IMPLEMENTASI METODE HIERARCHICAL CLUSTERING PADA OZONE LEVEL DETECTION DATA SET

SKRIPSI

Untuk memenuhi sebagian persyaratan
untuk mencapai gelar Sarjana Komputer

Disusun Oleh:
MUHAMMAD YAMIN
0610963032

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
MALANG
2013

LEMBAR PERSETUJUAN

IMPLEMENTASI METODE HIERARCHICAL CLUSTERING PADA OZONE LEVEL DETECTION DATASET

SKRIPSI
Untuk memenuhi sebagian persyaratan untuk
mencapai gelar Sarjana Komputer

Disusun Oleh:
MUHAMMAD YAMIN
0610963032

Telah diperiksa dan disetujui oleh

Dosen Pembimbing

Pembimbing I	Pembimbing II
Indriati, ST., M.Kom	**Candra Dewi, S.Kom., MSc**
NIP. 83101306120035	**NIP. 19771114 200312 2 001**

LEMBAR PENGESAHAN

IIMPLEMENTASI METODE HIERARCHICAL CLUSTERING PADA OZONE LEVEL DETECTION DATASET

Disusun oleh :

MUHAMMAD YAMIN

NIM. 0610963032

Skripsi ini telah diuji dan dinyatakan lulus tanggal 31Agustus 2013

Penguji I	Penguji II
Drs. Achmad Ridok, M.Kom	**Budi Darma Setiawan, S.Kom., M.Cs**
NIP. 19680825 199403 1 002	**NIK. 84101506110090**

Penguji III

Ahmad Afif Supianto, S.Si., M.Kom
NIK. 820623 16 1 1 0425

Mengetahui
Ketua Program Studi Teknik Informatika

Drs. Marji, MT
NIP. 196708011992031001

PERNYATAAN

ORISINALITAS SKRIPSI

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

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

Malang, Juli 2013
Mahasiswa,

Muhammad Yamin
NIM. 0610963032

ABSTRAK

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*. Salah satu algoritma *clustering* adalah *Qualitative Hierarchical Clustering*. Algoritma tersebut adalah metode analisis *cluster* yang membangun hirarki dari *cluster*. Jarak antara tiap obyek merupakan *input* dari algoritma ini. Iterasi terus berlanjut sampai semua obyek telah di*cluster* menjadi sebuah *cluster* saja. Dengan menggunakan sebuah metode *Hierarchical Clustering* untuk mengukur jarak antara data *time series* pada *ozone level detection dataset* yang telah ditentukan. Algoritma *Hierarchical* dimulai dengan menjadikan tiap objek menjadi sebuah *cluster* dan secara iterasi menggabungkan tiap *cluster* yang mirip. Jarak antar objek merupakan input dari algoritma ini. Iterasi terus berlanjut sampai semua objek telah di *cluster* hingga menjadi *cluster* saja. Sistem perangkat lunak yang dibangun dengan algoritma pada *Data time series, Hierarchical Clustering,* dan metode *Qualitative* dapat melakukan proses *clustering* pada *ozone level detection data set*. Tingkat akurasi dari metode *hierarchical clustering* dapat dilihat dari nilai rata-rata akurasi *f-measure* yang di dapat. Akurasi dari masing-masing percobaan yang memiliki nilai *f-measure* paling tinggi terdapat pada skenario ke dua yaitu 0.9792.

Kata Kunci: *Data Mining, Clustering,Hierarchical Clustering,Time Series, Short Time Series.*

ABSTRACT

Clustering is the process of grouping or classification of objects based on information obtained from data describing a relationship between objects with the principle of maximizing the similarity between members of the class and minimize the similarity between classes / clusters. One is Qualitative Hierarchical clustering algorithm Clustering. The algorithm is a method of cluster analysis to build a hierarchy of clusters. The distance between each object is the input of the algorithm. Iteration continues until all objects have been clustered into a cluster only. By using a hierarchical clustering method to measure the distance between the time series data on ozone level detection dataset which has been determined. Hierarchical algorithm begins by making each object into a cluster and each cluster iteratively combines the similar. Distance between objects is the input of the algorithm. Iteration continues until all objects have been in the cluster to be the cluster only. Software system built by the algorithm on time series data, Hierarchical Clustering, and Qualitative methods can make the process of clustering the time series data. The accuracy of hierarchical clustering methods can be seen from the average value of ¬ f-measure accuracy in the can. Accuracy of each of the experiments that have f-measure value is highest in the second scenario where the value of f-measure in the can is 0.9792.

Keywords: Data Mining, Clustering, Hierarchical Clustering, Time Series, Short Time Series.

KATA PENGANTAR

Alhamdulillahirabbil'alamin. Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayah-Nya, Skripsi yang berjudul **"Implementasi Metode Hierarchical Clustering Pada Ozone Level Detection Data Set"** ini dapat berjalan dengan baik. Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, Program Teknologi Informatika dan Ilmu Komputer (PTIIK), Universitas Brawijaya Malang.

Shalawat serta salam tetap tercurahkan kepada Baginda Rasulullah Muhammad *Shalallahu 'alaihi wasallam*, makhluk paling mulia yang senantiasa memberikan cahaya petunjuk, seorang uswatun hasanah yang telah membawa agama Allah yaitu agama Islam menjadi agama yang *Rahmatan Lil 'Alamin*.

Dalam penyelesaian skripsi ini, penulis telah mendapat begitu banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Indriati, ST., M.Kom., selaku dosen pembimbing utama yang telah meluangkan waktu untuk memberikan pengarahan dan masukan bagi penulis.
2. Candra Dewi, S.Kom., MSc., selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan pengarahan dan masukan bagi penulis.
3. Drs. Marji, MT., selaku Ketua Program Studi Informatika/Ilmu Komputer Program Teknologi Informasi & Ilmu Komputer Universitas Brawijaya Malang.
4. Issa Arwani, S.Kom., M.Sc. selaku sekertaris jurusan.
5. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Teknik Informatika / Ilmu Komputer Program Teknologi Informasi dan Ilmu Komputer (PTIIK) Universitas Brawijaya.
6. Unit konseling Program Teknologi Informatika dan Ilmu Komputer (PTIIK) Universitas Brawijaya yang telah memberikan dukungan motivasi dan berbagai masukan terkait pelaksanaan penyusunan skripsi ini.

7. Segenap staff dan karyawan di Program Teknologi Informatika dan Ilmu Komputer (PTIIK) Universitas Brawijaya yang telah banyak membantu penulis dalam pelaksanaan penyusunan skripsi ini.
8. Kedua orang tua dan keluarga tercinta, terima kasih atas semua doa, kasih sayang dan perhatian yang tulus serta dukungan yang telah diberikan.
9. Teman-teman angkatan 2006, dan 2007 yang sudah banyak membantu mengajarkan tentang konsep dan coding pada skripsi ini.
10. Aditya Hari, S.Kom, Ardhi Prhiagung, dan Wijayadi Saputra yang sudah berperan besar telah mengajarkan konsep dan coding pada skripsi ini.
11. Rekan-rekan di Program Studi Ilmu Komputer,Program Teknologi Informatika dan Ilmu Komputer (PTIIK) Universitas Brawijaya yang telah banyak memberikan bantuannya demi kelancaran pelaksanaan penyusunan skripsi ini.
12. Dan semua pihak yang telah terlibat baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu persatu terima kasih atas semua bantuan yang telah diberikan.

Semoga skripsi ini bermanfaat bagi pembaca sekalian. Akhirnya, penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan dan mengandung banyak kekurangan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca.

Malang, 1 Agustus 2013

Penulis

DAFTAR ISI

LEMBAR PERSETUJUAN ..... **i**
LEMBAR PENGESAHAN ..... **ii**
PERNYATAAN ORISINALITAS SKRIPSI ..... **iii**
ABSTRAK ..... **iv**
ABTRACT ..... **v**
KATA PENGANTAR ..... **vi**
DAFTAR ISI ..... **viii**
DAFTAR GAMBAR ..... **x**
DAFTAR TABEL ..... **xi**
DAFTAR SOURCE CODE ..... **xii**
BAB I : PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Rumusan Masalah 2
1.3. Batasan Masalah 2
1.4. Tujuan Penelitian 3
1.5. Manfaat Penelitian 3
1.6. Sistematika Penulisan 3
BAB II : KAJIAN PUSTAKA DAN DASAR TEORI 5
2.1. Kajian Pustaka 5
2.2. Data Mining 5
2.2.1. Kegunaan Data Mining 8
2.2.2. Tranasformasi Data 10
2.3. Ozon 10
2.3.1. Dampak Akibat Penipisan Ozon 12
2.4. Clustering 12
2.4.1. Subspace Clustering 13
2.4.2. Teknologi Cluster 15
2.4.3. Macam-Macam Metode Clustering 15
2.5. Data Time Series 17
2.6. Perhitungan Jarak Untuk Short Time Series 18
2.7. Evaluasi 19

2.7.1. Precision 20
2.7.2. Recall 20
2.7.3. F-Measure 20
BAB III : METODOLOGI PENELITIAN DAN PERANCANGAN 21
3.1. Studi Literatur 21
3.2. Data Penelitian 21
3.3. Deskripsi Umum Sistem 22
3.4. Perancangan Proses 22
3.4.1. Perancangan Proses Inisialisasi 24
3.4.2. Perancangan Proses Perhitungan Jarak 24
3.5. Perhitungan Manual 25
3.5.1. Perhitungan Clustering 26
3.5.2. Perhitungan Manual Evaluasi 29
3.6. Rancangan Uji Coba 30
3.6.1. Rancangan Desain Antarmuka 30
3.7. Skenario Pengujian 30
BAB IV : IMPLEMENTASI 31
4.1. Implementasi Perangkat Keras 31
4.2. Implementasi Perangkat Lunak 31
4.3. Implementasi Proses 31
4.3.1. Implementasi Proses Inisialisasi 31
4.3.2. Implementasi Perhitungan Jarak 32
4.3.3. Implementasi Proses Perhitungan Jarak Minimum 33
4.4. Implementasi Antarmuka 37
BAB V : PENGUJIAN DAN ANALISIS 40
5.1. Bahan Pengujian 40
5.2. Skenario Pengujian 40
5.3. Pengujian dan Analisa Hasil 41
5.4. Kesimpulan Pengujian 45
BAB VI : KESIMPULAN DAN SARAN 47

6.1. Kesimpulan 47
6.2. Saran 47
DAFTAR PUSTAKA 48

DAFTAR GAMBAR

Gambar 2.1. Fase-fase dalam data mining 7
Gambar 2.2. Contoh Sebuah Dendogram..... 17
Gambar 2.3. Time Series Dengan Pnjang 5 18
Gambar 3.1. Flowchart Sistem 23
Gambar 3.2. Flowchart Proses Inisialisasi 24
Gambar 3.3 Flowchart Proses Perhitungan Jarak 25
Gambar 3.5. Rancangan Antarmuka Sistem 30
Gambar 4.4. Tampilan Program Dengan Jumlah Data 100 37
Gambar 4.5. Tampilan Program Dengan Jumlah Data 500 38
Gambar 4.6. Tampilan Program Dengan Jumlah Data 2536 38
Gambar 5.1. Grafik Hasil Pengujian Data Pertama 41
Gambar 5.2. Grafik Waktu Hasil Pengujian Data Pertama 42
Gambar 5.3. Grafik Hasil Pengujian Data Kedua..... 43
Gambar 5.4. Grafik Waktu Hasil Pengujian Data Kedua 43
Gambar 5.5. Grafik Hasil Pengujian Data Ke Tiga 44
Gambar 5.6. Grafik Waktu Hasil Pengujian Data Ke Tiga 45
Gambar 5.7. Grafik Rata-rata Nilai F-Measure..... 46
Gambar 5.8. Grafik Rata-rata Waktu 46

DAFTAR TABEL

Tabel 2.1. Definisi Dari Fungsi Diff 19
Tabel 2.2. Matriks Confusion 19
Tabel 3.1. Tabel Dataset Uji 1 26
Tabel 5.1 Tabel Pengujian 40
Tabel 5.2 Tabel Hasil Pengujian Data pertama 41
Tabel 5.3. Tabel Hasil Pengujian Data Kedua 42
Tabel 5.4. Tabel Hasil PengujianData ke-Tiga 44

DAFTAR SOURCE CODE

Source Code 4.1. Proses Inisialisasi 32
Source Code 4.2. Proses Perhitungan Jarak 33
Source Code 4.3. Proses Perhitungan Jarak Minimum..... 34

BAB I
PENDAHULUAN

1.1. Latar Belakang

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*. Salah satu algoritma *clustering* adalah *Qualitative Hierarchical Clustering* yang dimana algoritma tersebut adalah metode analisis *cluster* yang membangun hirarki dari *cluster*. Jarak antara tiap obyek merupakan *input* dari algoritma ini. Iterasi terus berlanjut hingga semua obyek telah di*cluster* menjadi sebuah *cluster* saja. *Output* dari algoritma ini adalah sebuah *dendrogram* (*Hierarchical tree*). *Dendrogram* adalah sebuah *binary tree* di mana setiap *cluster* awalnya berisi satu elemen saja yaitu *leave* dari *tree* tersebut.

Salah satu penggunaan algoritma ini adalah pada *clustering dataset time series*. Dataset *time series* adalah data sebuah kumpulan dari pengamatan yang dibuat secara berurutan berdasarkan waktu, misalnya total penjualan per bulan, panggilan telepon per hari, perubahan inventori per minggu dan lain sebagainya.

Pada penelitian yang dilakukan oleh Peter Ljubic dengan judul *Time Series Analysis Of UK Traffic Accident Data* [LJU-02] menganalisis *database* yang berisi data kecelakaan lalu lintas dengan *time series* yang singkat, untuk memberikan pemahaman yang lebih baik pada data melalaui penggunaan statistik sederhana dan visualisasi, sekaligus menemukan beberapa trend dan pola pada perubahan dinamis pada jumlah kecelakaan lalu lintas. Pada jurnal ini memakai metode yang sama yaitu metode *Qualitative Hierarchical Clustering yang* mempunyai akurasi menghasilkan 2 *cluster,* yang berarti bahwa pada kasus ini memiliki dua jenis dinamika. *Cluster* pertama berisi 40 data dengan tren yang menurun dalam jumlah kecelakaanya, sedangkan *cluster* kedua berisi 11 data dengan tren yang meningkat. Pada penelitian Monica Sanwlani yang bejudul *Forecasting Sales Trough Time Series Clustering* [SAN-13] pada tahun 2013 tentang *clustering* pada data *time series* ini menggunakan *dataset* adalah data penjualan. Data tersebut dibagi menjadi komponen-komponen seperti trend,

ketidak teraturan, dan musiman. Hasil peramalan para ahli digunakan untuk meramalkan setiap komponen tersebut. Untuk meramalkan kompoen-komponen tersebut digunakan metode *statistical ARIMA, hot winter,* dan *exponential smoothing* yang digunakan untuk meramal data-data tersebut. Perbedaan dengan skripsi yang akan dibuat yaitu studi kasus dan *output* diagram yang dimana pada skripsi ini akan di ambil studi kasus *dataset* tentang *ozone level dataset*.

