

**KLASIFIKASI DOKUMEN BERITA BERBAHASA INDONESIA  
DENGAN ALGORITMA AGGLOMERATIVE SINGLE LINKAGE  
HIERARCHICAL CLUSTERING**

**SKRIPSI**

Oleh:

**IFTICHA DINA PRATIWI**

**0510963024-96**



**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2010**

UNIVERSITAS BRAWIJAYA



**KLASIFIKASI DOKUMEN BERITA BERBAHASA INDONESIA  
DENGAN ALGORITMA AGGLOMERATIVE SINGLE LINKAGE  
HIERARCHICAL CLUSTERING**

**SKRIPSI**

Sebagai salah satu syarat untuk memperoleh  
gelar Sarjana Komputer dalam bidang Ilmu Komputer

Oleh :

**IFTICHA DINA PRATIWI**  
**0510963024-96**



**PROGRAM STUDI ILMU KOMPUTER  
JURUSAN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS BRAWIJAYA  
MALANG  
2010**

UNIVERSITAS BRAWIJAYA



## LEMBAR PENGESAHAN SKRIPSI

### KLASIFIKASI DOKUMEN BERITA BERBAHASA INDONESIA DENGAN ALGORITMA *AGGLOMERATIVE SINGLE LINKAGE HIERARCHICAL CLUSTERING*

Oleh:  
**IFTICHA DINA PRATIWI**  
**0510963024-96**

Setelah dipertahankan di depan Majelis Penguji  
Pada tanggal 8 Maret 2010  
dan dinyatakan memenuhi syarat untuk memperoleh  
gelar Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I

Drs. Ach. Ridok, Mkom  
NIP. 196808251994031002

Pembimbing II

Bayu Rahayudi, ST., MT.  
NIP. 197407122006041001

Mengetahui,  
Ketua Jurusan Matematika  
Fakultas MIPA Universitas Brawijaya

Dr. Agus Suryanto, MSc.  
NIP. 196908071994121001

UNIVERSITAS BRAWIJAYA



## LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Ifticha Dina Pratiwi  
NIM : 0510963024-96  
Jurusan : Matematika  
Program Studi : Ilmu Komputer  
Penulis Skripsi berjudul : Klasifikasi dokumen berita berbahasa Indonesia dengan algoritma *Agglomerative Single Linkage Hierarchical Clustering*

Dengan ini menyatakan bahwa :

1. Isi dari Tugas Akhir yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Tugas Akhir ini.
2. Apabila dikemudian hari ternyata Tugas Akhir yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 8 Maret 2010  
Yang menyatakan,

(Ifticha Dina Pratiwi)  
NIM. 0510963024-96

UNIVERSITAS BRAWIJAYA



# KLASIFIKASI DOKUMEN BERITA BERBAHASA INDONESIA DENGAN ALGORITMA *AGGLOMERATIVE SINGLE LINKAGE HIERARCHICAL CLUSTERING*

## ABSTRAK

Berita sudah menjadi kebutuhan masyarakat Indonesia sehari – hari. Berita yang disajikan dalam bentuk media *online* berupa dokumen yang jumlahnya sangat banyak dan sulit sekali jika dokumen – dokumen berita tersebut diklasifikasikan secara manual, sehingga dibutuhkan sistem pengklasifikasian berita secara otomatis, untuk mempermudah pencarian berita mengenai suatu kejadian tertentu. Sistem pengklasifikasian berita merupakan penerapan dari *text mining* yang berusaha mengelompokkan secara otomatis suatu berita ke dalam suatu kategori tertentu. *Text mining* merupakan varian data mining yang berusaha untuk mengekstraksi dan menemukan informasi yang belum diketahui sebelumnya dari sekumpulan data tekstual yang besar jumlahnya.

Tahapan – tahapan yang dilakukan dalam sistem adalah, pertama dilakukan proses *case folding* yaitu merubah semua huruf menjadi huruf kecil, tahap kedua yaitu penguraian kata (*parsing*), tahap ketiga yaitu mengubah kata-kata ke dalam bentuk dasarnya (*stemming*), tahap keempat yaitu penghilangan *stopword* (kata-kata yang bukan merupakan ciri dari suatu dokumen) dan yang terakhir yaitu tahapan terpenting yang berusaha menemukan pola atau pengetahuan dari keseluruhan teks, yang dalam penelitian ini menggunakan algoritma *Agglomerative Single Linkage Hierarchical Clustering*. Untuk mengevaluasi efektifitas sistem klasifikasi, digunakan standar pengukuran *recall*, *precision* dan  $F_1$ -*measure*.

Hasil uji coba menunjukkan nilai *recall*, *precision*, dan  $F_1$  *measure* yang rendah, sehingga dapat disimpulkan metode yang diterapkan pada pembuatan sistem ini tidak cocok digunakan untuk mengklasifikasikan dokumen berita berbahasa Indonesia.

UNIVERSITAS BRAWIJAYA



# INDONESIAN LANGUAGE DOCUMENT CLASSIFICATION WITH AGGLOMERATIVE SINGLE LINKAGE HIERARCHICAL CLUSTERING ALGORITHM

## ABSTRACT

News has become Indonesian society needs in their daily life. News is presented in the form of documents online media is a large scale of documents and very difficult to be clasified manually, so is needed automatic news classification system, to facilitate the search for news about a particular event. News classification system is the application of text mining which automatically classifies a message into a specific category. Text mining is a variant of data mining which are extracting and finding unknown information previously from a large amount of collection of textual data.

There are some steps in this system, first step is the case folding process that changes all letters to lower case, second step is parsing the word (parsing), third step is stopword removal (the words that are not characteristic of a document), fourth step is changing the words into basic shapes (stemming) and the last step is the most important which tries to find patterns or knowledge of the entire text, which in this research using Agglomerative Single Linkage Hierarchical Clustering algorithm. To evaluate the effectiveness of the classification system, using standard measurement consist of *recall*, *precision* and  $F_1$ -*measure*.

The test result perform low value of *recall*, *precision*, and  $F_1$  *measure*. So, it can be concluded that applied method in this system is not suitable to be used in classify the Indonesian news document.

UNIVERSITAS BRAWIJAYA



## KATA PENGANTAR

Segala puji syukur kepada Allah SWT atas rahmat, hidayah-Nya, dan atas izin-Nyalah maka penulis dapat menyelesaikan tugas akhir dengan judul Klasifikasi Dokumen Berita Berbahasa Indonesia dengan Algoritma *Agglomerative Single Linkage Hierarchical Clustering*. Tugas akhir ini disusun sebagai syarat untuk memperoleh gelar Sarjana Komputer dalam bidang Ilmu Komputer.

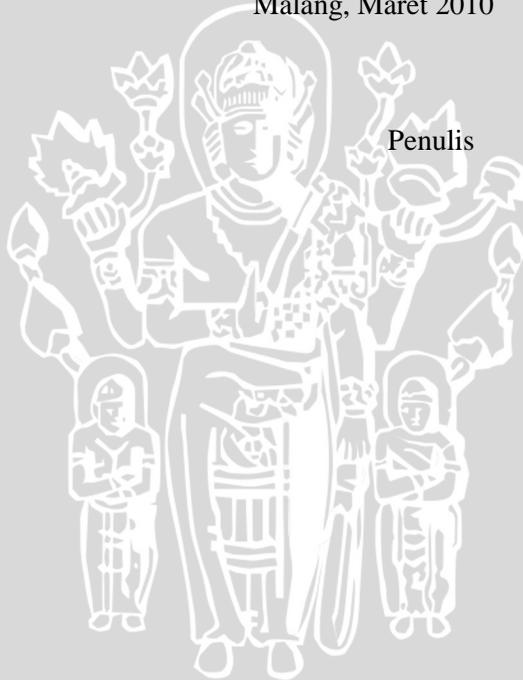
Dalam penyelesaian tugas akhir ini, penulis telah mendapat begitu banyak bantuan dari berbagai pihak. Atas bantuan yang telah diberikan, penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Drs. Achmad Ridok, M.Kom. dan Bayu Rahayudi, ST., MT., selaku pembimbing utama dan pembimbing pendamping dalam penulisan tugas akhir, serta Wayan Firdaus Mahmudy, SSi, MT, terima kasih atas semua waktu dan bimbingan yang telah diberikan.
2. Dr. Agus Suryanto, Msc selaku ketua jurusan Matematika Fakultas Mipa Universitas Brawijaya.
3. Drs. Marji, MT sebagai Ketua Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Mipa, Universitas Brawijaya.
4. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya.
5. Langgeng Widiyantoro dan Ninik Mulyani sebagai orang tua penulis, serta Yurida Dila Pratiwi sebagai adik penulis yang telah memberikan dorongan, semangat baik materiil maupun spiritual dan doa restu.
6. Mbah Uti dan almarhum mbah Kakung, terima kasih atas doa yang selalu dipanjatkan untuk penulis.
7. Rekan-rekan di Program Studi Ilmu Komputer Universitas Brawijaya terutama kilimanjaro tim (Lusi, Rini, Fifin, Ratna, Pipit) yang telah memberikan semangat kepada penulis.
8. Semua teman – teman di pondokan putri Sumpersari 292 C, terima kasih atas semua peristiwa yang kita lewati bersama.

