

Pengaruh *Singular Value Decomposition* Terhadap Metode-Metode
Clustering dalam Pengelompokkan Dokumen Berdimensi Tinggi

*Diajukan Sebagai Syarat Untuk Menyelesaikan
Pendidikan Program Strata-1 Pada
Jurusan Teknik Informatika*



Oleh:

Novita Hidayati
NIM : 09021181320030

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA
2017**

LEMBAR PENGESAHAN TUGAS AKHIR

PENGARUH SINGULAR VALUE DECOMPOSITION TERHADAP METODE-METODE CLUSTERING DALAM PENGELOMPOKAN DOKUMEN BERDIMENSI TINGGI

Oleh :

NOVITA HIDAYATI
NIM : 09021181320030

Indralaya, 17 Oktober 2017

Pembimbing I,

Pembimbing II,

Ir. M. Ihsan Jambak, M.Sc.
NIP. 196804052013081201

Danny Matthew Saputra, M.T
NIP. 198505102015041002

Mengetahui,
Dekan Fakultas Ilmu Komputer

Jaidan Jauhari, S.Pd., M.T.
NIP. 197107212005011005

TANDA LULUS UJIAN SIDANG SKRIPSI

Pada hari Selasa, 17 Oktober 2017 telah dilaksanakan ujian sidang tugas akhir oleh Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya.

Nama : Novita Hidayati
NIM : 09021181320030
Judul : Pengaruh Singular Value Decomposition Terhadap Metode-
Metode Clustering dalam Pengelompokan Dokumen Berdimensi
Tinggi

1. Ketua Penguji

Ir. M. Ihsan Jambak, M.Sc.
NIP. 196804052013081201

2. Sekretaris Penguji

Danny Matthew Saputra, M.T
NIP. 198505102015041002

3. Penguji I

M. Fachrurrozi, S.Si, M.T
NIP. 198005222008121002

4. Penguji II

Osvari Arsalan, M.T
NIP.

Mengetahui,
Ketua Jurusan Teknik Informatika

Rifkie Primartha, M.T
NIP. 197706012009121004

HALAMAN PERNYATAAN

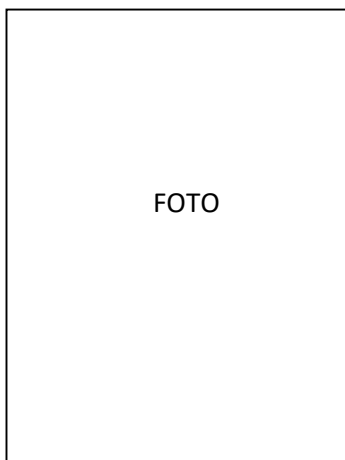
Yang bertanda tangan di bawah ini :

Nama : Novita Hidayati
NIM : 09021181320030
Program Studi : Teknik Informatika
Judul Skripsi : Pengaruh *Singular Value Decomposition* Terhadap
Metode-Metode Clustering dalam Pengelompokkan
Dokumen Berdimensi Tinggi
Hasil Pengecekan Software *iThenticate/Turnitin* : **8 %**

Menyatakan bahwa Laporan Projek saya merupakan hasil karya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan projek ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya sesuai dengan ketentuan yang berlaku.

Demikian, pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan oleh siapapun.

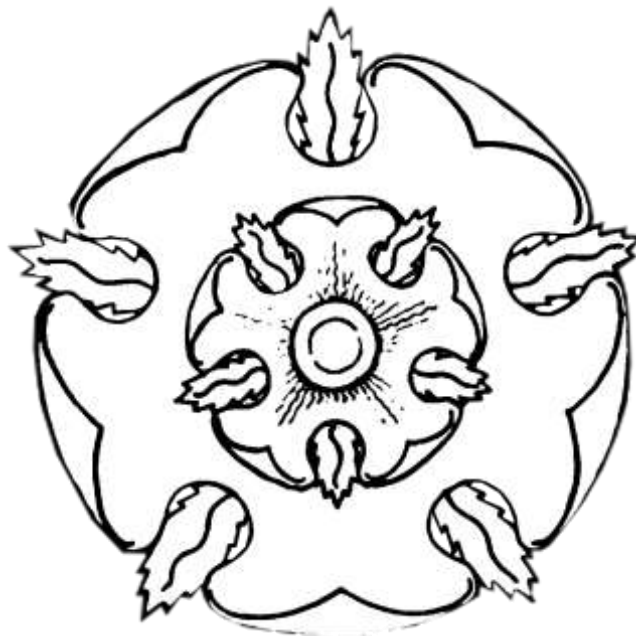
Indralaya, 01 November 2017



(Novita Hidayati)
NIM. 09021181320030

GROWING STRONG

HOUSE TYRELL



GAME OF THRONES

THE EFFECT OF SINGULAR VALUE DECOMPOSITION IN VARIOUS CLUSTERING METHODS ON HIGH DIMENSIONAL DATA CLUSTERING

by:

Novita Hidayati
09021181320030

ABSTRACT

Documents that have a large scale of attributes or dimension can be an obstacle in a clustering process. Most clustering algorithms are good at handling low-dimensional data. Finding clusters of data objects in a high-dimensional space is challenging, because data can be very sparse and highly skewed. Therefore, this research implements Singular Value Decomposition (SVD) to reduce the high dimension and analyses its impact on clustering methods k-means, k-medoids, and fuzzy c-means. This research shows that combining SVD with k-means and k-medoids increases both accuracy result compared with clustering without dimension reduction. The combination of SVD and k-means and k-medoids increases accuracy result by 10-11% compared to clustering without dimensional reduction. The execution time is also proven to be faster. On the contrary, the combination between SVD and fuzzy c-means shows a lower accuracy result compared with clustering without dimension reduction.

Kata Kunci: dimension reduction, *singular value decomposition*, *k-means clustering*, *k-medoids clustering*, *fuzzy c-means clustering*.

Indralaya, 17 Oktober 2017

Pembimbing I,

Pembimbing II,

Ir. M. Ihsan Jambak, M.Sc.

NIP. 196804052013081201

Danny Matthew Saputra, M.T

NIP. 198505102015041002

Mengetahui,
Ketua Jurusan Teknik Informatika

Rifkie Primartha, M.T
NIP. 1997706012009121004

PENGARUH SINGULAR VALUE DECOMPOSITION TERHADAP
METODE-METODE CLUSTERING DALAM PENGELOMPOKKAN
DOKUMEN BERDIMENSI TINGGI

Oleh:
Novita Hidayati
09021181320030

ABSTRAK

Dokumen yang mengandung banyak atribut atau dimensi dapat menjadi hambatan dalam proses clustering. Kebanyakan algoritma clustering baik dalam menangani data berdimensi rendah, sehingga untuk menemukan cluster dari objek data berdimensi tinggi adalah sebuah tantangan karena data berdimensi tinggi cenderung memiliki *noise*, *sparse*, dan bersifat *skew*. Dalam *data mining*, reduksi dimensi dapat dilakukan untuk meningkatkan akurasi hasil clustering data berdimensi tinggi. Untuk itu, Penelitian ini mengkombinasikan Singular Value Decomposition (SVD) untuk mereduksi dimensi dan melihat pengaruhnya terhadap akurasi hasil clustering metode k-means, k-medoids, dan fuzzy c-means. Penelitian menunjukkan bahwa kombinasi SVD dan k-means dan k-medoids meningkatkan hasil akurasi sebesar 10-11 % dibandingkan dengan clustering tanpa reduksi dimensi. Waktu komputasi proses clustering menjadi lebih cepat. Namun, sebaliknya kombinasi SVD dan fuzzy c-means menunjukkan hasil akurasi clustering yang lebih buruk dibandingkan dengan clustering tanpa reduksi dimensi.

Kata Kunci: Reduksi Dimensi, *Singular Value Decomposition*, *K-Means*, *K-Medoids*, *Fuzzy C-Means*, Dimensi Tinggi.

Indralaya, 17 Oktober 2017

Pembimbing I,

Pembimbing II,

Ir. M. Ihsan Jambak, M.Sc.

Danny Matthew Saputra, M.T

NIP. 196804052013081201

NIP. 198505102015041002

Mengetahui,
Ketua Jurusan Teknik Informatika

Rifkie Primartha, M.T
NIP. 1997706012009121004

KATA PENGANTAR



Puji syukur kepada Allah swt. atas berkat dan rahmat-Nya yang telah diberikan kepada Penulis sehingga dapat menyelesaikan Tugas Akhir ini dengan baik. Tugas akhir ini disusun untuk memenuhi salah satu syarat guna menyelesaikan pendidikan program Strata-1 pada Fakultas Ilmu Komputer Program Studi Teknik Informatika di Universitas Sriwijaya.

Dalam menyelesaikan Tugas Akhir ini banyak pihak yang telah memberikan bantuan dan dukungan baik secara langsung maupun secara tidak langsung. Untuk itu Penulis ingin menyampaikan rasa terima kasih kepada:

1. Orang tuaku, Kailani Akhmad (Almarhum) dan Jamilah, saudaraku, Firmansyah, Marlina, Tri Mulyani, Meirani Jayanti, dan Firsty Herawati dan seluruh keluarga beasrku yang selalu mendokan serta memberikan dukungan baik moril maupun materil
2. Bapak Jaidan Jauhari, S.Pd., M.T. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya
3. Bapak Rifkie Primartha, S.T., M.T. selaku Ketua Jurusan Teknik Informatika
4. Bapak Ir. M. Ihsan Jambak, M.Sc. selaku dosen pembimbing I dan Bapak Danny Matthew Saputra, M.T selaku dosen pembimbing II yang telah memberikan arahan serta dukungan dalam proses pengerjaan Tugas Akhir
5. Bapak M. Fachrurrozi, S.Si, M.T selaku dosen penguji I dan Bapak Osvari Arsalan, M.T selaku dosen penguji II yang telah memberikan masukan dan dorongan dalam proses pengerjaan Tugas Akhir
6. Bapak Syamsuryadi, S.Si., M.Kom., Ph.D. selaku dosen pembimbing akademik

7. Seluruh dosen Program Studi Teknik Informatika dan staf tata usaha Fakultas Ilmu Komputer Universitas Sriwijaya yang telah membantu dalam kelancaran proses administrasi dan akademik selama masa perkuliahan
8. Dininta Isnurthina, sahabatku yang selalu ada buatku dalam suka maupun duka dari semester 1 hingga penulis menyelesaikan studi
9. Keluarga besar Bapak Prof. Dr. Ir. Andy Mulyana, M.Sc. dan Ibu Dr. Lifianthi, S.P., M.Si. , nenek, ayuk, dan Rahma yang telah Penulis anggap sebagai keluarga sendiri
10. EC Techphoria 2015, Taya, Lala, Icil, Onik, Ningrum, Dini, dan Mamat, yang selalu mengerti, mendukung, dan mewarnai hidup Penulis
11. Tim PKPA, Latifah, Lala, Suwar,Koko Alvin, Niu dan Dininta yang telah memotivasi dan berjuang bersama Penulis selama masa perkuliahan
12. Keluargaku di BEM KM Fakultas Ilmu Komputer Universitas, DPM KM Fakultas Ilmu Komputer Universitas, INTEL, dan Ikatan Bujang Gadis Fakultas Ilmu Komputer Universitas yang telah memberikan ruang bagi Penulis untuk berprestasi dan berkarya.
13. Keluargaku, seluruh anggota IF Reguler 2013, terutama Rachmad Algani yang secara tidak langsung menghibur Penulis melalui gurauannya.
14. Semua pihak yang tidak bisa Penulis sebutkan satu persatu yang telah membantu dan berperan dalam Tugas Akhir ini.

Penulis menyadari dalam penyusunan Tugas Akhir ini masih terdapat banyak kekurangan disebabkan keterbatasan pengetahuan dan pengalaman, oleh karena itu kritik dan saran yang membangun sangat diharapkan untuk kemajuan penelitian selanjutnya.

Akhir kata dengan segala kerendahan hati, semoga Tugas Akhir ini dapat berguna dan bermanfaat bagi kita semua.

Indralaya, Oktober 2017

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERSETUJUAN KOMISI PENGUJI	iii
HALAMAN PERNYATAAN PLAGIAT	iv
HALAMAN MOTTO DAN PERSEMBAHAN.....	v
ABSTRACT.....	vi
ABSTRAKSI	viii
KATA PENGANTAR	x
DAFTAR TABEL.....	xviii
DAFTAR GAMBAR	xxi
DAFTAR LAMPIRAN	
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah.....	I-1
1.2 Pendahuluan	I-1
1.3 Rumusan Masalah	I-3
1.4 Tujuan Penelitian	I-4
1.5 Manfaat Penelitian	I-4
1.6 Batasan Masalah.....	I-5
1.7 Sistematika Penulisan	I-6

BAB II KAJIAN LITERATUR

2.1 Pendahuluan	II-1
2.2 K-Means	II-1
2.3 K-Medoids.....	II-3
2.4 Fuzzy C-Means	II-4
2.5 Singular Value Decomposition	II-6
2.6 Penelitian Lain yang Relevan.....	II-9

BAB III METODOLOGI PENELITIAN

3.1 Pendahuluan	III-1
3.2 Unit Penelitian.....	III-1
3.3 Metode Pengumpulan Data	III-1
3.4 Tahapan Penelitian	III-1
3.4.1 Menentukan Ruang Lingkup dan Unit Penelitian	III-2
3.4.2 Menemukan Dasar Teori yang Berkaitan dengan Penelitian ...	III-2
3.4.3 Menetapkan Kriteria Pengujian	III-2
3.4.4 Menentukan Alat yang Digunakan dalam Pelaksanaan Penelitian	III-4
3.4.5 Melakukan Pengujian Penelitian	III-5
3.4.6 Melakukan Analisa Hasil Pengujian dan Membuat Kesimpulan Penelitian	III-7
3.5 Metode Pengembangan Perangkat Lunak	III-9
3.5.1 Fase Insepsi	III-10
3.5.2 Fase Elaborasi.....	III-9
3.5.3 Fase Konstruksi	III-11

3.5.4 Fase Transisi	III-11
3.6 Penjadwalan Penelitian	III-12

BAB IV PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan	IV-1
4.2 Fase Insepsi	IV-1
4.2.1 Pemodelan Bisnis	IV-1
4.2.2 Kebutuhan Sistem.....	IV-1
4.2.3 Analisis dan Desain	IV-1
4.2.3.1 Analisis Perangkat Lunak	IV-1
a. Analisis Kebutuhan Perangkat Lunak	IV-1
b. Analisis Data	IV-1
c. Analisis Prapengolahan	IV-1
d. Analisis Pembobotan Kata	IV-1
e. Analisis Metode <i>Singular Value Decomposition</i>	IV-1
f. Analisis Metode <i>K-means</i>	IV-1
g. Analisis Metode <i>K-medoids</i>	IV-1
h. Analisis Metode <i>Fuzzy C-Means</i>	IV-1
4.2.3.2 Desain Perangkat Lunak	IV-1
1. Model <i>Use Case</i>	IV-1
2. Diagram Aktivitas	IV-1
4.3 Fase Elaborasi	IV-1
4.3.1 Pemodelan Bisnis	IV-1
4.3.1.1 Perancangan Data.....	IV-1

4.3.1.2 Perancangan Antar Muka.....	IV-1
4.3.2 Kebutuhan Sistem	IV-1
4.3.3 Diagram Sequence.....	IV-1
4.4 Fase Konstruksi.....	IV-1
4.4.1 Kebutuhan Sistem	IV-1
4.4.2 Diagram Kelas.....	IV-1
4.4.3 Implementasi	IV-1
4.4.3.1 Implementasi Kelas	IV-1
4.4.3.2 Implementasi Antarmuka.....	IV-1
4.5 Fase Transisi.....	IV-1
4.5.1 Pemodelan Bisnis	IV-1
4.5.2 Kebutuhan Sistem	IV-1
4.5.3 Rencana Pengujian	IV-1
4.5.3.1 Rencana Pengujian Use Case Memasukkan Dokumen	IV-1
4.5.3.2 Rencana Pengujian Use Case Melakukan Clustering dengan K-means	IV-1
4.5.3.3 Rencana Pengujian Use Case Melakukan Clustering dengan K-medoids	IV-1
4.5.3.4 Rencana Pengujian Use Case Melakukan Clustering dengan Fuzzy C-Means	IV-1
4.5.3.5 Rencana Pengujian Use Case Melakukan Clustering dengan K-means dan SVD.....	IV-1

4.5.3.6 Rencana Pengujian Use Case Melakukan Clustering dengan K-medoids dan SVD	IV-1
4.5.3.7 Rencana Pengujian Use Case Melakukan Clustering dengan Fuzzy C-Means dan SVD.....	IV-1
4.5.4 Implementasi	IV-1
4.5.4.1 Pengujian Use Case Memasukkan Dokumen	IV-1
4.5.4.2 Pengujian Use Case Melakukan Clustering dengan K-means	IV-1
4.5.4.3 Pengujian Use Case Melakukan Clustering dengan K-medoids.....	IV-1
4.5.4.4 Pengujian Use Case Melakukan Clustering dengan Fuzzy C-Means	IV-1
4.5.4.5 Pengujian Use Case Melakukan Clustering dengan K-Means dan SVD.....	IV-1
4.5.4.6 Pengujian Use Case Melakukan Clustering dengan K-medoids dan SVD	IV-1
4.5.4.7 Pengujian Use Case Melakukan Clustering dengan Fuzzy C-Means dan SVD	IV-1

BAB V HASIL DAN ANALISIS PENELITIAN

5.1 Pendahuluan	V-1
5.2 Hasil Percobaan Penelitian.....	V-1
5.2.1 Hasil Clustering K - Means	V-1
5.2.2 Hasil Clustering K - Means dan Singular Value Decomposition	V-1

5.2.3 Hasil Clustering K - Medoids.....	V-1
5.2.4 Hasil Clustering K - Medoids dan Singular Value Decomposition	V-1
5.2.5 Hasil Clustering Fuzzy C-Means	V-1
5.2.6 Hasil Clustering Fuzzy C-Means dan Singular Value Decomposition	V-1
5.3 Tabel Confussion	V-1
5.4 Nilai Precision, Recall, F-Measure, dan Rand Index	V-1
5.5 Waktu Komputasi dan Iterasi.....	V-1
5.6 Analisis Penelitian.....	V-1

BAB VI KESIMPULAN DAN SARAN

6.1 Pendahuluan	VI-1
6.2 Kesimpulan	VI-1
6.3 Saran.....	VI-2

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Halaman

III-1. Tabel Rancangan Hasil Cluster.....	III-5
III-2. Tabel Rancangan Perhitungan F-Measure, Recall dan Precision untuk Tiap Kategori	III -6
III-3. Tabel Rancangan Hasil Waktu Eksekusi Program	III-7
III-4. Tabel Penjadwalan Penelitian dalam Bentuk <i>Work Breakdown Structure</i> (WBS)	III-13
IV-1. Tabel Kebutuhan Fungsional.....	IV-4
IV-2. Tabel Kebutuhan Non Fungsional.....	IV-5
IV-3. Tabel Hasil <i>Tokenizing</i> Contoh Dokumen	IV-9
IV-4. Tabel Hasil <i>Stop Words Removal</i> Contoh Dokumen.....	IV-10
IV-5. Tabel Hasil <i>Stemming</i> Contoh Dokumen	IV-11
IV-6. Tabel Hasil <i>Pembobotan</i> Contoh Dokumen.....	IV-12
IV-7. Tabel Contoh Perhitungan <i>Singular Value Decomposition</i>	IV-14
IV-8. Tabel Hasil Clustering Kalimat menggunakan k-means	IV-19
IV-9. Tabel Hasil Pusat Cluster menggunakan k-means	IV-19
IV-10. Tabel Hasil Clustering Kalimat menggunakan k-medoids.....	IV-20
IV-11. Tabel Hasil Clustering Kalimat menggunakan fuzzy c-means	IV-20
IV-12. Tabel Hasil Pusat Cluster menggunakan fuzzy c-means.....	IV-21
IV-13. Tabel Definisi Aktor <i>Use Case</i>	IV-23
IV-14. Tabel Definisi <i>Use Case</i>	IV-24
IV-15. Tabel Skenario Use Case Memasukkan Dokumen.....	IV-25
IV-16. Tabel Skenario Use Case Melakukan Clustering dengan K-Means.....	IV-26
IV-17. Tabel Skenario Use Case Melakukan Clustering dengan K-Means dan SVD	IV-29
IV-18. Tabel Skenario Use Case Melakukan Clustering dengan K-Medoids	IV- 31
IV-19. Tabel Skenario Use Case Melakukan Clustering dengan K-Medoids dan SVD.....	IV-33

IV-20. Tabel Skenario Use Case Melakukan Clustering dengan Fuzzy C-Means	IV-35
IV-21. Tabel Skenario Use Case Melakukan Clustering dengan Fuzzy C-Means dan SVD	IV-37
IV-22. Tabel Implementasi Kelas	IV-60
IV-23. Tabel Rencana Pengujian <i>Use Case</i> Memasukkan Dokumen.....	IV-67
IV-24. Tabel Rencana Pengujian <i>Use Case</i> Melakukan Clustering dengan K-Means	IV-68
IV-25. Tabel Rencana Pengujian <i>Use Case</i> Melakukan Clustering dengan K-Means dan SVD	IV-69
IV-26. Tabel Rencana Pengujian <i>Use Case</i> Melakukan Clustering dengan K- Medoids.....	IV-70
IV-27. Tabel Rencana Pengujian <i>Use Case</i> Melakukan Clustering dengan K-Means dan SVD	IV- 71
IV-28. Tabel Rencana Pengujian <i>Use Case</i> Melakukan Clustering dengan Fuzzy C -Means	IV-72
IV-29. Tabel Rencana Pengujian <i>Use Case</i> Melakukan Clustering dengan Fuzzy C-Means dan SVD	IV-73
IV-30. Tabel Pengujian <i>Use Case</i> Memasukkan Dokumen.....	IV-75
IV-31. Tabel Pengujian <i>Use Case</i> Melakukan Clustering dengan K-Means	IV-77
IV-32. Tabel Pengujian <i>Use Case</i> Melakukan Clustering dengan K-Medoids	IV-79
IV-33. Tabel Pengujian <i>Use Case</i> Melakukan Clustering dengan Fuzzy C-Means	IV-81
IV-34. Tabel Pengujian <i>Use Case</i> Melakukan Clustering dengan K-Means dan SVD	IV-82
IV-35. Tabel Pengujian <i>Use Case</i> Melakukan Clustering dengan K-Medoids dan SVD	IV-85
IV-36. Tabel Pengujian <i>Use Case</i> Melakukan Clustering dengan Fuzzy C-Means dan SVD	IV-89
V-1. Tabel Hasil Clustering K-Means	V-4

V-2. Tabel Hasil Clustering K-Means dan Singular Value Decomposition	V-5
V-3. Tabel Hasil Clustering K-Medoids	V-7
V-4. Tabel Hasil Clustering KMedoids dan Singular Value Decomposition	V-8
V-5. Tabel Hasil Clustering Fuzzy C-Means	V-10
V-6. Tabel Hasil Clustering Fuzzy C-Means dan Singular Value Decomposition.....	
.....	V-12
V-7. Tabel Confussion Matrix Hasil Clustering K- Means	V-14
V-8. Tabel Confussion Matrix Hasil Clustering K-Means dan Singular Value Decomposition.....	V-14
V-9. Tabel Confussion Matrix Hasil Clustering K-Medoids	V-15
V-10. Tabel Confussion Matrix Hasil Clustering K-Medoids dan Singular Value Decomposition.....	V-16
V-11. Tabel Confussion Matrix Hasil Clustering Fuzzy C-Means.....	V-16
V-12. Tabel Confussion Matrix Hasil Clustering Fuzzy C-Means dan Singular Value Decomposition	V-17
V-13. Tabel Evaluasi Performa Clustering K-Means.....	V-19
V-14. Tabel Evaluasi Performa Clustering Kombinasi Singular Value Decomposition dan K-Means	V-20
V-15. Tabel Evaluasi Performa Clustering K-Medoids	V-21
V-16. Tabel Evaluasi Performa Clustering Kombinasi Singular Value Decomposition dan K-Medoids	V-22
V-17. Tabel Evaluasi Performa Clustering Fuzzy C-means	V-23
V-18. Tabel Evaluasi Performa Clustering Kombinasi Singular Value Decomposition dan Fuzzy C-means	V-24
V-19. Tabel Waktu Komputasi Clustering dengan dan tanpa SVD.....	V-25
V-20. Tabel Banyak Iterasi Clustering dengan dan tanpa SVD	V-26

DAFTAR GAMBAR

	Halaman
II-1. Gambar Dekomposisi Singular Value Decomposition	II-9
III-1. Gambar Diagram Tahapan Penelitian.....	III-8
III-2. Gambar Penjadwalan untuk Tahap Menentukan Ruang Lingkup dan Unit Penelitian	III-18
III-3. Gambar Penjadwalan untuk Tahap Menentukan Dasar Teori yang Berkaitan dengan Penelitian dan Menentukan Kriteria Pengujian.....	III-19
III-4. Gambar Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Insepsi	III-19
III-5. Gambar Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Elaborasi	III-20
III-6. Gambar Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Konstruksi.....	III-20
III-7. Gambar Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Transisi	III-21
III-8. Gambar Penjadwalan untuk Tahap Melakukan Pengujian Penelitian, Analisa Hasil Pengujian Penelitian dan Membuat Kesimpulan	III-21
IV-1. Gambar Diagram <i>Use Case Current Existing</i>	IV-2
IV-2. Gambar Diagram <i>Use Case</i>	IV-22
IV-3. Gambar Diagram Aktivitas Use Case Memasukkan Dokumen	IV-40
IV-4. Gambar Diagram Aktivitas Use Case Melakukan Clustering dengan K-Means	IV-41
IV-5. Gambar Diagram Aktivitas Use Case Melakukan Clustering dengan K-Means dan SVD	IV-42
IV-6. Gambar Diagram Aktivitas Use Case Melakukan Clustering dengan K-Medoids	IV-43
IV-7. Gambar Diagram Aktivitas Use Case Melakukan Clustering dengan	

K- Medoids dan SVD	IV-44
IV-8. Gambar Diagram Aktivitas Use Case Melakukan Clustering dengan Fuzzy C-Means	IV-45
IV-9. Gambar Diagram Aktivitas Use Case Melakukan Clustering dengan Fuzzy C-Means dan SVD	IV-46
IV-10. Gambar Rancangan Antar Muka Halaman Utama	IV-48
IV-11. Gambar Rancangan Antar Muka Pusat Cluster	IV-49
IV-12. Gambar Diagram Sequence Use Case Memasukkan Dokumen.....	IV-51
IV-13. Gambar Diagram Sequence Use Case Melakukan Clustering dengan K-Means	IV-52
IV-14. Gambar Diagram Sequence Use Case Melakukan Clustering dengan K-Means dan SVD	IV-53
IV-15. Gambar Diagram Sequence Use Case Melakukan Clustering dengan K-Medoids.....	IV-54
IV-16. Gambar Diagram Sequence Use Case Melakukan Clustering dengan K-Medoids dan SVD	IV-55
IV-17. Gambar Diagram Sequence Use Case Melakukan Clustering dengan Fuzzy C-Means	IV-56
IV-18. Gambar Diagram Sequence Use Case Melakukan Clustering dengan Fuzzy C-Means dan SVD	IV-57
IV-19. Gambar Diagram Kelas	IV-59
IV-20. Gambar Antar Muka Halaman Utama Perangkat Lunak.....	IV-65
IV-21. Gambar Antar Muka Pusat Cluster.....	IV-66
V-1. Gambar Perbandingan Performa Clustering k-means dengan dan tanpa SVD	V-28
V-2. Gambar Perbandingan Performa Clustering k-medoids dengan dan tanpa SVD	V-29
V-3. Gambar Perbandingan Performa Clustering fuzzy c-means dengan dan tanpa SVD	V-30
V-4. Gambar Perbandingan Waktu Komputasi Clustering dengan dan tanpa SVD	V-31

V-5. Gambar Perbandingan Banyak Iterasi Clustering dengan dan tanpa SVD	
.....	V-32

DAFTAR LAMPIRAN

1. Biodata penulis
2. Dokumentasi *source code*
3. Jurnal penelitian

BAB I

PENDAHULUAN

1.1 Pendahuluan

Bab 1 memberikan penjelasan umum mengenai keseluruhan penelitian. Antara lain membahas latar belakang masalah penelitian, perumusan masalah penelitian, tujuan penelitian dan manfaat penelitian. Pendahuluan dimulai dengan penjelasan mengenai proses clustering pada dokumen jurnal serta penelitian yang berkaitan dengan penggabungan teknik reduksi dimensi dan clustering pada data berdimensi tinggi yang menjadi latar belakang dari penelitian ini.

1.2 Latar Belakang Masalah

Dokumen jurnal yang disajikan dalam jumlah dan jenis yang banyak memerlukan proses *clustering* untuk mempermudah pencarian informasi tentang bidang tertentu (Indranandita, Susanto, & Rahmat, 2011). Sebuah jurnal dapat mengandung banyak dimensi atau atribut, dimana setiap kata kunci dipresentasikan sebagai satu dimensi. Kata kunci biasanya merupakan kata-kata yang unik, istilah yang penting dan bukan merupakan kata-kata umum yang digunakan seperti kata depan (di, ke, dari, daripada, kepada). Algoritma clustering konvensional hanya mampu mengatasi data berdimensi rendah (Han, Pei, & Kamber, 2011). Sehingga, untuk menemukan cluster dari obyek data berdimensi tinggi adalah sebuah tantangan dan menjadi salah satu fokus penelitian dalam clustering.

Untuk meningkatkan hasil clustering data berdimensi tinggi dapat dilakukan reduksi dimensi (Han et al., 2011). Reduksi dimensi merupakan teknik dalam *text mining* dengan mengurangi dimensi sehingga clustering memproses data dengan jumlah fitur yang telah berkurang. Oleh karena itu, metode yang menggabungkan reduksi dimensi dan algoritma clustering dianjurkan (Han et al., 2011). Penelitian mengenai penggabungan teknik reduksi dimensi dan analisis cluster telah dilakukan dengan berbagai obyek penelitian. Kadhim, Cheah, dan Ahamed (2014) menerapkan *TF-IDF* dan teknik reduksi dimensi *Singular Value Decomposition (SVD)* yang membantu proses clustering dokumen berdimensi tinggi dengan metode *k-means*. Penelitian ini menggunakan dua data uji yang masing-masing tingkat akurasi clusteringnya sebesar 95% dan 94,6667%. Hasil penelitian ini menunjukkan bahwa reduksi dimensi dapat meningkatkan derajat pengenalan antar dokumen dan kinerja sistem untuk clustering dokumen. Nur'aini, Najahaty, Hidayati, Murfi, dan Nurrohmah (2015) menggabungkan teknik reduksi dimensi *Singular Value Decomposition (SVD)* dan metode *k-means* untuk mendeteksi topik di *Twitter*. Hasil penelitian menunjukkan hasil *recall*, *precision*, dan akurasi yang komparatif. Di sisi lain, penggunaan *Singular Value Decomposition (SVD)* untuk mengurangi dimensi dari *tweet* meningkatkan waktu komputasi secara signifikan.

Selama ini, algoritma clustering yang sering digunakan pada penelitian penggabungan teknik reduksi dimensi dan analisis cluster adalah *k-means*. Sedangkan, terdapat algoritma clustering partisi lainnya yang merupakan peningkatan dari *k-means*, yaitu *k-medoids* dan *fuzzy c-means*. Menurut

Madhulatha (2011) yang melakukan perbandingan algoritma clustering *k-means* dengan *k-medoids* diketahui bahwa untuk data yang lebih besar, *k-medoids* bekerja lebih baik daripada *k-means*. *K-medoids* juga lebih tahan terhadap data yang memiliki noise dan penciran dibandingkan dengan *k-means* (Kaufman & Rousseeuw, 1987). Sibagariang (2015) pada penelitian tugas akhirnya melakukan analisis cluster pada 100 dokumen abstrak jurnal berbahasa Indonesia menggunakan metode *k-medoids*. Penelitian ini, menunjukkan bahwa clustering yang menggunakan *k-medoids* berkualitas baik dengan nilai rata-rata *purity* sebesar 0.827.

Astuti (2016) pada penelitian tugas akhirnya melakukan pengelompokkan terhadap 200 dataset yang diambil dari media online menggunakan metode *fuzzy c-means*. Hasilnya menunjukkan tingkat akurasi *f-measure* tertinggi yaitu 0,90. Berdasarkan hal tersebut, maka penelitian ini akan menguji pengaruh teknik reduksi dimensi *Singular Value Decomposition (SVD)* terhadap hasil pengelompokkan dokumen berdimensi tinggi menggunakan beberapa metode clustering.

1.3 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah apa pengaruh teknik reduksi dimensi *Singular Value Decomposition (SVD)* terhadap hasil pengelompokkan dokumen berdimensi tinggi menggunakan beberapa metode clustering. Untuk menjawab rumusan masalah tersebut, diuraikan beberapa *research question* sebagai berikut:

1. Bagaimana mekanisme metode *k-means*, *k-medoids*, dan *fuzzy c-means* untuk clustering dokumen?
2. Jika metode *k-means*, *k-medoids*, dan *fuzzy c-means* masing-masing akan dikombinasikan dengan *Singular Value Decomposition (SVD)*, maka apa peranan *Singular Value Decomposition (SVD)* pada metode-metode tersebut untuk mengelompokkan dokumen berdimensi tinggi?
3. Bagaimana *Singular Value Decomposition (SVD)* bekerja dengan metode *k-means*, *k-medoids*, dan *fuzzy c-means* untuk clustering dokumen berdimensi tinggi?

1.4 Tujuan Penelitian

Tujuan dilakukannya penelitian adalah sebagai berikut:

1. Mengetahui mekanisme metode *k-means*, *k-medoids*, dan *fuzzy c-means* untuk clustering dokumen;
2. Mengetahui peranan *Singular Value Decomposition (SVD)* pada metode *k-means*, *k-medoids*, dan *fuzzy c-means* untuk clustering dokumen berdimensi tinggi;
3. Mengetahui mekanisme *Singular Value Decomposition (SVD)* bekerja dengan metode *k-means*, *k-medoids*, dan *fuzzy c-means* untuk clustering dokumen berdimensi tinggi.

1.5 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian ini adalah sebagai berikut :

1. Memahami *k-means*, *k-medoids*, dan *fuzzy c-means* sebagai metode clustering dokumen;
2. Memahami peran *Singular Value Decomposition (SVD)* pada metode *k-means*, *k-medoids*, dan *fuzzy c-means* untuk clustering dokumen berdimensi tinggi;
3. Memahami mekanisme *Singular Value Decomposition (SVD)* bekerja dengan metode *k-means*, *k-medoids*, dan *fuzzy c-means* untuk clustering dokumen berdimensi tinggi;
4. Mampu menerapkan metode *k-means*, *k-medoids*, dan *fuzzy c-means* dengan teknik reduksi dimensi *Singular Value Decomposition (SVD)* pada sistem pengelompokan otomatis dokumen jurnal berbahasa Indonesia.

1.6 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut :

1. Karena kaidah pada praproses diatur dalam tata bahasa Indonesia, maka data yang digunakan adalah dokumen jurnal berbahasa Indonesia yang diperoleh dari situs *id.portalgaruda.org*;
2. Metode clustering yang digunakan adalah *k-means*, *k-medoids*, dan *fuzzy c-means*;
3. Evaluasi kualitas clustering dilakukan dengan mengukur *precision*, *recall*, dan *f-measure*. Kemudian hasil clustering *k-means*, *k-medoids*,

dan *fuzzy c-means* yang telah dilakukan reduksi dimensi *Singular Value Decomposition (SVD)* akan dibandingkan.

1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini adalah sebagai berikut :

BAB I. PENDAHULUAN

Pada bab ini diuraikan mengenai latar belakang, perumusan masalah, tujuan dan manfaat penelitian, batasan masalah/ruang lingkup, metodologi penelitian, dan sistematika penulisan.

BAB II. KAJIAN LITERATUR

Pada bab ini akan dibahas dasar-dasar teori yang digunakan dalam penelitian, seperti definisi-definisi *k-means*, *k-medoids*, *fuzzy c-means*, dan *Singular Value Decomposition (SVD)*.

BAB III. METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai tahapan yang akan dilaksanakan pada penelitian ini. Masing-masing rencana tahapan penelitian dideskripsikan dengan rinci dengan mengacu pada suatu kerangka kerja. Di akhir bab ini berisi perancangan manajemen proyek pada pelaksanaan penelitian.

BAB IV. PENGEMBANGAN PERANGKAT LUNAK

Pada bab ini akan dibahas mengenai perancangan dan lingkungan implementasi penggabungan teknik reduksi dimensi dan beberapa metode clustering untuk mengelompokkan dokumen berdimensi tinggi, hasil eksekusi, dan hasil pengujian.

BAB V. HASIL DAN ANALISIS PENELITIAN

Pada bab ini, hasil pengujian berdasarkan langkah-langkah yang telah direncanakan disajikan. Analisis diberikan sebagai basis dari kesimpulan yang diambil dalam penelitian ini.

BAB VI. KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dari semua uraian-uraian pada bab-bab sebelumnya dan juga berisi saran-saran yang diharapkan berguna dalam penerapan GIS demam berdarah ini.

BAB II

TINJAUAN PUSTAKA

1.1 Pendahuluan

Pada bab I dijelaskan bahwa rumusan masalah penelitian adalah apa pengaruh teknik reduksi dimensi *Singular Value Decomposition (SVD)* terhadap hasil pengelompokan dokumen berdimensi tinggi menggunakan beberapa metode clustering. Metode-metode clustering tersebut antara lain *k-means*, *k-medoids*, dan *fuzzy c-means*. Untuk memahami fundamental objek penelitian, penulis melakukan *literature review* terhadap jurnal, buku, dan artikel yang terkait dengan teknik reduksi *Singular Value Decompositions*, *k-means*, *k-medoids*, dan *fuzzy c-means*.

1.2 K-Means

K-Means adalah salah satu metode *clustering* partisi yang sering digunakan. Teknik ini mengelompokkan objek ke dalam K kelompok. Untuk melakukan pengelompokan ini, nilai K harus ditentukan terlebih dahulu. Proses yang dilakukan *K-Means* adalah memilih k *center* sebagai *centroid* yang akan digunakan dalam pengelompokan data. Langkah-langkah dalam algoritma *K-Means* menurut (Younus et al., 2014) yaitu :

1. Tentukan K sebagai jumlah pengelompokan yang diinginkan.
2. Bangkitkan k *centroid* (titik pusat) awal secara random. Pada pemilihan k *centroid* dapat dilakukan dengan memilih data awal secara *random* sebagai *centroid* atau dapat juga ditentukan dari awal data yang menjadi k *centroid*.

3. Hitung masing-masing jarak setiap data ke masing-masing *centroid* menggunakan rumus jarak *Euclidean*. Jarak *Euclidean* menghitung akar dari kuadrat perbedaan masing-masing dimensi dari dua titik, jika dituliskan sebagai rumus adalah sebagai berikut :

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (\text{II-1})$$

Keterangan :

d_{xy} = tingkat perbedaan (*dissimilarity degree*)

n = jumlah kata

x_i = bobot kata pada dokumen masukan

y_i = bobot kata dokumen pembanding

4. Tentukan posisi *centroid* baru dengan cara menghitung nilai rata-rata dari data yang berada pada *centroid* yang sama.
5. Hitung jarak setiap data dengan *centroid* yang baru menggunakan rumus jarak *Euclidean*.
6. Apabila terjadi perubahan pada data *centroid* awal dan *centroid* yang baru maka kembali ke langkah 4.
7. Apabila tidak terjadi perubahan pada data yang terletak dalam satu *centroid* maka algoritma berhenti.

Titik *centroid* akan menentukan keberhasilan dalam algoritma *K-Means*. Apabila titik *centroid* yang dipilih pada awal salah, maka hasil yang didapat tidak akan maksimal. Dalam *K-Means* titik *centroid* sering jadi permasalahan yang harus diatasi karena pemilihan nya secara acak.

1.3 K-Medoids

K-medoids adalah metode clustering partisi yang merupakan peningkatan dari metode *k-means*. Dibandingkan dengan *k-means*, *k-medoids* lebih tahan terhadap *noise* karena pengelompokan obyek-obyek ke dalam *k* cluster dengan cara meminimalisir *absolute error*. Dalam menemukan *k* cluster, pertama kali dilakukan pemilihan secara acak wakil dari data pada tiap-tiap cluster. Representasi cluster adalah salah satu titik yang dipilih menjadi perwakilan cluster yaitu *medoid*. Cluster dibangun dengan menghitung kedekatan yang dimiliki antara *medoid* dengan objek non-*medoid*.

Untuk menghitung jarak kedekatan menggunakan rumus *Cosine distance* (Huang, 2008).

$$d=1-Similarity \quad (II-2)$$

$$Similarity = \cos A, B = \frac{A \cdot B}{|A||B|} = \frac{\sum_{k=1}^n A_k B_k}{\sqrt{\sum_{k=1}^n A_k^2} \sqrt{\sum_{k=1}^n B_k^2}} \quad (II-3)$$

Menurut Han, Pei, dan Kamber (2011) algoritma *k-medoids* adalah,

1. Tentukan *k* sebagai jumlah pengelompokan yang diinginkan;
2. Bangkitkan *medoid* awal secara random;
3. Hitung kemiripan *medoid* dengan *nonmedoid* dengan rumus *Cosine distance*;
4. Tempatkan objek *nonmedoid* kedalam *cluster* yang paling dekat dengan *medoid*;
5. Secara acak pilih satu dari *nonmedoid* yang disebut dengan O_{random} ;

6. Hitung kemiripan O_{random} dengan *nonmedoid* yang lain dengan rumus *cosine distance*;
7. Hitung selisih antara total kemiripan *medoid* awal (O_j) dengan total kemiripan O_{random} . $S = \text{total kemiripan sebelumnya } (O_j) - \text{total kemiripan terbaru } (O_{random})$;
8. Jika $S < 0$ maka terjadi pertukaran *medoid*, tukar O_{random} menjadi O_j dengan kata lain O_{random} menjadi *medoid*;
9. Ulangi langkah 3-8 hingga tidak ada perubahan.

1.4 Fuzzy C-Means

Fuzzy C-Means adalah suatu teknik peng-*cluster*-an yang mana keberadaannya tiap-tiap titik data dalam suatu *cluster* ditentukan oleh derajat keanggotaan. Teknik ini pertama kali diperkenalkan oleh Jim Bezdek pada tahun 1981. Konsep dari *Fuzzy C-Means* pertama kali adalah menentukan pusat *cluster*, yang akan menandai lokasi rata-rata untuk tiap-tiap *cluster*. Pada kondisi awal, pusat *cluster* ini masih belum akurat. Tiap-tiap titik data memiliki derajat keanggotaan untuk tiap-tiap *cluster*. Dengan cara memperbaiki pusat *cluster* dan derajat keanggotaan tiap-tiap titik data secara berulang, maka akan dapat dilihat bahwa pusat *cluster* akan bergerak menuju lokasi yang tepat. Perulangan ini didasarkan pada minimasi fungsi obyektif yang menggambarkan jarak dari titik data yang diberikan kepusat cluster yang terbobot oleh derajat keanggotaan titik data tersebut. Fungsi obyektif yang digunakan pada FCM adalah (Ross, 2005). Algoritma Fuzzy C-Means adalah sebagai berikut:

1. Menentukan banyak kelompok (c), *fuzzifier* (m), maksimum iterasi ($MaxIter$), perubahan nilai fungsi objektif terkecil yang diharapkan (ϵ), fungsi objektif awal ($P_0 = 0$), dan iterasi awal ($t = 1$)
2. Membangkitkan bilangan random U_{ik} dengan i merupakan banyak data dan k merupakan banyak kelompok sebagai elemen-elemen awal matriks keanggotaan awal U .
3. Menghitung pusat kelompok ke- i dengan persamaan :

$$p_i = \frac{\sum_{k=1}^N (u_{ik})^m x_k}{\sum_{k=1}^N (u_{ik})^m} \quad (II-4)$$

dimana u_{ik} nilai keanggotaan objek ke- k dengan pusat kelompok ke- i , x_k adalah objek data ke- k , N adalah banyaknya objek penelitian, dan m adalah *fuzzifier*.

4. Menghitung fungsi objektif pada iterasi ke- t dengan persamaan :

$$J(P, U, X, c, m) = \sum_{i=1}^c \sum_{k=1}^N (u_{ik})^m d^2_{ik}(x_k, p_i) \quad (II-5)$$

dimana c adalah banyak kelompok yang diinginkan, N adalah banyak objek penelitian, u_{ik} adalah nilai keanggotaan objek ke- k pada kelompok ke- i yang merupakan bagian dari matriks U , m adalah *fuzzifier*, dan $d^2_{ik}(x_k, p_i)$ adalah jarak antara vektor pengamatan ke- k dengan pusat kelompok ke- i .

5. Menghitung perubahan matriks keanggotaan dengan persamaan :

$$u_{ik} = \frac{1}{\sum_{i=1}^c \left(\frac{d^2_{ik}}{d^2_{jk}} \right)^{\frac{1}{m-1}}} \quad (II-6)$$

dimana u_{ik} adalah nilai keanggotaan objek ke- k dengan pusat kelompok ke- i , d_{ik}^i adalah jarak antara objek ke- k dengan pusat kelompok ke- i , d_{jk}^2 adalah jarak antara objek ke- k dengan pusat kelompok ke- j , dan m adalah *fuzzifier*.

6. Cek kondisi berhenti

- Jika $|J_t - J_{t-1}| < \varepsilon$ atau $t > MaxIter$ maka berhenti;
- Jika tidak : $t=t+1$, ulangi langkah ke-3.

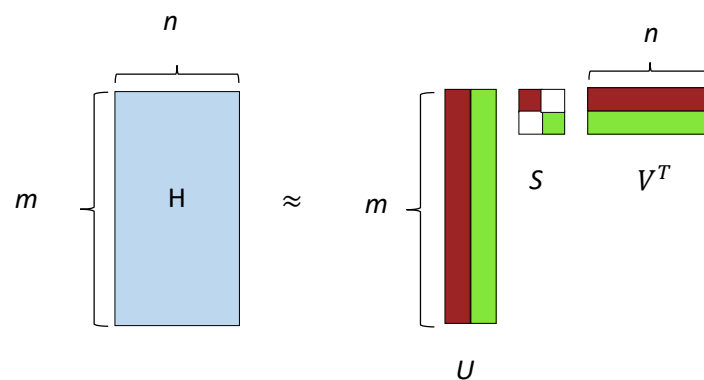
1.5 Singular Value Decomposition

Saat dimensi data tinggi, biasanya hanya sejumlah kecil dimensi yang relevan dengan cluster tertentu, namun data di dimensi yang tidak relevan dapat menghasilkan banyak *noise* dan menutupi cluster yang harusnya terbentuk. Selain itu saat dimensi meningkat, data biasanya menjadi semakin menjauh karena titik data umumnya berada diberbagai dimensi ruang bagian. Ketika data menjadi sangat jauh, titik data yang terletak diberbagai dimensi dapat dianggap memiliki jarak yang sama. Perhitungan jarak yang penting dalam *clustering* akan menjadi tidak berarti.

Untuk mengatasi masalah tersebut, reduksi dapat dilakukan dengan cara menangkap karakteristik data dengan memetakan set data dari dimensi semula ke dimensi lain yang relatif rendah. Pemetaan ini menghasilkan prinsipal komponen yang kemudian dapat diambil komponen atau fitur dari dimensi baru yang mempunyai pengaruh yang besar pada *dataset* dan membuang data yang tidak berpengaruh. Beberapa metode reduksi dimensi yang sudah dikenal secara luas

antara lain *wavelet*, analisis komponen utama, *independent component analysis*, *Information Gain*, *Singular Value Decomposition*, dan lain sebagainya (Han et al., 2011). Proses reduksi dimensi diterapkan sebelum proses clustering dan sesudah praproses dan pembobotan kata.

Singular Value Decomposition yaitu suatu bentuk analisa faktor pada matriks. Pada *SVD* matriks memuat frekuensi kemunculan kata kunci didekomposisi menjadi tiga komponen matriks (Leskovec, Rajaraman, & Ullman, 2014). Komponen matriks pertama (U) mendeskripsikan entitas baris sebagai vektor orthogonal matriks. Komponen matriks kedua (S) berupa matriks diagonal yang memuat nilai skalar matriks. Dan komponen yang ketiga (V) adalah matriks entitas kolom sebagai vektor orthogonal matriks. Dalam penelitian ini, *singular value decomposition* dari matriks TF-IDF akan digunakan dalam pendekatan transformasi linier. SVD pada dasarnya untuk melakukan estimasi *rank* dari matriks. Gambar II-1 adalah representasi dari *Singular Value Decomposition*.



Gambar II-1. Dekomposisi *Singular Value Decomposition*

Jika diketahui matriks H dengan dimensi $m \times n$, dimana nilai $m \geq n$ dan $\text{rank}(H) = r$ maka *singular value decomposition* dari H , didefinisikan melalui persamaan,

$$H = USV^T \quad (\text{II-7})$$

Keterangan:

H : matriks TF-IDF

U : vektor singular kiri

V : vektor singular kanan

T : transpose

S : nilai singular

m : dokumen

n : *term*

r : rank

dimana,

$$U^T U = V^T V = I_n \quad (\text{II-8})$$

dan memenuhi kondisi,

$$S = \text{diag}(\sigma_1, \dots, \sigma_n) \quad (\text{II-9})$$

dimana,

$$\sigma_i > 0 \text{ untuk } 1 \leq i \leq r \quad (\text{II-10})$$

$$\sigma_j = 0 \text{ untuk } j \geq r + 1 \quad (\text{II-11})$$

Kolom pertama dari matriks U dan V mendefinisikan vektor eigen orthonormal yang bersesuaian dengan r nilai vektor eigen tidak-nol dari matriks

HH^T dan H^TH berturut-turut. Kolom dari matriks U dan V berisi vektor, masing-masing disebut vektor singular kiri dan kanan. Nilai singular dari H merupakan elemen diagonal dari matriks S , dimana nilai singular didapat dari akar pangkat dua dari nilai atribut dari sejumlah n nilai eigen dari HH^T .