Pada skripsi ini menggunakan sebuah pendekatan *clustering* untuk menganalisa data *time series*. Dengan menggunakan sebuah metode *Hierarchical Clustering* untuk mengukur jarak antara data *time series* yang telah ditentukan. Algoritma *Hierarchical* dimulai dengan menjadikan tiap objek menjadi sebuah *cluster* dan secara iterasi menggabungkan tiap *cluster* yang mirip. Jarak antar objek merupakan input dari algoritma ini. Iterasi terus berlanjut sampai semua objek telah di *cluster* hingga menjadi *cluster* saja. Dari hasil pengujian pada algoritma ini akan diketahui tingkat akurasi dari metode *qualitative hierarchical clustering*. Karena itu dibuat skripsi ini yang berjudul Implementasi Metode *Hierarchical Clustering* Pada *Ozone Level Detection Data Set*.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka permasalahan pada skripsi ini adalah seperti berikut :

1. Bagaimana menerapkan metode *qualitative hierarchical clustering* pada data *time series.*
2. Bagaimana mengetahui tingkat akurasi dari metode *qualitative hierarchical clustering* pada data *Ozone Level Detection Data Set.*

1.3. Batasan Masalah

Batasan masalah yang digunakan untuk mencegah meluasnya permasalahan dalam skripsi ini adalah:

1. Perhitungan jarak tiap *cluster* ini memakai metode qualitative *clustering*.
2. Tidak ada perbandingan metode dalam skripsi ini.

3. Data yang digunakan pada penelitian ini adalah data tentang *Ozone Level Detection Data Set* yang diambil dari *UCI* (*www.archive.ics.UCI.edu.com*) yang diterbitkan pada tahun 1998.

1.4. Tujuan Penelitian

Tujuan dari skripsi yang ingin dicapai adalah:

1. Mengimplementasikan metode *qualitative hierarchical clustering* pada *Ozone Level Detection Data Set*.
2. Mengetahui kelayakan dari metode *Hierarchical Clustering* pada dataset ozone.

1.5. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian skripsi ini adalah:

1. Dapat mengetahui kelayakan metode *Hierarchical Clustering* pada *Ozone Level Detection Data Set*.

1.6. Sistematika Penulisan

Sistematika penulisan skripsi ini dibagi menjadi lima bab dengan masing-masing bab diuraikan sebagai berikut:

1. BAB I PENDAHULUAN

   Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penulisan, manfaat penulisan, dan sistematika penulisan.
2. BAB II TINJAUAN PUSTAKA

   Bab ini membahas mengenai teori-teori penunjang yang membahas teori *Data time series, Hierarchical Clustering*, dan metode *Qualitative* yang merupakan bahasan utama dalam penelitian ini.
3. BAB III METODE DAN PERANCANGAN SISTEM

   Membahas metode yang akan digunakan pada penelitian skripsi ini. Terdiri dari literature, perancangan system, implementasi algoritma, pengujian dan hasil implementasi, penulisan laporan.

4. BAB IV IMPLEMENTASI

   Membahas tentang implementasi metode yang akan digunakan dalam *Data time series*, *Hierarchical Clustering*, dan metode *Qualitative* dan uji coba program yang telah dibuat.

5. BAB V PENGUJIAN DAN ANALISIS

   Membahas proses dan melakukan analisa terhadap hasil pengujian algoritma pada *Data time series*, *Hierarchical Clustering*, dan metode *Qualitative*.

6. BAB VI KESIMPULAN DAN SARAN

   Berisi tentang kesimpulan yang didapat dari pembuatan skripsi ini dan saran-saran yang mungkin dapat berguna dalam penelitian lebih lanjut.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

2.1. Kajian Pustaka

Pada sebuah penelitian tentang data *time series* yang telah dilakukan oleh Ljupco Todorovski yang berjudul *Qualitative Clustering of Short Time-Series: A Case Study of Firms Reputation Data*, data yang digunakan adalah *dataset time series* tentang reputasi perusahaan. Metode *clustering* yang digunakan adalah metode hirarki *standart*. Perhitungan jaraknya menggunakan metode baru yang diajukan pada penelitian. Pendekatan *clustering* dilakukan pada perubahan pola reputasi perusahaan dari Slovenia sepanjang tahunnya. *Cluster* yang terbentuk pada penelitian ini adalah empat *cluster*.

Pada penelitian Monica Sanwlani yang bejudul *Forecasting Sales Trough Time Series Clustering* [SAN-13] pada tahun 2013 tentang *clustering* pada data *time series* ini menggunakan *dataset* yang berupa data penjualan. Data tersebut dibagi menjadi komponen-komponen seperti trend, ketidak teraturan, dan musiman. Hasil peramalan para ahli digunakan untuk meramalkan setiap komponen tersebut. Untuk meramalkan kompoen-komponen tersebut digunakan metode *statistical ARIMA, hot winter,* dan *exponential smoothing*. *Clustering* dilakukan pada peramalan tersebut dan akhirnya ditemukan para ahli yang memiliki peramalan terbaik, baik, dan buruk. Untuk membandingkan peramalan-peramalan digunakan *absolute percentage error*.

2.2. Data Mining

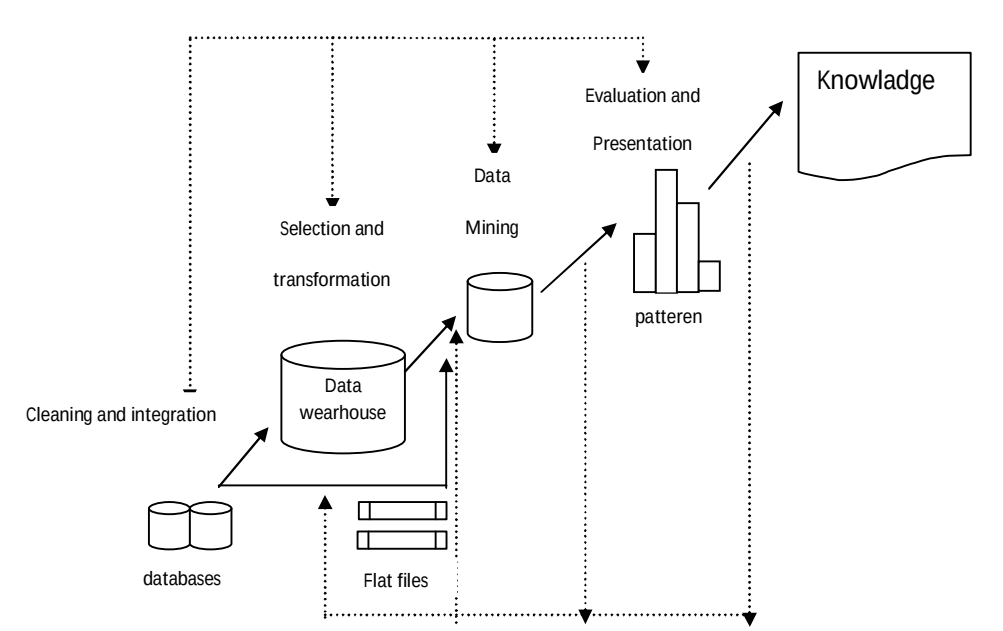
Data mining adalah suatu proses menggali (mining) pengetahuan dari sejumlah data yang besar, data mining juga disebut dengan *knowledge mining from databases, knowledge extraction*, data/*pattren analysis*, data *archeology*, dan data *design*. Data mining juga merupakan suatu proses *ekstraksi* informasi yang potensial, implisit, dan tidak diketahui sebelumnya (misalnya aturan-aturan, batasan-batasan, dan regularitas) dari sekumpulan data dalam *database* yang besar [SAN-07].

Data mining dapat dikatakan sebagai basil dari evolusi teknologi informasi. Yaitu mulai dari *system data collection, database creation*, data

management (termasuk *storage* dan *retrieval* dan *database transaction processing*), dan data *analysis* dan *understanding*. Semakin berkembangnya kecepatan dalam mengumpulkan dan penyimpanan data dalam jumlah yang besar makin mengakibatkan semakin sulitnya data yang banyak tersebut untuk dianalisa oleh manusia tanpa bantuan alat yang dapat menggali informasi yang ada dalam data yang banyak tersebut, akibatnya seringkali keputusan - keputusan yang diambil tidak berdasarkan informasi dari data yang ada, hal ini terjadi karena para pengambil keputusan tidak memiliki alat bantu yang benar atau memadai untuk melakukan pencarian informasi dari data yang banyak [SAN-07]. Sistem data mining yang baik dibangun dengan algoritma yang baik, terstruktur, cepat dan juga dapat menangani data dalam jumlah besar sehingga ketika menangani suatu *database* dalam jumlah kecil maupun besar, *running time*nya sesuai dengan jumlah data dalam *database* tersebut. Dengan melakukan data mining, *knowledge* yang baik, *high level information* dapat digali dari *database* atau di tampilkan dari berbagai sudut pandang. Data mining umumnya bisa dilakukan dari berbagai macam data yang tersimpan baik dan terstruktur pada relational *database*, *data warehouse*, *transaction aldatabase*, dan sistem *database* pada internet [SAN-07]. Data Mining memang salah satu cabang ilmu komputer yang relatif baru. Dan sampai sekarang orang masih memperdebatkan untuk menempatkan data mining di bidang ilmu mana, karena data mining menyangkut *database*, kecerdasan buatan (*artificial intelligence*), statistik, dan lain sebagainya. Ada pihak yang berpendapat bahwa data mining tidak lebih dari *machine learning* atau analisa statistik yang berjalan di atas *database*. Namun pihak lain berpendapat bahwa *database* berperanan penting di data mining karena data mining mengakses data yang ukurannya besar (bisa sampai *terabyte*) dan disini terlihat peran penting *database* terutama dalam optimisasi *query*-nya. Data mining yang juga melakukan kegiatan mengekstraksi atau menambang pengetahuan dari data yang berukuran/berjumlah besar, informasi inilah yang nantinya sangat berguna untuk pengembangan. Kehadiran data mining dilatar belakangi dengan problema data *explosion* yang dialami akhir-akhir ini dimana banyak organisasi telah mengumpulkan data sekian tahun lamanya (data pembelian, data penjualan, data nasabah, data transaksi dan masih banyak lagi).

Hampir semua data tersebut dimasukkan dengan menggunakan aplikasi komputer yang digunakan untuk menangani transaksi sehari-hari yang kebanyakan adalah OLTP (*On Line Transaction Processing*). Bayangkan berapa transaksi yang dimasukkan oleh *hypermarket* dalam seharinya dan bayangkan betapa besarnya ukuran data mereka jika nanti telah berjalan beberapa tahun. Pertanyaannya sekarang, apakah data tersebut akan dibiarkan menggunung, tidak berguna lalu dibuang, ataukah kita dapat menambangnya untuk informasi yang berguna bagiorganisasi kita. Banyak diantara kita yang kebanjiran data tapi miskin informasi [SAN-07].

Langkah-langkah untuk melakukan data mining di tunjukan pada Gambar 2.1 ini adalah langkah-langkah data mining dari proses awal hingga proses akhir.



Gambar 2.1. Fase-fase dalam data mining

- *Data cleaning* (untuk menghilangkan *noise* data yang tidak konsisten)
  Data *integration* (di mana sumber data yang terpecah dapat disatukan).
- *Data selection* (di mana data yang relevan dengan tugas analisis dikembalikan ke dalam *database*).
- *Data transformation* (di mana data berubah atau bersatu menjadi bentuk yang tepat untuk menambang dengan ringkasan performa atau operasi agresi).

- Data mining (proses esensial di mana metode yang intelejen digunakan untuk mengekstrak pola data).
- *Pattern evolution* (untuk mengidentifikasi pola yang benar-benar menarik yang mewakili pengetahuan berdasarkan atas beberapa tindakan yang menarik).
- *Knowledge presentation* (di mana gambaran teknik visualisasi dan pengetahuan digunakan untuk memberikan pengetahuan yang telah ditambang kepada user) [SAN-07].

Ada beberapa jenis data dalam data mining yaitu : *Relation Database* yaitu sebuah sistem *database*, atau disebut juga *database* management *system* (DBMS), mengandung sekumpulan data yang saling berhubungan, dikenal sebagai sebuah *database*, dan satu set program perangkat lunak untuk mengatur dan mengakses data tersebut. *Data Warehouse* adalah sebuah ruang penyimpaan informasi yang terkumpul dari beraneka macam sumber, disimpan dalam skema yang menyatu, dan biasanya terletak pada sebuah *site* [SAN-07]. Ada beberapa teknik data mining yang berkembang saat ini, di antaranya: pencarian aturan asosiasi, pencarian pola sekuensial, klasifikasi data dan klasterisasi data. Sebetulnya masih ada teknik lain, namun tidak dibahas dalam skripsi ini.

2.2.1 Kegunaan data mining

Kegunaan data mining adalah untuk menspesifikasikan pola yang harus diemukan dalam tugas data mining. Secara umum tugas data mining dapat diklasifikasikan ke dalam dua kategori: deskriptif dan prediktif. Tugas menambang secara deskritpif adalah untuk mengklasikasikan sifat umum suatu data di dalam *database*. Tugas data mining secara prediktif adalah untuk mengambil keputusan terhadap data terakhir untuk membuat prediksi [FAD-11].

1. **Konsep/Class Description**

Data dapat dialokasikan dengan pembagian *class* atau konsep. Untuk contohnya, ditoko *All Electronics, pembagian class* untuk barang yang akan dijual termasuk *computer* dan *printer*, dan konsep konsumen adalah *big spenders* dan *budget spender*. Hal tersebut berguna untuk menggambarkan pembagian *class*secara individual dan konsep secara ringkas, laporan ringkas, dan juga

pengaturan harga. Deskripsi suatu class atau konsep seperti itu disebut *class/concept description* [FAD-11].

2. **Assocition Analysis**

Assocition Analysis adalah penemuan *association rules* yang menunjukan kondisi suatu *attribut*e yang terjadi bersama-sama secara terus-menerus dalam memberikan set data. *Association* analisis secara luas dipakai untuk *market basket* atau analisa data transaksi [FAD-11].

3. **Klasifikasi dan prediksi**

Klasifikasi dan prediksi mungkin perlu diproses oleh analisis relevan, yang berusaha untuk mengidentifikasi atribut-atribut yang ditambahkan pada proses klasifikasi dan prediksi. Atribut-atribut ini kemudian dapat di kleuarkan [FAD-11].

4. **Cluster Analysis**

Tidak seperti klasifikasi dan prediksi, yang menganalisis objek data dengan kelas yang terlabel, clustering menganalisis objek data tanpa mencari keterangan pada label kelas yang diketahui. Pada umumnya, label kleas tidak ditampilkan di dalam latihan data *simply*, karena mereka tidak tahu bagaimana memulainya. *Clustering* dapat digunakan untuk menghasilkan label-label [FAD-11].

5. Outliner Analysis

Outliner dapat dideteksi menggunakan test yang bersifat statistik yang menagmbil sebuah distribusi atau probabilitas model untuk data, atau menggunakan langkah-langkah jarak jauh dimana objek yang penting jauh dari *cluster* lainnya dianggap *outliner*. Sebuah *database* mungkin mengandung objek data yang tidak di ikuti tingkah laku yang umum atau model dari data. Data ini disebut *outliner* [FAD-11].