9. Semua pihak yang telah membantu penulis baik secara langsung maupun tidak langsung sehingga terselesaikannya Tugas Akhir ini. Dan terima kasih telah menjadi bagian dari perjalanan hidupku.

Penulis menyadari tugas akhir ini masih banyak memiliki kekurangan sehingga penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan mutu isi tugas akhir ini untuk kelanjutan penelitian serupa di masa mendatang. Semoga penulisan tugas akhir ini bermanfaat bagi pembaca sekalian.

Malang, Maret 2010



Penulis

## DAFTAR ISI

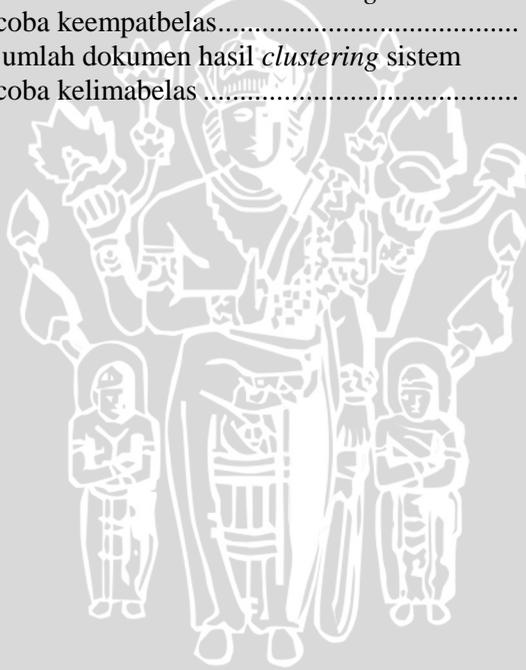
	Halaman
<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PENGESAHAN</b> .....	iii
<b>HALAMAN PERNYATAAN</b> .....	v
<b>ABSTRAK</b> .....	vii
<b>ABSTRACT</b> .....	ix
<b>KATA PENGANTAR</b> .....	xi
<b>DAFTAR ISI</b> .....	xiii
<b>DAFTAR GAMBAR</b> .....	xv
<b>DAFTAR TABEL</b> .....	xvii
<b>DAFTAR LAMPIRAN</b> .....	xxi
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Manfaat .....	3
1.6 Sistematika Penulisan .....	3
<b>BAB II TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1 Berita .....	5
2.2 Pemrosesan Text .....	6
2.2.1 <i>Case Folding</i> .....	6
2.2.2 <i>Parsing</i> .....	6
2.2.3 <i>Stemming</i> .....	7
2.2.4 Pembuangan <i>Stopwords</i> .....	7
2.3 <i>Stemming</i> pada Bahasa Indonesia .....	8
2.3.1 Morfologi .....	8
2.3.2 Proses <i>Stemming</i> Bahasa Indonesia .....	9
2.4 <i>Clustering</i> .....	11
2.4.1 Dokumen <i>Clustering</i> .....	11
2.4.2 Jenis-jenis <i>Custering</i> .....	12
2.4.3 <i>Hierarchical Clustering</i> .....	13

2.5 Implementasi Algoritma <i>Agglomerative Single Linkage Hierarchical Clustering</i> .....	17
2.6 Ukuran Kualitas Dokumen <i>Clustering</i> .....	19
<b>BAB III METODE PENELITIAN.....</b>	<b>21</b>
3.1 Perancangan Sistem .....	22
3.1.1 Deskripsi Umum Sistem.....	22
3.1.2 Batasan Sistem .....	22
3.2 Proses Klasifikasi.....	23
3.2.1 <i>Preprocessing</i> .....	23
3.2.2 Pemrosesan dengan <i>Agglomerative Single Linkage Hierarchical Clustering</i> .....	27
3.4 Implementasi Perhitungan .....	30
3.4.1 Ukuran Kualitas Dokumen.....	38
3.5 Rancangan Uji Coba .....	38
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>41</b>
4.1 Lingkungan Implementasi .....	41
4.1.1 Lingkungan Perangkat Keras .....	41
4.1.2 Lingkungan Perangkat Lunak .....	41
4.2 Struktur Data.....	41
4.3 Implementasi Algoritma <i>Agglomerative Single Linkage Hierarchical Clustering</i> .....	43
4.3.1 Implementasi <i>Preprocessing</i> .....	43
4.3.2 Implementasi <i>Parsing</i> dan <i>Case Folding</i> .....	43
4.3.3 Implementasi <i>Stemming</i> .....	44
4.3.4 Implementasi <i>Stopword Removal</i> .....	51
4.3.5 Implementasi <i>Agglomerative Single Linkage Hierarchical Clustering</i> .....	51
4.4 Implementasi Antar Muka .....	55
4.5 Implementasi Uji Coba .....	58
4.5.1 Skenario Evaluasi.....	58
4.5.2 Hasil Evaluasi.....	58
4.5.3 Analisa Hasil .....	91
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>93</b>
5.1 Kesimpulan .....	93
5.2 Saran .....	93
<b>DAFTAR PUSTAKA .....</b>	<b>95</b>

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur berita .....	6
Gambar 2.2 <i>Clustering</i> .....	12
Gambar 2.3 Ilustrasi algoritma <i>hierarchical clustering</i> .....	14
Gambar 2.4 Ilustrasi <i>single linkage</i> .....	15
Gambar 2.5 Ilustrasi <i>centroid linkage</i> .....	16
Gambar 2.6 Ilustrasi <i>complete linkage</i> .....	16
Gambar 2.7 Ilustrasi <i>average linkage</i> .....	17
Gambar 2.8 <i>Single linkage</i> dendogram untuk jarak antara lima objek.....	19
Gambar 3.1 Diagram Alir Penelitian.....	21
Gambar 3.2 Blok diagram proses sistem .....	23
Gambar 3.3 <i>Flowchart parsing</i> .....	24
Gambar 3.4 <i>Flowchart</i> proses <i>stemming</i> .....	25
Gambar 3.5 <i>Flowchart</i> penghilangan <i>stopwords</i> .....	26
Gambar 3.6 <i>Flowchart</i> penghitungan frekuensi.....	28
Gambar 3.7 <i>Flowchart Agglomerative Single Linkage hierarchical clustering</i> .....	29
Gambar 3.8 Dendogram hasil klasifikasi .....	37
Gambar 4.1 Antar muka tahap <i>preprocessing</i> .....	56
Gambar 4.2 Antar muka proses <i>clustering</i> .....	57
Gambar 4.3 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba pertama .....	59
Gambar 4.4 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba kedua .....	61
Gambar 4.5 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba ketiga .....	63
Gambar 4.6 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba keempat .....	65
Gambar 4.7 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba kelima .....	68
Gambar 4.8 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba keenam .....	70
Gambar 4.9 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba ketujuh .....	72

Gambar 4.10 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba kedelapan .....	74
Gambar 4.11 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba kesembilan.....	77
Gambar 4.12 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba kesepuluh .....	79
Gambar 4.13 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba kesebelas .....	81
Gambar 4.14 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba keduabelas.....	83
Gambar 4.15 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba ketigabelas .....	84
Gambar 4.16 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba keempatbelas.....	88
Gambar 4.17 Grafik jumlah dokumen hasil <i>clustering</i> sistem pada uji coba kelimabelas .....	89



## DAFTAR TABEL

	Halaman
Tabel 2.1 Aturan partikel infleksional.....	9
Tabel 2.2 Aturan kata ganti infleksional.....	10
Tabel 2.3 Aturan yang menangani urutan prefiks derivasional Pertama .....	10
Tabel 2.4 Aturan prefiks derivasional kedua.....	10
Tabel 2.5 Aturan sufiks derivasional .....	11
Tabel 2.6 Kategori hasil klasifikasi.....	19
Tabel 3.1 Aturan peleburan.....	27
Tabel 3.2 Hasil <i>preprocessing</i> dokumen.....	31
Tabel 3.3 Matrik dokumen .....	37
Tabel 3.4 Matrik iterasi 1 .....	37
Tabel 3.5 Hasil uji coba.....	39
Tabel 4.1 Jumlah dokumen setiap kategori pada uji coba pertama.....	56
Tabel 4.2 Evaluasi efektivitas uji coba pertama (8 dokumen). .....	59
Tabel 4.3 Jumlah dokumen setiap kategori pada uji coba kedua.....	60
Tabel 4.4 Evaluasi efektivitas uji coba kedua (16 dokumen). .....	61
Tabel 4.5 Jumlah dokumen setiap kategori pada uji coba ketiga .....	62
Tabel 4.6 Evaluasi efektivitas uji coba ketiga (24 dokumen). .....	63
Tabel 4.7 Jumlah dokumen setiap kategori pada uji coba keempat .....	64
Tabel 4.8 Evaluasi efektivitas uji coba keempat (32 dokumen). .....	66
Tabel 4.9 Jumlah dokumen setiap kategori pada uji coba kelima .....	66
Tabel 4.10 Evaluasi efektivitas uji coba kelima (40 dokumen). .....	68

