



TUGAS AKHIR - KI091391

Analisis Kinerja Algoritma *Clustering* dengan Metode *Evidence Accumulation*

Ratih Kirana Diantari
NRP. 5110 100 051

Dosen Pembimbing 1
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.

Dosen Pembimbing 2
Victor Hariadi, S.Si., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2014



FINAL PROJECT - KI091391

Analysis Performance Clustering Algorithm with Evidence Accumulation Method

Ratih Kirana Diantari
NRP. 5110 100 051

Advisor 1
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.

Advisor 2
Victor Hariadi, S.Si., M.Kom.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2014

Analisis Kinerja Algoritma *Clustering* dengan Metode *Evidence Accumulation*

Nama : Ratih Kirana Diantari
NRP : 5110100051
Jurusan : Teknik Informatika – FTIf ITS
Dosen Pembimbing I : Dr. Eng. Chastine Fatichah, S.Kom.,
M.Kom.
Dosen Pembimbing II : Victor Hariadi, S.Si., M.Kom.

Abstrak

Analisis cluster merupakan salah satu prosedur dalam proses penggalan data. Informasi tentang distribusi dan karakteristik data bisa diperoleh melalui proses clustering. Clustering telah digunakan dalam berbagai bidang, seperti segmentasi pasar, pengelompokan dokumen, pengenalan pola, dan analisis data spatial.

Pada Tugas Akhir ini mengimplementasikan metode clustering berbasis Consensus Clustering & Evidence Accumulation (EA). Pada metode Evidence Accumulation ada 2 parameter yg mempengaruhi kinerja algoritma clustering, yaitu k (jumlah cluster) dan t (threshold). Metode Consensus Clustering menggunakan metode K-means. Pada Tugas Akhir ini menganalisis kinerja algoritma Evidence Accumulation berdasarkan variasi parameter.

Beberapa dataset yang digunakan untuk menguji kinerja metode Evidence Acumulation adalah Halfring, Spiral, dan dataset UCI. Berdasarkan hasil uji coba kedua parameter tersebut metode Evidence Accumulation mempengaruhi hasil clustering. Hasil pada metode Evidence Accumulation membentuk pola yang berbeda-beda.

Kata kunci: analisis cluster, consensus clustering, k-means clustering, evidence accumulation.

ANALYSIS PERFORMANCE CLUSTERING ALGORITHM WITH EVIDENCE ACCUMULATION METHOD

Name : Ratih Kirana Diantari
NRP : 5110100051
Department : Informatics Engineering – FTIf ITS
Advisor I : Dr. Eng. Chastine Fatichah, S.Kom.,
M.Kom.
Advisor II : Victor Hariadi, S.Si., M.Kom.

Abstract

Cluster analysis is an important data processing procedure in data mining. Through clustering, valuable information such as data distribution and characteristics can be acquired. Clustering has been widely applied in various fields, such as market segmentation, document clustering, pattern recognition, and spatial data analysis.

In this final project, clustering method which is used are Consensus Clustering and Evidence Accumulation (EA). There are two parameters of EA that affect the performance of clustering algorithm, namely k (number of clusters) and t (threshold). Consensus clustering method used a K-means clustering method. The purpose of this final project to analyze performance of EA algorithms based on parameter variations.

Some datasets used to test the performance of the EA method are Halfring, Spiral, and the UCI datasets. Based on the results of the both test parameters, EA methods affect the results of clustering. The results of the EA methods form different patterns.

Keywords : cluster analysis, consensus clustering, k-means clustering, evidence accumulation.

LEMBAR PENGESAHAN

**ANALISIS KINERJA ALGORITMA CLUSTERING
DENGAN METODE EVIDENCE ACCUMULATION**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visualisasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:
RATH KIRANA DIANTARI
NRP: 5110 100 051

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr. Eng. Chastine Fatchah, S.Kom., M.Kom. (Pembimbing 1)
NIP: 197512202001122002
2. Victor Hariadi, S.Si., M.Kom. (Pembimbing 2)
NIP: 196912281994121001



SURABAYA
Juli, 2014

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillah rabbil'alam, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“ANALISIS KINERJA ALGORITMA CLUSTERING DENGAN METODE EVIDENCE ACCUMULATION”**.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Papa, mama, mbak, dan rasya yang telah memberikan dukungan moral dan material serta doa yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan Tugas Akhir ini.
2. Bapak Rully Soelaiman, S.Kom., M.Kom., yang telah memberikan motivasi, nasehat, bimbingan dan bantuan yang banyak kepada penulis dalam mengerjakan Tugas Akhir ini
3. Ibu Dr. Eng. Chastine Faticah, S.Kom., M.Kom. selaku dosen pembimbing I yang telah memberikan nasihat, arahan, dan bantuan dalam menyelesaikan Tugas Akhir ini.
4. Bapak Victor Hariadi, S.Si., M.Kom. selaku dosen pembimbing II yang telah membantu dan membimbing penulis dalam menyelesaikan Tugas Akhir ini.

5. Ibu Dr. Ir. Siti Rochimah, M.T. selaku dosen wali penulis, Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Dr. Eng. Radityo Anggoro, S.Kom., M.Sc. selaku Sekprodi Jurusan, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.
6. Pak Yudi, Pak Sugeng, Mbak Fathin, Bu Wartani, Pak Sholeh, Pak Pri dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kuliah di Teknik Informatika ITS.
7. Kawan-kawan angkatan 2010 yang selalu menjaga kebersamaan, kakak-kakak angkatan 2007, 2008, dan 2009 serta adik-adik angkatan 2011 dan 2012 yang membuat penulis untuk selalu belajar.
8. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juli 2014

DAFTAR ISI

| | |
|---|------|
| Halaman Judul..... | i |
| LEMBAR PENGESAHAN..... | v |
| Abstrak | vii |
| Abstract | ix |
| KATA PENGANTAR..... | xi |
| DAFTAR ISI..... | xiii |
| DAFTAR GAMBAR | xvii |
| DAFTAR TABEL | xix |
| DAFTAR KODE SUMBER | xxi |
| BAB 1 PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah..... | 2 |
| 1.4 Tujuan..... | 2 |
| 1.5 Metodologi | 3 |
| 1.6 Sistematika Penulisan..... | 4 |
| BAB 2 DASAR TEORI | 5 |
| 2.1 <i>Clustering</i> | 5 |
| 2.1.1 Consensus Clustering | 6 |
| 2.1.2 <i>K-means clustering</i> | 8 |
| 2.2 Metode <i>Evidence Accumulation (EA)</i> | 8 |
| 2.3 Single Link (SL)..... | 10 |
| BAB 3 PERANCANGAN PERANGKAT LUNAK | 13 |

| | | |
|-----------------------------------|--|----|
| 3.1 | Perancangan Data | 13 |
| 3.1.1 | Data Masukan | 13 |
| 3.1.2 | Data Keluaran | 13 |
| 3.2 | Perancangan Metode Secara Umum..... | 14 |
| 3.3 | Perancangan Tahap <i>K-means Clustering</i> | 16 |
| 3.4 | Perancangan Tahap <i>Combine</i> | 17 |
| 3.5 | Perancangan Tahap <i>Single Link (SL)</i> | 19 |
| BAB 4 IMPLEMENTASI | | 21 |
| 4.1 | Lingkungan Implementasi Program | 21 |
| 4.2 | Implementasi | 21 |
| 4.2.1 | Implementasi Program Utama | 21 |
| 4.2.2 | Implementasi Tahap <i>K-means Clustering</i> | 24 |
| 4.2.3 | Implementasi Tahap <i>Single Link</i> | 26 |
| 4.2.4 | Implementasi Tahap <i>Plot</i> | 27 |
| 4.2.5 | Implementasi Tahap Grafik | 29 |
| BAB 5 UJI COBA DAN EVALUASI | | 31 |
| 5.1 | Lingkungan Uji Coba | 31 |
| 5.2 | Data Uji Coba | 31 |
| 5.3 | Skenario Uji Coba | 32 |
| 5.3.1 | Uji Coba Halfring | 32 |
| 5.3.2 | Uji Coba Spiral | 33 |
| 5.3.3 | Uji Coba Iris | 34 |
| 5.3.4 | Uji Coba Wine | 35 |

| | | |
|--|---------------------|-----------|
| 5.3.5 | Uji Coba Glass..... | 36 |
| 5.3.6 | Uji Coba Vowel..... | 37 |
| 5.3.7 | Uji Coba Ecoli..... | 38 |
| 5.3.8 | Uji Coba Seeds..... | 39 |
| BAB 6 KESIMPULAN DAN SARAN..... | | 41 |
| 6.1 | Kesimpulan..... | 41 |
| 6.2 | Saran..... | 41 |
| DAFTAR PUSTAKA..... | | 43 |
| A. | LAMPIRAN A..... | 45 |
| B. | LAMPIRAN B..... | 49 |
| BIODATA PENULIS..... | | 63 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Contoh <i>Clustering</i> Data..... | 5 |
| Gambar 2.2 Consensus Clustering | 7 |
| Gambar 2.3 Ilustrasi <i>Single Link</i> | 11 |
| Gambar 3.1 Diagram Alir <i>Evidence Accumulation</i> | 15 |
| Gambar 3.2 Diagram Alir Tahap <i>K-means Clustering</i> | 16 |
| Gambar 3.3 <i>Pseudocode</i> Tahap <i>K-means Clustering</i> | 17 |
| Gambar 3.4 <i>Pseudocode</i> Tahap <i>Combine</i> | 18 |
| Gambar 3.5 <i>Pseudocode</i> Tahap SL | 19 |
| Gambar 5.1 Grafik Halfring | 33 |
| Gambar 5.2 Grafik Spiral | 34 |
| Gambar 5.3 Grafik Iris | 35 |
| Gambar 5.4 Grafik Wine | 36 |
| Gambar 5.5 Grafik Glass..... | 37 |
| Gambar 5.6 Grafik Vowel..... | 38 |
| Gambar 5.7 Grafik Ecoli | 39 |
| Gambar 5.8 Grafik Seeds | 40 |
| Gambar B.1 <i>Dataset</i> Halfring | 49 |
| Gambar B.2 Uji Coba Halfring yang Optimal $k = 15$ & $t = 0,2$.. | 50 |
| Gambar B.3 Uji Coba Halfring yang Tidak Optimal $k = 5$ & $t =$ 0,5..... | 50 |
| Gambar B.4 <i>Dataset</i> Spiral | 51 |
| Gambar B.5 Uji Coba Spiral yang Optimal $k = 20$ & $t = 0,7$ | 51 |
| Gambar B.6 Uji Coba Spiral yang Tidak Optimal $k = 3$ & $t = 0,9$ | 52 |
| Gambar B.7 <i>Dataset</i> Iris..... | 52 |
| Gambar B.8 Uji Coba Iris yang Optimal $k = 3$ & $t = 0,7$ | 53 |
| Gambar B.9 Uji Coba Iris yang Tidak Optimal $k = 5$ & $t = 0,7$.. | 53 |
| Gambar B.10 <i>Dataset</i> Wine | 54 |
| Gambar B.11 Uji Coba Wine yang Optimal $k = 3$ & $t = 0,2$ | 54 |

| | |
|--|----|
| Gambar B.12 Uji Coba Wine yang Tidak Optimal $k = 10$ & $t = 0,5$ | 55 |
| Gambar B.13 <i>Dataset</i> Glass | 55 |
| Gambar B.14 Uji Coba Glass yang Optimal $k = 10$ & $t = 0,4$ | 56 |
| Gambar B.15 Uji Coba Glass yang Tidak Optimal $k = 3$ & $t = 0,7$ | 56 |
| Gambar B.16 <i>Dataset</i> Vowel..... | 57 |
| Gambar B.17 Uji Coba Vowel yang Optimal $k = 31$ & $t = 0,6$ | 57 |
| Gambar B.18 Uji Coba Vowel yang Tidak Optimal $k = 5$ & $t = 0,8$ | 58 |
| Gambar B.19 <i>Dataset</i> Ecoli..... | 58 |
| Gambar B.20 Uji Coba Ecoli yang Optimal $k = 10$ & $t = 0,7$ | 59 |
| Gambar B.21 Uji Coba Ecoli yang Tidak Optimal $k = 18$ & $t = 0,6$ | 59 |
| Gambar B.22 <i>Dataset</i> Seeds | 60 |
| Gambar B.23 Uji Coba Seeds yang Optimal $k = 3$ & $t = 0,6$ | 60 |
| Gambar B.24 Uji Coba Seeds yang Tidak Optimal $k = 10$ & $t = 0,5$ | 61 |

DAFTAR KODE SUMBER

| | |
|---|----|
| Kode Sumber 4.1 Implementasi Program Utama (Bagian 1)..... | 22 |
| Kode Sumber 4.2 Implementasi Program Utama (Bagian 2)..... | 23 |
| Kode Sumber 4.3 Implementasi Program Utama (Bagian 3)..... | 24 |
| Kode Sumber 4.4 Implementasi <i>K-means Clustering</i> | 25 |
| Kode Sumber 4.5 Implementasi Fungsi <i>DistMatrix.m</i> | 26 |
| Kode Sumber 4.6 Implementasi Tahap <i>Single Link</i> (Bagian 1) .. | 26 |
| Kode Sumber 4.7 Implementasi Tahap <i>Single Link</i> (Bagian 2) .. | 27 |
| Kode Sumber 4.8 Implementasi Tahap <i>Plot</i> | 28 |
| Kode Sumber 4.9 Program Pembentukan Grafik | 29 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 3.1 Daftar Parameter dalam Tahap <i>K-means Clustering</i> ... | 17 |
| Tabel 3.2 Parameter dalam Tahap <i>Combine</i> | 18 |
| Tabel 3.3 Daftar Parameter dalam Tahap SL | 19 |
| Tabel 5.1 Data Uji Coba..... | 32 |
| Tabel A.1 Uji Coba Halfring..... | 45 |
| Tabel A.2 Uji Coba Spiral..... | 45 |
| Tabel A.3 Uji Coba Iris | 46 |
| Tabel A.4 Uji Coba Wine..... | 46 |
| Tabel A.5 Uji Coba Glass..... | 47 |
| Tabel A.6 Uji Coba Vowel..... | 47 |
| Tabel A.7 Uji Coba Ecoli | 48 |
| Tabel A.8 Uji Coba Seeds | 48 |

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Clustering adalah metode penganalisaan data yang bertujuan untuk mengelompokkan data dengan karakteristik yang sama ke suatu “wilayah” yang sama dan data dengan karakteristik yang berbeda ke “wilayah” yang lain. Ada dua pendekatan utama dalam metode *clustering* yaitu pendekatan partisi (*Partition-based Clustering*) dan pendekatan hierarki (*Hierarchical Clustering*). Salah satu contoh untuk pendekatan partisi adalah *Consensus Clustering*.