6. Evolution Analysis

Data analis evolusi menggambarkan ketetapan model atau kecenderungan objek yang dimiliki kebiasaan berubah setip waktu. Meskipun ini mungkin termasuk karakteristik, deskriminasi, asosiasi, klasifikasi, atau *clustering data* berdasarkan waktu, kelebihan yang jelas seperti analisa termasuk analisa data *time*

series, urutan atau pencocokan pola secra berkala, dan kesamaan berdasarkan analisa data [FAD-11].

2.2.2 Transformasi data

Transformasi data melakukan peringkasan data dengan mengasumsikan bahwa data telah tersimpan dalam tempat penyimpanan tunggal. Pada langkah terakhir, data telah di ekstrak dari banyak basis data ke dalam basis data tunggal. Tipe peringkasan yang dikerjakan dalam langkah ini mirip dengan peringkasan yang dikerjakan selama tahap ekstraksi. Beberapa perusahaan memilih untuk memangkas data dalam sebuah tempat penyimpanan tunggal. Fungsi-fungsi *Agregate* yang sering digunakan antara lain: *summarizations, averages, minimum, maximum, dan count*. Data-data yang telah terkumpul selanjutnya akan mengalami proses pembersihan. Proses pembersihan data dilakukan untuk membuang *record* yang tidak di gunakan pada input yang akan di *cluster*, menstandarkan atribut-atribut, merasionalisasi struktur data, dan mengendalikan data yang hilang. Data yang tidak konsisten dan banyak kekeliruan membuat hasil data mining tidak akurat. Adalah sangat penting untuk membuat data konsisten dan seiagam. Pembersihan data juga dapat membantu perusahaan untuk mengkonsolidasikan *record*. ini sangat berguna ketika sebuah perusahaan mempunyai banyak *record* untuk seorang pelanggan. Setiap *record* atau *file* pelanggan mempunyai nomor pelanggan yang sama, tetapi informasi dalam tiap *file*nya berbeda [HAN-06].

Untuk melakukan *clustertime* series dengan perhitungan jarak menggunakan metode *qualitative*, data yang ada pada *database* harus ditransformasi terlebih dahulu menjadi bentuk yang telah dikelompokkan. Pengelompokkan dilakukan berdasarkan kebutuhan informasi yang dianalisa [HAN-06].

2.3. Ozon

Ozon adalah gas beracun sehingga bila berada dekat permukaan tanah akan berbahaya bila terhisap dan dapat merusak paru-paru. Sebaliknya, lapisan ozon di *atmosfer* melindungi kehidupan di Bumi karena ia melindunginya dari

radiasi sinar *ultraviolet* yang dapat menyebabkan kanker. Oleh karena itu, para ilmuwan sangat khawatir ketika mereka menemukan bahwa bahan kimia *Cloro Flouro Carbon* (CFC) yang biasa digunakan sebagai media pendingin dan gas pendorong *spray aerosol*, memberikan ancaman terhadap lapisan ini. Bila dilepas ke *atmosfer*, zat yang mengandung *klorin* ini akan dipecah oleh sinar matahari yang menyebabkan klorin dapat bereaksi dan menghancurkan molekul-molekul ozon. Setiap satu molekul *Cloro Flouro Carbon* mampu menghancurkan hingga 100.000 molekul ozon. Oleh karena itu, penggunaan CFC dalam *aerosol* dilarang di Amerika Serikat dan negara-negara lain di dunia. Bahan-bahan kimia lain seperti bromin halokarbon, dan juga *nitrogen oksida* dari pupuk, juga dapat menyerang lapisan ozon [WES-98].

Menipisnya lapisan ozon dalam *atmosfer* bagian atas diperkirakan menjadi penyebab meningkatnya penyakit kanker kulit dan katarak pada manusia, merusak tanaman pangan tertentu, memengaruhi *plankton* yang akan berakibat pada rantai makanan di laut, dan meningkatnya *karbondioksida* (lihat pemanasan global) akibat berkurangnya tanaman dan *plankton*. Sebaliknya, terlalu banyak ozon di bagian bawah *atmosfer* membantu terjadinya kabut campur asap, yang berkaitan dengan iritasi saluran pernapasan dan penyakit pernapasan akut bagi mereka yang menderita masalah *kardiopulmoner* [WES-98].

Pada awal tahun 1980-an, para peneliti yang bekerja di antartika mendeteksi hilangnya ozon secara periodik di atas benua tersebut. Keadaan yang dinamakan lubang ozon (suatu area ozon tipis pada lapisan ozon) ini, terbentuk saat musim semi di Antartika dan berlanjut selama beberapa bulan sebelum menebal kembali. Studi-studi yang dilakukan dengan balon pada ketinggian tinggi dan satelit satelit-satelit cuaca menunjukkan bahwa persentase ozon secara keseluruhan di Antartika sebenarnya terus menurun. Penerbangan-penerbangan yang dilakukan untuk meneliti hal ini juga memberikan hasil yang sama. Pada tahun 1987, ditandatangani Protokol Montreal, suatu perjanjian untuk perlindungan terhadap lapisan ozon. Protokol ini kemudian diratifikasi oleh 36 negara termasuk Amerika Serikat. Pelarangan total terhadap penggunaan CFC sejak 1990 diusulkan oleh Komunitas Eropa sekarang Uni Eropa pada tahun 1989, yang juga disetujui oleh Presiden AS George Bush. Pada Desember 1995, lebih

dari 100 negara setuju untuk secara bertahap menghentikan produksi pestisida metil bromida di negara-negara maju. Bahan ini diperkirakan dapat menyebabkan pengurangan lapisan ozon hingga 15 persen pada tahun 2000. CFC tidak diproduksi lagi di negara maju pada akhir tahun 1995 dan dihentikan secara bertahap di negara berkembang hingga tahun 2010. *Hidrofluorokarbon* atau HCFC, yang lebih sedikit menyebabkan kerusakan lapisan ozon bila dibandingkan CFC, digunakan sementara sebagai pengganti CFC, hingga 2020 pada negara maju dan 2016 di negara berkembang. Untuk memonitor berkurangnya ozon secara global, pada tahun 1991, "National Aeronautics and Space Administration" NASA meluncurkan Satelit Peneliti Atmosfer. Satelit dengan berat 7 ton ini mengorbit pada ketinggian 600 km (372 mil) untuk mengukur variasi ozon pada berbagai ketinggian dan menyediakan gambaran jelas pertama tentang kimiawi atmosfer di atas [WES-98].

2.3.1. Dampak Akibat Penipisan Ozon

Ancaman yang diketahui terhadap keseimbangan ozon adalah *kloroflorokarbon* (CFC) buatan manusia yang meningkatkan kadar penipisan ozon menyebabkan kemerosotan berangsur-angsur dalam tingkat ozon global. CFC digunakan oleh masyarakat modern dengan cara yang tidak terkira banyaknya, dalam kulkas, bahan dorong dalam penyembur, pembuatan busa dan bahan pelarut terutama bagi kilang-kilang elektronik. Masa hidup CFC berarti 1 molekul yang dibebaskan hari ini bisa ada 50 hingga 100 tahun dalam atmosfer sebelum dihapuskan. Dalam waktu kira-kira 5 tahun, CFC bergerak naik dengan perlahan ke dalam *stratosfer* (10 – 50 km). Di atas lapisan ozon utama, pertengahan *julat* ketinggian 20 – 25 km, kurang sinar UV diserap oleh ozon. Molekul CFC terurai setelah bercampur dengan UV, dan membebaskan atom klorin. Atom klorin ini juga berupaya untuk memusnahkan ozon dan menghasilkan lubang ozon [WES-98].

2.4. Clustering

Clustering adalah 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* dalam data mining berguna untuk menemukan pola distribusi di dalam sebuah dataset yang berguna untuk proses analisa data. Kesamaan objek biasanya diperoleh dari kedekatan nilai-nilai atribut yang menjelaskan objek-objek data, sedangkan objek-objek data biasanya direpresentasikan sebagai sebuah titik dalam ruang multi dimensi [HAN-06].

Clustering digunakan untuk mendapatkan *high availability* dan scalability. Pada *high available cluster*, dapat digunakan *fail over database cluster*, dimana hanya ada satu *node* yang aktif melayani *user*, sedangkan *node* lainnya *standby*. *Storage* yang digunakan mempunyai koneksi ke setiap *node* pada *cluster*, sehingga jika *primary node* mati, *database* engine, *listener process*, dan *logical host ip address* akan dijalankan pada *secondary node* tanpa perlu menunggu *operating system boot*, sehingga *downtime* dapat diminimalisasi. *Highavailability* mempunyai standar *duptime* 99.999 persen, atau hanya boleh mati selama 5 menit dalam setahun. Beberapa contoh *software* yang dapat digunakan untuk membuat *HA cluster* adalah *Sun Cluster* dan *Veritas Cluster*. Pada *scalable cluster*, digunakan produk *Oracle RAC*, dimana setiap *node* aktif melayani user, sehingga diperoleh performa yang semakin baik dengan menggunakan lebih banyak *node*. *Sun cluster* dapat digunakan sampai 16 *node*, sedangkan *Veritas Storage Foundation for Oracle RA*C bias sampai 32 *node*. Jika ada *node* yang mati, tentu akan menurunkan performa, namun tidak terjadi *downtime*. Pada *scalable cluster*, seluruh *node* dapat terhubung secara langsung ke *share* di *storage*, namun dapat juga tidak mempunyai koneksi fisik ke *storage*, melainkan melalui *private cluster transport* [HAN-06].

2.4.1. Subspace Clustering

Subspace clustering adalah suatu teknik *clustering* yang mencoba menemukan *cluster* pada dataset multi dimensi dengan pemilihan dimensi yang paling relevan untuk setiap *cluster*, karena pada data multi dimensi kemungkinan terdapat dimensi-dimensi yang tidak relevan yang dapat membingungkan algoritma *clustering* sehingga bisa mengaburkan *cluster* sebenarnya yang

seharusnya dapat ditemukan. Masalah lainnya, *cluster* dapat saja berada dalam *subspace* yang berbeda, dimana setiap *subspace* dibentuk dari kombinasi dimensi yang berbeda-beda. Akibatnya, semakin banyak dimensi yang digunakan, *cluster* akan sulit ditemukan. *Subspace clustering* secara otomatis akan menemukan unit-unit yang padat pada tiap *subspace* [HAN-06].

Berbeda dengan *association rule mining* dan *classification* dimana kelas data telah ditentukan sebelumnya, *clustering* melakukan pengelompokan data tanpa berdasarkan kelas data tertentu. Bahkan *clustering* dapat dipakai untuk memberikan label pada kelas data yang belum diketahui itu. Karena itu *clustering* sering digolongkan sebagai metode *unsupervised learning*. Prinsip dari *clustering* adalah memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar *class*/*cluster*. *Clustering* dapat dilakukan pada data yang memiliki beberapa atribut yang dipetakan sebagai ruang multidimensi. Banyak algoritma *clustering* memerlukan fungsi jarak untuk mengukur kemiripan antar data, diperlukan juga metode untuk normalisasi bermacam atribut yang dimiliki data. Beberapa kategori algoritma *clustering* yang banyak dikenal adalah metode partisi dimana pemakai harus menentukan jumlah partisi yang diinginkan lalu setiap data dicoba untuk dimasukkan pada salah satu partisi, metode lain yang telah lama dikenal adalah metode hierarki yang terbagi dua lagi : *bottom-up* yang menggabungkan *cluster* kecil menjadi *cluster* lebih besar dan *top-down* yang memecah *cluster* besar menjadi *cluster* yang lebih kecil. Kelemahan 3 metode ini adalah bila salah satu penggabungan/pemecahan dilakukan pada tempat yang salah, tidak dapat didapatkan *cluster* yang optimal. Pendekatan yang banyak diambil adalah menggabungkan metode hierarki dengan metode *clustering* lainnya seperti yang dilakukan oleh Chameleon [HAN-06].

Akhir-akhir ini dikembangkan juga metode berdasar kepadatan data, yaitu jumlah data yang ada disekitar suatu data yang sudah teridentifikasi dalam suatu *cluste*r. Bila jumlah data dalam jangkauan tertentu lebih besar dari nilai ambang batas, data-data tersebut dimasukkan dalam *cluster*. Kelebihan metode ini adalah bentuk *cluster* yang lebih *fleksibel*. Algoritma yang terkenal adalah DBSCAN [HAN-06].

2.4.2. Teknologi Cluster

Kebutuhan akan komunikasi data dewasa ini sangat penting seiring dengan kemajuan dan perkembangan teknologi komunikasi data yang semakin canggih. Hal yang sangat penting bagi teknologi komunikasi data adalah *database* sebagai penyedia data [HAN-06]. Aplikasi-aplikasi *database* dituntut untuk mampu melayani banyak akses data. Hal ini dapat dimaklumi karena *database* server telah di rancang untuk dapat melayani beragam jenis akses data. Saat ini aplikasi *database* semakin berkembang, baik dalam hal kegunaan, ukuran, maupun kompleksitas. Hal ini secara langsung akan berdampak pada server *database* sebagai penyedia layanan terhadap akses *database*, konsekuensi dari semua itu adalah beban *database server* akan semakin bertambah berat dan mengakibatkan kurang optimalnya kinerja dari server tersebut. Oleh sebab itu, diperlukan perancangan yang tepat dan handal dalam membangun *database* server. Solusi praktis dan tepat yang dapat diterapkan untuk mengatasi permasalahan diatas antara lain dengan penerapan Teknologi *Cluster*. Banyak sekali manfaat yang diperoleh dari teknologi *cluster* diantaranya meningkatkan ketersediaan dan performansi *system* [HAN-06].

Banyak sekali manfaat yang diperoleh dari teknologi *cluster* diantaranya meningkatkan ketersediaan dan performansi *system*. Pada *server database* yang besar dalam pelayanannya tidak menggunakan *server database* tunggal, tetapi dilayani oleh sekelompok *server database*, beberapa buah server *database* dihubungkan menjadi satu pada lingkungan yang sangat kompleks. Arsitektur yang digunakan harus menjamin bahwa sistem bekerja dengan baik, setiap *server* mengerjakan beban semestinya, dan tidak ada *bottleneck*. Berbagai jenis *server* diikat menjadi satu untuk menjadikan suatu pelayanan tunggal (*one stop shopping*). *Database* ter*cluster* tersebut dapat dianggap *personal desktop* atau *database* tunggal bagi penggunanya [HAN-06].

2.4.3. Macam-macam Metode Clustering

- ***Partition Clustering***

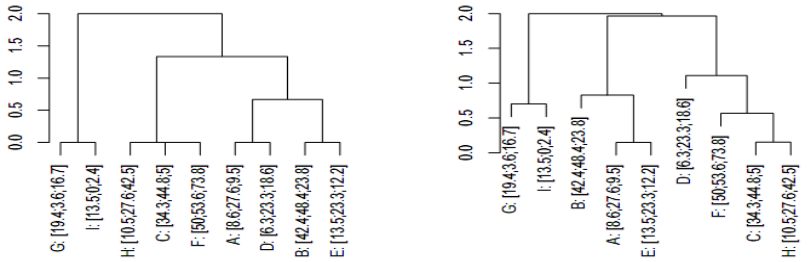
Membangun berbagai partisi dan kemudian mengevaluasi partisi tersebut dengan beberapa kriteria. Yang termasuk dalam metode ini meliputi algoritma *k*-

Means, k-Medoid, PROCLUS, CLARA, CLARANS, dan *PAM*. Dalam sub bab ini tidak akan dijelaskan karena tidak menyinggung dalam skripsi yang akan di kerjakan.