Setelah memperoleh tiga matriks dari proses SVD, proses berikutnya untuk mereduksi dimensi dari matriks adalah dengan mengurangi dimensi dari matriks S yang berupa matriks diagonal. Nilai skalar matriks terkecil milik matriks S akan dihilangkan sehingga satu kolom dari matriks U dan satu baris matriks V juga hilang mengikuti letak nilai singular matriks S . Selanjutnya, dilakukan perkalian matriks baru U dan S sehingga menghasilkan matriks H baru. Matriks H baru tersebut akan memuat *principal component* data.

1.6 Penelitian Lain yang Relevan

Penelitian mengenai penggabungan teknik reduksi dimensi dan analisis cluster dilakukan oleh Kadhim, Cheah, dan Ahamed (2014) yang menerapkan pembobotan kata *TF-IDF* dan teknik reduksi dimensi yaitu *singular value decomposition (SVD)* yang membantu proses clustering dokumen berdimensi tinggi dengan metode *k-means*. Penelitian dilakukan pada dua data uji, data uji pertama berjumlah 2.225 dokumen teks yang diambil dari situs berita BBC dengan 9.636 kata kunci. Data uji pertama memiliki lima kelas alami yaitu bisnis, hiburan, politik, olahraga, dan teknologi. Data uji kedua diambil dari situs olahraga BBC yang berjumlah 737 dokumen teks dan 4.613 kata kunci. Data uji kedua memiliki lima kelas alami yaitu atletis, kriket, sepakbola, ragbi, dan tenis. Dimensi yang semula berjumlah 9.636 dan 4.613 (dari situs berita dan olahraga

BBC) masing-masing berkurang menjadi 100. Sedangkan akurasi clustering untuk data uji pertama sebesar 95% dan data uji kedua sebesar 94,6667%. Hasil penelitian menunjukkan bahwa reduksi dimensi dapat meningkatkan derajat pengenalan antar dokumen dan kinerja sistem untuk clustering dokumen.

Nur'aini, Najahaty, Hidayati, Murfi, dan Nurrohmah (2015) menggabungkan teknik reduksi dimensi *Singular Value Decomposition (SVD)* dan metode *k-means* untuk mendeteksi topik di *Twitter* dengan tiga *dataset* twitter yaitu *The US Super Tuesday*, *The US Presedential elections*, dan *The English FA Cup*. Jumlah topik ditentukan sebanyak 100 untuk semua dataset. Untuk membuat matriks kata, *tweet* diuraikan, dan kamus kata dibentuk melalui proses tokenisasi. Karakter selain huruf alfabet dihilangkan, dan dilanjutkan proses *stop-words removal*. Setelah itu, matriks kata diberi bobot menggunakan metode *TF-IDF*. Proses selanjutnya adalah menerapkan *SVD* dan metode *k-means*. Biasanya sulit untuk menentukan jumlah optimal dimensi (p). Maka, penelitian ini menggunakan pendekatan heuristik dengan mengkombinasikan *truncated SVD* dan *k-means* selama beberapa kali dan kelipatan nilai p untuk menentukan jumlah dimensi. Nilai p yang menghasilkan kinerja terbaik akan dipilih menjadi optimal p . Nilai optimal berurutan untuk *The US Super Tuesday*, *The US Presedential elections*, dan *The English FA* adalah 200, 200, dan 2. Hasil penelitian menunjukkan hasil *recall*, *precision*, dan akurasi yang komparatif. Di sisi lain, penggunaan *Singular Value Decomposition (SVD)* untuk mengurangi dimensi dari *tweet* meningkatkan waktu komputasi secara signifikan.

Penelitian mengenai metode *k-medoids* pun telah banyak dilakukan. Madhulatha (2011) membandingkan metode *k-means* dan *k-medoids* berdasarkan pendekatan dasarnya. Dari penelitian ini diketahui bahwa metode *k-means* bekerja lebih baik untuk data yang lebih sedikit. Namun untuk data dokumen yang lebih besar, *k-medoids* bekerja lebih baik daripada *k-means*. Sibagariang (2015) pada penelitian tugas akhirnya melakukan penelitian clustering abstrak dokumen jurnal berbahasa Indonesia menggunakan metode *k-medoids* dan metode TF-IDF. Penelitian ini menggunakan 100 dokumen jurnal dengan 5 kelas alami yaitu ekonomi, pendidikan, peternakan, sistem informasi, dan politico. Masing-masing kelas mempunyai 20 dokumen jurnal. Percobaan dilakukan sebanyak 10 kali karena *k-medoids* menggunakan fungsi random saat inisialisasi cluster dan pemilihan medoid. Setelah dilakukan 10 kali percobaan maka nilai rata-rata purity yang dihasilkan adalah 0.827. Dengan kata lain clustering yang menggunakan *k-medoids* berkualitas baik.

Penelitian yang mengimplementasikan *fuzzy c-means* untuk mengelompokkan dokumen berita berbahasa Indonesia telah dilakukan oleh Astuti (2016). Data berita yang digunakan berasal dari berbagai website berita internet dengan 8 kategori yaitu berita ekonomi, berita olahraga, berita pendidikan, berita kesehatan, berita pariwisata, berita politik, dan berita teknologi. Hasil uji coba terhadap 200 dataset dengan masing-masing kategori terdapat 25 dokumen berita menunjukkan tingkat akurasi *f-measure* tertinggi yaitu 0,90. Nilai akurasi yang didapat pada saat ujicoba menunjukkan hasil yang didapat sangat tergantung pada jumlah variasi kata yang digunakan dalam pembobotan.

BAB III

METODOLOGI PENELITIAN

3.1 Pendahuluan

Bab ini menjelaskan unit penelitian, tahapan penelitian yang diimplementasikan, metodologi penelitian, serta penjadwalan penelitian. Tahapan penelitian dijadikan sebagai acuan pada setiap fase pengembangan dan memberikan sebuah solusi untuk rumusan masalah dan mencapai tujuan penelitian.

3.2 Unit Penelitian

Dalam penelitian ini yang digunakan sebagai unit penelitian adalah situs Indonesian Publication Index (*id.portalgaruda.org*) yang menyediakan layanan penelusuran, indeksasi, abstraksi, pemantauan, dan peningkatan standar kualitas berbagai publikasi ilmiah di Indonesia.

3.3 Metode Pengumpulan Data

Data yang digunakan sebagai obyek penelitian adalah jenis data sekunder berupa dokumen teks jurnal berbahasa Indonesia. Jumlah data yang digunakan direncanakan sebanyak 100 dokumen teks dengan lima kategori yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan.

Data didapatkan secara manual dengan mengunduh dokumen jurnal di situs Indonesian Publication Index (*id.portalgaruda.org*), kemudian konten

dokumen disalin ke dalam file berekstensi *.txt* dan disimpan dalam satu folder yang sama.

3.4 Tahapan Penelitian

Untuk mengetahui pengaruh *Singular Value Decomposition (SVD)* terhadap hasil pengelompokkan dokumen berdimensi tinggi menggunakan beberapa metode clustering, maka penelitian akan dilakukan dengan tahapan-tahapan yang akan dijelaskan pada subbab 3.3.1 sampai dengan 3.3.6.

3.4.1 Menentukan Ruang Lingkup dan Unit Penelitian

Tahapan ini menggunakan metode studi kepustakaan. Pembahasan mengenai batasan masalah telah dijelaskan pada bab I dan unit penelitian dibahas pada bab III subbab 3.2.

3.4.2 Menemukan Dasar Teori yang Berkaitan dengan Penelitian

Tahapan ini menggunakan metode studi kepustakaan. Pembahasan mengenai dasar teori yang berkaitan dengan penelitian telah dijelaskan pada bab II.

3.4.3 Menetapkan Kriteria Pengujian

Untuk melakukan pengujian penelitian sebelumnya data uji melalui tahapan praproses, pembobotan kata, reduksi dimensi, dan pengelompokkan menggunakan metode *k-means*, *k-medoids*, dan *fuzzy c-means*.

a. Praproses

Tahap *praproses* merupakan tahapan dalam mengelola data masukan. Tahap pertama dalam *praproses* pada perangkat lunak ini adalah *case folding*. Dalam penelitian ini, proses *Case folding* akan menyeragamkan semua huruf alfabet kapital menjadi huruf kecil. Selanjutnya, kalimat akan dipecah menjadi potongan kata atau yang dikenal dengan *tokenizing*. Tahap ketiga, potongan kata-kata dilakukan *stop words removal* yang berfungsi menghilangkan kata-kata yang sering muncul dalam kumpulan kata dan tidak memiliki makna. Tahap terakhir

adalah *stemming* menggunakan metode Nazief dan Adriani (1996) yang berfungsi mengubah kata yang memiliki imbuhan kembali ke bentuk dasarnya.

b. Pembobotan Kata

Setelah dilakukan praproses, kata-kata akan diberi nilai atau bobot. Pembobotan dilakukan dengan menggunakan metode gabungan *term frequency* dan *inverse document frequency* yang dikenal dengan istilah TF-IDF. TF-IDF adalah nilai bobot dari suatu kata yang diambil dari nilai *Term Frequency (TF)* dan inverse dari nilai *document frequency (DF)*.

c. Reduksi Dimensi

Matriks kata hasil pembobotan dilakukan proses clustering dokumen menggunakan metode *k-means*, *k-medoids*, dan *fuzzy c-means*. Dalam mereduksi dimensi matriks kata digunakan metode *singular value decomposition (SVD)*. Tahapan ini akan mengurangi ukuran data dengan menghasilkan hasil analisis yang hampir sama. Pada *SVD*, matriks yang memuat frekuensi kemunculan kata kunci (matriks H) didekomposisi menjadi tiga buah matriks yang jika tiga buah matriks tersebut dikalikan maka akan muncul kembali matriks asalnya. Matriks pertama atau matriks U dan matriks ketiga atau matriks V masing-masing mendeskripsikan entitas kolom dan baris sebagai nilai vektor orthogonal matriks. Matriks kedua atau matriks S berupa matriks diagonal yang memuat nilai skalar matriks. Kemudian, nilai skalar matriks terkecil milik matriks S akan dihilangkan sehingga satu kolom dari matriks U dan satu baris matriks V juga hilang mengikuti letak nilai singular matriks S . Selanjutnya, dilakukan perkalian matriks

baru U , dan S sehingga menghasilkan matriks H yang baru. Matriks H baru akan memuat *principal component* data.

d. Pengelompokkan menggunakan metode *k-means*

Jumlah cluster (k) yang akan dibentuk sebanyak lima sesuai dengan kategori awal yaitu pertanian, bahasa dan sastra, ilmu komputer, kesehatan masyarakat, dan hukum. Centroid awal atau pusat cluster ditentukan secara acak, setelah itu dilakukan pengukuran jarak setiap dokumen dan centroid menggunakan *euclidean*. Pengukuran ini bertujuan untuk mengetahui kemiripan atau kesamaan dari dokumen jurnal. Kemudian, cari centroid baru dengan menghitung nilai rata-rata data yang berada pada *centroid* yang sama, lakukan perhitungan jarak antara centroid baru dengan dokumen menggunakan cosine distance. Jika terjadi perubahan pada data centroid awal dan centroid yang baru maka kembali ulangi cari centroid. Jika tidak algoritma berhenti.

e. Pengelompokkan menggunakan metode *k-medoids*

Jumlah cluster (k) yang akan dibentuk sebanyak lima sesuai dengan kategori awal yaitu pertanian, bahasa dan sastra, ilmu komputer, kesehatan masyarakat, dan hukum. Medoid awal atau pusat *cluster* ditentukan secara acak, setelah itu dilakukan pengukuran jarak setiap dokumen dan medoid menggunakan *cosine distance*. Pengukuran ini bertujuan untuk mengetahui kemiripan atau kesamaan dari dokumen jurnal. Kemudian, selisih antara total jarak medoid dengan total jarak medoid acak dihitung untuk ditempatkan pada cluster dengan medoid yang sama. Setelah melalui proses pengelompokkan dokumen maka akan dihasilkan clustering dokumen jurnal.

f. Pengelompokkan menggunakan metode *fuzzy c-means*

Jumlah cluster (k) yang akan dibentuk sebanyak lima. Kemudian pangkat, maksimum iterasi, error terkecil yang diharapkan, fungsi objektif awal, dan iterasi awal ditentukan. Bangkitkan bilangan random untuk matriks partisi awal U Hitung pusat cluster V . Hitung fungsi objektif pada iterasi pertama, dan hitung perubahan matriks partisi U . Cek kondisi berhenti sampai dengan $|J_t - J_{t-1}| < \varepsilon$ atau $t > MaxIter$ Setelah melalui proses pengelompokkan dokumen maka akan dihasilkan clustering dokumen jurnal.

3.4.4 Menentukan Alat yang Digunakan dalam Pelaksanaan Penelitian

Untuk melaksanakan penelitian mengenai pengaruh *Singular Value Decomposition* terhadap hasil clustering dokumen berdimensi tinggi menggunakan beberapa metode clustering dibutuhkan alat penelitian. Oleh karena itu, penulis akan mengembangkan sebuah perangkat lunak yang dapat melakukan proses clustering dokumen baik menggunakan kombinasi *Singular Value Decomposition* dengan *k-means*, *k-medoids*, dan *fuzzy c-means*.

3.4.5 Melakukan Pengujian Penelitian

Pembahasan mengenai tahapan ini akan dijelaskan pada bab V. Pada tahapan pengujian penelitian, data hasil pengujian clustering dokumen akan digambarkan dalam tabel III-1.

Tabel III-1. Rancangan Tabel Hasil Clustering

Percobaan ke - n	Cluster i					Cluster B				
	E	H	K	PN	PT	E	H	K	PN	PT
n1										
n2										

Hasil pengujian secara keseluruhan menggunakan *recall*, *precision*, *F-*

Measure akan digambarkan dalam tabel III-2.

Tabel III-2. Rancangan Tabel Confussion Matrix Hasil Clustering

Percobaan ke - n	Nama Metode			
	TP	FP	TN	FN
.				
.				
n				

Keterangan :

TP : Jumlah dokumen yang secara benar dimasukkan ke dalam X

FP : Jumlah dokumen yang secara salah dimasukkan ke dalam X

FN: Jumlah teks yang secara salah tidak dimasukkan ke dalam X

Tabel III-3. Rancangan Tabel Evaluasi Performa Clustering

Percobaan ke - n	Nama Metode			
	Precision	Recall	F-Measure	Rand Index
.				
.				
n				

Sedangkan, waktu komputasi dan jumlah iterasi digambarkan dalam tabel III-4 dan III-5.

Tabel III-5. Rancangan Tabel waktu komputasi

Pengujian ke- n	Waktu Komputasi (s)		
	<i>K-means</i>	<i>K-medoids</i>	<i>Fuzzy C-means</i>

Tabel III-6. Rancangan Tabel waktu komputasi

Pengujian ke- n	Banyak Iterasi		
	<i>K-means</i>	<i>K-medoids</i>	<i>Fuzzy C-means</i>

3.4.6 Melakukan Analisa Hasil Pengujian dan Membuat Kesimpulan

Penelitian

Untuk mengetahui pengaruh *Singular Value Decomposition* terhadap hasil clustering dokumen berdimensi tinggi menggunakan beberapa metode clustering, maka efektivitas dan efisiensi dilakukan dengan melihat kedekatan setiap *cluster* yang dihasilkan perangkat lunak dengan kategori yang sebelumnya telah diidentifikasi secara manual, dengan menggunakan metode evaluasi standar dalam *clustering* yaitu *precision*, *recall* dan *F-measure* yang akan dibahas pada bab V.

Precision merupakan cara mengukur ketepatan cluster yang dibentuk oleh perangkat lunak. **Recall** adalah tingkat keberhasilan perangkat lunak dalam memisahkan dokumen yang mirip ke dalam cluster yang sama. **F-measure** adalah fungsi *harmonic mean* dari *precision* dan *recall*. Semakin nilai f-measure mendekati 1 maka menunjukkan hasil clustering semakin baik. Sedangkan **rand index** mempresentasikan akurasi hasil clustering aktual dengan nilai prediksi. Sehingga, dapat didefinisikan dengan persamaan sebagai berikut:

$$Precision = \frac{TP}{TP+FP}, \quad (III-1)$$

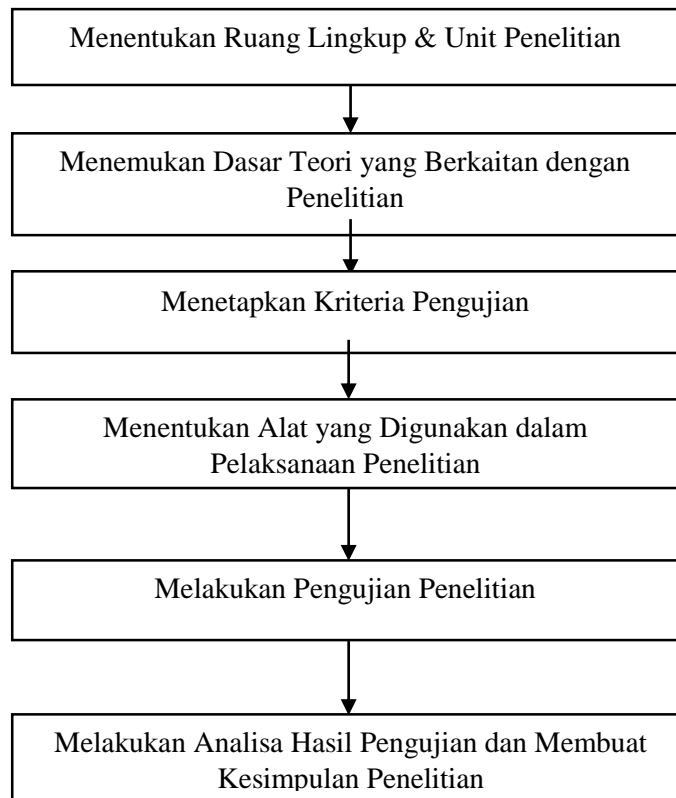
$$Recall = \frac{TP}{TP+FN}, \quad (III-2)$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (III-3)$$

$$Rand Index = \frac{TP+TN}{TP+TN+FP+FN} \quad (III-4)$$

Dimana TP menunjukkan jumlah dokumen yang secara benar dimasukkan ke dalam cluster ke- i , FP adalah jumlah dokumen yang secara salah dimasukkan ke dalam cluster ke- i , FN adalah jumlah dokumen yang secara salah tidak dimasukkan ke dalam cluster ke- i , dan TN adalah jumlah dokumen yang secara benar tidak dimasukkan ke dalam cluster ke- i .

Setelah mendapatkan hasil pengujian penelitian, maka langkah selanjutnya adalah membuat kesimpulan penelitian yang akan dijelaskan pada bab VI. Berikut adalah tahapan penelitian yang ditunjukkan pada gambar III-1.



Gambar III-1. Diagram Tahapan Penelitian

3.5 Metode Pengembangan Perangkat Lunak

Metodologi yang diterapkan dalam pengembangan perangkat lunak sebagai alat penelitian tugas akhir ini berorientasi pada objek menggunakan metode *Rational Unified Process* (RUP). Secara umum, langkah-langkah yang akan dilakukan pada pengembangan perangkat lunak adalah fase inepsi, elaborasi, konstruksi, dan transisi (Pressman, 2005).

3.5.1 Fase Inepsi

Pada tahapan pemodelan bisnis, penulis menentukan *user requirements* dan fungsionalitas atau fitur-fitur yang dibutuhkan pada perangkat lunak. Pada tahapan pengumpulan kebutuhan, penulis mengumpulkan data penelitian berupa

jurnal berbahasa Indonesia di situs Indonesian Publication Index (*id.portalgaruda.org*). Pada tahap analisis dan desain, penulis membuat diagram *use case*. Pada tahap implementasi, penulis mendokumentasikan *user requirements*, fungsionalitas perangkat lunak dan diagram *use case*. Pada tahap pengujian, penulis memastikan apakah *user requirements* dan fungsionalitas perangkat lunak sudah valid.

3.5.2 Fase Elaborasi

Pada tahapan pemodelan bisnis, penulis menentukan arsitektur perangkat lunak, desain basis data, dan desain antar muka sesuai dengan *user requirements* dan fungsionalitas perangkat lunak yang telah didapatkan. Penulis dapat melengkapi *user requirement*, apabila dirasa belum lengkap, pada tahap pengumpulan kebutuhan. *Activity diagram* dan *sequence diagram* dibuat pada tahap analisis dan desain. Penulis menyusun dokumentasi yang memuat arsitektur perangkat lunak, desain basis data, desain antar muka, *activity diagram*, dan *sequence diagram* pada tahap implementasi lalu memastikan seluruhnya sudah valid pada tahap pengujian.

3.5.3 Fase Konstruksi

Pada tahapan pemodelan bisnis, penulis menentukan kelas-kelas yang dibutuhkan pada perangkat lunak. Pada tahap pengumpulan kebutuhan, ditentukan bahasa pemrograman yang digunakan untuk mengembangkan perangkat lunak, yaitu Java. Kebutuhan lain dalam proses pengembangan perangkat lunak juga

diidentifikasi, seperti perangkat keras dengan Processor Intel(R) Core(TM) i3-5005U CPU @ 2.0 GHz, RAM 4 GB, dan Harddisk 500 GB, *staruml*, dan Netbeans IDE 8.0.2. *Class diagram* dibuat pada tahap analisis dan desain. Pada tahapan implementasi, penulis mengembangkan perangkat lunak dengan mengimplementasi kelas-kelas yang telah ditentukan ke kode program dalam bahasa Java. Selanjutnya, penulis melakukan *unit testing* terhadap perangkat lunak yang telah dikembangkan.

3.5.4 Fase Transisi

Pada tahapan pemodelan bisnis, penulis membuat rencana atau skenario pengujian terhadap perangkat lunak. Penulis menentukan *tools* pengujian yang diperlukan di tahap pengumpulan kebutuhan. *Tools* pengujian merupakan perangkat keras yang sama saat digunakan untuk pengembangan perangkat lunak yaitu laptop dengan Processor Intel(R) Core(TM) i3-5005U CPU @ 2.0 GHz, RAM 4 GB, dan Harddisk 500 GB. Penulis lalu mendesain tabel skenario pada tahap analisis dan desain. Pada tahapan implementasi, penulis melakukan pengujian terhadap perangkat lunak berdasarkan skenario atau rencana pengujian. Skenario pengujian ditinjau ulang pada tahap pengujian.

3.6 Penjadwalan Penelitian

Penjadwalan merupakan perencanaan aktivitas penelitian dari tahap inisialisasi masalah sampai dengan pada tahap kesimpulan dari penelitian. Adapun kegiatan-kegiatan yang berlangsung selama penelitian dapat dilihat dalam *Work Breakdown Structure* (WBS) yang tertera pada Tabel III-4, dan *Gantt Chart* pada

Gambar III-2, Gambar III-3, Gambar III-4, Gambar III-5, Gambar III-6, Gambar
III-7, dan Gambar III-8.

Tabel III-3. Penjadwalan Penelitian dalam Bentuk *Work Breakdown Structure* (WBS)

ID	Task Name	Duration	Start	Finish	Predecessors
	Pengaruh Singular Value Decomposition Terhadap Metode K-Medoids dalam Pengelompokan Dokumen Bahasa Indonesia Berdimensi Tinggi	157 days	Sun 10/23/16	Thu 5/25/17	
	Menentukan Ruang Lingkup dan Unit Penelitian	25 days	Sun 10/23/16	Sat 11/26/16	
T1	Menentukan masalah penelitian	7 days	Sun 10/23/16	Sat 10/29/16	
T2	Membuat latar belakang dan rumusan masalah	8 days	Sun 10/30/16	Tue 11/8/16	T1
T3	Menentukan tujuan dan manfaat penelitian	3 days	Wed 11/9/16	Fri 11/11/16	T2
T4	Menentukan batasan masalah	5 days	Tue 11/15/16	Sat 11/19/16	T3
T5	Menentukan unit penelitian	6 days	Sun 11/20/16	Fri 11/25/16	T4
M1	Tersedia dokumen hasil tahapan penelitian	0 days	Sat 11/26/16	Sat 11/26/16	T5
	Menentukan Dasar Teori yang Berkaitan dengan Penelitian	25 days	Sun 11/27/16	Mon 1/2/17	
T6	Mengumpulkan jurnal, paper, dan literatur ilmiah yang berkaitan dengan penelitian	12 days	Sun 11/27/16	Mon 12/12/16	T1
T7	Mempelajari metode K-Medoids, K-Means dan <i>Singular Value Decomposition</i>	15 days	Tue 12/13/16	Sun 1/1/17	T6
M2	Tersedia dokumen hasil tahapan penelitian	0 days	Mon 1/2/17	Mon 1/2/17	T7
	Menentukan Kriteria Pengujian	21 days	Tue 1/3/17	Wed 2/1/17	

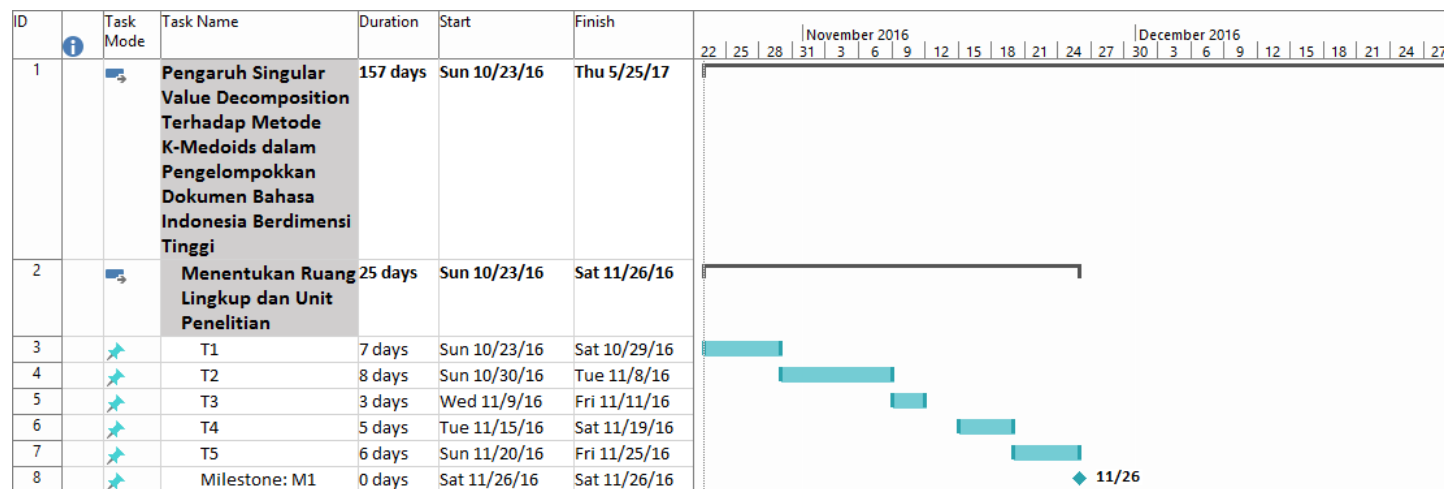
T8	Menentukan teknik yang digunakan untuk praproses data	13 days	Tue 1/3/17	Thu 1/19/17	T6
T9	Menentukan metode yang digunakan untuk pembobotan	8 days	Fri 1/20/17	Tue 1/31/17	T6
M3	Tersedia dokumen hasil tahapan penelitian	0 days	Wed 2/1/17	Wed 2/1/17	T8, T9
	Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian	60 days	Mon 2/27/17	Tue 5/16/17	
	<i>Inception</i>	8 days	Mon 2/27/17	Wed 3/8/17	
	<i>Business Modelling</i>	3 days	Mon 2/27/17	Wed 3/1/17	
T10	Menentukan <i>user requirements</i> dan fungsionalitas perangkat lunak	3 days	Mon 2/27/17	Wed 3/1/17	T1, T4, T5
	<i>Requirements</i>	4 days	Wed 3/1/17	Mon 3/6/17	
T11	Mengumpulkan dataset penelitian	4 days	Wed 3/1/17	Mon 3/6/17	T4, T5
	<i>Construction</i>	2 days	Tue 3/7/17	Wed 3/8/17	
T12	Membuat <i>use case diagram</i>	2 days	Tue 3/7/17	Wed 3/8/17	T10
	<i>Implementation</i>	2 days	Tue 3/7/17	Wed 3/8/17	
T13	Membuat dokumentasi	2 days	Tue 3/7/17	Wed 3/8/17	T10
	<i>Testing</i>	2 days	Tue 3/7/17	Wed 3/8/17	
T14	Memastikan <i>user requierements</i> dan fungsionalitas sudah valid	2 days	Tue 3/7/17	Wed 3/8/17	T13
	<i>Elaboration</i>	8 days	Thu 3/9/17	Sun 3/19/17	
	<i>Business Modelling</i>	4 days	Thu 3/9/17	Tue 3/14/17	
T15	Menentukan arsitektur perangkat lunak, desain basis data, dan desain antar muka	4 days	Thu 3/9/17	Tue 3/14/17	T14

	Requirements	3 days	Wed 3/15/17	Fri 3/17/17	
T16	Melengkapi <i>user requirements</i> yang telah didefinisikan di fase <i>inception</i>	3 days	Wed 3/15/17	Fri 3/17/17	T10
	Analysis & Design	3 days	Wed 3/15/17	Fri 3/17/17	
T17	Membuat <i>activity</i> dan <i>sequence diagram</i>	3 days	Wed 3/15/17	Fri 3/17/17	T12, T15
	Implementation	3 days	Wed 3/15/17	Fri 3/17/17	
T18	Membuat dokumentasi	3 days	Wed 3/15/17	Fri 3/17/17	T17
	Testing	1 day	Sat 3/18/17	Sun 3/19/17	
T19	Memastikan arsitektur perangkat lunak, desain basis data, dan desain antar muka sudah valid	1 day	Sat 3/18/17	Sun 3/19/17	T18
	Construction	38 days	Mon 3/20/17	Mon 5/8/17	
	Business Modelling	5 days	Mon 3/20/17	Fri 3/24/17	
T20	Menentukan kelas-kelas pada perangkat lunak	5 days	Mon 3/20/17	Fri 3/24/17	T19
	Requirements	2 days	Fri 3/24/17	Sat 3/25/17	
T21	Menentukan bahasa pemrograman yang digunakan untuk mengembangkan perangkat lunak	2 days	Fri 3/24/17	Sat 3/25/17	T7, T8, T9
T22	Menentukan kebutuhan perangkat keras yang digunakan	2 days	Fri 3/24/17	Sat 3/25/17	T21
	Analysis & Design	2 days	Sun 3/26/17	Tue 3/28/17	
T23	Membuat <i>class diagram</i>	3 days	Sun 3/26/17	Tue 3/28/17	T20, T21
	Implementation	30 days	Wed 3/29/17	Mon 5/8/17	
T24	Mengimplementasi kelas-kelas ke dalam kode program	30 days	Wed 3/29/17	Mon 5/8/17	T23

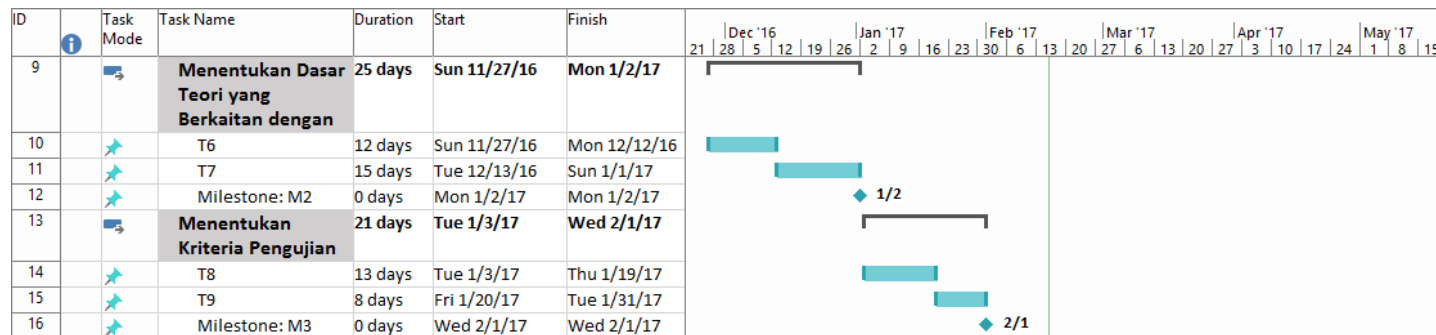
	Testing	30 days	Wed 3/29/17	Mon 5/8/17	
T25	Melakukan <i>unit testing</i>	30 days	Wed 3/29/17	Mon 5/8/17	T23
	Transition	6 days	Tue 5/9/17	Mon 5/15/17	
	Business Modelling	2 days	Tue 5/9/17	Wed 5/10/17	
T26	Membuat rencana atau skenario pengujian	2 days	Tue 5/9/17	Wed 5/10/17	T25
	Requirements	2 days	Thu 5/11/17	Fri 5/12/17	
T27	Menentukan <i>tools</i> pengujian yang diperlukan dan sampel pertanyaan	2 days	Thu 5/11/17	Fri 5/12/17	T25
	Analysis & Design	1 day	Fri 5/12/17	Sat 5/13/17	
T28	Membuat tabel skenario pengujian	2 days	Fri 5/12/17	Sat 5/13/17	T26, T27
	Implementation	2 days	Sun 5/14/17	Mon 5/15/17	
T29	Melakukan pengujian terhadap perangkat lunak berdasarkan skenario atau rencana pengujian	2 days	Sun 5/14/17	Mon 5/15/17	T26, T27
	Testing	2 days	Sun 5/14/17	Mon 5/15/17	
T30	Meninjau atau menguji skenario pengujian	2 days	Sun 5/14/17	Mon 5/15/17	T26, T27, T28
M4	Tersedia dokumen hasil tahapan penelitian	0 days	Tue 5/16/17	Tue 5/16/17	T30
	Melakukan Pengujian Penelitian	4 days	Tue 5/16/17	Sat 5/20/17	
T31	Menentukan rancangan hasil penelitian	2 days	Tue 5/16/17	Wed 5/17/17	T2, T4
T32	Melakukan pengujian penelitian berdasarkan hasil pengujian perangkat lunak	2 days	Thu 5/18/17	Fri 5/19/17	T29
M5	Tersedia dokumen hasil tahapan penelitian	0 days	Sat 5/20/17	Sat 5/20/17	T31, T32
	Melakukan Analisa Hasil Pengujian dan Membuat Kesimpulan	3 days	Sun 5/21/17	Thu 5/25/17	

T33	Melakukan analisa terhadap hasil pengujian penelitian dengan menghitung rata-rata, simpangan baku, dan varian	2 days	Sun 5/21/17	Mon 5/22/17	T32
T34	Membuat kesimpulan dan saran berdasarkan analisa terhadap hasil pengujian	2 days	Tue 5/23/17	Wed 5/24/17	T33
M6	Tersedia dokumen hasil tahapan penelitian	0 days	Thu 5/25/17	Thu 5/25/17	T34

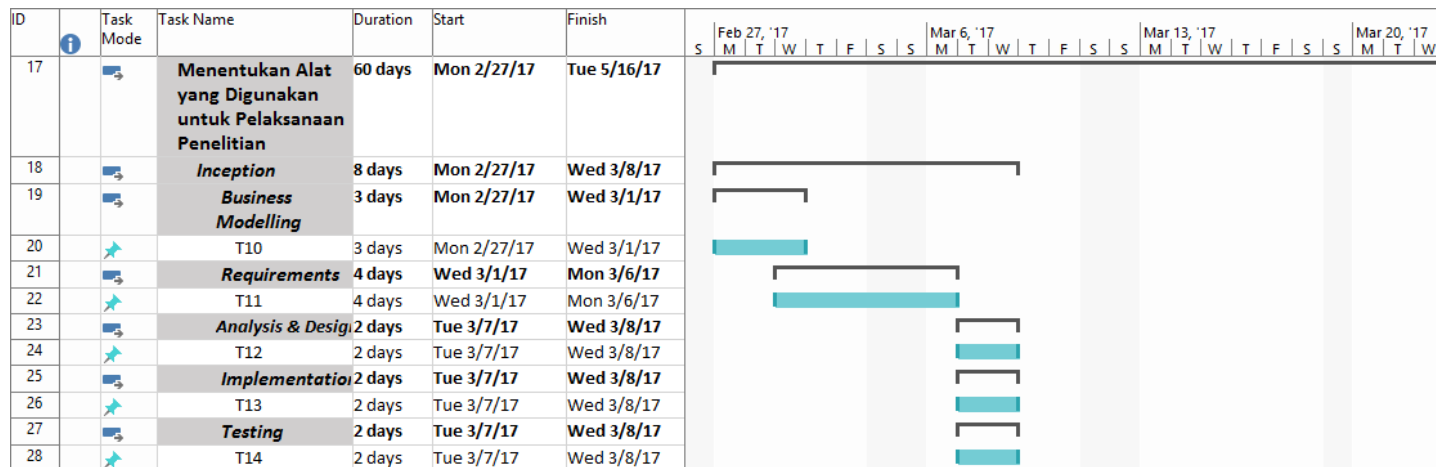
Penjadwalan penelitian dalam bentuk *Gantt Chart* dibuat dengan *tool* Microsoft Project 2013. Gambar III-2, Gambar III-3, Gambar III-4, Gambar III-5, Gambar III-6, Gambar III-7, dan Gambar III-8 menampilkan *Gantt Chart* untuk penjadwalan penelitian ini.



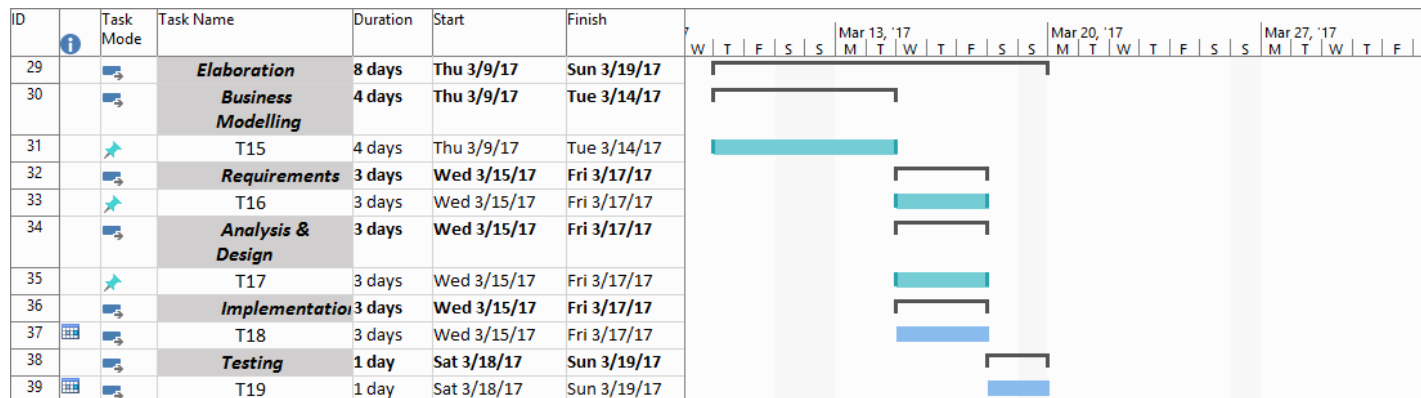
Gambar III-2. Penjadwalan untuk Tahap Menentukan Ruang Lingkup dan Unit Penelitian



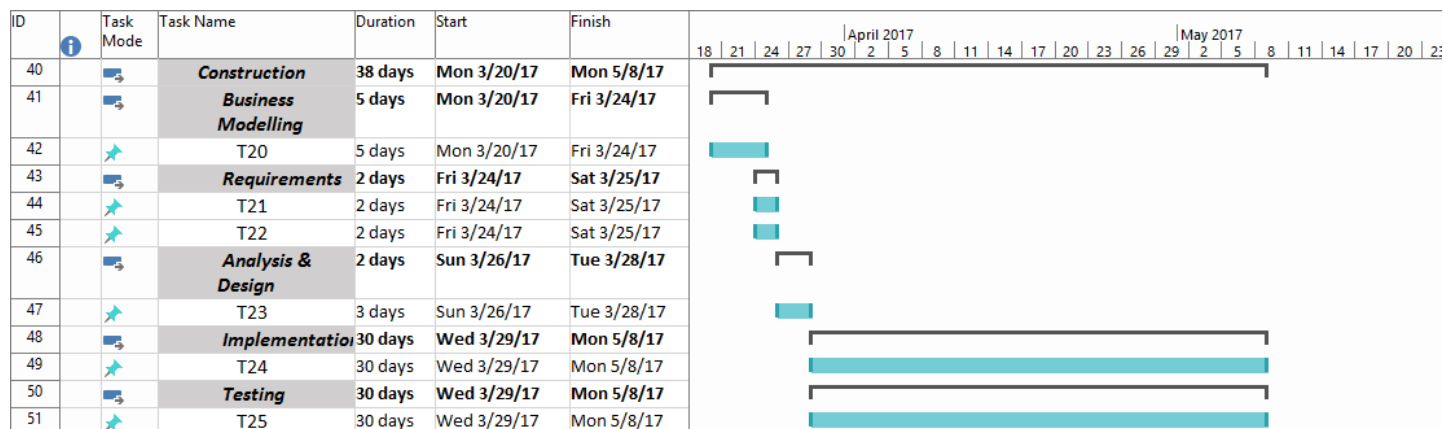
Gambar III-3. Penjadwalan untuk Tahap Menentukan Dasar Teori yang Berkaitan dengan Penelitian dan Menentukan Kriteria Pengujian



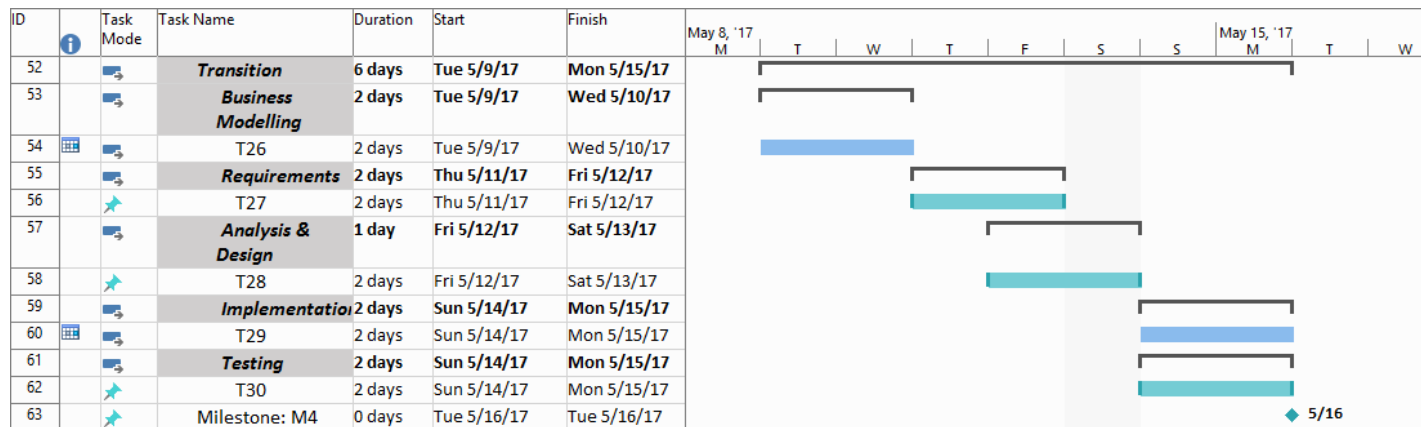
Gambar III-4. Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Insepsi



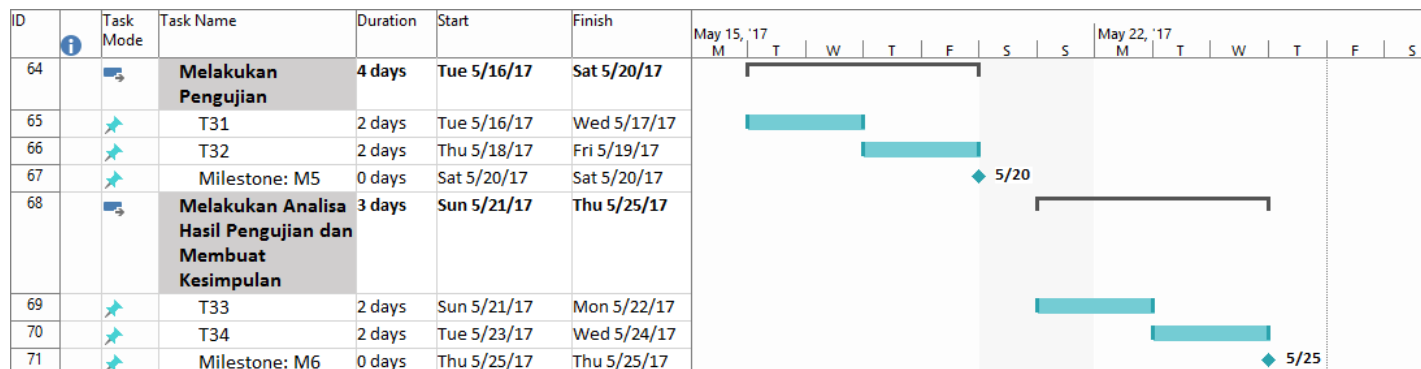
Gambar III-5. Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Elaborasi



Gambar III-6. Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Konstruksi



Gambar III-7. Penjadwalan untuk Tahap Menentukan Alat yang Digunakan untuk Pelaksanaan Penelitian Fase Transisi



Gambar III-8. Penjadwalan untuk Tahap Melakukan Pengujian Penelitian, Analisa Hasil Pengujian Penelitian dan Membuat Kesimpulan

BAB IV

PENGEMBANGAN PERANGKAT LUNAK

4.1 Pendahuluan

Pada bab 3 disebutkan bahwa diperlukan sebuah alat berupa perangkat lunak yang digunakan dalam pelaksanaan penelitian ini, maka penulis mengembangkan perangkat lunak dengan metode pemrograman berorientasi obyek berdasarkan panduan *Rational Unified Process*. Di dalam Rational Unified Process terdapat empat fase pengembangan perangkat lunak yaitu fase insepisi, elaborasi, konstruksi, dan transisi. Setiap fase terdiri dari pemodelan bisnis, kebutuhan, analisis dan desain, implementasi, dan pengujian. Pada bab ini dibahas proses pengembangan perangkat lunak yang digunakan sebagai alat penelitian.

4.2 Fase Insepisi

Tahapan pertama dalam pengembangan perangkat lunak adalah melakukan identifikasi terhadap sistem yang dikembangkan. Aktivitas yang dilakukan mencakup analisis sistem, identifikasi kebutuhan, perumusan kebutuhan pengujian, pemodelan diagram *use case* dan diagram aktivitas.

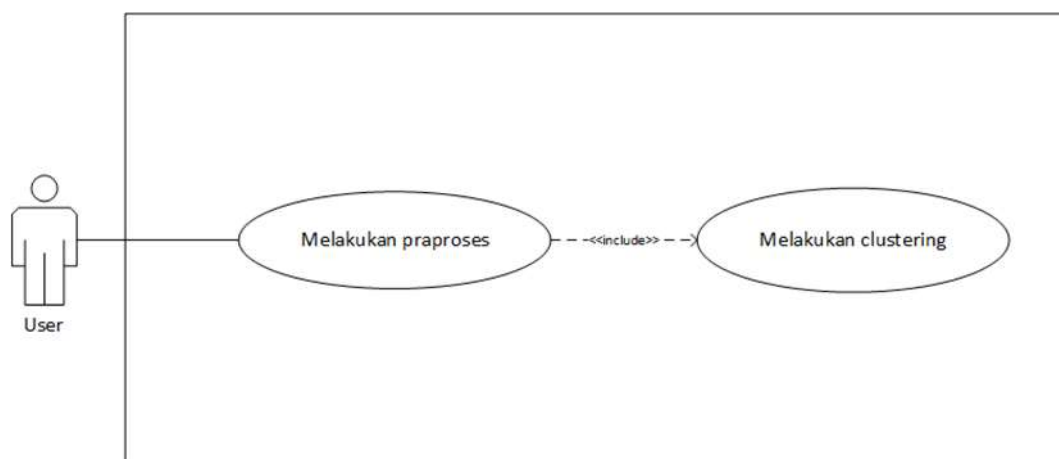
4.2.1 Pemodelan Bisnis

Jurnal merupakan kebutuhan untuk mendapatkan informasi tentang bidang yang akan diteliti. Seiring kemajuan zaman, semakin banyak orang yang mempublikasikan jurnal penelitiannya di internet. Sebuah jurnal dapat terdiri dari ribuan kosakata dengan minimal jumlah halaman sebanyak 5. Jika jurnal disajikan dalam jumlah yang banyak maka akan menyulitkan pembaca dalam melakukan

pencarian informasi. Untuk mempermudah pencarian, jurnal dapat dikelompokkan berdasarkan topik penelitiannya. Namun pengelompokan secara manual menimbulkan permasalahan sebagai berikut:

1. dibutuhkan ketelitian yang tinggi untuk membaca jurnal satu persatu;
2. membutuhkan waktu yang lama untuk mengelompokkan;
3. dibutuhkan banyak sumber daya manusia.

Dari permasalahan tersebut, dibutuhkan pengelompokan otomatis



terhadap jurnal dengan melakukan clustering. Pada gambar IV-1 menunjukkan proses pengelompokkan dokumen menggunakan clustering.

Gambar IV-1. Diagram *Use Case Current Existing*

Namun, untuk jumlah data yang banyak maka algoritma clustering konvensional tidak mampu menghasilkan kualitas cluster yang baik. Dibutuhkan teknik reduksi dimensi untuk memaksimalkan kinerja algoritma clustering konvensional. Perangkat lunak yang dibangun merupakan perangkat lunak berbasis

desktop yang digunakan untuk clustering jurnal berbahasa Indonesia dengan menggunakan metode *k-means*, *k-medoids*, dan *fuzzy c-means*. Perangkat lunak mampu melakukan clustering jurnal secara otomatis tanpa teknik reduksi dimensi ataupun dengan terlebih dahulu dilakukan proses reduksi dimensi oleh metode *singular value decomposition*. Masukan untuk perangkat lunak yang dikembangkan berupa berkas bertipe *.txt* yang berisikan konten jurnal berbahasa Indonesia. Keluaran yang dihasilkan berupa tabel berisi data dokumen yang telah terkelompok berdasarkan *cluster* nya.