Consensus Clustering menggabungkan beberapa *clustering* tanpa akses ke fitur-fitur yang mendasari data, sehingga menghasilkan *clustering* akhir yang kuat dibandingkan dengan beberapa *clustering* lainnya. Algoritma *consensus clustering* menghasilkan *clustering* yang lebih baik, menemukan pengelompokan *clustering* yang tidak terjangkau oleh algoritma *clustering* tunggal, kurang sensitif terhadap *noise*, *outlier* atau variasi sampel dan mampu mengintegrasikan solusi dari berbagai sumber data atau atribut. *Consensus Clustering* juga dapat berguna dalam berbagai domain. Misalnya, mengelompokkan data kategori dapat dianggap sebagai masalah *consensus clustering* dimana masing-masing fitur diskrit dipandang sebagai pengelompokan sederhana dari data [1]. Kelemahan dari *consensus clustering* ini adalah kesamaan berpasangan seringkali tidak menggambarkan kesamaan ukuran yang baik antara titik data, terutama ketika jumlah basis-*clustering* terbatas [2]. Sebagian besar algoritma ini kembali ke *consensus clustering* tunggal sebagai hasil akhir. Ada dua pendekatan utama pada solusi *consensus*, yaitu berpatokan pada model probabilistik dan mengukur kesamaan antara dua *clustering*.

Pada Tugas Akhir ini, *Consensus Clustering* diolah menggunakan metode *Evidence Accumulation* (EA). Metode ini

mampu menangani partisi lengkap dalam *ensemble* serta parsial. Ada beberapa parameter yang diidentifikasi, misalnya nilai k (jumlah *cluster*) dan (t) *threshold* dimana beberapa parameter ini akan mempengaruhi kinerja *clustering*.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Memahami konsep *Consensus Clustering*.
2. Merancang metode *Consensus Clustering* dan *Evidence Accumulation*.
3. Mengimplementasikan metode *Consensus Clustering* dan *Evidence Accumulation* yang melibatkan beberapa parameter.
4. Melakukan uji coba menggunakan metode *Evidence Accumulation* pada *Consensus Clustering* yang melibatkan beberapa parameter dengan beberapa dataset.

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki batasan sebagai berikut:

1. Implementasi dan uji coba dilakukan dengan menggunakan MATLAB 7.6.0 atau MATLAB R2008a.
2. *Dataset* yang digunakan adalah *Half-ring*, *Spiral*, dan *dataset* dari <http://archive.ics.uci.edu/ml/> yaitu *Ecoli*, *Glass*, *Iris*, *Seeds*, *Vowel*, dan *Wine*. Kedelapan *dataset* tersebut dipilih karena memiliki jumlah atribut dan jumlah data yang beragam. Untuk data *Iris*, *Wine*, *Glass*, dan *Ecoli* tidak memiliki *missing value*.
3. Pada bagian *Consensus Clustering* menggunakan K-means.

1.4 Tujuan

Tugas Akhir ini memiliki beberapa tujuan yang rinciannya dapat ditulis sebagai berikut:

1. Mengembangkan aplikasi clustering berbasis Consensus Clustering dan Evidence Accumulation.
2. Mengidentifikasi kinerja Consensus Clustering dengan metode Evidence Accumulation yang dipengaruhi beberapa parameter, yaitu k (jumlah *cluster*) dan t (*threshold*).

1.5 Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1. Penyusunan Proposal Tugas Akhir
Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan Proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan untuk mengimplementasikan *clustering* berbasis *Evidence Accumulation*.
2. Studi Literatur
Pada tahapan ini akan dilakukan studi literatur mengenai metode yang digunakan untuk implementasi *algoritma clustering*, yaitu *Evidence Accumulation*.
3. Perancangan Perangkat Lunak
Pada tahap ini dilakukan proses perancangan perangkat lunak, berdasarkan literatur yang telah dikaji kemudian dibuat desain model, dan diagram alir proses-proses yang ada. Kemudian untuk selanjutnya dilakukan implementasi.
4. Pengimplementasian Perangkat Lunak
Pada tahap ini dilakukan proses pengimplementasian, dengan berdasar pada rancangan awal perangkat lunak dan literatur yang telah dikaji. Aplikasi dibuat secara menyeluruh.
5. Uji Coba dan Evaluasi
Pada tahap ini dilakukan uji coba aplikasi yang telah selesai diimplementasikan.
6. Penyusunan Laporan Tugas Akhir
Pada tahap ini disusun buku sebagai dokumentasi dari pelaksanaan Tugas Akhir yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

1.6 Sistematika Penulisan

Laporan Tugas Akhir ini dibagi menjadi beberapa bab, sebagai berikut:

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Perancangan Perangkat Lunak

Bab ini berisi pembahasan mengenai desain yang digunakan dalam implementasi *clustering* dengan metode *Evidence Accumulation*. Pada bab ini juga dibahas mengenai perancangan data yang terdiri dari data proses, data masukan, data keluaran.

Bab IV Implementasi

Bab ini akan membahas implementasi dari aplikasi yang telah dibuat, akan dilakukan pembuatan aplikasi yang dibangun dengan komponen-komponen yang telah ada yang sesuai dengan permasalahan dan batasannya yang telah dijabarkan pada bab pertama

Bab V Uji Coba dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan dan Saran

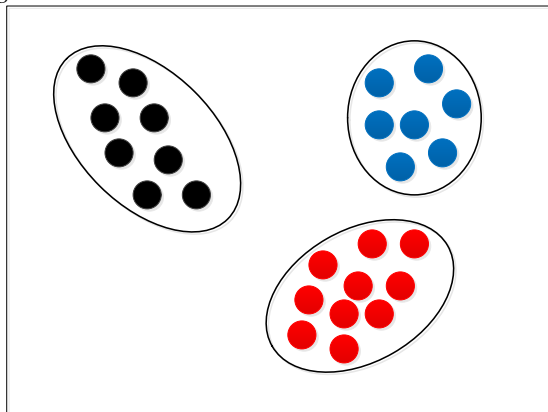
Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB 2 DASAR TEORI

Bab ini berisi uraian mengenai teori-teori yang digunakan dalam pengerjaan Tugas Akhir ini. Teori yang akan dibahas adalah *Consensus Clustering*, *Clustering*, dan metode *Evidence Accumulation (EA)* yang terdiri dari *K-means Clustering* dan *Single Link (SL)*.

2.1 *Clustering*

Clustering adalah teknik data mining yang digunakan untuk mengelompokkan data dengan karakteristik yang sama ke wilayah yang sama, dan karakteristik yang berbeda ke wilayah yang lain. Tujuan *clustering* adalah menganalisis kesamaan data dan membaginya ke dalam beberapa *cluster*. Data yang berada pada satu *cluster* akan memiliki similaritas yang tinggi, sedangkan data yang berada pada *cluster* berbeda akan memiliki similaritas yang rendah. Gambar 2.1 berikut ini adalah contoh *clustering* data.



Gambar 2.1 Contoh *Clustering* Data

Pada dasarnya, dalam *clustering* terdapat dua metode, yaitu *hierarchical* dan *partitional*. Pada metode *hierarchical*, *output* dari *clustering* berupa sebuah *tree* yang menggambarkan data yang telah dibagi menjadi beberapa *cluster*. Kelebihan teknik *hierarchical* adalah jumlah *cluster* tidak perlu ditentukan di awal. Langkah-langkah *hierarchical clustering* [3]:

1. Kelompokkan setiap objek ke dalam *clusternya* sendiri
2. Temukan pasangan paling mirip untuk dimasukkan ke dalam *cluster* yang sama dengan melihat data dalam matriks kemiripan
3. Gabungkan kedua objek dalam satu *cluster*
4. Ulangi sampai tersisa hanya satu *cluster*

Menggabungkan dua atau lebih objek menjadi satu *cluster* dilihat dari kemiripan dan ketidakmiripan. Untuk ukuran kemiripan bisa menggunakan *cosinus*, *kovarian*, dan korelasi. Untuk ketidakmiripan dihitung menggunakan jarak, yaitu *Single Linkage*, *Average Linkage*, *Centroid Linkage*, dan *Ward Linkage*. Pada teknik *partitional*, jumlah *cluster* harus ditentukan di awal. Penentuan jumlah *cluster* ini sangat berpengaruh terhadap letak *centroid* dan hasil *clustering*. Jika jumlah *cluster* kecil, hasil *clustering* akan bagus tetapi tidak dapat menemukan informasi tersembunyi dari data. Di sisi lain, jika menggunakan jumlah *cluster* yang besar, informasi tersembunyi dari data akan dapat diketahui, akan tetapi hasil *clustering* akan tidak bagus karena memiliki tingkat homogenitas yang tinggi antara satu *cluster* dengan *cluster* lain.

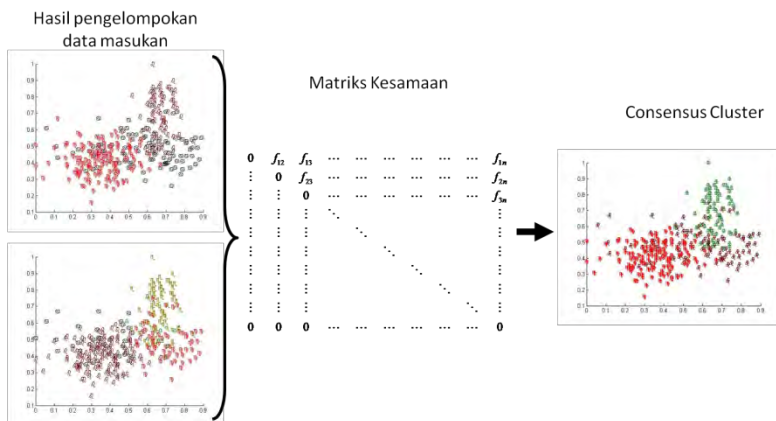
2.1.1 Consensus Clustering

Menggabungkan beberapa *cluster* tanpa akses ke fitur-fitur yang mendasari data disebut sebagai *Consensus Clustering*. Hasil yang diperoleh berupa *cluster* akhir yang kuat dan stabil dibandingkan dengan *cluster* tunggal. Istilah ini juga bisa disebut sebagai *clustering ensembles*, atau *clustering aggregation* [1].

Metode *clustering ensemble* memanfaatkan *consensus* di beberapa solusi *clustering* dan menggabungkan beberapa solusi ini menjadi satu *consensus* tunggal. Keuntungan metode ini adalah hasil lebih akurat dan kuat [4].

Pendekatan utama *consensus clustering* adalah berpatokan pada model probabilistik dan mengukur kesamaan antara dua *clustering*. Metode ini juga dapat berguna dalam berbagai domain.

Clustering ini dapat digunakan dalam *clustering multi objective* sebagai hubungan antara *clustering* tunggal dengan fungsi tujuan yang saling bertentangan. Metode ini memiliki dua tahap algoritma, yaitu menyimpan beberapa hasil independen yang berjalan pada *K-means* dan menggunakan fungsi *consensus* tertentu untuk menemukan partisi akhir dari hasil yang disimpan [5]. Penjelasan secara garis besar pada metode ini dapat dilihat pada Gambar 2.2.



Gambar 2.2 Consensus Clustering

2.1.2 *K-means clustering*

Metode *K-means* diperkenalkan oleh *J.B. MacQueen* pada tahun 1976, merupakan metode yang sangat umum untuk mengelompokkan data sesuai dengan karakteristik yang serupa. *K-means clustering* merupakan metode sederhana, umum, dan cepat dalam pengelompokkan data (*clustering*). Teknik ini berguna untuk mengelompokkan objek ke dalam k kelompok atau *cluster*. Nilai k harus ditentukan terlebih dahulu, bisa juga diacak atau *random*. Jika jarak dua objek tidak cukup dekat, maka dua objek itu tidak mirip, dan sebaliknya. Semakin tinggi nilai jarak, semakin tinggi ketidakmiripannya.

Pada metode ini, jumlah *cluster* ditentukan terlebih dahulu. Setiap data dihitung jaraknya dengan titik pusat *cluster* (*centroid*). Data akan dikelompokkan berdasarkan jarak terdekat terhadap *centroid*. Jika ada sebuah *cluster* yang tidak memiliki anggota, maka *cluster* tersebut akan dihapus.

Langkah-langkah dalam algoritma *k-means clustering* ini adalah sebagai berikut [6]:

1. Inisialisasi titik k sebagai titik pusat (*centroid*) awal dari setiap *cluster*.
2. Kelompokkan tiap data ke dalam *cluster* yang mempunyai jarak terdekat.
3. Setelah selesai untuk semua data, hitung kembali posisi *centroid* yang baru.
4. Ulangi langkah 2 dan 3 sampai titik pusat tidak berubah.

2.2 *Metode Evidence Accumulation (EA)*

Ide *Evidence Accumulation* adalah menggabungkan hasil dari beberapa *cluster*. Pertama, *clustering ensemble* – satu set partisi objek dibentuk. Cara yang berbeda untuk menghasilkan partisi data adalah menerapkan algoritma *clustering* yang berbeda, dan menerapkan algoritma *clustering* yang sama dengan nilai parameter atau inisialisasi yang berbeda. Kombinasi dari representasi data yang berbeda dan algoritma *clustering* juga

dapat memberikan banyak data partisi yang berbeda secara signifikan [3]. Algoritma ini memiliki konsep bahwa setiap partisi dipandang sebagai bukti independen organisasi data, partisi data individu yang digabungkan, berdasarkan mekanisme *voting*, untuk menghasilkan $n \times n$ kesamaan matrik baru antara pola n . Keuntungan algoritma ini dapat menghindari masalah korespondensi label, yang mempengaruhi skema pengelompokan *ensemble* lainnya. EA memiliki pola dimensi n - d , strategi yang diusulkan mengikuti pendekatan *split* dan *merge*:

1. *Split*

Mengurangi data multidimensi menjadi beberapa *cluster* kecil. Algoritma *K-Means* melakukan pengurangan ini dengan berbagai hasil *cluster* diperoleh inisialisasi acak algoritma.

2. *Combine*

Mengatasi partisi dengan jumlah *cluster* yang berbeda, menggunakan “*voting*” untuk menggabungkan hasil *clustering*, sesuai ke ukuran baru dari kesamaan antara pola. Asumsinya adalah pola-pola *cluster* yang alami sangat mungkin menjadi “letak” di *cluster* yang sama di *clustering* yang berbeda. Mengambil *co-occurrences* dalam pola berpasangan di *cluster* yang sama sebagai hasil untuk asosiasi mereka, partisi data dijalankan dalam *K-Means*. Mengambil pola berpasangan *co-occurrence* dalam *cluster* yang sama sebagai “*voting*” untuk asosiasi, data partisi yang dihasilkan oleh *K-Means* dipetakan pada matriks *co-association* $n \times n$, yang ditunjukkan oleh Persamaan(2.1):

$$CO_{assoc}(i,j) = \frac{votes_{ij}}{N} \quad (2.1)$$

dimana N merupakan jumlah *clustering* dan $votes_{ij}$ merupakan berapa kali pasangan berpola (i,j) ditugaskan untuk *cluster* yang sama antara *clustering* N .