- ***Hierachical Clustering***

Beberapa tehnik *clustering hierarchical* bekerja dengan sederetan dari penggabungan yang berurutan atau sederetan dari pembagian yang berurutan. Metode *agglomerative* berawal dari objek-objek individual. Jadi pada awalnya banyaknya *cluster* sama dengan banyaknya objek. Pertama-tama objek dikumpulkan yang paling mirip, dan kelompok-kelompok awal ini digabungkan sesuai dengan kemiripannya. Akhirnya, sewaktu kemiripan berkurang semua subkelompok digabungkan menjadi satu *cluster* tunggal. Algoritma *Hierarchical* dimulai dengan menjadikan tiap obyek menjadi sebuah *cluster* dan secara iterasi menggabungkan tiap *cluster* yang mirip. Jarak antara tiap obyek merupakan inputdari algoritma ini. Iterasi terus berlanjut sampai semua obyek telah di*cluster* menjadi sebuah *cluster* saja. *Output* dari algoritma ini adalah sebuah *dendrogram* (*Hierarchical tree*). *Dendrogram* adalah sebuah *binary tree* di mana setiap *cluster* awalnya berisi satu elemen saja yaitu *leave* dari *tree* tersebut. Setiap internal *node* mewakili *cluster* yang merupakan gabungan dari obyek dari dua *cluster* pada *node* anak. Tingkat dari *node* tersebut sesuai dengan jarak antara *cluster* yang digabung. *Dendrogram* memiliki dua keterbatasan, karena masing-masing pengamatan harus ditampilkan sebagai daun yang mereka hanya dapat digunakan untuk sejumlah kecil pengamatan. Stata 7 memungkinkan hingga 100 pengamatan, bahkan dengan 75 pengamatan sulit untuk membedakan individu daun. Sumbu vertikal mewakili kriteria tingkat di mana setiap dua *cluster* dapat bergabung. Berturut-turut bergabung dengan *cluster* menyiratkan struktur hirarkis, yang berarti bahwa *dendrogram* hanya cocok untuk analisis *cluster hirarki. Dendrogram* seperti ponsel dan dapat secara bebas diputar di setiap *node*. Dalam contoh di atas, dendrogram bisa berputar sedemikian rupa sehingga sampel muncul dalam urutan yang berbeda, ditunjukkan di bawah ini, dengan kendala yang *dendrogram* tidak salib itu sendiri. Seperti pemintalan *dendrogram* adalah cara yang berguna untuk menonjolkan pola *chaining* atau kekhasan *cluster* (meskipun tidak membantu dalam kasus ini). Untuk program analisis *cluster* banyak, berputar harus dilakukan

(*tediously*) dalam sebuah program grafis yang terpisah, seperti *Illustrator*, tetapi dapat berputar dilakukan jauh lebih mudah langsung [TOD-02].



Gambar 2.2. Contoh Sebuah Dendrogram

Jarak hasil penggabungan tiap *cluster* yang mirip dapat menggunakan *single-link, average-link* atau *complete-link. Single-link* menghasilkan jarak antarasuatu *cluster* dengan *cluster* yang lain sama dengan jarak yang paling kecil. *Average-link* menghasilkan jarak antara *cluster* dengan *cluster* yang lain sama dengan jarak dari rata-rata. *Complete-link* menghasilkan jarak antara suatu *cluster* dengan *cluster* yang lain sama dengan jarak yang paling besar [TOD-02].

Di metode *hierarchical clustering* terdapat 3 metode yaitu metode *Single Linkage, Metode Complete Linkage, Metode Average Linkage.*

2.5. Data Time Series

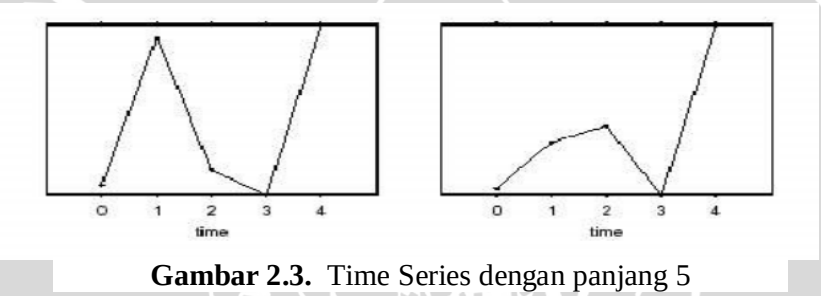
Data *time series* adalah sebuah kumpulan dari pengamatan yang dibuat secara berurutan berdasarkan waktu, misalnya total penjualan per bulan, panggilan telepon per hari, perubahan inventori per minggu dan lain sebagainya. Jadi frekuensi dari *time* series tergantung dari masalah atau problem yang di hadapi. Satuan waktu pada data *time* series dapat berupa harian, mingguan, bulanan, per *quater*, tahunan dan satuan waktu lainnya. Pemilihan satuan waktu ini sangat penting untuk permodelan dalam pengambilan keputusan nantinya. *Time* series melakukan pencarian terhadap *event-event* pada satuan waktu tertentu. Dengan memakai metode *time* series pada data *time series* maka akan membantu mengurangi jumlah informasi yang harus di analisa [TOD-02].

Salah satu metode yang digunakan adalah trend analysis, yang merepresentasikan pembahan berdasarkan waktu. Ada empat komponen dalam karakteristik *time series* yaitu: (1) *long term or trend movements*, yang melakukan

identifikasi dalam jangka waktu yang lama, di mana ditampilkan dalam bentuk kurva atau garis, (2) *cyclic movements or cyclic variations*, yang menampilkan kurva seperti *long term* namun dapat dilakukan secara *unperiodic*, (3) *seasonal movements or seasonal variations*, berdasarkan kejadian tertentu, seperti pada saat Natal, (4) *irregular or random movements*, merupakan pengambilan kejadian secara random [TOD-02]. Untuk menghasilkan analisis tentang trend yang terjadi dalam *system* informasi, digunakan pendekatan *cluster*ing berdasarkan analisis *qualitative* perhitungan jarak [TOD-02].

2.6. Perhitungan Jarak Untuk Short Time Series

Perhitungan jarak dapat dilakukan dengan menggunakan *qualitative analysist* dan membandingkan antar suatu *time series* [TOD-02].



Gambar 2.3. Time Series dengan panjang 5

Pada Gambar 2.2, dimisalkan bahwa gambar yang scbclah kiri adalah *time series* X dan gambar sebelah kanan adalah *time* series Y dengan panjang 5. Apabila dipilih suatu titik i dan j, maka pcrubahan yang tcrjadi pada X dan Y yang dilambangkan dengan q (Xi,Xj) (dan q(Yi,Yj)) mempunyai 3 macam kcmungkinan yaitu kenaikan, ketika Xi> Xj; tidak bcrubah, ketika Xi = Xj; pcnurunan, ketika Xi< Xj. Scbagai contoh, antara titik l dan 3 pada X dan Y, keduanya mcngalami kcnaikan; antara titik 2 dan 3 pada X dan Y, X mengalami penurunan dan Y mengalami kenaikan. Untuk menghasilkan jarak tiap *time series*, digunakan persamaan *qualitative* [TOD-02].

$$Dq(X,Y) = \frac{4}{N.(N-1)} . \sum_{i<j} \mathrm{Diff}(q(Xi, Xj), q(Yi, Yj)) \qquad (\mathbf{2.1})$$

Dimana persamaan (2.2) digunakan untuk melihat perubahan pada V, dan Diff merupakan fungsi untuk menentukan nilai dari 3 kemungkinan persamaan (2.2). Faktor 4/N (N-1) digunakan untuk melakukan normalisasi terhadap nilai

jarak yang ada. Fungsi Diff merupakan hasil dari perbedaan titik tersebut [TOD-02].

Tabel 2.1. Definisi dari fungsi Diff

Diff (q1,q2)	q1 Kenaikan	Tidak berubah	Penurunan
Kenaikan	0	0.5	1
q2 Tidak berubah	0,5	0	0,5
Penurunan	1	0,5	0

2.7. Evaluasi

Dalam melakukan evaluasi akurasi, perhitungan tersebut dibuat berdasarkan *matriks confusion* yang di gambarkan pada tabel 2.2, dimana matriks ini berisikan informasi mengenai prediksi atas suatu *class* yang telah dilakukan oleh sistem. *Actual* menyatakan *class* dari dataset latih yang sebenarnya, sedangkan *Predicted* menyatakan *class* dari dataset latih yang ditentukan oleh sistem. *True Positive Label* menyatakan jumlah dataset latih yang dikategorikan benar, sedangkan *False Negative Label* menyatakan jumlah dataset latih yang dikategorikan salah. *Positive Label* menyatakan jumlah dataset latih yang dikategorikan benar oleh sistem, sedangkan *Negative Label* menyatakan jumlah dataset latih yang dikategorikan salah oleh sistem [NUR-09].

Tabel 2.2. Matriks Confusion

	Predicted	
Actual	*Positive Label*	*Negative Label*
True Positive Label	A	B
False Negative Label	C	D

Misalkan diberikan suatu kategori C_m, maka :

- A menyatakan jumlah dataset yang dikategorikan sebagai kategori C_m oleh sistem yang juga merupakan member dari kategori C_m.
- B menyatakan jumlah dataset yang tidak dikategorikan benar/tidak relevan oleh sistem sebagai kategori C_m, namun sebenarnya merupakan member dari kategori C_m.

- C menyatakan jumlah dataset yang dikategorikan benar/relevan oleh sistem sebagai kategori C_m, namun sebenarnya bukan member dari kategori C_m.
- D menyatakan jumlah dataset yang tidak dikategorikan benar/relevan oleh sistem sebagai kategori C_m, dan juga bukan merupakan member dari kategori C_m.

2.7.1. Precision

Precision dapat didefinisikan sebagai kemiripan dari hasil kategorisasi suatu dataset. Nilai *precision* didapatkan menggunakan persamaan (2.3) dimana menyatakan perbandingan antara dataset yang dinyatakan benar (A) dengan seluruh dataset yang dikategorikan pada kategori tersebut (A+C). *Precision* dihitung dengan persamaan (2.3) [NUR-09].

$$Precision = \frac{A}{A+C} \quad \textbf{(2.3)}$$

2.7.2. Recall

Recall didefinisikan sebagai perbandingan antara jumlah dataset yang benar/relevan yang ditemukan oleh sistem (A) dengan jumlah dataset pada kategori (A+B). Nilai *recall* didapatkan dengan dengan menggunakan persamaan (2.4) [NUR-09].

$$Recall = \frac{A}{A+B} \quad \textbf{(2.4)}$$

2.7.3. F_1 Measure

F_1 Measure disini berfungsi sebagai titik estimasi yang mana akan menyeimbangkan fungsi dari *precision* dan *recall.* Nilai *F_1Measure* akan didapatkan dengan menggunakan persamaan 2.5 [NUR-09].

$$F_1 = \frac{2.precision \cdot recall}{precision + recall} \quad \textbf{(2.5)}$$

BAB III
METODE PENELITIAN DAN PERANCANGAN

Pada bab ini menjelaskan langkah-langkah yang akan ditempuh dalam menyusun penelitian. Tahapan penelitian ini dirancang untuk membangun sistem penerapan metode *qualitative hierarchical clustering* pada data *ozone level detection data set*.

3.1. Studi Literatur

Dalam penelitian ini dibutuhkan studi literatur untuk merealisasikan tujuan dan penyelesaian masalah. Teori-teori berkaitan dengan metode *qualitative hierarchical clustering* pada data *time series* digunakan sebagai dasar penelitian yang didapat dari buku, jurnal dan *browsing* dari internet. Dari data yang diperoleh akan diolah sehingga dapat digunakan untuk analisis. Setelah dilakukan analisis maka dapat diimplementasikan ke dalam aplikasi perangkat lunak.

3.2. Data Penelitian

Untuk mengetahui bagaimana kinerja sistem rekomendasi, dibutuhkan *dataset* yang menentukan level *Ozone*. Dataset yang digunakan dalam penelitian ini diambil dari *UCI* (*www.archive.ics.UCI.edu.com*) yang diterbitkan pada tahun 1998 hingga tahun 2004.

Pada skripsi ini terdapat statistik dataset yang akan digunakan pada penelitian, dimana jumlah data keseluruhan adalah 2536 dataset, jumlah keseluruhan atribut 73, dan jumlah keseluruhan *class* ada 2 *class*. Jumlah dataset yang berjumlah 2536 data dalam skripsi ini akan di tampilkan pada lampiran 1. Berikut adalah nama-nama atribut pada data penelitian yang akan saya dimana yang akan dijelaskan pada lampiran 1.

Berikut ini adalah spesifikasi untuk beberapa atribut yang paling penting yang diakui oleh Texas Commission on Environmental Quality (TCEQ). Rincian lebih lanjut dijelaskan sebagai berikut :

1. 3 - Local ozone peak prediction.
2. Upwind - Upwind ozone background level.
3. EmFactor - Precursor emissions related factor.

4. Tmax - Maximum temperature in degrees F.
5. Tb - Base temperature where net ozone production begins (50 F).
6. SRd - Solar radiation total for the day.
7. WSa - Wind speed near sunrise (using 09-12 UTC forecast mode).
8. WSp - Wind speed mid-day (using 15-21 UTC forecast mode).

3.3. Deskripsi Umum Sistem

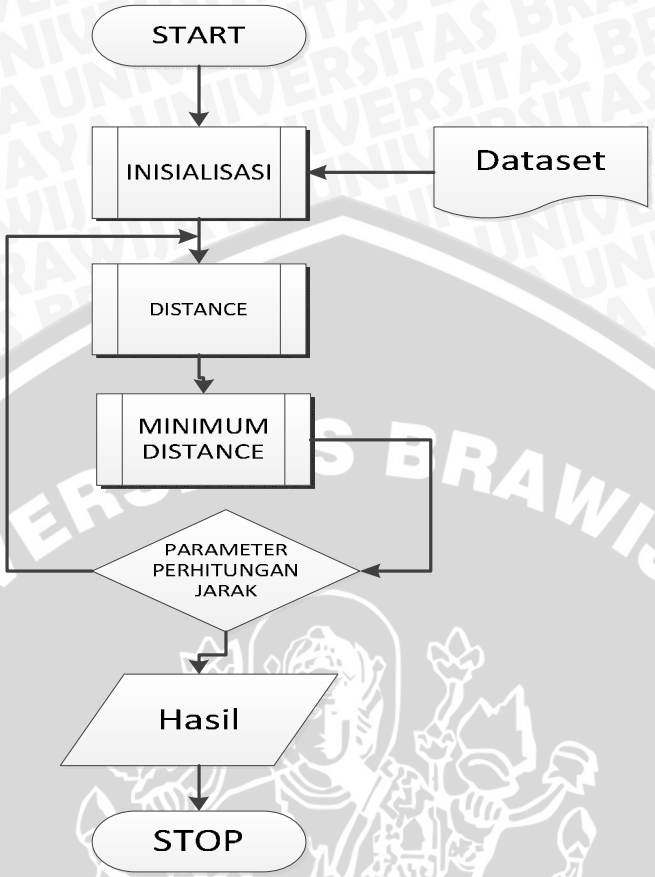
Secara umum sistem yang dibangun adalah perangkat lunak yang melakukan *clustering*. Metode yang digunakan dalam sistem ini adalah *qualitative hierarchical clustering*.