4.2.2 Kebutuhan Sistem

Spesifikasi kebutuhan perangkat lunak *clustering* dokumen berbahasa Indonesia dibangun berdasarkan pemodelan bisnis. Fitur-fitur utama yang disediakan perangkat lunak antara lain fitur prapengolahan, reduksi dimensi, dan clustering.

a. Fitur Prapengolahan

Perangkat lunak dilengkapi fitur prapengolahan yang digunakan oleh pengguna untuk mengkonversi dokumen teks ke dalam bentuk numerik. Prapengolahan yang dilakukan yaitu *casefolding*, *tokenizing*, *stop words removal*, *stemming*, dan pembobotan kata *tf-idf*.

b. Fitur Reduksi Dimensi

Fitur selanjutnya yang dimiliki perangkat lunak, pengguna dapat melakukan proses reduksi dimensi menggunakan *singular value decomposition*. Fitur ini hanya dapat digunakan jika pengguna memilih melakukan clustering dengan teknik reduksi dimensi.

c. Fitur *Clustering*

Fitur utama yang merupakan fokus penelitian yaitu clustering dokumen. Pengguna dapat memilih metode clustering yang digunakan untuk mengelompokkan dokumen yaitu metode *k-means*, *k-medoids*, dan *fuzzy c-means*.

Untuk dapat merealisasikan fitur-fitur tersebut, perangkat lunak harus memenuhi kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional menjelaskan kebutuhan atau fasilitas utama perangkat lunak yang dibangun pada tabel IV-1. Sedangkan, kebutuhan non fungsional menjelaskan kebutuhan atau fasilitas yang tidak wajib dimiliki oleh perangkat lunak, dalam artian hanya merupakan pelengkap agar perangkat lunak lebih baik kinerjanya yang dapat dilihat pada tabel IV-2.

Tabel IV-1 Kebutuhan Fungsional

No.	Kebutuhan
1	Perangkat lunak dapat melakukan prapengolahan dokumen.
2	Perangkat lunak dapat melakukan reduksi dimensi dokumen.
3	Perangkat lunak dapat melakukan proses clustering dokumen.

Tabel IV-2 Kebutuhan Non Fungsional

No.	Kebutuhan
1	Perangkat lunak dapat menampilkan pesan kesalahan jika terdapat aksi pengguna yang salah dan konfirmasi tindakan pengguna.
2	Perangkat lunak memiliki antarmuka yang mudah dimengerti dan digunakan oleh pengguna.

4.2.3 Analisis dan Desain

Pada fase insepri, tahapan analisis dan desain adalah tahapan terpenting untuk memahami kebutuhan perangkat lunak yang tepat dalam menyelesaikan permasalahan penelitian.

4.2.3.1 Analisis Perangkat Lunak

Kegiatan yang dilakukan pada tahapan ini adalah menganalisis kebutuhan perangkat lunak, data, prapengolahan, pembobotan kata, metode perhitungan jarak dengan *cosine distance*, reduksi dimensi dengan *singular value decomposition*, clustering dengan *k-means*, *k-medoids*, dan *fuzzy c-means*.

a. Analisis Kebutuhan Perangkat Lunak

Dari pemodelan bisnis yang telah dijabarkan, untuk menyelesaikan permasalahan yang terjadi diperlukan perangkat lunak yang mampu mengelompokkan dokumen jurnal secara otomatis baik menggunakan *Singular Value Decomposition* ataupun tidak. Untuk itu, perangkat lunak harus memiliki kemampuan sebagai berikut.

1. menyamakan bentuk huruf;
2. menghilangkan simbol yang dianggap delimiter;
3. memecah kalimat menjadi bentuk string tunggal;
4. membuang kata-kata yang tidak penting atau tidak memiliki makna;
5. mengubah bentuk kata berimbuhan menjadi kata dasar;

6. mengetahui jumlah kemunculan setiap kata pada dokumen menggunakan perhitungan bobot kata;
7. melakukan reduksi dimensi pada dokumen menggunakan *singular value decomposition*;
8. melakukan clustering secara otomatis menggunakan pengukuran jarak kedekatan antar dokumen berdasarkan bobot kata dari setiap dokumen untuk metode *k-means*, *k-medoids*, dan *fuzzy c-means*.

Pengembangan perangkat lunak dimulai dengan mengumpulkan jurnal-jurnal dari internet yang diunduh dari website Indonesian Publication Index (id.portalgaruda.org). Selanjutnya file abstrak jurnal disimpan dalam format file *.txt*. Tahap selanjutnya ialah melakukan prapengolahan (*case folding*, *tokenizing*, *stop words removal*, dan *stemming*). Dokumen yang telah melewati tahap prapengolahan berubah menjadi kata. Setelah itu, diberi bobot dengan metode pembobotan *TF-IDF*. Matriks kata hasil pembobotan memiliki dimensi yang tinggi, sehingga dilakukan proses reduksi dimensi oleh metode *singular value decomposition*. Selanjutnya nilai hasil reduksi digunakan untuk menghitung jarak antar dokumen. Pengukuran jarak antar dokumen merupakan pendekatan yang dapat digunakan untuk menyelesaikan masalah tersebut. Pengukuran ini bertujuan untuk mengetahui kemiripan atau kesamaan dari dokumen jurnal.

Pengukuran tingkat kemiripan dua buah dokumen dilakukan menggunakan metode *Cosine Distance*. *Cosine Distance* merupakan metode yang cocok untuk data teks yang berdimensi besar dalam perhitungan jarak

antar dokumen. Dua buah dokumen dikatakan sama jika jarak kedekatannya sama dengan 0 (Huang A., 2011). Perhitungan jarak antar dokumen tersebut dapat dilakukan setelah setiap kata diberi bobot. Jika sudah memperoleh jarak antar dokumen maka dapat dilakukan clustering dengan menggunakan metode clustering.

b. Analisis Data

Data yang dibutuhkan pada perangkat lunak yang dibangun ada tiga yaitu data jurnal, kamus stopwords, dan kamus kata dasar bahasa Indonesia.

1. Data Jurnal

Data jurnal berfungsi sebagai data masukan pada proses clustering, berisi konten jurnal dengan mengabaikan tabel dan rumus. Data jurnal yang digunakan dalam tugas akhir ini diperoleh dari internet yang diunduh dari website Indonesian Publication Index (*id.portalgaruda.org*). Data yang diperoleh disimpan dalam format file *.txt* dan diberi nama sesuai topik jurnal. Data yang digunakan sebanyak 100 jurnal dengan jumlah term sebanyak 8088 kata.

2. Kamus *Stop Word*

Kamus *stop word* berisikan daftar kata yang tidak memiliki makna tetapi sering muncul dalam dokumen. Kamus disimpan dalam file berformat *.txt*.

3. Kamus Kata Dasar Bahasa Indonesia

Kamus kata dasar bahasa Indonesia berisikan daftar kata bahasa Indonesia yang baku digunakan untuk proses *stemming*. Kamus disimpan dalam file berformat *.txt*.

c. Analisis Prapengolahan

Prapengolahan yang digunakan dalam perangkat lunak yaitu *case folding*, *tokenizing*, *stop words removal*, dan *stemming*. Contoh dokumen yang digunakan sebanyak 4 buah dengan masing-masing berisi 1 kalimat.

D1= Sistem pakar untuk diagnosa penyakit kanker.

D2= Kesehatan masyarakat sangat tergantung kebersihan lingkungannya.

D3= Sistem pakar diagnosa penyakit hepatitis.

D4= Lingkungan yang tidak bersih akan merendahkan kualitas kesehatan masyarakatnya.

Hasil tahapan prapengolahan yang dilakukan pada keempat dokumen dapat dilihat di bawah ini.

1. Case Folding

Berikut adalah hasil *case folding* dari contoh dokumen.

D1= sistem pakar untuk diagnosa penyakit kanker

D2= kesehatan masyarakat sangat tergantung pada kebersihan lingkungannya

D3= sistem pakar diagnosa penyakit hepatitis

D4= lingkungan yang tidak bersih akan merendahkan kualitas kesehatan masyarakatnya

2. *Tokenizing*

Berikut adalah hasil *tokenizing* dari contoh dokumen yang digambarkan pada tabel IV-3.

Tabel IV-3 Hasil *Tokenizing* Contoh Dokumen

D1	D2	D3	D4
sistem	kesehatan	sistem	lingkungan
pakar	masyarakat	pakar	yang
untuk	sangat	diagnosa	tidak
diagnosa	tergantung	penyakit	bersih
penyakit	pada	hepatitis	akan
kanker	kebersihan		merendahkan
	lingkungannya		kualitas
			kesehatan
			masyarakatnya

3. *Stop Words Removal*

Berikut adalah hasil *stop words removal* dari contoh dokumen yang digambarkan pada tabel IV-4.

Tabel IV-4 Hasil *Stop Words Removal* Contoh Dokumen

D1	D2	D3	D4
sistem	kesehatan	sistem	lingkungan
pakar	masyarakat	pakar	bersih

diagnosa	tergantung	diagnosa	merendahkan
penyakit	kebersihan	penyakit	kualitas
kanker	lingkungannya	hepatitis	kesehatan
			masyarakatnya

4. *Stemming*

Berikut adalah hasil *stemming* dari contoh dokumen yang digambarkan pada tabel IV-5.

Tabel IV-5 Hasil *Stemming* Contoh Dokumen

D1	D2	D3	D4
sistem	sehat	sistem	lingkung
D1	D2	D3	D4
pakar	masyarakat	pakar	bersih
diagnosa	gantung	diagnosa	rendah
penyakit	bersih	penyakit	kualitas
kanker	lingkung	hepatitis	sehat
			masyarakat

d. Analisis Pembobotan Kata

Setelah tahapan prapengolahan selesai maka dilakukan tahapan pembobotan kata yaitu dengan menghitung frekuensi kata. Hasil prapengolahan yang didapatkan sebagai berikut.

D1 : sistem pakar diagnosa sakit kanker

D2 : sehat masyarakat gantung bersih lingkung

D3 : sistem pakar diagnosa sakit hepatitis

D4 : lingkung bersih rendah kualitas sehat masyarakat

Perhitungan bobot kata dapat dilakukan dengan menghitung tf-idf menggunakan pada persamaan sebagai berikut.

$$w_{ik} = tf_{ik} \times idf_k \quad (IV-1)$$

dimana, faktor tf yang digunakan adalah

$$tf_{ik} = freq_{ik} \quad (IV-2)$$

faktor idf adalah inverse dari nilai df yang memberikan nilai tinggi pada kata yang jarang muncul untuk suatu dokumen tertentu adalah

$$idf_k = \log \frac{N}{n_k} \quad (IV-3)$$

Keterangan:

w_{ik} : bobot kata k ;

tf_{ik} : banyaknya kemunculan suatu kata k dalam suatu dokumen;

df_k : banyaknya kemunculan suatu kata k dalam seluruh dokumen;

idf_k : banyaknya kemunculan suatu kata k dalam dokumen tertentu tetapi jarang muncul dalam dokumen lain;

$freq_{ik}$: jumlah kemunculan kata k pada sebuah dokumen D_i ;

N : jumlah keseluruhan dokumen;

n_k : jumlah dokumen yang mengandung kata k sedikitnya satu

Dengan persamaan IV-1, IV-2, dan IV-3 didapatkan hasil yang digambarkan pada tabel IV-6.

Tabel IV-6 Hasil Pembobotan Kata Contoh Dokumen

Keterangan :

$$- idf_{(sistem, pakar, diagnosa, sakit, sehat, masyarakat, bersih, lingkung)}$$

$$= \log \frac{4}{2} = 0.3010$$

$$- idf_{(kanker, gantung, hepatitis, rendah, kualitas)} = \log \frac{4}{1} = 0.6020$$

Term	D1	D2	D3	D4	df	idf	$tf_{ik} \times idf_k$			
							D1	D2	D3	D4
sistem	1	0	1	0	2	0.3010	0.3010	0	0.3010	0
pakar	1	0	1	0	2	0.3010	0.3010	0	0.3010	0
diagnosa	1	0	1	0	2	0.3010	0.3010	0	0.3010	0
sakit	1	0	1	0	2	0.3010	0.3010	0	0.3010	0
kanker	1	0	0	0	1	0.6020	0.6020	0	0	0
sehat	0	1	0	1	2	0.3010	0	0.3010	0	0.3010
masyarakat	0	1	0	1	2	0.3010	0	0.3010	0	0.3010
gantung	0	1	0	0	1	0.6020	0	0.6020	0	0
bersih	0	1	0	1	2	0.3010	0	0.3010	0	0.3010
lingkung	0	1	0	1	2	0.3010	0	0.3010	0	0.3010
hepatitis	0	0	1	0	1	0.6020	0	0	0.6020	0
rendah	0	0	0	1	1	0.6020	0	0	0	0.6020
kualitas	0	0	0	1	1	0.6020	0	0	0	0.6020

Setelah dilakukan pembobotan kata, matriks hasil pembobotan diproses tergantung dengan pilihan pengguna. Apabila pengguna memilih melakukan clustering tanpa proses reduksi dimensi, maka matriks hasil pembobotan akan langsung digunakan untuk masukan metode *cosine distance*. Untuk kemudian dilakukan clustering. Sebaliknya, apabila pengguna memilih melakukan clustering dengan proses reduksi dimensi, maka matriks hasil pembobotan kata direduksi menggunakan metode *singular value decomposition*. Pada penjelasan ini, penulis menjabarkan cara kerja proses clustering dokumen dengan *singular value decomposition*.

e. Analisis Metode Singular Value Decomposition

Proses clustering kalimat pada penelitian ini bertujuan untuk memperoleh cluster-cluster dokumen dimana masing-masing cluster berisi kumpulan dokumen yang digabung berdasarkan tingkat rasio kemiripan. Metode *singular value decomposition* (SVD) digunakan untuk mendapatkan tingkat kemiripan antardokumen dengan mencocokkan hubungan antara term-term disuatu dokumen dengan dokumen yang lain. Bentuk komputasi yang dilakukan SVD yaitu mengubah matriks hasil pembobotan kata yang diperoleh pada tahap sebelumnya menjadi 3 buah matriks yakni, matriks U (matriks baris), matriks Σ (matriks singular) dan matriks V^t (matriks kolom). Pada perangkat lunak ini menggunakan metode *Singular Value Decomposition* yang ada pada package colt-1.2.0 untuk melakukan perhitungan metode aljabar matriks *Singular Value Decomposition*. Untuk menggunakan metode tersebut, diperlukan matriks yang berasal dari hasil

pembobotan kata. Setelah matriks pembobotan kata tersebut terbentuk, maka SVD akan memecah matriks tersebut menjadi U , Σ , dan V^t . Ketika ketiga matriks tersebut telah terbentuk, maka tingkat kemiripan dari antara kalimat ke- i dan ke- j dapat dihitung dengan Cosine Distance. Contoh perhitungan tingkat kemiripan antarkalimat menggunakan *Singular Value Decomposition* dapat dilihat pada Tabel IV-7 .

Tabel IV-7 Contoh Perhitungan *Singular Value Decomposition*

Hasil pemecahan matriks pembobotan kata menggunakan SVD ($U\Sigma V^t$) :			
Matriks U :			
1.966374E-017	0.408248	1.966374E-017	-5.551115E-017
1.401548E-017	0.408248	1.401548E-017	1.387779E-016
-5.257865E-017	0.408248	-5.257865E-017	5.551115E-017
6.299813E-018	0.408248	6.299813E-018	5.551115E-017
6.299813E-018	0.408248	6.299813E-018	-0.707107
0.361803	0	-0.138197	0
0.361803	0	-0.138197	0
0.276393	0	-0.138197	0
0.361803	0	-0.138197	0
0.361803	0	-0.138197	0
6.299813E-018	0.408248	6.299813E-018	0.707107
0.447214	0	0.447214	0
0.447214	0	0.447214	0

Matriks S :

1.145186	0	0	0
0	1.042798	0	0
0	0	0.707764	0
0	0	0	0.60206

Matriks V^t :

0	0.525731	0	0.850651
0.707107	0	0.707107	0
0	-0.850651	0	0.525731
-0.707107	0	0.707107	0

Kata-kata dalam matriks pembobotan kata tersebut diwakili pada vektor-vektor baris dari U sedangkan kalimat diwakili pada kolom-kolom dari vektor V^t . Untuk menskalakan kedua vektor dilakukan pengkalian nilai-nilai singular pada Σ . Dengan demikian representasi dari kata-kata tersebut diperoleh dari vektor baris $U\Sigma$ dan kalimat diperoleh dari vektor kolom ΣV^t .

Matriks $U\Sigma$:

$$\text{sistem} = \begin{bmatrix} 2.2518638050104062E - 17 \\ 0.4257207025491161 \\ 1.391728369532097E - 17 \\ -3.3421043228896296E - 17 \end{bmatrix},$$

$$\text{pakar} = \begin{bmatrix} 1.6050330949228712E - 17 \\ 0.4257207025491162 \\ 9.91965005730771E - 18 \\ 8.355260807224074E - 17 \end{bmatrix},$$

$$\text{diagnosa} = \begin{bmatrix} -6.021234638919387E - 17 \\ 0.42572070254911615 \\ -3.721327661090382E - 17 \\ 3.3421043228896296E - 17 \end{bmatrix},$$

$$\text{sakit} = \begin{bmatrix} 7.214459129953694E - 18 \\ 0.42572070254911615 \\ 4.458780952758379E - 18 \\ 3.3421043228896296E - 17 \end{bmatrix},$$

$$\text{kanker} = \begin{bmatrix} 7.21445912995369E - 18 \\ 0.4257207025491162 \\ 4.458780952758376E - 18 \\ -0.42572070254911615 \end{bmatrix},$$

$$\text{sehat} = \begin{bmatrix} 0.4143322435514195 \\ 0.0 \\ -0.09781057474813386 \\ 0.0 \end{bmatrix},$$

$$\text{masyarakat} = \begin{bmatrix} 0.4143322435514195 \\ 0.0 \\ -0.09781057474813386 \\ 0.0 \end{bmatrix},$$

$$\text{gantung} = \begin{bmatrix} 0.3165216688032857 \\ 0.0 \\ -0.5121428182995534 \\ 0.0 \end{bmatrix},$$

$$\text{bersih} = \begin{bmatrix} 0.4143322435514195 \\ 0.0 \\ -0.09781057474813386 \\ 0.0 \end{bmatrix}$$

$$\text{lingkung} = \begin{bmatrix} 0.4143322435514195 \\ 0.0 \\ -0.09781057474813386 \\ 0.0 \end{bmatrix},$$

$$\text{hepatitis} = \begin{bmatrix} 7.21445912995369E - 18 \\ 0.4257207025491162 \\ 4.458780952758376E - 18 \\ 0.42572070254911615 \end{bmatrix},$$

$$= \frac{\begin{bmatrix} 2.2518638050104062E-17 \\ 0.4257207025491161 \\ 1.391728369532097E-17 \\ -3.3421043228896296E-17 \end{bmatrix} + \begin{bmatrix} 1.6050330949228712E-17 \\ 0.4257207025491162 \\ 9.91965005730771E-18 \\ 8.355260807224074E-17 \end{bmatrix} + \begin{bmatrix} -6.021234638919387E-17 \\ 0.42572070254911615 \\ -3.721327661090382E-17 \\ 3.3421043228896296E-17 \end{bmatrix} + \begin{bmatrix} 7.214459129953694E-18 \\ 0.42572070254911615 \\ 4.458780952758379E-18 \\ 3.3421043228896296E-17 \end{bmatrix}}{5}$$

$$\text{rendah} = \begin{bmatrix} 0.5121428182995534 \\ 0.0 \\ 0.3165216688032857 \\ 0.0 \end{bmatrix},$$

$$\text{kualitas} = \begin{bmatrix} 0.5121428182995534 \\ 0.0 \\ 0.3165216688032857 \\ 0.0 \end{bmatrix}.$$

Matriks ΣV^t :

$$\text{Kalimat 1 (D1)} = \begin{bmatrix} 0.0 \\ 0.7373698866489866 \\ 0.0 \\ -0.42572070254911615 \end{bmatrix},$$

$$\text{Kalimat 2 (D2)} = \begin{bmatrix} 0.6020599913279626 \\ 0.0 \\ -0.6020599913279626 \\ 0.0 \end{bmatrix},$$

$$\text{Kalimat 3 (D3)} = \begin{bmatrix} 0.0 \\ 0.7373698866489865 \\ 0.0 \\ 0.42572070254911626 \end{bmatrix}.$$

Contoh perhitungan tingkat kemiripan antara kalimat ke-1 dan kalimat ke-2 :

D1: sistem pakar diagnosa sakit kanker

D2: sehat masyarakat gantung bersih lingkung

$D_1 =$

$$= \begin{bmatrix} -1.4428918259907428E - 18 \\ 0.42572070254911615 \\ -8.91756190551677E - 19 \\ -0.0851441405098232 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 0.6020599913279626 \\ 0.0 \\ -0.6020599913279626 \\ 0.0 \end{bmatrix}$$

$$|D_1| = \sqrt{-1.4428918259907428E - 0.42572070254911615^2 + -8.91756190551677E + -0.0851441405098232^2}$$

$$|D_2| = \sqrt{0.6020599913279626^2 + 0.0^2 + -0.6020599913279626^2 + 0.0^2}$$

$$\text{Distance}(\text{Kal}_1, \text{Kal}_2) = \frac{D_1 \cdot D_2}{|D_1| |D_2|} = \frac{0.3501612494985513}{0.3696546772170727} = 0.05273415682246119$$

Setelah masing-masing jarak antarkalimat dihitung, maka dibentuklah matriks reduksi dimensi yang digunakan sebagai masukan dalam proses clustering.

<i>Distance</i>	D1	D2	D3	D4
D1		1.0	0.2488502919 6064523	1.0
D2	1.0		1.0	0.299061052 82507645
D3	0.248850291 96064523	1.0		1.0
D4	1.0	0.2990610528 2507645	1.0	

Setelah reduksi dimensi selesai, maka proses selanjutnya adalah melakukan *cluster* dokumen menggunakan metode-metode clustering (k-means, k-medoids, atau fuzzy c-means).

f. Analisis Metode *K-Means*

Dengan mengikuti algoritma clustering k-means pada sub bab 2.3 didapatkan Hasil clustering yang didapatkan ditunjukkan pada tabel IV-8 dengan pusat cluster seluruh term dari contoh kalimat pada tabel IV-9. Perhitungan lengkapnya terdapat pada lampiran II.

Tabel IV-8 Hasil Clustering Kalimat menggunakan k-means

Cluster A	Cluster B
D1	D2
D3	D4

Tabel IV-9 Pusat Cluster k-means

Cluster A	Cluster B
0.3010299956639812	0.0
0.3010299956639812	0.0
0.3010299956639812	0.0
0.3010299956639812	0.0
0.3010299956639812	0.0
0.0	0.3010299956639812
0.0	0.3010299956639812
0.0	0.3010299956639812
0.0	0.3010299956639812
0.0	0.3010299956639812
0.3010299956639812	0.0

0.0	0.3010299956639812
0.0	0.3010299956639812

g. Analisis Metode *K-Medoids*

Hasil reduksi dimensi digunakan sebagai masukan untuk proses clustering mengikuti algoritma *k-medoids* pada sub bab 2.4. Hasil clustering yang didapatkan ditunjukkan pada Tabel IV-10 dengan pusat cluster D3 dan D2. Perhitungan lengkapnya terdapat pada lampiran II.

Tabel IV-10 Hasil Clustering Kalimat menggunakan *k-medoids*

Cluster A	Cluster B
D1	D2
D3	D4

h. Analisis *Fuzzy C-Means*

Dengan mengikuti algoritma clustering fuzzy c-means pada sub bab 2.5. Hasil clustering yang didapatkan ditunjukkan pada tabel IV-11 dengan pusat cluster seluruh term dari contoh kalimat pada tabel IV-12. Perhitungan lengkapnya terdapat pada lampiran II.

Tabel IV-11 Hasil Clustering Kalimat menggunakan fuzzy c-means

Cluster A	Cluster B
D1	D2
D3	D4

Tabel IV-12 Pusat Cluster fuzzy c-means

Cluster A	Cluster B
0.005810328067836311	0.2913816893759715
0.005810328067836311	0.2913816893759715
0.005810328067836311	0.2913816893759715
0.005810328067836311	0.2913816893759715
0.005810325740239751	0.29138170532586294
0.2952196675961449	0.00964830628800971
0.2952196675961449	0.00964830628800971
0.24959415289448436	0.016932370377333562
0.2952196675961449	0.00964830628800971
0.2952196675961449	0.00964830628800971
0.0058103303954328694	0.29138167342608007
0.3408451822978054	0.0023642421986858585
0.3408451822978054	0.0023642421986858585

4.2.3.2 Desain Perangkat Lunak

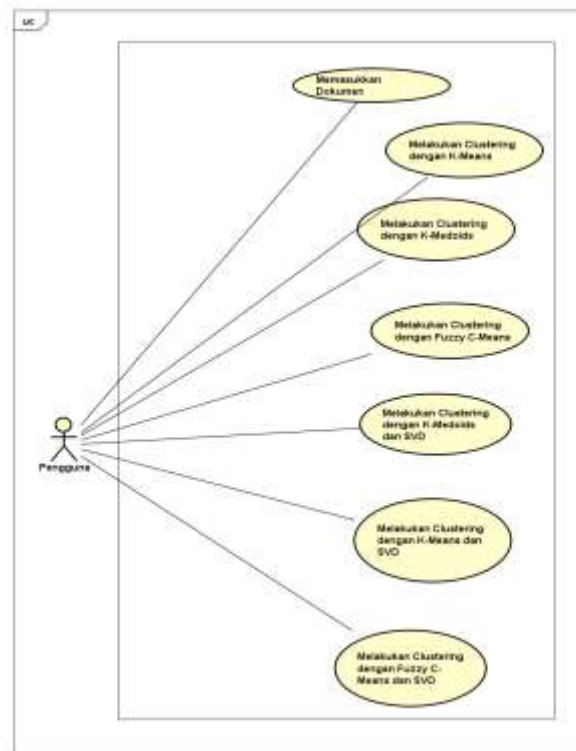
Desain perangkat lunak digambarkan dengan diagram *use case* dan diagram aktivitas.

1. Model *Use Case*

Pada subbab ini dijelaskan gambaran fungsionalitas perangkat lunak yang dibangun dengan menggunakan pemodelan *Use Case*.

a) Diagram *Use Case*

Diagram *Use Case* menjelaskan secara umum kegiatan yang dilakukan oleh aktor (pengguna) terhadap perangkat lunak yang dapat dilihat pada gambar IV-2.



Gambar IV-2. Diagram *Use Case*

b) Tabel Definisi Aktor

Dalam penelitian ini, penulis, dosen pembimbing, dan penguji menjadi aktor yang dijelaskan pada tabel IV-13.

Tabel IV-13 Definisi Aktor *Use Case*

Nomor	Use Case	Definisi
1	Pengguna	Pengguna adalah orang yang berhubungan dengan perangkat lunak aplikasi untuk

		memasukkan dokumen, melakukan prapengolahan, reduksi dimensi, dan clustering dokumen berbahasa Indonesia
--	--	--

c) Tabel Definisi *Use Case*

Definisi *Use case* yang dijelaskan pada tabel IV-14 merupakan penjelasan dari kerja perangkat lunak secara spesifik pada gambar IV-2 disimbolkan dengan sebuah notasi lingkaran yang agak lonjong atau berbentuk oval.

Tabel IV-14 Definisi *Use Case*

Nomor	<i>Use Case</i>	Deskripsi
1	Memasukkan Dokumen	Kegiatan memilih dan memasukkan dokumen untuk dilakukan proses prapengolahan tahapan <i>case folding, tokenizing, filtering, stemming</i> , dan pembobotan kata.
2	Melakukan Clustering dengan K-means	Kegiatan ini digunakan untuk melakukan clustering dokumen menggunakan metode K-means.
3	Melakukan Clustering dengan K-means dan	Kegiatan ini digunakan untuk melakukan clustering dokumen menggunakan metode K-means, dengan terlebih dahulu dilakukan reduksi dimensi pada hasil prapengolahan.

	Singular Value Decomposition (SVD)	
4	Melakukan Clustering dengan K- medoids	Kegiatan ini digunakan untuk melakukan clustering dokumen menggunakan metode K-medoids.
5	Melakukan Clustering dengan K- medoids dan Singular Value Decomposition (SVD)	Kegiatan ini digunakan untuk melakukan clustering dokumen menggunakan metode K-medoids, dengan terlebih dahulu dilakukan reduksi dimensi pada hasil prapengolahan.
6	Melakukan Clustering dengan Fuzzy c-means	Kegiatan ini digunakan untuk melakukan clustering dokumen menggunakan metode Fuzzy c-means.
7	Melakukan Clustering dengan Fuzzy c-means dan Singular Value	Kegiatan ini digunakan untuk melakukan clustering dokumen menggunakan metode Fuzzy c-means, dengan terlebih dahulu dilakukan reduksi dimensi pada hasil prapengolahan.

	Decomposition (SVD)	
--	------------------------	--

d) Skenario *Use Case*

Skenario adalah urutan spesifik dari aksi dan interaksi antara aktor dan sistem. Berikut ini adalah skenario dari *use case* yang telah didefinisikan pada subbab sebelumnya.

1. Skenario *Use Case* Memasukkan Dokumen

No	: 001
Nama <i>Use Case</i>	: Memasukkan Dokumen
Aktor	: Pengguna
Tujuan	: Mengubah dokumen menjadi data yang siap
	untuk di <i>clustering</i>
Deskripsi	: <i>Use Case</i> ini digunakan pada proses awal pembentukan data yang digunakan untuk proses <i>clustering</i>
Kondisi Awal	: Belum terdapat data dokumen
Kondisi Akhir	: Mendapatkan data dokumen yang telah dilakukan prapengolahan

Skenario *use case* Memasukkan Dokumen dijelaskan pada tabel IV-15 .

Tabel IV-15 Skenario Memasukkan Dokumen

Aktor	Sistem
-------	--------

1. Pengguna menekan tombol “Muat”	
	2. Menampilkan <i>window</i> yang menunjukkan direktori penyimpanan dokumen
3. Pengguna memilih folder dokumen yang akan diproses	
	4. Melakukan prapengolahan terhadap kumpulan dokumen (casefolding, tokenizing, stopwords removal, stemming, dan pembobotan).
	5. Menampilkan direktori penyimpanan dokumen
	6. Mengaktifkan <i>radio button</i> untuk proses clustering

2. Skenario *Use Case* Melakukan Clustering dengan K-means

No : 002

Nama Use Case : Melakukan Clustering dengan K-means

Aktor : Pengguna

Tujuan : Mengetahui kelompok dokumen-dokumen

Deskripsi : Use case ini digunakan untuk melakukan *clustering* dengan metode K-means untuk mengelompokkan dokumen-dokumen ke dalam *cluster*.

Kondisi Awal : Terdapat dokumen yang sudah dilakukan pembobotan

Kondisi Akhir : Hasil *clustering* dokumen

Skenario *use case* Melakukan Clustering dengan K-means dijelaskan pada tabel IV-16.

Tabel IV-16 Skenario *Use Case* Melakukan Clustering dengan K-means

Aktor	Sistem
1. Pengguna memilih metode <i>k-means</i> pada <i>radio button</i>	
2. Pengguna memasukkan jumlah cluster yang ingin dibentuk	
3. Pengguna menekan tombol “Mulai”	
Aktor	Sistem
	4. Menentukan centroid secara acak
	5. Membentuk hasil clustering sementara
	6. Menentukan <i>centroid</i> baru
	7. Memilih centroid terbaik
	8. Menampilkan hasil clustering
Skenario Alternatif (Pengguna tidak memasukkan jumlah cluster)	
1. Pengguna tidak memasukkan jumlah cluster	
	2. Menampilkan pesan “masukkan jumlah cluster”
Skenario Alternatif (Pengguna memasukkan tipe data nilai <i>k</i> yang salah)	

1. Pengguna memasukkan nilai k selain tipe data <i>integer</i>	
	2. Menampilkan pesan “ Nilai k harus angka”
Skenario Alternatif (Pengguna memasukkan nilai k yang melebihi jumlah dokumen)	
1. Pengguna memasukkan nilai yang lebih besar dari jumlah data	
	2. Menampilkan pesan “ Nilai k harus lebih kecil dari jumlah data”

3. Skenario *Use Case* Melakukan Clustering dengan K - Medoids

No : 003

Nama Use Case : Melakukan *Clustering* dengan K - Medoids

Aktor : Pengguna

Tujuan : Mengetahui kelompok dokumen-dokumen

Deskripsi : Use case ini digunakan untuk melakukan *clustering* dengan metode K - Medoids untuk mengelompokkan dokumen-dokumen ke dalam *cluster*.

Kondisi Awal : Terdapat dokumen yang sudah dilakukan pembobotan

Kondisi Akhir : Hasil *clustering* dokumen

Skenario *use case* melakukan clustering dengan k-medoids dijelaskan pada tabel IV-17.

Tabel IV-17 Skenario *Use Case* Melakukan Clustering dengan K - Medoids

Aktor	Sistem
1. Pengguna memilih metode <i>k-medoids</i> pada <i>radio button</i>	
2. Pengguna memasukkan jumlah cluster yang ingin dibentuk	
3. Pengguna menekan tombol “Mulai”	
	4. Menentukan <i>medoid</i>
	5. Menghitung jarak antar dokumen
	6. Membentuk hasil clustering sementara
	7. Menentukan <i>medoid</i> baru
	8. Memilih <i>medoid</i> terbaik
	9. Menampilkan hasil clustering
Skenario Alternatif (Pengguna tidak memasukkan jumlah cluster)	
1. Pengguna tidak memasukkan jumlah cluster	
Skenario Alternatif (Pengguna tidak memasukkan jumlah cluster)	
	2. Menampilkan pesan “ masukkan jumlah cluster”
Skenario Alternatif (Pengguna memasukkan tipe data nilai <i>k</i> yang salah)	
1. Pengguna memasukkan nilai <i>k</i> selain tipe data <i>integer</i>	
	2. Menampilkan pesan “ Nilai <i>k</i> harus angka”
Skenario Alternatif (Pengguna memasukkan nilai <i>k</i> yang melebihi jumlah dokumen)	
1. Pengguna memasukkan nilai yang lebih besar dari jumlah data	

	2. Menampilkan pesan “ Nilai k harus lebih kecil dari jumlah data”
--	--

3. Skenario *Use Case* Melakukan Clustering dengan Fuzzy C-Means

No : 004

Nama Use Case : Melakukan *Clustering* dengan Fuzzy C-Means

Aktor : Pengguna

Tujuan : Mengetahui kelompok dokumen-dokumen

Deskripsi : Use case ini digunakan untuk melakukan *clustering* dengan metode Fuzzy C-Means untuk mengelompokkan dokumen-dokumen ke dalam *cluster*.

Kondisi Awal : Terdapat dokumen yang sudah dilakukan pembobotan

Kondisi Akhir : Hasil *clustering* dokumen

Skenario *use case* melakukan Melakukan *Clustering* dengan Fuzzy C-Means dijelaskan pada tabel IV-18.

Tabel IV-18 Skenario *Use Case* Melakukan *Clustering* dengan Fuzzy C-Means

Aktor	Sistem
1. Pengguna memilih metode <i>fuzzy c - means</i> pada <i>radio button</i>	
3. Pengguna memasukkan jumlah cluster yang ingin dibentuk	

4. Pengguna menekan tombol “Mulai”	
	5. Menentukan matriks U
	6. Menghitung pusat kelompok
	7. Menghitung fungsi objektif
	8. Menghitung perubahan matriks keanggotaan
	9. Memeriksa kondisi berhenti
	10. Menampilkan hasil clustering
Skenario Alternatif (Pengguna tidak memasukkan jumlah cluster)	
1. Pengguna tidak memasukkan jumlah cluster	
	2. Menampilkan pesan “ masukkan jumlah cluster”
Skenario Alternatif (Pengguna memasukkan tipe data nilai k yang salah)	
1. Pengguna memasukkan nilai k selain tipe data <i>integer</i>	
	2. Menampilkan pesan “ Nilai k harus angka”
Skenario Alternatif (Pengguna memasukkan nilai k yang melebihi jumlah dokumen)	
1. Pengguna memasukkan nilai yang lebih besar dari jumlah data	
	2. Menampilkan pesan “ Nilai k harus lebih kecil dari jumlah data”

5. Skenario *Use Case* Melakukan Clustering dengan K-means dan SVD

No	: 005
Nama Use Case	: Melakukan Clustering dengan K-means dan SVD
Aktor	: Pengguna
Tujuan	: Mengetahui kelompok dokumen-dokumen
Deskripsi	: Use case ini digunakan untuk melakukan <i>clustering</i> dengan metode K-means dan SVD untuk mengelompokkan dokumen-dokumen ke dalam <i>cluster</i> .
Kondisi Awal	: Terdapat dokumen yang sudah dilakukan pembobotan
Kondisi Akhir	: Hasil <i>clustering</i> dokumen

Skenario *use case* Melakukan Clustering dengan K-means dan SVD dijelaskan pada tabel IV-19.

Tabel IV-19 Skenario *Use Case* Melakukan Clustering dengan K-means dan SVD

Aktor	Sistem
1. Pengguna memilih metode <i>k-means</i> dan SVD pada <i>radio button</i>	
2. Pengguna memasukkan jumlah cluster yang ingin dibentuk	
3. Pengguna menekan tombol “Mulai”	
	4. Menghitung nilai matriks U, S, dan V menggunakan singular value

	decomposition
	5. Melakukan perkalian matriks U dan S
	6. Melakukan perkalian matriks S dan V
	7. Menentukan centroid secara acak
	8. Membentuk hasil clustering sementara
	9. Menentukan <i>centroid</i> baru
	10. Memilih centroid terbaik
	11. Menampilkan hasil clustering
Skenario Alternatif (Pengguna tidak memasukkan jumlah cluster)	
1. Pengguna tidak memasukkan jumlah cluster	
	2. Menampilkan pesan “ masukkan jumlah cluster”
Skenario Alternatif (Pengguna memasukkan tipe data nilai k yang salah)	
1. Pengguna memasukkan nilai k selain tipe data <i>integer</i>	
	2. Menampilkan pesan “ Nilai k harus angka”
Skenario Alternatif (Pengguna memasukkan nilai k yang melebihi jumlah dokumen)	
1. Pengguna memasukkan nilai yang lebih besar dari jumlah data	
	2. Menampilkan pesan “ Nilai k harus

	lebih kecil dari jumlah data”
--	-------------------------------

6. Skenario *Use Case* Melakukan Clustering dengan K-medoids dan SVD

No : 006

Nama Use Case : Melakukan Clustering dengan K-medoids dan SVD

Aktor : Pengguna

Tujuan : Mengetahui kelompok dokumen-dokumen

Deskripsi : Use case ini digunakan untuk melakukan *clustering* dengan metode K-medoids dan SVD untuk mengelompokkan dokumen-dokumen ke dalam *cluster*.

Kondisi Awal : Terdapat dokumen yang sudah dilakukan pembobotan

Kondisi Akhir : Hasil *clustering* dokumen

Skenario *use case* Melakukan Clustering dengan K-medoids dan SVD dijelaskan pada tabel IV-20.

Tabel IV-20 Skenario *Use Case* Melakukan Clustering dengan K-medoids dan SVD

Aktor	Sistem
1. Pengguna memilih metode <i>k-medoids</i> dan SVD pada <i>radio button</i>	

2. Pengguna memasukkan jumlah cluster yang ingin dibentuk	
3. Pengguna menekan tombol “Mulai”	
	4. Menghitung nilai matriks U, S, dan V menggunakan singular value decomposition
	5. Melakukan perkalian matriks U dan S
	6. Melakukan perkalian matriks S dan V
	7. Menentukan medoid
	8. Menghitung jarak antar dokumen
	9. Membentuk hasil clustering sementara
Aktor	Sistem
	10. Menentukan medoid baru
	11. Memilih medoid terbaik
	12. Menampilkan hasil clustering
Skenario Alternatif (Pengguna tidak memasukkan jumlah cluster)	
1. Pengguna tidak memasukkan jumlah cluster	
	2. Menampilkan pesan “ masukkan jumlah cluster”

Skenario Alternatif (Pengguna memasukkan tipe data nilai k yang salah)	
1. Pengguna memasukkan nilai k selain tipe data <i>integer</i>	
	2. Menampilkan pesan “ Nilai k harus angka”
Skenario Alternatif (Pengguna memasukkan nilai k yang melebihi jumlah dokumen)	
1. Pengguna memasukkan nilai yang lebih besar dari jumlah data	
	2. Menampilkan pesan “ Nilai k harus lebih kecil dari jumlah data”

7. Skenario *Use Case* Melakukan Clustering dengan Fuzzy C-Means dan SVD

No : 007

Nama Use Case : Melakukan Clustering dengan Fuzzy C-Means dan SVD

Aktor : Pengguna

Tujuan : Mengetahui kelompok dokumen-dokumen

Deskripsi : Use case ini digunakan untuk melakukan *clustering* dengan metode Fuzzy C-Means dan SVD untuk mengelompokkan dokumen-dokumen ke dalam *cluster*.

Kondisi Awal : Terdapat dokumen yang sudah dilakukan pembobotan

Kondisi Akhir : Hasil *clustering* dokumen

Skenario *use case* Melakukan Clustering dengan Fuzzy C-Means dan SVD dijelaskan pada tabel IV-21.

Tabel IV-21 Skenario *Use Case* Melakukan Clustering dengan Fuzzy C-Means dan SVD

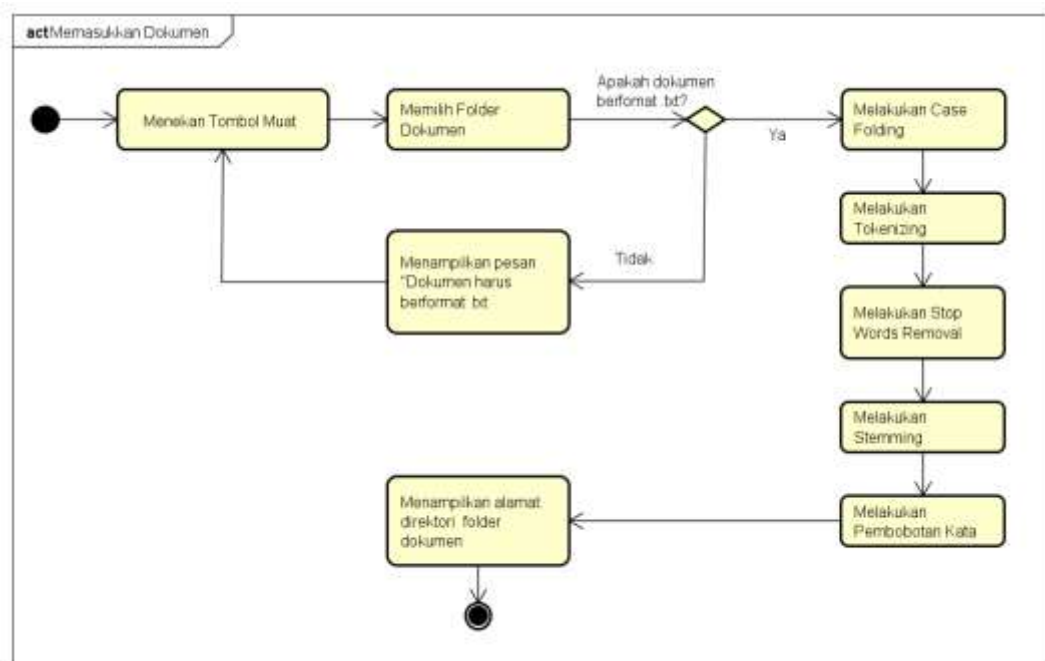
Aktor	Sistem
1. Pengguna memilih metode Fuzzy C-Means dan SVD pada <i>radio button</i>	
2. Pengguna memasukkan jumlah cluster yang ingin dibentuk	
3. Pengguna menekan tombol “Mulai”	
Aktor	Sistem
	4. Menghitung nilai matriks U, S, dan V menggunakan singular value decomposition
	5. Melakukan perkalian matriks U dan S
	6. Melakukan perkalian matriks S dan V
	7. Menentukan matriks U
	8. Menghitung pusat kelompok

	9. Menghitung fungsi objektif
	10. Menghitung perubahan matriks keanggotaan
	11. Memeriksa kondisi berhenti
	12. Menampilkan hasil clustering
Skenario Alternatif (Pengguna tidak memasukkan jumlah cluster)	
1. Pengguna tidak memasukkan jumlah cluster	
	2. Menampilkan pesan “ masukkan jumlah cluster”
Skenario Alternatif (Pengguna memasukkan tipe data nilai k yang salah)	
1. Pengguna memasukkan nilai k selain tipe data <i>integer</i>	
	2. Menampilkan pesan “ Nilai k harus angka”
Skenario Alternatif (Pengguna memasukkan nilai k yang melebihi jumlah dokumen)	
1. Pengguna memasukkan nilai yang lebih besar dari jumlah data	
	2. Menampilkan pesan “ Nilai k harus lebih kecil dari jumlah data”

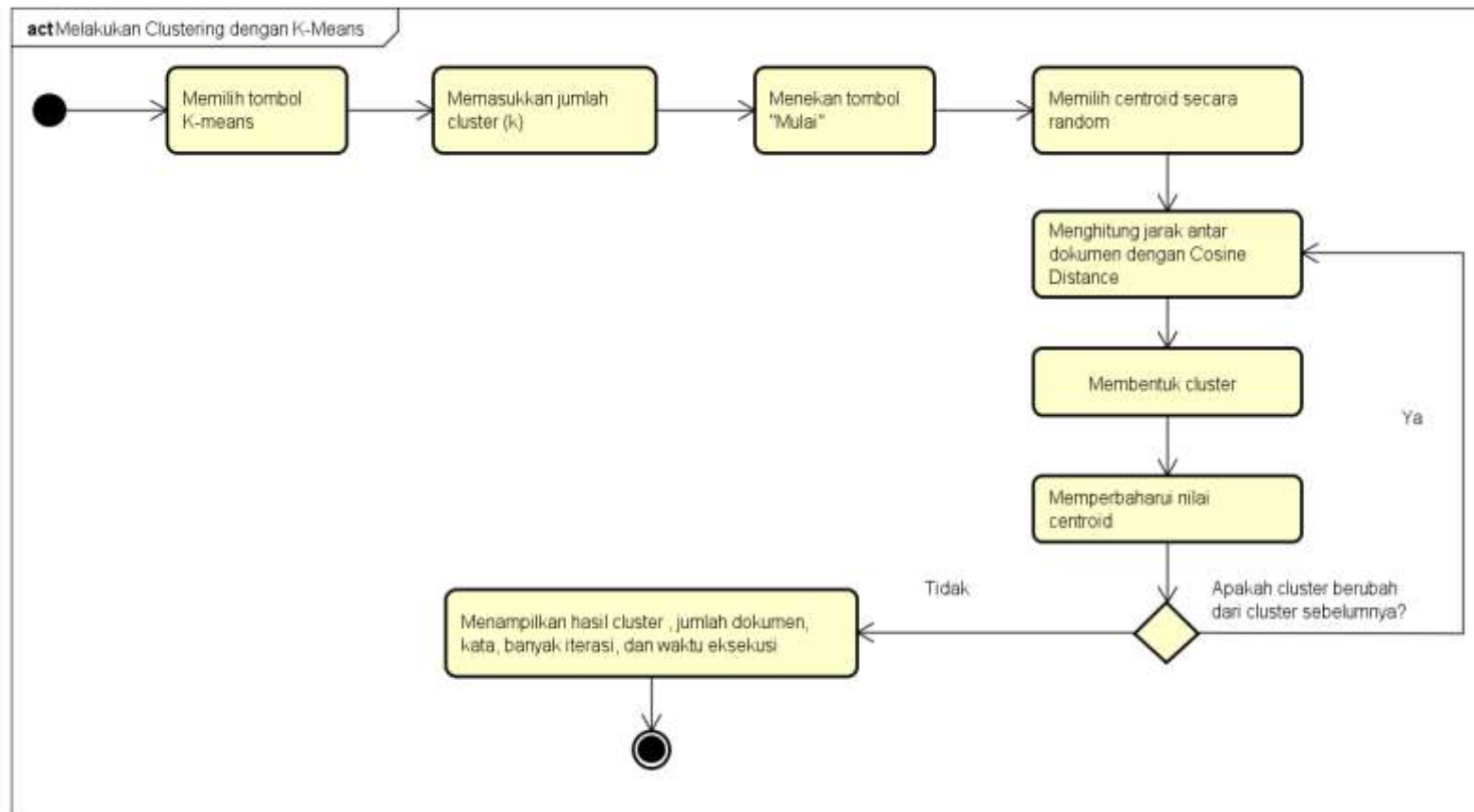
2. Diagram Aktivitas

Diagram aktivitas menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Diagram aktivitas memasukkan dokumen dapat dilihat

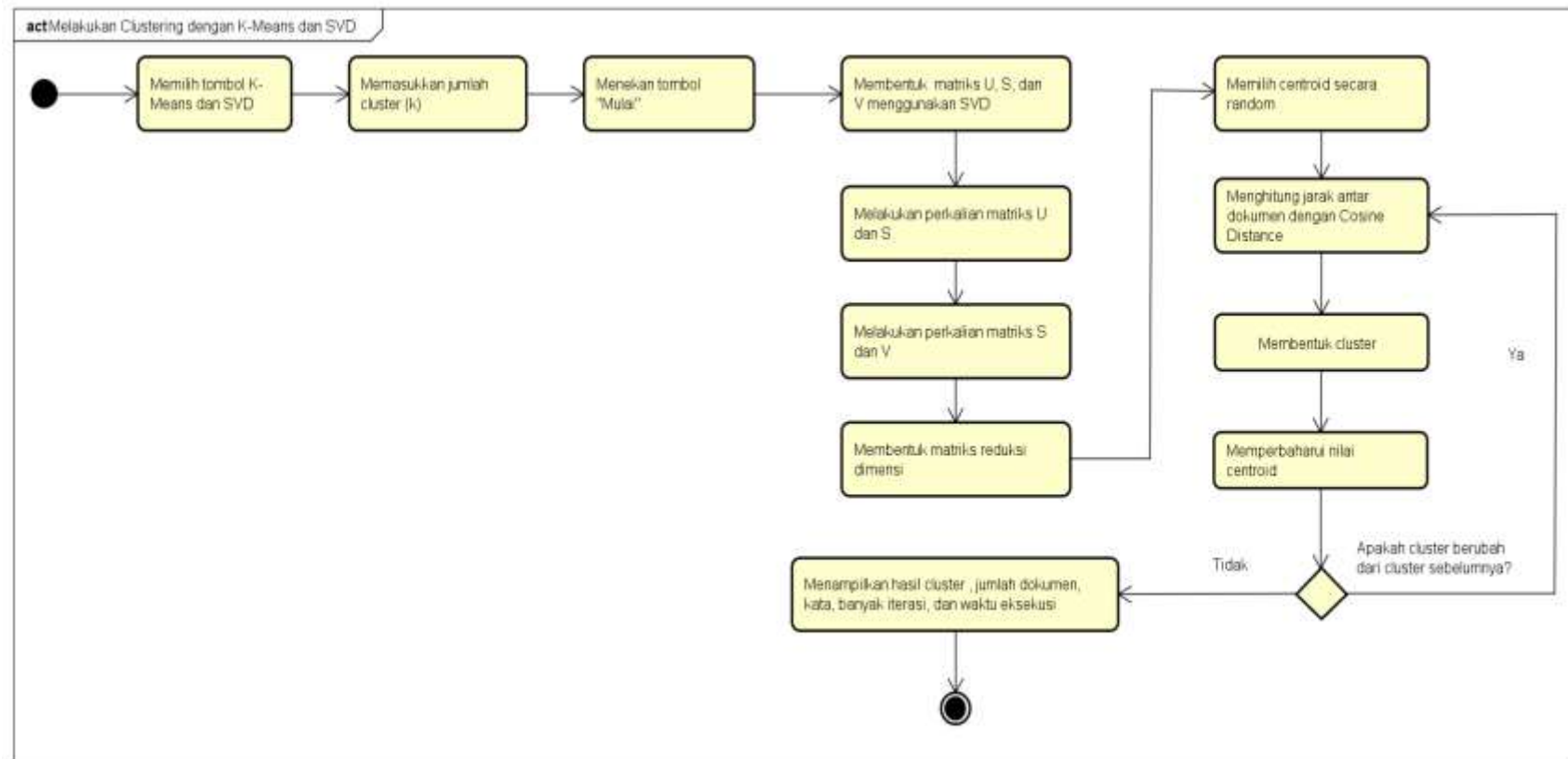
pada gambar IV-3, diagram aktivitas melakukan clustering dengan k-means dapat dilihat pada gambar IV-4, diagram aktivitas melakukan clustering dengan k-means dan SVD dapat dilihat pada gambar IV-5, diagram aktivitas melakukan clustering dengan k-medoids dapat dilihat pada gambar IV-6, diagram aktivitas melakukan clustering dengan k-medoids dan SVD dapat dilihat pada gambar IV-7, diagram aktivitas melakukan clustering dengan fuzzy c-means dapat dilihat pada gambar IV-8, dan diagram aktivitas melakukan clustering dengan fuzzy c-means dan SVD pada gambar IV-9 .