Tabel 4.11 Jumlah dokumen setiap kategori pada uji coba keenam.....	69
Tabel 4.12 Evaluasi efektivitas uji coba keenam (48 dokumen). .....	70
Tabel 4.13 Jumlah dokumen setiap kategori pada uji coba ketujuh.....	71
Tabel 4.14 Evaluasi efektivitas uji coba ketujuh (56 dokumen). .....	72
Tabel 4.15 Jumlah dokumen setiap kategori pada uji coba kedelapan .....	73
Tabel 4.16 Evaluasi efektivitas uji coba kedelapan (64 dokumen) .....	74
Tabel 4.17 Jumlah dokumen setiap kategori pada uji coba kesembilan .....	75
Tabel 4.18 Evaluasi efektivitas uji coba kesembilan (72 dokumen) .....	77
Tabel 4.19 Jumlah dokumen setiap kategori pada uji coba kesepuluh .....	78
Tabel 4.20 Evaluasi efektivitas uji coba kesepuluh (80 dokumen) .....	79
Tabel 4.21 Jumlah dokumen setiap kategori pada uji coba kesebelas .....	80
.....	66
Tabel 4.22 Evaluasi efektivitas uji coba kesebelas (88 dokumen) .....	81
Tabel 4.23 Jumlah dokumen setiap kategori pada uji coba keduabelas.....	82
Tabel 4.24 Evaluasi efektivitas uji coba keduabelas (96 dokumen) .....	84
Tabel 4.25 Jumlah dokumen setiap kategori pada uji coba ketigabelas .....	84
Tabel 4.26 Evaluasi efektivitas uji coba ketigabelas (104 dokumen) .....	86
Tabel 4.27 Jumlah dokumen setiap kategori pada uji coba keempabelas.....	87

Tabel 4.28 Evaluasi efektivitas uji coba keempatbelas (112 dokumen) .....	88
Tabel 4.29 Jumlah dokumen setiap kategori pada uji coba kelimabelas .....	89
Tabel 4.30 Evaluasi efektivitas uji coba kelimabelas (120 dokumen) .....	90

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



## DAFTAR LAMPIRAN

Halaman

Lampiran 1	Daftar <i>stopwords</i> .....	99
Lampiran 2	Contoh dokumen yang diklasifikasikan.....	103

UNIVERSITAS BRAWIJAYA



UNIVERSITAS BRAWIJAYA



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Berita sudah menjadi kebutuhan masyarakat Indonesia sehari – hari. Setiap hari terdapat berbagai macam peristiwa yang dilaporkan kepada masyarakat dalam bentuk berita. Konsumsi masyarakat terhadap berita meningkat seiring berjalannya waktu. Oleh sebab itu, semakin banyak pula penerbit berita di negeri ini. Penerbit berita tersebut tidak hanya menyajikan berita dalam bentuk media cetak tetapi juga media *online* (internet). Berita yang disajikan dalam bentuk media *online* berupa dokumen yang jumlahnya sangat banyak dan sulit sekali jika dokumen – dokumen berita tersebut diklasifikasikan secara manual, sehingga kebutuhan akan proses pengklasifikasian berita secara otomatis, yaitu penggolongan suatu berita ke dalam suatu kategori diperlukan untuk mempermudah pencarian berita mengenai suatu kejadian tertentu.

Penelitian pengelompokkan berita berbahasa Indonesia sebelumnya sudah pernah dilakukan, yaitu pengelompokkan berita menggunakan metode klasifikasi *naïve bayes* yang sering disebut sebagai *naïve bayes classifier*. *Naïve bayes Classifier* menggunakan teori probabilitas sebagai dasar teori (Ma'lufatul, 2007). Juga penelitian dengan algoritma *Single Pass Clustering*, yaitu dengan menggunakan perhitungan tingkat kemiripan (*Similarity*) dengan *standard cosine Similarity*. *Similarity* yang telah dihasilkan selanjutnya dievaluasi untuk menentukan pasangan – pasangan dokumen yang dinyatakan mirip berdasarkan nilai *threshold* tertentu, (Zainal dan Ari Novan, 2001).

*Text mining* merupakan varian data mining yang berusaha untuk mengekstraksi dan menemukan sesuatu yang belum diketahui sebelumnya dan mengandung informasi yang sangat berguna dari sekumpulan data tekstual yang besar jumlahnya (Hearst, 2003). Dalam *text mining* terdapat dua teknik pembelajaran, yaitu *unsupervised learning* dan *supervised learning*. *Clustering* adalah contoh dari *unsupervised learning*, dimana sekelompok data langsung dikelompokkan berdasarkan tingkat kemiripannya tanpa dilakukan *supervise* (Even dan Zohar, 2002).

*Clustering* dapat digunakan untuk membantu menganalisis berita dengan mengelompokkan secara otomatis berita yang memiliki kesamaan. *Clustering* dokumen adalah aktifitas klasifikasi otomatis dokumen tanpa supervisi ke dalam *clusters* atau grup-grup (Yudi dan Masayu, 2005).

Terdapat beberapa metode *clustering* data, diantaranya adalah *partitioning clustering*, *overlapping clustering*, *hybrid clustering*, dan *hierarchical clustering* (Ridho, 2006). *Hierarchical clustering* dapat digunakan untuk aplikasi *automated text integration*, dan dari hasil survei terhadap 100 orang responden, sebanyak 78% responden mengatakan bahwa integrasi dokumen yang dihasilkan telah benar, (Budhi, 2008). Sehingga dalam tugas akhir ini akan digunakan metode *hierarchical clustering*. Metode ini mengelompokkan data menggunakan hasil *clustering* yang sebelumnya (*nested clustering*). Dalam *hierarchical clustering* ini, hasil *clustering* pada level 1 akan di-*cluster* lagi dengan *cluster* yang lain berdasarkan kemiripan yang terdapat pada *cluster* yang dihasilkan pada *clustering* level 1.

Berdasarkan latar belakang yang telah dipaparkan, maka judul yang diambil dalam skripsi ini “Klasifikasi dokumen berita berbahasa Indonesia dengan algoritma *Agglomerative Single Linkage Hierarchical Clustering Hierarchical Clustering*”.

## **1.2 Rumusan Masalah**

Rumusan masalah dalam skripsi adalah :

1. Bagaimana sistem pengklasifikasian berita berbahasa Indonesia dengan memakai algoritma *Agglomerative Single Linkage Hierarchical Clustering Hierarchical Clustering*?
2. Bagaimana efektivitas algoritma *Agglomerative Single Linkage Hierarchical Clustering Hierarchical Clustering* tersebut pada sejumlah sampel berita berbahasa Indonesia?

## **1.3 Batasan Masalah**

Batasan Masalah yang diangkat dalam skripsi ini adalah:

1. Klasifikasi hanya pada berita dalam bahasa Indonesia, tidak dalam bahasa lainnya.
2. Dokumen berita yang akan digunakan dalam skripsi ini berupa dokumen dalam format txt.

3. Kata yang dihasilkan dari *headline* atau judul beritanya berdasarkan frekuensi kata, teras berita dan tubuh berita dianggap sama. Probabilitas yang dibangun hanya berdasarkan frekuensi kemunculan kata tidak mempedulikan posisi kata tersebut pada teks berita.
4. Suatu kata dianggap berdiri sendiri dan lepas dari kata disekitarnya. Hal ini dimaksudkan untuk menyederhanakan proses pengolahan jika kata diproses secara berpasangan tentu saja akan muncul berbagai faktor yang harus diperhitungkan.
5. Pada proses *stemming* tidak memperhitungkan adanya infiks (sisipan). Proses *stemming* yang dibangun hanya melakukan penghilangan prefiks dan sufiks.

#### **1.4 Tujuan**

Beberapa tujuan yang ingin dicapai dalam pelaksanaan skripsi ini adalah :

1. Mempelajari dan menerapkan teknik pengklasifikasian berita berbahasa Indonesia dengan algoritma *Agglomerative Single Linkage Hierarchical Clustering*.
2. Mengetahui efektivitas sistem untuk mengklasifikasikan berita dalam kategori yang sesuai kemudian menganalisa hasil dari efektivitas sistem.

#### **1.5 Manfaat**

Manfaat yang dapat diambil dari skripsi ini adalah untuk memudahkan proses pengklasifikasian berita secara otomatis dan mempunyai nilai keakuratan yang baik, sehingga memudahkan pencarian berita mengenai suatu kejadian tertentu.

#### **1.6 Sistematika Penulisan**

Di dalam skripsi ini, pembahasan materi disusun menjadi beberapa bab, dengan sistematika sebagai berikut :

##### **BAB I PENDAHULUAN**

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penulisan, manfaat penulisan, dan sistematika penulisan skripsi.