Dalam metode *qualitative hierarchical clustering* ini terdapat beberapa tahapan proses yang harus dilakukan, yaitu :

- *User* menentukan dataset yang digunakan.
- *User* menentukan parameter jumlah objek, parameter jumlah atribut dan jumlah *class* yang akan digunakan.
- Sistem menentukan jarak antara tiap-tiap data.
- Sistem menentukan jarak data yang terkecil.
- *User* mengulangi proses kedua dan ketiga hingga terbentuk *cluster* dengan jumlah dataset yang sama dengan *class*.

3.4. Perancangan Proses

Tahapan proses dalam skripsi yang menggunakan metode *qualitative hierarchical clustering* dibagi menjadi empat tahapan yaitu menentukan dataset yang akan digunakan, menentukan jarak antar data, memilih jarak data yang terkecil, dan mengulangi proses yang ke dua dan ketiga untuk mendapatkan *cluster* dengan jumlah dataset yang sama dengan *class*. Berikut perancangan proses tersebut digambarkan pada Gambar 3.1.



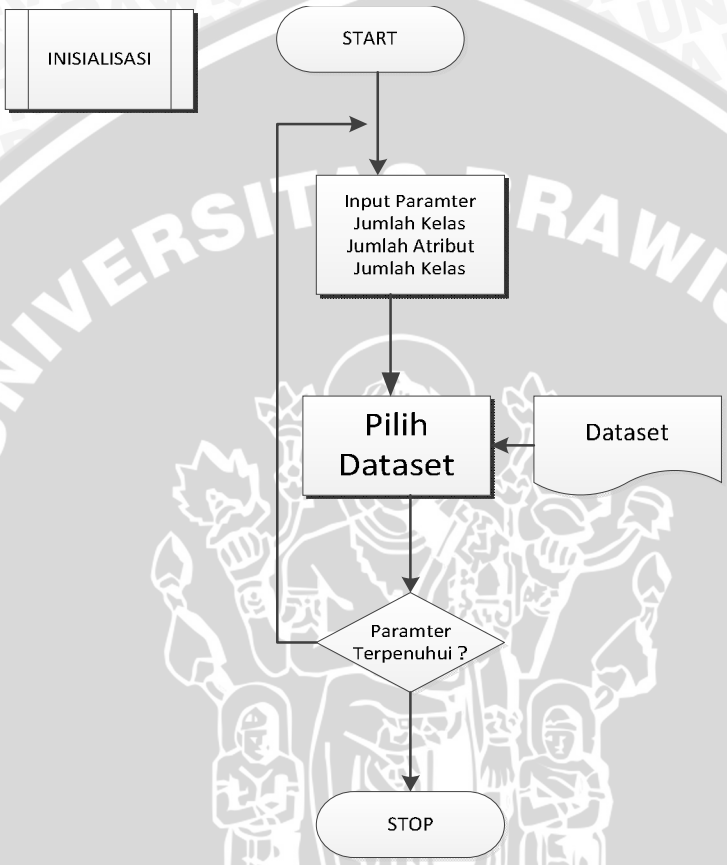
Gambar 3.1 *Flowchart* Sistem

Pada *flowchart* diatas dapat dijelaskan sebagai berikut :

- *User* menginputkan parameter dan memilih dataset yang akan diproses.
- Sistem melakukan proses inisialisasi data dan parameter dari user serta kebutuhan data pada sistem.
- Sistem melakukan proses perhitungan jarak pada masing-masing data dan kemudian membentuk sub *cluster*.
- Sistem menghitung jarak terpedek dari data yang sudah dihitung dan kemudian dilakukan pencocokan data.
- Sistem mengulangi proses pada proses perhitungan jarak dan proses akan berhenti jika jumlah *class* mencapai atau sama dengan jumlah class yang diinputakan pada parameter *user*.
- Sistem menampilkan hasil *clustering*.

3.4.1 Perancangan Proses Inisialisasi

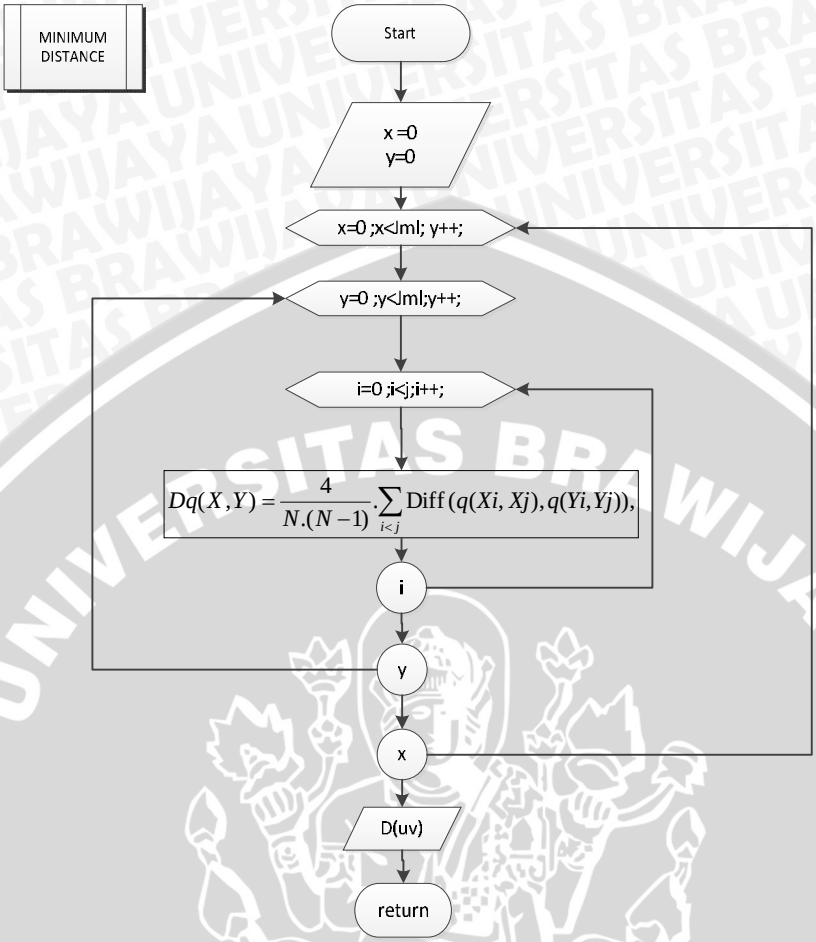
Pada proses ini dilakukan proses input pada parameter dan dataset yang akan digunakan pada proses *clustering*. Berikut ini *flowchart* dari proses inisialisasi diperlihatkan pada Gambar 3.2.



Gambar 3.2. Flowchart Proses Inisialisasi

3.4.2 Perancangan Proses Perhitungan Jarak

Pada proses ini sistem akan melakukan proses perhitungan jarak pada tiap-tiap dari *dataset* yang sudah dipilih menggunakan persamaan (2.1). Selain itu juga dilakukan proses *subcluster* yang digunakan untuk memudahkan dan mempercepat proses perhitungan pada tahap selanjutnya. Berikut ini *flowchart* dari proses perhitungan jarak diperlihatkan pada Gambar 3.3.



Gambar 3.3. *Flowchart* Proses Perhitungan Jarak.

3.5. Perhitungan Manual

Perhitungan manual adalah perhitungan dari dataset sehingga diperoleh hasil klaster yang sesuai dengan metode yang akan digunakan. Proses perhitungan manual terdapat tahap pengujian data yang sangat berguna untuk menentukan hasil klaster data. Pengujian dilakukan terhadap sekumpulan dataset. Proses pengujian klaster data yang dilakukan antara lain:

- Pengguna memasukkan dokumen dataset.
- Pengguna memasukkan parameter yang akan diujikan.
- Dilakukan perhitungan nilai jarak tiap-tiap data pada dataset sesuai dengan persamaan (2.1).
- Dari hasil perhitungan jarak akan mendapatkan nilai jarak antar data
- Dari nilai terkecil akan diambil atribut data dan akan di rata-rata

- Nilai hasil rata-rata tersebut akan dijadikan atribut baru dari data yang mempunyai nilai terkecil di nilai jarak sebelumnya.
- Proses perulangan terjadi hingga mendapatkan dua *cluster*.
- Pembentukan klaster data sebagai hasil dari dokumen dataset uji.
- Perhitungan evaluasi akan menghasilkan nilai precision, recall, dan f-measure yang didapatkan dengan menggunakan persamaan (2.3), (2.4), dan (2.5).

Dataset yang digunakan adalah dataset *time series* dari *ozone*. Berikut sampel data uji diperlihatkan pada tabel 3.2.

Tabel 3.1. Tabel Dataset Uji 1

	WSR0	WSR1	WSR2	WSR3	WSR4	Class
Data 1	1.333333333	1.857142857	5.5	2.666666667	1.1	1
Data 2	11	7	4.333333333	1	4	0
Data 3	1	7	2	1.666666667	1.6	1
Data 4	1	4.5	1.571428571	4.5	8	1
Data 5	4.333333333	3.75	4.666666667	0.75	1.125	0

Keteragan pada tabel 3.2 adalah sebagai berikut :

1. WSR0 adalah WSA - kecepatan angin di dekat matahari terbit (menggunakan 09-12 UTC modus perkiraan)
2. WSR1 adalah WSA - kecepatan angin di dekat matahari terbit (menggunakan 09-12 UTC modus perkiraan)
3. WSR2 adalah WSA - kecepatan angin di dekat matahari terbit (menggunakan 09-12 UTC modus perkiraan)
4. WSR3 adalah WSA - kecepatan angin di dekat matahari terbit (menggunakan 09-12 UTC modus perkiraan)
5. WSR4 adalah WSA - kecepatan angin di dekat matahari terbit (menggunakan 09-12 UTC modus perkiraan)

Dari Tabel 3.2 tersebut digunakan paramter jumlah *class* 2 dimana class 0 adalah normal day, dan 1 adalah ozone day.

3.5.1. Perhitungan Clustering

Pada perhitungan manual diperlukan proses perhitungan *clustering* yang digunakan untuk proses mendapatkan *cluster* dari dataset yang akan di gunakan untuk perhitungan manual. Berikut ini adalah porses perhitungan *clustering*.

	WSR0	WSR1	WSR2	WSR3	WSR4	Class
Data 1	1.333333333	1.857142857	5.5	2.666666667	1.1	1
Data 2	11	7	4.333333333	1	4	0
Data 3	1	7	2	1.666666667	1.6	1
Data 4	1	4.5	1.571428571	4.5	8	1
Data 5	4.333333333	3.75	4.666666667	0.75	1.125	0

Data 1-2	0	1	0	1	2	0.4
Data 1-3	1	1	0	1	3	0.6
Data 1-4	1	1	1	0	3	0.6
Data 1-5	0	1	1	0	2	0.4
Data 2-3	0.5	0.5	1	1	3	0.6
Data 2-4	0	0	1	1	2	0.4
Data 2-5	0	1	0	0	1	0.2
Data 3-4	0.5	0	1	0	1.5	0.3
Data 3-5	1	1	1	0	3	0.6
Data 4-5	1	1	1	0	3	0.6

Nilai Jarak I	Data 1	Data 2	Data 3	Data 4	Data 5
Data 1					
Data 2	0.4				
Data 3	0.6	0.6			
Data 4	0.6	0.4	0.3		
Data 5	0.4	0.2	0.6	0.6	

Nilai Atribut Data 2	11	7	4.333333333	1	4
Nilai Atribut Data 5	4.333333333	3.75	4.666666667	0.75	1.125
Nilai Rata - Rata	7.666666667	5.375	4.5	0.875	2.5625

	Atribut 1	Atribut 2	Atribut 3	Atribut 4	Atribut 5
Data 1	1.333333333	1.857142857	5.5	2.666666667	1.1
Data 2 & 5	7.666666667	5.375	4.5	0.875	2.5625
Data 3	1	7	2	1.666666667	1.6
Data 4	1	4.5	1.571428571	4.5	8

Data 1 - Data 2 & 5	0	1	0	1	2	0.4
Data 1 - Data 3	1	1	0	1	3	0.6
Data 1 - Data 4	1	1	1	0	3	0.6
Data 2 & 5 - Data 3	1	1	1	1	4	0.8
Data 2 & 5 - Data 4	0	0	1	0	1	0.2
Data 3 - Data 4	0.5	0	1	0	1.5	0.3

Nilai Jarak II	Data 1	Data 2 & 5	Data 3	Data 4
Data 1				
Data 2 & 5	0.4			
Data 3	0.6	0.8		
Data 4	0.6	0.2	0.3	

Nilai Atribut Data 2 & 5	7.666666667	5.375	4.5	0.875	2.5625
Nilai Atribut Data 4	1	4.5	1.571428571	4.5	8
Nilai Rata - Rata	4.333333333	4.9375	3.035714286	2.6875	5.28125

	Atribut 1	Atribut 2	Atribut 3	Atribut 4	Atribut 5
Data 1	1.333333333	1.857142857	5.5	2.666666667	1.1
Data 2 & 5 & 4	4.333333333	4.9375	3.035714286	2.6875	5.28125
Data 3	1	7	2	1.666666667	1.6

Data 1 - Data 2, 5, 4	0	1	1	0	2	0.4
Data 1 - Data 3	1	1	0	1	3	0.6
Data 3 - Data 2, 5 & 4	1	1	0	1	3	0.6

Nilai Jarak IV	Data 1	Data 2 & 5 & 4	Data 3
Data 1			
Data 2 & 5 & 4	0.4		
Data 3	0.6	0.6	

Nilai Atribut Data 1	1.333333333	1.857142857	5.5	2.666666667	1.1
Nilai Atribut Data 2 & 5 & 4	4.333333333	4.9375	3.035714286	2.6875	5.28125
Nilai Rata - Rata	2.833333333	3.397321429	4.267857143	2.677083334	3.190625

	Atribut 1	Atribut 2	Atribut 3	Atribut 4	Atribut 5
Data 1 & 2 & 5 & 4	4.333333333	4.9375	3.035714286	2.6875	5.28125
Data 3	1	7	2	1.666666667	1.6

Data 1 & 2 & 5 & 4 dengan Data 3	1	1	0	0	2	0.4

Nilai Jarak V	Data 1 & 2 & 5 & 4	Data 3
Data 1 & 2 & 5 & 4		
Data 3	0.4	

3.5.1. Perhitungan Manual Evaluasi

Pada proses perhitungan evaluasi akan menghasilkan nilai precision, recall, dan f-measure yang akan digunakan untuk mengetahui kelayakan dari metode yang akan dipakai. Berikut adalah perhitungan manual evaluasi.