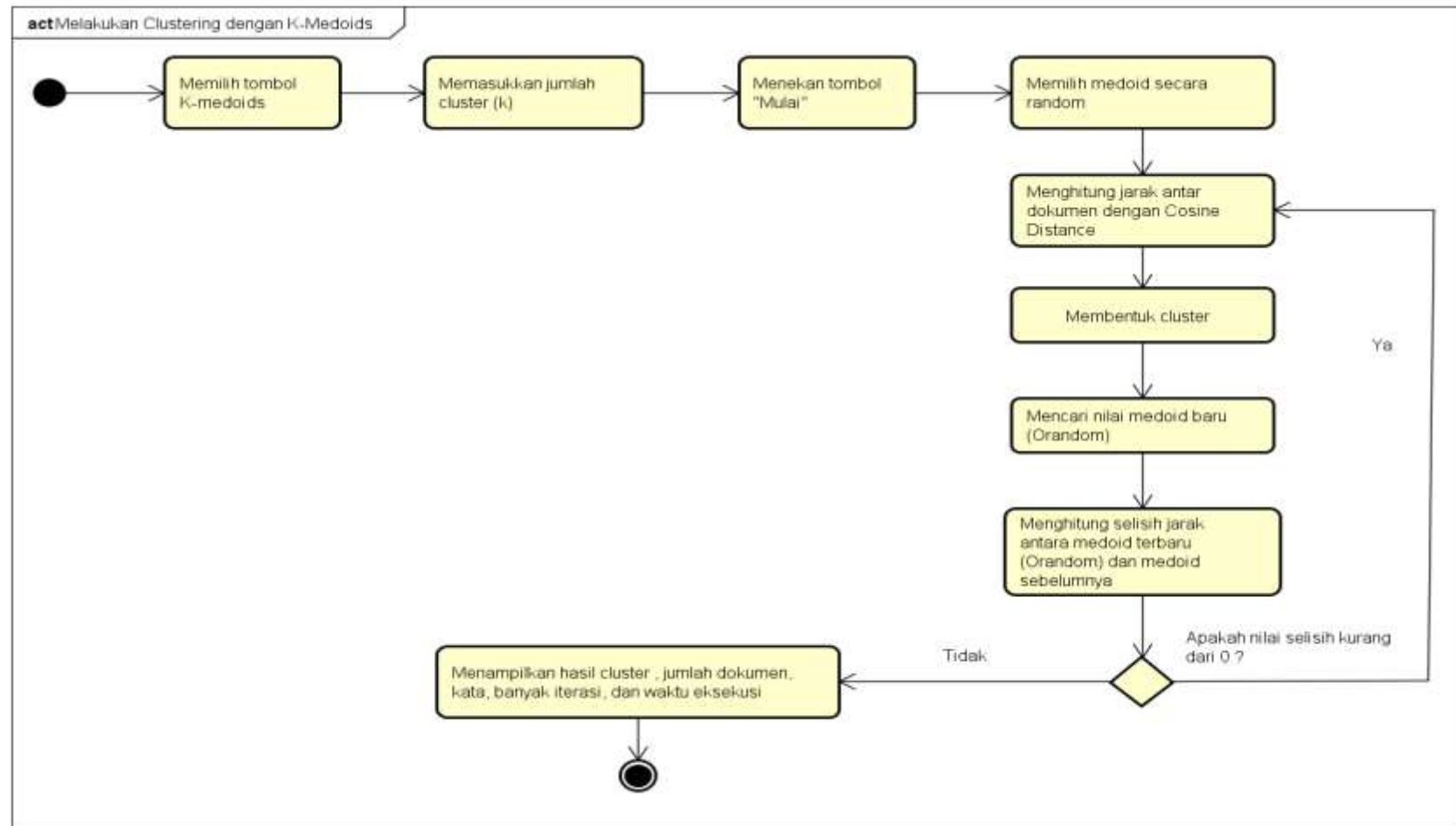
Gambar IV-3. Diagram Aktivitas Use Case Memasukkan Dokumen



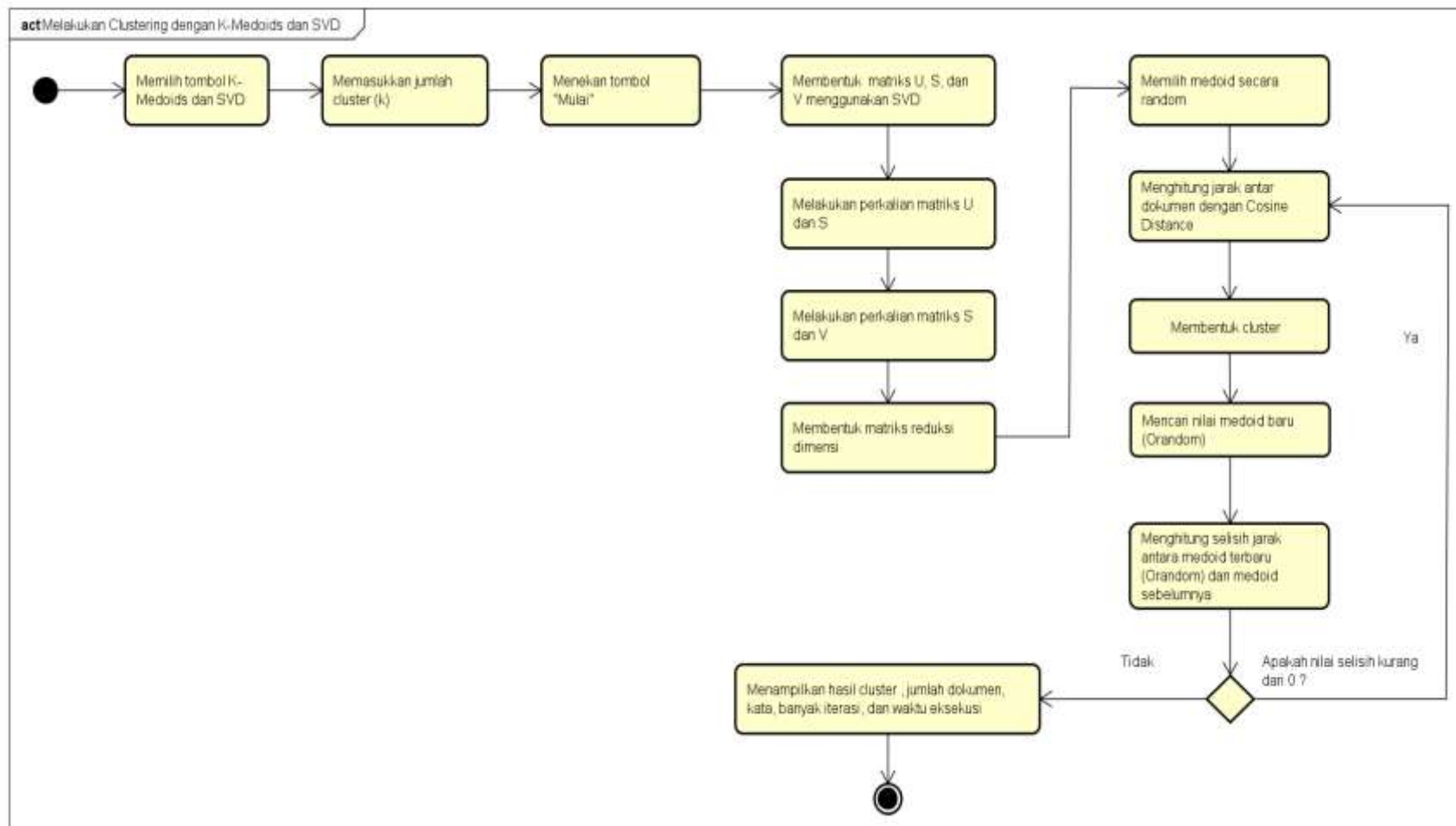
Gambar IV-4. Diagram Aktivitas Use Case Melakukan Clustering dengan K-means



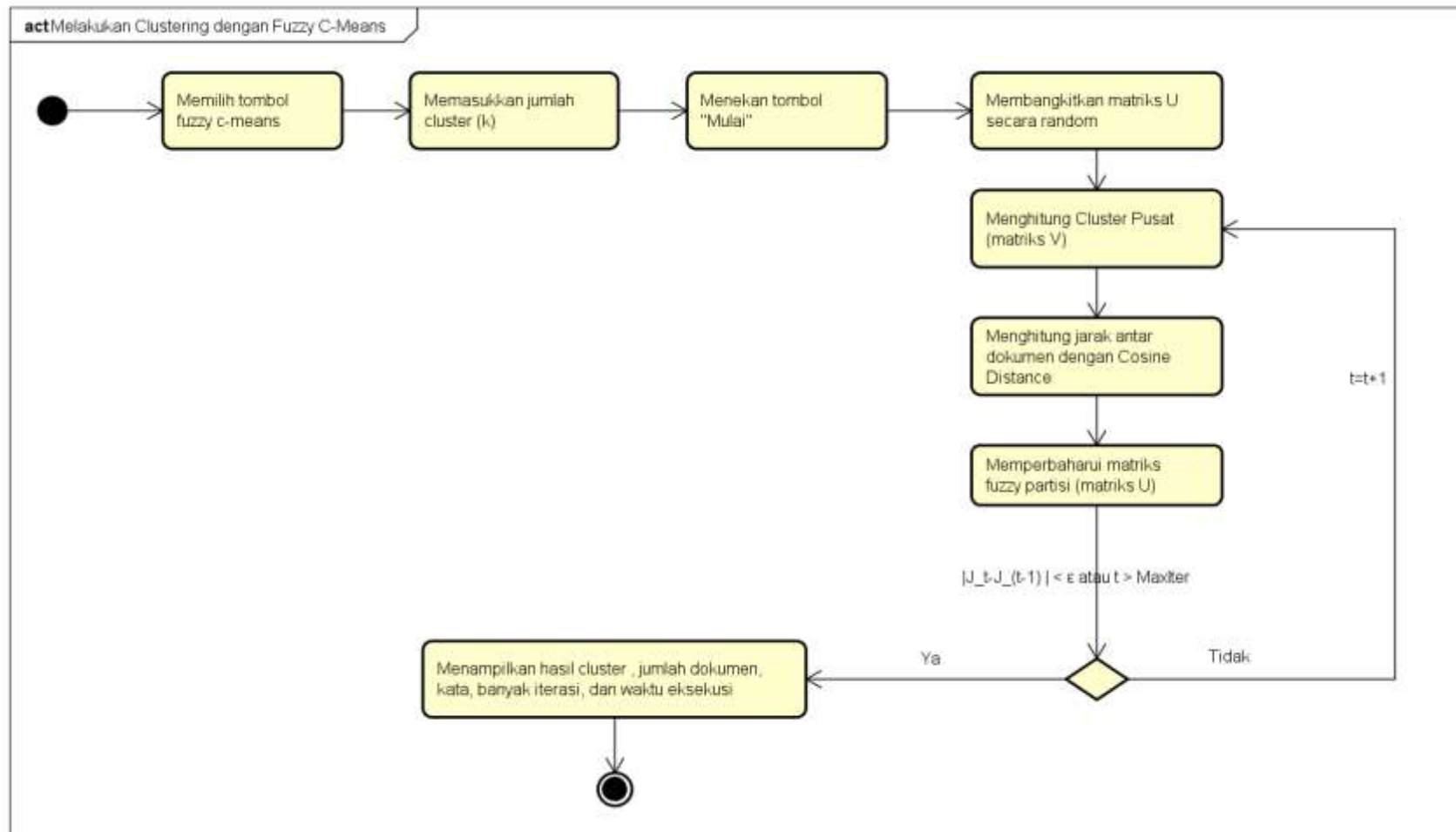
Gambar IV-5. Diagram Aktivitas Use Case Melakukan Clustering dengan K-means dan SVD



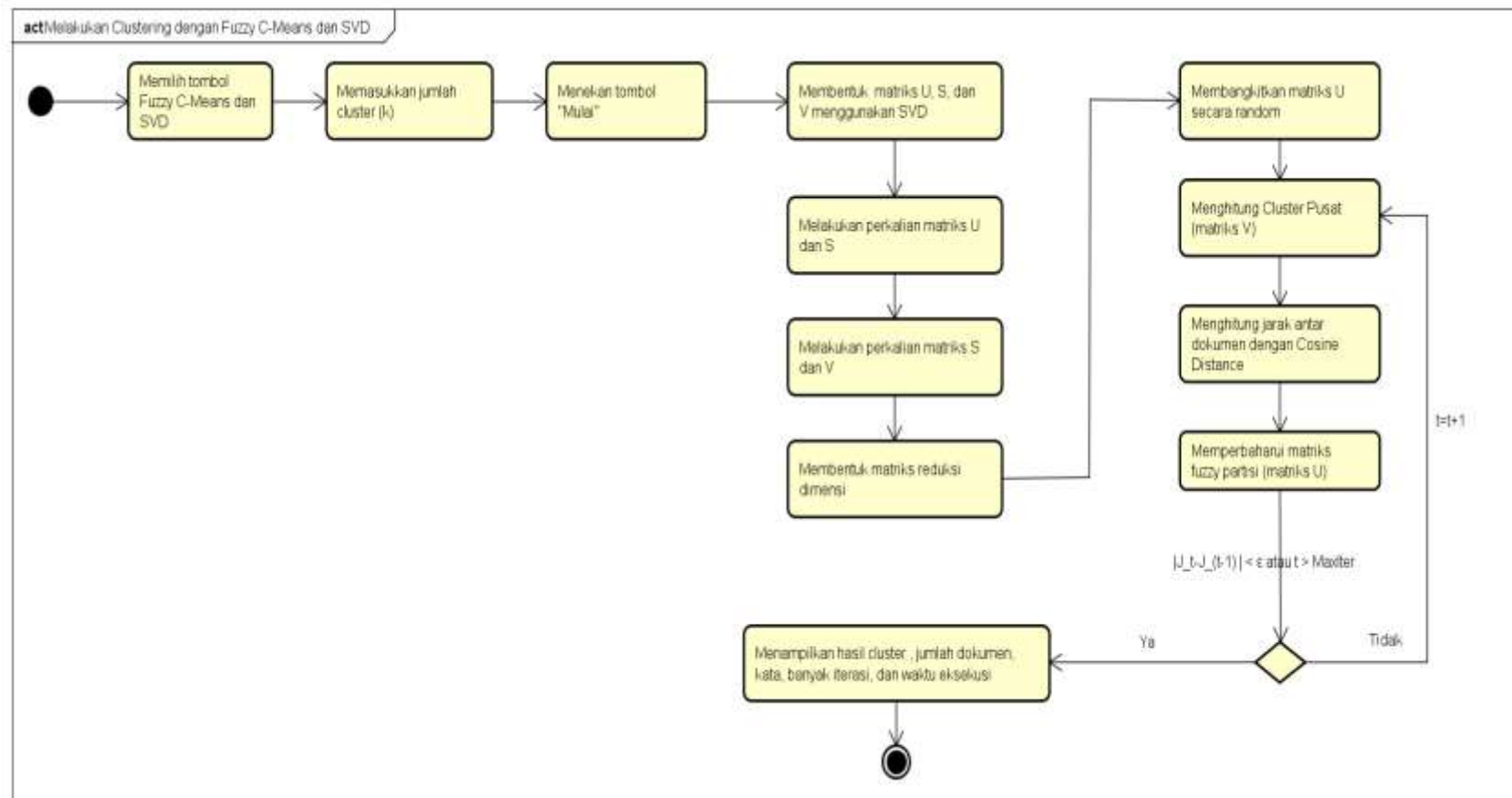
Gambar IV-6. Diagram Aktivitas Use Case Melakukan Clustering dengan K-medoids



Gambar IV-7. Diagram Aktivitas Use Case Melakukan Clustering dengan K-medoids dan SVD



Gambar IV-8. Diagram Aktivitas Use Case Melakukan Clustering dengan Fuzzy C-Means



Gambar IV-9. Diagram Aktivitas Use Case Melakukan Clustering dengan Fuzzy C-Means dan SVD

4.3 Fase Elaborasi

Tahapan kedua dalam pengembangan perangkat lunak adalah melakukan identifikasi terhadap sistem yang dikembangkan. Aktivitas yang dilakukan mencakup perancangan data, perancangan antarmuka, identifikasi kebutuhan, perumusan kebutuhan pengujian, pemodelan diagram sequence, dan pembuatan dokumentasi.

4.3.1 Pemodelan Bisnis

Pada subbab ini akan dibahas mengenai perancangan perangkat lunak yang akan dibangun. Perancangan dilakukan berdasarkan hasil analisa yang telah dilakukan pada fase insepisi. Perancangan yang dibahas pada subbab ini meliputi perancangan data dan perancangan antar muka.

4.3.1.1 Perancangan Data

Perangkat lunak yang akan dibangun memiliki kemampuan *clustering* terhadap data. Adapun data yang akan melalui proses *clustering* adalah data dokumen jurnal yang disimpan dalam file berformat .txt. selanjutnya terdapat kamus besar bahasa Indonesia dan kamus *stopword* yang disimpan juga dalam file berformat .txt. rancangan data yang digunakan adalah sebagai berikut:

1. kamusID.txt : kamus besar bahasa Indonesia;
2. stoplist.txt : berisi kata yang berperan sebagai *stopword*.

4.3.1.2 Perancangan Antar Muka

Pada subbab ini membahas tentang perancangan antar muka dari perangkat lunak yang dibangun. Adapun rancangan antarmuka Menu Utama digambarkan pada gambar IV-10 dan rancangan antarmuka pusat cluster pada gambar IV-11.

Program TA Navita

Muat

Cluster Method

☐ K - Means
☐ K - Medoids
☐ Fuzzy C - Means
☐ SVD + K - Means
☐ SVD + K - Medoids

Jumlah Cluster (k):

Mulai

Cluster Output

Jumlah Kata

Jumlah Dokumen

Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5

Iterasi

Waktu Eksekusi

Gambar IV-10. Rancangan Antarmuka Menu Utama

Gambar IV-10 merupakan halaman utama yang berfungsi untuk melakukan *clustering* jurnal. Terdapat enam bagian pada antarmuka ini yaitu tombol memasukkan, *radio button* untuk memilih metode clustering, text field untuk memasukkan jumlah cluster yang ingin dibentuk, *text field* untuk menampilkan direktori dokumen, *text field* yang menampilkan jumlah kata, jumlah dokumen, dan tabel hasil clustering, text field waktu eksekusi dan iterasi.

3. RAM 4 GB ;

4. *Hard Disk* 320 GB .

Sedangkan perangkat lunak yang digunakan untuk implementasi yaitu:

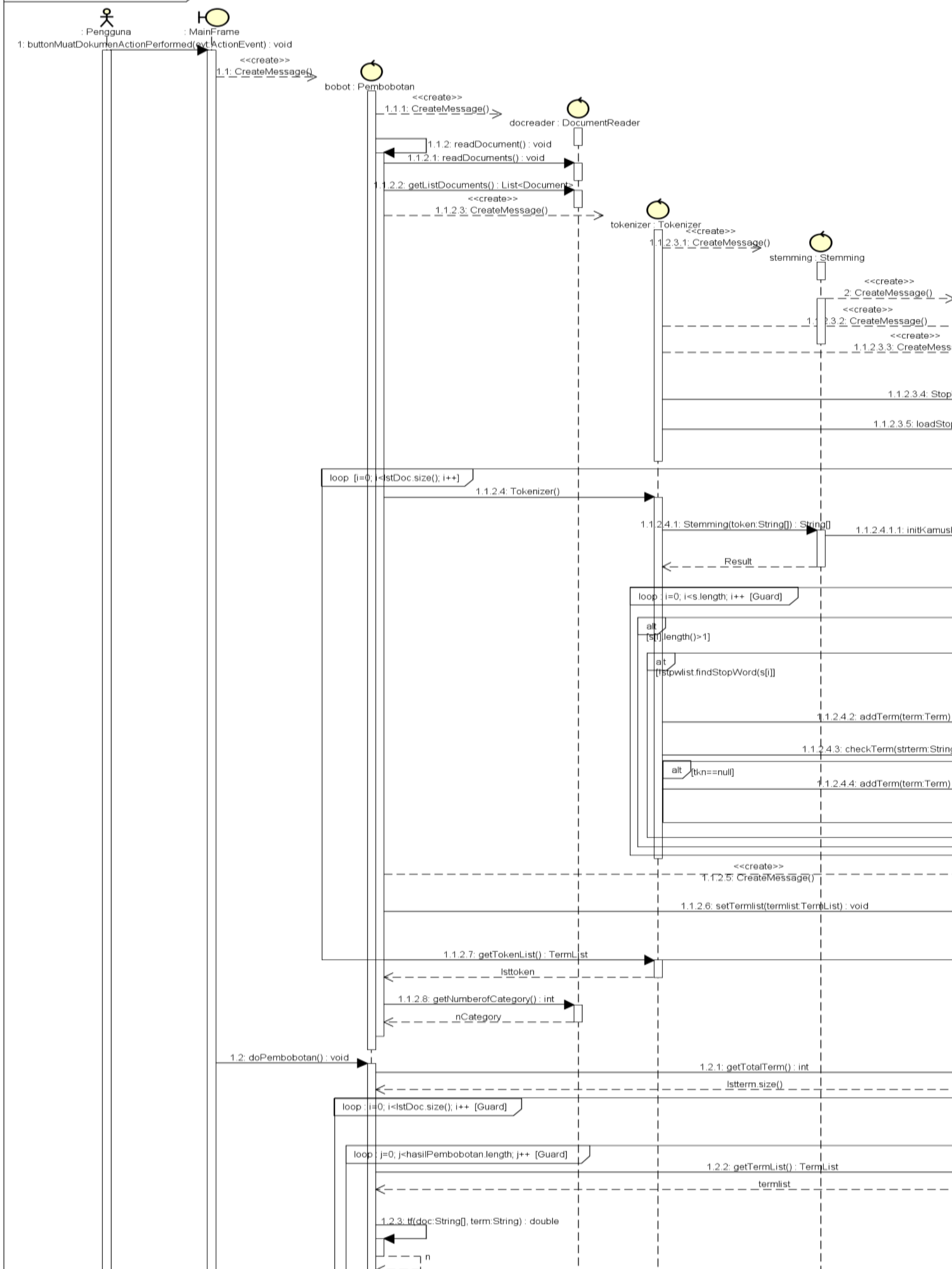
1. Sistem operasi Windows 10 Pro 64-bit ;

2. *Compiler* Netbeans IDE 8.2 .

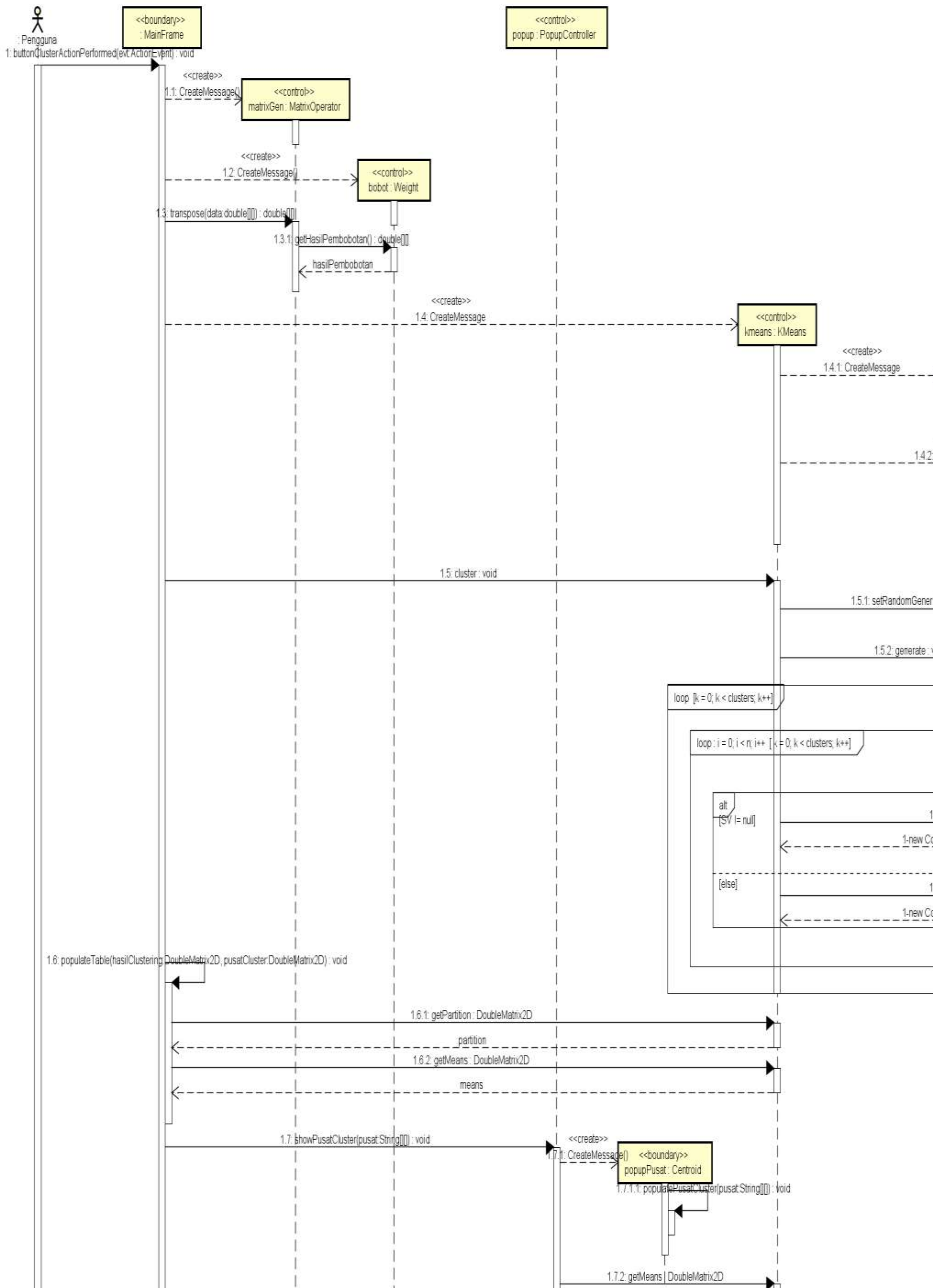
4.3.3 Diagram Sequence

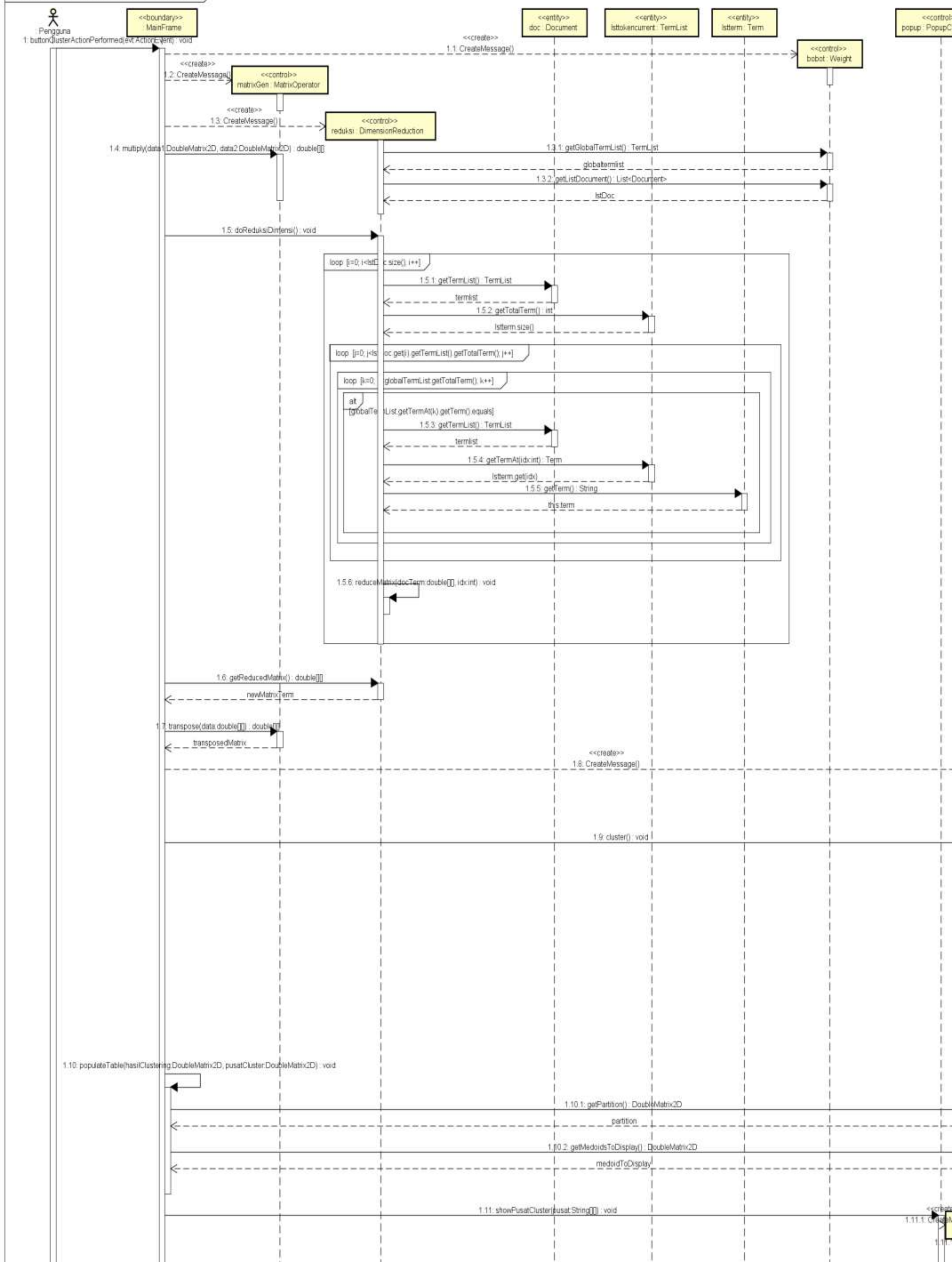
Diagram *sequence* adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah objek. Berdasarkan *use case* yang dibentuk, diagram sequence yang dibentuk pada pengembangan perangkat lunak ini berjumlah tujuh buah. Diagram sequence memasukkan dokumen dapat dilihat pada gambar IV-12, diagram sequence melakukan clustering dengan k-means dapat dilihat pada gambar IV-13, diagram sequence melakukan clustering dengan k-means dan SVD dapat dilihat pada gambar IV-14, diagram sequence melakukan clustering dengan k-medoids dapat dilihat pada gambar IV-15, diagram sequence melakukan clustering dengan k-medoids dan SVD dapat dilihat pada gambar IV-16, diagram sequence melakukan clustering dengan fuzzy c-means dapat dilihat pada gambar IV-17, dan diagram sequence melakukan clustering dengan fuzzy c-means dan SVD pada gambar IV-18 .

sd Sequence Memasukkan Dokumen

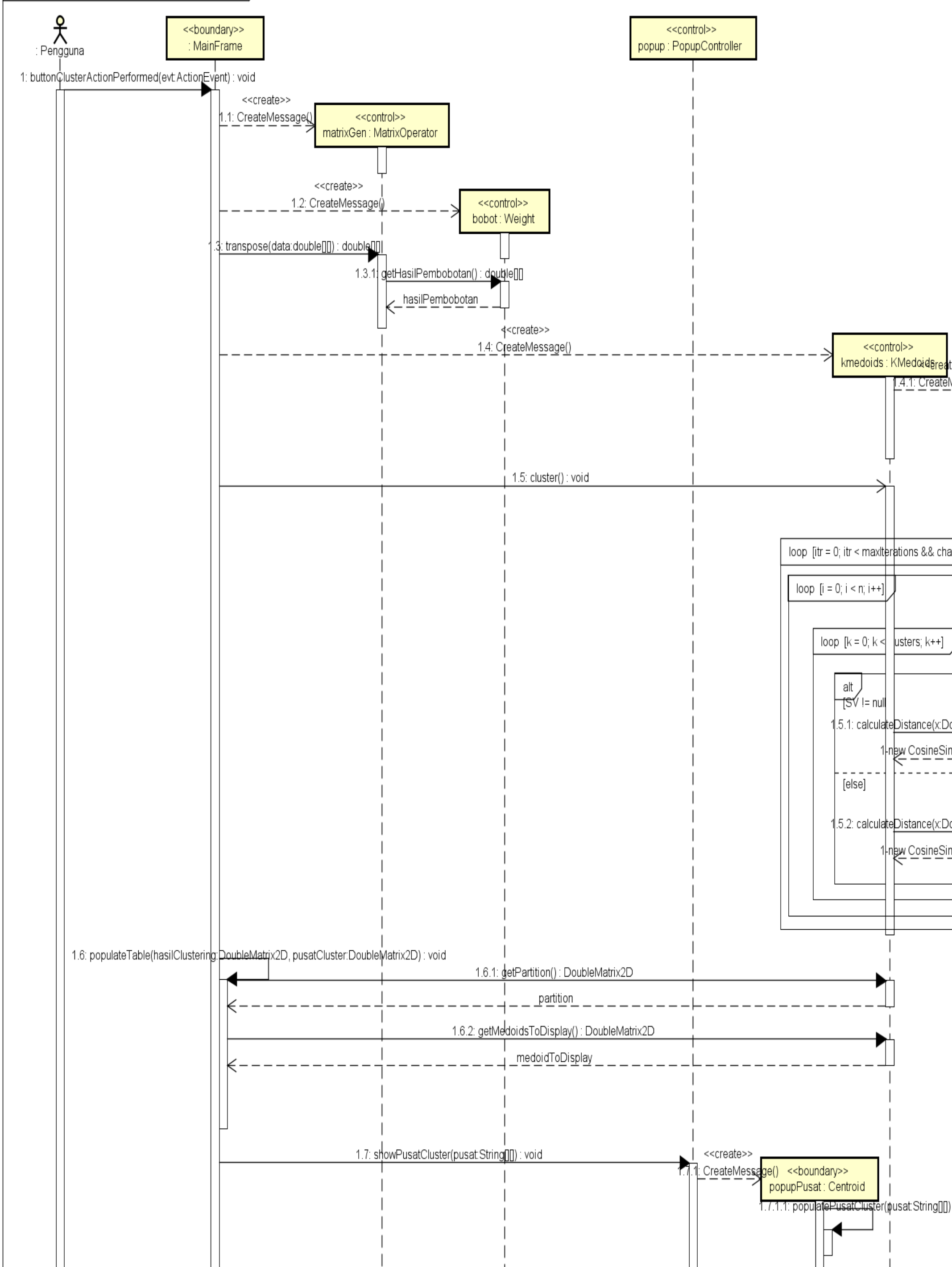


sd Sequence Diagram Clustering dengan K-Means

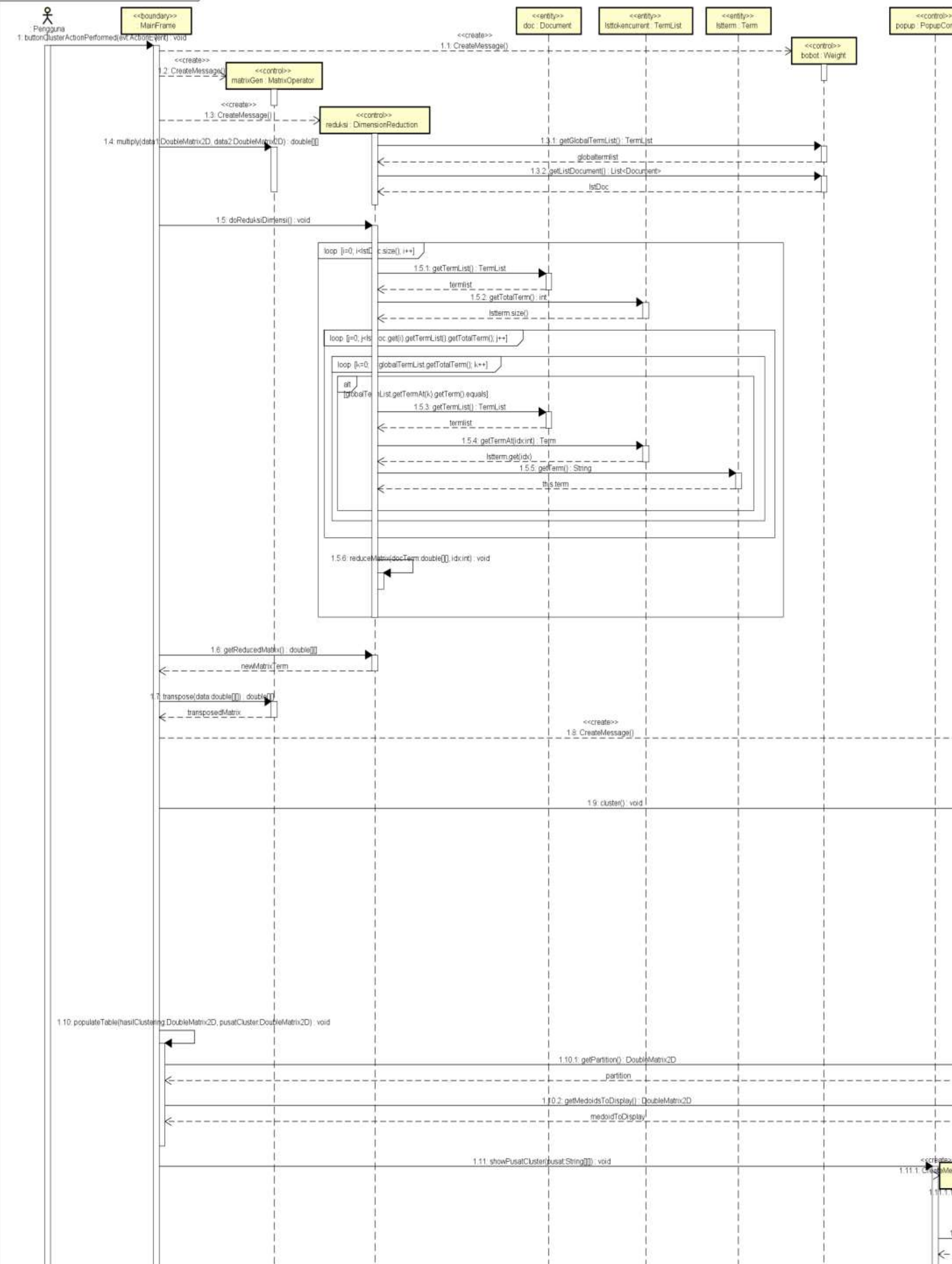




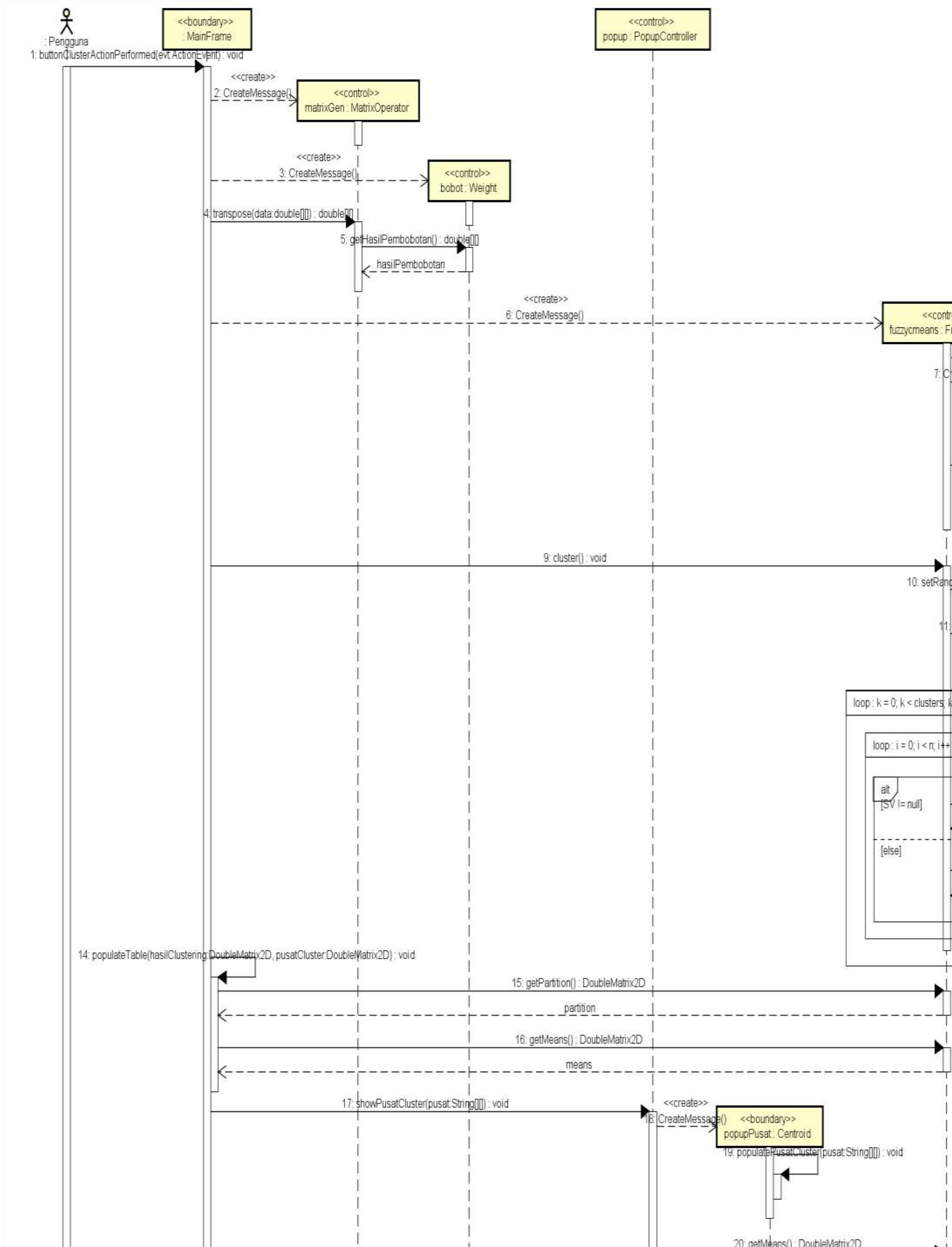
sd Sequence Diagram Clustering dengan K-Medoids



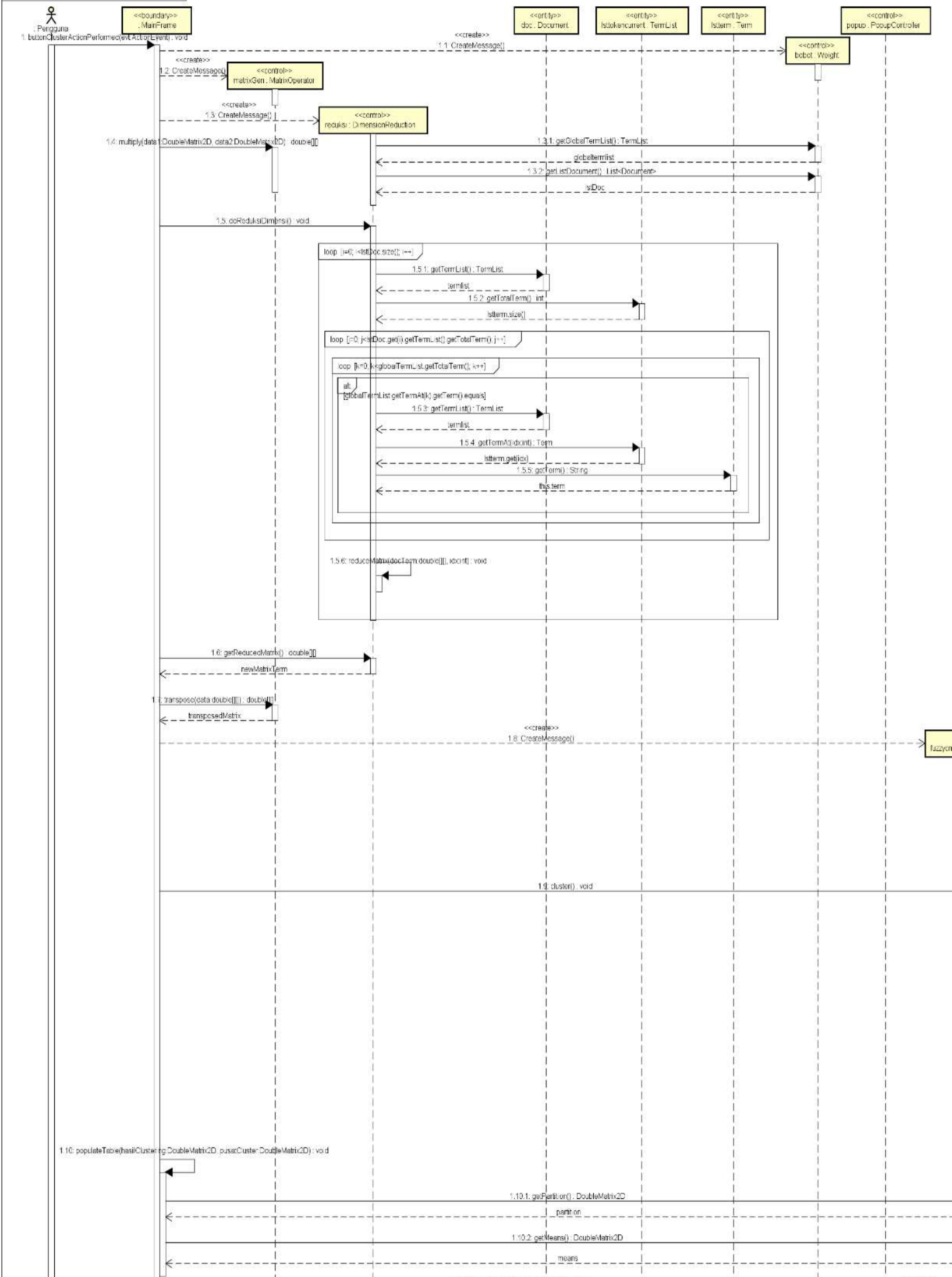
sd Sequence Diagram Clustering dengan K-Medoids & SVD



sd Sequence Diagram Clustering dengan Fuzzy C-Means



sd Sequence Diagram: Clustering Fuzzy C-Means & SVD



4.4 Fase Konstruksi

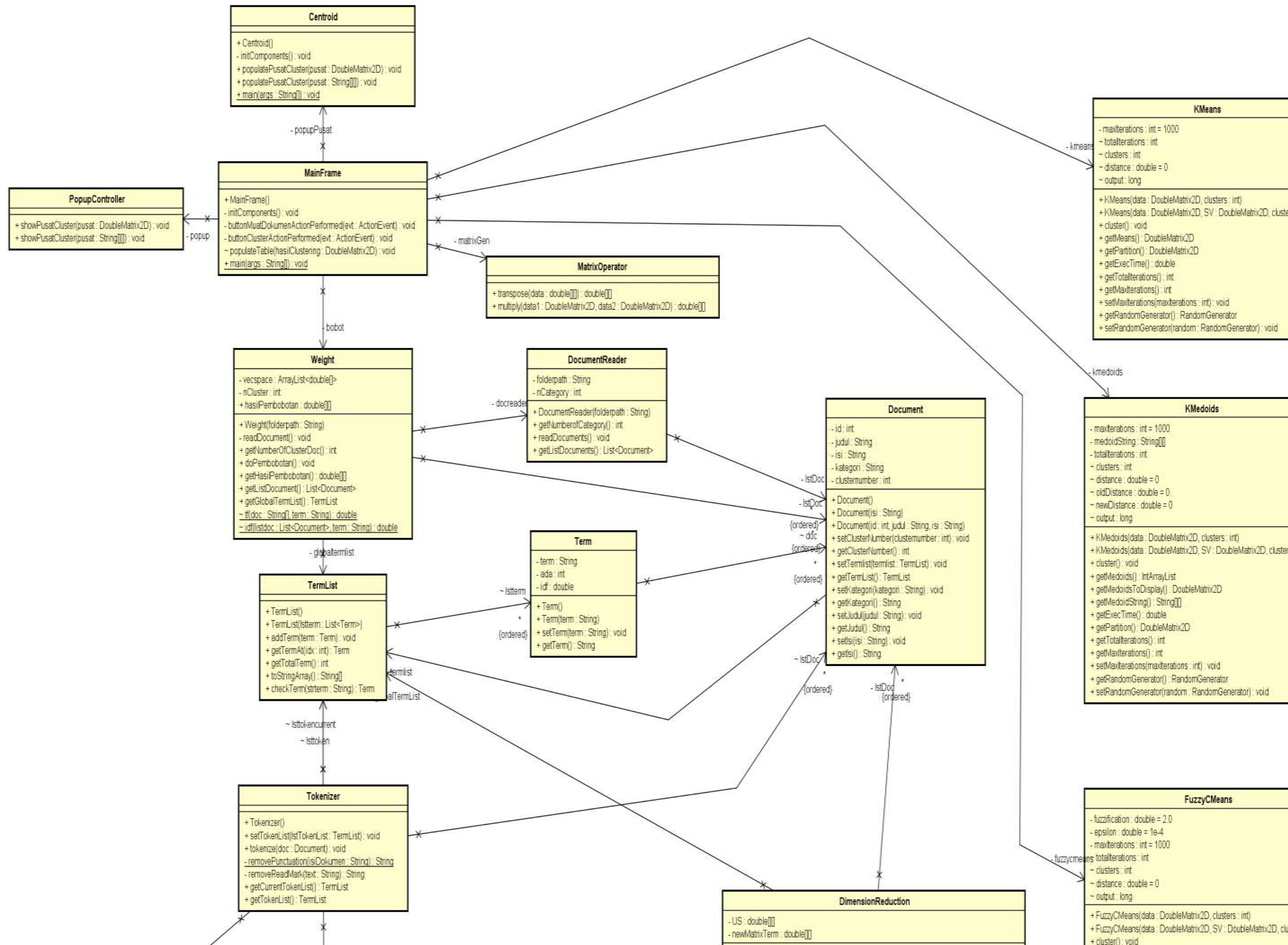
Fase konstruksi berfokus pada pengembangan perangkat lunak baik komponen utama maupun fitur-fitur pendukung dengan melakukan sederet iterasi. Disetiap iterasi terdapat proses analisa, desain, implementasi, dan pengujian. Dalam proses pengembangannya dapat menggunakan konstruksi paralel agar mempercepat hasil perangkat lunak. Hasil yang diharapkan dari fase ini adalah sebuah produk perangkat lunak yang siap digunakan oleh end-user, yaitu sebuah produk perangkat lunak yang dapat digunakan sebagai alat penelitian.