## BAB II LANDASAN TEORI

Pada bab ini berisi tentang teori – teori yang menjadi acuan untuk pelaksanaan penulisan skripsi.

## BAB III METODOLOGI DAN PERANCANGAN

Pada bab ini menjelaskan metode penelitian dan tentang langkah-langkah yang digunakan untuk pengklasifikasian berita berbahasa Indonesia dengan algoritma *Agglomerative Single Linkage Hierarchical Clustering Hierarchical Clustering*.

## BAB IV PEMBAHASAN

Pada bab ini dibahas implementasi metode *Agglomerative Single Linkage Hierarchical Clustering Hierarchical Clustering* dan beberapa hasil uji-coba klasifikasi.

## BAB V PENUTUP

Bab ini berisi kesimpulan dan saran.



## BAB II TINJAUAN PUSTAKA

### 2.1 Berita

Berita adalah laporan tentang peristiwa atau pendapat yang memiliki nilai yang penting, menarik bagi sebagian besar khalayak masih baru dan dipublikasikan secara luas melalui media masa periodik. Sebuah berita sudah pasti suatu informasi, tetapi suatu informasi belum tentu sebuah berita. Hal ini dikarenakan informasi baru dapat dikatakan berita apabila suatu informasi memiliki unsur-unsur yang mempunyai nilai berita atau nilai jurnalistik dan disebarluaskan pada khalayak.

Sebuah berita tersusun dari bagian-bagian yang terdiri dari organ-organnya. Bagian - bagian tersebut selengkapnya mencakup:

a. Judul

Judul mewakili isi berita yang ingin disampaikan dan memiliki daya tarik yang kuat. Gaya penulisannya kadang tidak mengikuti kaidah bahasa Indonesia.

b. Baris tanggal (*dateline*)

Waktu dan tempat berita tersebut diperoleh

c. Teras berita (*lead* atau *intro*)

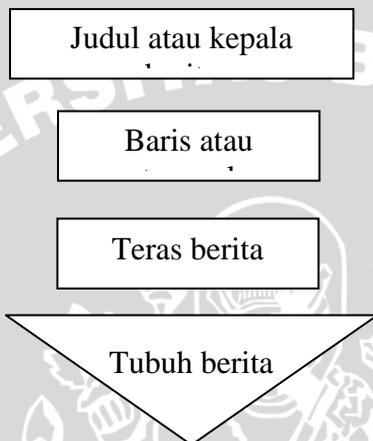
*Lead* terletak sebelum tubuh berita. *Lead* biasanya berisi ringkasan dan berupa kalimat yang menarik agar pembaca merasa penasaran sehingga ingin terus melanjutkan membaca ke tubuh berita. *Lead* juga biasanya berisi fakta paling penting dengan mengedepankan unsur 5W+1H (*what, who, when, where, why, dan how*)

d. Tubuh berita (*body*)

Tubuh berita berisi penjelasan atau uraian rinci unsur 5W+1H, baik yang sudah dikemukakan dalam teras maupun yang belum diungkapkan. Penulisan tubuh berita relative tidak sesulit menulis *lead*. Penulisan tubuh berita hanya melanjutkan apa yang sudah tertuang dalam teras yang mencerminkan pokok – pokok terpenting isi berita.

Bagian-bagian tersebut membentuk sebuah anatomi berita yang tersusun sebagai sebuah struktur yang utuh dan padu, yang sering

dinamakan sebagai gaya piramida terbalik (*inverted pyramid style*). Disebut demikian karena bagian tubuh berita disusun dengan pola pengembangan umum - khusus (dimulai dari hal umum, lalu secara berangsur-angsur menuju ke hal-hal yang semakin khusus) atau klimaks-antiklimaks (dari yang paling pokok atau penting beralih secara berturut-turut ke yang kurang pokok atau penting).



**Gambar 2.1** Struktur berita

## 2.2 Pemrosesan Teks

Menurut Yudi Wibisono, dalam pemrosesan teks tahap *preprocessing*, langkah-langkah yang dilakukan adalah *case folding*, *parsing*, pembuangan *stopwords*, dan *stemming*.

### 2.2.1 Case Folding

*Case folding* adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap *delimiter*.

### 2.2.2 Parsing

*Parsing* adalah memilah dokumen menjadi unit-unit yang lebih kecil misalnya kata, frasa, atau kalimat. Unit hasil pemrosesan tersebut dinamakan token. Proses parsing merujuk pada proses pengenalan token yang terdapat dalam rangkaian teks. Bagian dasar dalam parsing adalah algoritma pengambilan token dari teks, yang disebut sebagai tokenizer. Proses ini memerlukan pengetahuan

bahasa untuk menangani karakter –karakter khusus, dan menentukan batasan satuan unit dalam dokumen (Arsono, 2009).

Contoh :

Input : Fahri menghampiri apartemen Maria untuk minta bantuan karena komputernya error.

Output : Fahri	menghampiri	apartemen	Maria
untuk	minta	bantuan	karena
komputernya	error		

### 2.2.3 Stemming

*Stemming* adalah penghilangan prefiks dan suffiks dari istilah yang terdapat dalam kueri dan dokumen hasil dari proses *parsing* sebelumnya. *Stemming* dilakukan atas dasar asumsi bahwa kata – kata yang memiliki stem yang sama memiliki makna yang serupa pula (Arsono, 2009).

*Stemming* mempunyai beberapa algoritma salah satunya yaitu Porter Stemmer. Porter stemmer dikembangkan oleh Martin Porter dari Universitas Cambridge pada tahun 1980. *Stemmer* ini berdasarkan ide bahwa akhiran dalam bahasa Inggris (mendekati 1200) terbentuk dari kombinasi akhiran yang lebih kecil dan sederhana. *Stemmer* ini merupakan *stemmer* dengan tahapan yang berurutan yaitu terdiri dari lima tahapan yang mengaplikasikan aturan dalam setiap tahapannya. Apabila sebuah aturan akhiran dicocokkan dengan sebuah kata kemudian sesuai dengan aturan itu diuji sehingga dapat menghasilkan *stem* yang dilakukan pada setiap tahapan. Apabila sesuai dengan aturan maka akhiran dapat dihilangkan.

Algoritma Porter *stemmer* merupakan algoritma yang sering digunakan untuk *stemmer* pada dokumen Bahasa Inggris.

### 2.2.4 Pembuangan Stopwords

*Stopwords* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopwords* adalah “yang”, “dan”, “di”, “dari” dan seterusnya.

## 2.3 *Stemming* pada bahasa Indonesia

### 2.3.1 Morfologi

Morfologi merupakan bagian dari ilmu bahasa yang mengkaji struktur, bentuk dan pengolahan kata. Struktur kata ialah susunan bentuk bunyi ujaran atau lambang (tulisan) yang menjadi unit bahasa yang berbentuk tunggal atau hasil dari proses pengimbuhan, pemajemukan dan penggandaan. Struktur kata disebut dengan morfem dasar dan morfem terikat. Morfem dasar merupakan kata yang dapat berdiri sendiri tanpa adanya tambahan, sedangkan morfem terikat merupakan kata yang terdiri dari morfem dasar atau gabungan morfem dasar.

Morfologi kata bahasa Indonesia dibagi ke dalam struktur infleksional dan derivasional, pembagian struktur tersebut digunakan untuk melakukan proses *stemming* pada bahasa Indonesia. Infleksional adalah struktur yang paling sederhana yang dinyatakan dalam penambahan sufiks dimana tidak mempengaruhi arti sebenarnya dari kata dasar yang dilekati. Sufiks infleksional dapat dibagi menjadi 2 jenis. Sufiks infleksional tersebut adalah sebagai berikut :

- a. sufiks *-lah, -kah, -pun, -tah*. Sufiks ini sebenarnya adalah partikel yang tidak mempunyai arti. Keberadaannya pada suatu kata adalah untuk penekanan. Contoh :  
dia + kah → diakah  
duduk + lah → duduklah
- b. sufiks *-ku, -mu, -nya*. Sufiks ini berfungsi sebagai kata ganti kepemilikan. Contoh :  
tas + ku → tasku  
buku + ku → bukumu

Penambahan sufiks infleksional tidak akan merubah bentuk dasar dari kata berimbuhan (Tala, 2003). Dengan kata lain, tidak ada penghilangan atau peleburan kata dasar pada kata berimbuhan. Kata dasar dapat ditentukan dengan mudah pada struktur infleksional dapat ditulis sebagai berikut :

Infleksional = (kata dasar + kata ganti) | (kata dasar + partikel) | (kata dasar + kata ganti + partikel)

Prefiks dapat ditambahkan pada suatu kata yang telah terdapat konfiks atau prefik, dan menghasilkan struktur prefiks ganda, tidak semua prefiks dapat ditambahkan pada kata yang telah mendapatkan prefiks atau konfiks.