3. Merge

Untuk membentuk *cluster* ke bentuk semula, menggunakan metode *Single Link* (SL).

Evidence Accumulation adalah metode untuk menggabungkan hasil dari *multiple clustering* menjadi data partisi individu, dengan melihat setiap hasil *cluster* sebagai data yang independen. Keuntungan algoritma ini dapat menghindari masalah korespondensi label, yang mempengaruhi skema pengelompokan *ensemble* lainnya. Kekurangan dari algoritma ini adalah memiliki kompleksitas yang sangat tinggi dalam waktu dan ruang yang berasal dari menyimpan dan mengakses inputan matriks *co-association*.

2.3 Single Link (SL)

Linked List adalah sejumlah objek yang dihubungkan satu dengan yang lainnya sehingga membentuk suatu *list*. Obyek yang dimaksud adalah gabungan beberapa elemen data (*variabel*) yang dijadikan satu kelompok. Salah satu bagian dari *linked list* adalah *Single Link*.

Metode *Single Link* ditemukan oleh S.C. Johnson pada 1967, merupakan metode untuk menggabungkan beberapa *cluster*. Jarak antara satu *cluster* dengan *cluster* lainnya sama dengan jarak terpendek dari setiap anggota *cluster* untuk setiap anggota dari *cluster* lainnya. Data yang memiliki kesamaan jarak akan digabungkan ke dalam satu *cluster*. Istilah lain dari metode ini adalah *neighbor joining*. Perhitungan *single link* dapat dilihat pada persamaan (2.2) berikut ini:

$$d(r,s) = \min(\text{dist}(x_{r_i}, x_{s_j})), i \in (1, \dots, n_r), j \in (1, \dots, n_s) \quad (2.2)$$

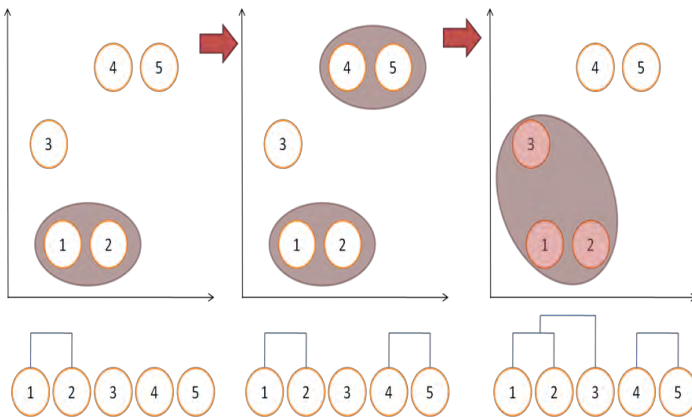
Metode *Single Link* termasuk dalam *clustering hierarchical* yang bersifat *bottom up* yaitu penggabungan n buah objek menjadi satu *cluster* tunggal atau sampai jumlah yang telah kita tentukan. Waktu kompleksitas metode ini adalah $O(n^2)$. Metode ini awalnya dimulai dengan meletakkan setiap objek data

sebagai sebuah *cluster* tersendiri dan selanjutnya menggabungkan *cluster-cluster* tersebut menjadi *cluster* yang lebih besar dan akhirnya semua objek menyatu dalam sebuah *cluster* tunggal. Metode ini dapat menghemat memori apabila data yang digunakan sangat besar.

Langkah-langkah untuk metode *Single Link* [8]:

1. Menentukan k sebagai jumlah *cluster* yang ingin dibentuk.
2. Setiap data dianggap sebagai *cluster*. Jika $n =$ jumlah data dan $c =$ jumlah *cluster*, berarti ada $c = n$.
3. Menghitung jarak/*similarity/dissimilarity* antar data.
4. Mencari 2 *cluster* yang mempunyai jarak antar *cluster* yang paling minimal dan digabungkan ($c = c - 1$).
5. Jika $c > 3$, kembali ke langkah 3.

Ilustrasi metode *Single Link* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Ilustrasi *Single Link*

BAB 3

PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dibahas mengenai perancangan perangkat lunak yang akan digunakan untuk menyelesaikan Tugas Akhir. Pembahasan perancangan perangkat lunak akan meliputi, penjelasan tentang *dataset* yang digunakan untuk uji coba, perancangan data, serta perancangan proses-proses yang ada dalam Tugas Akhir ini secara lebih mendetail. Perancangan sistem pada bagian ini meliputi dua bagian penting, yaitu perancangan data yang akan digunakan dalam sistem dan algoritma yang akan digunakan dalam sistem yang digambarkan dengan *pseudocode*.

3.1 Perancangan Data

Perancangan data merupakan hal penting untuk diperhatikan karena diperlukan data yang tepat agar perangkat lunak beroperasi secara benar. Data yang diperlukan dalam pengoperasian perangkat lunak, yaitu data masukan (*input*) dan data keluaran (*output*).

3.1.1 Data Masukan

Data masukan (*input*) merupakan data yang dimasukkan ke dalam perangkat lunak. Pada tugas akhir ini, terdapat 8 *dataset* yang digunakan sebagai data masukan pada uji coba evaluasi, yaitu *dataset Halfring*, *Spiral*, dan *dataset* yang didapat pada situs *The UCI KDD Archive*, yaitu *Iris*, *Wine*, *Glass*, *Vowel*, *Ecoli*, dan *Seeds*.

3.1.2 Data Keluaran

Data keluaran (*output*) yang dihasilkan berupa hasil *cluster* setiap data yang sudah digabungkan dengan beberapa percobaan parameter. Data yang sudah bergabung menghasilkan *cluster* yang lebih kuat dan stabil.

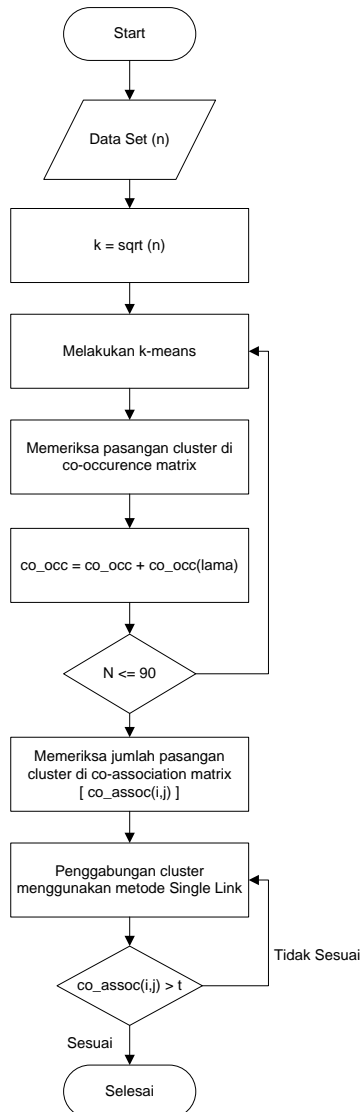
3.2 Perancangan Metode Secara Umum

Langkah awal pada metode *Evidence Accumulation* (EA) adalah membagi data menjadi beberapa bagian dengan menggunakan *k-means*. Data akan dibagi sesuai dengan letak *cluster*-nya, dengan jumlah *cluster* yang telah ditentukan di awal. Beberapa pasangan *cluster* yang muncul di setiap perulangan akan disimpan di matriks *co-occurrence*. Perulangan ini dilakukan sebanyak 90 kali, untuk mendapatkan pola pasangan yang tetap. Pasangan *cluster* yang ada akan dihitung seberapa banyak yang muncul dan disimpan di matriks *co-association*. Jumlah pasangan *cluster* akan digabungkan dengan metode *single link*, dengan melihat nilai *co-association* lebih besar dari nilai ambang (*threshold*).

Secara umum, langkah-langkah dalam metode *Evidence Accumulation* adalah sebagai berikut:

1. Tentukan letak *centroid* pada setiap data
2. Tahap *k-means clustering*. Tahap ini akan mengelompokkan data sesuai dengan *centroid* terdekat.
3. Tahap *co-occurrence*. Tahap ini melihat pasangan *cluster* yang muncul di setiap perulangan.
4. Tahap *co-association*. Tahap ini melihat ada berapa banyak pasangan *cluster* yang muncul.
5. Tahap *single link*. Tahap ini penggabungan *cluster* hasil dari tahap *co-association*.

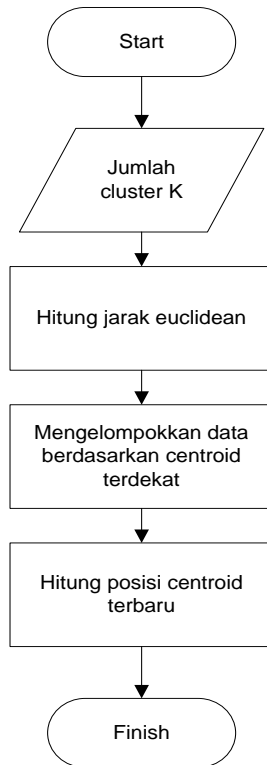
Pada Gambar 3.1 menjelaskan diagram alir dari perancangan metode secara umum (program utama). Tahap *k-means* dijelaskan pada subbab 3.3 dan tahap *single link* dijelaskan pada subbab 3.5.



Gambar 3.1 Diagram Alir *Evidence Accumulation*

3.3 Perancangan Tahap *K-means Clustering*

K-means clustering merupakan tahap untuk mengolah data menjadi beberapa bagian, yang disesuaikan dengan letak *cluster*. Dari tahap ini, diperoleh jumlah *cluster* dan letak *centroid*. Gambar 3.2 menunjukkan diagram alir tahap *K-means clustering*.



Gambar 3.2 Diagram Alir Tahap *K-means Clustering*

Daftar parameter yang digunakan pada tahap *K-means clustering* dapat dilihat pada Tabel 3.1.

Tabel 3.1 Daftar Parameter dalam Tahap *K-means Clustering*

| No | Parameter | Tipe | Keterangan |
|----|-----------|--------|---|
| 1 | k | double | Jumlah <i>cluster</i> |
| 2 | c | int | <i>Centroid</i> dengan ukuran koordinat (1:k, 1:maxCol) |
| 3 | n | double | <i>Dataset</i> |
| 4 | maxRow | matrik | Jumlah kolom dalam matrik data |
| 5 | p | double | Inisialisasi data acak |
| 6 | d | double | Nilai jarak antar data |
| 7 | g | double | Kelompok ukuran matrik saat ini (1:maxRow) |
| 8 | temp | double | Kelompok iterasi sebelumnya dengan ukuran (1:maxRow) |
| 9 | i | double | Iterator skala |
| 10 | f | double | Jumlah baris data yang merupakan anggota grup i |

Parameter pada Tabel 3.1 diimplementasikan dalam bentuk *pseudocode* seperti pada Gambar 3.3 sebelum akhirnya diimplementasikan dalam kode Matlab.

| | |
|----------|---|
| Masukan | <ol style="list-style-type: none"> 1. Inputan (n) 2. Jumlah cluster (k) 3. Inputan awal akan menetapkan data k pertama sebagai centroid awal (<i>isRand</i>) |
| Keluaran | <p>Hasil cluster setiap data (y)</p> <ol style="list-style-type: none"> 1. Hitung nilai centroid 2. Menentukan posisi data di setiap cluster 3. Inisialisasi nilai centroid 4. Proses perubahan posisi centroid 5. Menghitung jarak |

Gambar 3.3 Pseudocode Tahap *K-means Clustering*

3.4 Perancangan Tahap *Combine*

Partisi data yang ada memiliki pola berpasangan yang disimpan di *co-occurences*. Pola berpasangan data akan dipetakan

pada matriks *co_association* dengan bentuk $n \times n$. Daftar parameter yang digunakan pada tahap *Combine* dapat dilihat pada Tabel 3.2.

Tabel 3.2 Parameter dalam Tahap *Combine*

| No | Parameter | Tipe | Keterangan |
|----|-------------|--------|-------------------------------------|
| 1 | N | double | Perulangan proses <i>clustering</i> |
| 2 | y | double | Hasil <i>k-means clustering</i> |
| 3 | co_occ_baru | double | Menampung pasangan baru |
| 4 | i | double | Baris |
| 5 | j | double | Kolom |
| 6 | co_assoc | double | Hasil kesamaan data |

Parameter pada Tabel 3.2 diimplementasikan dalam bentuk *pseudocode* seperti pada Gambar 3.4.

| Masukan | Hasil cluster setiap data (y) |
|--|--|
| Keluaran | File yang berisi <i>co_assoc</i> (<i>co_assoc.dat</i>) |
| <pre> 1. Inisialisasi N dan p 2. Membuat matrik co_occ 3. for p ← 1 to N 4. y ← kMeansCluster(data, k, 1) 5. co_occ_baru ← zeros(n(1), n(1)) 6. for i ← 1 to n(1) 7. for j ← 1 to n(1) 8. if y(i) == y(j) 9. co_occ_baru(i, j) ← 1 10. end if 11. end for 12. end for 13. co_occ = co_occ + co_occ_baru 14. end for 15. co_assoc = co_occ / N </pre> | |

Gambar 3.4 Pseudocode Tahap *Combine*

3.5 Perancangan Tahap *Single Link* (SL)

Single Link merupakan tahap untuk menggabungkan beberapa *cluster* menjadi satu *cluster* yang utuh. Tahap ini melihat data dari *co_assoc* yang berisi kesamaan data, kemudian dibandingkan dengan nilai ambang batas (*threshold*). Perbandingan data dilihat dari $co_assoc > t$, data akan digabung menjadi satu *cluster*. Daftar parameter yang digunakan pada tahap *single link* dapat dilihat pada Tabel 3.3.

