Dan Aplikasi Menggunakan Matlab.
Yogyakarta: Andi.

Prasetyo, E (2014). Data Mining : Pengolahan Data Menjadi Informasi
Menggunakan Matlab. Yogyakarta: Andi.

Astuti, F. H. (2013). Data Mining. Yogyakarta: Andi.

Supianto, A (2014). Pengenalan Pola Hierarchical Clustering. Diakses dari
<http://afif.lecture.ub.ac.id/files/2014/05/Slide-12-Klasterisasi-Hierarchical-Clustering.pdf> pada 1 Desember 2015 pukul 18.34 WIB.

Anderson, A. Find The Error Sum Of Error When Constructing The Test Statistic
For Anova. Diakses dari
<http://www.dummies.com/education/math/business-statistics/find-the-error-sum-of-squares-when-constructing-the-test-statistic-for-anova/> pada tanggal
8 September 2016 pukul 08.00 WIB.



LAMPIRAN 1

Tabel Percobaan Standar Deviasi *Single Linkage*

NO	NIS	BID	MTK	IPA	BID	MTK	IPA	JUMLAH	RANGK	KELAS	Cluster
1	200	81	70	78	56	79	80	444	45	A	2
2	218	74	40	65	69	42	78	368	68	A	3
3	184	93	100	94	86	100	97	570	1	C	1
4	182	88	89	97	83	100	94	551	2	C	1
5	180	94	75	99	91	92	98	549	3	C	1
6	159	94	87	99	85	85	98	548	4	C	1
7	197	95	72	99	89	85	98	538	5	C	1
8	188	93	90	100	82	80	92	537	6	C	1
9	211	91	71	99	84	90	94	529	7	C	1
10	214	96	69	81	90	96	94	526	8	C	1
11	220	89	80	90	83	95	87	524	9	C	1
12	210	89	93	94	68	89	89	522	10	C	1
13	217	91	63	97	88	80	100	519	11	C	1
14	565	84	82	93	77	100	81	517	12	C	1
15	194	93	73	94	78	76	94	508	13	C	1
16	187	88	88	84	84	75	88	507	14	C	1
17	185	91	70	94	77	89	80	501	15	C	1
18	157	85	66	95	84	75	94	499	16	C	1
19	199	91	65	93	68	94	88	499	17	C	1
20	174	88	62	89	80	89	87	495	18	C	1
21	221	80	72	96	80	78	88	494	19	C	1
22	216	83	84	79	70	92	84	492	20	C	1
23	225	89	58	96	80	82	85	490	21	C	1
24	212	86	68	84	81	71	94	484	22	C	1
25	309	85	59	89	89	71	89	482	23	C	1
26	470	81	70	91	73	85	81	481	24	B	1
27	172	86	60	83	77	96	76	478	25	B	1
28	192	91	68	89	72	75	80	475	26	B	1
29	173	85	62	82	85	75	85	474	27	B	1
30	224	83	62	93	82	68	83	471	28	B	1
31	223	78	67	86	74	85	80	470	29	B	1
32	228	86	55	81	83	80	85	470	30	B	1
33	226	83	60	86	73	83	84	469	31	B	1
34	227	91	60	81	87	58	91	468	32	B	1
35	204	79	60	89	76	75	87	466	33	B	1
36	207	84	54	83	83	91	70	465	34	B	1
37	209	81	72	79	77	83	71	463	35	B	1
38	215	91	49	93	83	55	92	463	36	B	1
39	219	86	57	96	75	61	88	463	37	B	1
40	472	74	66	81	83	88	69	461	38	B	1
41	310	83	62	84	86	71	74	460	39	B	1
42	198	88	47	87	74	76	86	458	40	B	1
43	179	85	38	89	83	68	91	454	41	B	1
44	167	86	52	79	81	74	81	453	42	B	1
45	163	86	54	88	83	55	84	450	43	B	1
46	158	85	56	83	78	63	83	448	44	B	1
47	162	83	47	87	74	72	80	443	46	A	1
48	170	86	45	79	76	74	79	439	47	A	1
49	201	86	40	92	81	52	88	439	48	A	1
50	169	81	54	89	65	65	81	435	49	A	1
51	177	84	53	90	73	57	75	432	51	A	1
52	203	83	40	87	76	68	78	432	50	A	1
53	161	78	45	85	80	61	82	431	52	A	1
54	570	81	35	91	71	61	87	426	53	A	1
55	181	81	49	86	74	65	70	425	54	A	1
56	213	86	48	86	71	53	80	424	55	A	1
57	202	69	46	79	84	53	92	423	56	A	1
58	164	77	54	71	74	72	72	420	58	A	1
59	190	88	35	87	69	50	91	420	57	A	1
60	475	80	49	77	76	61	71	414	59	A	1
61	165	75	37	92	60	58	84	406	60	A	1
62	196	66	46	79	69	58	88	406	61	A	1
63	189	79	34	83	68	55	85	404	63	A	1
64	232	79	37	84	73	51	80	404	62	A	1
65	168	82	47	73	78	58	63	401	64	A	1
66	156	83	52	60	73	58	63	389	66	A	1
67	186	63	38	89	71	41	85	387	67	A	1