4.4.1 Kebutuhan Sistem

Dalam pengembangan perangkat lunak ini, penulis menggunakan beberapa *library*. Diantaranya adalah Colt, JavaML, dan Commons Math. Library Colt digunakan untuk penerapan perhitungan matematis *cosine distance*, metode clustering k-means, k-medoids, dan fuzzy c-means menggunakan Commons Math. Sedangkan untuk mendapatkan nilai matriks U, S, dan V dari *Singular Value Decomposition* menggunakan library JavaML.

4.4.2 Diagram Kelas

Diagram kelas adalah diagram UML yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain. Terdapat serta dimasukkan pula atribut dan operasi. Terdapat 21 kelas yang terdiri dari 2 kelas entitas (kelas Document dan Term), 2 kelas *boundary* (kelas MainFrame dan PusatCluster), dan 17 kelas *control*. Di gambar IV-19 menunjukkan hubungan antar kelas tersebut.



4.4.3 Implementasi

Fase implementasi dalam konstruksi adalah mengembangkan perangkat lunak berdasarkan diagram kelas dan rancangan antarmuka yang telah dibuat dalam fase sebelumnya.

4.4.3.1 Implementasi Kelas

Kelas-kelas yang telah dirancang pada diagram kelas diimplementasikan dalam bahasa pemrograman Java. Tabel (IV-22) menunjukkan implementasi kelas dalam bahasa Java.

Tabel IV-22 Implementasi Kelas

No	Nama Kelas	Nama File	Keterangan
1	MainFrame	MainFrame.java	Kelas MainFrame merupakan kelas <i>boundary</i> yang menyediakan akses untuk melakukan memasukkan dokumen dan clustering.
2	Centroid	Centroid.java	Kelas Centroid merupakan kelas <i>boundary</i> yang menampilkan hasil clustering dokumen.

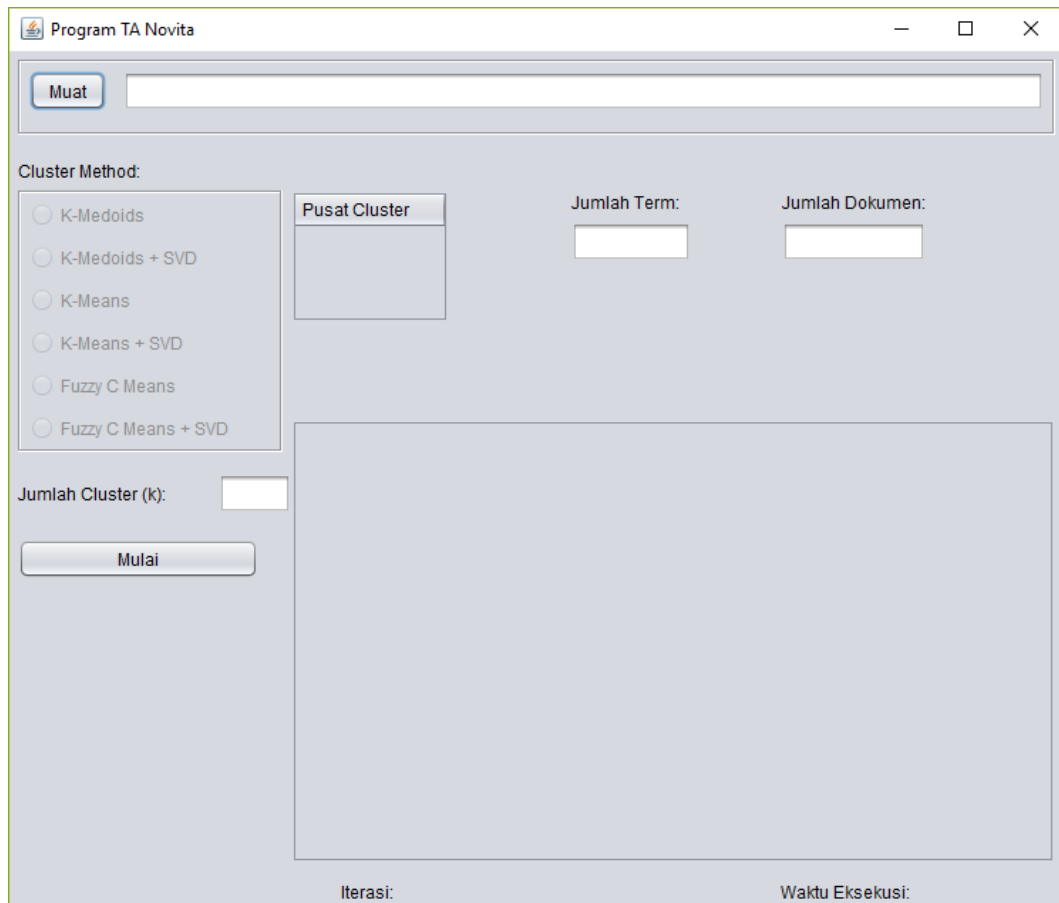
No	Nama Kelas	Nama File	Keterangan
3	PopupController	PopupController.java	Kelas FrameHasilClustering merupakan kelas <i>control</i> yang mengatur kelas PusatCluster.
4	Document	Document.java	Kelas Document merupakan kelas <i>entitas</i> yang bertujuan untuk menyimpan data dokumen.
5	DocumentReader	DocumentReader.java	Kelas DocumentReader merupakan kelas <i>control</i> yang menangani proses dokumen.
6	Stemmer	Stemmer.java	Kelas Stemmer merupakan kelas <i>control</i> yang menangani proses stemming dari teks dokumen.
7	DictGenerator	DictGenerator.java	Kelas DictGenerator merupakan kelas <i>control</i> yang bertujuan untuk menyimpan data kamus kata dasar yang akan digunakan untuk prapengolahan
8	StopWordList	StopWordList.java	Kelas StopWordList merupakan kelas <i>control</i> yang menangani proses filtering dari teks dokumen.
9	Term	Term.java	Kelas Term merupakan kelas <i>entitas</i> yang

			menyimpan kata (term) sebuah dokumen.
10	TermList	TermList.java	Kelas TermList merupakan kelas <i>control</i> untuk menyimpan kumpulan kata (term) seluruh dokumen.
11	Tokenizer	Tokenizer.java	Kelas Tokenizer merupakan kelas <i>control</i> yang menangani proses tokenizing dari teks dokumen.
12	Weight	Weight.java	Kelas Weight merupakan kelas <i>control</i> yang menangani proses pembobotan dari teks dokumen.
13	CosineDistance	CosineDistance.java	Kelas CosineDistance merupakan kelas <i>control</i> yang menangani proses perhitungan cosine distance dari teks dokumen.
14	FuzzyCMeans	FuzzyCMeans.java	Kelas FuzzyCMeans merupakan kelas <i>control</i> yang menangani proses clustering fuzzy c-means.
15	KMeans	Kmeans.java	Kelas Kmeans merupakan kelas <i>control</i> yang menangani proses clustering k-means.
16	KMedoids	Kmedoids.java	Kelas Kmedoids merupakan

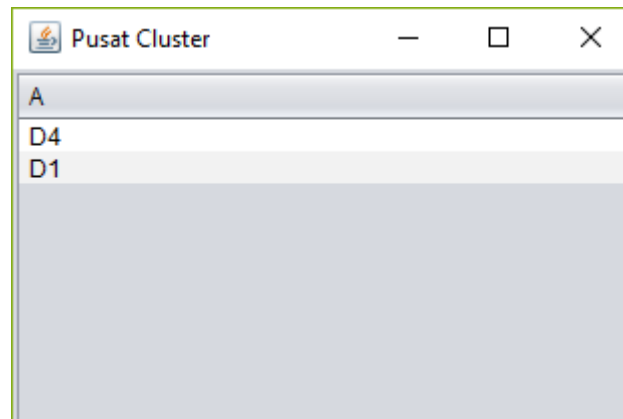
			kelas <i>control</i> yang menangani proses clustering k-medoids.
17	FuzzyRandom PartitionGenerator	FuzzyRandom PartitionGenerator. java	Kelas FuzzyRandom PartitionGenerator merupakan kelas <i>control</i> yang menghasilkan nilai random untuk matriks U.
18	HardRandom PartitionGenerator	HardRandom PartitionGenerator. java	Kelas HardRandom PartitionGenerator merupakan kelas <i>control</i> yang menghasilkan nilai random untuk medoid awal.
19	PartitionGenerator	PartitionGenerator. java	Kelas PartitionGenerator merupakan <i>interface</i> yang memiliki method yang diimplementasikan oleh kelas FuzzyRandom PartitionGenerator dan HardRandom PartitionGenerator
20	MatrixOperator	MatrixOperator.java	Kelas MatrixOperator merupakan kelas <i>control</i> yang menangani perhitungan perkalian dan transpose matriks.
21	DimensionReduction	DimensionReduction.java	Kelas DimensionReduction merupakan kelas <i>control</i> yang menangani reduksi dimensi.

4.4.3.2 Implementasi Antarmuka

Implementasi antarmuka dilakukan berdasarkan perancangan antarmuka pada fase elaborasi. Gambar IV-19 merupakan antarmuka halaman utama perangkat lunak dan gambar IV-20 merupakan antarmuka pusat cluster.



Gambar IV-19 Antarmuka Halaman Utama Perangkat Lunak



Gambar IV-20 Antarmuka Pusat Cluster

4.5 Fase Transisi

Pada subbab ini dibahas mengenai pengujian dari perangkat lunak *clustering* dokumen berbahasa Indonesia yang telah dibangun. Pengujian dilakukan berdasarkan perangkat lunak hasil pengembangan di fase konstruksi.

4.5.1 Pemodelan Bisnis

Pengujian perangkat lunak secara *black box* dan *white box* dengan terlebih dahulu membuat rencana pengujian berdasarkan *Use Case* yang dibuat pada fase insepasi.

4.5.2 Kebutuhan Sistem

Lingkungan pengujian yang digunakan pada fase transisi adalah perangkat keras yang sama saat membangun perangkat lunak *clustering* dokumen berbahasa Indonesia dengan spesifikasi sebagai berikut:

1. Laptop merk Lenovo;
2. Processor Intel(R) Core(TM) i3-5005U CPU @ 2.00 GHz 2.00 GHz;
3. RAM 4 GB ;

4. *Hard Disk* 320 GB .

Sedangkan perangkat lunak yang digunakan untuk pengujian yaitu:

1. Sistem operasi Windows 10 Pro 64-bit ;
2. *Compiler* Netbeans IDE 8.2.

4.5.3 Rencana Pengujian

Rencana pengujian pada perangkat lunak *clustering* dokumen berbahasa Indonesia digambarkan dalam tabel-tabel. Kolom pada tabel meliputi identifikasi, pengujian, jenis pengujian, serta tingkat pengujian.

4.5.3.1 Rencana Pengujian Use Case Memasukkan Dokumen

Tabel IV-23 menerangkan rencana pengujian memasukkan dokumen perangkat lunak berdasarkan *Use Case*.

Tabel IV-23 . Rencana Pengujian *Use Case* Memasukkan Dokumen

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-1-101	Masukkan folder yang berisi file dokumen berekstensi <i>.txt</i> .	<i>Black Box</i>	Pengujian Unit
2.	U-1-102	Masukkan folder yang berisi file dokumen bukan berekstensi <i>.txt</i> .	<i>Black Box</i>	Pengujian Unit
3.	U-1-103	Melakukan keseluruhan prapengolahan dokumen.	<i>White Box</i>	Pengujian Unit

4.5.3.2 Rencana Pengujian Use Case Melakukan Clustering dengan K - Means

Tabel IV-24 menerangkan rencana pengujian melakukan clustering dengan k-means pada perangkat lunak berdasarkan Use Case.

Tabel IV-24 . Rencana Pengujian Use Case Melakukan Clustering dengan

K - Means

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-2-101	Menekan radio button “k-means”.	<i>Black Box</i>	Pengujian Unit
2.	U-2-102	Memasukkan nilai pada kolom jumlah cluster.	<i>Black Box</i>	Pengujian Unit
3.	U-2-103	Menekan tombol “Mulai”	<i>Black Box</i>	Pengujian Unit
4.	U-2-104	Melakukan perhitungan Clustering dokumen dengan metode k-means.	<i>White Box</i>	Pengujian Unit

4.5.3.3 Rencana Pengujian Use Case Melakukan Clustering dengan K - Medoids

Tabel IV-25 menerangkan rencana pengujian melakukan clustering dengan k-medoids pada perangkat lunak berdasarkan Use Case.

Tabel IV-25 . Rencana Pengujian Use Case Melakukan Clustering dengan

K - Medoids

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-3-101	Menekan radio button “k-medoids”.	<i>Black Box</i>	Pengujian Unit
2.	U-3-102	Memasukkan nilai pada kolom jumlah cluster.	<i>Black Box</i>	Pengujian Unit
3.	U-3-103	Menekan tombol “Mulai”	<i>Black Box</i>	Pengujian Unit
4.	U-3-104	Melakukan perhitungan Clustering dokumen dengan metode k-medoids.	<i>White Box</i>	Pengujian Unit

4.5.3.4 Rencana Pengujian Use Case Melakukan Clustering dengan Fuzzy C-Means

Tabel IV-26 menerangkan rencana pengujian melakukan clustering dengan fuzzy c-means pada perangkat lunak berdasarkan Use Case.

Tabel IV-26 . Rencana Pengujian Use Case Melakukan Clustering dengan

Fuzzy C-Means

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-4-101	Menekan radio button “Fuzzy c-means”.	<i>Black Box</i>	Pengujian Unit
2.	U-4-102	Memasukkan nilai	<i>Black Box</i>	Pengujian Unit

		pada kolom jumlah cluster.		
3.	U-4-103	Menekan tombol “Mulai”	<i>Black Box</i>	Pengujian Unit
4.	U-4-104	Melakukan perhitungan Clustering dokumen dengan metode fuzzy c-means.	<i>White Box</i>	Pengujian Unit

4.5.3.5 Rencana Pengujian Use Case Melakukan Clustering dengan K-means dan SVD

Tabel IV-27 menerangkan rencana pengujian melakukan clustering dengan k-means dan SVD pada perangkat lunak berdasarkan Use Case.

Tabel IV-27 Rencana Pengujian Use Case Melakukan Clustering dengan

K– Means dan SVD

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-5-101	Menekan radio button “K-means + SVD”.	<i>Black Box</i>	Pengujian Unit
2.	U-5-102	Memasukkan nilai pada kolom jumlah cluster.	<i>Black Box</i>	Pengujian Unit
3.	U-5-103	Menekan tombol “Mulai”	<i>Black Box</i>	Pengujian Unit

4.	U-5-104	Melakukan perhitungan Clustering dokumen dengan metode k-means dan SVD	<i>White Box</i>	Pengujian Unit
----	---------	--	------------------	----------------

4.5.3.6 Rencana Pengujian Use Case Melakukan Clustering dengan K-medoids dan SVD

Tabel IV-28 menerangkan rencana pengujian melakukan clustering dengan k-medoids dan SVD pada perangkat lunak berdasarkan Use Case.

Tabel IV-28 Rencana Pengujian Use Case Melakukan Clustering dengan K– Medoids dan SVD

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-6-101	Menekan radio button “K-medoids + SVD”.	<i>Black Box</i>	Pengujian Unit
2.	U-6-102	Memasukkan nilai pada kolom jumlah cluster.	<i>Black Box</i>	Pengujian Unit
3.	U-6-103	Menekan tombol “Mulai”	<i>Black Box</i>	Pengujian Unit
4.	U-6-104	Melakukan perhitungan Clustering dokumen dengan metode k-medoids dan SVD	<i>White Box</i>	Pengujian Unit

4.5.3.7 Rencana Pengujian Use Case Melakukan Clustering dengan Fuzzy C - Means dan SVD

Tabel IV-29 menerangkan rencana pengujian melakukan clustering dengan fuzzy c-means dan SVD pada perangkat lunak berdasarkan Use Case.

Tabel IV-29 Rencana Pengujian Use Case Melakukan Clustering dengan Fuzzy c-Means dan SVD

No	Identifikasi	Pengujian	Jenis Pengujian	Tingkat Pengujian
1.	U-7-101	Menekan radio button “Fuzzy c-Means + SVD”.	<i>Black Box</i>	Pengujian Unit
2.	U-7-102	Memasukkan nilai pada kolom jumlah cluster.	<i>Black Box</i>	Pengujian Unit
3.	U-7-103	Menekan tombol “Mulai”	<i>Black Box</i>	Pengujian Unit
4.	U-7-104	Melakukan perhitungan Clustering dokumen dengan metode fuzzy c-means dan SVD	<i>White Box</i>	Pengujian Unit

4.5.4 Implementasi

Berikut ini adalah kasus uji yang dilakukan terhadap perangkat lunak yang dibangun. Kasus uji dilakukan berdasarkan rencana uji yang telah dipaparkan sebelumnya.

4.5.4.1 Pengujian Use Case Memasukkan Dokumen

Tabel IV-30 menerangkan pengujian memasukkan dokumen pada perangkat lunak berdasarkan Use Case.

Tabel IV-30. Pengujian Use Case Memasukkan Dokumen

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
U-1-101	Masukkan folder yang berisi file dokumen berekstensi .txt	Menekan tombol “Muat”.	Tidak ada	List nama dokumen dan isi file .txt.	List nama dokumen dan isi file .txt.	Terpenuhi
U-1-102	Masukkan folder yang berisi file dokumen bukan berekstensi .txt.	Menekan tombol “Muat”	Tidak ada	Dokumen tidak berhasil dimuat. Kemudian, Sistem menampilkan pesan “Gagal. File yang dimasukkan harus berekstensi .txt”.	Dokumen tidak berhasil dimuat. Kemudian, Sistem menampilkan pesan “Gagal. File yang dimasukkan harus berekstensi .txt”.	Terpenuhi
U-1-103	Melakukan keseluruhan	Menekan tombol “prapengolahan”	Masukan berupa	Matriks Hasil Pembobotan	Matriks Hasil Pembobotan	Terpenuhi

	prapengolahan dokumen.		dokumen berekstensi .txt.			
--	------------------------	--	---------------------------	--	--	--

4.5.4.2 Pengujian Use Case Melakukan Clustering dengan K-means

Tabel IV-31 menerangkan pengujian clustering dokumen dengan k - means pada perangkat lunak berdasarkan Use Case.

Tabel IV-31 Rencana Pengujian Use Case Melakukan Clustering dengan K-means

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
U-2-101	Menekan radio button “k-means”.	Menekan radio button “k-means”.	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan	Tombol dapat ditekan	Terpenuhi
U-2-102	Memasukkan nilai pada kolom jumlah cluster.	Mengetikkan nilai ke dalam kolom teks	Masukan berupa nilai integer	Nilai dapat ditangkap dan ditampilkan di antarmuka	Nilai dapat ditangkap dan ditampilkan di antarmuka	Terpenuhi
U-2-103	Menekan tombol	Menekan tombol	Trigger untuk	Tombol dapat	Tombol dapat	Terpenuhi

	“Mulai”	“Mulai”	mengaktifkan tombol	ditekan dan mengaktifkan perhitungan clustering	ditekan dan mengaktifkan perhitungan clustering	
U-2-104	Melakukan perhitungan Clustering dokumen dengan metode k-means.	Menekan tombol “Muat”	Masukan berupa hasil prapengolahan	Hasil cluster yang terkelompok sebanyak jumlah masukan cluster oleh pengguna	Hasil cluster yang terkelompok sebanyak jumlah masukan cluster oleh pengguna	Terpenuhi

4.5.4.3 Pengujian Use Case Melakukan Clustering dengan K-medoids

Tabel IV-32 menerangkan pengujian clustering dokumen dengan k-medoids pada perangkat lunak berdasarkan Use Case.

Tabel IV-32 Rencana Pengujian Use Case Melakukan Clustering dengan K-medoids

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
U-3-101	Menekan radio button “k-”	Menekan radio button “k-medoids”.	Trigger untuk mengaktifkan	Tombol dapat ditekan	Tombol dapat ditekan	Terpenuhi

	medoids”.		tombol			
U-3-102	Memasukkan nilai pada kolom jumlah cluster.	Mengetikkan nilai ke dalam kolom teks	Masukan berupa nilai integer	Nilai dapat ditangkap dan ditampilkan di antarmuka	Nilai dapat ditangkap dan ditampilkan di antarmuka	Terpenuhi
U-3-103	Menekan tombol “Mulai”	Menekan tombol “Mulai”	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Terpenuhi
U-3-104	Melakukan perhitungan Clustering dokumen dengan metode k-medoids.	Menekan tombol “Muat”	Masukan berupa hasil prapengolahan	Hasil cluster yang terkelompok sebanyak jumlah masukan cluster oleh pengguna	Hasil cluster yang terkelompok sebanyak jumlah masukan cluster oleh pengguna	Terpenuhi

4.5.4.4 Pengujian Use Case Melakukan Clustering dengan Fuzzy C-Means

Tabel IV-33 menerangkan pengujian clustering dokumen dengan fuzzy c-means pada perangkat lunak berdasarkan Use Case.

Tabel IV-33 Rencana Pengujian Use Case Melakukan Clustering dengan Fuzzy C-Means

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
U-4-101	Menekan radio button “fuzzy c-means”.	Menekan radio button “fuzzy c-means”.	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan	Tombol dapat ditekan	Terpenuhi
U-4-102	Memasukkan nilai pada kolom jumlah cluster.	Mengetikkan nilai ke dalam kolom teks	Masukan berupa nilai integer	Nilai dapat ditangkap dan ditampilkan di antarmuka	Nilai dapat ditangkap dan ditampilkan di antarmuka	Terpenuhi
U-4-103	Menekan tombol “Mulai”	Menekan tombol “Mulai”	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Terpenuhi
U-4-104	Melakukan	Menekan tombol	Masukan	Hasil cluster yang	Hasil cluster yang	Terpenuhi

	perhitungan Clustering dokumen dengan metode fuzzy c-means.	“Muat”	berupa hasil prapengolahan	terkelompok sebanyak jumlah masukan cluster oleh pengguna	terkelompok sebanyak jumlah masukan cluster oleh pengguna	
--	---	--------	----------------------------	---	---	--

4.5.4.5 Pengujian Use Case Melakukan Clustering dengan K-means dan SVD

Tabel IV-34 menerangkan pengujian clustering dokumen dengan k-means dan SVD pada perangkat lunak berdasarkan Use Case.

Tabel IV-34 Rencana Pengujian Use Case Melakukan Clustering dengan k-means dan SVD

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
U-5-101	Menekan radio button “k-means + SVD”.	Menekan radio button “k-means + SVD”.	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan	Tombol dapat ditekan	Terpenuhi
U-5-102	Memasukkan nilai pada kolom	Mengetikkan nilai ke dalam kolom teks	Masukan berupa nilai	Nilai dapat ditangkap dan	Nilai dapat ditangkap dan	Terpenuhi

	jumlah cluster.		integer	ditampilkan di antarmuka	ditampilkan di antarmuka	
U-5-103	Menekan tombol “Mulai”	Menekan tombol “Mulai”	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Terpenuhi
U-5-104	Melakukan perhitungan Clustering dokumen dengan metode k-means dan SVD	Menekan tombol “Muat”	Masukan berupa hasil prapengolahan	Hasil cluster yang terkelompok sebanyak jumlah masukan cluster oleh pengguna	Hasil cluster yang terkelompok sebanyak jumlah masukan cluster oleh pengguna	Terpenuhi

4.5.4.6 Pengujian Use Case Melakukan Clustering dengan K-medoids dan SVD

Tabel IV-35 menerangkan pengujian clustering dokumen dengan k-medoids dan SVD pada perangkat lunak berdasarkan Use Case.

Tabel IV-35 Rencana Pengujian Use Case Melakukan Clustering dengan k-medoids dan SVD

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
U-6-101	Menekan radio button “k-medoids + SVD”.	Menekan radio button “k-medoids + SVD”.	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan	Tombol dapat ditekan	Terpenuhi
U-6-102	Memasukkan nilai pada kolom jumlah cluster.	Mengetikkan nilai ke dalam kolom teks	Masukan berupa nilai integer	Nilai dapat ditangkap dan ditampilkan di antarmuka	Nilai dapat ditangkap dan ditampilkan di antarmuka	Terpenuhi
U-6-103	Menekan tombol “Mulai”	Menekan tombol “Mulai”	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Terpenuhi
U-6-104	Melakukan perhitungan Clustering	Menekan tombol “Muat”	Masukan berupa hasil prapengolaha	Hasil cluster yang terkelompok sebanyak jumlah	Hasil cluster yang terkelompok sebanyak jumlah	Terpenuhi

	dokumen dengan metode k-medoids dan SVD		n	masukan cluster oleh pengguna	masukan cluster oleh pengguna	
--	---	--	---	-------------------------------	-------------------------------	--

4.5.4.7 Pengujian Use Case Melakukan Clustering dengan Fuzzy C-Means dan SVD

Tabel IV-36 menerangkan pengujian clustering dokumen dengan fuzzy c-means dan SVD pada perangkat lunak berdasarkan Use Case.

Tabel IV-36 Rencana Pengujian Use Case Melakukan Clustering dengan fuzzy c-means dan SVD

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Hasil yang Didapat	Kesimpulan
U-7-101	Menekan radio button “fuzzy c-means + SVD”.	Menekan radio button “fuzzy c-means + SVD”.	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan	Tombol dapat ditekan	Terpenuhi
U-7-102	Memasukkan nilai pada kolom jumlah cluster.	Mengetikkan nilai ke dalam kolom teks	Masukan berupa nilai integer	Nilai dapat ditangkap dan ditampilkan di antarmuka	Nilai dapat ditangkap dan ditampilkan di antarmuka	Terpenuhi

U-7-103	Menekan tombol “Mulai”	Menekan tombol “Mulai”	Trigger untuk mengaktifkan tombol	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Tombol dapat ditekan dan mengaktifkan perhitungan clustering	Terpenuhi
U-7-104	Melakukan perhitungan Clustering dokumen dengan metode fuzzy c- means dan SVD	Menekan tombol “Muat”	Masukan berupa hasil prapengolaha n	Hasil cluster yang terkelompok sebanyak jumlah masukan cluster oleh pengguna	Hasil cluster yang terkelompok sebanyak jumlah masukan cluster oleh pengguna	Terpenuhi

Berdasarkan hasil pengujian yang telah dilakukan terhadap kesesuaian perancangan perangkat lunak dan implementasi perangkat lunak maka dapat disimpulkan bahwa unit dan antar muka yang dibangun dapat berjalan dengan baik. Hal ini ditandai dengan kesimpulan hasil skenario pada kasus uji semuanya memberikan kesimpulan yang sama, yaitu terpenuhi

BAB V

ANALISIS PENELITIAN

5.1 Pendahuluan

Pada bab IV telah dilakukan pengembangan perangkat lunak yang menjadi alat penelitian pengaruh teknik reduksi dimensi *Singular Value Decomposition (SVD)* terhadap hasil pengelompokan dokumen berdimensi tinggi menggunakan metode clustering *k-means*, *k-medoids*, dan *fuzzy c-means*. Pada bab V, hasil clustering data uji penelitian menggunakan perangkat lunak digambarkan pada tabel V-1 sampai dengan tabel V-6. Untuk membuktikan pengaruh SVD terhadap metode clustering, maka digunakan nilai *precision*, *recall*, *f-measure*, *rand index* untuk mengevaluasi performa clustering serta waktu komputasi dan jumlah iterasi yang diperlukan untuk menyelesaikan proses clustering. Untuk mendapatkan nilai *precision*, *recall*, *f-measure*, dan *rand index* diperlukan *confussion matrix* yang mengandung nilai True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Untuk mendapatkan nilai TP, TN, FP, dan FN penelitian ini menggunakan bantuan library NLP stanford.

5.2 Hasil Percobaan Penelitian

Pengujian dilakukan dengan menggunakan data uji berupa dokumen jurnal sebanyak 100 buah, yang terdiri dari lima buah topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan. Masing-masing kategori

mempunyai 20 dokumen jurnal. Data diunduh dari situs Indonesian Publication Index (*id.portalgaruda.org*) yang kemudian disalin dan disimpan dalam dokumen bereksistensi *.txt*. Proses pengujian dilakukan sesuai dengan arsitektur perangkat lunak yang telah dijabarkan pada subbab 4.2.2, yakni prapengolahan, reduksi dimensi, dan *clustering* dokumen. Berdasarkan hasil prapengolahan, jumlah kata kunci atau *term* dari keseluruhan data uji penelitian adalah 8088 kata. Percobaan dilakukan sebanyak 10 kali pada masing-masing metode clustering k-means, k-medoids, dan fuzzy c-means dengan pertimbangan bahwa pusat cluster yang diambil secara acak di awal akan mempengaruhi hasil setiap clustering. Jumlah cluster yang dibentuk adalah lima sesuai dengan banyaknya kelas murni jurnal. Pada tabel V-1, V-2, V-3, V-4, V-5, dan V-6 masing-masing huruf ‘E’, ‘H’, ‘K’, ‘PN’, ‘PT’ mewakili inisial topik ekonomi, hukum, komputer, pertanian dan peternakan secara berurutan.

5.2.1 Hasil Clustering K - Means

Tabel V-1 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan. Keterangan judul dokumen lebih lengkap terdapat pada Lampiran Percobaan Clustering K-means (Lampiran III.1). Dapat dilihat pada percobaan ke-5, pada cluster 1 topik jurnal komputer dominan terhadap topik lain dengan jumlah dokumen sebanyak 17. Di cluster 2 topik jurnal hukum dominan dengan jumlah dokumen sebanyak 19. Di cluster 3 terdapat 15 dokumen jurnal ekonomi dan 18 dokumen jurnal pendidikan, namun topik jurnal pendidikan tetap dominan.

Di cluster 4, tidak ada topik jurnal yang dominan, sehingga tidak dapat dipastikan cluster yang terbentuk merupakan cluster topik ekonomi, komputer, pendidikan, ataupun peternakan. Sedangkan pada cluster 5, topik peternakan dominan dengan jumlah dokumen sebanyak 6.

Berbeda dengan percobaan ke-7, setiap cluster cenderung memiliki satu topik jurnal yang dominan. Di cluster 1, topik jurnal ekonomi dominan terhadap topik jurnal lain dengan 15 dokumen. Di cluster 2, topik jurnal peternakan dominan dengan 8 dokumen. Di cluster 3, topik jurnal pendidikan dominan dengan 8 dokumen. Di cluster 3, topik jurnal hukum dominan dengan 19 dokumen dan di cluster 5 topik jurnal komputer dominan dengan 15 dokumen. Sehingga masing-masing cluster dapat dinamakan berdasarkan topik jurnal yang dominan di dalamnya.

Tabel V-1. Hasil Clustering K-means

Percobaan ke - <i>n</i>	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT
1	2	0	1	0	7	1	19	0	0	0	1	0	0	19	2	3	0	17	0	10	13	1	3	0	1
2	2	0	17	0	0	2	0	0	0	12	4	2	2	3	1	11	18	0	15	0	1	0	1	2	7
3	1	0	5	8	0	16	1	1	0	13	2	19	0	0	0	1	0	1	12	7	0	0	13	0	0
4	17	0	2	0	2	1	0	17	0	1	1	0	0	2	11	1	1	0	1	5	0	19	1	17	1
5	0	0	17	0	9	1	19	1	0	2	15	0	0	18	1	2	1	2	2	2	2	0	0	0	6
6	3	0	0	19	2	3	0	1	1	1	10	19	12	0	0	2	1	5	0	5	2	0	2	0	12
7	15	0	0	12	2	1	0	1	0	8	3	0	0	8	2	1	19	1	0	7	1	1	18	0	1
8	0	20	0	1	1	1	0	3	0	6	3	0	0	19	0	4	0	1	0	12	12	0	16	0	1
9	1	18	0	0	5	0	2	12	15	0	16	0	4	0	2	2	0	0	0	13	1	0	4	5	0
10	3	0	1	2	12	14	0	1	2	5	3	0	1	1	3	0	0	16	3	0	0	20	1	16	0

5.2.2 Hasil Clustering K-means dan Singular Value Decomposition

Tabel V-2 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan clustering k-means dan singular value decomposition. Keterangan judul dokumen lebih lengkap terdapat pada Lampiran Percobaan Clustering k-means dan singular value decomposition (Lampiran III.2). Dapat dilihat pada tabel V-2, persebaran topik jurnal lebih merata dibandingkan tanpa reduksi dimensi. Terutama pada percobaan ke-4, hanya topik jurnal komputer yang berjumlah 19 dokumen pada cluster 1. Sedangkan pada cluster 2, 3, 4, dan 5 tiap topik jurnal *tercluster* secara

baik dan benar. Sehingga dapat disimpulkan bahwa teknik reduksi dimensi singular value decomposition memberikan pengaruh terhadap ketepatan persebaran topik jurnal di setiap cluster menggunakan metode k-means.

Tabel V-2. Hasil Clustering K-means dan Singular Value Decomposition

Percobaan ke - <i>n</i>	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT
1	0	0	1	20	6	0	0	0	0	14	0	20	0	0	0	0	0	17	0	0	20	0	2	0	0
2	0	0	1	20	6	0	20	0	0	0	19	0	0	0	0	1	0	0	0	14	0	0	19	0	0
3	0	20	0	0	0	1	0	0	0	14	0	0	18	0	0	19	0	2	0	6	0	0	1	19	0
4	0	0	19	0	0	0	0	0	0	20	0	20	0	0	0	20	0	0	0	0	0	0	1	20	0
5	19	0	1	0	0	0	20	0	0	0	0	0	18	0	0	0	0	0	0	14	1	0	1	20	6
6	0	0	1	20	0	0	0	17	0	0	20	0	0	0	0	0	0	1	0	20	0	20	1	0	0
7	0	20	1	0	0	1	0	0	0	20	0	0	17	0	0	0	0	2	20	0	19	0	0	0	0
8	0	20	1	0	0	0	0	18	0	0	0	0	0	20	0	16	0	0	0	13	4	0	1	0	7
9	0	20	0	0	0	0	0	1	0	14	0	0	0	0	6	0	0	19	20	0	20	0	0	0	0
10	1	20	0	0	0	0	0	15	0	0	0	0	1	20	0	2	0	4	0	1	17	0	0	0	19

5.2.3 Hasil Clustering K-medoids

Tabel V-3 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan clustering k-medoids. Keterangan judul dokumen lebih lengkap terdapat pada Lampiran Percobaan Clustering K-medoids (Lampiran III.3). Persebaran topik jurnal terburuk terdapat pada percobaan ke-10, dikarenakan topik jurnal peternakan pada cluster 3 dan 4 memiliki jumlah dokumen yang dominan dibanding topik lain sebanyak 6 dan 14

dokumen secara berurutan. Di cluster 1, topik jurnal pendidikan sebanyak 20 dokumen. Sedangkan di cluster 2, hanya terdapat 1 dokumen jurnal ekonomi. Di cluster 5, topik jurnal hukum sebanyak 19 dokumen. Berdasarkan percobaan tersebut, dapat disimpulkan bahwa persebaran data uji penelitian menggunakan metode k-medoids tidak merata.

Tabel V-3. Hasil Clustering K-medoids

Percobaan ke - <i>n</i>	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT
1	1	1	0	0	6	0	0	0	0	12	19	4	0	3	0	0	14	0	9	2	0	1	20	8	0
2	1	6	0	11	0	15	14	20	6	0	0	0	0	3	2	0	0	0	0	18	4	0	0	0	0
3	5	0	0	0	7	8	3	1	0	0	0	0	11	20	0	7	12	4	0	0	0	0	4	0	13
4	15	20	0	1	0	0	0	0	0	14	0	0	6	1	0	0	0	14	0	0	5	0	0	18	6
5	10	20	1	0	0	9	0	14	0	8	1	0	5	19	0	0	0	0	0	12	0	0	0	1	0
6	19	0	9	2	0	0	5	8	8	0	1	0	1	0	12	0	0	1	0	8	0	15	1	12	0
7	0	0	19	0	2	3	0	0	0	2	0	20	0	0	1	17	0	0	0	10	0	0	1	20	5
8	14	12	3	0	0	0	0	4	0	2	6	0	2	0	0	0	6	8	20	0	0	2	3	0	18
9	2	20	0	10	2	3	0	0	3	0	15	0	0	2	1	0	0	20	5	0	0	0	0	0	18
10	1	0	0	20	0	1	0	0	0	0	1	1	3	0	6	1	0	9	0	14	16	19	8	0	0

5.2.4 Hasil Clustering K-medoids dan Singular Value Decomposition

Tabel V-4 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan. Keterangan judul dokumen lebih lengkap terdapat pada Lampiran Percobaan Clustering K-medoids dan singular value decomposition (Lampiran III.4). Seperti hasil percobaan clustering menggunakan kombinasi teknik

reduksi dimensi dan metode k-means, persebaran topik jurnal kombinasi teknik reduksi dimensi dan metode k-medoids pun lebih merata dibandingkan tanpa reduksi dimensi. Terutama ditunjukkan pada percobaan ke-9, pada cluster 1 ada 20 dokumen yang bertopik hukum. Di cluster 2 terdapat topik jurnal komputer sebanyak 17 dokumen, di cluster 3 jurnal pendidikan dominan dengan 19 dokumen, di cluster 4 jurnal peternakan dominan dengan 20 dokumen, dan di cluster 5 jurnal ekonomi juga dominan dengan 20 dokumen. Sehingga berdasarkan percobaan di tabel V-4 dapat disimpulkan bahwa teknik reduksi dimensi singular value decomposition memberikan pengaruh terhadap ketepatan persebaran topik jurnal di setiap cluster menggunakan metode k-medoids.

Tabel V-4. Hasil Clustering K-medoids dan Singular Value Decomposition

Percobaan ke - <i>n</i>	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT
1	4	0	0	0	0	0	0	0	0	20	0	20	1	0	0	16	0	4	0	0	0	0	15	20	0
2	9	0	0	0	19	0	0	20	0	0	0	20	0	0	1	11	0	0	0	0	0	0	0	20	0
3	0	0	2	0	0	0	20	0	0	0	2	0	1	20	2	0	0	17	0	1	18	0	0	0	17
4	0	0	4	0	0	0	20	0	15	0	0	0	0	20	20	0	0	5	0	0	0	0	16	0	0
5	10	8	0	0	0	0	0	19	0	0	1	0	0	0	20	0	0	1	20	0	9	12	0	0	0
6	0	0	0	0	16	0	0	0	20	0	20	2	0	0	4	0	0	20	0	0	0	18	0	0	0
7	4	20	0	0	0	16	0	0	0	0	0	0	0	20	0	0	0	20	0	0	0	20	0	0	0
8	0	0	2	5	1	0	0	0	15	0	8	0	0	0	19	0	0	18	0	0	12	20	0	0	0
9	0	20	0	0	0	0	0	17	1	0	0	0	1	19	0	0	0	0	0	20	20	0	2	0	0
10	0	0	3	0	12	3	0	0	0	8	0	0	1	20	0	17	0	16	0	0	0	20	0	0	0

5.2.5 Hasil Clustering Fuzzy C-Means

Tabel V-5 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan. Keterangan judul dokumen lebih lengkap terdapat pada Lampiran Percobaan Clustering Fuzzy C-Means (Lampiran III.5). Kekhasan dari metode fuzzy c-means clustering adalah satu data dapat masuk ke lebih dari satu buah cluster. Namun pada tabel V-5, peneliti hanya menampilkan dokumen yang nilai kemiripannya terbesar di cluster tersebut. Sehingga pada sepuluh kali percobaan clustering dengan fuzzy c-means, data uji penelitian cenderung membentuk 3 buah cluster walaupun masukan jumlah k (cluster) di awal adalah lima. Hal ini dikarenakan nilai kemiripan pada 2 cluster lainnya lebih rendah dibandingkan dengan 3 cluster tersebut. Seolah-olah dalam 2 cluster tersebut tidak memiliki dokumen yang masuk padahal dalam kenyataannya ada walaupun nilai kemiripannya rendah.

Apabila diperhatikan dengan seksama, hasil clustering pada percobaan ke-1 , 3 , 4, 5, 6, 7 , 8, dan 9 menghasilkan pola persebaran topik jurnal yang mirip. Pola yang dibentuk adalah 2 cluster yang tidak memiliki topik jurnal yang dominan. Cluster berikutnya terdapat 6 dokumen jurnal ekonomi, 20 dokumen jurnal hukum, dan 1 dokumen jurnal pendidikan. Cluster ke-3 yang hanya terdapat 2 dokumen jurnal ekonomi, dan cluster terakhir terdapat 12 dokumen jurnal ekonomi, 20 dokumen jurnal komputer, dan 20 dokumen jurnal peternakan. Pola cluster ini berulang walaupun urutan clusternya tidak sama antara tiap percobaan.

Tabel V-5. Hasil Clustering Fuzzy C-Means

Percobaan ke - <i>n</i>	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT
1	6	20	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	20	19	20
2	0	0	0	0	0	6	20	0	1	0	3	0	20	16	0	11	0	0	3	20	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	6	20	0	1	0	12	0	20	19	20
4	0	0	0	0	0	12	0	20	19	20	2	0	0	0	0	6	20	0	1	0	0	0	0	0	0
5	12	0	20	19	20	0	0	0	0	0	6	20	0	1	0	2	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	12	0	20	19	20	0	0	0	0	0	6	20	0	1	0	2	0	0	0	0
7	2	0	0	0	0	6	20	0	1	0	0	0	0	0	0	12	0	20	19	20	0	0	0	0	0
8	6	20	0	1	0	2	0	0	0	0	12	0	20	19	20	0	0	0	0	0	0	0	0	0	0
9	12	0	20	19	20	2	0	0	0	0	0	0	0	0	0	6	20	0	1	0	0	0	0	0	0
10	0	0	0	0	0	10	0	20	19	20	6	20	0	1	0	2	0	0	0	0	2	0	0	0	0

5.2.6 Hasil Clustering Fuzzy C-Means dan Singular Value Decomposition

Tabel V-6 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan. Keterangan judul dokumen lebih lengkap terdapat pada Lampiran Percobaan Clustering Fuzzy C-Means dan Singular Value Decomposition (Lampiran III.6). Sama halnya dengan percobaan clustering fuzzy c-means, tabel V-6 hanya menampilkan dokumen yang nilai kemiripannya terbesar di cluster tersebut. Bedanya, apabila percobaan clustering fuzzy c-means cenderung membentuk 3 cluster. Pada percobaan clustering dengan kombinasi singular value decomposition dan fuzzy c-means cenderung membentuk 2 cluster. Hasil cluster nya pun memiliki pola persebaran data yang mirip di setiap percobaan

walaupun urutan persebaran data pada cluster berbeda. Pola yang ditampilkan adalah 3 cluster yang bernilai 0 dokumen, cluster yang memiliki 1 dokumen jurnal ekonomi dan 20 dokumen jurnal hukum, cluster yang memiliki 19 dokumen jurnal ekonomi, 20 dokumen jurnal komputer, 20 dokumen jurnal pendidikan, dan 20 dokumen jurnal peternakan.

Tabel V-6. Hasil Clustering Fuzzy C-Means dan Singular Value Decomposition

Percobaan ke - n	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	P N	PT	E	H	K	P N	P T	E	H	K	P N	P T	E	H	K	P N	P T
1	1	20	0	0	0	0	0	0	0	0	19	0	20	20	20	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	19	0	20	20	20	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	20	20	20	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	19	0	20	20	20	1	20	0	0	0	0	0	0	0	0
5	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	20	20	20	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	19	0	20	20	20	1	20	0	0	0	0	0	0	0	0
7	19	0	20	20	20	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	20	0	0	0	0	0	0	0	0	19	0	20	20	20
9	0	0	0	0	0	1	20	0	0	0	19	0	20	20	20	0	0	0	0	0	0	0	0	0	0
10	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	20	20	20

5.3 Tabel Confussion

Berdasarkan hasil percobaan clustering pada tabel V-1 sampai dengan V-6, selanjutnya didapatkan tabel confussion yang berisi nilai *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, dan *False Negative (FN)*. Dimana *TP* menunjukkan jumlah

dokumen yang secara benar dimasukkan ke dalam cluster ke- i , FP adalah jumlah dokumen yang secara salah dimasukkan ke dalam cluster ke- i , FN adalah jumlah dokumen yang secara salah tidak dimasukkan ke dalam cluster ke- i , dan TN adalah jumlah dokumen yang secara benar tidak dimasukkan ke dalam cluster ke- i . Nilai-nilai TP, FP, TN, dan FN digunakan untuk mengevaluasi hasil clustering. Untuk mengevaluasi akurasi clustering, digunakan metode evaluasi clustering eksternal yaitu ***precision, recall, f-measure, dan rand index***. Tabel V-7 menampilkan confusion matrix hasil clustering k-means pada sepuluh kali percobaan berdasarkan nilai pada tabel V-1.

Tabel V-7. Confussion Matrix Hasil Clustering K-means

Percobaan ke - n	K-means			
	TP	FP	TN	FN
1	637	430	3570	313
2	523	721	3119	390
3	573	481	3519	377
4	658	512	3488	292
5	622	421	3579	328
6	549	496	3504	401
7	577	540	3540	393
8	640	409	3591	310
9	574	485	3515	376
10	621	559	3441	329
rata - rata	597,4	505,4	3486,6	350,9

Tabel V-8 menampilkan confusion matrix hasil clustering kombinasi metode k-means dan singular value decomposition pada sepuluh kali percobaan berdasarkan nilai pada tabel V-2.

Tabel V-8. Confussion Matrix Hasil Clustering Kombinasi Singular Value

Decomposition dan K-means

Percobaan ke - n	Kombinasi SVD dan K-means			
	TP	FP	TN	FN
1	813	186	3814	137
2	828	160	3840	122
3	794	198	3802	156
4	931	20	3980	19
5	810	179	3821	140
6	896	60	3940	54

Percobaan ke - n	K-means dan SVD			
	TP	TP	TP	TP
7	878	80	3920	72
8	758	267	3733	192
9	847	394	3606	103
10	799	377	3623	151
rata - rata	835,4	192,1	3807,9	114,6

Tabel V-9 menampilkan confusion matrix hasil clustering kombinasi k - medoids pada sepuluh kali percobaan berdasarkan nilai pada tabel V-3.

Tabel V-9. Confussion Matrix Hasil Clustering K-medoids

Percobaan ke - n	K-medoids			
	TP	FP	TN	FN
1	602	520	3479	349
2	634	573	3427	316
3	554	557	3443	396
4	679	568	3432	271
5	636	638	3362	314
6	478	420	2780	327
7	748	595	3405	202
8	581	716	3364	389
9	670	497	3503	280
10	670	807	3193	280
rata-rata	625,2	589,1	3338,8	312,4

Tabel V-10 menampilkan confusion matrix hasil clustering kombinasi k-medoids dan singular value decomposition pada sepuluh kali percobaan berdasarkan nilai pada tabel V-4.

Tabel V-10. Confussion Matrix Hasil Clustering Kombinasi Singular Value

Decomposition dan K-medoids

Percobaan ke - n	Kombinasi SVD dan K-medoids			
	TP	FP	TN	FN
1	788	388	3612	162
2	813	211	3789	137
3	808	431	3569	142
4	811	400	3600	139
5	707	250	3750	243
6	850	128	3872	100
7	886	80	3920	64
8	724	409	3591	226
9	878	76	3924	72
10	736	352	3648	214
rata-rata	802	272,1	3727,9	148

Tabel V-11 menampilkan confusion matrix hasil clustering kombinasi fuzzy c-means pada sepuluh kali percobaan berdasarkan nilai pada tabel V-5.