Tabel 3.3 Daftar Parameter dalam Tahap SL

| No | Parameter | Tipe | Keterangan |
|----|------------------|--------|---|
| 1 | <i>co_assoc</i> | double | Pasangan data |
| 2 | <i>t</i> | double | Nilai ambang batas |
| 3 | <i>cluster</i> | double | Nilai awal <i>cluster</i> |
| 4 | <i>c1</i> | double | Ukuran 1xn berisi penggabungan <i>cluster</i> |
| 5 | <i>i</i> | double | Posisi baris |
| 6 | <i>j</i> | double | Posisi kolom |
| 7 | <i>cG</i> | double | Kode <i>cluster</i> pada <i>j</i> |
| 8 | <i>n_cluster</i> | double | Jumlah <i>cluster</i> penggabungan |
| 9 | <i>cU</i> | double | Menyimpan nilai <i>c1</i> |

Parameter pada Tabel 3.3 diimplementasikan dalam bentuk *pseudocode* seperti pada Gambar 3.5.

| | |
|---|--|
| Masukan | File yang berisi <i>co_assoc</i> (<i>co_assoc.dat</i>) |
| Keluaran | Hasil penggabungan <i>cluster</i> (<i>c1</i>) |
| <ol style="list-style-type: none"> 1. $co_assoc \leftarrow \text{d1mread}('co_assoc.dat')$ 2. Inisialisasi nilai <i>t</i> 3. $cluster = 1$ 4. Menentukan posisi data di setiap <i>cluster</i> 5. Membentuk vector untuk menyimpan data 6. Proses $co_assoc > t$ 7. Pengelompokan data | |

Gambar 3.5 Pseudocode Tahap SL

BAB 4 IMPLEMENTASI

Bab ini berisi penjelasan mengenai implementasi perangkat lunak yang meliputi algoritma dan kode sumber yang terdapat dalam perangkat lunak.

4.1 Lingkungan Implementasi Program

Lingkungan implementasi perangkat lunak yang digunakan dalam pembuatan Tugas Akhir ini meliputi perangkat lunak dan perangkat keras yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™2 Duo CPU U9600 @ 1.60 GHz
 - b. Memory (RAM): 4 GB
 - c. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak
 - a. Sistem operasi: Windows 7 Professional
 - b. Perangkat pengembang: MATLAB 7.13.0 (R2011b)

4.2 Implementasi

Implementasi dilakukan sesuai dengan proses yang dijelaskan pada bab sebelumnya.

4.2.1 Implementasi Program Utama

Program utama ini terdiri dari proses masukan meliputi pemilihan *dataset* yang akan digunakan, jumlah *cluster* dan jumlah *threshold*. Proses selanjutnya adalah tahap membagi data sesuai dengan letak cluster-nya menggunakan *K-means*. Data yang ada akan digabungkan menggunakan *Single Link*. Implementasi program utama dapat dilihat pada Kode Sumber 4.1 sampai Kode Sumber 4.3 berikut ini.

| | |
|----|--------------------------------------|
| 1 | clc; |
| 2 | clear all |
| 3 | close all |
| 4 | disp data; |
| 5 | selData = input('Pilih Data set: '); |
| 6 | [dataSet, name] = get_data(selData); |
| 7 | dim selData=size(dataSet); |
| 8 | n = size(dataSet); |
| 9 | <i>%proses penentuan k</i> |
| 10 | k = round(sqrt(n(1))); |
| 11 | y = kMeansCluster(dataSet,k,1); |
| 12 | n cluster = size(unique(y),q); |
| 13 | N = 90; |
| 14 | p = 1;; |
| 15 | co occ = zeros(n(1),n(1)) |
| 16 | <i>%proses co assoc</i> |
| 17 | for p = 1:N |
| 18 | y = kMeansCluster(dataSet,k,1); |
| 19 | co occ baru = zeros(n(1),n(1)); |
| 20 | for i = 1:n(1) |
| 21 | for j = 1:n(1) |
| 22 | if y(i)==y(j) |
| 23 | co occ baru(i,j) = 1; |
| 24 | end |
| 25 | end |
| 26 | end |
| 27 | co occ = co occ + co occ baru; |
| 28 | end |
| 29 | co assoc = co occ / N; |
| 30 | dlmwrite('co assoc.dat',co assoc); |
| 31 | <i>%proses pembacaan file</i> |
| 32 | co assoc = dlmread('co assoc.dat'); |
| 33 | t = 0,01; |
| 34 | cluster = 1; |
| 35 | c1 = zeros(n(1),1); |

Kode Sumber 4.1 Implementasi Program Utama (Bagian 1)

| | |
|----|--|
| 36 | <code>%proses penggabungan cluster</code> |
| 37 | <code>for i = 1:n(1)</code> |
| 38 | <code> for j = 1:i</code> |
| 39 | <code> if co_assoc(i,j) > t</code> |
| 40 | <code> if cl(i)==0 && cl(j)==0</code> |
| 41 | <code> cl(i)=cluster;</code> |
| 42 | <code> cl(j)=cluster;</code> |
| 43 | <code> cluster = cluster + 1;</code> |
| 44 | <code> elseif cl(i)>0 && cl(j)==0</code> |
| 45 | <code> cl(j)=cl(i);</code> |
| 46 | <code> elseif cl(i)==0 && cl(j)>0</code> |
| 47 | <code> cl(i)=cl(j);</code> |
| 48 | <code> elseif cl(i)>0 && cl(j)>0</code> |
| 49 | <code> cG = cl(j);</code> |
| 50 | <code> cl(cl==cG)=cl(i);</code> |
| 51 | <code> end</code> |
| 52 | <code> else</code> |
| 53 | <code> if cl(i)==0</code> |
| 54 | <code> cl(i)=cluster;</code> |
| 55 | <code> cluster = cluster + 1;</code> |
| 56 | <code> end</code> |
| 57 | <code> if cl(j)==0</code> |
| 58 | <code> cl(j)=cluster;</code> |
| 59 | <code> cluster = cluster + 1;</code> |
| 60 | <code> end</code> |
| 61 | <code> end</code> |
| 62 | <code> end</code> |
| 63 | <code>end</code> |
| 64 | <code>%proses plot penggabungan data</code> |
| 65 | <code>cU = unique(cl);</code> |
| 66 | <code>figure</code> |
| 67 | <code>hold on</code> |
| 68 | <code>r = 0;</code> |
| 69 | <code>g = 0;</code> |
| 70 | <code>b = 0;</code> |
| 71 | <code>font = 'Wingdis';</code> |
| 72 | <code>m = 'a';</code> |

Kode Sumber 4.2 Implementasi Program Utama (Bagian 2)

| | |
|----|---|
| 73 | for i = 1:n_cluster |
| 74 | plot(data(find(c1==cU(i),1),data(find(c1==cU(i),2),'+', 'Color',[r g b])); |
| 75 | text(data(find(c1==cU(i),1),data(find(c1==cU(i),2),m,'fontname',font,'HorizontalAl', 'center', 'color',[r g b])); |
| 76 | m = char(m+1); |
| 77 | r = r + 0.5; |
| 78 | if r > 1 |
| 79 | r = 0; |
| 80 | g = g + 0.5; |
| 81 | end |
| 82 | if g > 1; |
| 83 | g = 0; |
| 84 | b = b + 0.5; |
| 85 | end |
| 86 | if b > 1 |
| 87 | r = 0 ; |
| 88 | g = 0; |
| 89 | b = 0; |
| 90 | end |
| 91 | if m > 'z' |
| 92 | m = 'A'; |
| 93 | end |
| 94 | end |

Kode Sumber 4.3 Implementasi Program Utama (Bagian 3)

4.2.2 Implementasi Tahap *K-means Clustering*

Kode Sumber 4.4 menunjukkan implementasi *K-means clustering*. Baris 2 menunjukkan jumlah inputan untuk proses *K-means clustering*. Baris 5 menunjukkan nilai *centroid*. Baris 8 menunjukkan posisi data di setiap *cluster*. Baris 10 - 20 menjelaskan proses inisialisasi nilai *centroid*. Baris 21 - 35 menunjukkan proses perubahan *centroid*.

| | |
|----|--|
| 1 | <code>function y = kMeansCluster(n,k,isRand)</code> |
| 2 | <code>if nargin<2,</code> |
| 3 | <code>k = 1;</code> |
| 4 | <code>end</code> |
| 5 | <code>c = zeros(k,size(n,2));</code> |
| 6 | <code>[maxRow,~]=size(n)</code> |
| 7 | <code>if maxRow<=k</code> |
| 8 | <code> y = [n,1:maxRow]</code> |
| 9 | <code>else</code> |
| 10 | <code> %inisialisasi nilai centroid</code> |
| 11 | <code> if isRand,</code> |
| 12 | <code> p = randperm(size(n,1));</code> |
| 13 | <code> for i=1:k</code> |
| 14 | <code> c(i,:)=n(p(i),:);</code> |
| 15 | <code> end</code> |
| 16 | <code> else</code> |
| 17 | <code> for i=1:k</code> |
| 18 | <code> c(i,:)=n(i,:);</code> |
| 19 | <code> end</code> |
| 20 | <code> end</code> |
| 21 | <code>temp = zeros(maxRow,1);</code> |
| 22 | <code>while 1,</code> |
| 23 | <code> d = DistMatrix(n,c);</code> |
| 24 | <code> [~,g]=min(d,[],2);</code> |
| 25 | <code> if g==temp,</code> |
| 26 | <code> break;</code> |
| 27 | <code> else</code> |
| 28 | <code> temp=g;</code> |
| 29 | <code> end</code> |
| 30 | <code> for i=1:k</code> |
| 31 | <code> f=find(g==i);</code> |
| 32 | <code> if f</code> |
| 33 | <code> c(i,:)=mean(n(find(g==i),:),1);</code> |
| 34 | <code> end</code> |
| 35 | <code> end</code> |
| 36 | <code>end y=g;</code> |
| 37 | <code>end</code> |

Kode Sumber 4.4 Implementasi *K-means Clustering*

4.2.2.1 Implementasi Fungsi *DistMatrix.m*

Kode Sumber 4.5 menunjukkan implementasi fungsi *DistMatrix.m*. Baris 6 menunjukkan perhitungan jarak antar data. Pada baris 12 menghitung jarak *euclidean*.

| | |
|----|---|
| 1 | <code>function d=DistMatrix(A,B)</code> |
| 2 | <code>[hA,~]=size(A);</code> |
| 3 | <code>[hB,~]=size(B);</code> |
| 4 | <code>d = zeros(hA,hB);</code> |
| 5 | <code>if hA==1 && hB==1</code> |
| 6 | <code> d=sqrt(dot((A-B),(A-B)));</code> |
| 7 | <code>else</code> |
| 8 | <code> for i=1:hA</code> |
| 9 | <code> for j=1:hB</code> |
| 10 | <code> pA = A(i,:);</code> |
| 11 | <code> pB = B(j,:);</code> |
| 12 | <code> d(i,j)=sum((pA-pB).^2).^0.5;</code> |
| 13 | <code> end</code> |
| 14 | <code> end</code> |
| 15 | <code>end</code> |

Kode Sumber 4.5 Implementasi Fungsi *DistMatrix.m*

4.2.3 Implementasi Tahap *Single Link*

Kode Sumber 4.6 sampai Kode Sumber 4.7 menunjukkan implementasi tahap *Single Link*. Baris 1 menunjukkan *file* yang akan diproses. Baris 2 merupakan inisialisasi nilai ambang batas (*threshold*). Baris 4 - 31 menjelaskan proses penggabungan *cluster*.

| | |
|---|--|
| 1 | <code>co assoc = dlmread('co assoc.dat');</code> |
| 2 | <code>t = 0,01;</code> |
| 3 | <code>cluster = 1;</code> |
| 4 | <code>c1 = zeros(n(1),1);</code> |
| 5 | <code>%proses penggabungan cluster</code> |
| 6 | <code>for i = 1:n(1)</code> |

Kode Sumber 4.6 Implementasi Tahap *Single Link* (Bagian 1)

| | |
|----|-----------------------------------|
| 7 | for j = 1:i |
| 8 | if co_assoc(i,j) > t |
| 9 | if cl(i)==0 && cl(j)==0 |
| 10 | cl(i)=cluster; |
| 11 | cl(j)=cluster; |
| 12 | cluster = cluster + 1; |
| 13 | elseif cl(i)>0 && cl(j)==0 |
| 14 | cl(j)=cl(i); |
| 15 | elseif cl(i)==0 && cl(j)>0 |
| 16 | cl(i)=cl(j); |
| 17 | elseif cl(i)>0 && cl(j)>0 |
| 18 | cG = cl(j); |
| 19 | cl(cl==cG)=cl(i); |
| 20 | end |
| 21 | else |
| 22 | if cl(i)==0 |
| 23 | cl(i)=cluster; |
| 24 | cluster = cluster + 1; |
| 25 | end |
| 26 | if cl(j)==0 |
| 27 | cl(j)=cluster; |
| 28 | cluster = cluster + 1; |
| 29 | end |
| 30 | end |
| 31 | end |
| 32 | end |

Kode Sumber 4.7 Implementasi Tahap *Single Link* (Bagian 2)

4.2.4 Implementasi Tahap *Plot*

Kode Sumber 4.8 menunjukkan implementasi tahap *plot* (gambar). Baris 5-7 digunakan untuk menginisialisasi warna pada setiap *cluster*. Baris 8 merupakan perintah *plot* berbentuk teks. Pada baris 11-20 adalah menjelaskan proses *plot* yang berlangsung.

| | |
|----|---|
| 1 | <code>n_cluster = size(unique(y),q);</code> |
| 2 | <code>cU = unique(c1);</code> |
| 3 | <code>figure</code> |
| 4 | <code>hold on</code> |
| 5 | <code>r = 0;</code> |
| 6 | <code>g = 0;</code> |
| 7 | <code>b = 0;</code> |
| 8 | <code>font = 'Wingdis';</code> |
| 9 | <code>m = 'a';</code> |
| 10 | <code>for i = 1:n_cluster</code> |
| 11 | <code>plot(data(find(c1==cU(i),1),data(find(c1==cU(i),2),'+', 'Color',[r g b]));</code> |
| 12 | <code>text(data(find(c1==cU(i)),1),data(find(c1==cU(i),2),m,' fontname', font,'HorizontalAl', 'center', 'color',[r g b]));</code> |
| 13 | <code> m = char(m+1);</code> |
| 14 | <code> r = r + 0.5;</code> |
| 15 | <code> if r > 1</code> |
| 16 | <code> r = 0;</code> |
| 17 | <code> g = g + 0.5;</code> |
| 18 | <code> end</code> |
| 19 | <code> if g > 1;</code> |
| 20 | <code> g = 0;</code> |
| 21 | <code> b = b + 0.5;</code> |
| 22 | <code> end</code> |
| 23 | <code> if b > 1</code> |
| 24 | <code> r = 0 ;</code> |
| 25 | <code> g = 0;</code> |
| 26 | <code> b = 0;</code> |
| 27 | <code> end</code> |
| 28 | <code> if m > 'z'</code> |
| 29 | <code> m = 'A';</code> |
| 30 | <code> end</code> |
| 31 | <code>end</code> |

Kode Sumber 4.8 Implementasi Tahap *Plot*

4.2.5 Implementasi Tahap Grafik

Program pembentukan grafik ini terpisah dari program yang lain. Pembentukan grafik dilakukan setelah hasil dari proses *Evidence Accumulation* selesai. Pembentukan grafik ini untuk mengetahui hubungan antara parameter dan hasil jumlah cluster yang didapatkan. Program pembentukan grafik ditunjukkan pada Kode Sumber 4.9.

| | |
|----|--|
| 1 | <code>clc</code> |
| 2 | <code>clear all</code> |
| 3 | <code>data =</code> <code>xlsread('grafikcoba.xlsx','Halfring');</code> |
| 4 | <code>x = data(:,1);</code> |
| 5 | <code>z = data(:,3);</code> |
| 6 | <code>y = data(:,2);</code> |
| 7 | <code>zbar = mean(z);</code> |
| 8 | <code>xgrid = linspace (-3, 13, N);</code> |
| 9 | <code>ygrid = linspace (-0.5, 1.5, N);</code> |
| 10 | <code>[XI, YI] = meshgrid(xgrid,ygrid);</code> |
| 11 | <code>ZI = griddata(x,y,z,XI,YI);</code> |
| 12 | <code>for I = 1:N;</code> |
| 13 | <code>for J = 1:N;</code> |
| 14 | <code>if isnan(ZI(I,J)) == 1</code> |
| 15 | <code>ZI(I,J) = zbar;</code> |
| 16 | <code>end</code> |
| 17 | <code>end</code> |
| 18 | <code>end</code> |
| 19 | <code>surf(XI,YI,ZI)</code> |
| 20 | <code>grid on;</code> |
| 21 | <code>light;</code> |
| 22 | <code>lighting phong;</code> |
| 23 | <code>camlight('left');</code> |

Kode Sumber 4.9 Program Pembentukan Grafik

BAB 5

UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisis dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, data uji coba, dan skenario uji coba yang meliputi uji perbandingan dan uji kinerja serta analisis setiap pengujian.