Sehingga struktur morfologi pada kata derivasional adalah :  
 Derivasional = (prefiks +kata dasar) | (kata dasar |sufiks) | (prefiks +  
 kata dasar +sufiks) | (prefiks 1 + prefiks 2 + kata  
 dasar) | (prefiks 1 + prefiks 2 +kata dasar +sufiks).

Contoh dari struktur kata derivasional tersebut adalah sebagai berikut :

mem + beri → memberi  
 ambil + kan → ambilkan  
 meng + ambil + kan → mengambilkan  
 mem + per + indah → memperindah  
 mem + per + main + kan → mempermainkan

Struktur lain yang mungkin terjadi dalam morfologi bahasa Indonesia adalah penambahan sufiks infleksional pada struktur derivasional.

Sehingga dapat disimpulkan secara umum struktur morfologi kata bahasa Indonesia adalah :

Struktur morfologi = [prefiks 1] + [prefiks 2] + kata dasar [sufiks] + [kata ganti] + partikel.

Macam – macam dan penggolongan prefiks dan sufiks dijelaskan pada sub bab selanjutnya.

### 2.3.2 Proses *Steming* Bahasa Indonesia

Terdapat 5 aturan tahapan pada proses *stemming* dalam bahasa Indonesia (Tala, 2003). Aturan – aturan tersebut adalah

a. Aturan yang menangani partikel infleksional

**Tabel 2.1** Aturan partikel infleksional

Sufiks	Pengganti	Kondisi Tambahan	Contoh
Kah	NULL	NULL	Diakah → dia
Lah	NULL	NULL	Adalah → ada
Tah	NULL	NULL	Apatah → apa
Pun	NULL	NULL	Bukupun → buku

b. Aturan yang menangani kata ganti infleksional

**Tabel 2.2** Aturan kata ganti infleksional

Sufiks	Pengganti	Kondisi Tambahan	Contoh
Ku	NULL	NULL	bukuku→buku
Mu	NULL	NULL	bukumu→buku
nya	NULL	NULL	bukunya→buku

c. Aturan yang menangani urutan prefiks derivasional pertama

**Tabel 2.3** Aturan prefiks derivasional pertama

prefiks	Pengganti	Kondisi Tambahan	Contoh
Meng	NULL	NULL	mengukur→ukur
Meny	S	V...*	Menyapu→sapu
Men	T	V...*	Menuduh→tuduh
Men	NULL	NULL	Menduga→duga
Mem	P	V...*	Memukul→pukul
Mem	NULL	NULL	Membakar→bakar
Me	NULL	NULL	Merusak→rusak
Peng	NULL	NULL	Pengukur→ukur
Peny	S	V...*	Penyelam→selam
Pen	T	V...*	Penari→tari
Pen	NULL	NULL	Penduga→duga
Pem	P	V...*	Pemandu→pandu
Pem	NULL	NULL	Pembaca→baca
Di	NULL	NULL	Diukur→ukur
Ter	NULL	NULL	Tersipu→sipu
ke	NULL	NULL	Kekasih→kasih

\*kata dasar dimulai huruf vocal

d. aturan yang menangani urutan prefiks derivasional kedua

**Tabel 2.4** Aturan prefiks derivasional kedua

Prefiks	Pengganti	Kondisi Tambahan	Contoh
Ber	NULL	NULL	berlari→lari
Bel	NULL	Ajar	belajar→ajar
Be	NULL	kerja	bekerja→kerja
Per	NULL	NULL	Perjelas→jelas
Pel	NULL	ajar	Pelajar→ajar
Pe	NULL	NULL	pekerja→kerja

e. aturan yang menangani sufiks derivasional

**Tabel 2.5** Aturan sufiks derivasional

suffiks	Pengganti	Kondisi Tambahan	Contoh
Kan	NULL	Prefiks $\ell$ {ke,peng}	Tarikkan $\rightarrow$ tarik (meng)ambilkan $\rightarrow$ ambil
An	NULL	Prefik $\ell$ {di, meng, ter}	makanan $\rightarrow$ makan (per)janjian $\rightarrow$ janji
i	NULL	Prefiks $\ell$ {ber,ke,peng}	Tandai $\rightarrow$ tanda (men)dapati $\rightarrow$ dapat

(Tala, 2003)

## 2.4 Clustering

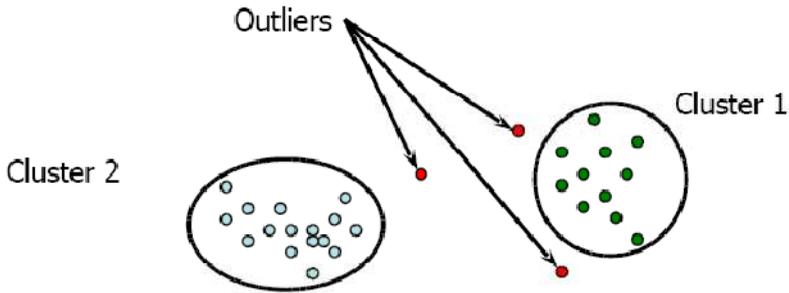
*Clustering* merupakan metode tidak terawasi (*unsupervised*), karena pola kelas data atau label sampel data tidak ditentukan terlebih dahulu. *Clustering* bergantung pada algoritma *clustering* dan *similarity measure* (Manning, 2006).

*Clustering* adalah suatu alat untuk analisa data, yang memecahkan permasalahan penggolongan. Objek dari *clustering* adalah untuk kasus pendistribusian (orang-orang, objek, peristiwa dan lain - lain) ke dalam kelompok, sedemikian sehingga derajat tingkat keterhubungan antar anggota *cluster* yang sama adalah kuat dan lemah antar anggota dari *cluster* yang berbeda. Dengan Cara ini masing-masing *cluster* menguraikan, dalam kaitan dengan kumpulan atau koleksi data, kelas dimana milik anggota-anggotanya.

*Cluster* adalah alat penemuan yang mengungkapkan hubungan dan struktur di dalam data, yang sebelumnya tidak jelas. Algoritma *clustering* berusaha untuk menemukan kelompok komponen secara natural, berdasarkan pada beberapa kesamaan. Ilustrasi *clustering* dapat dilihat pada gambar 2.2.

### 2.4.1 Dokumen Clustering

Salah satu aplikasi *clustering* adalah *document clustering*. Adapun tujuan dari *document clustering* adalah untuk mengelompokkan dokumen dimana *cluster* dalam satu kelompok mirip satu dengan yang lain berdasarkan frekuensi kemunculan kata yang ada di setiap dokumen.



**Gambar 2.2** Clustering

### 2.4.2 Jenis-jenis Clustering

Metode *clustering* dibagi menjadi dua yaitu *hierarchical clustering* dan *nonhierarchical clustering*. *Hierarchical clustering* adalah pengelompokan dokumen menjadi kelompok-kelompok yang mempunyai suatu hirarki yaitu dari suatu kelompok kecil menuju ke kelompok besar atau sebaliknya. *Nonhierarchical clustering* adalah pengelompokan dengan membaca dokumen kemudian mengelompokkan berdasarkan kedekatannya secara langsung (Manning, 2006).

*Nonhierarchical clustering* mempunyai 2 macam algoritma yaitu (Steinbach, 2003) :

a. Metode *Single-pass*

Algoritma *Single-pass* ini melewati data dari suatu kumpulan data sekali saja untuk menciptakan *cluster*.

b. Metode *K-means*

Algoritma *K-means* ini melewati data berulang kali untuk mengadakan pengujian terhadap pusat *cluster* dan menentukan pengelompokan yang terbaik. Algoritma ini mendefinisikan terlebih dahulu berapa jumlah *cluster* yang diinginkan.

Pemilihan metode *clustering* yang tepat dilakukan dengan membandingkan metode *clustering* yang ada. Berikut ini adalah perbandingan antara metode *nonhierarchical clustering* dan *hierarchical clustering* (Teknomo, 2005).

a. *Hierarchical clustering*

- Jarak dihitung untuk semua variabel, sehingga dapat menunjukkan relasi antar semua variable

- Kurang handal dalam menangani jumlah data yang besar karena harus menghitung jarak antar data satu per satu.
  - Penggunaan waktu yang cukup banyak dengan jumlah anggota yang besar
- b. *Non – hierarchical clustering*
- Algoritma lebih cepat karena tidak melakukan perhitungan jarak terhadap semua variabel.
  - Tidak menunjukkan semua relasi antar variabel yang disebabkan karena jarak hanya dihitung dari titik ke pusat *cluster*.
  - Lebih handal dalam menangani jumlah data yang besar.

### 2.4.3 Hierarchical Clustering

Dengan metode ini, data tidak langsung dikelompokkan ke dalam beberapa *cluster* dalam 1 tahap, tetapi dimulai dari 1 *cluster* yang mempunyai kesamaan, dan berjalan seterusnya selama beberapa iterasi, hingga terbentuk beberapa *cluster* tertentu (Ridho dan Arai, 2004).