True Positif

0 : 3

1 : 2

False Negatif

Untuk a = 0 Nilai = 0+ = 0

Untuk a =1 Nilai = 0 + = 0

False Positif

Untuk a =0 Nilai = 0 + = 0

Untuk a =1 Nilai = 0 + = 0

Perhitungan Masing-masing data :

Recall [0] = 3 / (3 + 0) = 1

Precision [0] = 3 / (3 + 0) = 1

F-Measure [0] = 2 * 1 * 1 / (1 + 1) = 1

Recall [1] = 2 / (2 + 0) = 1

Precision [1] = 2 / (2 + 0) = 1

F-Measure [1] = 2 * 1 * 1 / (1 + 1) = 1

Perhitungan total masing-masing evaluasi :

Precision total = 1+ 1

Recall total = 1+ 1

F-Measure total = 1+ 1

Dari hasil total masing-masing diperoleh :

F-Measure = 2/2 = 1

Precision = 2/2 = 1

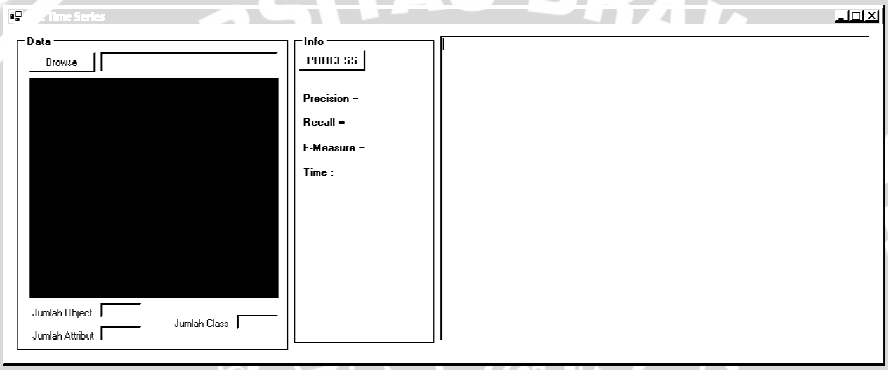
Recall = 2/2 = 1

3.6. Rancangan Uji Coba

Rancangan uji coba meliputi desain antarmuka dan hasil evaluasi. Dalam sub bab ini akan dijelaskan bagaimana bentuk dari sistem yang akan dibuat dan bagaimana tingkat akurasi yang dihasilkan oleh sistem nantinya.

3.6.1. Rancangan Antarmuka

Antarmuka adalah hal pertama yang perlu diperhatikan dalam membuat sebuah sistem aplikasi, karena antarmuka inilah yang menghubungkan langsung dengan *user*. pada Gambar 3.6 berikut adalah desain antarmuka dari penerapan metode *qualitative hierarchical clustering* pada data *time series.*



Gambar 3.5. Rancangan Antar Muka Sistem

Pada gambar perancangan antar muka sistem yang ditunjukkan pada Gambar 3.6, terdapat beberapa tombol yang masing- masing memiliki fungsi yang berbeda-beda yakni: pada tombol *browse* digunakan untuk mengambil dokumen dataset yang akan diuji. Pada tombol proses digunakan untuk memulai proses pengujian *dataset*, sedangkan pada bagian informasi berfungsi untuk memberikan informasi tetang akurasi dari hasil pengujian.

3.7. Skenario Pengujian

Hasil *clustering* dataset oleh sistem dibandingkan dengan tiga kali pengujian hasil dari dataset yang berjumlah 100, 500, dan 1000, pada setiap pengujian akan menggunakan data random sesuai jumlah data yang akan di ujikan. Dimana hal ini bertujuan untuk mengetahui hasil akurasi, dan efisiensi waktu dari sistem.

BAB IV
IMPLEMENTASI

Bab ini menjelaskan implementasi dari perancangan pada bab sebelumnya. Implementasi ini meliputi implementasi perangkat keras dan implementasi perangkat lunak.

4.1. Implementasi Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem adalah sebagai berikut :

1. *Processor* : Core i7- 3970X , 3.5Ghz
2. *Memorry RAM* : 16 Gb
3. *Harddisk* : 2 Tera

4.2. Implementasi Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem ini adalah :

1. Sistem Operasi : Microsoft Windows 7 Sp.1 64 bit
2. Bahasa Pemrograman : Microsoft Visual Studio Express 2010

4.3. Implementasi Proses

Berdasarkan analisa dan perancangan proses yang terdapat dalam bab 3, maka pada sub bab ini akan dijelaskan implementasi proses-proses tersebut.

4.3.1. Implementasi Proses Inisialisasi

Pada proses ini dilakukan proses input pada parameter dan *dataset* yang akan digunakan pada proses *clustering*.

```
DialogResult file = openFile.ShowDialog();
if (file == DialogResult.OK)
{
List<string[]> parsedData = newList<string[]>();
try
        {
using (StreamReader readFile = newStreamReader(openFile.FileName))
                {
while ((line = readFile.ReadLine()) != null)
                    {
row = line.Split(',');
parsedData.Add(row);
                    }
                }
         }
```

```
catch (Exception er)
        {
MessageBox.Show(er.Message);
        }
dataset = parsedData.ToArray();
DataTable testTable = newDataTable();
foreach (string column in dataset[0])
        {
testTable.Columns.Add();
attribute++;
        }
foreach (string[] r in dataset)
        {
        testTable.Rows.Add(r);
        record++;
        }
        DGset.DataSource = testTable;
        }
        textBoxDataset.Text = openFile.FileName;
        textBox2.Text = Convert.ToString(record);
        textBox3.Text = Convert.ToString(attribute);
cl = newstring[record];
for (int a = 0; a < record; a++)
           {
           cl[a] = dataset[a][attribute - 1];
if (!kelas.Contains(cl[a]))
               {
               kelas.Add(cl[a]);
               }
           }
k = kelas.Count;
textBox4.Text = Convert.ToString(kelas.Count);
```

Source Code 4.1. Proses Inisialisasi

4.3.2. Implementasi Proses Perhitungan Jarak

Pada proses ini sistem akan melukukan prose perhitungan jarak pada tiap-tiap dari *dataset* yang sudah dipilih menggunakan persamaan (2.1). Selain itu juga dilakukan proses *subcluster* yang digunakan untuk memudahkan dan mempercepat proses perhitungan pada tahap selanjutnya.

```
int[] ixCl = newint[record];
for (int a = 0; a < k; a++)
{
for (int b = 0; b < cluster[a].Count; b++)
      {
      ixCl[cluster[a][b]] = a;
      }
}
richTextBox1.Text += "\n";
for (int a = 0; a < record; a++)
{
```

```
richTextBox1.Text += ixCl[a] + "\n";
}

#region precision, recall, F-Measure
evaluation eval = newevaluation();
List<evaluation> dataEval = newList<evaluation>();
for (int a = 0; a < record; a++)
{
eval.recordord = a;
eval.cluster = ixCl[a];
eval.kelas = Convert.ToString(ddataset[a][attribute-1]);
dataEval.Add(eval);
}
List<string> classList = newList<string>();
for (int a = 0; a < record; a++)
{
if (!classList.Contains(dataEval[a].kelas))
       {
        classList.Add(dataEval[a].kelas);
       }
}
int[][] dataM = newint[k][];
for (int a = 0; a < k; a++)
{
dataM[a] = newint[classList.Count];
}

for (int a = 0; a < record; a++)
{
for (int b = 0; b < k; b++)
       {
for (int c = 0; c < classList.Count; c++)
              {
if ((Convert.ToString(ddataset[a][attribute-1]) == classList[c]) &&
(dataEval[a].cluster == b))
                        {
                         dataM[b][c]++;
                         break;
                        }
              }
        }
}
```

Source Code 4.2. Proses Perhitungan Jarak

4.3.3. Implementasi Proses Perhitungan Jarak Minimum

Dari hasil proses perhitungan jarak pada masing-masing data di dataset, kemudian dilakukan proses lagi yakni melakukan perhitungan jarak minimum dari masing-masing data yang telah di*cluster* dengan menggunakan persamaan (2.2). Hal ini bertujuan untuk mendapatkan nilai-nilai yang paling mendekati atau mirip dari tiap-tiap klaster yang terbentuk. Selain itu pada proses ini juga dilakukan

pengecekan pada jumlah *class* yang telah terbentuk, jika telah memenuhi atau sama dengan parameter input maka proses akan berhenti, dan sebaliknya jika tidak maka mengulai proses perhitungan jarak lagi.

```
ddataset = newdouble[record][];
for (int a = 0; a < record; a++)
{
ddataset[a] = newdouble[attribute];
}
for (int a = 0; a < record; a++)
{
for (int b = 0; b < attribute; b++)
      {
if (dataset[a][b] == "?")
           {
            ddataset[a][b] = 0;
           }
else
           {
            ddataset[a][b] = Convert.ToDouble(dataset[a][b]);
           }
      }
}

List<List<double>> dlDataset = newList<List<double>>();
for (int a = 0; a < record; a++)
{
 List<double> subdl = newList<double>();
 for (int b = 0; b < attribute; b++)
      {
       subdl.Add(Math.Round(ddataset[a][b], 3));
      }
dlDataset.Add(subdl);
}

List<List<int>> cluster = newList<List<int>>();
for (int a = 0; a < record; a++)
{
List<int> subcluster = newList<int>();
subcluster.Add(a);
cluster.Add(subcluster);
}
double[][] distance = newdouble[record][];
for (int a = 0; a < record; a++)
{
distance[a] = newdouble[record];
}
for (int a = 0; a < record; a++)
{
Console.WriteLine("Data Ke-"+a.ToString());
for (int b = 0; b < a; b++)
      {
double temp = 0;
Console.WriteLine(" Jarak ke-" + b.ToString());
for (int c = 0; c < attribute; c++)
           {
if (dlDataset[a][c] < dlDataset[b][c])
                {
```

```
                temp += 0;
               }
elseif (dlDataset[a][c] > dlDataset[b][c])
               {
                temp += 1;
               }
elseif (dlDataset[a][c] == dlDataset[b][c])
               {
                temp += 0.5;
               }
Console.Write("   +" + temp.ToString()+"\t");
                    }
Console.WriteLine(" = "+ temp);
distance[a][b] = temp;
                        }
                            }
                            richTextBox1.Text += "JARAK : \n";
                            for (int a = 0; a < cluster.Count; a++)
                            {
for (int b = 0; b < attribute; b++)
                              {
                               richTextBox1.Text += dlDataset[a][b] + "\t";
                              }
                               richTextBox1.Text += "\n";
                    }
do
                   {
int pair1 = 0, pair2 = 0;
double min = 99999;
Console.WriteLine(" Jarak Minimum ");
for (int a = 0; a < cluster.Count; a++)
                    {
Console.Write(" Jarak Cluster ke-" + a.ToString());
for (int b = 0; b < a; b++)
                       {
Console.WriteLine();
Console.Write(" \tdata-"+b.ToString());
Console.Write(" \t\tJarak=" + distance[a][b].ToString());
if (distance[a][b] < min)
                       {
                        min = distance[a][b];
                        pair1 = a;
                        pair2 = b;
Console.WriteLine();
Console.Write(" \t\tJarak Minimum = " + min.ToString());
Console.Write(" \t\tSehingga data ke-" + b.ToString()+"
cluster="+a.ToString());
                       }
Console.WriteLine();
                    }
Console.WriteLine();
                }
                richTextBox1.Text += "Pair 1 = " + pair1 + "\n";
                richTextBox1.Text += "Pair 2 = " + pair2 + "\n";
                if (pair1 < pair2)
                {
                 cluster[pair1].AddRange(cluster[pair2]);
                 cluster.RemoveAt(pair2);
                 for (int a = 0; a < attribute; a++)
                 {
```

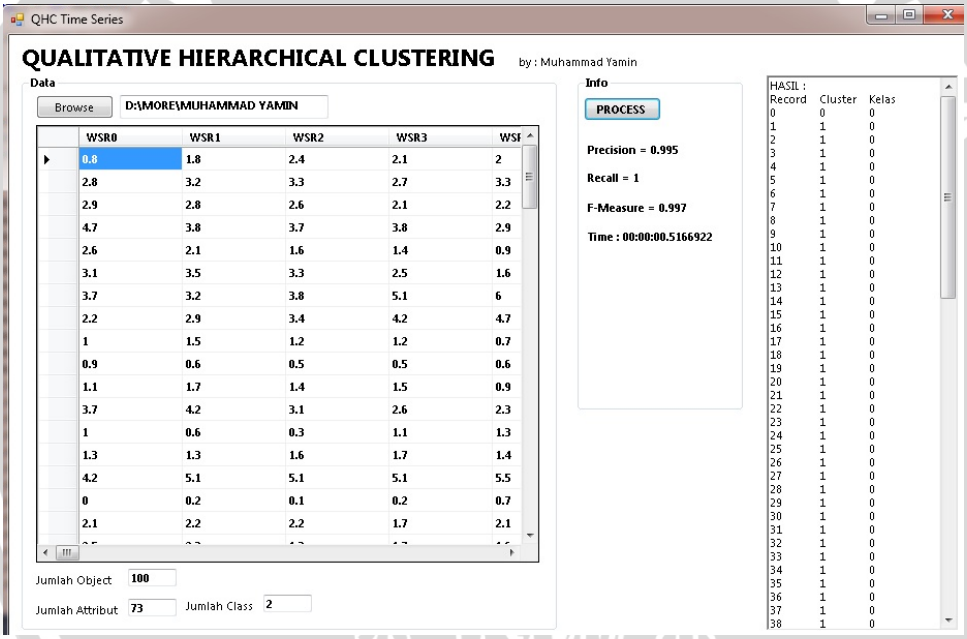
```
double temp = 0;
for (int c = 0; c < cluster[pair1].Count; c++)
                {
                 temp += ddataset[cluster[pair1][c]][a];
                }
dlDataset[pair1].Add(Math.Round(temp / cluster[pair1].Count, 3));
            }
          }
else
          {
           cluster[pair2].AddRange(cluster[pair1]);
           cluster.RemoveAt(pair1);
for (int a = 0; a < attribute; a++)
            {
double temp = 0;
for (int c = 0; c < cluster[pair2].Count; c++)
                {
                 temp += ddataset[cluster[pair2][c]][a];
                }
                 dlDataset[pair2].Add(Math.Round(temp /
cluster[pair2].Count, 3));
            }
for (int a = 0; a < cluster.Count; a++)
            {
for (int b = 0; b < attribute; b++)
                {
                 richTextBox1.Text += dlDataset[a][b] + "\t";
                }
                 richTextBox1.Text += "\n";
            }
distance = newdouble[cluster.Count][];
for (int a = 0; a < cluster.Count; a++)
            {
             distance[a] = newdouble[cluster.Count];
            }
for (int a = 0; a < cluster.Count; a++)
            {
for (int b = 0; b < a; b++)
              {
double temp = 0;
for (int c = 0; c < attribute; c++)
                {
if (dlDataset[a][c] < dlDataset[b][c])
                    {
                    temp += 0;
                    }
elseif (dlDataset[a][c] > dlDataset[b][c])
                    {
                     temp += 1;
                    }
elseif (dlDataset[a][c] == dlDataset[b][c])
                    {
                     temp += 0.5;
                    }
                }
                 distance[a][b] = temp;
                }
            }
richTextBox1.Text += "\n";
for (int a = 0; a < cluster.Count; a++)
```

```
                {
for (int b = 0; b < cluster.Count; b++)
                {
                 richTextBox1.Text += distance[a][b] + "\t";
                }
                 richTextBox1.Text += "\n";
                }
            }
while (cluster.Count > k);
```