LAMPIRAN 2

Tabel Percobaan Standar Deviasi *Average Linkage*

NO	NIS	BID	MTK	IPA	BID	MTK	IPA	JUMLAH	RANGK	KELAS	Cluster
1	184	93	100	94	86	100	97	570	1	C	2
2	182	88	89	97	83	100	94	551	2	C	2
3	180	94	75	99	91	92	98	549	3	C	2
4	159	94	87	99	85	85	98	548	4	C	2
5	197	95	72	99	89	85	98	538	5	C	2
6	188	93	90	100	82	80	92	537	6	C	2
7	211	91	71	99	84	90	94	529	7	C	2
8	214	96	69	81	90	96	94	526	8	C	2
9	220	89	80	90	83	95	87	524	9	C	2
10	210	89	93	94	68	89	89	522	10	C	2
11	217	91	63	97	88	80	100	519	11	C	2
12	565	84	82	93	77	100	81	517	12	C	2
13	187	88	88	84	84	75	88	507	14	C	2
14	216	83	84	79	70	92	84	492	20	C	2
15	172	86	60	83	77	96	76	478	25	B	2
16	173	85	62	82	85	75	85	474	27	B	2
17	224	83	62	93	82	68	83	471	28	B	2
18	207	84	54	83	83	91	70	465	34	B	2
19	209	81	72	79	77	83	71	463	35	B	2
20	472	74	66	81	83	88	69	461	38	B	2
21	310	83	62	84	86	71	74	460	39	B	2
22	200	81	70	78	56	79	80	444	45	A	2
23	194	93	73	94	78	76	94	508	13	C	1
24	185	91	70	94	77	89	80	501	15	C	1
25	157	85	66	95	84	75	94	499	16	C	1
26	199	91	65	93	68	94	88	499	17	C	1
27	174	88	62	89	80	89	87	495	18	C	1
28	221	80	72	96	80	78	88	494	19	C	1
29	225	89	58	96	80	82	85	490	21	C	1
30	212	86	68	84	81	71	94	484	22	C	1
31	309	85	59	89	89	71	89	482	23	C	1
32	470	81	70	91	73	85	81	481	24	B	1
33	197	91	68	89	72	75	80	475	26	B	1
34	223	78	67	86	74	85	80	470	29	B	1
35	226	83	60	86	73	83	84	469	31	B	1
36	204	79	60	89	76	75	87	466	33	B	1
37	228	86	65	81	83	80	85	470	30	B	3
38	227	91	60	81	87	58	91	468	32	B	3
39	215	91	69	93	83	55	92	461	36	B	3
40	219	86	57	96	75	61	88	463	37	B	3
41	198	88	47	87	74	76	86	458	40	B	3
42	179	85	38	83	83	68	91	454	41	B	3
43	167	86	52	79	81	74	81	453	42	B	3
44	163	86	54	88	83	55	84	450	43	B	3
45	158	85	56	83	78	63	83	448	44	B	3
46	162	83	47	87	74	72	80	443	46	A	3
47	170	86	45	79	76	74	79	439	47	A	3
48	201	86	40	92	81	52	88	439	48	A	3
49	169	81	54	89	65	65	81	435	49	A	3
50	177	84	53	90	73	57	75	432	51	A	3
51	203	83	40	87	76	68	78	432	50	A	3
52	161	78	45	85	80	61	82	431	52	A	3
53	570	81	35	91	71	61	87	426	53	A	3
54	181	81	49	86	74	65	70	425	54	A	3
55	213	86	48	86	71	53	80	424	55	A	3
56	202	69	46	79	84	53	92	423	56	A	3
57	164	77	54	71	74	72	72	420	58	A	3
58	190	88	35	87	69	50	91	420	57	A	3
59	475	80	49	77	76	61	71	414	59	A	3
60	165	75	37	92	60	58	84	406	60	A	3
61	196	66	46	79	69	58	88	406	61	A	3
62	189	79	34	83	68	55	85	404	63	A	3
63	232	79	37	84	73	51	80	404	62	A	3
64	168	82	47	73	78	58	63	401	64	A	3
65	156	83	52	60	73	58	63	389	66	A	3
66	186	63	38	89	71	41	85	387	67	A	3
67	218	74	40	65	69	42	78	368	68	A	3