Arah *hierarchical clustering* dibagi 2, yaitu :

a. *Divisive*

- Dari 1 *cluster* ke *k cluster*
- Pembagian dari atas ke bawah (*top to down division*)

b. *Agglomerative*

- Dari *N cluster* ke *k cluster*
- Penggabungan dari bawah ke atas (*down to top merge*).

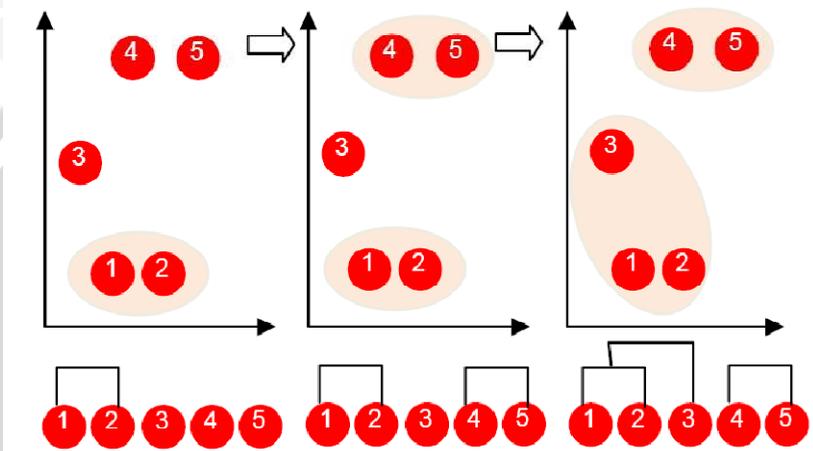
Algoritma *Hierarchical clustering* :

- Menentukan *k* sebagai jumlah *cluster* yang ingin dibentuk
- Setiap data dianggap sebagai *cluster*. Kalau  $N$  = jumlah data dan  $c$  = jumlah *cluster*, berarti ada  $c = N$ .
- Menghitung jarak antar *cluster*
- Cari 2 *cluster* yang mempunyai jarak antar *cluster* yang paling minimal dan gabungkan (berarti  $c=c-1$ ).
- Jika  $c > k$ , kembali ke langkah 3.

Penghitungan jarak antar objek, maupun antar *clusternya* dilakukan dengan *Euclidian distance*, khususnya untuk data numerik (Ridho dan Arai, 2004). Untuk data 2 dimensi, digunakan persamaan 2.1 :

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

Ilustrasi algoritma *Hierarchical Clustering* dapat dilihat pada gambar 2.3.



**Gambar 2.3** Ilustrasi Algoritma *Hierarchical Clustering*

Algoritma *hierarchical clustering* banyak diaplikasikan pada metode peng-*clusteran* berikut :

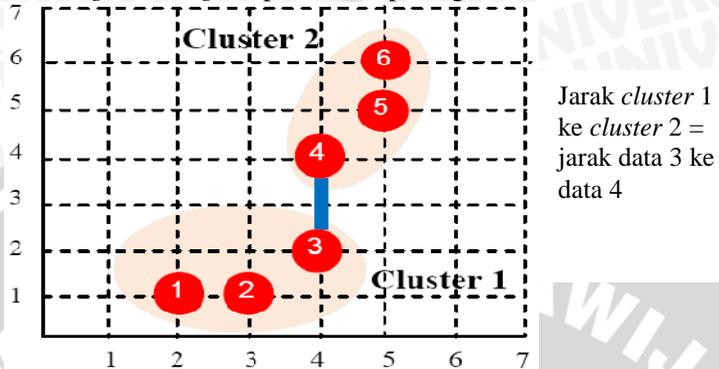
a. *Single Linkage Hierarchical Method* (SLHM)

*Single Linkage* adalah proses peng-*clusteran* yang didasarkan pada jarak terdekat antar objeknya (*minimum distance*) (Karypis, 1999).

Metode SLHM sangat bagus untuk melakukan analisa pada tiap tahap pembentukan *cluster*. Algoritma *Single Linkage Hierarchical Method* :

- diasumsikan setiap data dianggap sebagai *cluster*. Kalau  $n$ =jumlah data dan  $c$ =jumlah *cluster*, berarti ada  $c=n$ .
- menghitung jarak antar *cluster* dengan *Euclidian distance*.
- mencari 2 *cluster* yang mempunyai jarak antar *cluster* yang paling minimal dan digabungkan (*merge*) kedalam *cluster* baru (sehingga  $c=c-1$ )
- kembali ke langkah 3, dan diulangi sampai dicapai *cluster* yang diinginkan.

Ilustrasi *Single Linkage* dapat dilihat pada gambar 2.4.



**Gambar 2.4** Ilustrasi *Single Linkage*

b. *Centroid Linkage Hierarchical Method*

*Centroid Linkage Hierarchical Method* *Centroid Linkage* adalah proses pengclusteran yang didasarkan pada jarak antar centroidnya (Ridho, 2006). Metode ini bagus untuk memperkecil *variance within cluster* karena melibatkan centroid pada saat penggabungan antar *cluster*. Metode ini juga baik untuk data yang mengandung outlier.

Algoritma *Centroid Linkage Hierarchical Method* :

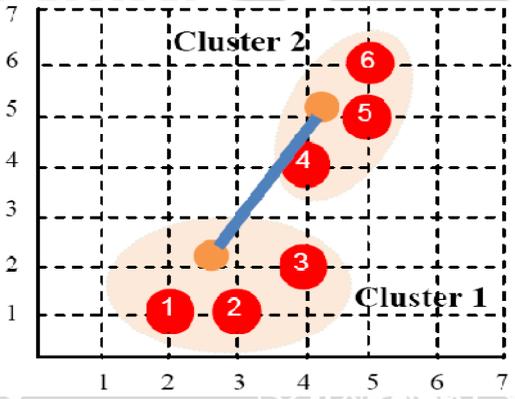
- Diasumsikan setiap data dianggap sebagai *cluster*. Kalau  $n$ =jumlah data dan  $c$ =jumlah *cluster*, berarti ada  $c=n$ .
- Menghitung jarak antar *cluster* dengan *Euclidian distance*.
- Mencari 2 *cluster* yang mempunyai jarak centroid antar *cluster* yang paling minimal dan digabungkan (*merge*) ke dalam *cluster* baru (sehingga  $c=c-1$ )
- Kembali ke langkah 3, dan diulangi sampai dicapai *cluster* yang diinginkan.

Ilustrasi *Centroid Linkage* dapat dilihat pada gambar 2.5.

c. *Complete Linkage Hierarchical Method* *Complete Linkage*

*Complete Linkage Hierarchical Method* *Complete Linkage* adalah proses pengclusteran yang didasarkan pada jarak terjauh antar objeknya (*maksimum distance*) (Ridho, 2006). Metode ini baik untuk kasus *clustering* dengan normal *data set distribution*.

Akan tetapi, metode ini tidak cocok untuk data yang mengandung outlier.



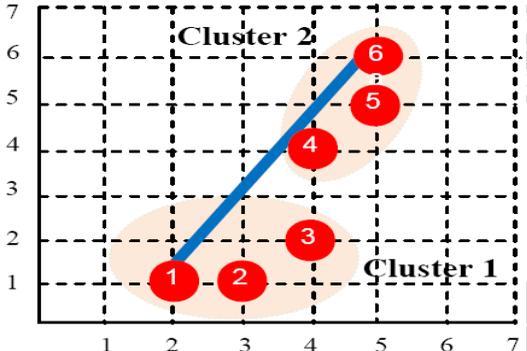
Jarak cluster 1 ke cluster 2 =  
 Jarak centroid cluster 1 ke centroid cluster 2

Gambar 2.5 Ilustrasi Centroid Linkage

Algoritma Complete Linkage Hierarchical Method :

- Diasumsikan setiap data dianggap sebagai cluster. Kalau  $n$ =jumlah data dan  $c$ =jumlah cluster, berarti ada  $c=n$ .
- Menghitung jarak antar cluster dengan Euclidian distance.

Ilustrasi Complete Linkage dapat dilihat pada gambar 2.6.



Jarak cluster 1 ke cluster 2 =  
 Jarak data 1 ke data 6

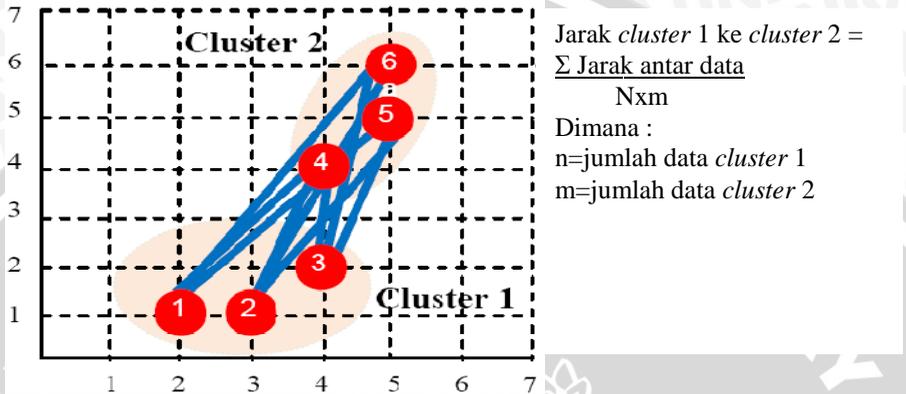
Gambar 2.6 Ilustrasi Complete Linkage

d. Average Linkage Hierarchical Method Average Linkage

Average Linkage Hierarchical Method Average Linkage adalah proses pengclusteran yang didasarkan pada jarak rata-rata antar objeknya ( average distance) (Ridho, 2006). Metode ini relatif yang terbaik dari metode-metode hierarchical. Namun, ini harus dibayar

dengan waktu komputasi yang paling tinggi dibandingkan dengan metode-metode *hierarchical* yang lain.