5.1 Lingkungan Uji Coba

Lingkungan uji coba yang digunakan dalam pembuatan Tugas Akhir mengimplementasi kinerja algoritma *clustering* dengan metode *Evidence Accumulation*. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor: Intel® Core™2 Duo CPU U9600 @ 1.60 GHz
 - b. Memory (RAM): 4 GB
 - c. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak
 - a. Sistem operasi: Windows 7 Professional
 - b. Perangkat pengembang: MATLAB 7.13.0 (R2011b)

5.2 Data Uji Coba

Pada bagian ini akan ditunjukkan data uji coba dengan beberapa *dataset* untuk menguji keakuratan solusi dari metode yang digunakan. Tabel 5.1 berikut ini menjelaskan tentang data uji. Data uji coba berjumlah delapan dengan jumlah data dan jumlah atribut yang bervariasi, mulai yang sederhana seperti data Iris sampai data yang rumit seperti data Vowel. Jumlah class juga bervariasi mulai dari 3 *cluster* sampai 11 *cluster*.

Tabel 5.1 Data Uji Coba

| No | Data set | Jumlah Data | Jumlah Atribut | Jumlah Cluster | |
|----|-----------|-------------|----------------|----------------|------------|
| | | | | Sebenarnya | \sqrt{n} |
| 1 | Half-ring | 400 | 2 | 2 | 20 |
| 2 | Spiral | 200 | 2 | 2 | 14 |
| 3 | Iris | 150 | 4 | 3 | 12 |
| 4 | Wine | 178 | 13 | 3 | 13 |
| 5 | Glass | 214 | 10 | 6 | 15 |
| 6 | Vowel | 990 | 10 | 11 | 31 |
| 7 | Ecoli | 336 | 8 | 8 | 18 |
| 8 | Seeds | 210 | 7 | 3 | 14 |

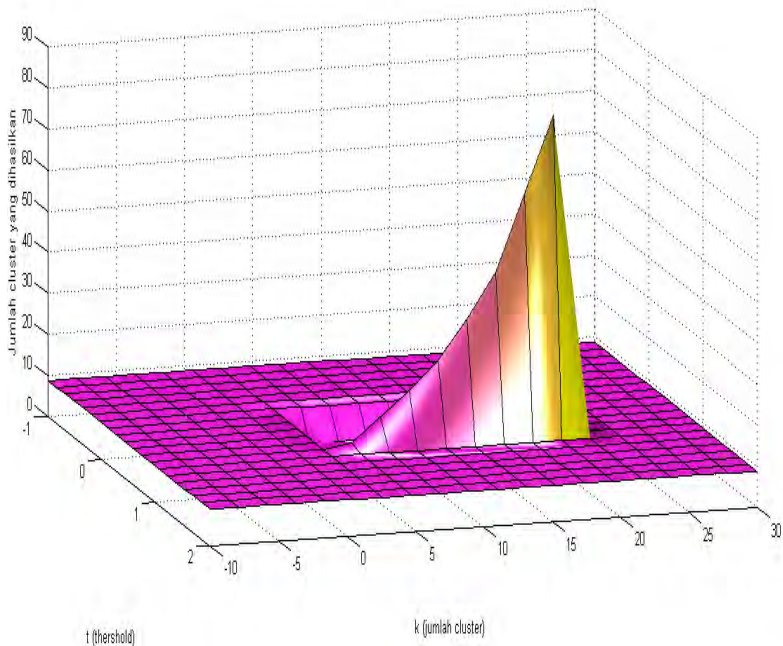
5.3 Skenario Uji Coba

Uji coba dilakukan untuk menguji apakah fungsionalitas aplikasi telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya. Uji coba akan didasarkan pada beberapa skenario untuk menguji kesesuaian dan kinerja aplikasi.

Skenario parameter yang diujikan terdiri dari 2 macam, yaitu k (jumlah *cluster*) dan t (*threshold*). Percobaan nilai yang diambil dari kedua parameter menunjukkan kinerja algoritma *clustering* dengan metode *Evidence Accumulation*. Parameter k (jumlah *cluster*) memiliki ukuran 3, 5, 10, 15, 20, 25, 30 dan \sqrt{n} . Ukuran \sqrt{n} merupakan akar dari jumlah data yang dimasukkan di awal. Nilai k (jumlah *cluster*) akan diproses pada metode *Evidence Accumulation*, dimana data akan disimpan di *co_assoc*. Parameter t (*threshold*) dari 0,1 - 0,9 bertujuan untuk membandingkan kesamaan pasangan data, yaitu *co_assoc*, apabila *co_assoc* lebih besar daripada t maka data akan digabungkan.

5.3.1 Uji Coba Halfring

Uji coba *Evidence Accumulation* pada *dataset* Halfring dilakukan dengan variasi parameter, yaitu k (jumlah *cluster*) dan t (*threshold*). Variasi parameter yang berpengaruh pada kinerja algoritma ditunjukkan dalam Gambar 5.1.

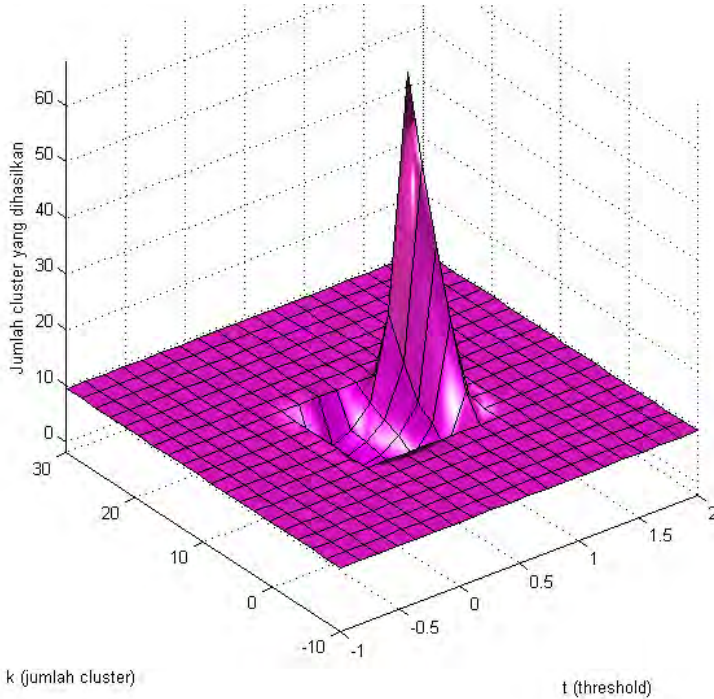


Gambar 5.1 Grafik Halfring

Pada percobaan *dataset* Halfring menunjukkan hasil *cluster* yang bervariasi, karena pengaruh parameter yang ada. Dari hasil uji coba terdapat *cluster* yang optimal yaitu $t = 0,1$ dan $k = 10$; $t = 0,1 - 0,3$ dan $k = 15$; $t = 0,1 - 0,2$ dan $k = 20$. Nilai optimal yang didapatkan pada Gambar 5.1 merupakan hasil perbandingan ilustrasi hasil plot data asli terhadap hasil plot data uji coba.

5.3.2 Uji Coba Spiral

Uji coba *Evidence Accumulation* pada *dataset* Spiral dilakukan dengan variasi parameter, yaitu k (jumlah *cluster*) dan t (*threshold*). Variasi parameter yang berpengaruh pada kinerja algoritma ditunjukkan dalam .

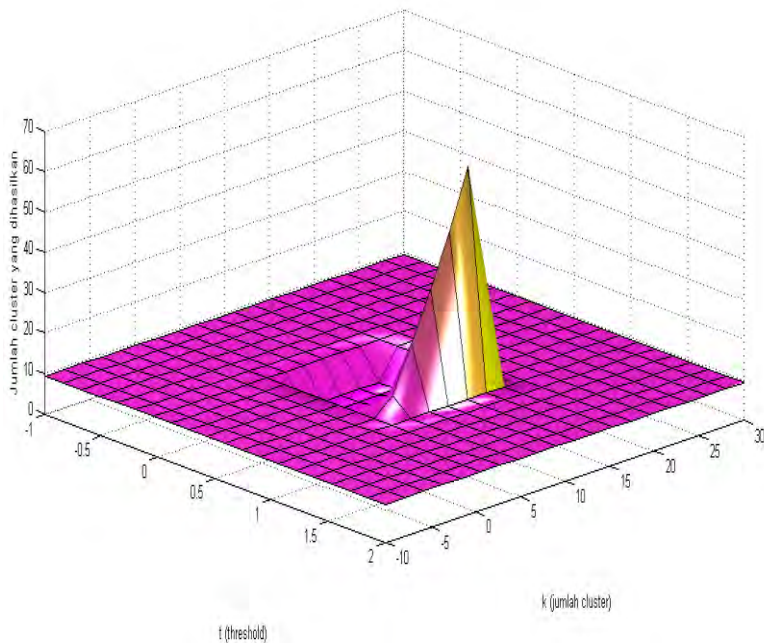


Gambar 5.2 Grafik Spiral

Pada percobaan *dataset* Spiral menunjukkan hasil *cluster* yang bervariasi, karena pengaruh parameter yang ada. Dari hasil uji coba terdapat *cluster* yang optimal yaitu $t = 0,7$ dan $k = 20$; $t = 0,6$ dan $k = 25$; $t = 0,5 - 0,6$ dan $k = 30$. Nilai optimal yang didapatkan pada Gambar 5.2 merupakan hasil perbandingan ilustrasi hasil plot data asli terhadap hasil plot data uji coba.

5.3.3 Uji Coba Iris

Uji coba *Evidence Accumulation* pada *dataset* Iris dilakukan dengan variasi parameter k (jumlah *cluster*) dan t (*threshold*). Variasi parameter yang berpengaruh pada kinerja algoritma ditunjukkan dalam Gambar 5.3.

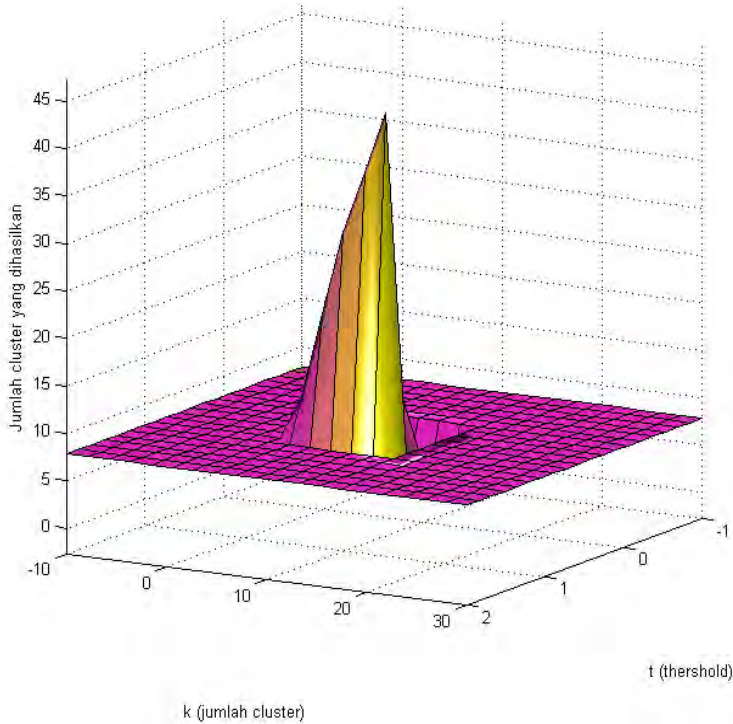


Gambar 5.3 Grafik Iris

Pada percobaan *dataset* Iris menunjukkan hasil *cluster* yang bervariasi, karena pengaruh parameter yang ada. Dari hasil uji coba terdapat *cluster* yang optimal yaitu $t = 0,7 - 0,8$ dan $k = 3$. Nilai yang optimal pada *dataset* Iris diambil yang mendekati jumlah cluster sebenarnya. Nilai optimal yang didapatkan pada Gambar 5.3 merupakan hasil perbandingan ilustrasi hasil plot data asli terhadap hasil plot data uji coba.

5.3.4 Uji Coba Wine

Uji coba *Evidence Accumulation* pada *dataset* Wine dilakukan dengan variasi parameter k (jumlah *cluster*) dan t (*threshold*). Variasi parameter yang berpengaruh pada kinerja algoritma ditunjukkan dalam Gambar 5.4.