LAMPIRAN 3

Tabel Percobaan Standar Deviasi *Complete Linkage*

NO	NIS	BID	MTK	IPA	BID	MTK	IPA	JUMLAH	RANGK	KELAS	Cluster
1	184	93	100	94	86	100	97	570	1	C	2
2	182	88	89	97	83	100	94	551	2	C	2
3	180	94	75	99	91	92	98	549	3	C	2
4	159	94	87	99	85	85	98	548	4	C	2
5	197	95	72	99	89	85	98	538	5	C	2
6	188	93	90	100	82	80	92	537	6	C	2
7	211	91	71	99	84	90	94	529	7	C	2
8	214	96	69	81	90	96	94	526	8	C	2
9	220	89	80	90	83	95	87	524	9	C	2
10	210	89	93	94	68	89	89	522	10	C	2
11	217	91	63	97	88	80	100	519	11	C	2
12	565	84	82	93	77	100	81	517	12	C	2
13	194	93	73	94	78	76	94	508	13	C	2
14	187	88	88	84	84	75	88	507	14	C	2
15	157	85	66	95	84	75	94	499	16	C	2
16	221	80	72	96	80	78	88	494	19	C	2
17	216	83	84	79	70	92	84	492	20	C	2
18	212	86	68	84	81	71	94	484	22	C	2
19	309	85	59	89	89	71	89	482	23	C	2
20	172	86	60	83	77	96	76	478	25	B	2
21	173	85	62	82	85	75	85	474	27	B	2
22	224	83	62	93	82	68	83	471	28	B	2
23	228	86	55	81	83	80	85	470	30	B	2
24	227	91	60	81	87	58	91	468	32	B	2
25	207	84	54	83	83	91	70	465	34	B	2
26	209	81	72	79	77	83	71	463	35	B	2
27	219	86	57	94	75	61	88	463	37	B	2
28	472	74	66	81	83	88	69	461	38	B	2
29	310	83	62	84	86	71	74	460	39	B	2
30	198	88	47	87	74	76	86	458	40	B	2
31	167	86	52	79	81	74	81	453	42	B	2
32	200	81	70	78	56	79	89	444	45	A	2
33	170	86	46	79	76	74	79	439	47	A	2
34	185	94	70	94	79	69	80	401	15	C	1
35	199	91	65	93	68	94	88	499	17	C	1
36	174	88	62	89	80	82	87	495	18	C	1
37	225	80	58	94	80	82	85	490	21	C	1
38	470	81	70	91	73	85	81	481	24	B	1
39	192	91	68	89	72	75	80	475	26	B	1
40	223	78	67	84	74	85	80	470	29	B	1
41	226	83	60	84	73	83	84	469	31	B	1
42	204	79	60	89	76	75	87	466	33	B	1
43	215	91	49	93	83	55	92	463	36	B	3
44	179	85	38	89	83	68	91	424	41	B	3
45	163	86	54	88	83	55	84	420	43	B	3
46	198	85	56	83	78	63	83	418	44	B	3
47	162	83	47	87	74	72	80	443	46	A	3
48	201	86	40	92	81	52	88	439	48	A	3
49	169	81	54	89	65	65	81	435	49	A	3
50	177	84	53	90	73	57	75	432	51	A	3
51	203	83	40	87	76	68	78	432	50	A	3
52	161	78	45	85	80	61	82	431	52	A	3
53	570	81	35	91	71	61	87	426	53	A	3
54	181	81	49	86	74	65	70	425	54	A	3
55	213	86	48	86	71	53	80	424	55	A	3
56	202	69	46	79	84	53	92	423	56	A	3
57	164	77	54	71	74	72	72	420	58	A	3
58	190	88	35	87	69	50	91	420	57	A	3
59	475	80	49	77	76	61	71	414	59	A	3
60	165	75	37	92	60	58	84	406	60	A	3
61	196	66	46	79	69	58	88	406	61	A	3
62	189	79	34	83	68	55	85	404	63	A	3
63	232	79	37	84	73	51	80	404	62	A	3
64	168	82	47	73	78	58	63	401	64	A	3
65	156	83	52	60	73	58	63	389	66	A	3
66	186	63	38	89	71	41	85	387	67	A	3
67	218	74	40	65	69	42	78	368	68	A	3