Tabel V-11. Confussion Matrix Hasil Clustering Fuzzy C-Means

Percobaan ke - n	Fuzzy C-Means			
	TP	FP	TN	FN
1	823	2014	1986	127
2	937	887	4633	564
3	823	2014	1986	127
4	823	2014	1986	127
5	823	2014	1986	127
6	823	2014	1986	127
7	823	2014	1986	127

Percobaan ke - n	Fuzzy C-Means			
	TP	FP	TN	FN
8	823	2014	1986	127
9	823	2014	1986	127
10	803	1896	2104	147
rata-rata	832,4	1889,5	2262,5	172,7

Tabel V-12 menampilkan confusion matrix hasil clustering kombinasi fuzzy c-means dan singular value decomposition pada sepuluh kali percobaan berdasarkan nilai pada tabel V-6.

Tabel V-12. Confussion Matrix Hasil Clustering Kombinasi Singular Value

Decomposition dan Fuzzy C-Means

Percobaan ke - n	Kombinasi SVD dan Fuzzy C-Means			
	TP	FP	TN	FN
1	931	2360	1640	19
2	931	2360	1640	19
3	931	2360	1640	19
4	931	2360	1640	19
5	931	2360	1640	19
6	931	2360	1640	19
7	931	2360	1640	19
8	931	2360	1640	19
9	931	2360	1640	19
10	931	2360	1640	19
rata-rata	931	2360	1640	19

5.4 Nilai Precision, Recall, F-Measure, dan Rand Index

Evaluasi performa dilakukan dengan membandingkan akurasi dan lama waktu eksekusi antara proses clustering dengan reduksi dimensi dan tanpa reduksi dimensi. Evaluasi hasil *clustering* dilakukan untuk mengukur seberapa baik hasil *clustering* yang didapat. Untuk mengevaluasi akurasi clustering, digunakan metode evaluasi clustering eksternal yaitu ***precision, recall, f-measure, dan rand index***. ***Precision*** adalah merupakan cara mengukur ketepatan cluster yang dibentuk oleh perangkat lunak. ***Recall*** adalah tingkat keberhasilan perangkat lunak dalam memisahkan dokumen yang mirip ke dalam cluster yang sama. ***F-measure*** adalah fungsi *harmonic mean* dari *precision* dan *recall*. Semakin nilai *f-measure* mendekati 1 maka menunjukkan hasil clustering semakin baik. Sedangkan ***rand index*** mempresentasikan akurasi hasil clustering aktual dengan nilai prediksi. Perhitungan dilakukan dengan menggunakan persamaan-persamaan pada bab III yaitu persamaan III-1 untuk *precision*, persamaan III-2 untuk *recall*, persamaan III-3 untuk *f-measure*, dan persamaan III-4 untuk *rand index*. Semakin besar nilai *FP* dan *FN* maka nilai *rand index* akan menurun. Sebaliknya, semakin besar nilai *TP* dan *TN* maka nilai *rand index* akan naik menyebabkan akurasi hasil clustering meningkat.

Tabel IV-13 menunjukkan nilai *precision, recall, f-measure, dan rand index* pada sepuluh kali percobaan clustering menggunakan *k-means*. Percobaan ke-2 menunjukkan nilai *precision, recall, f-measure, dan rand index* terkecil dalam sepuluh kali percobaan tersebut. Persebaran topik jurnal yang tidak merata di tiap cluster menyebabkan nilai *TP* dan *TN* menjadi kecil, sedangkan nilai *FP* dan *FN*

menjadi lebih besar. Sehingga akurasi clustering pada percobaan ke-2 pun hanya sebesar 0,766253. Pada percobaan clustering tanpa teknik reduksi, percobaan ke-8 memiliki nilai precision, recall, f-measure, dan rand index terbesar dalam sepuluh kali percobaan. Akurasi hasil clustering pada percobaan ke-8 adalah 0,854747.

Tabel IV-13. Evaluasi Performa Clustering K-means

Percobaan ke - <i>n</i>	K-means			
	Precision	Recall	F - Measure	Rand Index
1	0,597001	0,670526	0,631631	0,849899
2	0,420418	0,572837	0,484933	0,766253
3	0,543643	0,603158	0,571856	0,826667
4	0,562393	0,692632	0,620755	0,837576
5	0,517041	0,654737	0,577798	0,816364
6	0,525359	0,577895	0,550376	0,818788
7	0,516562	0,594845	0,552947	0,815248
8	0,610105	0,673684	0,64032	0,854747
9	0,542021	0,604211	0,571429	0,826061
10	0,526271	0,653684	0,583099	0,820606
Rata - rata	0,5360814	0,6298209	0,5785144	0,8232209

Tabel IV-14 menunjukkan nilai precision, recall, f-measure, dan rand index pada sepuluh kali percobaan clustering menggunakan kombinasi singular value decomposition dan k-means. Percobaan ke-8 menunjukkan nilai precision, recall, f-measure, dan rand index terkecil dalam sepuluh kali percobaan tersebut. Akurasi clustering pada percobaan ke-8 sebesar 0,907273. Pada percobaan clustering tanpa teknik reduksi, percobaan ke-4 memiliki nilai precision, recall, f-measure, dan rand index terbesar dalam sepuluh kali percobaan. Akurasi hasil clustering pada percobaan ke-4 adalah 0,992121.

Tabel IV-14. Evaluasi Performa Clustering Kombinasi Singular Value

Decomposition dan K-means

Percobaan ke - n	Kombinasi SVD dan K-means			
	Precision	Recall	F - Measure	Rand Index
1	0,813814	0,855789	0,834274	0,934747
2	0,838057	0,871579	0,8544889	0,94303
3	0,800403	0,835789	0,817714	0,928485
4	0,97897	0,98	0,979484	0,992121
5	0,819009	0,852632	0,835482	0,935556
6	0,937238	0,943158	0,940189	0,97696
7	0,916493	0,924211	0,920335	0,969293
8	0,739512	0,797895	0,767595	0,907273
9	0,682514	0,891579	0,773163	0,899596
10	0,679422	0,841053	0,751646	0,893333
Rata - rata	0,82054	0,87937	0,847437	0,93804

Tabel IV-15 menunjukkan nilai precision, recall, f-measure, dan rand index pada sepuluh kali percobaan clustering menggunakan k-medoids. Percobaan ke-10 menunjukkan nilai precision, recall, f-measure, dan rand index terkecil dalam sepuluh kali percobaan tersebut. Akurasi clustering pada percobaan ke-8 sebesar 0,830505. Pada percobaan clustering tanpa teknik reduksi, percobaan ke-10 memiliki nilai precision, recall, f-measure, dan rand index terbesar dalam sepuluh kali percobaan. Akurasi hasil clustering pada percobaan ke-10 adalah 0,780404.

Tabel IV-15. Evaluasi Performa Clustering K-medoids

Percobaan ke - n	K-medoids			
	Precision	Recall	F - Measure	Rand Index
1	0,536542	0,633018	0,580801	0,824444

2	0,525269	0,667368	0,587854	0,820404
3	0,49865	0,583158	0,537603	0,807475
4	0,544507	0,714737	0,618116	0,830505
5	0,499215	0,669474	0,571942	0,807677
6	0,532294	0,593789	0,561362	0,813483
7	0,556962	0,787368	0,65242	0,83899
8	0,447957	0,598969	0,512572	0,781188
9	0,574122	0,705263	0,632971	0,84303
10	0,453622	0,705263	0,552122	0,780404
Rata - rata	0,51691	0,66584	0,58078	0,81476

Tabel IV-16 menunjukkan nilai precision, recall, f-measure, dan rand index pada sepuluh kali percobaan clustering menggunakan kombinasi singular value decomposition dan k-medoids. Percobaan ke-8 menunjukkan nilai precision, recall, f-measure, dan rand index terkecil dalam sepuluh kali percobaan tersebut. Akurasi clustering pada percobaan ke-8 sebesar 0,871717. Pada percobaan clustering tanpa teknik reduksi, percobaan ke-7 memiliki nilai precision, recall, f-measure, dan rand index terbesar dalam sepuluh kali percobaan. Akurasi hasil clustering pada percobaan ke-7 adalah 0,970909.

Tabel IV-16. Evaluasi Performa Clustering Kombinasi Singular Value Decomposition dan K-medoids

Percobaan ke - n	Kombinasi SVD dan K-medoids			
	Precision	Recall	F - Measure	Rand Index
1	0,677582	0,849474	0,753853	0,893535
2	0,793945	0,855789	0,823708	0,929697
3	0,652139	0,850526	0,738237	0,884242
4	0,669694	0,853684	0,750578	0,891111
5	0,738767	0,744211	0,741479	0,900404
6	0,869121	0,894737	0,881743	0,953939

7	0,917184	0,932632	0,924843	0,970909
8	0,639011	0,762105	0,695151	0,871717
9	0,920335	0,924211	0,922269	0,970101
10	0,676471	0,774737	0,722277	0,885657
Rata - rata	0,75542	0,84421	0,79541	0,91513

Tabel IV-17 menunjukkan nilai precision, recall, f-measure, dan rand index pada sepuluh kali percobaan clustering menggunakan fuzzy c-means. Percobaan ke-1,3,4,5,6,7,8 dan 9 menunjukkan nilai precision, recall, f-measure, dan rand index terkecil dalam sepuluh kali percobaan tersebut. Akurasi clustering pada percobaan ke-1,3,4,5,6,7,8 dan 9 sebesar 0,567475. Pada percobaan clustering tanpa teknik reduksi, percobaan ke-2 memiliki nilai precision, recall, f-measure, dan rand index terbesar dalam sepuluh kali percobaan. Akurasi hasil clustering pada percobaan ke-2 adalah 0,793334. Dibandingkan dengan metode clustering k-means dan k-medoids, nilai f-measure milik fuzzy c-means sangat kecil. Hal ini dikarenakan besarnya selisih nilai precision dan recall. Sehingga dapat dikatakan bahwa cluster yang terbentuk memiliki relevansi yang rendah.

Tabel IV-17. Evaluasi Performa Clustering Fuzzy C-means

Percobaan ke - <i>n</i>	Fuzzy C-means			
	Precision	Recall	F - Measure	Rand Index
1	0,290095	0,866316	0,434645	0,567475
2	0,513706	0,62425	0,563609	0,793334
3	0,290095	0,866316	0,434645	0,567475
4	0,290095	0,866316	0,434645	0,567475
5	0,290095	0,866316	0,434645	0,567475
6	0,290095	0,866316	0,434645	0,567475
7	0,290095	0,866316	0,434645	0,567475
8	0,290095	0,866316	0,434645	0,567475

9	0,290095	0,866316	0,434645	0,567475
10	0,297518	0,845263	0,440121	0,587273
Rata - rata	0,3132	0,84	0,44809	0,59204

Tabel IV-18 menunjukkan nilai precision, recall, f-measure, dan rand index pada sepuluh kali percobaan clustering menggunakan kombinasi SVD dan fuzzy c-means. Nilai precision, recall, f-measure, dan rand index memiliki nilai yang sama yaitu 0.282893 , 0.98, 0.439047, 0.519394 secara berurutan. Sama dengan hasil clustering fuzzy c-means, nilai f-measure milik kombinasi SVD dan fuzzy c-means sangat kecil. Nilai akurasi yang menurun pun menunjukkan bahwa reduksi dimensi tidak meningkatkan kualitas hasil cluster yang terbentuk.

Tabel IV-18. Evaluasi Performa Clustering Kombinasi Singular Value

Decomposition dan Fuzzy C-means

Percobaan ke - <i>n</i>	Kombinasi SVD dan Fuzzy C-means			
	Precision	Recall	F - Measure	Rand Index
1	0,282893	0,98	0,439047	0,519394
2	0,282893	0,98	0,439047	0,519394
3	0,282893	0,98	0,439047	0,519394
4	0,282893	0,98	0,439047	0,519394
5	0,282893	0,98	0,439047	0,519394
6	0,282893	0,98	0,439047	0,519394
7	0,282893	0,98	0,439047	0,519394
8	0,282893	0,98	0,439047	0,519394
9	0,282893	0,98	0,439047	0,519394
10	0,282893	0,98	0,439047	0,519394
Rata - rata	0,28289	0,98	0,43905	0,51939

5.5 Waktu Komputasi dan Iterasi

Waktu komputasi didapatkan dari lamanya waktu eksekusi proses clustering. Waktu komputasi tidak menghitung waktu proses prapengolahan dan reduksi dimensi. Tabel V-19 menggambarkan waktu komputasi seluruh metode clustering dengan dan tanpa SVD dalam satuan waktu detik atau sekon (*s*) pada sepuluh kali percobaan.

Tabel V-19. Waktu Komputasi Clustering dengan dan tanpa SVD

Percobaan ke - <i>n</i>	Waktu Komputasi (<i>s</i>)					
	K- Means	SVD dan K- Means	K- Medoids	SVD dan K- Medoids	Fuzzy C- Means	SVD dan Fuzzy C- Means
1	269,963	9,569	247,201	29,752	1186,54	33,113
2	118,681	12,08	187,481	22,838	839,795	40,141
3	263,023	9,766	211,707	25,543	1025,935	41,8
4	244,289	9,331	241,417	24,295	1076,837	32,288
5	218,688	11,203	223,917	26,076	928,845	37,06
6	135,392	16,514	270,137	26,766	981,279	39,183
7	134,919	14,009	276,616	25,701	1192,382	33,625
8	184,01	12,375	298,933	22,81	973,182	40,538
9	300,325	9,239	299,787	30,758	878,949	39,486
10	244,051	6,732	367,987	22,186	1055,265	38,828
Rata - rata	211,3341	11,0818	262,5183	25,6725	1013,9009	37,6062

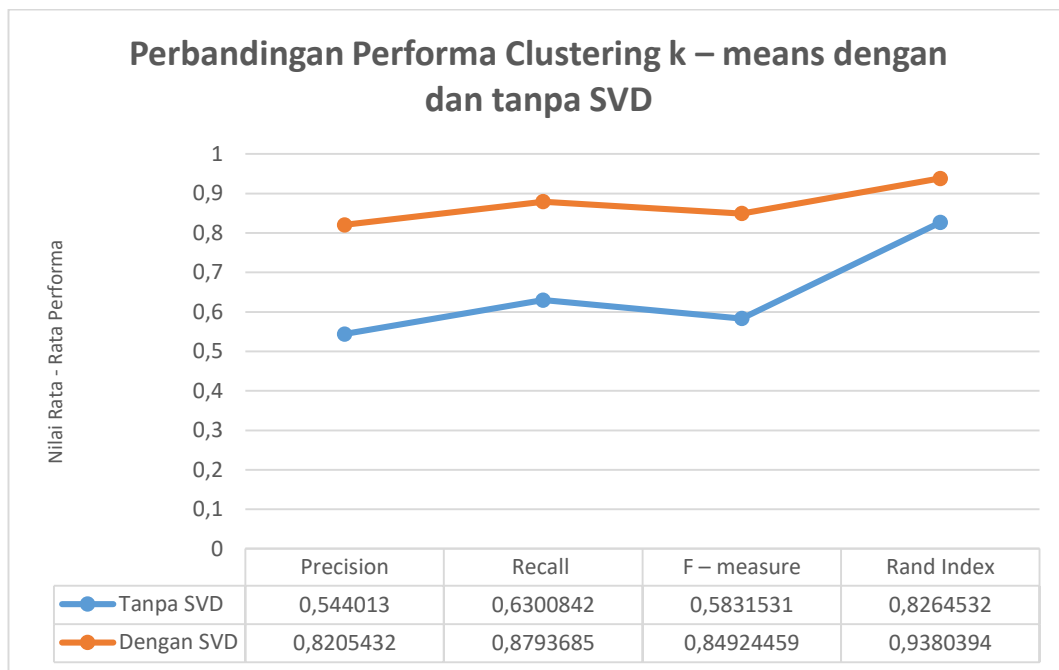
Jumlah Iterasi didapatkan dari banyaknya perulangan terjadi dalam menemukan pusat cluster pada proses clustering. Tabel V-20 menggambarkan jumlah iterasi seluruh metode clustering dengan dan tanpa SVD pada sepuluh kali percobaan.

Tabel V-20. Banyak Iterasi Clustering dengan dan tanpa SVD

Percobaan ke - n	Banyak Iterasi					
	K-Means	SVD dan K-Means	K-Medoids	SVD dan K-Medoids	Fuzzy C-Means	SVD dan Fuzzy C-Means
1	7	4	1	1	22	12
2	4	5	1	1	18	12
3	8	4	1	1	22	14
4	9	4	1	1	23	13
5	8	5	1	1	20	13
6	5	6	1	1	21	14
7	5	6	1	1	21	14
8	6	5	1	1	21	13
9	11	4	1	1	19	15
10	9	3	1	1	23	14
Rata - rata	7,2	4,6	1	1	21	13,4

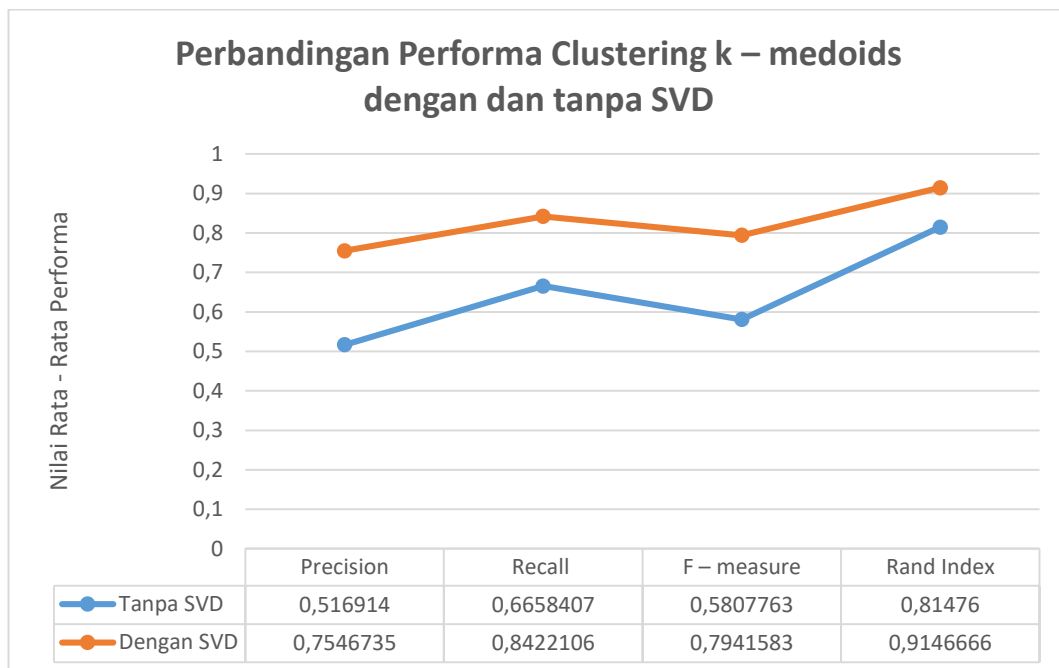
5.6 Analisis Penelitian

Pada gambar V-1 menunjukkan peningkatan kualitas clustering kombinasi SVD dan k-means berdasarkan nilai rata-rata *precision*, *recall*, *f-measure*, dan *rand index* dibandingkan dengan hasil clustering tanpa SVD. Kombinasi SVD dan k-means menyebabkan nilai rata-rata *rand index* sebesar 0,93 selisih 0,11 dibandingkan nilai *rand index* k-means tanpa reduksi dimensi sebesar 0,82. Sehingga, secara langsung reduksi dimensi SVD memberikan pengaruh yang baik terhadap akurasi hasil clustering data uji penelitian menggunakan metode k-means.



Gambar V-1. Perbandingan Performa Clustering k-means dengan dan tanpa SVD

Gambar V-2 menunjukkan nilai rata-rata precision, recall, f-measure, dan rand index yang dihasilkan oleh proses clustering k-medoids dengan dan tanpa SVD. Walaupun pada kombinasi SVD dan k-medoids nilai precision dan f-measure tidak mencapai 0,8 seperti pada percobaan k-means. Namun nilai rand index tetap meningkat sebesar 0,10 yaitu 0,91 dibandingkan dengan nilai rand index clustering tanpa reduksi dimensi yang hanya sebesar 0,81. Sehingga pada hasil clustering data uji penelitian menggunakan metode k-medoids, teknik reduksi dimensi SVD memberikan pengaruh yang baik pada peningkatan akurasi.

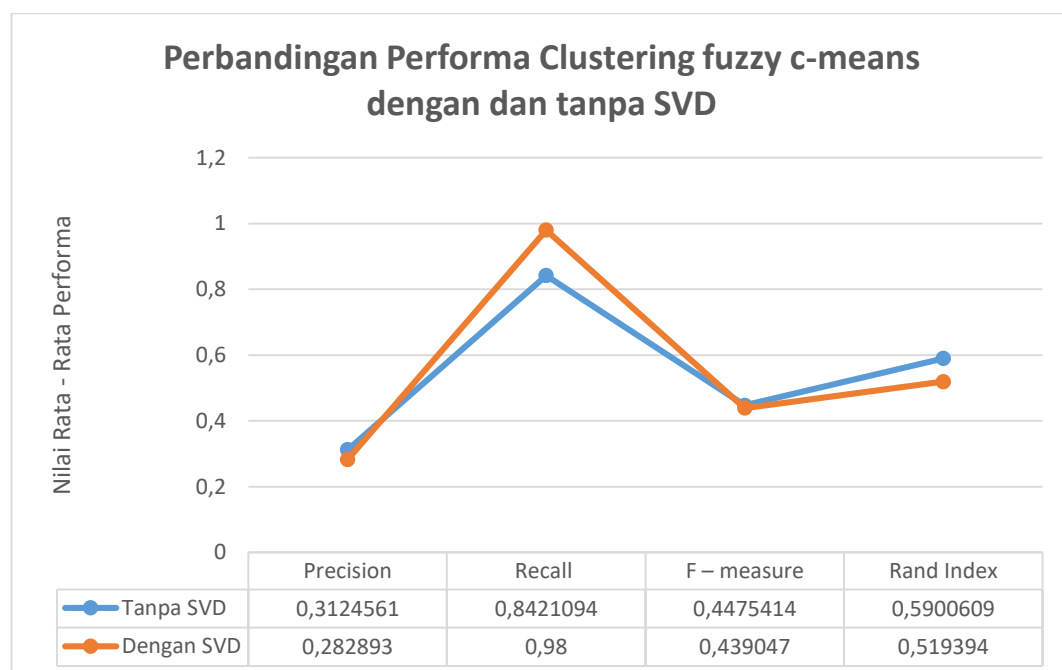


Gambar V-2. Perbandingan Performa Clustering k-medoids dengan dan tanpa
SVD

Gambar V-3 menunjukkan nilai rata-rata precision, recall, f-measure, dan rand index yang dihasilkan oleh proses clustering fuzzy c-means dengan dan tanpa SVD. Berbeda dengan metode k-means dan k-medoids, akurasi hasil clustering fuzzy c-means tidak menunjukkan peningkatan dibandingkan dengan tanpa menggunakan SVD. Sebaliknya hasil akurasi menjadi turun. Hanya terlihat kombinasi SVD dan Fuzzy C-means meningkatkan nilai recall sebesar 0,98 dibandingkan tanpa reduksi dimensi yaitu 0,84.

Dari total 10 kali percobaan, kombinasi antara clustering fuzzy c-means dan SVD cenderung membentuk 2 cluster. Kecenderungan inilah yang menyebabkan kecilnya hasil dari precision, sedangkan nilai recall sangat bergantung pada nilai kategori yang diambil pada masing-masing cluster yang terbentuk. Nilai rand index kombinasi SVD dan fuzzy c-means sebesar 0,51

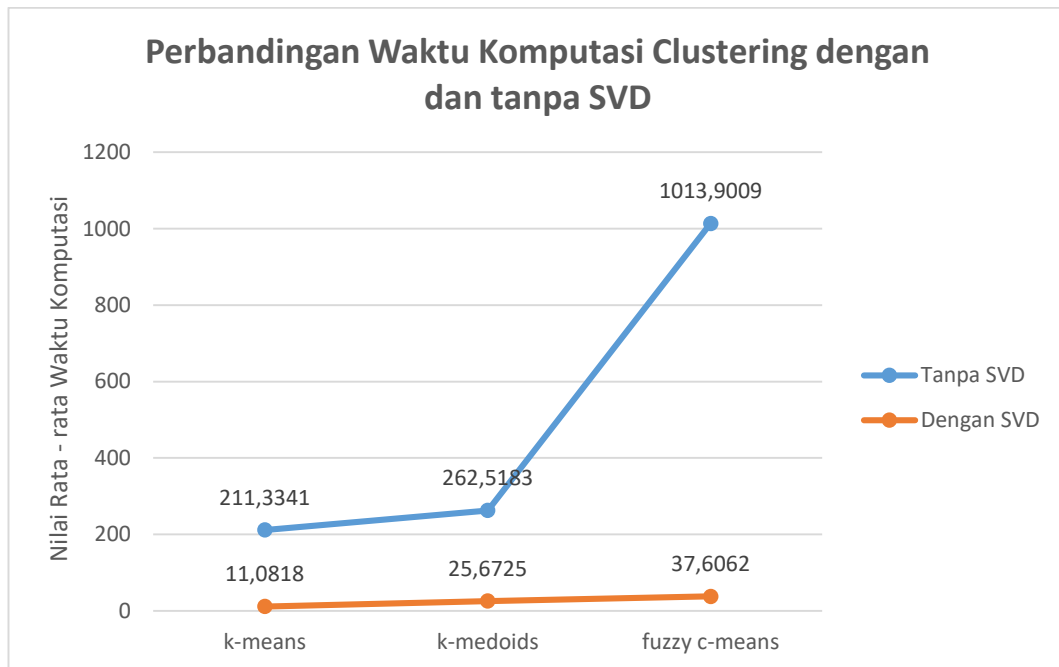
mengalami penurunan dibandingkan nilai rand index tanpa SVD yaitu 0,59. Akurasi yang rendah kemungkinan disebabkan oleh prapengolahan dokumen yang kurang baik serta reduksi dimensi pada fitur sehingga berkurangnya ragam term untuk dilakukan fuzzifikasi. Sehingga dapat disimpulkan, bahwa untuk data uji penelitian ini teknik reduksi SVD tidak memberikan pengaruh yang baik pada akurasi hasil clustering menggunakan metode fuzzy c-means.



Gambar V-3. Perbandingan Performa Clustering fuzzy c-means dengan dan tanpa SVD

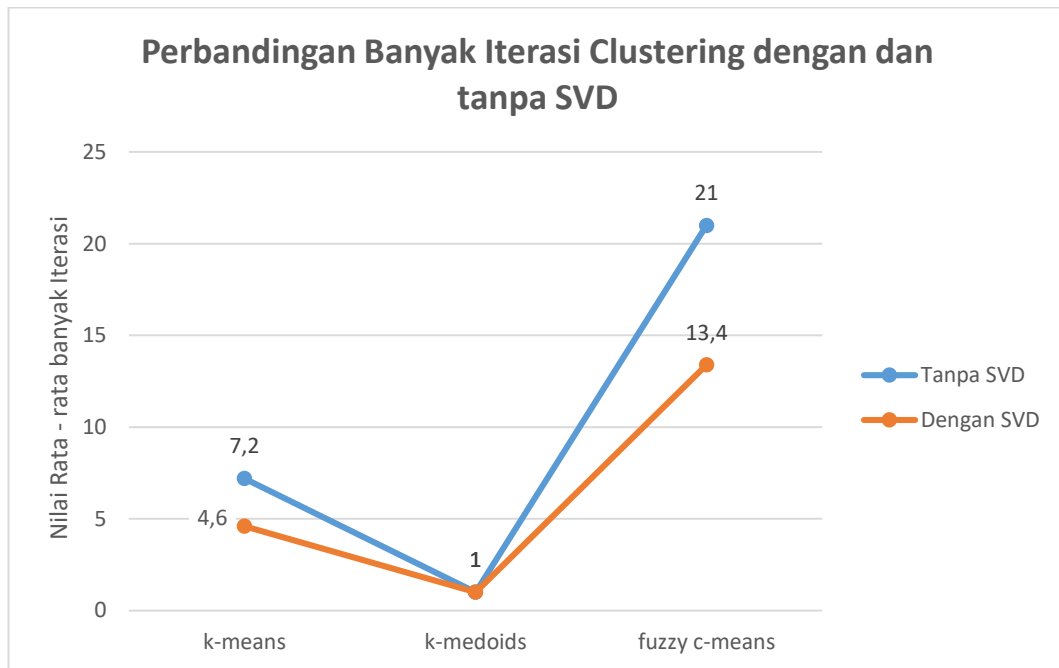
Gambar V-4 menunjukkan nilai rata-rata waktu eksekusi proses clustering dengan dan tanpa SVD. Berdasarkan waktu eksekusi pembentukan clustering yang digambarkan pada gambar V-4 menunjukkan bahwa SVD memberikan pengaruh signifikan pada waktu komputasi. Pada tiap metode clustering waktu komputasi pun menjadi lebih cepat dibandingkan tanpa reduksi

dimensi. Hal ini dikarenakan, clustering dengan reduksi dimensi hanya memproses fitur yang lebih sedikit pada data yang belum direduksi dimensinya.



Gambar V-4. Perbandingan Waktu Komputasi Clustering dengan dan tanpa SVD

Gambar V-5 menunjukkan nilai rata-rata banyak iterasi pada proses clustering dengan dan tanpa SVD. Berdasarkan banyak iterasi pembentukan clustering yang digambarkan pada gambar V-5 menunjukkan bahwa SVD memberikan pengaruh signifikan pada jumlah iterasi kecuali pada k-medoids.



Gambar V-5. Perbandingan Waktu Komputasi Clustering dengan dan tanpa SVD

5.7 Kesimpulan

BAB VI

KESIMPULAN DAN SARAN

6.1 Pendahuluan

Pada bab VI telah dilakukan analisis penelitian mengenai pengaruh teknik reduksi dimensi *Singular Value Decomposition (SVD)* terhadap hasil pengelompokkan dokumen berdimensi tinggi menggunakan beberapa metode clustering. Pada bab VI penulis memberikan kesimpulan dan saran yang diharapkan dapat menjadi acuan bagi penelitian selanjutnya di bidang yang sama.

6.2 Kesimpulan

Pada penelitian ini, teknik reduksi dimensi Singular Value Decomposition dikombinasikan dengan metode clustering k-means, k-medoids, dan fuzzy c-means untuk mengelompokkan dokumen. Berdasarkan hasil pengujian, ditemukan bahwa SVD berpengaruh terhadap clustering k-means dan k-medoids. Dibuktikan dengan meningkatnya akurasi rand index hasil clustering k-means dan k-medoids sebesar 0,11 dan 0,10. Sebaliknya, kombinasi fuzzy c-means dan SVD tidak memberikan hasil akurasi yang baik. Fenomena ini kemungkinan disebabkan oleh proses prapengolahan dokumen yang kurang baik dan reduksi dimensi pada fitur dokumen. Sedangkan berdasarkan waktu eksekusi dan jumlah iterasi pembentukan clustering, SVD mampu meningkatkan waktu eksekusi dan mengurangi jumlah iterasi secara signifikan pada semua metode clustering.

6.3 Saran

Pada penelitian selanjutnya, diharapkan :

1. Dapat menemukan solusi untuk meningkatkan akurasi fuzzy c-means baik menggunakan SVD maupun teknik reduksi lainnya.
2. Melakukan percobaan clustering dengan teknik reduksi dimensi lain untuk data yang memiliki lebih banyak *noise* dan *sparse*.

DAFTAR PUSTAKA

- Astuti, D. (2016). *Pengelompokkan Dokumen Berita Menggunakan Fuzzy C-means*. (Strata 1), Universitas Sriwijaya, Palembang.
- Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*: Elsevier.
- Huang, A. (2008). *Similarity measures for text document clustering*. Paper presented at the Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand.
- Indranandita, A., Susanto, B., & Rahmat, A. (2011). Sistem Klasifikasi dan Pencarian Jurnal dengan Menggunakan Metode Naive Bayes dan Vector Space Model. *Jurnal Informatika*, 4(2).
- Kadhim, A. I., Cheah, Y.-N., & Ahamed, N. H. (2014). *Text Document Preprocessing and Dimension Reduction Techniques for Text Document Clustering*. Paper presented at the Artificial Intelligence with Applications in Engineering and Technology (ICAIET), 2014 4th International Conference on.
- Kaufman, L., & Rousseeuw, P. (1987). *Clustering by means of medoids*: North-Holland.
- Madhulatha, T. S. (2011). Comparison between k-means and k-medoids clustering algorithms *Advances in Computing and Information Technology* (pp. 472-481): Springer.
- Nazief, B. A., & Adriani, M. (1996). Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia. *Internal publication, Faculty of Computer Science, University of Indonesia, Depok, Jakarta*, 41.
- Nur'aini, K., Najahaty, I., Hidayati, L., Murfi, H., & Nurrohmah, S. (2015). *Combination of singular value decomposition and K-means clustering methods for topic detection on Twitter*. Paper presented at the Advanced Computer Science and Information Systems (ICACSIS), 2015 International Conference on.

Pressman, R. S. (2005). *Software engineering: a practitioner's approach*: Palgrave Macmillan.

Ross, Tj. (2005). *Fuzzy Logic With Engineering Application Third Edition*: Grava Media.

Sibagariang, S. (2015). *Clustering Dokumen Berbahasa Indonesia Menggunakan K-Medoids*. (Strata 1), Universitas Sriwijaya, Palembang.

Lampiran Dokumentasi Source Code

MainFrame.java

```
package boundary;

import
cern.colt.matrix.DoubleMatrix2D;
import
cern.colt.matrix.impl.DenseDoubleMatr
ix2D;
import control.MatrixOperator;
import java.io.IOException;
import javax.swing.JFileChooser;
import control.Weight;
import control.DimensionReduction;
import entity.Document;
import control.KMedoids;
import
cern.colt.matrix.linalg.SingularValue
Decomposition;
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import javax.swing.JTable;
import
javax.swing.table.DefaultTableModel;
import control.FuzzyCMeans;
import control.KMeans;
import control.PopupController;
import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;
import javax.swing.JOptionPane;
import javax.swing.RowFilter;
import javax.swing.RowSorter;
import javax.swing.SortOrder;
import javax.swing.table.TableModel;
import
javax.swing.table.TableRowSorter;

/**
 *
 * @author WIN8
 */
public class MainFrame extends
javax.swing.JFrame {

    private Weight bobot;
    private final MatrixOperator
matrixGen;
    private PopupController popup;
    private Centroid popupPusat;
    /**
     * Creates new form MainFrame
     */
    public MainFrame() {
        initComponents();
        this.setTitle("Sidang Program
Tugas Akhir Novita");

        this.setLocationRelativeTo(null);
        matrixGen = new
MatrixOperator();
        popupPusat = new Centroid();
        popup = new
PopupController();

        radioButtonKMedoids.setEnabled(false)
;

        radioButtonKMedoidsSVD.setEnabled(fal
se);

        radioButtonKMeans.setEnabled(false);

        radioButtonKMeansSVD.setEnabled(false
);

        radioButtonFuzzy.setEnabled(false);

        radioButtonFuzzySVD.setEnabled(false)
;

        /**
         * This method is called from
within the constructor to initialize
the form.
         * WARNING: Do NOT modify this
code. The content of this method is
always
         * regenerated by the Form
Editor.
         */
        @SuppressWarnings("unchecked")
        // <editor-fold
defaultstate="collapsed"
desc="Generated Code">
        private void initComponents() {

            buttonGroupCluster = new
javax.swing.ButtonGroup();
            jPanel1 = new
javax.swing.JPanel();
            buttonMuatDokumen = new
javax.swing.JButton();
            textFieldPathDokumen = new
javax.swing.JTextField();
            jPanel2 = new
javax.swing.JPanel();
            radioButtonKMedoids = new
javax.swing.JRadioButton();
            radioButtonKMedoidsSVD = new
javax.swing.JRadioButton();
            radioButtonKMeans = new
javax.swing.JRadioButton();
            radioButtonKMeansSVD = new
javax.swing.JRadioButton();
            radioButtonFuzzy = new
javax.swing.JRadioButton();
            radioButtonFuzzySVD = new
javax.swing.JRadioButton();
            buttonCluster = new
javax.swing.JButton();
            textFieldK = new
javax.swing.JTextField();
            labelK = new
javax.swing.JLabel();
            jScrollPane1 = new
javax.swing.JScrollPane();
            tableCluster = new
javax.swing.JTable();
```

```

        labelClusterMethod = new
javafx.swing.JLabel();
        textFieldDokumen = new
javafx.swing.JTextField();
        labelTerm = new
javafx.swing.JLabel();
        labelDokumen = new
javafx.swing.JLabel();
        textFieldTerm = new
javafx.swing.JTextField();
        labelWaktu = new
javafx.swing.JLabel();
        waktuEksekusi = new
javafx.swing.JLabel();
        labelIterasi = new
javafx.swing.JLabel();
        iterasi = new
javafx.swing.JLabel();

setDefaultCloseOperation(javafx.swing.
WindowConstants.EXIT_ON_CLOSE);
setTitle("Program TA
Novita");

jPanel1.setBorder(javafx.swing.BorderF
actory.createEtchedBorder());

buttonMuatDokumen.setText("Muat");

buttonMuatDokumen.addActionListener(n
ew java.awt.event.ActionListener() {
    public void
actionPerformed(java.awt.event.Action
Event evt) {

buttonMuatDokumenActionPerformed(evt)
;

    }
});

javafx.swing.GroupLayout
jPanel1Layout = new
javafx.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(jav
afx.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequent
ialGroup()

.addContainerGap()

.addComponent(buttonMuatDokumen)

.addPreferredGap(javafx.swing.LayoutSt
yle.ComponentPlacement.UNRELATED)

.addComponent(textFieldPathDokumen,
javafx.swing.GroupLayout.DEFAULT_SIZE,
745, Short.MAX_VALUE)

.addContainerGap()

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(jav
afx.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequent
ialGroup()

.addContainerGap()

.addComponent(buttonMuatDokumen)

.addComponent(textFieldPathDokumen,
javafx.swing.GroupLayout.PREFERRED_SIZ
E,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZ
E))

.addContainerGap(15,
Short.MAX_VALUE)

);

jPanel2.setBorder(javafx.swing.BorderF
actory.createEtchedBorder());

buttonGroupCluster.add(radioButtonKMe
doids);

radioButtonKMedoids.setText("K-
Medoids");

buttonGroupCluster.add(radioButtonKMe
doidsSVD);

radioButtonKMedoidsSVD.setText("K-
Medoids + SVD");

buttonGroupCluster.add(radioButtonKMe
ans);

radioButtonKMeans.setText("K-
Means");

buttonGroupCluster.add(radioButtonKMe
ansSVD);

radioButtonKMeansSVD.setText("K-Means
+ SVD");

buttonGroupCluster.add(radioButtonFuz
zy);

radioButtonFuzzy.setText("Fuzzy C
Means");

buttonGroupCluster.add(radioButtonFuz
zySVD);

radioButtonFuzzySVD.setText("Fuzzy C
Means + SVD");

```

```

        javax.swing.GroupLayout
jPanel2Layout = new
javax.swing.GroupLayout (jPanel2);

jPanel2.setLayout (jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup (jav
ax.swing.GroupLayout.Alignment.LEADING)

.addGroup (jPanel2Layout.createSequent
ialGroup ()
.addContainerGap ()

.addGroup (jPanel2Layout.createParalle
lGroup (javax.swing.GroupLayout.Alignm
ent.LEADING)

.addComponent (radioButtonKMedoids)

.addComponent (radioButtonKMedoidsSVD)

.addComponent (radioButtonKMeans)

.addComponent (radioButtonKMeansSVD)

.addComponent (radioButtonFuzzy)

.addComponent (radioButtonFuzzySVD))

.addContainerGap (javax.swing.GroupLay
out.DEFAULT_SIZE, Short.MAX_VALUE))
);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup (jav
ax.swing.GroupLayout.Alignment.LEADING)

.addGroup (jPanel2Layout.createSequent
ialGroup ()
.addContainerGap ()

.addComponent (radioButtonKMedoids)

.addPreferredGap (javax.swing.LayoutSt
yle.ComponentPlacement.UNRELATED)

.addComponent (radioButtonKMedoidsSVD)

.addPreferredGap (javax.swing.LayoutSt
yle.ComponentPlacement.UNRELATED)

.addComponent (radioButtonKMeans)

.addPreferredGap (javax.swing.LayoutSt
yle.ComponentPlacement.UNRELATED)

.addComponent (radioButtonKMeansSVD)

.addPreferredGap (javax.swing.LayoutSt
yle.ComponentPlacement.UNRELATED)

.addComponent (radioButtonFuzzy)

.addPreferredGap (javax.swing.LayoutSt
yle.ComponentPlacement.UNRELATED)

.addComponent (radioButtonFuzzySVD)

.addContainerGap (javax.swing.GroupLay
out.DEFAULT_SIZE, Short.MAX_VALUE))
);

buttonCluster.setText ("Mulai");

buttonCluster.addActionListener (new
java.awt.event.ActionListener () {
    public void
actionPerformed (java.awt.event.Action
Event evt) {

buttonClusterActionPerformed (evt);
    }
});

labelK.setText ("Jumlah
Cluster (k):");

tableCluster.setModel (new
javax.swing.table.DefaultTableModel (
    new Object [][] {

    },
    new String [] {

    }
));

jScrollPane1.setViewportViewView (tableClu
ster);

labelClusterMethod.setText ("Cluster
Method:");

textfieldDokumen.setEditable (false);

labelTerm.setText ("Jumlah
Term:");

labelDokumen.setText ("Jumlah
Dokumen:");

textfieldTerm.setEditable (false);

labelWaktu.setText ("Waktu
Eksekusi:");

labelIterasi.setText ("Iterasi:");

        javax.swing.GroupLayout
layout = new
javax.swing.GroupLayout (getContentPan
e());

getContentPane ().setLayout (layout);
        layout.setHorizontalGroup(

layout.createParallelGroup (javax.swin
g.GroupLayout.Alignment.LEADING)

.addGroup (layout.createSequentialGrou
p()

```

```

        .addContainerGap()

        .addGroup(layout.createParallelGroup(
            javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

        .addComponent(labelClusterMethod)
            .addGap(0, 0,
                Short.MAX_VALUE))

        .addGroup(layout.createSequentialGroup()

        .addGroup(layout.createParallelGroup(
            javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jPanel1,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)

        .addGroup(layout.createSequentialGroup()

        .addGroup(layout.createParallelGroup(
            javax.swing.GroupLayout.Alignment.TRAILING, false)

        .addGroup(layout.createSequentialGroup()

        .addComponent(labelK)

        .addGap(36, 36, 36)

        .addComponent(textFieldK,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jPanel2,
            javax.swing.GroupLayout.Alignment.LEADING,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)

        .addComponent(buttonCluster,
            javax.swing.GroupLayout.Alignment.LEADING,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

        .addGroup(layout.createParallelGroup(
            javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(jScrollPane1)

        .addGroup(layout.createSequentialGroup()

```

```

        .addGroup(layout.createParallelGroup(
            javax.swing.GroupLayout.Alignment.LEADING)

        .addComponent(labelTerm)

        .addComponent(textfieldTerm,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(37, 37, 37)

        .addGroup(layout.createParallelGroup(
            javax.swing.GroupLayout.Alignment.LEADING, false)

        .addComponent(textfieldDokumen)

        .addComponent(labelDokumen))

        .addGap(0, 0, Short.MAX_VALUE))))

        .addContainerGap()))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            layout.createSequentialGroup()
                .addGap(233, 233,
                    233)

        .addComponent(labelIterasi)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(iterasi)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            Short.MAX_VALUE)

        .addComponent(labelWaktu)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(waktuEksekusi)
            .addGap(102, 102,
                102))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.
                GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()

        .addComponent(jPanel1,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(17, 17, 17)

```

```

.addComponent(labelClusterMethod)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(
    javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(
    javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(labelTerm)

.addComponent(labelDokumen))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(
    javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(textfieldDokumen,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(textfieldTerm,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 121,
    Short.MAX_VALUE)

.addComponent(jScrollPane1,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(
    javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(labelWaktu)

.addComponent(waktuEksekusi)

.addComponent(labelIterasi)

.addComponent(iterasi)

.addGap(4, 4)

.addGroup(layout.createSequentialGroup()

```

```

.addComponent(jPanel2,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(15, 15)

.addGroup(layout.createParallelGroup(
    javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(textFieldK,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(labelK)

.addGap(18, 18)

.addComponent(buttonCluster)

.addGap(0, 0,
    Short.MAX_VALUE)))

);

pack();
} // </editor-fold>

private void
buttonMuatDokumenActionPerformed(java
    .awt.event.ActionEvent evt) {
    // TODO add your handling
    code here:
    //
    textFieldPathDokumen.setText("");
    JFileChooser chooser = new
    JFileChooser();

    chooser.setCurrentDirectory(new
    java.io.File("."));

    chooser.setDialogTitle("Select
    Folder");

    chooser.setFileSelectionMode(JFileCho
    oser.DIRECTORIES_ONLY);

    chooser.setAcceptAllFileFilterUsed(fa
    lse);

    if
    (chooser.showOpenDialog(null) ==
    JFileChooser.APPROVE_OPTION) {
        try {
            String selectedFolder
            =
            String.valueOf(chooser.getSelectedFil
            e());

            textFieldPathDokumen.setText(selected
            Folder);

            System.out.println("getSelectedFile()
            : " + selectedFolder);
            bobot = new
            Weight(selectedFolder);

```

```

        bobot.doPembobotan();

if(bobot.getListDocument().isEmpty())
{

JOptionPane.showMessageDialog(null,"F
ile di dalam folder terpilih tidak
dapat diproses",
        "Gagal
Muat", JOptionPane.ERROR_MESSAGE);
}else{

textfieldTerm.setText(String.valueOf(
bobot.getGlobalTermList().getTotalTer
m()));

textfieldDokumen.setText(String.value
Of(bobot.getListDocument().size()));

radioButtonKMedoids.setEnabled(true);

radioButtonKMedoidsSVD.setEnabled(tru
e);

radioButtonKMeans.setEnabled(true);

radioButtonKMeansSVD.setEnabled(true)
;

radioButtonFuzzy.setEnabled(true);

radioButtonFuzzySVD.setEnabled(true);
        }
        catch
        (NullPointerException e) {

JOptionPane.showMessageDialog(null,"F
ile di dalam folder terpilih tidak
dapat diproses",
        "Gagal Muat",
JOptionPane.ERROR_MESSAGE);
        }

        } else {
            System.out.println("Gagal
Muat");
        }
    }

    private void
buttonClusterActionPerformed(java.awt
.event.ActionEvent evt) {
        // TODO add your handling
code here:

if(textFieldK.getText().isEmpty() ||
Integer.valueOf(textFieldK.getText())
< 1){

JOptionPane.showMessageDialog(null,"N
ilai K tidak boleh kosong atau nol",
        "Gagal",
JOptionPane.ERROR_MESSAGE);
        }else
if(Integer.valueOf(textFieldK.getText
())
        >=
bobot.getListDocument().size()){

JOptionPane.showMessageDialog(null,"N
ilai K tidak boleh sama atau lebih
dari jumlah dokumen",

```

```

        "Gagal",
JOptionPane.ERROR_MESSAGE);
        }else{

if(radioButtonKMedoids.isSelected()){

System.out.println("\n-----
KMedoids-----\n");
            DoubleMatrix2D data =
new
DenseDoubleMatrix2D(matrixGen.transpo
se(bobot.getHasilPembobotan()));
            KMedoids kmedoids =
new
            KMedoids(data,
Integer.valueOf(textFieldK.getText())
);
            kmedoids.cluster();

System.out.println(kmedoids.getMedoid
sToDisplay().get(0, 0));

System.out.println(kmedoids.getPartit
ion());

populateTable(kmedoids.getPartition()
);

popup.showPusatCluster(kmedoids.getMe
doidString());

waktuEksekusi.setText(String.valueOf(
kmedoids.getExecTime()) + " s");

iterasi.setText(String.valueOf(kmedoi
ds.getTotalIterations()));

        }else
if(radioButtonKMedoidsSVD.isSelected(
)){

System.out.println("\n-----
KMedoids + SVD-----
-\n");
            DoubleMatrix2D data =
new
DenseDoubleMatrix2D(bobot.getHasilPem
bobotan());

SingularValueDecomposition svd = new
SingularValueDecomposition(data);

System.out.println("U:");

System.out.println(svd.getU()
        +
"\n");

System.out.println("S:");

System.out.println(svd.getS()
        +
"\n");

System.out.println("Vt:");

System.out.println(svd.getV().viewDic
e() + "\n");

System.out.println("U*S:");

```

```

        double[][] US =
matrixGen.multiply(svd.getU(),
svd.getS());

System.out.println("\nS*V:");
        double[][] SV =
matrixGen.multiply(svd.getS(),
svd.getV().viewDice());

System.out.println("\n");
        DimensionReduction
reduksi = new
DimensionReduction(bobot.getGlobalTer
mList(), bobot.getListDocument(),
US);

reduksi.doReduksiDimensi();

        DoubleMatrix2D
reducedMatrix = new
DenseDoubleMatrix2D(reduksi.getReduce
dMatrix());

        DoubleMatrix2D
transposedSV = new
DenseDoubleMatrix2D(matrixGen.transpo
se(SV));

        KMedoids kmedoids =
new
KMedoids(reducedMatrix,
transposedSV,
Integer.valueOf(textFieldK.getText())
);

        kmedoids.cluster();

System.out.println(kmedoids.getMedoid
s().size());

System.out.println(kmedoids.getPartit
ion());

populateTable(kmedoids.getPartition()
);

popup.showPusatCluster(kmedoids.getMe
doidString());

waktuEksekusi.setText(String.valueOf(
kmedoids.getExecTime()) + " s");

iterasi.setText(String.valueOf(kmedoi
ds.getTotalIterations()));

    }else
if (radioButtonKMeans.isSelected()) {

System.out.println("\n-----
KMeans-----\n");
        DoubleMatrix2D data =
new
DenseDoubleMatrix2D(matrixGen.transpo
se(bobot.getHasilPembobotan()));
        KMeans kmeans = new
KMeans(data,
Integer.valueOf(textFieldK.getText())
);

        kmeans.cluster();

System.out.println(kmeans.getPartitio
n());

        populateTable(kmeans.getPartition());

        popup.showPusatCluster(kmeans.getMean
s());

        waktuEksekusi.setText(String.valueOf(
kmeans.getExecTime()) + " s");

        iterasi.setText(String.valueOf(kmeans
.getTotalIterations()));

    }else
if (radioButtonKMeansSVD.isSelected())
{

System.out.println("\n-----
KMeans + SVD-----
\n");
        DoubleMatrix2D data =
new
DenseDoubleMatrix2D(bobot.getHasilPem
bobotan());

        SingularValueDecomposition svd = new
SingularValueDecomposition(data);

System.out.println("U:\n");

System.out.println(svd.getU()
+
"\n");

System.out.println("S:\n");

System.out.println(svd.getS()
+
"\n");

System.out.println("Vt:\n");

System.out.println(svd.getV().viewDic
e() + "\n");

System.out.println("U*S:");
        double[][] US =
matrixGen.multiply(svd.getU(),
svd.getS());

System.out.println("S*V:");
        double[][] SV =
matrixGen.multiply(svd.getS(),
svd.getV().viewDice());

System.out.println("\n");
        DimensionReduction
reduksi = new
DimensionReduction(bobot.getGlobalTer
mList(), bobot.getListDocument(),
US);

reduksi.doReduksiDimensi();

        DoubleMatrix2D
reducedMatrix = new
DenseDoubleMatrix2D(reduksi.getReduce
dMatrix());

        DoubleMatrix2D
transposedSV = new
DenseDoubleMatrix2D(matrixGen.transpo
se(SV));