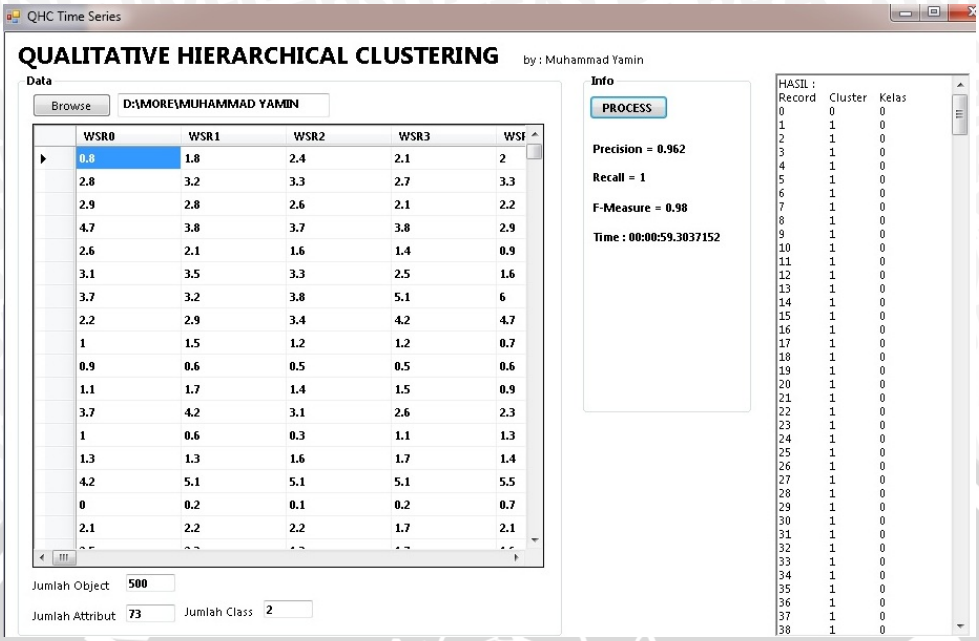
Source Code 4.3. Proses Perhitungan Jarak Minimum

4.4. Implementasi Antarmuka

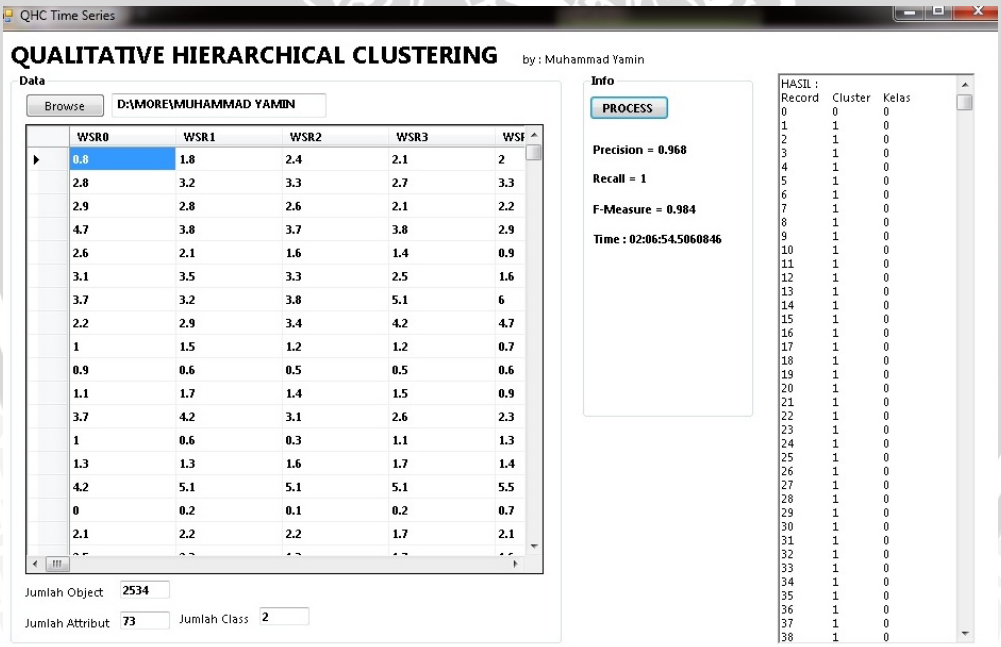
Seperti yang telah dijelaskan pada perancangan pada bab 3, halaman atarmuka digunakan sebagai media interaksi atar user dan sistem. Halaman antar muka diperlihatkan pada Gambar 4.4, 4.5, dan 4.6.



Gambar 4.4. Tampilan Program Dengan Jumlah Data 100



Gambar 4.5. Tampilan Program Dengan Jumlah Data 500



Gambar 4.6. Tampilan Program Dengan Jumlah Data 2536

Pada halaman atarmuka terdapat output dari program skripsi ini, output di Gambar 4.4, 4.5, 4.6 ini terdapat informasi tentang jumlah object, jumlah atribut, jumlah *class*, dimana yang berfungsi untuk mengetahui jumlah dari masing-

masing info dataset. Ada beberapa tombol navigasi, yakni tombol *browse* dan tombol proses, tombol *browse* digunakan utuk mengambil *dataset* dari berkas yang tersedia, kemudian sistem secara otmomatis menampilkan atribut dan *item* dari dataset yang dipilih. Kemudian tombol *process* digunakan untuk memproses klasifikasi, pada tombol proses dilakukan proses perhitungan sesuai dengan perancangan pada bab 3. Setelah proses perhitungan selesai sistem lakukan, kemudian sistem menampilkan hasil perhitungan dan hasil analisa pada halaman antarmuka.

BAB V
PENGUJIAN DAN ANALISIS

5.1. Bahan Pengujian

Untuk melakukan pengujian diperlukan data *time series* inputan, yakni dataset *ozone level detection data set*. Dataset yang digunakan dalam penelitian ini diambil dari *UCI* (www.archive.ics.UCI.edu.com) yang diterbitkan pada tahun 1998. Data yang akan digunakan dalam 3 kali pengujian berjumlah 100, 500, dan 1000 sebanyak 5 file dengan record data yang berbeda sesuai jumlah file yang akan dijadikan bahan percobaan, data yang akan digunakan adalah dataset yang diambil dari dataset yang berjumlah 2536. Data yang diambil pada percobaan ini adalah data random yang record datanya sudah di tentukan oleh pengguna. Keterangan dataset pengujian dapat dilihat pada Tabel 5.1.

Tabel 5.1 Tabel Pengujian

No.	Banyak Data	Banyak Atribut	Banyak Class	Banyak Percobaan
1	100	73	2	5 Kali
2	500	73	2	5 Kali
3	1000	73	2	5 Kali

5.2. Skenario Pengujian

Sesuai dengan rancangan pengujian pada bab 3, proses pengujian dilakukan dengan melakukkan pengujian sebanyak 5 kali pada masing-masing dataset pada Tabel 5.1, setiap pengujian akan menggunakan data random sesuai banyak data yang diujikan. Kemudian dari hasil percobaan didapatkan nilai akurasinya yakni *precision*, *recall* dan *f-measure*. Dimana nilai precision akan menghasilkan nilai titik estimasi yang mana akan menyeimbangkan fungsi dari *precision* dan *recall*, nilai recall akan menghasilkan nilai perbandingan antara jumlah dataset yang benar/relevan yang ditemukan oleh sistem (A) dengan jumlah dataset pada kategori, dan nilai f-measure akan menghasilkan nilai kemiripan dari hasil kategorisasi suatu dataset. Pada proses perhitungan analisis recall, precision, f-measure akan menggunakan nilai inputan dari *class* asli pada dataset pengujian

yang sudah ditentukan dari dataset *ozone level detection dataset*. Dari hasil evaluasi yang akan diketahui kelayakan dari metode *Hierarchical Clustering*. Untuk mengambil kesimpulan, dari hasil analisa terebut dibuat sebuah grafik *precision*, *recall* dan *f-measure* dan *time*.

5.3. Pengujian dan Analisis Hasil

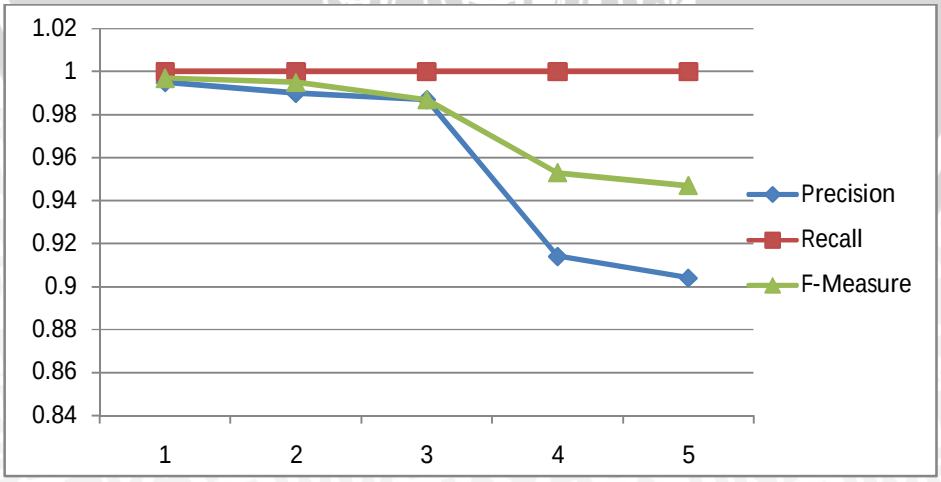
Pada pengujian sekenario, dilakukan implementasi metode yang akan digunakan dalam *Data time series*, *Hierarchical Clustering*, dan metode *Qualitative*.

1. Skenario Pengujian Pertama

Pada Skenario ini akan dilakukan pengujian sebnayak 5 kali pada 100 data yang diambil secara random, dimana pada pengujian ini dilakukan sebanyak 5 kali proses pengujian dengan menggunakan data yang berbeda-beda. Hasil dari pengujian data pertama diperlihatkan pada Tabel 5.2.

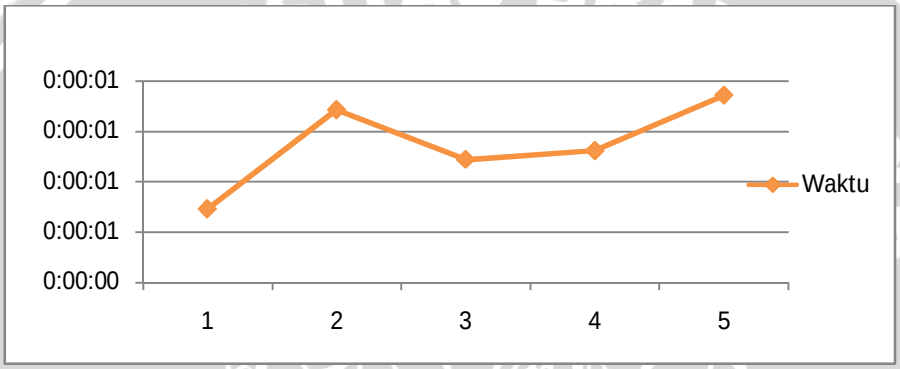
Tabel 5.2 Tabel Hasil Pengujian Data pertama

Percobaan	precision	recall	measure	Time (detik)
1	0.995	1	0.997	00:00:00.5086731
2	0.99	1	0.995	00:00:00.5431960
3	0.987	1	0.987	00:00:00.5256921
4	0.914	1	0.953	00:00:00.5293056
5	0.904	1	0.947	00:00:00.5477105
Nilai Rata rata	0.958	1	0.9758	



Gambar 5.1. Grafik Hasil Pengujian Data Pertama

Dari gambar grafik Gambar 5.1, memberikan informasi bahwa nilai rata-rata *precision* dari data percobaan pertama adalah 0,958 dimana nilai ini merupakan nilai kemiripan dari katergori yang ada pada dokumen dibandingkan dengan nilai hasil proses pengujian dari perangkat lunak. Kemudian nilai recall adalah 1 dimana nilai ini meberikan informasi bahwa hasil perbandingan antara dataset uji dan datahasil pengujian, memiliki relevansi yang sama atau benar. Sedangkan nilai *F-Measure* adalah 0,97 dimana nilai ini memberikan informasi mengenai estimasi antara hasil dari *precision* dan *recall*. Untuk waktu komputasi dari masing-masing percobaan pada data pertama diperlihatkan pada Gambar 5.2.



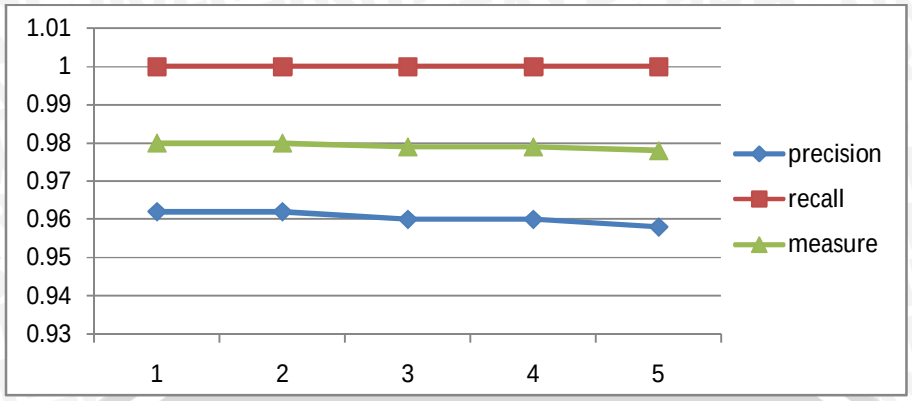
Gambar 5.2. Grafik Waktu Hasil Pengujian Data Pertama

2. Pengujian data kedua

Pada Skenario ini akan dilakukan pengujian sebnayak 5 kali pada 500 data yang diambil secara random, dimana pada pengujian ini dilakukan sebanyak 5 kali proses pengujian dengan menggunakan data yang berbeda-beda. Hasil dari pengujian data pertama diperlihatkan pada Tabel 5.3.