LAMPIRAN 4

Source Code

1. AHCMMain.fig



2. AHCMMain.m

```
function varargout = AHCMMain(varargin)
% AHCMMAIN MATLAB code for AHCMMain.fig
% AHCMMAIN, by itself, creates a new AHCMMAIN or raises the
existing
% singleton*.
%
%
%
% H = AHCMMAIN returns the handle to a new AHCMMAIN or the
handle to
% the existing singleton*.
%
% AHCMMAIN('CALLBACK',hObject,eventData,handles,...) calls the
local
% function named CALLBACK in AHCMMAIN.M with the given input
arguments.
%
% AHCMMAIN('Property','Value',...) creates a new AHCMMAIN or
raises the
% existing singleton*. Starting from the left, property
value pairs are
% applied to the GUI before AHCMMain_OpeningFcn gets called.
An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to AHCMMain_OpeningFcn via
varargin.
```

```

%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help AHCMMain

% Last Modified by GUIDE v2.5 31-Oct-2016 14:35:20

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @AHCMMain_OpeningFcn, ...
                  'gui_OutputFcn',  @AHCMMain_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before AHCMMain is made visible.
function AHCMMain_OpeningFcn(hObject,...
    eventdata,...
    handles,...
    varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to AHCMMain (see VARARGIN)

axes(handles.LogoUSDAxes)
matlabImage = imread('D:\AHC\USD.png');
image(matlabImage)
axis off
axis image

% Choose default command line output for AHCMMain
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes AHCMMain wait for user response (see UIRESUME)

```

```

% uiwait(handles.figure1);
set(gcf, 'units','normalized','outerposition',[0 1 1 1]);

% --- Outputs from this function are returned to the command line.
function varargout = AHCMMain_OutputFcn(hObject,...
    eventdata,...
    handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in InputBT.
function InputBT_Callback(hObject, eventdata, handles)
% hObject     handle to InputBT (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

[namaFile, alamatFile] = uigetfile({'*.xls'; '*.xlsx'}, 'Pilih file
Anda',...
    'Data\DataAwal\');

if namaFile ~=0
alamatNamaFile=fullfile(alamatFile,namaFile);
set(handles.InputTF, 'String', alamatNamaFile);
set(handles.InputTF, 'String', namaFile);

% data = xlsread(alamatNamaFile);
[dataAngka, dataText]=xlsread(alamatNamaFile);
handles.dataAngka=dataAngka;
featureAwal=dataText(1,2:end);
siswaAwal =dataAngka(:,1);
dataAwal = dataAngka(:,2:end);

set(handles.InfoJumlahSiswaAwalTF, 'String', size(siswaAwal, 1));

set(handles.JumlahFeatureAwalTF, 'String', size(featureAwal, 2));
    %Tampil di tabel data awal
set(handles.TabelDataAwal, 'ColumnName', featureAwal);
set(handles.TabelDataAwal, 'RowName', siswaAwal);
set(handles.TabelDataAwal, 'Data', dataAwal);

```

```

set(handles.TabelDataAwal, 'ColumnWidth', {77});
set(handles.TabelDataAwal, 'Visible', 'on');

    %Pembentukan atribut dan nilai/handles structure
handles.featureAwal=featureAwal;
handles.siswaAwal=siswaAwal;
handles.dataAwal=dataAwal;

% komponenPeoses_aktif(hObject, handles);

end

    guidata(hObject,handles);
% --- Executes on button press in ProsesBT.
function ProsesBT_Callback(hObject, eventdata, handles)
% hObject    handle to ProsesBT (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Clear cmd window
clc

%Pembentukan penggunaan function handle FunctionAHC
func = FunctionAHC;

% jmlCluster=3;

jmlCluster=str2double(get(handles.jmlClusterTF, 'String'));

[data_Hasil,nilaiSSE]=completeLink(handles, func, jmlCluster);
handles.hasilComplete=data_Hasil;
handles.ClusterSSEComplete=nilaiSSE;

[data_Hasil,nilaiSSE]=singleLink(handles, func, jmlCluster);
handles.hasilSingle=data_Hasil;
handles.ClusterSSESingle=nilaiSSE;

[data_Hasil,nilaiSSE]=averageLink(handles, func, jmlCluster);
handles.hasilAverage=data_Hasil;
handles.ClusterSSEAverage=nilaiSSE;

function[data_Hasil,nilaiSSE]=completeLink(handles, func, jmlCluster
)
    data_Hasil=func.ahc(handles.dataAwal, 'complete');

% jmlCluster=3;
axes(handles.DendroAxes);

labelDendro=func.tampilDendrogram(data_Hasil, jmlCluster);