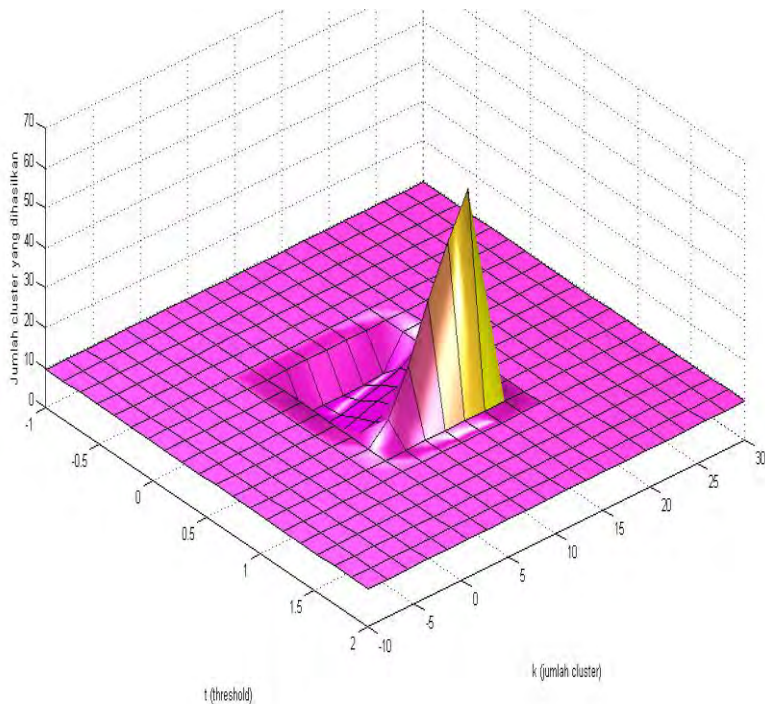


Gambar 5.4 Grafik Wine

Pada percobaan *dataset* Wine menunjukkan hasil *cluster* yang bervariasi, karena pengaruh parameter yang ada. Dari hasil uji coba terdapat *cluster* yang optimal yaitu $t = 0,2 - 0,8$ dan $k = 3$. Nilai optimal yang didapatkan pada Gambar 5.4 merupakan hasil perbandingan ilustrasi hasil plot data asli terhadap hasil plot data uji coba.

5.3.5 Uji Coba Glass

Uji coba *Evidence Accumulation* pada *dataset* Glass dilakukan dengan variasi parameter k (jumlah *cluster*) dan t (*threshold*). Variasi parameter yang berpengaruh pada kinerja algoritma ditunjukkan dalam Gambar 5.5.

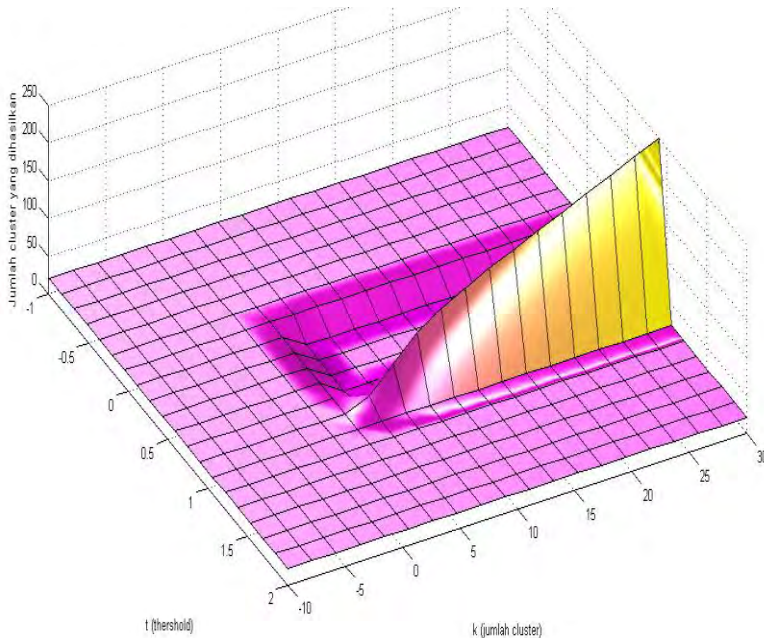


Gambar 5.5 Grafik Glass

Pada percobaan *dataset* Glass menunjukkan hasil *cluster* yang bervariasi, karena pengaruh parameter yang ada. Dari hasil uji coba terdapat *cluster* yang optimal yaitu $t = 0,4$ dan $k = 10$. Nilai optimal yang didapatkan pada Gambar 5.5 merupakan hasil perbandingan ilustrasi hasil plot data asli terhadap hasil plot data uji coba.

5.3.6 Uji Coba Vowel

Uji coba *Evidence Accumulation* pada *dataset* Vowel dilakukan dengan variasi parameter k (jumlah *cluster*) dan t (*threshold*). Variasi parameter yang berpengaruh pada kinerja algoritma ditunjukkan dalam Gambar 5.6.

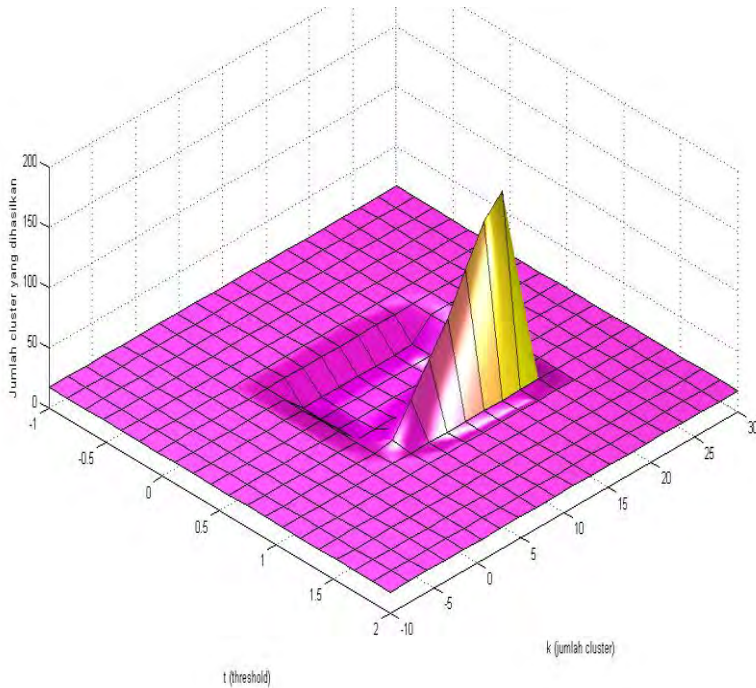


Gambar 5.6 Grafik Vowel

Pada percobaan *dataset* Vowel menunjukkan hasil *cluster* yang bervariasi, karena pengaruh parameter yang ada. Dari hasil uji coba terdapat *cluster* yang optimal yaitu $t = 0,6$ & $k = 31$. Nilai yang optimal pada *dataset* Vowel diambil yang mendekati jumlah cluster sebenarnya. Nilai optimal yang didapatkan pada Gambar 5.6 merupakan hasil perbandingan ilustrasi hasil plot data asli terhadap hasil plot data uji coba.

5.3.7 Uji Coba Ecoli

Uji coba *Evidence Accumulation* pada *dataset* Ecoli dilakukan dengan variasi parameter k (jumlah *cluster*) dan t (*threshold*). Variasi parameter yang berpengaruh pada kinerja algoritma ditunjukkan dalam Gambar 5.7.

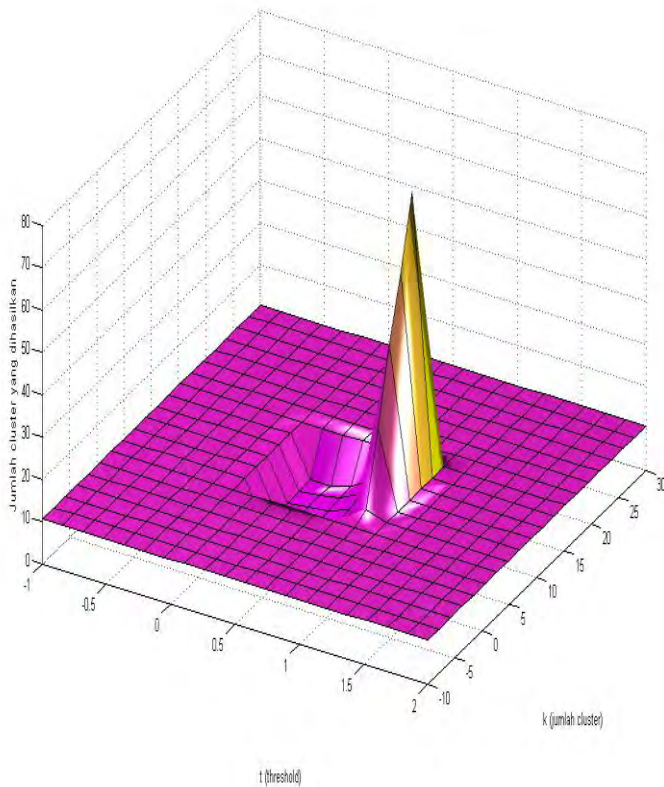


Gambar 5.7 Grafik Ecoli

Pada percobaan *dataset* Ecoli menunjukkan hasil *cluster* yang bervariasi, karena pengaruh parameter yang ada. Dari hasil uji coba terdapat *cluster* yang optimal yaitu $t = 0,7$ dan $k = 10$. Nilai yang optimal pada *dataset* Ecoli diambil yang mendekati jumlah cluster sebenarnya. Nilai optimal yang didapatkan pada Gambar 5.7 merupakan hasil perbandingan ilustrasi hasil plot data asli terhadap hasil plot data uji coba.

5.3.8 Uji Coba Seeds

Uji coba *Evidence Accumulation* pada *dataset* Seeds dilakukan dengan variasi parameter k (jumlah *cluster*) dan t (*threshold*). Variasi parameter yang berpengaruh pada kinerja algoritma ditunjukkan Gambar 5.8.



Gambar 5.8 Grafik Seeds

Pada percobaan *dataset* Seeds menunjukkan hasil *cluster* yang bervariasi, karena pengaruh parameter yang ada. Dari hasil uji coba terdapat *cluster* yang optimal yaitu $t = 0,6$ dan $k = 3$. Nilai optimal yang didapatkan pada Gambar 5.8 merupakan hasil perbandingan ilustrasi hasil plot data asli terhadap hasil plot data uji coba.

BAB 6

KESIMPULAN DAN SARAN

Bab ini berisi pembahasan mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, bab ini berisi saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1 Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap implementasi *Evidence Accumulation* dapat diambil kesimpulan sebagai berikut:

1. Metode *Evidence Accumulation* dan *Consensus Clustering* dapat menyelesaikan permasalahan *clustering*, tergantung pada parameter yang diujikan.
2. Parameter k (jumlah *cluster*) dan t (*threshold*) yang optimal setiap dataset memiliki nilai yang berbeda dengan jumlah *cluster* sebenarnya.
3. Semakin besar nilai t (*threshold*), maka jumlah cluster semakin besar.

6.2 Saran

Saran yang diberikan untuk pengembangan metode *Evidence Accumulation* pada Tugas Akhir ini adalah:

1. Pada metode *Evidence Accumulation* proses penyimpanan nilai `co_assoc` dapat direduksi.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] N. Nguyen and R. Caruana. (2007, Oct.) Consensus Clusterings. [Online]. <http://www.cs.cornell.edu/>
- [2] A. Lourenço, et al., "Consensus Clustering Using Partial Evidence Accumulation," *Pattern Recognition and Image Analysis*, pp. 69-78, 2013.
- [3] Michael. www.econ.upf.edu. [Online]. <http://www.econ.upf.edu/~michael/stanford/maeb7.pdf>
- [4] Acharya, Joydeep Ghosh and Ayan, "Cluster Ensembles," vol. 1, pp. 305-315, Jul. 2011.
- [5] Reza Ghaemi, Md. Nasir Sulaiman, Hamidah Ibrahim, Norwati Mustapha, "A Survey: Clustering Ensembles Techniques," *World Academy of Science*, vol. 26, pp. 636-645, 2009.
- [6] Youguo Li, Haiyan Wu, "A Clustering Method Based on K-Means Algorithm,," *International Conference on Solid State Devices and Materials Science*, vol. 25, pp. 1104-1109, 2012.
- [7] A. L. Fred, Member, IEEE, a. A. K. Jain, and Fellow, "Combining Multiple Clusterings Using Evidence Accumulation," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 27, Jun. 2005.
- [8] Wiley, "Minimum Spanning Trees and Single Linkage Analysis," *Journal of the Royal Statistical Society*. , vol. 18, pp. 54-64, 2013.

- [9] A. K. Jain and A. L. N. Fred, "Data Clustering Using Evidence Accumulation," *IEEE Pattern Recognition*, vol. 4, pp. 276-280, 2002.

BIODATA PENULIS



Ratih Kirana Diantari, lahir di Banjarbaru pada tanggal 17 Januari 1993. Penulis menempuh pendidikan mulai dari SD Pertamina Balikpapan (1998), SD Pucang III Sidoarjo (1998-2004), SMP Al-Muslim Surabaya (2004-2007), SMA Al-Falah Surabaya (2007-2010) dan S1 Teknik Informatika ITS (2010-2014).

Selama masa kuliah, penulis aktif di organisasi Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS.

Penulis dapat dihubungi melalui email: ratih.051@gmail.com

LAMPIRAN A

Pada Tabel A.1 sampai Tabel A.8 berisi tabel uji coba setiap *dataset* yang mempresentasikan semua nilai parameter dan jumlah *cluster* yang dihasilkan.