Ilustrasi *Average Linkage* dapat dilihat pada gambar 2.7.



Gambar 2.7 Ilustrasi *Average Linkage*

## 2.5 Implementasi Algoritma *Agglomerative Single Linkage Hierarchical Clustering*

Untuk menggambarkan algoritma linkage kita pandang jarak - jarak hipotetis antara 5 objek berikut ini (Hartini, 2009):

$$D = \{d_{ik}\} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{bmatrix} \end{matrix}$$

Dengan memperlakukan setiap objek sebagai cluster, kita memulai pengklasteran dengan menggabungkan dua item yang paling dekat. Karena  $\min(d_{ik}) = d_{53} = 2$ , objek 5 dan 3 digabung untuk membentuk cluster (35). Untuk memperoleh tingkat pengklasteran berikutnya, kita memerlukan jarak-jarak antara cluster

(35) dan objek – objek yang lain yang tersisa yaitu 1, 2 dan 4. Jarak – jarak yang berdekatan adalah:

$$d(35)1 = \min \{d_{31}, d_{51}\} = \min \{3, 11\} = 3$$

$$d(35)2 = \min \{d_{32}, d_{52}\} = \min \{7, 10\} = 7$$

$$d(35)4 = \min \{d_{34}, d_{54}\} = \min \{9, 8\} = 8$$

Dengan menghapus baris-baris dan kolom-kolom dari  $D$  yang bersesuaian dengan objek 3 dan 5 dan ading baris dan kolom untuk cluster (35), kita dapatkan matrik jarak yang baru

$$\begin{matrix} & 35 & 1 & 2 & 4 \\ 35 & \left[ \begin{array}{cccc} 0 & & & \\ 3 & 0 & & \\ 7 & 9 & 0 & \\ 8 & 6 & 5 & 0 \end{array} \right] \\ 1 & & & & \\ 2 & & & & \\ 4 & & & & \end{matrix}$$

Jarak terkecil antara pasangan-pasangan cluster sekarang adalah  $d(35)1 = 3$  dan kita gabung cluster (1) dengan cluster (35) untuk mendapatkan cluster berikutnya. Dengan menghitung

$$d(135)2 = \min \{d(35)2, d_{12}\} = \min \{7, 9\} = 7$$

$$d(135)4 = \min \{d(35)4, d_{14}\} = \min \{8, 6\} = 6$$

kita mendapatkan bahwa matrik jarak untuk tingkat pengklasteran berikutnya adalah:

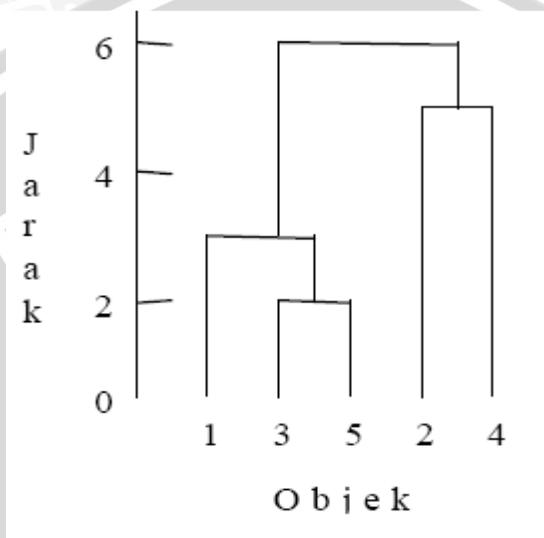
$$\begin{matrix} & 135 & 2 & 4 \\ 135 & \left[ \begin{array}{ccc} 0 & & \\ 7 & 0 & \\ 6 & 5 & 0 \end{array} \right] \\ 2 & & & \\ 4 & & & \end{matrix}$$

Jarak terdekat yang paling kecil antara pasangan cluster adalah  $d_{42} = 5$ , dan kita gabung objek 4 dan 2 untuk mendapatkan cluster (24). Pada saat ini kita punya 2 cluster yang berlainan, (135) dan (24). Jarak terdekat mereka adalah  $d(135)24 = \min \{d(135)2, d(135)4\} = \min \{7, 6\} = 6$  matriks jarak yang terakhir adalah:

$$\begin{matrix} & 135 & 24 \\ 135 & \left[ \begin{array}{cc} 0 & \\ 6 & 0 \end{array} \right] \\ 24 & & \end{matrix}$$

Jadi cluster (135) dan (24) digabung membentuk cluster tunggal dari semua 5 objek, (12345), sewaktu jarak terdekat mencapai 6.

Dendrogram yang menggambarkan pengklasteran hitarki (pengelompokan dan tingkat-tingkat jarak yang diperoleh). digambarkan pada gambar 2.8.



**Gambar 2.8** Single linkage dendrogram untuk jarak antara lima objek

## 2.6 Ukuran kualitas Dokumen *Clustering*

Untuk mengevaluasi secara manual kesamaan diantara dokumen dalam *cluster-cluster* yang telah dikelompokkan digunakan standar sebagaimana Tabel 2.6. Tabel tersebut berisi berbagai kemungkinan hasil klasifikasi pada tiap *event* (*Per Event contingency table*) (Yiming, 1999).

**Tabel 2.6** Kategori hasil klasifikasi

	In event	Not In event
In cluster	a	b
Not In Cluster	c	d

Tabel 2.6 menunjukkan bahwa hasil klasifikasi adakalanya memang termasuk event (a) yang dimaksud dan adakalanya tidak (b). Sedangkan dokumen yang tidak termasuk dalam hasil klasifikasi

suatu *event*, adakalanya memang bukan anggota *event* itu (d) dan adakalanya ternyata seharusnya menjadi anggota *event* tersebut (c). Dalam hal ini, keempat parameter di atas digunakan untuk menghitung 2 parameter evaluasi, yakni :

a. *Recall*, yakni tingkat keberhasilan mengenali suatu *event* dari seluruh *event* yang seharusnya dikenali. Rumusnya adalah  $r = a/(a+c)$  untuk  $a+c > 0$  2.2

b. *Precision*, yakni tingkat ketepatan hasil klasifikasi terhadap suatu *event*. Artinya, dari seluruh dokumen hasil klasifikasi, berapa persenkah yang dinyatakan benar. Rumusnya adalah  $p = a/(a+b)$  jika  $a+b > 0$  2.3

c. Sedangkan *F1 measure* merupakan gabungan antara *recall* dan *precision* yang dapat didefinisikan sebagai berikut :

$$F_1 \text{ measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad 2.4$$



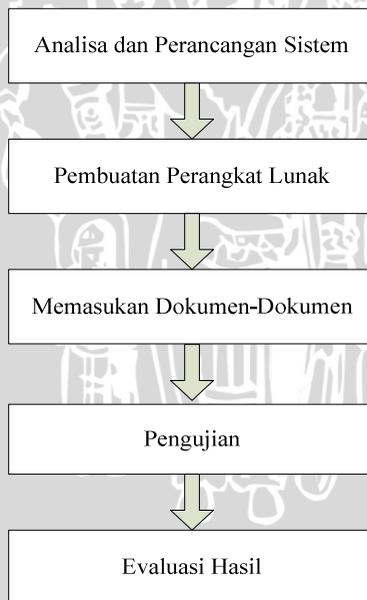
### BAB III METODOLOGI DAN PERANCANGAN

Pada BAB III ini akan dibahas metode dan perancangan yang digunakan, rancangan yang digunakan dan langkah – langkah yang dilakukan dalam penelitian klasifikasi dokumen berita dengan *Agglomerative Single Linkage Hierarchical Clustering*.

Penelitian dilakukan dengan langkah – langkah sebagai berikut :

1. Menganalisa dan merancang sistem yang akan digunakan untuk klasifikasi dokumen
2. Membuat perangkat lunak berdasarkan analisa dan perancangan yang telah dilakukan.
3. Memasukkan dokumen – dokumen yang bersumber dari *website* koran tertentu pada perangkat lunak yang dibuat.
4. Pengujian metode yang digunakan.
5. Evaluasi hasil

Langkah – langkah penelitian ini dapat dilihat pada gambar 3.1.