```

```

[leafNode]=func.leafNodeDendro(data_Hasil,jmlCluster,...
labelDendro,handles.siswaAwal);
set(handles.CompleteTabel,'ColumnName',{'Cluster','Siswa'});

set(handles.CompleteTabel,'Data',leafNode(:,2:3));
set(handles.CompleteTabel,'ColumnWidth',{20 30});

[jmlLabelCluster]=func.tampilCluster(data_Hasil,jmlCluster);
set(handles.TabelClusterComplete,'ColumnName',{'Cluster','Jumlah'});
);
set(handles.TabelClusterComplete,'Data',jmlLabelCluster);
set(handles.TabelClusterComplete,'ColumnWidth',{50 50});

[nilaiSSE]=func.hitungSSE(data_Hasil);

set(handles.TFSSE,'String',nilaiSSE);

function[data_Hasil,nilaiSSE]=singleLink(handles,func,jmlCluster)
data_Hasil=func.ahc(handles.dataAwal,'single');

axes(handles.DendroAxesSingle);

labelDendro=func.tampilDendrogram(data_Hasil,jmlCluster);

[leafNode]=func.leafNodeDendro(data_Hasil,jmlCluster,...
labelDendro,handles.siswaAwal);
set(handles.SingleTabel,'ColumnName',{'Cluster','Siswa'});

set(handles.SingleTabel,'Data',leafNode(:,2:3));
set(handles.SingleTabel,'ColumnWidth',{20 30});

[jmlLabelCluster]=func.tampilCluster(data_Hasil,jmlCluster);
set(handles.TabelClusterSingle,'ColumnName',{'Cluster','Jumlah'});
set(handles.TabelClusterSingle,'Data',jmlLabelCluster);
set(handles.TabelClusterSingle,'ColumnWidth',{50 50});

[nilaiSSE]=func.hitungSSE(data_Hasil);

set(handles.TFSSESingle,'String',nilaiSSE);

```

```

function [data_Hasil, nilaiSSE]=averageLink(handles, func, jmlCluster)
    data_Hasil=func.ahc(handles.dataAwal, 'average');

    axes(handles.DendroAxesAverage);

labelDendro=func.tampilDendrogram(data_Hasil, jmlCluster);

[leafNode]=func.leafNodeDendro(data_Hasil, jmlCluster, ...
labelDendro, handles.siswaAwal);
set(handles.AverageTabel, 'ColumnName', {'Cluster', 'Siswa'});

set(handles.AverageTabel, 'Data', leafNode(:, 2:3));
set(handles.AverageTabel, 'ColumnWidth', {20 30});

[jmlLabelCluster]=func.tampilCluster(data_Hasil, jmlCluster);
set(handles.TabelClusterAverage, 'ColumnName', {'Cluster', 'Jumlah'})
;
set(handles.TabelClusterAverage, 'Data', jmlLabelCluster);
set(handles.TabelClusterAverage, 'ColumnWidth', {50 50});

[nilaiSSE]=func.hitungSSE(data_Hasil);

    set(handles.TFSSEAverage, 'String', nilaiSSE);

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit3 as text
%        str2double(get(hObject, 'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))

```

```

    set(hObject, 'BackgroundColor', 'white');
end

function InfoJumlahSiswaAwalTF_Callback(hObject, eventdata,
handles)
% hObject    handle to InfoJumlahSiswaAwalTF (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of
InfoJumlahSiswaAwalTF as text
%         str2double(get(hObject, 'String')) returns contents of
InfoJumlahSiswaAwalTF as a double

% --- Executes during object creation, after setting all
properties.
function InfoJumlahSiswaAwalTF_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to InfoJumlahSiswaAwalTF (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function InputTF_Callback(hObject, eventdata, handles)
% hObject    handle to InputTF (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of InputTF as text
%         str2double(get(hObject, 'String')) returns contents of
InputTF as a double

% --- Executes during object creation, after setting all
properties.
function InputTF_CreateFcn(hObject, eventdata, handles)
% hObject    handle to InputTF (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB

```

```

% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function JumlahFeatureAwalTF_Callback(hObject, eventdata, handles)
% hObject    handle to JumlahFeatureAwalTF (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
JumlahFeatureAwalTF as text
%       str2double(get(hObject,'String')) returns contents of
JumlahFeatureAwalTF as a double

% --- Executes during object creation, after setting all
properties.
function JumlahFeatureAwalTF_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to JumlahFeatureAwalTF (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function jmlClusterTF_Callback(hObject, eventdata, handles)
% hObject    handle to jmlClusterTF (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of jmlClusterTF as
text
%       str2double(get(hObject,'String')) returns contents of
jmlClusterTF as a double

```



```

% --- Executes during object creation, after setting all
properties.
function jmlClusterTF_CreateFcn(hObject, eventdata, handles)
% hObject    handle to jmlClusterTF (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function TFSSE_Callback(hObject, eventdata, handles)
% hObject    handle to TFSSE (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TFSSE as text
%       str2double(get(hObject,'String')) returns contents of
TFSSE as a double

% --- Executes during object creation, after setting all
properties.
function TFSSE_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TFSSE (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function TFSSESingle_Callback(hObject, eventdata, handles)
% hObject    handle to TFSSESingle (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TFSSESingle as
text

```

```

%         str2double(get(hObject,'String')) returns contents of
TFSSESingle as a double

% --- Executes during object creation, after setting all
properties.
function TFSSESingle_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TFSSESingle (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function TFSSEAverage_Callback(hObject, eventdata, handles)
% hObject    handle to TFSSEAverage (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TFSSEAverage as
text
%         str2double(get(hObject,'String')) returns contents of
TFSSEAverage as a double

% --- Executes during object creation, after setting all
properties.
function TFSSEAverage_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TFSSEAverage (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes when entered data in editable cell(s) in
CompleteTabel.
function CompleteTabel_CellEditCallback(hObject, eventdata,
handles)
% hObject    handle to CompleteTabel (see GCBO)

```

```

% eventdata structure with the following fields (see
MATLAB.UI.CONTROL.TABLE)
% Indices: row and column indices of the cell(s) edited
% PreviousData: previous data for the cell(s) edited
% EditData: string(s) entered by the user
% NewData: EditData or its converted form set on the Data
property. Empty if Data was not changed
% Error: error string when failed to convert EditData to
appropriate value for Data
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in ResetBT.

% --- Executes on button press in pushbutton5.

% --- Executes on button press in SimpanBT.
function SimpanBT_Callback(hObject, eventdata, handles)
% hObject handle to SimpanBT (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

warning('off','MATLAB:xlswrite:AddSheet');

[namaFile, alamatFile] = uiputfile({'*.xls'; '*.xlsx'},
'Simpan',...
'Data\');

if isequal(namaFile,0) || isequal(alamatFile,0)
    uiwait(msgbox('Anda membatalkan untuk simoan.', 'modal'));
else
    alamatNamaFile=fullfile(alamatFile,namaFile);
    dataTabel=get(handles.CompleteTabel, 'Data');
    featureTabel=get(handles.CompleteTabel, 'ColumnName');

    dataTabelDitulis=mat2cell(dataTabel,ones(size(dataTabel,1),1),ones
(size(dataTabel,2),1));

    siswaBaruDanDataIsi=dataTabelDitulis;

    featureTabelBaruComplete=featureTabel';
    dataDitulisComplete=[featureTabelBaruComplete;siswaBaruDanDataIsi]
    ;
    xlswrite(alamatNamaFile,dataDitulisComplete,1);

    dataTabel=get(handles.AverageTabel, 'Data');

```

```

featureTabel=get(handles.AverageTabel,'ColumnName');

%

dataTabelDitulis=mat2cell(dataTabel,ones(size(dataTabel,1),1),ones(
(size(dataTabel,2),1)));

siswaBaruDanDataIsi=dataTabelDitulis;

featureTabelBaruAverage=featureTabel';
dataDitulisAverage=[featureTabelBaruAverage;siswaBaruDanDataIsi];
xlswrite(alamatNamaFile,dataDitulisAverage,2);

dataTabel=get(handles.SingleTabel,'Data');
featureTabel=get(handles.SingleTabel,'ColumnName');

%

dataTabelDitulis=mat2cell(dataTabel,ones(size(dataTabel,1),1),ones(
(size(dataTabel,2),1)));

siswaBaruDanDataIsi=dataTabelDitulis;

featureTabelBaruSingle=featureTabel';
dataDitulisSingle=[featureTabelBaruSingle;siswaBaruDanDataIsi];
xlswrite(alamatNamaFile,dataDitulisSingle,3);

act = actxserver('Excel.Application');
ewb=act.Workbooks.Open(alamatNamaFile);

ewb.Worksheets.Item(3).Name='Hasil Single';
%
ewb.Worksheets.Item(2).Name='Hasil Average';
%
ewb.Worksheets.Item(1).Name='Hasil Complete';

ewb.Save
ewb.Close(false)
act.Quit
act.delete

end

% --- Executes on button press in ZoomInBT.
function ZoomInBT_Callback(hObject, eventdata, handles)
% hObject handle to ZoomInBT (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

inValue=get(handles.ZoomInBT,'Value');
hZoomIn=zoom;

if inValue==1

hZoomIn.Enable='on';
hZoomIn.Direction='in';

hZoomIn.Enable='off';
set(handles.ZoomInBT,'Value',0);
end

% --- Executes on button press in ZoomOutBT.
function ZoomOutBT_Callback(hObject, eventdata, handles)
% hObject    handle to ZoomOutBT (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
inValue=get(handles.ZoomOutBT,'Value');
hZoomIn=zoom;

if inValue==1

hZoomIn.Enable='on';
hZoomIn.Direction='out';

else
hZoomIn.Enable='off';
set(handles.ZoomOutBT,'Value',0);
end