Tabel A.1 Uji Coba Halfring

| t/k | 3 | 5 | 10 | 15 | 20 |
|-----|---|----|----|----|----|
| 0,1 | 1 | 1 | 2 | 2 | 2 |
| 0,2 | 1 | 1 | 3 | 2 | 2 |
| 0,3 | 1 | 1 | 4 | 2 | 4 |
| 0,4 | 1 | 1 | 5 | 4 | 4 |
| 0,5 | 1 | 2 | 7 | 4 | 5 |
| 0,6 | 5 | 5 | 7 | 6 | 8 |
| 0,7 | 5 | 6 | 7 | 7 | 12 |
| 0,8 | 5 | 7 | 10 | 25 | 36 |
| 0,9 | 5 | 11 | 25 | 48 | 92 |

Tabel A.2 Uji Coba Spiral

| t/k | 3 | 5 | 10 | 14 | 15 | 20 | 25 | 30 |
|-----|---|----|----|-----|-----|-----|-----|-----|
| 0,1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0,2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0,3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0,4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0,5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| 0,6 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 |
| 0,7 | 1 | 1 | 1 | 1 | 1 | 2 | 5 | 4 |
| 0,8 | 1 | 1 | 4 | 11 | 17 | 28 | 42 | 64 |
| 0,9 | 2 | 32 | 77 | 115 | 124 | 150 | 158 | 176 |

Tabel A.3 Uji Coba Iris

| t/k | 3 | 5 | 10 | 12 | 15 |
|-----|---|----|----|----|----|
| 0,1 | 1 | 2 | 2 | 2 | 2 |
| 0,2 | 1 | 2 | 2 | 2 | 2 |
| 0,3 | 2 | 2 | 2 | 3 | 2 |
| 0,4 | 2 | 2 | 3 | 3 | 4 |
| 0,5 | 2 | 2 | 3 | 5 | 6 |
| 0,6 | 2 | 2 | 7 | 7 | 9 |
| 0,7 | 4 | 3 | 8 | 10 | 15 |
| 0,8 | 4 | 7 | 15 | 26 | 43 |
| 0,9 | 8 | 15 | 43 | 58 | 78 |

Tabel A.4 Uji Coba Wine

| t/k | 3 | 5 | 10 | 13 | 15 |
|-----|---|----|----|----|----|
| 0,1 | 1 | 1 | 1 | 1 | 1 |
| 0,2 | 3 | 1 | 1 | 2 | 1 |
| 0,3 | 3 | 1 | 1 | 3 | 4 |
| 0,4 | 3 | 1 | 2 | 4 | 4 |
| 0,5 | 3 | 1 | 3 | 5 | 6 |
| 0,6 | 3 | 4 | 5 | 7 | 7 |
| 0,7 | 3 | 6 | 9 | 10 | 12 |
| 0,8 | 3 | 9 | 18 | 21 | 28 |
| 0,9 | 5 | 14 | 35 | 44 | 53 |

Tabel A.5 Uji Coba Glass

| t/k | 15 | 3 | 5 | 10 | 15 |
|-----|----|----|----|----|----|
| 0,1 | 1 | 1 | 1 | 1 | 2 |
| 0,2 | 7 | 1 | 1 | 2 | 5 |
| 0,3 | 8 | 1 | 1 | 4 | 8 |
| 0,4 | 8 | 1 | 2 | 6 | 8 |
| 0,5 | 8 | 1 | 3 | 7 | 9 |
| 0,6 | 11 | 4 | 5 | 8 | 10 |
| 0,7 | 19 | 6 | 6 | 9 | 14 |
| 0,8 | 36 | 8 | 9 | 14 | 37 |
| 0,9 | 70 | 14 | 15 | 48 | 74 |

Tabel A.6 Uji Coba Vowel


| t/k | 3 | 5 | 10 | 15 | 31 |
|-----|---|----|-----|-----|-----|
| 0,1 | 1 | 1 | 1 | 1 | 1 |
| 0,2 | 1 | 1 | 1 | 1 | 1 |
| 0,3 | 1 | 1 | 1 | 1 | 1 |
| 0,4 | 1 | 1 | 1 | 1 | 2 |
| 0,5 | 1 | 1 | 1 | 1 | 4 |
| 0,6 | 3 | 1 | 1 | 2 | 15 |
| 0,7 | 6 | 5 | 7 | 20 | 56 |
| 0,8 | 6 | 10 | 40 | 63 | 118 |
| 0,9 | 9 | 41 | 115 | 154 | 261 |

Tabel A.7 Uji Coba Ecoli

| t/k | 3 | 5 | 10 | 15 | 18 |
|-----|---|----|----|-----|-----|
| 0,1 | 1 | 1 | 1 | 1 | 1 |
| 0,2 | 2 | 1 | 1 | 1 | 1 |
| 0,3 | 2 | 1 | 1 | 1 | 1 |
| 0,4 | 2 | 1 | 1 | 1 | 3 |
| 0,5 | 3 | 1 | 2 | 4 | 4 |
| 0,6 | 3 | 1 | 3 | 6 | 8 |
| 0,7 | 4 | 3 | 9 | 19 | 31 |
| 0,8 | 4 | 7 | 25 | 58 | 103 |
| 0,9 | 5 | 22 | 83 | 160 | 184 |

Tabel A.8 Uji Coba Seeds

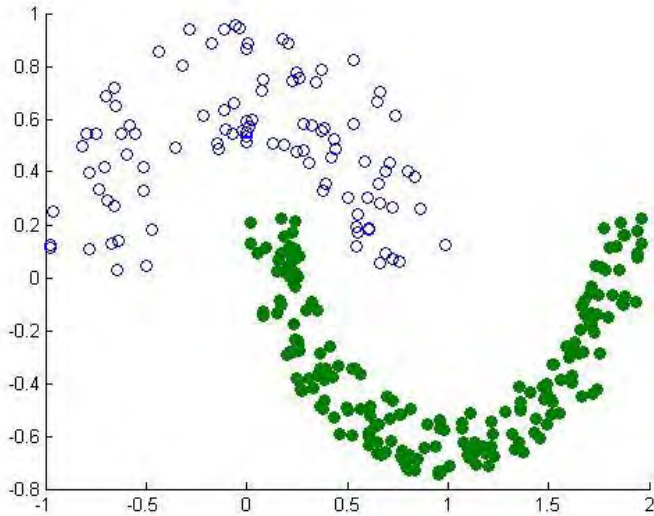
| t/k | 3 | 5 | 10 | 14 | 15 |
|-----|---|----|----|----|----|
| 0,1 | 2 | 1 | 1 | 1 | 1 |
| 0,2 | 2 | 1 | 1 | 1 | 1 |
| 0,3 | 2 | 1 | 2 | 1 | 1 |
| 0,4 | 3 | 1 | 2 | 2 | 2 |
| 0,5 | 3 | 2 | 3 | 5 | 4 |
| 0,6 | 3 | 2 | 5 | 6 | 7 |
| 0,7 | 4 | 2 | 9 | 16 | 18 |
| 0,8 | 4 | 7 | 19 | 34 | 41 |
| 0,9 | 4 | 14 | 55 | 86 | 94 |

Merah  : Tidak Optimal

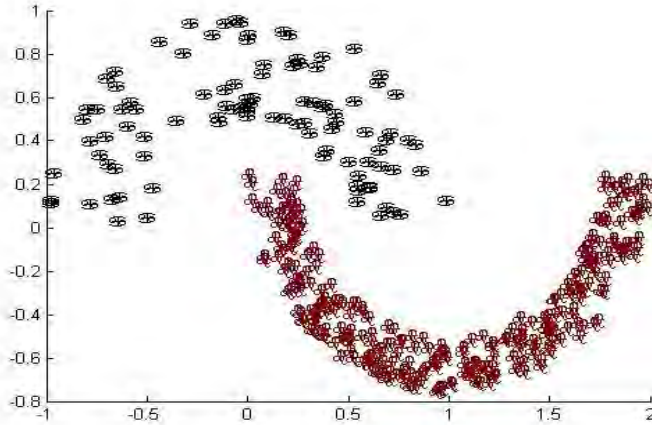
Hijau  : Optimal

LAMPIRAN B

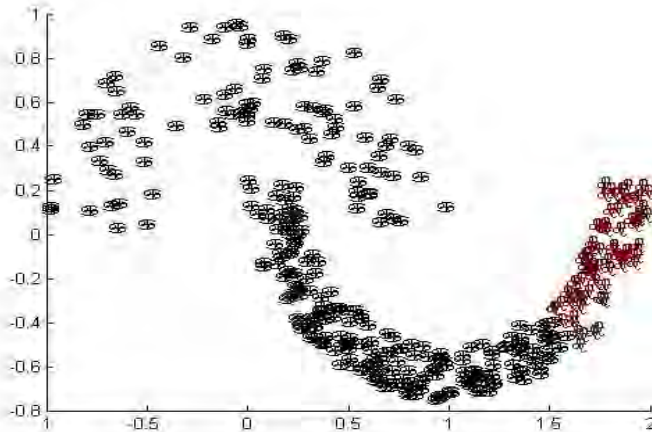
Pada Gambar B.1 sampai Gambar B.24 berisi gambar setiap *dataset* yang terdiri dari gambar plot data asli, gambar plot data uji coba yang optimal dan tidak optimal.



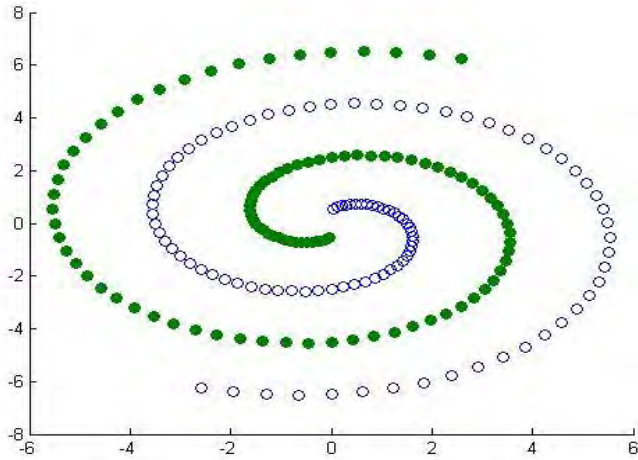
Gambar B.1 *Dataset Halfring*



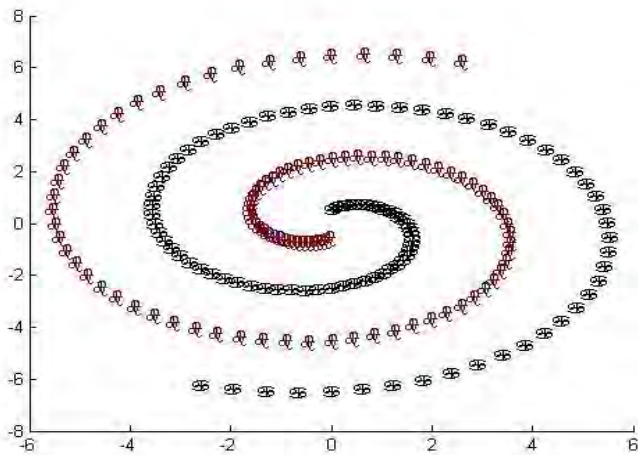
Gambar B.2 Uji Coba Halfring yang Optimal $k = 15$ & $t = 0,2$



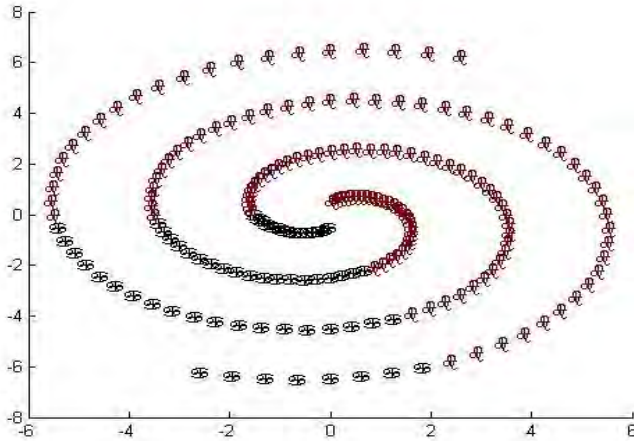
Gambar B.3 Uji Coba Halfring yang Tidak Optimal $k = 5$ & $t = 0,5$



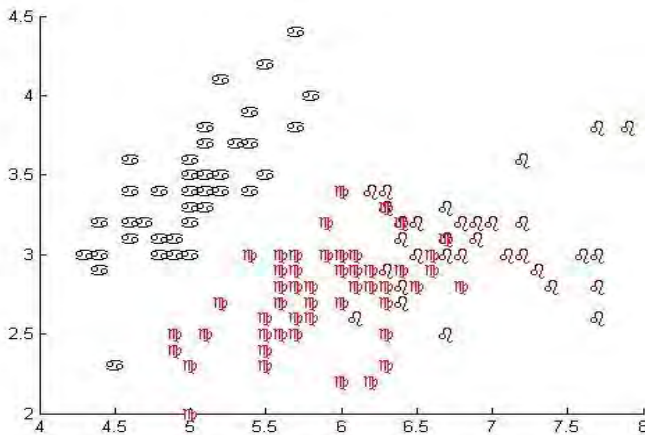
Gambar B.4 *Dataset* Spiral



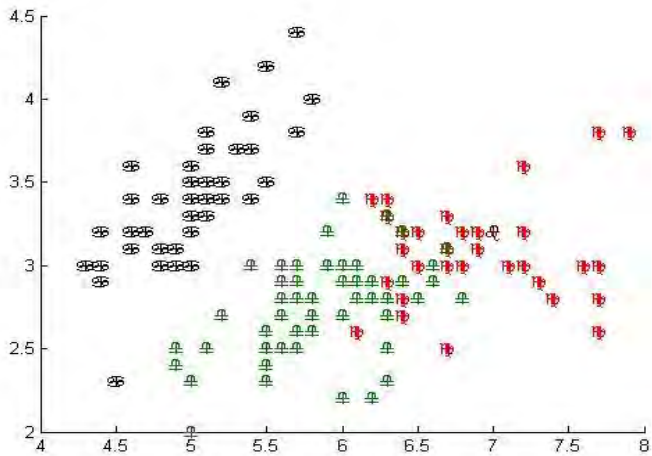
Gambar B.5 Uji Coba Spiral yang Optimal $k = 20$ & $t = 0,7$



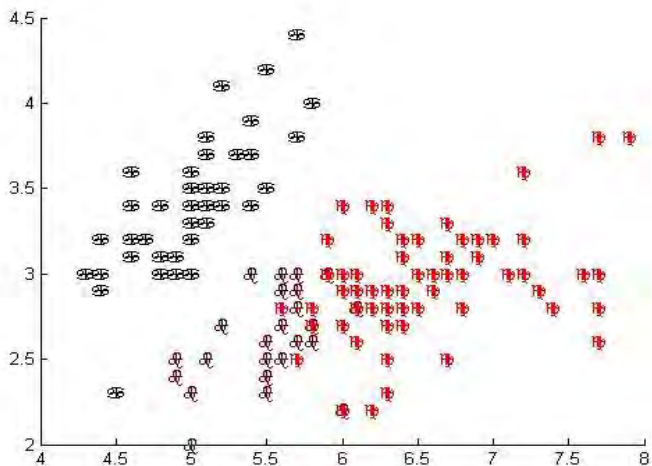
Gambar B.6 Uji Coba Spiral yang Tidak Optimal $k = 3$ & $t = 0,9$



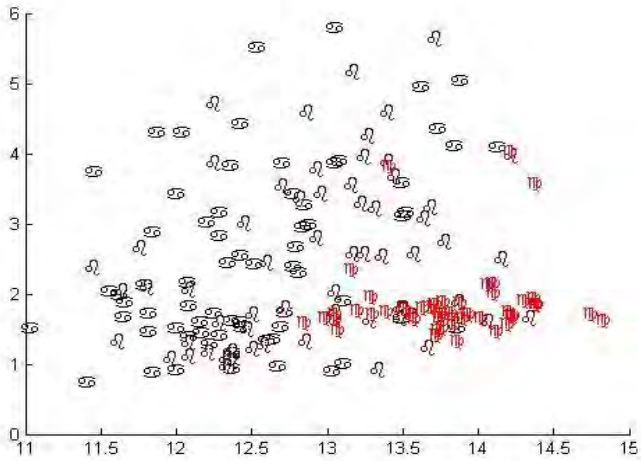
Gambar B.7 Dataset Iris



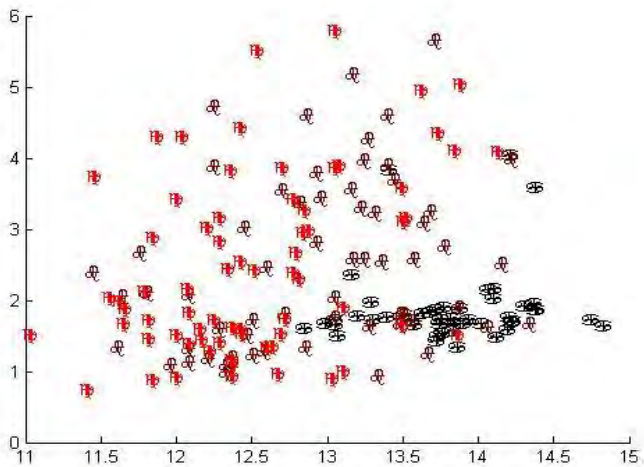
Gambar B.8 Uji Coba Iris yang Optimal $k = 3$ & $t = 0,7$