```



```

        KMeans kmeans = new
KMeans(reducedMatrix, transposedSV,
Integer.valueOf(textFieldK.getText())
);
        kmeans.cluster();

System.out.println(kmeans.getPartitio
n());

populateTable(kmeans.getPartition());

popup.showPusatCluster(kmeans.getMean
s());

waktuEksekusi.setText(String.valueOf(
kmeans.getExecTime()) + " s");

iterasi.setText(String.valueOf(kmeans
.getTotalIterations()));

        }else
if (radioButtonFuzzy.isSelected()){

System.out.println("\n-----
Fuzzy C Means-----
\n");
        DoubleMatrix2D data =
new
DenseDoubleMatrix2D(matrixGen.transpo
se(bobot.getHasilPembobotan()));
        FuzzyCMeans
fuzzycmeans = new FuzzyCMeans(data,
Integer.valueOf(textFieldK.getText())
);

fuzzycmeans.cluster();

System.out.println(fuzzycmeans.getPar
tition());

populateTable(fuzzycmeans.getPartitio
n());

popup.showPusatCluster(fuzzycmeans.ge
tMeans());

waktuEksekusi.setText(String.valueOf(
fuzzycmeans.getExecTime()) + " s");

iterasi.setText(String.valueOf(fuzzyc
means.getTotalIterations()));

        }else
if (radioButtonFuzzySVD.isSelected()){

System.out.println("\n-----
Fuzzy C Means + SVD-----
-----\n");
        DoubleMatrix2D data =
new
DenseDoubleMatrix2D(bobot.getHasilPem
bobotan());

SingularValueDecomposition svd = new
SingularValueDecomposition(data);

System.out.println("U:\n");

System.out.println(svd.getU() +
"\n");

System.out.println("S:\n");

System.out.println(svd.getS() +
"\n");

System.out.println("Vt:\n");

System.out.println(svd.getV().viewDic
e() + "\n");

System.out.println("U*S:");
        double[][] US =
matrixGen.multiply(svd.getU(),
svd.getS());

System.out.println("S*V:");
        double[][] SV =
matrixGen.multiply(svd.getS(),
svd.getV().viewDice());

System.out.println("\n");
        DimensionReduction
reduksi = new
DimensionReduction(bobot.getGlobalTer
mList(), bobot.getListDocument(),
US);

reduksi.doReduksiDimensi();

        DoubleMatrix2D
reducedMatrix = new
DenseDoubleMatrix2D(reduksi.getReduce
dMatrix());

        DoubleMatrix2D
transposedSV = new
DenseDoubleMatrix2D(matrixGen.transpo
se(SV));

        FuzzyCMeans
fuzzycmeans = new
FuzzyCMeans(reducedMatrix,
transposedSV,
Integer.valueOf(textFieldK.getText())
);

fuzzycmeans.cluster();

System.out.println(fuzzycmeans.getPar
tition());

populateTable(fuzzycmeans.getPartitio
n());

popup.showPusatCluster(fuzzycmeans.ge
tMeans());

waktuEksekusi.setText(String.valueOf(
fuzzycmeans.getExecTime()) + " s");

iterasi.setText(String.valueOf(fuzzyc
means.getTotalIterations()));

        }
    }
}

void populateTable(DoubleMatrix2D
hasilClustering){

```

```

        DefaultTableModel tabelModel
        = new
        DefaultTableModel(hasilClustering.rows(), hasilClustering.columns());
        for(int i=0;
        i<hasilClustering.rows(); i++){
            double max =
            Integer.MIN_VALUE;
            int idx = 0;
            for(int j=0;
            j<hasilClustering.columns(); j++){
                //
                tabelModel.setValueAt(hasilClustering
                .get(i, j), i, j);

                if(hasilClustering.get(i, j) > max){
                    max =
                    hasilClustering.get(i, j);
                    idx = j;
                }
                else{
                    //
                    //
                    tabelModel.setValueAt("", i, j);
                    //
                }

                tabelModel.setValueAt(bobot.getListDo
                cument().get(i).getJudul(), i, idx);
            }

            tableCluster.setModel(tabelModel);
        }

        /**
         * @param args the command line
         arguments
         */
        public static void main(String
        args[]) {
            /* Set the Nimbus look and
            feel */
            //<editor-fold
            defaultstate="collapsed" desc=" Look
            and feel setting code (optional) ">
            /* If Nimbus (introduced in
            Java SE 6) is not available, stay
            with the default look and feel.
             * For details see
            http://download.oracle.com/javase/tut
            orial/uiswing/lookandfeel/plaf.html
             */
            try {
                for
                (javax.swing.UIManager.LookAndFeelInfo
                info
                :
                javax.swing.UIManager.getInstalledLo
                okAndFeels()) {
                    if
                    ("Nimbus".equals(info.getName())) {
                        javax.swing.UIManager.setLookAndFeel(
                        info.getClassName());
                        break;
                    }
                }
            } catch
            (ClassNotFoundException ex) {
                java.util.logging.Logger.getLogger(Ma

```

```

inFrame.class.getName()).log(java.util
logging.Level.SEVERE, null, ex);
            } catch
            (InstantiationException ex) {
                java.util.logging.Logger.getLogger(Ma
                inFrame.class.getName()).log(java.util
                logging.Level.SEVERE, null, ex);
            } catch
            (IllegalAccessException ex) {
                java.util.logging.Logger.getLogger(Ma
                inFrame.class.getName()).log(java.util
                logging.Level.SEVERE, null, ex);
            } catch
            (javax.swing.UnsupportedLookAndFeelEx
            ception ex) {
                java.util.logging.Logger.getLogger(Ma
                inFrame.class.getName()).log(java.util
                logging.Level.SEVERE, null, ex);
            }
        }
    }
}
//</editor-fold>

/* Create and display the
form */

java.awt.EventQueue.invokeLater(new
Runnable() {
    public void run() {
        new
        MainFrame().setVisible(true);
    }
});

// Variables declaration - do not
modify
private javax.swing.JButton
buttonCluster;
private javax.swing.ButtonGroup
buttonGroupCluster;
private javax.swing.JButton
buttonMuatDokumen;
private javax.swing.JLabel
iterasi;
private javax.swing.JPanel
jPanel1;
private javax.swing.JPanel
jPanel2;
private javax.swing.JScrollPane
jScrollPane1;
private javax.swing.JLabel
labelClusterMethod;
private javax.swing.JLabel
labelDokumen;
private javax.swing.JLabel
labelIterasi;
private javax.swing.JLabel
labelK;
private javax.swing.JLabel
labelTerm;
private javax.swing.JLabel
labelWaktu;
private javax.swing.JRadioButton
radioButtonFuzzy;
private javax.swing.JRadioButton
radioButtonFuzzySVD;
private javax.swing.JRadioButton
radioButtonKMeans;

```

```

        private javax.swing.JRadioButton
radioButtonKMeansSVD;
        private javax.swing.JRadioButton
radioButtonKMedoids;
        private javax.swing.JRadioButton
radioButtonKMedoidsSVD;
        private javax.swing.JTable
tableCluster;
        private javax.swing.JTextField
textFieldK;
        private javax.swing.JTextField
textFieldPathDokumen;
        private javax.swing.JTextField
textFieldDokumen;
        private javax.swing.JTextField
textFieldTerm;
        private javax.swing.JLabel
waktuEksekusi;
        // End of variables declaration
    }

```

Document.java

```

package preprocessing;

import java.awt.Component;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JFileChooser;
import javax.swing.JTextField;

/**
 *
 * @author User-pc
 */
public class Document {
    private int id;
    private String judul;
    private String isi;
    private TermList termlist;
    private String kategori;
    private int clusternumber;

    public Document(){
        termlist = new TermList();
    }

    public Document(String isi){
        this.isi = isi;
    }

    public Document(int id, String
judul, String isi){
        this.id = id;
        this.judul = judul;
        this.isi = isi;
    }

    public void setClusterNumber(int
clusternumber){
        this.clusternumber =
clusternumber;
    }

    public int getClusterNumber(){
        return clusternumber;
    }
}

```

```

        public void setTermlist(TermList
termlist){
            this.termlist = termlist;
        }

        public TermList getTermList(){
            return termlist;
        }

        public void setKategori(String
kategori){
            this.kategori = kategori;
        }

        public String getKategori(){
            return kategori;
        }

        public void setJudul(String
judul){
            this.judul = judul;
        }

        public String getJudul(){
            return judul;
        }

        public void setIs(String isi){
            this.isi = isi;
        }

        public String getIsi(){
            return isi;
        }
    }
}

```

DocumentReader.java

```

package preprocessing;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileFilter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

/**
 *
 * @author User-pc
 */
public class DocumentReader {
    private String folderpath;
    private List<Document> lstDoc;
    private int nCategory;

    public DocumentReader(String
folderpath){
        this.folderpath = folderpath;
    }

    public int getNumberOfCategory(){
        return nCategory;
    }

    public void readDocuments()
throws FileNotFoundException,
IOException{

```

```

        lstDoc = new ArrayList<>();
        File folders = new
File(folderpath);

System.out.println(folderpath);
        nCategory =
folders.list().length;
        String[] folderNames = new
String[folders.list().length];
        folderNames = folders.list();
        for (String folderName :
folderNames) {

System.out.println(folderName);

System.out.println(folders);
        File folder = new
File(folders + "\\\" + folderName +
"\\");
        List<File> files =
Arrays.asList(folder.listFiles(new
FileFilter(){
            @Override
            public boolean
accept(File f){
                return f.isFile()
&& f.getName().endsWith(".txt");
            }
        }));
        if (files != null){
            BufferedReader in =
null;
            int id=0;
            for (File f : files)
            {
                in = new
BufferedReader(new FileReader(f));

//System.out.println(f.getAbsolutePath(
));
                String isidokumen
= new String();
                String s = null;
                while((s =
in.readLine()) != null){
                    isidokumen =
isidokumen + (s+"\n");
                }
                id++;

System.out.println(f.getAbsolutePath(
));
                Document myDoc
= new Document(id,
f.getAbsolutePath(), sb.toString());
                Document myDoc =
new Document(id, f.getAbsolutePath(),
isidokumen);

myDoc.setKategori(folderName);

lstDoc.add(myDoc);
            }
        }
        System.out.println("\n-----
-----tf idf-----
-\n");
    }
}

```

```

        public List<Document>
getListDocuments(){
            return lstDoc;
        }
    }

Weight.java
package pembobotan;

import
com.sun.istack.internal.logging.Logge
r;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import
sun.util.logging.PlatformLogger;
import preprocessing.Document;
import preprocessing.DocumentReader;
import preprocessing.TermList;
import preprocessing.Tokenizer;

/**
 *
 * @author User-pc
 */
public class Pembobotan {

    private List<Document> lstDoc;
    private DocumentReader docreader;
    private TermList globaltermlist;
    private ArrayList<double[]>
vecsapace;
    private int nCluster;
    public double[][]
hasilPembobotan;

    public Pembobotan(String
folderpath){
        docreader = new
DocumentReader(folderpath);
        readDocument();
    }

    private void readDocument(){
        try{

docreader.readDocuments();
            lstDoc =
docreader.getListDocuments();
            Tokenizer tokenizer = new
Tokenizer();
            for(int i=0;
i<lstDoc.size(); i++){
                Document doc =
lstDoc.get(i);

tokenizer.tokenize(doc);

doc.setTermlist(tokenizer.getCurrentT
okenList());

//System.out.println(doc.getJudul() +
" = " +
Arrays.toString(tokenizer.getCurrentT
okenList().toStringArray()) + "\n");
//
System.out.println(doc.getJudul() + "
= " + doc.getIsi() + "\n");

```

```

        }
        globaltermlist =
tokenizer.getTokenList();
        nCluster =
docreader.getNumberOfCategory();
    } catch (FileNotFoundException
ex) {

    } catch (IOException ex) {

    }
}

public int
getNumberOfClusterDoc() {
    return nCluster;
}

public void doPembobotan() {
    double[][] d = new
double[globaltermlist.getTotalTerm()]
[1stDoc.size()];
    hasilPembobotan = new
double[globaltermlist.getTotalTerm()]
[1stDoc.size()];
    for(int i=0; i<1stDoc.size();
i++) {
        String sdocs[] =
1stDoc.get(i).getTermList().toStringA
rray();

        System.out.println(1stDoc.get(i).getJ
udul() + " : " +
Arrays.toString(1stDoc.get(i).getTerm
List().toStringArray()));
        for(int j=0;
j<hasilPembobotan.length; j++){
            hasilPembobotan[j][i]
=
            tf(sdocs,
globaltermlist.getTermAt(j).getTerm()
) *
            idf(1stDoc,
globaltermlist.getTermAt(j).getTerm()
);
            //
            hasilPembobotan[j][i] = tf(sdocs,
globaltermlist.getTermAt(j).getTerm()
);
            //
            System.out.println(tf(sdocs,
globaltermlist.getTermAt(j).getTerm()
) + " * "
            //
            +
            idf(1stDoc,
globaltermlist.getTermAt(j).getTerm()
) + " = ");

            //System.out.println(globaltermlist.g
etTermAt(j).getTerm()+" :
"+hasilPembobotan[j][i]+"
            =
            d["+j+"["+i+"]");
        }
        System.out.print("\n");
    }
}

System.out.println("-----
----Pembobotan selesai-----
");
//
for(int i=0;
i<globaltermlist.getTotalTerm();
i++){

```

```

//
System.out.println(globaltermlist.get
TermAt(i).getTerm());
//
}

public double[][]
getHasilPembobotan() {
    return hasilPembobotan;
}

public List<Document>
getListDocument() {
    return 1stDoc;
}

public TermList
getGlobalTermList() {
    return globaltermlist;
}

static double tf(String[] doc,
String term) {
    double n = 0;
    for(String s: doc) {
        if(s.equalsIgnoreCase(term)) {
            n++;
        }
    }
    return n;
//
return n/doc.length;
}

static double idf(List<Document>
listdoc, String term) {
    double n = 0;
    //
    System.out.println("n =
"+listdoc.size());
    for(int i=0;
i<listdoc.size(); i++){
        for(int j=0;
j<listdoc.get(i).getTermList().getTot
alTerm(); j++){
            String s =
listdoc.get(i).getTermList().getTermA
t(j).getTerm();
            if(s.equalsIgnoreCase(term)) {
                n++;
                break;
            }
        }
    }
    //
    System.out.println("df =
"+n);
    //
    System.out.println("idf =
"+Math.log10(2));
    return
    Math.log10(listdoc.size()/n);
}
}

```

Tokenizer.java

```

package preprocessing;

/**
 *
 * @author User-pc

```

```

*/
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Tokenizer
{
    TermList lsttoken;
    TermList lsttokencurrent;
    StopWordList stpwlist;
    List<Document>lstDoc;
    Stemming stemming = new
Stemming();

    public Tokenizer()
    {
        lsttoken = new
TermList();
        stpwlist = new
StopWordList();

        stpwlist.loadStopWord();
    }

    public void
setTokenList(TermList lstTokenList)
    {
        lsttoken =
lstTokenList;
    }

    public void tokenize(Document
doc)
    {
        // String[]
t=doc.getIsi().toLowerCase().replaceA
ll("[\\W&[^\s]]", "
").split("[\\W+");
        String[] t=
removePunctuation(doc.getIsi().toLowe
rCase()).split("[\\s+");
        String[]
s=stemming.Stemming(t);
        lsttokencurrent = new
TermList();
        // for(int i=0;
i<t.length; i++)
        // {
        //
        System.out.println("Token: " +t[i]);
        ////
        if(t[i].length()>1)
        {
            ////
            if(!stpwlist.findStopWord(t[i]
))
            {
                ////
                t[i] = t[i].toLowerCase();
                ////
                t[i] = removeReadMark(t[i]);
                ////
                lsttokencurrent.addTerm(new
Term(t[i]));
                ////
                Term tkn =
lsttoken.checkTerm(t[i]);
                ////
                if(tkn==null)
                {

```

```

////
                lsttoken.addTerm(new
Term(t[i]));
                ////
            }
        }
        //
        }
        for(int i=0;
i<s.length; i++){
        //
        System.out.println("Stem: "+s[i]);

        if(s[i].length()>1)
        {
            if(!stpwlist.findStopWord(s[i]
))
            {
                s[i] = s[i].toLowerCase();

                //s[i] =
removePunctuation(s[i]);

                lsttokencurrent.addTerm(new
Term(s[i]));

                Term tkn =
lsttoken.checkTerm(s[i]);

                if(tkn==null)
                {
                    lsttoken.addTerm(new
Term(s[i]));
                }
            }
        }

        private static String
removePunctuation(String isiDokumen){
            // Filter Punctuation
            String rslt = isiDokumen;
            String P =
"[!\"$%&'()*+,-./:;<=>? \\[\\]^_`{-
~{|}...0987654321]";
            return rslt.replaceAll(P,
" ");
        }

        private String
removeReadMark(String text)
        {
            String rm[] = new
String[]{".", ",", "?", "!", ";", "/", "(",
")", "{", "}", "*", "+", "-", "%", "\n"};
            char tb =
text.charAt(text.length()-1);
            boolean foundmark =
false;
            for(int i=0;
i<rm.length; i++)
            {
                if(tb==rm[i].toCharArray()[0])
                {

```

```

        foundmark = true;
        break;
    }
    if (foundmark)
    {
        text
        text.substring(0, text.length()-1);
        // text =
        text.replace(String.valueOf(tb), "
");
    }
    return text;
}

public TermList
getCurrentTokenList()
{
    return
    lsttokencurrent;
}

public TermList getTokenList()
{
    return lsttoken;
}
}

```

Stemmer.java

```

package preprocessing;

import generator.GeneratorKamus;

/**
 *
 * @author User-pc
 */
public class Stemming {

    public Stemming(){

        GeneratorKamus.initKamusKDid();

        // hasil stemming untuk
        mengembalikan kata yang sesuai kamus
        setelah diproses nazief adriani
        public String[] Stemming
        (String[] token){
            String[] Filter = token;
            String HasilStem="";
            String[] Result;
            for(int i=0; i<Filter.length;
            i++){
                HasilStem +=
                Stem(Filter[i])+ "\n";
            }
            Result =
            HasilStem.split("\n");
            return Result;
        }

        //proses stem
        //proses stem
        private String Stem(String
        kata){

            if(GeneratorKamus.getIndexAlphabet().
            contains(getFirstChart(kata))){

```

```

//System.out.println(kata);
        if (!stringMatcher(kata,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(kata)))){
            //
            System.out.println(kata+" masuk ke
            proses stemming");
            return
            this.Hps_derivation_prefixes(this.Hps
            _derivation_suffixes(this.Hps_inflect
            ion_Suffixes(kata)));
        }
        //
        System.out.println(kata+"
        tidak masuk ke proses stemming");
        return kata;
    }

    //untuk menghilangkan inflection
    suffixes( perubahan akhiran)
    private String
    Hps_inflection_Suffixes(String kata){
        //
        System.out.println(kata + "
        mampir ke Hps_inflection_Suffixes");
        String inflection = kata;
        String Hasil= "";
        String Hasil2= "";
        if
        (kata.endsWith("kah")||kata.endsWith(
        "lah")||kata.endsWith("pun")||kata.en
        dsWith("ku")||kata.endsWith("mu")||ka
        ta.endsWith("nya")){
            Hasil =
            kata.replaceAll("(kah|lah|pun|tah|ku|
            mu|nya)$", "");

            if(kata.endsWith("kah")||kata.endsWit
            h("lah")||kata.endsWith("pun")||kata.
            endsWith("tah")){

                if(kata.endsWith("ku")||kata.endsWith
                ("mu")||kata.endsWith("nya")){
                    Hasil2=
                    kata.replaceAll("(ku|mu|nya)$", "");
                    //
                    System.out.println("Hasil2:
                    "+Hasil2);
                    return Hasil2;
                }
            }
            //
            System.out.println("Hasil: "+Hasil);
            return Hasil;
        }
        //
        System.out.println("inflection:
        "+inflection);
        return inflection;
    }

    //untuk menghilangkan derivation
    suffixes (akhiran yang pasti)
    private String
    Hps_derivation_suffixes(String kata){
        //
        System.out.println(kata + "
        mampir ke Hps_derivation_suffixes");
        String der_suffixes= kata;
        String Hasil="";
        String Hasil2="";

```

```

        if
        (kata.endsWith("i") || kata.endsWith("a
n") || kata.endsWith("kan")) {
            Hasil=
            kata.replaceAll("(i|an|kan)$", "");
            if (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) {
                //
                System.out.println("Hasil: "+Hasil);
                return Hasil;
            }
            if (kata.endsWith("k")) {
                Hasil2 =
                kata.replaceAll("(k)$", "");
                if
                (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) {
                    //
                    System.out.println("Hasil2:
"+Hasil2);
                    return Hasil2;
                }
            }
            //
            System.out.println("der_suffixes:
"+der_suffixes);
            return der_suffixes;
        }
        //
        System.out.println("der_suffixes:
"+der_suffixes);
        return der_suffixes;
    }

    //Hapus Derivation prefixes
    private String
Hps_derivation_prefixes(String kata) {
        String der_prefixes = kata;
        String Hasil = "";
        String Hasil2 = "";
        if
        (kata.startsWith("di") || kata.startsWi
th("ke") || kata.startsWith("se")) {
            Hasil =
            kata.replaceAll("(di|ke|se)", "");
            if (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) return Hasil;
            Hasil2 =
            Hps_derivation_suffixes(Hasil);
            if (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;
            if
            (kata.startsWith("diper")) {
                Hasil =
                kata.replaceAll("(^diper)", "");
                if
                (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) return Hasil;
                Hasil2 =
                Hps_derivation_suffixes(Hasil);
                if
                (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;

```

```

                Hasil =
                kata.replaceAll("(^diper)", "r");
                //luluh "r"
                if
                (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) return Hasil;
                Hasil2 =
                Hps_derivation_suffixes(Hasil);
                if
                (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;
            }
        }
        if
        (kata.startsWith("te") || kata.startsWi
th("me") || kata.startsWith("be") || kata
.startsWith("pe")) {
            if(kata.startsWith("te")) {
                if
                (kata.startsWith("terr")) return
der_prefixes;
                if
                (kata.startsWith("ter")) {
                    Hasil =
                    kata.replaceAll("(^ter)", "");
                    if
                    (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) return Hasil;
                    Hasil2 =
                    Hps_derivation_suffixes(Hasil);
                    if
                    (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;
                }
            }
            if
            (kata.startsWith("te")) {
                Hasil =
                kata.replaceAll("(^te)", "");
                if
                (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) return Hasil;
                Hasil2 =
                Hps_derivation_suffixes(Hasil);
                if
                (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;
            }
        }
        if
        (kata.startsWith("me")) {
            if
            (kata.startsWith("menga") || kata.start
sWith("mengi") || kata.startsWith("meng
u") ||
            kata.startsWith("menge") || kata.starts
With("mengo") || kata.startsWith("mengk
")
            || kata.startsWith("mengg") || kata.star
tsWith("mengh") || kata.startsWith("men
gq")) {
                Hasil =
                kata.replaceAll("(^meng)", "");

```



```

        if
        (stringMatcher(Hasil,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil)))) return Hasil;
        Hasil2 =
        Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil2)))) return Hasil2;
        Hasil =
        kata.replaceAll("^ (meng)", "k");
        if
        (stringMatcher(Hasil,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil)))) return Hasil;
        Hasil2 =
        Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil2)))) return Hasil2;
        }
        if
        (kata.startsWith("meny")){
        Hasil =
        kata.replaceAll("^ (meny)", "s");
        if
        (stringMatcher(Hasil,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil)))) return Hasil;
        Hasil2 =
        Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil2)))) return Hasil2;
        }
        if
        (kata.startsWith("mem")){
        if
        (kata.startsWith("memb") || kata.starts
        With("memf") || kata.startsWith("memp")
        || kata.startsWith("memv")){
        Hasil =
        kata.replaceAll("^ (mem)", "");
        if
        (stringMatcher(Hasil,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil)))) return Hasil;
        Hasil2 =
        Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil2)))) return Hasil2;
        Hasil =
        kata.replaceAll("^ (mem)", "p");
        if
        (stringMatcher(Hasil,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil)))) return Hasil;
        Hasil2 =
        Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil2)))) return Hasil2;
        if
        (kata.startsWith("memper")){

```

```

        Hasil =
        kata.replaceAll("^ (memper)", "");
        if
        (stringMatcher(Hasil,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil)))) return Hasil;
        Hasil2 =
        Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil2)))) return Hasil2;
        Hasil =
        kata.replaceAll("^ (memper)", "r");
        if
        (stringMatcher(Hasil,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil)))) return Hasil;
        Hasil2 =
        Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil2)))) return Hasil2;
        }
        if
        (kata.startsWith("mema") || kata.starts
        With("memi") || kata.startsWith("memu")
        || kata.startsWith("meme") || kata.start
        sWith("memo")){
        Hasil =
        kata.replaceAll("^ (mem)", "p");
        if
        (stringMatcher(Hasil,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil)))) return Hasil;
        Hasil2 =
        Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil2)))) return Hasil2;
        }
        if
        (kata.startsWith("men")){
        if
        (kata.startsWith("menc") || kata.starts
        With("mend") || kata.startsWith("menj")
        || kata.startsWith("mens") || kata.start
        sWith("menz")){
        Hasil =
        kata.replaceAll("^ (men)", "");
        if
        (stringMatcher(Hasil,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil)))) return Hasil;
        Hasil2 =
        Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
        GeneratorKamus.getKamusKDid().get(get
        FirstChart(Hasil2)))) return Hasil2;
        }
        if
        (kata.startsWith("mena") || kata.starts
        With("meni") || kata.startsWith("menu")

```

```

||kata.startsWith("mene")||kata.start
sWith("meno")){
    Hasil =
kata.replaceAll("^ (men)", "t");
    if
    (stringMatcher (Hasil,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil)))) return Hasil;
    Hasil2 =
Hps_derivation_suffixes (Hasil);
    if
    (stringMatcher (Hasil2,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil2)))) return Hasil2;
    }
    if
    (kata.startsWith("me")){
        Hasil =
kata.replaceAll("^ (me)", "");
        if
        (stringMatcher (Hasil,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil)))) return Hasil;
        Hasil2 =
Hps_derivation_suffixes (Hasil);
        if
        (stringMatcher (Hasil2,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil2)))) return Hasil2;
        }
        if
        (kata.startsWith("be")){
            if
            (kata.startsWith("ber")){
                if
                (kata.startsWith("bera")||kata.starts
With("beri")||kata.startsWith("beru"))

||kata.startsWith("bere")||kata.start
sWith("bero")){
                    Hasil =
kata.replaceAll("^ (ber)", "");
                    if
                    (stringMatcher (Hasil,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil)))) return Hasil;
                    Hasil2 =
Hps_derivation_suffixes (Hasil);
                    if
                    (stringMatcher (Hasil2,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil2)))) return Hasil2;
                    Hasil =
kata.replaceAll("^ (ber)", "r");
                    if
                    (stringMatcher (Hasil,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil)))) return Hasil;
                    Hasil2 =
Hps_derivation_suffixes (Hasil);
                    if
                    (stringMatcher (Hasil2,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil2)))) return Hasil2;
                    }
                    if
                    (! (kata.startsWith("bera")||kata.star

```

```

tsWith("beri")||kata.startsWith("beru
")
||kata.startsWith("bere")||kata.start
sWith("bero")){
    Hasil =
kata.replaceAll("^ (ber)", "");
    if
    (stringMatcher (Hasil,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil)))) return Hasil;
    Hasil2 =
Hps_derivation_suffixes (Hasil);
    if
    (stringMatcher (Hasil2,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil2)))) return Hasil2;
    }
    return
der_prefixes;
    }
    if
    (kata.startsWith("bek")){
        Hasil =
kata.replaceAll("^ (be)", "");
        if
        (stringMatcher (Hasil,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil)))) return Hasil;
        Hasil2 =
Hps_derivation_suffixes (Hasil);
        if
        (stringMatcher (Hasil2,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil2)))) return Hasil2;
        }
        if
        (kata.startsWith("pe")){
            if
            (kata.startsWith("penga")||kata.start
sWith("pengi")||kata.startsWith("peng
u"))

||kata.startsWith("penge")||kata.star
tsWith("pengo")||kata.startsWith("pen
gk"))

||kata.startsWith("pengg")||kata.star
tsWith("pengh")||kata.startsWith("pen
gq")){
                Hasil =
kata.replaceAll("^ (peng)", "");
                if
                (stringMatcher (Hasil,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil)))) return Hasil;
                Hasil2 =
Hps_derivation_suffixes (Hasil);
                if
                (stringMatcher (Hasil2,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil2)))) return Hasil2;
                Hasil2 =
Hps_derivation_suffixes (Hasil.substring(1));
                if
                (stringMatcher (Hasil2,
GeneratorKamus.getKamusKDid().get (get
FirstChart (Hasil2)))) return Hasil2;

```

```

        Hasil = "k" +
Hasil;
        Hasil2 =
Hps_derivation_suffixes(Hasil);
        if
        (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;
        }
        if
        (kata.startsWith("peny")){
            Hasil =
kata.replaceAll("^ (peny)", "s");
            if
            (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) return Hasil;
            Hasil2 =
Hps_derivation_suffixes(Hasil);
            if
            (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;
            }
            if
            (kata.startsWith("pemb") || kata.starts
With("pemf") || kata.startsWith("pemp")
||
kata.startsWith("pembv") || kata.startsW
ith("pemr") || kata.startsWith("pem")){
                Hasil =
kata.replaceAll("^ (pem)", "");
            if (Hasil.startsWith("belajar")) Hasil=
"ajar";

            if
            (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) return Hasil;
            Hasil2 =
Hps_derivation_suffixes(Hasil);
            if
            (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;
            Hasil =
kata.replaceAll("^ (pem)", "p");
            if
            (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) return Hasil;
            Hasil2 =
Hps_derivation_suffixes(Hasil);
            if
            (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;
            }
            if
            (kata.startsWith("penc") || kata.starts
With("pend") || kata.startsWith("penj")
|| kata.startsWith("pens") || kata.start
sWith("penz")){
                Hasil =
kata.replaceAll("^ (pen)", "");
                if
                (stringMatcher(Hasil,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil)))) return Hasil;
                Hasil2 =
Hps_derivation_suffixes(Hasil);
                if
                (stringMatcher(Hasil2,
GeneratorKamus.getKamusKDid().get(get
FirstChart(Hasil2)))) return Hasil2;
                }
            }
        }
    }
}

```

```

        if
        (kata.startsWith("memper")){
            Hasil
            kata.replaceAll("^ (memper)", "");
            if
            (stringMatcher(Hasil,
            GeneratorKamus.getKamusKDid().get(get
            FirstChart(Hasil)))) return Hasil;
            Hasil2
            Hps_derivation_suffixes(Hasil);
            if
            (stringMatcher(Hasil2,
            GeneratorKamus.getKamusKDid().get(get
            FirstChart(Hasil2)))) return Hasil2;
            Hasil
            kata.replaceAll("^ (memper)", "r");
            if
            (stringMatcher(Hasil,
            GeneratorKamus.getKamusKDid().get(get
            FirstChart(Hasil)))) return Hasil;
            Hasil2
            Hps_derivation_suffixes(Hasil);
            if
            (stringMatcher(Hasil2,
            GeneratorKamus.getKamusKDid().get(get
            FirstChart(Hasil2)))) return Hasil2;
        }
        if
        (! (kata.startsWith("di") || kata.starts
        With("ke") || kata.starts With("se") ||
        kata.starts With("te") || kata.starts Wit
        h("me") || kata.starts With("be") || kata.
        starts With("pe"))){
            return der_prefixes;
        }
        return der_prefixes;
    }

    private Character
    getFirstChart(String teks){
        try {
            //System.out.println(teks
            + " : " + teks.charAt(0));
            return teks.charAt(0);
        } catch (Exception e) {
            return '';
        }
    }

    // method check apakah isi
    stringnya sama
    private static boolean
    stringMatcher(String temp, String[]
    temp2){
        for (int j=0; j<temp2.length;
        j++){
            if(temp.equals(temp2[j])){
                return true;
            }
        }
        return false;
    }
}

```

StopWordList.java

```
package preprocessing;
```

```

/**
 *
 * @author User-pc
 */
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class StopWordList
{
    List<String>lststopword;

    public StopWordList()
    {
        lststopword = new
        ArrayList<String>();
    }

    public void loadStopWord()
    {
        try
        {
            //
            FileReader
            freader = new
            FileReader("folderkamus\\stopwordlist
            .txt");
            //
            int data=0;
            //
            String str="";
            //
            BufferedReader in
            = new
            BufferedReader(new
            FileReader("folderkamus\\stopwordlist
            .txt"));
            //
            StringBuffer sb =
            new StringBuffer();
            //
            String s = null;
            while((s
            =
            in.readLine()) != null){
                sb.append(s);
            }

            lststopword.addAll(Arrays.asList(sb.t
            oString().split(";")));
            //
            while((data=freader.read())!==-
            1)
            //
            {
                if
                (data==59)
                //
                {
                    lststopword.add(str);
                    //
                    //
                    str="";
                    //
                    //
                    //
                    //
                    //
                    str += (char)data;
                    //
                    //
                }
            }
        } catch (FileNotFoundException
        ex)
        {

```

```

System.out.println("File not found");
    }catch(IOException ex)
    {

System.out.println("IOException");
    }
}

    public boolean
findStopWord(String str)
    {
        boolean ada=false;
        for(int i=0;
i<lststopword.size(); i++)
        {

            if(lststopword.get(i).equals(s
tr))
            {

                ada=true;
                break;
            }
        }
        //
        System.out.println(ada);
        return ada;
    }

    public void printStopWord(){
        for(int i=0;
i<lststopword.size(); i++){
            System.out.println(i
+ ". " + lststopword.get(i));
        }
    }
}

```

DictGenerator.java

```

package generator;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 *
 * @author user
 */
public class GeneratorKamus {
    private static Map<Character,
String[]> KamusKDid;

    public static Map<Character,
String[]> getKamusKDid() {
        return KamusKDid;
    }

    private static List<Character>
indexAlphabet;

    public static List<Character>
getIndexAlphabet() {
        return indexAlphabet;
    }
}

```

```

    }

    public static void
initKamusKDid() {
        // init Kamus
        String pathKamusKDid =
"folderkamus\\kamusKDid\\";
        File FileFolder = new
File(pathKamusKDid);
        List<String> Files = new
ArrayList<String>(listFiles(FileFolde
r));
        Map<Character, String[]>
tmpKamusKDid = new
HashMap<Character, String[]>();
        try {
            char ix = 'a';
            for (String File : Files)
            {
                ArrayList<String> tmp
= new ArrayList<String>();
                String line;
                BufferedReader reader
= new BufferedReader(new
FileReader(pathKamusKDid+File));
                while ((line =
reader.readLine()) != null) {
                    tmp.add(line);
                }
                tmpKamusKDid.put(ix,
tmp.toArray(new String[tmp.size()]));
                ix++;
            }
            String[] s = {"`"};
            tmpKamusKDid.put('`', s);
            KamusKDid = tmpKamusKDid;
        }
        catch(IOException e){
        }
        // init Index
        List<Character>
tmpIndexAlphabet = new
ArrayList<Character>();
        for(char c='a'; c<='z'; c++)
        {
            tmpIndexAlphabet.add(c);
        }
        tmpIndexAlphabet.add('`');
        indexAlphabet =
tmpIndexAlphabet;
    }

    // list all Files found in folder
    private static ArrayList<String>
listFiles(final File pathFolder) {
        // * Retrieve all files
        inside a Folder
        ArrayList<String> results =
new ArrayList<String>();
        for (final File fileEntry :
pathFolder.listFiles()) {
            if
(fileEntry.isDirectory()) {
                listFiles(fileEntry);
            }
            else {
                results.add(fileEntry.getName());
            }
        }
    }
}

```

```

        return results;
    }
}

```

Term.java

```

package preprocessing;

import java.util.List;

/**
 *
 * @author User-pc
 */
public class Term {
    private String term;
    private int ada;
    private double idf;
    List<Document> doc;

    public Term(){
    }

    public Term(String term){
        this.term = term;
    }

    public void setTerm(String term){
        this.term = term;
    }

    public String getTerm(){
        return this.term;
    }
}

```

TermList.java

```

package preprocessing;

import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author User-pc
 */
public class TermList {
    List<Term> lstterm;

    public TermList(){
        lstterm = new ArrayList<>();
    }

    public TermList(List<Term>
lstterm){
        this.lstterm = lstterm;
    }

    public void addTerm(Term term){
        lstterm.add(term);
    }

    public Term getTermAt(int idx){
        return lstterm.get(idx);
    }

    public int getTotalTerm(){
        return lstterm.size();
    }
}

```

```

    public String[] toStringArray(){
        String[] temp = new
String[lstterm.size()];
        for(int i=0; i<temp.length;
i++){
            temp[i]
            =
            lstterm.get(i).getTerm();
        }
        return temp;
    }
}

```

```

    public Term checkTerm(String
strterm){
        Term term = null;
        for(int i=0;
i<lstterm.size(); i++){
            if(strterm.equals(lstterm.get(i).getT
erm())){
                term
                =
                lstterm.get(i);
                break;
            }
        }
        return term;
    }
}

```

DimensionReduction.java

```

package pembobotan;

import
cern.colt.matrix.DoubleMatrix2D;
import java.util.Arrays;
import java.util.List;
import preprocessing.Document;
import preprocessing.TermList;

/**
 *
 * @author WIN8
 */
public class ReduksiDimensi {

    private TermList globalTermList;
    private List<Document> lstDoc;
    private double[][] US;
    private double[][] newMatrixTerm;

    public ReduksiDimensi(TermList
globalTermList, List<Document>
lstDoc, double[][] US){
        this.globalTermList
        =
        globalTermList;
        this.lstDoc = lstDoc;
        this.US = US;
        newMatrixTerm
        =
        new
double[lstDoc.size()][lstDoc.size()];
    }

    public void doReduksiDimensi(){
        int termIdx = 0;
        for(int i=0; i<lstDoc.size();
i++){
            System.out.println(lstDoc.get(i).getJ
udul());
            double[][] docTerm = new
double[lstDoc.get(i).getTermList().ge
tTotalTerm()][lstDoc.size()];

```

```

        for(int j=0;
j<lstDoc.get(i).getTermList().getTotalTerm(); j++){
            for(int k=0;
k<globalTermList.getTotalTerm(); k++){
                if(globalTermList.getTermAt(k).getTerm().equals(
lstDoc.get(i).getTermList().getTermAt(j).getTerm())){
                    termIdx = k;
                    break;
                }
            }
            docTerm[j] =
US[termIdx];
System.out.println(Arrays.toString(docTerm[j]));
        }
        System.out.println("\n");
        reduceMatrix(docTerm, i);
System.out.println("\n\n");
    }
}

private void
reduceMatrix(double[][] docTerm, int
idx){
    System.out.println(idx+1);
    double[][] ans = new
double[docTerm[0].length][docTerm.length];
    for(int rows = 0; rows <
docTerm.length; rows++){
        for(int cols = 0; cols <
docTerm[0].length; cols++){
            ans[cols][rows] =
docTerm[rows][cols];
        }
    }
    for(int i=0; i<ans.length;
i++){//2D arrays are arrays of arrays
        double temp = 0;
        for(int j=0;
j<ans[i].length; j++){
            temp += ans[i][j];
        }
        newMatrixTerm[idx][i] =
temp/ans[i].length;
System.out.println(temp/ans[i].length);
//
System.out.println(Arrays.toString(i));
    }
}

public double[][]
getReducedMatrix(){
    return newMatrixTerm;
}
}

```

CosineDistance.java

```

package testta;

import
cern.colt.matrix.DoubleMatrix1D;
import net.sf.javaml.core.Dataset;
import net.sf.javaml.core.Instance;
import
net.sf.javaml.distance.AbstractDistance;
/**
 *
 * @author User-pc
 */
public class CosineDistance {

    public double
calculateDistance(DoubleMatrix1D x,
DoubleMatrix1D y) {
        if (x.size() != y.size()) {
            throw new
RuntimeException("Both instances
should contain the same number of
values.");
        }
        return 1-new
CosineSimilarity().calculateDistance(
x, y);
    }

    public double
getMaximumDistance(Dataset data) {
        //TODO implement
        throw new
RuntimeException("Method
getMaximumDistance is not implemented
in CosineDistance.");
    }

    public double
getMinimumDistance(Dataset data) {
        // TODO implement
        throw new
RuntimeException("Method
getMinimumDistance is not implemented
in CosineDistance.");
    }
}

```

FuzzyCMeans.java

```

package testta;

/**
 *
 * @author User-pc
 */
import
generator.FuzzyRandomPartitionGenerator;
import generator.PartitionGenerator;
import
cern.colt.matrix.DoubleMatrix2D;
import
cern.colt.matrix.doublealgo.Statistic;
import
cern.colt.matrix.doublealgo.Statistic
.VectorVectorFunction;

```



```

// Handle this
awkward case
    u = 1;
    } else {
        double sum = 0;
        for (int j = 0; j <
clusters; j++) {
            // Exact
            analytic solution given by Lagrange
            multipliers
            sum +=
Math.pow(distances.getQuick(i, k) /
distances.getQuick(i, j),
1.0 / (fuzzification - 1.0));
        }
        u = 1 / sum;
    }

    double u0 =
partition.getQuick(i, k);
    partition.setQuick(i,
k, u);

    // StepSize is
max(delta(U))
    if (u - u0 > stepSize)
    {
        stepSize = u - u0;
    }
}
totalIterations = itr;
}
long lEndTime =
System.currentTimeMillis();

    output = lEndTime -
lStartTime;
}

    public DoubleMatrix2D getMeans() {
        return means;
    }

    public DoubleMatrix2D
getPartition() {
        return partition;
    }

    public double getExecTime(){
        return (double) output/1000;
    }

    public int getTotalIterations(){
        return totalIterations+1;
    }

    public double getFuzzification() {
        return fuzzification;
    }

    public void
setFuzzification(double
fuzzification) {
        this.fuzzification =
fuzzification;
    }

    public double getEpsilon() {
        return epsilon;
    }

```

```

}

    public void setEpsilon(double
epsilon) {
        this.epsilon = epsilon;
    }

    public int getMaxIterations() {
        return maxIterations;
    }

    public void setMaxIterations(int
maxIterations) {
        this.maxIterations =
maxIterations;
    }

    public RandomGenerator
getRandomGenerator() {
        return randomGenerator;
    }

    public void
setRandomGenerator(RandomGenerator
random) {
        this.randomGenerator = random;
    }
}

```

Kmeans.java

```

package testta;

/**
 *
 * @author User-pc
 */

import
generator.HardRandomPartitionGenerato
r;
import generator.PartitionGenerator;
import
cern.colt.matrix.DoubleMatrix2D;
import
cern.colt.matrix.doublealgo.Statistic
;
import
cern.colt.matrix.doublealgo.Statistic
.VectorVectorFunction;
import
cern.colt.matrix.impl.DenseDoubleMatr
ix2D;
import
cern.colt.matrix.impl.SparseDoubleMat
rix2D;
import
org.apache.commons.math3.random.Merse
nneTwister;
import
org.apache.commons.math3.random.Rando
mGenerator;

public class KMeans {

    private DoubleMatrix2D means;
    private DoubleMatrix2D partition;
    private int maxIterations = 1000;

```

```

        private RandomGenerator
randomGenerator = new
MersenneTwister();
        private PartitionGenerator
partitionGenerator = new
HardRandomPartitionGenerator();
// private VectorVectorFunction
distanceMeasure = Statistic.EUCLID;
        private CosineDistance
distanceMeasure = new
CosineDistance();
        private DoubleMatrix2D data;
        private DoubleMatrix2D SV;
        int totalIterations;
        int clusters;
        double distance = 0;
        long output;

        public KMeans(DoubleMatrix2D data,
int clusters) {
            this.data = data;
            this.clusters = clusters;
        }

        public KMeans(DoubleMatrix2D
data, DoubleMatrix2D SV, int
clusters) {
            this.data = data;
            this.SV = SV;
            this.clusters = clusters;
        }

        public void cluster() {
            long lStartTime =
System.currentTimeMillis();
            int n = data.rows(); // Number
of features
            int p = data.columns(); //
Dimensions of features

            partition = new
SparseDoubleMatrix2D(n, clusters);

            partitionGenerator.setRandomGenerator
(randomGenerator);

            partitionGenerator.generate(partition
);

            means = new
DenseDoubleMatrix2D(p, clusters);

            boolean changedPartition =
true;

            // Begin the main loop of
alternating optimization
            for (int itr = 0; itr <
maxIterations && changedPartition;
++itr) {
                // Get new prototypes (v)
for each cluster using weighted
median
                for (int k = 0; k <
clusters; k++) {
                    for (int j = 0; j < p;
j++) {
                        double sumWeight = 0;
                        double sumValue = 0;

```

```

                        for (int i = 0; i < n;
i++) {
                            double Um =
partition.getQuick(i, k);
                            sumWeight += Um;
                            sumValue +=
data.getQuick(i, j) * Um;
                        }

                        means.setQuick(j, k,
sumValue / sumWeight);
                    }
                }

                // Calculate distance
measure d:
                DoubleMatrix2D distances =
new DenseDoubleMatrix2D(n, clusters);
                for (int k = 0; k <
clusters; k++) {
                    for (int i = 0; i < n;
i++) {
                        // Euclidean distance
calculation
                        if (SV != null) {
                            distance =
distanceMeasure.calculateDistance(means.viewColumn(k),
SV.viewRow(i));
                        } else {
                            distance =
distanceMeasure.calculateDistance(means.viewColumn(k),
data.viewRow(i));
                        }
                        distances.setQuick(i,
k, distance);
                    }
                }

                // Get new partition matrix
U:
                changedPartition = false;
                for (int i = 0; i < n; i++)
                {
                    double minDistance =
Double.MAX_VALUE;
                    int closestCluster = 0;

                    for (int k = 0; k <
clusters; k++) {
                        // U = 1 for the
closest prototype
                        // U = 0 otherwise

                        if
(distances.getQuick(i, k) <
minDistance) {
                            minDistance =
distances.getQuick(i, k);
                            closestCluster = k;
                        }
                    }

                    if (partition.getQuick(i,
closestCluster) == 0) {
                        changedPartition =
true;

                        for (int k = 0; k <
clusters; k++) {

```

```

partition.setQuick(i, k, (k ==
closestCluster) ? 1 : 0);
    }
    }
    totalIterations = itr;
    }
    long lEndTime =
System.currentTimeMillis();

    output = lEndTime -
lStartTime;
    }

    public DoubleMatrix2D getMeans() {
        return means;
    }

    public DoubleMatrix2D
getPartition() {
        return partition;
    }

    public double getExecTime(){
        return (double) output/1000;
    }

    public int getTotalIterations(){
        return totalIterations+1;
    }

    public int getMaxIterations() {
        return maxIterations;
    }

    public void setMaxIterations(int
maxIterations) {
        this.maxIterations =
maxIterations;
    }

    public RandomGenerator
getRandomGenerator() {
        return randomGenerator;
    }

    public void
setRandomGenerator(RandomGenerator
random) {
        this.randomGenerator = random;
    }
}

Kmedoids.java

package testta;

/**
 *
 * @author User-pc
 */
import cern.colit.list.IntArrayList;
import
cern.colit.matrix.DoubleMatrix2D;
import
cern.colit.matrix.doublealgo.Statistic
;
import
cern.colit.matrix.doublealgo.Statistic
.VectorVectorFunction;

```

```

import
cern.colit.matrix.impl.DenseDoubleMatr
ix2D;
import
cern.colit.matrix.impl.SparseDoubleMat
rix2D;
import
javax.swing.table.DefaultTableModel;
import net.sf.javaml.core.Instance;
import
org.apache.commons.math3.distribution
.UniformIntegerDistribution;
import
org.apache.commons.math3.random.Merse
nneTwister;
import
org.apache.commons.math3.random.Rando
mGenerator;

/**
 * Also known as: PAM (Partitioning
around medoids)
 * It is more robust to noise and
outliers
 * See:
http://en.wikipedia.org/wiki/K-
medoids
 * Also See: Computational Complexity
between K-Means and K-Medoids
Clustering Algorithms
 * for Normal and Uniform
Distributions of Data Points
 * T. Velmurugan and T. Santhanam
 * Department of Computer Science, DG
Vaishnav College, Chennai, India
 * @author tgee
 */
public class KMedoids {

    private DoubleMatrix2D partition;
    private int maxIterations = 1000;
    private RandomGenerator
randomGenerator = new
MersenneTwister();
    private IntArrayList medoids;
    private DoubleMatrix2D
medoidToDisplay;
    private String[][] medoidString;
    private int totalIterations;
    //private VectorVectorFunction
distanceMeasure = Statistic.EUCLID;
    private CosineDistance
distanceMeasure = new
CosineDistance();
    private DoubleMatrix2D data;
    private DoubleMatrix2D SV;
    int clusters;
    double distance = 0;
    double oldDistance = 0;
    double newDistance = 0;
    long output;

    public KMedoids(DoubleMatrix2D
data, int clusters) {
        this.data = data;
        this.clusters = clusters;
    }
}