```

3. FunctionAHC.m

```

function func = FunctionAHC

func.ahc=@ahc;
func.tampilDendrogram=@tampilDendrogram;
func.leafNodeDendro=@leafNodeDendro;
func.hitungSSE=@hitungSSE;
func.tampilCluster=@tampilCluster;
end
%-----AHC
function [dataHasil] = ahc(data,linkage)
%
%Perhitungan jarak dengan Euclidean
dataEuclidian = pdist(data,'euclidean');
%
%Set data matriks dari perhitungan jarak
kolom= size(dataEuclidian,2);
baris= (1+sqrt(1+8*kolom))/2;

dataHasil=zeros(baris-1,3);

```

```

jmlPoinDiSetiapCluster = zeros(1,2*baris-1);
jmlPoinDiSetiapCluster(1:baris) = 1;
kolom = baris;
vektorIndeks = 1:kolom;

loop=1;
if strcmp(linkage, 'complete')
while loop ~= kolom
[v,k] = min(dataEuclidian);
% Mencari dataHasil
[hasilData,i,j,I,J]=olahDataHasil(loop,baris,v,k,vektorIndeks);
dataHasil(loop,:)=hasilData(loop,:);
dataEuclidian(I) = max(dataEuclidian(I),dataEuclidian(J));
[dataEuclidian,~,baris,jmlPoinDiSetiapCluster,vektorIndeks]=update
Matriks(dataEuclidian,J,baris,jmlPoinDiSetiapCluster,kolom,loop,ve
ktorIndeks,i,j);
loop=loop+1;
end

elseif strcmp(linkage, 'single')
while loop ~= kolom
[v,k] = min(dataEuclidian);
% Mencari dataHasil
[hasilData,i,j,I,J]=olahDataHasil(loop,baris,v,k,vektorIndeks);
dataHasil(loop,:)=hasilData(loop,:);
dataEuclidian(I) = min(dataEuclidian(I),dataEuclidian(J));
[dataEuclidian,~,baris,jmlPoinDiSetiapCluster,vektorIndeks]=update
Matriks(dataEuclidian,J,baris,jmlPoinDiSetiapCluster,kolom,loop,ve
ktorIndeks,i,j);
loop=loop+1;
end

elseif strcmp(linkage, 'average')
while loop ~= kolom
[v,k] =
average(dataEuclidian,baris,jmlPoinDiSetiapCluster,vektorIndeks);
% Mencari dataHasil
[hasilData,i,j,I,J]=olahDataHasil(loop,baris,v,k,vektorIndeks);
dataHasil(loop,:)=hasilData(loop,:);
dataEuclidian(I) = dataEuclidian(I) + dataEuclidian(J);
[dataEuclidian,~,baris,jmlPoinDiSetiapCluster,vektorIndeks]=update
Matriks(dataEuclidian,J,baris,jmlPoinDiSetiapCluster,kolom,loop,ve
ktorIndeks,i,j);
loop=loop+1;
end

else
end

end
function
[v,k]=average(dataEuclidian,baris,jmlPoinDiSetiapCluster,vektorInd
eks)
p = (baris-1):-1:2;
I = zeros(baris*(baris-1)/2,1);