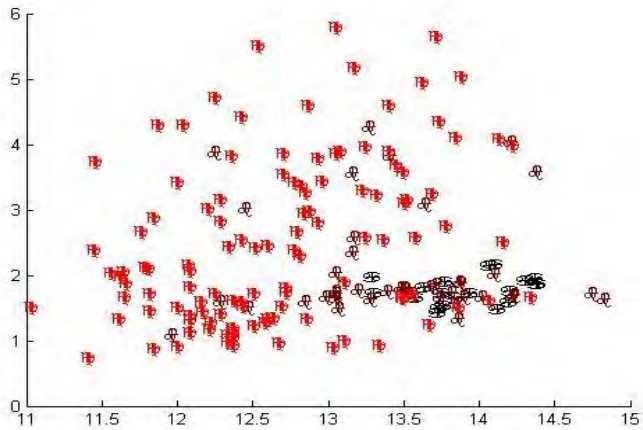
Gambar B.9 Uji Coba Iris yang Tidak Optimal $k = 5$ & $t = 0,7$



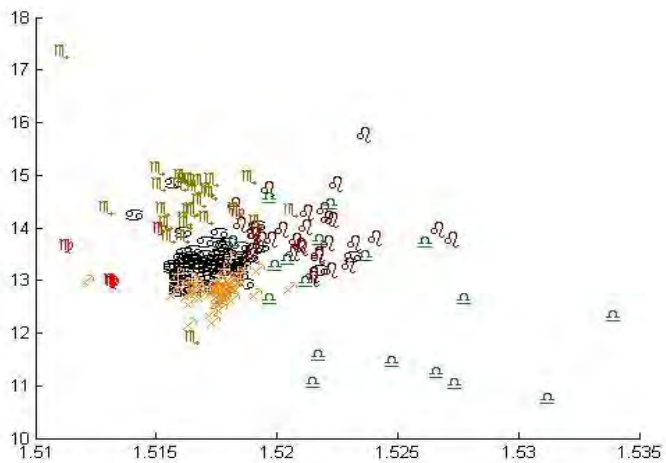
Gambar B.10 *Dataset Wine*



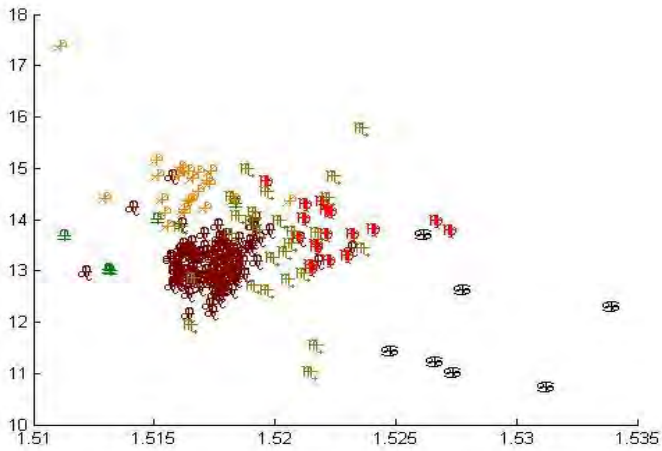
Gambar B.11 Uji Coba Wine yang Optimal $k = 3$ & $t = 0,2$



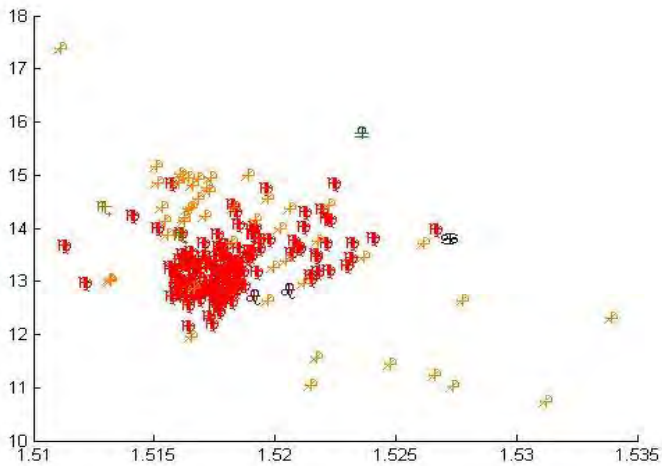
Gambar B.12 Uji Coba Wine yang Tidak Optimal $k = 10$ & $t = 0,5$



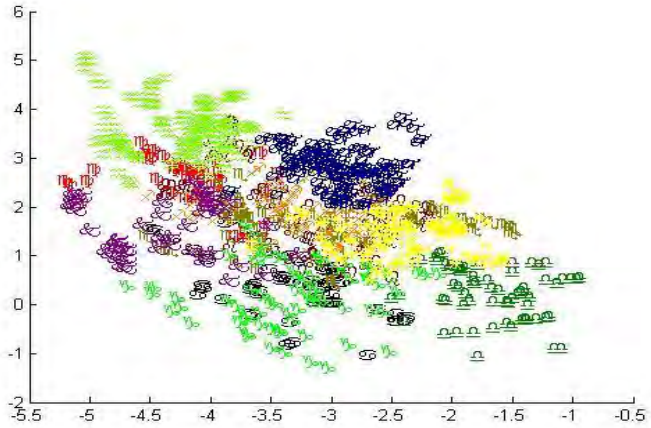
Gambar B.13 Dataset Glass



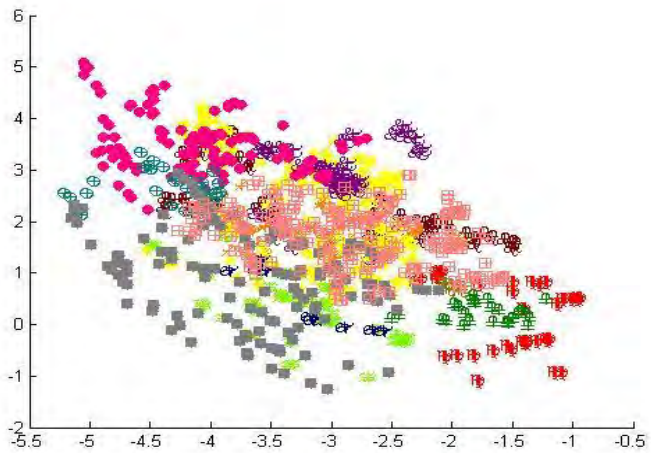
Gambar B.14 Uji Coba Glass yang Optimal $k = 10$ & $t = 0,4$



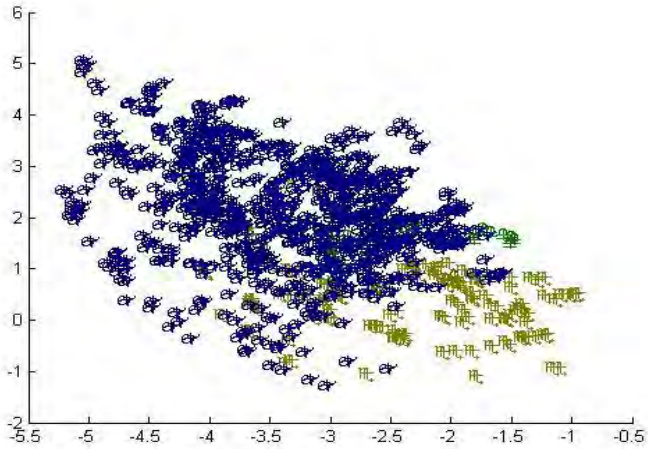
Gambar B.15 Uji Coba Glass yang Tidak Optimal $k = 3$ & $t = 0,7$



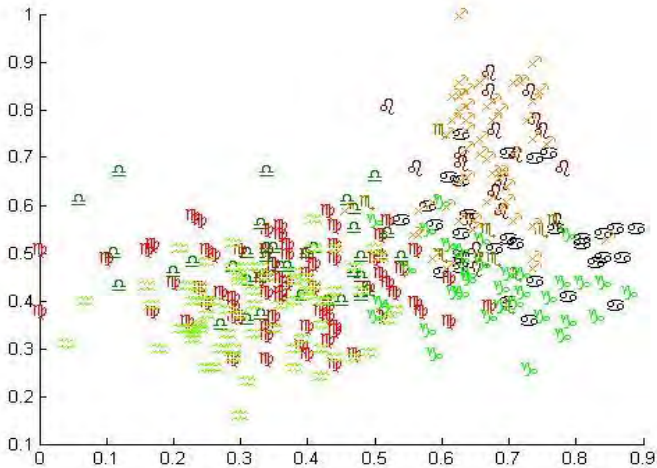
Gambar B.16 Dataset Vowel



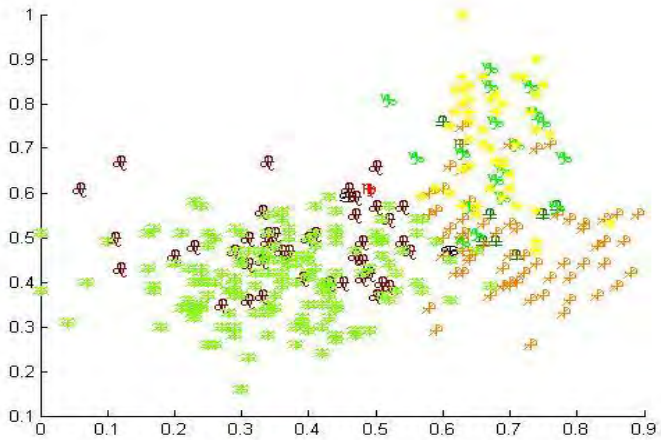
Gambar B.17 Uji Coba Vowel yang Optimal $k = 31$ & $t = 0,6$



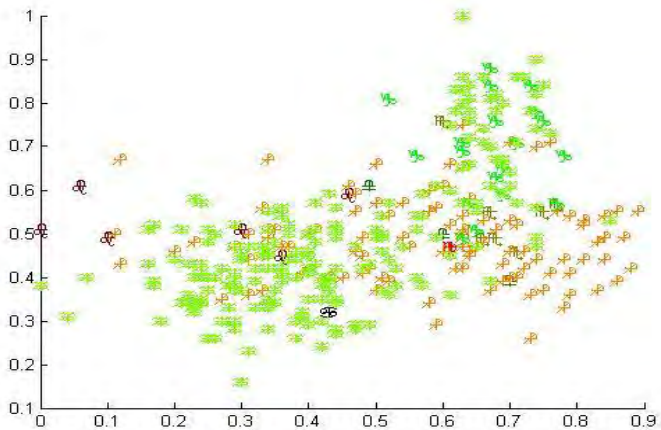
Gambar B.18 Uji Coba Vowel yang Tidak Optimal $k = 5$ & $t = 0,8$



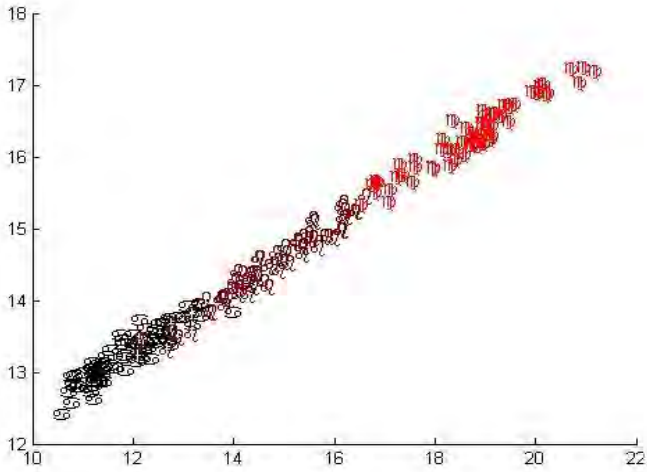
Gambar B.19 Dataset Ecoli



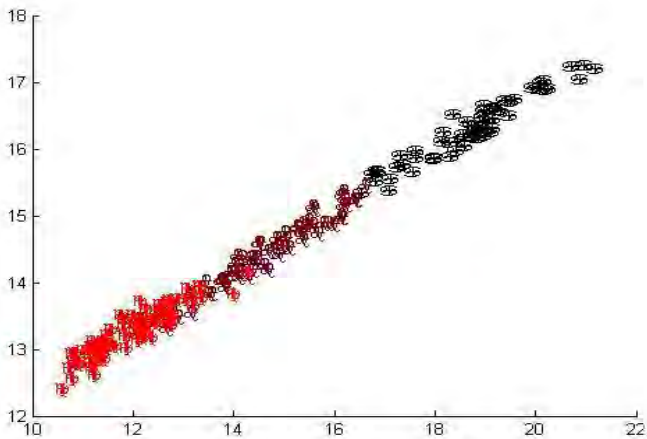
Gambar B.20 Uji Coba Ecoli yang Optimal $k = 10$ & $t = 0,7$



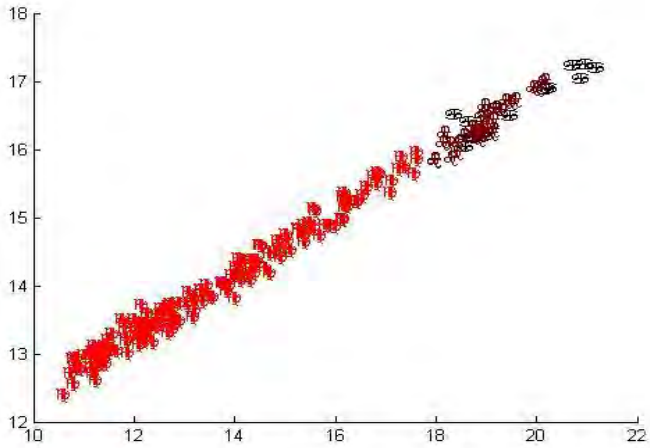
Gambar B.21 Uji Coba Ecoli yang Tidak Optimal $k = 18$ & $t = 0,6$



Gambar B.22 Dataset Seeds



Gambar B.23 Uji Coba Seeds yang Optimal $k = 3$ & $t = 0,6$



Gambar B.24 Uji Coba Seeds yang Tidak Optimal $k = 10$ & $t = 0,5$