```

```

    public KMedoids(DoubleMatrix2D
data, DoubleMatrix2D SV, int
clusters) {
    this.data = data;
    this.SV = SV;
    this.clusters = clusters;
}

    public void cluster() {
        long lStartTime =
System.currentTimeMillis();
        int n = data.rows(); // Number
of features
        System.out.println("Baris
data: "+n);
        int p = data.columns(); //
Dimensions of features
        medoidToDisplay = new
DenseDoubleMatrix2D(clusters, 1);
        medoidString = new
String[clusters][1];

        partition = new
SparseDoubleMatrix2D(n, clusters);
        medoids = new
IntArrayList(clusters);

        IntArrayList randomOrdering =
new IntArrayList(n);
        for (int i = 0; i < n; ++i) {
            randomOrdering.setQuick(i,
i);
        }

        // Choose the medoids by
shuffling the data
        System.out.println("Choose the
medoids by shuffling the data");
        for (int i = 0; i < clusters;
++i) {
            // k is the index of the
remaining possibilities
            UniformIntegerDistribution
uniform = new
UniformIntegerDistribution(randomGene
rator, i, n);
            int k = uniform.sample();

            // Swap x(i) and x(k)
            int medoid =
randomOrdering.getQuick(k);
            randomOrdering.setQuick(k,
i);

            medoids.setQuick(i, medoid);
            System.out.println("medoid
ke-"+(int)(i+1) + ": D" +
(int)(medoid+1));
            medoidToDisplay.set(i, 0,
medoid+1);
            medoidString[i][0] = "D"
+(int)(medoid+1);
        }

        boolean changedMedoid = true;

        // Begin the main loop of
alternating optimization
        System.out.println("\nBegin
the main loop of alternating
optimization");

        for (int itr = 0; itr <
maxIterations && changedMedoid;
++itr) {
            // Get new partition matrix
U by
            // assigning each object to
the nearest medoid
            for (int i = 0; i < n; i++)
            {
                double minDistance =
Double.MAX_VALUE;
                int closestCluster = 0;

                for (int k = 0; k <
clusters; k++) {
                    // U = 1 for the
closest medoid
                    // U = 0 otherwise
                    int medoid =
medoids.getQuick(k);
                    // double distance =
distanceMeasure.apply(data.viewRow(me
doid), data.viewRow(i));

                    System.out.println("medoid:
D"+(int)(medoid+1) + " dokumen:
D"+(int)(i+1));
                    if (SV != null) {
                        distance =
distanceMeasure.calculateDistance(dat
a.viewRow(medoid), SV.viewRow(i));
                    } else {
                        distance =
distanceMeasure.calculateDistance(dat
a.viewRow(medoid), data.viewRow(i));
                    }
                    if (distance <
minDistance) {
                        minDistance =
distance;
                        closestCluster = k;
                    }
                }

                if (partition.getQuick(i,
closestCluster) == 0) {
                    for (int k = 0; k <
clusters; k++) {
                        partition.setQuick(i, k, (k ==
closestCluster) ? 1 : 0);
                    }
                }
            }

            System.out.println(partition);

            // Try to find a better set
of medoids
            //System.out.println("\nTry
to find a better set of medoids");
            changedMedoid = false;
            for (int k = 0; k <
clusters; k++) {
                System.out.println("\nTry
to find a better set of medoids");
                // For each non-medoid in
the cluster

```

```

        int medoid =
medoids.getQuick(k);
        for (int i = 0; i < n;
++i) {
            int bestMedoid =
medoid;
            double lowestCostDelta
= 0;
            if (i != medoid &&
partition.getQuick(i, k) > 0) {
                // Calculate the
change in cost by swapping this
configuration
                int costDelta = 0;
                for (int j = 0; j <
n; ++j) {
                    if
(partition.getQuick(j, k) > 0) {
//System.out.println("medoid:
D"+(int)(medoid+1) + " dokumen:
D"+(int)(i+1));
                        if(SV !=
null){
oldDistance =
distanceMeasure.calculateDistance(dat
a.viewRow(medoid), SV.viewRow(j));

newDistance =
distanceMeasure.calculateDistance(dat
a.viewRow(i), SV.viewRow(j));
                        }else{
oldDistance =
distanceMeasure.calculateDistance(dat
a.viewRow(medoid), data.viewRow(j));

newDistance =
distanceMeasure.calculateDistance(dat
a.viewRow(i), data.viewRow(j));
                        }

                        System.out.println("dokumen:
D"+(int)(medoid+1) + " dokumen:
D"+(int)(j+1)+" : "+oldDistance);

                        System.out.println("dokumen:
D"+(int)(i+1) + " dokumen:
D"+(int)(j+1)+" : "+newDistance);
                        costDelta +=
newDistance - oldDistance;
                    }
                }

                if (costDelta <
lowestCostDelta) {
                    bestMedoid = i;
                    lowestCostDelta
= costDelta;
                }

                if (bestMedoid !=
medoid) {
medoids.setQuick(k, bestMedoid);
                    changedMedoid =
true;

                    System.out.println("Terjadi perubahan

```

```

medoid dari "+(int)(k+1)+" ke
"+(int)(bestMedoid+1)+"\n");
                }
            }else{

                System.out.println("Tidak terjadi
perubahan medoid\n");
            }
        }
    }

    totalIterations = itr;
}

    System.out.println("Medoids
here: "+medoids);
    long lEndTime =
System.currentTimeMillis();

    output = lEndTime -
lStartTime;
}

    public IntArrayList getMedoids() {
        return medoids;
    }

    public DoubleMatrix2D
getMedoidsToDisplay(){
        return medoidToDisplay;
    }

    public String[][]
getMedoidString(){
        return medoidString;
    }

    public double getExecTime(){
        return (double) output/1000;
    }

    public DoubleMatrix2D
getPartition() {
        return partition;
    }

    public int getTotalIterations(){
        return totalIterations+1;
    }

    public int getMaxIterations() {
        return maxIterations;
    }

    public void setMaxIterations(int
maxIterations) {
        this.maxIterations =
maxIterations;
    }

    public RandomGenerator
getRandomGenerator() {
        return randomGenerator;
    }

    public void
setRandomGenerator(RandomGenerator
random) {
        this.randomGenerator = random;
    }
}

```

PartitionGenerator.java

```
package generator;

/**
 *
 * @author User-pc
 */
import
cern.colt.matrix.DoubleMatrix2D;
import
org.apache.commons.math3.random.Rand
omGenerator;
public interface PartitionGenerator {

    void        generate(DoubleMatrix2D
partition);

    void
setRandomGenerator(RandomGenerator
randomGenerator);
}
```

FuzzyRandomPartitionGenerator.java

```
package generator;

/**
 *
 * @author User-pc
 */
import
cern.colt.matrix.DoubleMatrix2D;
import
org.apache.commons.math3.random.Merse
nneTwister;
import
org.apache.commons.math3.random.Rando
mGenerator;

public                                class
FuzzyRandomPartitionGenerator
implements PartitionGenerator {

    private                            RandomGenerator
randomGenerator;

    public
FuzzyRandomPartitionGenerator() {
        randomGenerator        =        new
MersenneTwister();
    }

    @Override
    public                                void
generate(DoubleMatrix2D partition) {
        for (int i = 0; i <
partition.rows(); ++i) {
            // Randomise
            double sum = 0;
            for (int k = 0; k <
partition.columns(); ++k) {
                double u =
randomGenerator.nextDouble();
                partition.setQuick(i, k,
u);
                sum += u;
            }
            // Normalise the weights
```

```
        for (int k = 0; k <
partition.columns(); ++k) {
            partition.setQuick(i, k,
partition.getQuick(i, k) / sum);
        }
    }
}

public                                RandomGenerator
getRandomGenerator() {
    return randomGenerator;
}

@Override
public                                void
setRandomGenerator(RandomGenerator
random) {
    this.randomGenerator = random;
}
}
```

HardRandomGenerator.java

```
package generator;

/**
 *
 * @author User-pc
 */
import
cern.colt.matrix.DoubleMatrix2D;
import
org.apache.commons.math3.distribution
.UniformIntegerDistribution;
import
org.apache.commons.math3.random.Merse
nneTwister;
import
org.apache.commons.math3.random.Rando
mGenerator;

public                                class
HardRandomPartitionGenerator
implements PartitionGenerator {

    private                            RandomGenerator
randomGenerator;

    public
HardRandomPartitionGenerator() {
        randomGenerator        =        new
MersenneTwister();
    }

    @Override
    public                                void
generate(DoubleMatrix2D partition) {
        // Initialise U randomly
        partition.assign(0);

        UniformIntegerDistribution
uniform = new
UniformIntegerDistribution(randomGene
rator, 0, partition.columns() - 1);

        for (int i = 0; i <
partition.rows(); ++i)
        {
            // Randomise
            int k = uniform.sample();
            partition.setQuick(i, k, 1);
        }
    }
}
```

```
    }  
}  
  
    public RandomGenerator  
getRandomGenerator() {  
    return randomGenerator;  
}  
    @Override  
    public void  
setRandomGenerator(RandomGenerator  
random) {  
        this.randomGenerator = random;  
    }  
}
```

PENGARUH SINGULAR VALUE DECOMPOSITION TERHADAP METODE – METODE CLUSTERING

Novita Hidayati ¹⁾ Muhammad Ihsan Jambak ²⁾ Danny Matthew Saputra ³⁾

- ¹⁾ Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya
Palembang, Indonesia
email : novitahidayati52@gmail.com
- ²⁾ Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya
Palembang, Indonesia
email : mihsanjambak@gmail.com
- ³⁾ Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Sriwijaya
Palembang, Indonesia
email : danny.saputra@gmail.com

ABSTRACT

Documents that have a large scale of attributes or dimension can be an obstacle in a clustering process. Most clustering algorithms are good at handling low-dimensional data. Finding clusters of data objects in a high-dimensional space is challenging, because data can be very sparse and highly skewed. Therefore, this research implements Singular Value Decomposition (SVD) to reduce the high dimension and analyses its impact on clustering methods k-means, k-medoids, and fuzzy c-means. This research shows that combining SVD with k-means and k-medoids increases both accuracy result compared with clustering without dimension reduction. The combination of SVD and k-means and k-medoids increases accuracy result by 10-11% compared to clustering without dimensional reduction. The execution time is also proven to be faster. On the contrary, the combination between SVD and fuzzy c-means shows a lower accuracy result compared with clustering without dimension reduction.

Key words

dimension reduction, singular value decomposition, k-means clustering, k-medoids clustering, fuzzy c-means clustering.

1. Pendahuluan

Dokumen yang disajikan dalam jumlah dan jenis yang banyak memerlukan proses clustering untuk mempermudah pencarian informasi di bidang tertentu [1]. Namun jika dalam sebuah dokumen mengandung banyak atribut atau *term*, dimana setiap *term* direpresentasikan sebagai satu dimensi, maka dokumen tersebut termasuk data berdimensi tinggi. Kebanyakan algoritma clustering baik dalam menangani data berdimensi rendah, sehingga untuk menemukan cluster dari objek data berdimensi tinggi adalah sebuah tantangan karena data berdimensi tinggi cenderung memiliki *noise*, *sparse*, dan bersifat *skew* [2].

Untuk meningkatkan akurasi hasil clustering data berdimensi tinggi dapat dilakukan reduksi dimensi [2]. Reduksi dimensi merupakan teknik dalam *text mining* dengan mengurangi dimensi sehingga clustering memproses data dengan jumlah fitur yang telah berkurang. *Singular Value Decomposition (SVD)* adalah metode populer untuk reduksi dimensi. Untuk data tekstual, metode ini juga dikenal dengan Latent Semantic Analysis (LSA) [3][4].

Penelitian ini berfokus untuk melihat pengaruh SVD terhadap akurasi dan waktu eksekusi hasil clustering dari

tiga metode clustering yaitu k-means, k-medoids, dan fuzzy c-means dalam mengelompokkan dokumen berdimensi tinggi.

2. METODOLOGI PENELITIAN

2.1 Data

Data uji penelitian yang digunakan adalah jurnal berbahasa Indonesia yang diunduh melalui situs Indonesian Publication Index (id.portalgaruda.org) yang disimpan dalam berkas ASCII (*.txt). Data tersebut memiliki lima kelas alami berdasarkan topik penelitiannya yaitu jurnal ekonomi, hukum, komputer, pendidikan, dan peternakan. Masing – masing topik jurnal berjumlah 20, sehingga total data uji sebanyak 100.

2.2 Prapengolahan

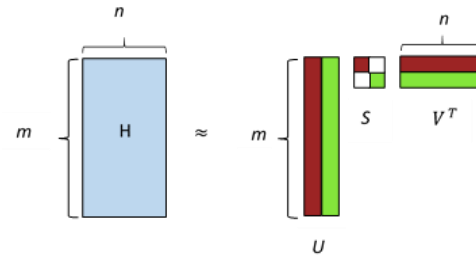
Tahapan prapengolahan teks bertujuan meningkatkan kualitas fitur dan mengurangi kesulitan dalam proses *text mining*. Tahapan yang dilakukan pada data uji penelitian adalah *case folding*, *tokenizing*, *stop words removal*, dan *stemming*. Selanjutnya, term dari data uji penelitian diberi bobot menggunakan skema pembobotan TF-IDF dengan menggunakan persamaan :

$$TF - IDF = \ln(TF) \times \ln(IDF) \quad \dots(1)$$

$$TF - IDF = TF \times IDF \quad \dots(2)$$

2.4 Reduksi Dimensi

Singular Value Decomposition yaitu suatu bentuk analisa faktor pada matriks. Pada *SVD* matriks memuat frekuensi kemunculan kata kunci didekomposisi menjadi tiga komponen matriks [5]. Komponen matriks pertama (U) mendeskripsikan entitas baris sebagai vektor orthogonal matriks. Komponen matriks kedua (S) berupa matriks diagonal yang memuat nilai skalar matriks. Dan komponen yang ketiga (V) adalah matriks entitas kolom sebagai vektor orthogonal matriks. Dalam penelitian ini, *singular value decomposition* dari matriks hasil pembobotan akan digunakan dalam pendekatan transformasi linier. *SVD* pada dasarnya untuk melakukan estimasi *rank* dari matriks. Gambar II-1 adalah representasi dari *Singular Value Decomposition* [2].



Gambar 1 Dekomposisi *Singular Value Decomposition*

Jika diketahui matriks H dengan dimensi $m \times n$, dimana nilai $m \geq n$ dan $\text{rank}(H) = r$ maka *singular value decomposition* dari H , didefinisikan melalui persamaan,

$$H = USV^T \quad \dots(3)$$

Keterangan:

H : matriks hasil pembobotan

U : vektor singular kiri

V : vektor singular kanan

T : transpose

S : nilai singular

m : term

n : dokumen

r : rank

dimana,

$$U^T U = V^T V = I_n \quad \dots(4)$$

dan memenuhi kondisi,

$$S = \text{diag}(\sigma_1, \dots, \sigma_n) \quad \dots(5)$$

dimana,

$$\sigma_i > 0 \text{ untuk } 1 \leq i \leq r$$

$$\dots(6)$$

$$\sigma_j = 0 \text{ untuk } j \geq r + 1$$

$$\dots(7)$$

Kolom pertama dari matriks U dan V mendefinisikan vektor eigen orthonormal yang bersesuaian dengan r nilai vektor eigen tidak-nol dari matriks HH^T dan $H^T H$ berturut-turut. Kolom dari matriks U dan V berisi vektor, masing-masing disebut vektor singular kiri dan kanan. Nilai singular dari H merupakan elemen diagonal dari matriks S , dimana nilai singular didapat dari akar pangkat dua dari nilai atribut dari sejumlah n nilai eigen dari HH^T .

Setelah memperoleh tiga matriks dari proses *SVD*, proses berikutnya untuk mereduksi dimensi dari matriks

adalah dengan mengurangi dimensi dari matriks S yang berupa matriks diagonal. Nilai skalar matriks terkecil milik matriks S akan dihilangkan sehingga satu kolom dari matriks U dan satu baris matriks V juga hilang mengikuti letak nilai singular matriks S . Selanjutnya, dilakukan perkalian matriks baru U dan S sehingga menghasilkan matriks H baru. Matriks H baru tersebut akan memuat *principal component* data. Algoritma SVD dibahas secara mendetail pada [6].

Matriks U , S , dan V hasil SVD kemudian digunakan sebagai masukan untuk Latent Semantic Analysis (LSA). LSA adalah sebuah teknik merepresentasikan suatu teks dalam bentuk matriks yang berisi frekuensi atau banyaknya kemunculan kata dalam dokumen. Setiap baris pada matriks tersebut berupa frekuensi kata dimasing-masing kolom kalimat. LSA membutuhkan proses *Singular Value Decomposition* (SVD) terhadap matriks yang dihasilkan dengan baris kata dan kolom kalimat tersebut. SVD dilakukan sebagai proses pendekomposisian matriks dengan maksud sebagai proses pengurangan noise, sehingga matriks yang diperoleh merupakan matriks yang bersih dari noise dimension.

Secara garis besar, algoritma kombinasi SVD dan metode clustering adalah sebagai berikut [7]:

1. Reduksi dimensi matriks hasil pembobotan seluruh dokumen menggunakan SVD. Sehingga dihasilkan matriks U , S , dan V^T .
2. Lakukan perkalian antara matriks U dan S (matriks H).
3. Lakukan perkalian antara matriks S dan V^T , kemudian matriks hasil perkalian ditranspose (matriks J).
4. Lakukan proses clustering dengan matriks H dan J yang telah direduksi sebagai masukannya. Untuk perhitungan kemiripan antar dokumen menggunakan *cosine distance*.

Berikut contoh perhitungan LSA menggunakan kalimat yang telah dilakukan prapengolahan:

Kal 1: novita yumi putus hengkang usaha

Kal 2: yumi putus undur usaha komputer

Kal 3: novita yumi jabat wakil direktur komputer undur diri

Kal 4: yumi umum facebook tinggal posisi

Tabel 1 menunjukkan *term frequency* masing – masing dokumen.

Tabel 1 *Term Frequency* Kata

Term	Kal 1	Kal 2	Kal 3	Kal 4
novita	1	0	1	0
yumi	1	1	1	1
putus	1	1	0	0
hengkang	1	0	0	0
usaha	1	1	0	0
undur	0	1	1	0
komputer	0	1	1	0
jabat	0	0	1	0

wakil	0	0	1	0
direktur	0	0	1	0
facebook	0	0	0	1
tinggal	0	0	0	1
posisi	0	0	0	1

Selanjutnya term frequency kata dipecah menggunakan SVD menjadi matriks U , S , dan V^T . Tabel 2 menunjukkan matriks U , tabel 3 menunjukkan matriks S , dan tabel 4 menunjukkan matriks S .

Tabel 2 Matriks U

-0.332	-0.085	-0.102	0.604
-0.561	0.183	0.262	0.019
-0.307	0.381	-0.285	-0.126
-0.144	0.275	-0.162	0.441
-0.307	0.381	-0.285	-0.126
-0.351	-0.253	-0.064	-0.406
-0.351	-0.253	-0.064	-0.406
-0.188	-0.359	0.059	0.162
-0.188	-0.359	0.059	0.162
-0.188	-0.359	0.059	0.162
-0.066	0.162	0.488	-0.017
-0.066	0.162	0.488	-0.017
-0.066	0.162	0.488	-0.017

Tabel 3 Matriks S

3.389	0	0	0
0	2.033	0	0
0	0	1.880	0
0	0	0	1.356

Tabel 3 Matriks V^T

Kal 1	Kal 2	Kal 3	Kal 4
-0.487	-0.554	-0.637	-0.224
0.558	0.216	-0.730	0.329
-0.305	-0.232	0.112	0.917
0.599	-0.770	0.220	-0.023

Kata-kata dalam matriks term frequency diwakili oleh vektor-vektor baris dari U sedangkan kalimat diwakili oleh kolom-kolom dari vektor V^T . Untuk menskalakan kedua vektor dilakukan pengkalian nilai - nilai singular pada matriks S . Kedua matriks hasil perkalian inilah yang akan menjadi masukan pada proses clustering. Dengan demikian representasi dari kata-kata diperoleh dari vektor baris $U \times S$ adalah sebagai berikut:

$$\begin{aligned} \text{novita} &= \begin{bmatrix} -1.124 \\ -0.172 \\ -0.193 \\ 0.819 \end{bmatrix}, \text{ yumi} = \begin{bmatrix} -1.902 \\ 0.373 \\ 0.493 \\ 0.026 \end{bmatrix}, \\ \text{putus} &= \begin{bmatrix} -1.041 \\ 0.774 \\ -0.536 \\ -0.171 \end{bmatrix}, \text{ hengkang} = \begin{bmatrix} -0.487 \\ 0.558 \\ -0.305 \\ 0.599 \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
\text{usaha} &= \begin{bmatrix} -1.041 \\ 0.774 \\ -0.536 \\ -0.171 \end{bmatrix}, \text{undur} = \begin{bmatrix} -1.191 \\ -0.514 \\ -0.120 \\ -0.550 \end{bmatrix}, \\
\text{komputer} &= \begin{bmatrix} -1.191 \\ -0.514 \\ -0.120 \\ -0.550 \end{bmatrix}, \text{jabat} = \begin{bmatrix} -0.637 \\ -0.730 \\ 0.112 \\ 0.220 \end{bmatrix}, \\
\text{wakil} &= \begin{bmatrix} -0.637 \\ -0.730 \\ 0.112 \\ 0.220 \end{bmatrix}, \text{direktur} = \begin{bmatrix} -0.637 \\ -0.730 \\ 0.112 \\ 0.220 \end{bmatrix}, \\
\text{facebook} &= \begin{bmatrix} -0.224 \\ 0.329 \\ 0.917 \\ -0.023 \end{bmatrix}, \text{tinggal} = \begin{bmatrix} -0.224 \\ 0.329 \\ 0.917 \\ -0.023 \end{bmatrix}, \\
\text{posisi} &= \begin{bmatrix} -0.224 \\ 0.329 \\ 0.917 \\ -0.023 \end{bmatrix}
\end{aligned}$$

dan representasi kalimat diperoleh dari vektor kolom $S \times V^t$ adalah :

$$\begin{aligned}
\text{Kal 1} &= \begin{bmatrix} -1.651 \\ 1.135 \\ -0.573 \\ 0.812 \end{bmatrix}, \text{kal 2} = \begin{bmatrix} -1.878 \\ 0.439 \\ -0.436 \\ -1.044 \end{bmatrix}, \text{kal 3} = \begin{bmatrix} -2.159 \\ -1.485 \\ 0.210 \\ 0.298 \end{bmatrix}, \\
\text{kal 4} &= \begin{bmatrix} -0.759 \\ 0.669 \\ 1.725 \\ -0.031 \end{bmatrix}
\end{aligned}$$

Untuk pengukuran nilai kemiripan/similarity pada clustering digunakan cosine distance. Cosine distance adalah perhitungan kemiripan berdasarkan besar sudut kosinus antara dua vektor. Berdasarkan kosinus sudut antara dua vektor tersebut, maka nilai similarity memiliki rentang antara 0-1. Semakin nilai similarity mendekati 1, maka semakin mirip pula pasangan kalimat tersebut. Berikut contoh perhitungan tingkat kemiripan antara kalimat ke-1 dan kalimat ke-2:

Kal 1: novita yumi putus hengkang usaha
Kal 2: yumi putus undur usaha komputer
Kal 1 =

$$\begin{aligned}
&\frac{\begin{bmatrix} -1.124 \\ -0.172 \\ -0.193 \\ 0.819 \end{bmatrix} + \begin{bmatrix} -1.902 \\ 0.373 \\ 0.493 \\ 0.026 \end{bmatrix} + \begin{bmatrix} -1.041 \\ 0.774 \\ -0.536 \\ -0.171 \end{bmatrix} + \begin{bmatrix} -0.487 \\ 0.558 \\ -0.305 \\ 0.599 \end{bmatrix} + \begin{bmatrix} -1.041 \\ 0.774 \\ -0.536 \\ -0.171 \end{bmatrix}}{5} \\
&= \begin{bmatrix} -1.119 \\ 0.462 \\ -0.215 \\ 0.220 \end{bmatrix} \\
&\text{Kal 2} = \begin{bmatrix} -1.878 \\ 0.439 \\ -0.436 \\ -1.044 \end{bmatrix} \\
&|\text{Kal}_1| = \sqrt{-1.119^2 + 0.462^2 + -0.215^2 + 0.220^2} = 1.249 \\
&|\text{Kal}_2| = \sqrt{-1.878^2 + 0.439^2 + -0.436^2 + -1.044^2} = 2.236
\end{aligned}$$

$$\begin{aligned}
\text{Sim}(\text{Kal}_1, \text{Kal}_2) &= \frac{\text{Kal}_1 \cdot \text{Kal}_2}{|\text{Kal}_1| |\text{Kal}_2|} = \frac{\begin{bmatrix} -1.119 \\ 0.462 \\ -0.215 \\ 0.220 \end{bmatrix} \cdot \begin{bmatrix} -1.878 \\ 0.439 \\ -0.436 \\ -1.044 \end{bmatrix}}{1.249 \times 2.236} = \frac{2.168}{2.792} = 0.776
\end{aligned}$$

Berdasarkan perhitungan di atas, didapatkan nilai kemiripan antara kalimat ke – 1 dan ke – 2 sebesar 0,776.

2.4 K-Means

K – means adalah metode clustering partisi berbasis titik pusat (centroid) yang paling sering digunakan karena kesederhanaannya. Proses penentuan titik tengah dan penempatan data dalam cluster diulangi sampai nilai titik tengah tidak berubah lagi.

Algoritma k – means sebagai berikut [2]:

1. Tentukan jumlah cluster (k) sebagai banyaknya cluster yang ingin dibentuk;
2. Bangkitkan k *centroid* (titik pusat) awal secara random;
3. Untuk setiap data, temukan pusat cluster terdekat. Sehingga setiap pusat cluster memiliki sebuah subset dari dataset;
4. Untuk masing – masing cluster k, temukan pusat luasan cluster, perbaharui lokasi dari setiap pusat cluster ke nilai baru dari pusat luasan;
5. Ulangi langkah ke – 3 dan ke – 5 hingga data – data pada tiap cluster menjadi terpusat atau selesai.

Titik *centroid* akan menentukan keberhasilan dalam algoritma k – means. Apabila nilai random titik *centroid* yang dipilih pada awal tidak tepat, maka hasil cluster yang didapat tidak maksimal.

2.5 K-Medoids

K – medoids adalah metode clustering partisi berbasis objek yang merupakan peningkatan dari metode k – means. Dibandingkan dengan k – means, k – medoids lebih tahan terhadap *noise* karena pengelompokan objek data ke dalam *k* cluster dengan cara meminimalisir *absolute error*.

Algoritma k – medoids sebagai berikut [8]:

1. Tentukan jumlah cluster (*k*) sebagai banyaknya cluster yang ingin dibentuk;
2. Bangkitkan *medoid* awal secara random;
3. Untuk setiap data, temukan pusat cluster terdekat. Sehingga setiap pusat cluster memiliki sebuah subset dari dataset;
4. Secara acak pilih salah satu dari non – medoid yang disebut O_{random} ;
5. Hitung selisih antara total kemiripan *medoid* awal (O_j) dengan total kemiripan O_{random} . S = total kemiripan sebelumnya (O_j) - total kemiripan terbaru (O_{random});
6. Jika $S < 0$ maka terjadi pertukaran medoid, tukar O_{random} menjadi O_j dengan kata lain O_{random} menjadi *medoid*;
6. Ulangi langkah 3 – 6 hingga data – data pada tiap cluster menjadi terpusat atau selesai.

2.6 Fuzzy C-Means

Fuzzy C-Means adalah suatu metode clustering partisi yang mana keberadaannya tiap-tiap titik data dalam suatu *cluster* ditentukan oleh derajat keanggotaan. Kekhasan lainnya dari metode fuzzy c-means clustering adalah satu data dapat masuk ke lebih dari satu buah cluster.

Algoritma fuzzy c – means sebagai berikut [9]:

1. Tentukan banyak kelompok (*c*), *fuzzifier* (*m*), maksimum iterasi (*MaxIter*), perubahan nilai fungsi objektif terkecil yang diharapkan (ϵ), fungsi objektif awal ($P_0 = 0$), dan iterasi awal ($t = 1$);
2. Bangkitkan bilangan random U_{ik} dengan *i* merupakan banyak data dan *k* merupakan banyak kelompok sebagai elemen-elemen awal matriks keanggotaan awal *U*;
3. Hitung pusat kelompok ke-*i* dengan persamaan :

$$p_i = \frac{\sum_{k=1}^N (u_{ik})^m x_k}{\sum_{k=1}^N (u_{ik})^m} \quad \dots(8)$$

dimana u_{ik} nilai keanggotaan objek ke-*k* dengan pusat kelompok ke-*i*, x_k adalah objek data ke-*k*, *N* adalah banyaknya objek penelitian, dan *m* adalah *fuzzifier*.

4. Hitung fungsi objektif pada iterasi ke-*t* dengan persamaan :

$$J(P, U, X, c, m) = \sum_{i=1}^c \sum_{k=1}^N (u_{ik})^m d^2_{ik}(x_k, p_i) \quad \dots(9)$$

dimana *c* adalah banyak kelompok yang diinginkan, *N* adalah banyak objek penelitian, u_{ik} adalah nilai keanggotaan objek ke-*k* pada kelompok ke-*i* yang merupakan bagian dari matriks *U*, *m* adalah *fuzzifier*, dan $d^2_{ik}(x_k, p_i)$ adalah jarak antara vektor pengamatan ke-*k* dengan pusat kelompok ke-*i*.

5. Hitung perubahan matriks keanggotaan dengan persamaan :

$$u_{ik} = \frac{1}{\sum_{i=1}^c \left(\frac{d^i_k}{d^2_{jk}} \right)^{\frac{1}{m-1}}} \quad \dots(10)$$

dimana u_{ik} adalah nilai keanggotaan objek ke-*k* dengan pusat kelompok ke-*i*, d^i_k adalah jarak antara objek ke-*k* dengan pusat kelompok ke-*i*, d^2_{jk} adalah jarak antara objek ke-*k* dengan pusat kelompok ke-*j*, dan *m* adalah *fuzzifier*.

6. Cek kondisi berhenti
 - Jika $|J_t - J_{t-1}| < \epsilon$ atau $t > \text{MaxIter}$ maka berhenti;
 - Jika tidak, maka $t = t + 1$, Ulangi langkah 3 – 5 hingga data – data pada cluster menjadi terpusat atau selesai.

3. Hasil Percobaan

Percobaan dilakukan sebanyak 10 kali pada masing – masing metode clustering k-means, k- medoids, dan fuzzy c-means dengan pertimbangan bahwa pusat cluster yang diambil secara acak di awal akan mempengaruhi hasil setiap clustering. Jumlah cluster yang dibentuk adalah lima sesuai dengan banyaknya kelas murni jurnal. Hasil prapengolahan data uji penelitian menunjukkan jumlah kata kunci atau *term* dari keseluruhan sebanyak 8088 kata. Pada tabel 4 , 5, 6, 7, 8, dan 9 masing – masing huruf ‘E’ , ‘H’ , ‘K’ , ‘PN’ , ‘PT’ mewakili inisial topik ekonomi, hukum, komputer, pertanian dan peternakan secara berurutan.

Tabel 4 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan. clustering k-means. Pada percobaan ke – 5, pada cluster 1 topik jurnal komputer dominan terhadap topik lain dengan jumlah dokumen sebanyak 17. Di cluster 2 topik jurnal hukum dominan dengan jumlah dokumen sebanyak 19. Di cluster 3 terdapat 15 dokumen jurnal ekonomi dan 18 dokumen jurnal pendidikan, namun topik jurnal pendidikan tetap dominan. Di cluster 4, tidak ada topik jurnal yang dominan, sehingga tidak dapat dipastikan cluster yang terbentuk merupakan cluster topik ekonomi, komputer, pendidikan, ataupun peternakan. Sedangkan pada cluster 5, topik peternakan dominan dengan jumlah dokumen sebanyak 6.

Berbeda dengan percobaan ke-7, setiap cluster cenderung memiliki satu topik jurnal yang dominan. Di cluster 1, topik jurnal ekonomi dominan terhadap topik jurnal lain dengan 15 dokumen. Di cluster 2, topik jurnal peternakan dominan dengan 8 dokumen. Di cluster 3, topik jurnal pendidikan dominan dengan 8 dokumen. Di cluster 3, topik jurnal hukum dominan dengan 19 dokumen dan di cluster 5 topik jurnal komputer dominan dengan 15 dokumen. Sehingga masing – masing cluster dapat dinamakan berdasarkan topik jurnal yang dominan di dalamnya.

Tabel 4 Hasil Clustering K-Means

P ke - n	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	P N	P T
1	2	0	1	0	7	1	19	0	0	0	1	0	0	19	2	3	0	17	0	10	13	1	3	0	1
2	2	0	17	0	0	2	0	0	0	12	4	2	2	3	1	11	18	0	15	0	1	0	1	2	7
3	1	0	5	8	0	16	1	1	0	13	2	19	0	0	0	1	0	1	12	7	0	0	13	0	0
4	17	0	2	0	2	1	0	17	0	1	1	0	0	2	11	1	1	0	1	5	0	19	1	17	1
5	0	0	17	0	9	1	19	1	0	2	15	0	0	18	1	2	1	2	2	2	2	0	0	0	6
6	3	0	0	19	2	3	0	1	1	1	10	19	12	0	0	2	1	5	0	5	2	0	2	0	12
7	15	0	0	12	2	1	0	1	0	8	3	0	0	8	2	1	19	1	0	7	1	1	18	0	1
8	0	20	0	1	1	1	0	3	0	6	3	0	0	19	0	4	0	1	0	12	12	0	16	0	1
9	1	18	0	0	5	0	2	12	15	0	16	0	4	0	2	2	0	0	0	13	1	0	4	5	0
10	3	0	1	2	12	14	0	1	2	5	3	0	1	1	3	0	0	16	3	0	0	20	1	16	0

Tabel 5 memuat persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan clustering k-means dan singular value decomposition. Pada tabel 5 persebaran topik jurnal lebih merata dibandingkan tanpa reduksi dimensi. Terutama pada percobaan ke-4, hanya

topik jurnal komputer yang berjumlah 19 dokumen pada cluster 1. Sedangkan pada cluster 2, 3, 4, dan 5 tiap topik jurnal *tercluster* secara baik dan benar. Sehingga dapat disimpulkan bahwa teknik reduksi dimensi singular value decomposition memberikan pengaruh terhadap ketepatan persebaran topik jurnal di setiap cluster menggunakan metode k – means.

Tabel 5 Hasil Clustering Kombinasi SVD dan K-Means

P ke - n	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	P N	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT
1	0	0		20	6	0	0	0	0	14	0	20	0	0	0	0	0	17	0	0	20	0	2	0	0
2	0	0		20	6	0	20	0	0	0	19	0	0	0	0	1	0	0	0	14	0	0	19	0	0
3	0	20		0	0	1	0	0	0	14	0	0	18	0	0	19	0	2	0	6	0	0	1	19	0
4	0	0	9	0	0	0	0	0	0	20	0	20	0	0	0	20	0	0	0	0	0	0	1	20	0
5	19	0	1	0	0	0	20	0	0	0	0	0	18	0	0	0	0	0	0	14	1	0	1	20	6
6	0	0	1	20	0	0	0	17	0	0	20	0	0	0	0	0	0	1	0	20	0	20	1	0	0
7	0	20	1	0	0	1	0	0	0	20	0	0	17	0	0	0	0	2	20	0	19	0	0	0	0
8	0	20	1	0	0	0	0	18	0	0	0	0	0	20	0	16	0	0	0	13	4	0	1	0	7
9	0	20	0	0	0	0	0	1	0	14	0	0	0	0	6	0	0	19	20	0	20	0	0	0	0
10	1	20	0	0	0	0	0	15	0	0	0	0	1	20	0	2	0	4	0	1	17	0	0	0	19

Tabel 6 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan clustering k-medoids. Persebaran topik jurnal terburuk terdapat pada percobaan ke-10, dikarenakan topik jurnal peternakan pada cluster 3 dan 4 memiliki jumlah dokumen yang dominan dibanding topik lain sebanyak 6 dan 14 dokumen secara berurutan. Di cluster 1, topik jurnal

pendidikan sebanyak 20 dokumen. Sedangkan di cluster 2, hanya terdapat 1 dokumen jurnal ekonomi. Di cluster 5, topik jurnal hukum sebanyak 19 dokumen. Berdasarkan percobaan tersebut, dapat disimpulkan bahwa persebaran data uji penelitian menggunakan metode k-medoids tidak merata.

Tabel 6 Confusion K – Medoids

P ke- n	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	P N	P T
1	1	1	0	0	6	0	0	0	0	12	19	4	0	3	0	0	14	0	9	2	0	1	20	8	0
2	1	6	0	11	0	15	14	20	6	0	0	0	0	3	2	0	0	0	0	18	4	0	0	0	0
3	5	0	0	0	7	8	3	1	0	0	0	0	11	20	0	7	12	4	0	0	0	0	4	0	13
4	15	20	0	1	0	0	0	0	0	14	0	0	6	1	0	0	0	14	0	0	5	0	0	18	6
5	10	20	1	0	0	9	0	14	0	8	1	0	5	19	0	0	0	0	0	12	0	0	0	1	0
6	19	0	9	2	0	0	5	8	8	0	1	0	1	0	12	0	0	1	0	8	0	15	1	12	0
7	0	0	19	0	2	3	0	0	0	2	0	20	0	0	1	17	0	0	0	10	0	0	1	20	5
8	14	12	3	0	0	0	0	4	0	2	6	0	2	0	0	0	6	8	20	0	0	2	3	0	18
9	2	20	0	10	2	3	0	0	3	0	15	0	0	2	1	0	0	20	5	0	0	0	0	0	18
10	1	0	0	20	0	1	0	0	0	0	1	1	3	0	6	1	0	9	0	14	16	19	8	0	0

Tabel 7 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan kombinasi SVD clustering k-medoids. Seperti hasil percobaan clustering menggunakan kombinasi teknik reduksi dimensi dan metode k – means, persebaran topik jurnal kombinasi teknik reduksi dimensi dan metode k-medoids pun lebih merata dibandingkan tanpa reduksi dimensi. Terutama ditunjukkan pada percobaan ke-9, pada cluster 1 ada 20 dokumen yang bertopik hukum. Di cluster 2 terdapat topik jurnal komputer sebanyak 17 dokumen, di cluster 3 jurnal pendidikan dominan dengan 19 dokumen, di cluster 4 jurnal peternakan dominan dengan 20 dokumen, dan di cluster 5 jurnal ekonomi juga dominan dengan 20 dokumen. Sehingga berdasarkan percobaan di tabel 7 dapat disimpulkan bahwa teknik reduksi dimensi singular value decomposition memberikan pengaruh terhadap ketepatan persebaran topik jurnal di setiap cluster menggunakan metode k – medoids.

Tabel 8 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan fuzzy c-means. Kekhasan dari metode fuzzy c-means clustering adalah satu data dapat masuk ke lebih dari satu buah cluster. Namun pada tabel 8, peneliti hanya menampilkan dokumen yang nilai kemiripannya terbesar di cluster tersebut. Sehingga pada sepuluh kali percobaan clustering dengan fuzzy c-means, data uji penelitian cenderung membentuk 3 buah cluster walaupun masukan jumlah k (cluster) di awal adalah lima. Hal ini dikarenakan nilai kemiripan pada 2 cluster lainnya lebih rendah

dibandingkan dengan 3 cluster tersebut. Seolah-olah dalam 2 cluster tersebut tidak memiliki dokumen yang masuk padahal dalam kenyataannya ada walaupun nilai kemiripannya rendah.

Apabila diperhatikan dengan seksama, hasil clustering pada percobaan ke-1, 3, 4, 5, 6, 7, 8, dan 9 menghasilkan pola persebaran topik jurnal yang mirip. Pola yang dibentuk adalah 2 cluster yang tidak memiliki topik jurnal yang dominan. Cluster berikutnya terdapat 6 dokumen jurnal ekonomi, 20 dokumen jurnal hukum, dan 1 dokumen jurnal pendidikan. Cluster ke – 3 yang hanya terdapat 2 dokumen jurnal ekonomi, dan cluster terakhir terdapat 12 dokumen jurnal ekonomi, 20 dokumen jurnal komputer, dan 20 dokumen jurnal peternakan. Pola cluster ini berulang walaupun urutan clusternya tidak sama antara tiap percobaan.

Tabel 9 menunjukkan persebaran lima topik jurnal yaitu ekonomi, hukum, komputer, pendidikan, dan peternakan dalam tiap cluster pada sepuluh kali percobaan kombinasi SVD dan fuzzy c-means. Sama halnya dengan percobaan clustering fuzzy c-means, tabel 9 hanya menampilkan dokumen yang nilai kemiripannya terbesar di cluster tersebut. Bedanya, apabila percobaan clustering fuzzy c-means cenderung membentuk 3 cluster. Pada percobaan clustering dengan kombinasi singular value decomposition dan fuzzy c – means cenderung membentuk 2 cluster. Hasil cluster nya pun memiliki pola persebaran data yang mirip di setiap percobaan walaupun urutan persebaran data pada cluster berbeda. Pola yang ditampilkan adalah 3 cluster yang bernilai 0 dokumen, cluster yang memiliki 1 dokumen jurnal ekonomi dan 20 dokumen jurnal hukum, cluster yang memiliki 19

dokumen jurnal ekonomi, 20 dokumen jurnal komputer, peternakan.
20 dokumen jurnal pendidikan, dan 20 dokumen jurnal

Tabel 7 Hasil Clustering Kombinasi SVD dan K- Medoids

P ke - n	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	P N	P T
1	4	0	0	0	0	0	0	0	0	20	0	20	1	0	0	16	0	4	0	0	0	0	15	20	0
2	9	0	0	0	19	0	0	20	0	0	0	20	0	0	1	11	0	0	0	0	0	0	0	20	0
3	0	0	2	0	0	0	0	20	0	0	2	0	1	20	2	0	0	17	0	1	18	0	0	0	17
4	0	0	4	0	0	0	0	20	0	15	0	0	0	0	20	20	0	0	5	0	0	0	16	0	0
5	10	8	0	0	0	0	0	19	0	0	1	0	0	0	20	0	0	1	20	0	9	12	0	0	0
6	0	0	0	0	16	0	0	0	20	0	20	2	0	0	4	0	0	20	0	0	0	18	0	0	0
7	4	20	0	0	0	16	0	0	0	0	0	0	0	0	20	0	0	0	20	0	0	0	20	0	0
8	0	0	2	5	1	0	0	0	15	0	8	0	0	0	19	0	0	18	0	0	12	20	0	0	0
9	0	20	0	0	0	0	0	17	1	0	0	0	1	19	0	0	0	0	0	20	20	0	2	0	0
10	0	0	3	0	12	3	0	0	0	8	0	0	1	20	0	17	0	16	0	0	0	20	0	0	0

Tabel 8 Hasil Clustering Fuzzy C-Means

P ke - n	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	P N	P T
1	6	20	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	20	19	20
2	0	0	0	0	0	6	20	0	1	0	3	0	20	16	0	11	0	0	3	20	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	6	20	0	1	0	12	0	20	19	20
4	0	0	0	0	0	12	0	20	19	20	2	0	0	0	0	6	20	0	1	0	0	0	0	0	0
5	12	0	20	19	20	0	0	0	0	0	6	20	0	1	0	2	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	12	0	20	19	20	0	0	0	0	0	6	20	0	1	0	2	0	0	0	0
7	2	0	0	0	0	6	20	0	1	0	0	0	0	0	0	12	0	20	19	20	0	0	0	0	0
8	6	20	0	1	0	2	0	0	0	0	12	0	20	19	20	0	0	0	0	0	0	0	0	0	0
9	12	0	20	19	20	2	0	0	0	0	0	0	0	0	0	6	20	0	1	0	0	0	0	0	0
10	0	0	0	0	0	10	0	20	19	20	6	20	0	1	0	2	0	0	0	0	2	0	0	0	0

Tabel 9 Hasil Clustering Kombinasi SVD dan Fuzzy C-Means

P ke - n	Cluster 1					Cluster 2					Cluster 3					Cluster 4					Cluster 5				
	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	PN	PT	E	H	K	P N	P T
1	1	20	0	0	0	0	0	0	0	0	19	0	20	20	20	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	19	0	20	20	20	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	20	20	20	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	19	0	20	20	20	1	20	0	0	0	0	0	0	0	0
5	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	19	0	20	20	20	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	19	0	20	20	20	1	20	0	0	0	0	0	0	0	0

clustering fuzzy c-means tidak menunjukkan peningkatan dibandingkan dengan tanpa menggunakan SVD. Sebaliknya hasil akurasi menjadi turun. Hanya terlihat kombinasi SVD dan fuzzy c-means meningkatkan nilai recall sebesar 0,98 dibandingkan tanpa reduksi dimensi yaitu 0,84.

Dari total 10 kali percobaan, kombinasi antara clustering fuzzy c-means dan SVD cenderung membentuk 2 cluster. Kecenderungan inilah yang menyebabkan kecilnya hasil dari precision, sedangkan nilai recall sangat bergantung pada nilai kategori yang diambil pada masing – masing cluster yang terbentuk. Nilai rand index kombinasi SVD dan fuzzy c-means sebesar 0,51 mengalami penurunan dibandingkan nilai rand index tanpa SVD yaitu 0,59. Akurasi yang rendah kemungkinan disebabkan oleh prapengolahan dokumen yang kurang baik serta reduksi dimensi pada fitur sehingga berkurangnya ragam term untuk dilakukan fuzzifikasi. Sehingga dapat disimpulkan, bahwa untuk data uji penelitian ini teknik reduksi SVD tidak memberikan pengaruh yang baik pada akurasi hasil clustering menggunakan metode fuzzy c-means.

Tabel 12 Perbandingan Performa clustering fuzzy c -means dengan dan tanpa SVD

	Fuzzy C- Means	
	Tanpa SVD	Dengan SVD
<i>Precision</i>	0,3131984	0,282893
<i>Recall</i>	0,8400041	0,98
<i>F – measure</i>	0,448089	0,439047
<i>Rand Index</i>	0,5920407	0,519394

Selanjutnya, berdasarkan waktu komputasi pembentukan clustering yang digambarkan pada tabel 13 menunjukkan bahwa SVD memberikan pengaruh signifikan pada waktu eksekusi. Pada tiap metode clustering waktu komputasi pun menjadi lebih cepat dibandingkan tanpa reduksi dimensi. Hal ini dikarenakan, clustering dengan reduksi dimensi hanya memproses fitur yang lebih sedikit pada data yang belum direduksi dimensinya.

Tabel 13 Perbandingan Waktu Komputasi Clustering dengan dan tanpa SVD

	Waktu Eksekusi (s)	
	Tanpa SVD	Dengan SVD
K- Means	211,3341	11,0818
K-Medoids	262,5183	25,6725
Fuzzy C-Means	1013,9009	37,6062

4. Kesimpulan

Pada penelitian ini, teknik reduksi dimensi Singular Value Decomposition dikombinasikan dengan metode clustering k-means, k-medoids, dan fuzzy c-means untuk

mengelompokkan dokumen. Berdasarkan hasil pengujian, ditemukan bahwa SVD berpengaruh terhadap clustering k-means dan k-medoids. Dibuktikan dengan meningkatnya akurasi hasil clustering k- means dan k- medoids sebesar 0,11 dan 0,10. Sebaliknya, kombinasi fuzzy c-means dan SVD tidak memberikan hasil akurasi yang baik. Fenomena ini kemungkinan disebabkan oleh proses prapengolahan dokumen yang kurang baik dan reduksi dimensi pada fitur dokumen. Sedangkan berdasarkan waktu eksekusi pembentukan clustering, SVD mampu meningkatkan waktu eksekusi secara signifikan pada semua metode clustering. Pada penelitian selanjutnya, diharapkan :

3. Dapat menemukan solusi untuk meningkatkan akurasi fuzzy c-means baik menggunakan SVD maupun teknik reduksi lainnya.
4. Melakukan percobaan clustering dengan teknik reduksi dimensi lain untuk data yang memiliki lebih banyak *noise* dan *sparse*.

REFERENSI

- Indranandita, A., Susanto, B., & Rahmat, A., 1998, “Sistem Klasifikasi dan Pencarian Jurnal dengan Menggunakan Metode Naïve Bayes dan Vector Space Model”, Jurnal Informatika, 4(2).
- Han, J., Pei, J., & Kamber, M., 2011, “Data Mining: concepts and techniques”, Elsevier.
- S. C. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. A. Harshman., 1990, “Indexing by latent semantic analysis”, *Journal of the American Society of Information Science*, 41(6), pp. 391-407.
- S. T. Dumais., 2005, “Latent semantic analysis”. *Annual Review of Information Science and Technology*, 38 (1), pp. 188-230.
- Leskovec, J., Rajaraman, A., & Ullman, J. D., 2014, “Mining of massive datasets”, Cambridge University Press.
- R. L. Burden, J. D. Faires., 2011, “*Numerical Analysis*”, Brooks/Cole Cengage Learning.
- Thomo, A., 2009, “Latent semantic analysis Tutorial”, *Victoria, Canada.*, 1–7.
- Kaufman, L. and Rousseeuw, P.J., 1987, “Clustering by means of Medoids”, in *Statistical Data Analysis Based on the Norm and Related Methods*, edited by Y. Dodge, North Holland, 405–416.
- Bezdek, James C., 1981, “*Pattern Recognition with Fuzzy Objective Function Algorithms*”, ISBN 0-306-40671-3.

Novita Hidayati, mahasiswa program, studi teknik informatika di fakultas ilmu komputer Universitas Sriwijaya. Saat ini sedang menyelesaikan tugas akhirnya dengan judul penelitian “*Pengaruh Singular Value Decomposition terhadap Metode – Metode Clustering pada Dokumen Berdimensi Tinggi*”.

Muhammad Ihsan Jambak, memperoleh gelar Insinyur Teknik Elektro dari Universitas Sriwijaya dan M.Sc dari Universiti Teknologi Malaysia, Johor Bahru, Malaysia. tahun 1992 dan 2004. Saat ini sebagai Staf Pengajar fakultas Ilmu Komputer Universitas Sriwijaya.

Danny Matthew Saputra, memperoleh gelar B.S dari ITT Telkom Indonesia dan M.S dari Eastern Kentucky University, Amerika tahun 2009 dan 2013. Saat ini sebagai Staf Pengajar fakultas Ilmu Komputer Universitas Sriwijaya.