```

```

I(cumsum([1 p])) = 1;
I = cumsum(I);
J = ones(baris*(baris-1)/2,1);
J(cumsum(p)+1) = 2-p;
J(1)=2;
J = cumsum(J);
W
=jmlPoinDiSetiapCluster(vektorIndeks(I)).*jmlPoinDiSetiapCluster(v
ektorIndeks(J));
[v, k]=min(dataEuclidian./W);
end

% Pencarian data hasil
function
[dataHasil,i,j,I,J]=olahDataHasil(loop,baris,v,k,vektorIndeks)
i = floor(baris+1/2-sqrt(baris^2-baris+1/4-2*(k-1)));
j = k - (i-1)*(baris-i/2)+i;
% Memasukkan nilai ke data hasil
dataHasil(loop,:)= [vektorIndeks(i) vektorIndeks(j) v];
% Penyimpanan sementara
I1 = 1:(i-1); I2 = (i+1):(j-1); I3 = (j+1):baris;
I = [I1.*(baris-(I1+1)/2)-baris+i i*(baris-(i+1)/2)-baris+I2
i*(baris-(i+1)/2)-baris+I3];
J = [I1.*(baris-(I1+1)/2)-baris+j I2.*(baris-(I2+1)/2)-baris+j
j*(baris-(j+1)/2)-baris+I3];
end
% Update matriks
function [dataEuclidian,...
J,baris,...
jmlPoinDiSetiapCluster,...
vektorIndeks]=updateMatriks(dataEuclidian,...
J,baris,...
jmlPoinDiSetiapCluster,...
kolom,loop,vektorIndeks,i,j)
J = [J i*(baris-(i+1)/2)-baris+j];
dataEuclidian(J) = [];
% update baris, jmlPoinDiSetiapCluster, vektorIndeks
baris = baris-1;
jmlPoinDiSetiapCluster(kolom+loop)
=jmlPoinDiSetiapCluster(vektorIndeks(i))
+jmlPoinDiSetiapCluster(vektorIndeks(j));
vektorIndeks(i) = kolom+loop;
vektorIndeks(j:(kolom-1))=vektorIndeks((j+1):kolom);
end

%-----Tampil Dendrogram
function[dataIndeks]=tampilDendrogram(dataAHC,jmlCluster)
warna = dataAHC(end-jmlCluster+2,3)-eps;
[dendro,~,dataIndeks]=
dendrogram(dataAHC,0,'colorthreshold',warna);
if size(dataIndeks,2)>100
set(dendro, 'LineWidth',1)
else
set(dendro, 'LineWidth',2)
end
set(gca, 'XTickLabel',[], 'TickLength',[0 0])
set(gca, 'YColor',[0.4,0.4,0.4]);
% set(gca, 'YTickLabel',[])

```

```

% Pengaturan YLim
yLim=get(gca, 'YLim');
if yLim(2)>100
yLim(2)=yLim(2)+100;
else
yLim(2)=yLim(2)+3;
end
set(gca, 'YLim', yLim)
end

%-----
LeafNode Dendro
function [leafNode]=leafNodeDendro(dataHasil, jmlCluster,
labelDendro, anggota)
    pengelompokanCluster = cluster(dataHasil, 'maxclust',
jmlCluster);

%Indeks /Cluster
noCluster=zeros(size(pengelompokanCluster));
kelompokCluster=1;
indeksPerCluster=1;
for i=1:size(pengelompokanCluster,1)
    for
j=1:length(find(pengelompokanCluster==kelompokCluster))
        noCluster(indeksPerCluster)=j;
        indeksPerCluster=indeksPerCluster+1;
    end
    kelompokCluster=kelompokCluster+1;
end

%leafNode
label=zeros(size(labelDendro));
for i=1:size(labelDendro,2);
    label(i)=anggota(labelDendro(i));
end
cls=sort(pengelompokanCluster);
leafNode=[noCluster(1:size(label'), :)
cls(1:size(label'), :) label'];
end

%-----Cluster
function[jmlLabelCluster]=tampilCluster(dataHasil,jmlCluster)

jmlCls=jmlCluster;
cls=cluster(dataHasil, 'maxclust', jmlCls);

jmlLabelCluster=zeros(jmlCls,2);
for i=1:jmlCls
    jmlLabelCluster(i,1)=i;
    jmlLabelCluster(i,2)=length(find(cls==i));
end

end

%-----SSE

```



```
function [nilaiSSE]=hitungSSE(data)
rerata=mean(data,2);
[m,n]=size(data);
nilaiSSETemp=zeros(m,n);
for i=1:m
    for j=1:n
        nilaiSSETemp1(i,j)=power(((data(i,j)-rerata(i,1))),2);
    end
end
nilaiSSETemp2=sum(nilaiSSETemp1,2);
nilaiSSE=double(sum(nilaiSSETemp2));

end
```