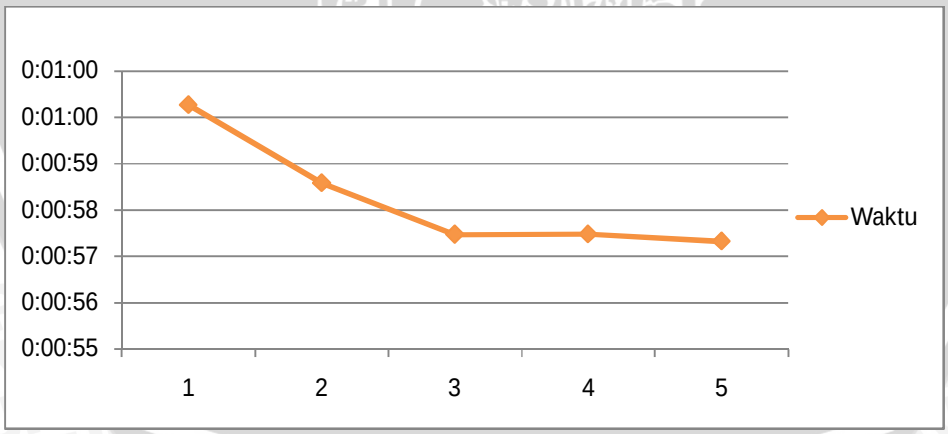
Tabel 5.3. Tabel Hasil Pengujian Data Kedua

Percobaan	Precision	Recall	F-Measure	Time (detik)
1	0.962	1	0.98	00:00:59.8497320
2	0.962	1	0.98	00:00:58.3929579
3	0.96	1	0.979	00:00:57.4334418
4	0.96	1	0.979	00:00:57.4396502
5	0.958	1	0.978	00:00:57.3070044
Rata-rata	0.9604	1	0.9792	



Gambar 5.3. Grafik Hasil Pengujian Data Kedua

Dari gambar grafik Gambar 5.3, memberikan informasi bahwa nilai *precision* dari data percobaan kedua adalah 0,9604 dimana nilai ini merupakan nilai kemiripan dari katergori yang ada pada dokumen dibandingkan dengan nilai hasil proses pengujian dari perangkat lunak. Kemudian nilai *recall* adalah 1 dimana nilai ini meberikan informasi bahwa hasil perbandingan antara dataset uji dan datahasil pengujian, memiliki relevansi yang sama atau benar. Sedangkan nilain *F-Measure* adalah 0.9792 dimana nilai ini memberikan informasi mengenai estimasi antara hasil dari *precision* dan recall. Untuk waktu komputasi dari masing-masing percobaan pada data kedua diperlihatkan pada Gambar 5.4.



Gambar 5.4. Grafik Waktu Hasil Pengujian Data Kedua

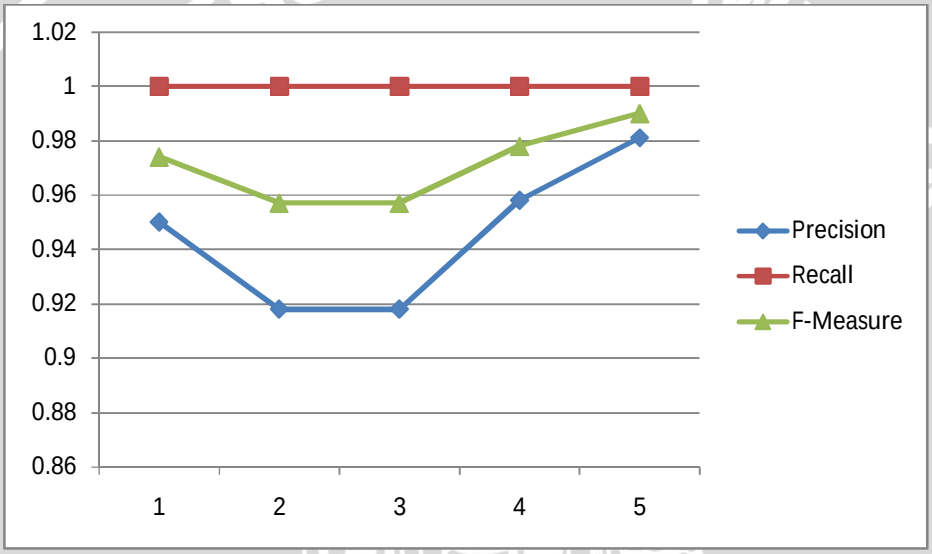
3. Pengujian data ketiga

Pada Skenario ini akan dilakukan pengujian sebnayak 5 kali pada 1000 data yang diambil secara random, dimana pada pengujian ini dilakukan sebanyak

5 kali proses pengujian dengan menggunakan data yang berbeda-beda. Hasil dari pengujian data pertama diperlihatkan pada Tabel 5.3.

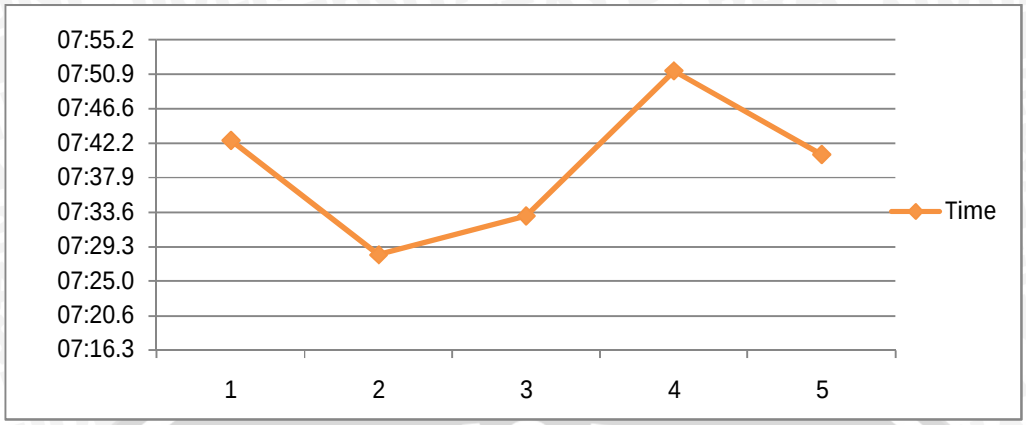
Tabel 5.4. Tabel Hasil Pengujian Data ke-Tiga

Percobaan	Precision	Recall	F-Measure	Time (detik)
1	0.95	1	0.974	00:07:42.5844881
2	0.918	1	0.957	00:07:28.2686134
3	0.918	1	0.957	00:07:33.1245412
4	0.958	1	0.978	00:07:51.2813852
5	0.981	1	0.99	00:07:40.8087161
Rata-rata	0.945	1	0.9712	



Gambar 5.5. Grafik Hasil Pengujian Data Ke Tiga

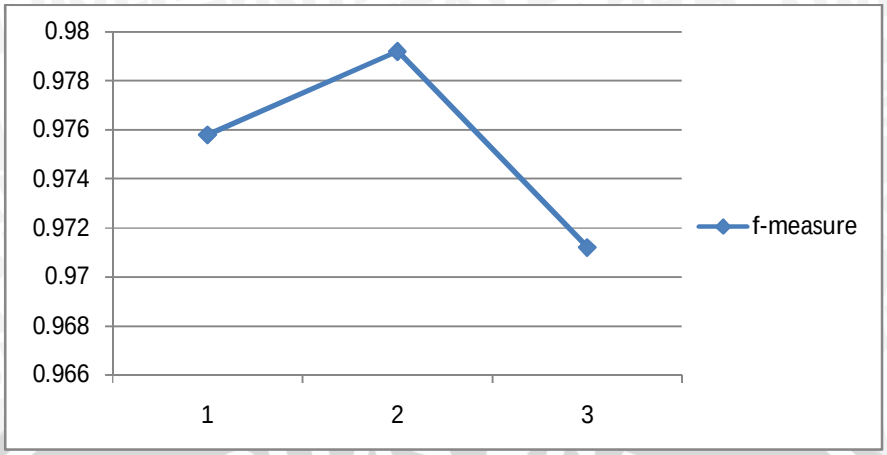
Dari gambar grafik Gambar 5.5, memberikan informasi bahwa nilai *precision* dari data percobaan ke tiga adalah 0,945 dimana nilai ini merupakan nilai kemiripan dari katergori yang ada pada dokumen dibandingkan dengan nilai hasil proses pengujian dari perangkat lunak. Kemudian nilai *recall* adalah 1 dimana nilai ini meberikan informasi bahwa hasil perbandingan antara dataset uji dan data hasil pengujian, memiliki relevansi yang sama atau benar. Sedangkan nilain *F-Measure* adalah 0,9712 dimana nilai ini memberikan informasi mengenai estimasi antara hasil dari *precision* dan *recall.* Untuk waktu komputasi dari masing-masing percobaan pada data ke tiga diperlihatkan pada Gambar 5.6.



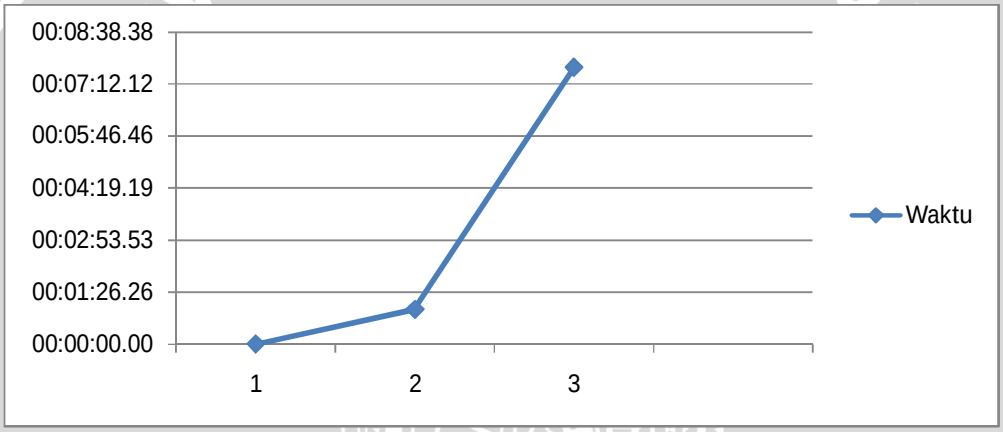
Gambar 5.6. Grafik Waktu Hasil Pengujian Data Ke Tiga

5.4. Kesimpulan Pengujian

Pada kesmipulan ini, hasil dari proses pengujian data sesuai dengan skenario pada Tabel 5.1 dari masing-masing percobaan pada data, memberikan informasi bahwa nilai rata-rata *precision* dari masing-masing percobaan adalah pada percobaan pertama adalah 0,958 pada percobaan kedua adalah 0.9604 dan pada percobaan ke tiga adalah 0,945 dimana nilai ini merupakan nilai kemiripan dari katergori yang ada pada dokumen dibandingkan dengan nilai hasil proses pengujian dari perangkat lunak. Kemudian nilai *recall* adalah 1 dimana nilai ini meberikan informasi bahwa hasil perbandingan antara dataset uji dan data hasil pengujian, memiliki relevansi yang sama atau benar. Sedangkan nilai *f-measure* adalah pada percobaan pertama adalah 0,9758 pada percobaan kedua adalah 0.9792 dan pada percobaan ketiga adalah 0,9712. Pada Skenario pengujian ke dua mendapatkan *f-measure* paling tinggi dengan nilai 0.9792 yang berarti pada pengujian ke dua ini memiliki tingkat akurasi yang baik. Dari rata-rata di *f-*measure tiga kali percobaan di atas adalah 0.9754 dimana nilai ini memberikan informasi mengenai estimasi antara hasil dari *precision* dan *recall*. Sehingga dari nilai-nilai tersebut, bahwa nilai relevansi atau kebenaran dari hasil analisis perangkat lunak, menunjukkan tingkat akurasi yang cukup baik, seperti yang di tunjukan pada Gambar 5.7. Dan dari grafik waktu pada masing-masing percobaan disimpulkan bahwa semakin banyak data yang digunakan maka semakin besar pula waktu komputasi yang digunakan yang seperti di tunjukan pada Gambar 5.8.



Gambar 5.7 Grafik Rata-rata F-Measure



Gambar 5.8 Grafik Rata-rata Waktu proses

BAB VI
KESIMPULAN DAN SARAN

Bab VI Penutup berisi kesimpulan dari hasil implementasi dan analisis sistem serta saran untuk pengembangan sistem.

6.1. Kesimpulan

Dari hasil implementasi, pengujian, dan analisis metode *qualitative hierarchical clustering* pada data *time series* dapat disimpulkan bahwa:

1. Sistem perangkat lunak yang dibangun dengan algoritma pada *Data time series*, *Hierarchical Clustering*, dan metode *Qualitative* dapat melakukan proses *clustering* pada data *time series*.
2. Tingkat akurasi dari metode *hierarchical clustering* dapat dilihat dari nilai rata-rata akurasi *f-measure* yang di dapat. Akurasi dari masing-masing percobaan yang memiliki nilai *f-measure* paling tinggi terdapat pada scenario ke dua dimana nilai *f-measure* yang di dapat adalah 0.9792. Semakin besar data yang diujikan menyebabkan waktu komputasi yang digunakan semakin lama.

6.2. Saran

Saran untuk pengembangan dan analisis metode *qualitative hierarchical clustering* pada data *time series*, adalah:

1. Perlunya pengembangan metode *qualitative hierarchical clustering* pada data *time series*, terhadapdata yang lebih komplek.
2. Perlunya pengembangan bahasa pemrograman agar diperoleh waktu komputasi yang lebih cepat pada data dengan jumlah yang besar.

DAFTAR PUSTAKA

[FAD-11] Fadli, A. 2011 Konsep Data Mining, *http://ilmukomputer.org/wp-content/uploads/2011/03/Ari Fadli Konsep Data Mining.pdf*. Tanggal akses : 8 April 2012

[HAN -06] Han, J. 2006. *Data Mining Concepts and techniques Second Edition*.University of Illinois, Illinois, United States.

[LJU-02] Ljubic, P., Todorovski, and Lavrac, N. 2002. *Time-Series Analysis Of UK Traffic Accident Data*. Ljubljana, Slovenia.

[NUR-09] Nur, D. 2009. Penerapan Algoritma *Improved K-Nearest Neighbors* Untuk Pengkategorian Dokumen Teks Berita Berbahasa Indonesia. Universitas Brawijaya. Malang

[SAN-13] Sanwlani, M, and Vijayalakshmi, M., 2013. *Forecasting Sale Through Time Series Clustering*. Mumbai, Maharashtra.

[SAN-07] Santosa, B. 2007. Data Mining: Teknik Pemanfaatan Data Untuk Keperluan Bisnis, Graha Ilmu, Yogyakarta.

[WES-98] West, B., Peter, M., and Michael, R. 1998. *Ozone Layer*.

LEMBAR LAMPIRAN

Lampiran 1. Nama Atribut

No	Nama Atribut	No	Nama Atribut	No	Nama Atribut	No	Nama Atribut
1	WSR0	37	T10	19	WSR18	55	U85
2	WSR1	38	T11	20	WSR19	56	V85
3	WSR2	39	T12	21	WSR20	57	HT85
4	WSR3	40	T13	22	WSR21	58	T70
5	WSR4	41	T14	23	WSR22	59	RH70
6	WSR5	42	T15	24	WSR23	60	U70
7	WSR6	43	T16	25	WSR_PK	61	V70
8	WSR7	44	T17	26	WSR_AV	62	HT70
9	WSR8	45	T18	27	T0	63	T50
10	WSR9	46	T19	28	T1	64	RH50
11	WSR10	47	T20	29	T2	65	U50
12	WSR11	48	T21	30	T3	66	V50
13	WSR12	49	T22	31	T4	67	HT50
14	WSR13	50	T23	32	T5	68	KI.
15	WSR14	51	T_PK	33	T6	69	TT
16	WSR15	52	T_AV	34	T7	70	SLP
17	WSR16	53	T85	35	T8	71	SLP_
18	WSR17	54	RH85	36	T9	72	Precp