**Gambar 3.1** Diagram alir penelitian

### 3.1 Perancangan Sistem

#### 3.1.1 Deskripsi Umum Sistem

Sistem yang akan dibuat merupakan sistem untuk mengklasifikasikan suatu berita berbahasa Indonesia dengan menginputkan sejumlah data dalam sistem dan ditentukan banyaknya *cluster* yang akan dibentuk. Secara garis besar sistem memiliki beberapa proses, antara lain *preprocessing* yang terdiri dari proses *case folding*, *parsing*, *stemming*, penghilangan *stopword*, dan proses *clustering* data dengan metode *agglomerative hierarchical clustering* dengan penghitungan jarak terdekat (*Single Linkage Hierarchical Method*).

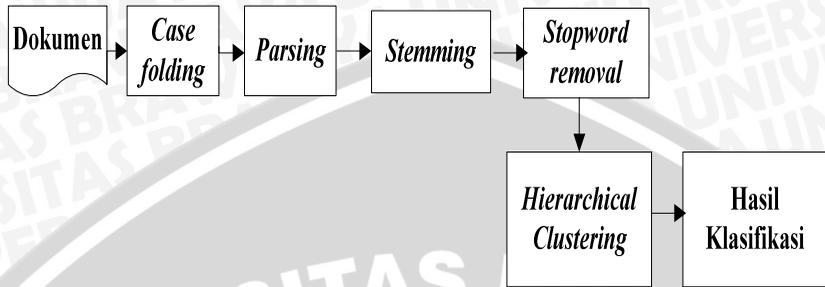
Proses sistem secara umum adalah sebagai berikut :

1. Proses pemasukan data dan menentukan jumlah *cluster* yang dilakukan oleh *user*.
  2. Proses *case folding*, yaitu mengubah semua huruf yang terdapat dalam dokumen menjadi huruf kecil.
  3. Proses *parsing*, yaitu memilah dokumen menjadi per kata.
  4. Proses *stemming*, yaitu melakukan *stemming* terhadap kata – kata dalam dokumen yang digunakan.
  5. Proses penghilangan *stopword*.
  6. Proses klasifikasi dokumen dengan menggunakan salah satu algoritma *hierarchical clustering* yaitu *agglomerative hierarchical clustering* dengan penghitungan jarak terdekat (*Single Linkage Hierarchical Method*).
  7. Hasil akhir adalah data yang terklusterisasi pada *cluster* tertentu.
- Blok gambar sistem dapat ditunjukkan pada gambar 3.2.

#### 3.1.2 Batasan Sistem

Batasan dari sistem yang dibuat adalah :

1. Sistem hanya mengklasifikasikan berita berbahasa Indonesia.
2. Tidak dilakukan pemisahan struktur kalimat, semua kata dalam kalimat dianggap mempunyai kedudukan yang sama.
3. Proses *stemming* yang dilakukan adalah proses *stemming* sederhana dan tidak secara mendalam.
4. Sistem tidak membedakan bobot judul atau bobot isi dalam dokumen.



**Gambar 3.2** Blok diagram proses sistem

### 3.2 Proses Klasifikasi

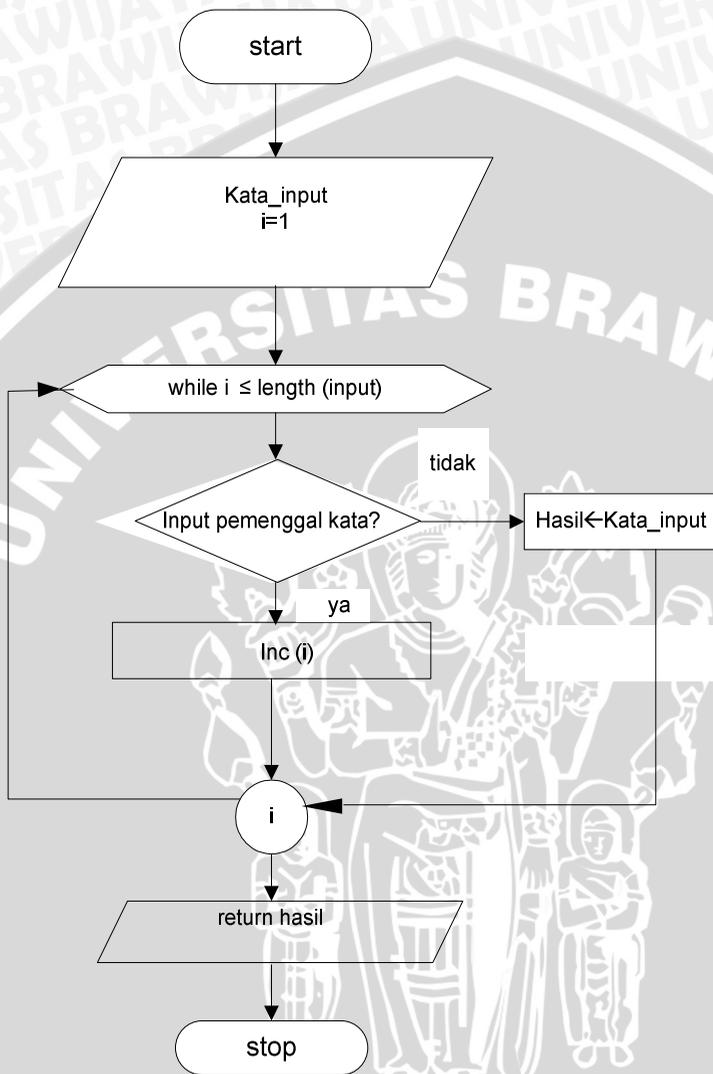
Dalam proses klasifikasi terdapat 2 tahapan yang dilakukan, yaitu *preprocessing* dan proses klasifikasi dengan cara *clustering* data. *Clustering* data dilakukan dengan menggunakan algoritma *agglomerative hierarchical clustering* dengan penghitungan jarak terdekat (*Single Linkage Hierarchical Method*).

#### 3.2.1 *Preprocessing*

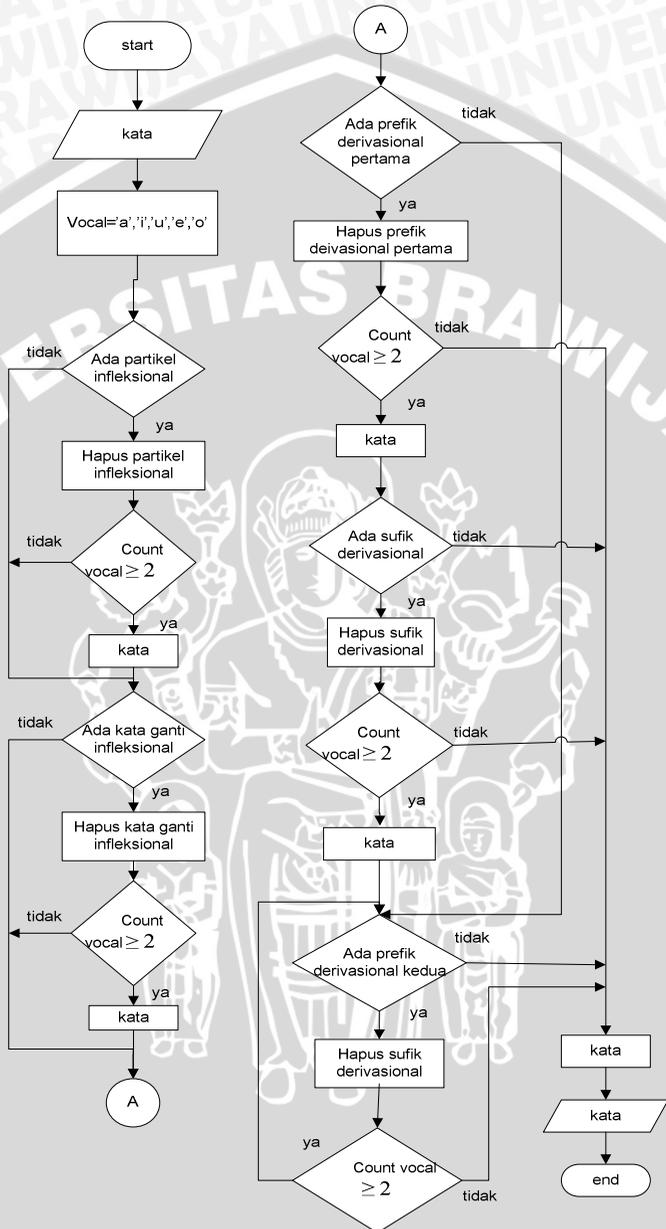
Pada tahapan *preprocessing*, dokumen – dokumen yang di inputkan dijadikan sebagai kumpulan kata dan pemisah kata. kata adalah kumpulan huruf alphabet, sedangkan tanda baca, angka, dan karakter selain huruf alphabet dianggap sebagai *delimiter* atau pemisah antar kata (Porter, 1980).

Pada tahapan ini, terdapat proses *case folding*, *parsing*, *stemming*, dan *stopword removal*. *Case folding* adalah pengubahan semua huruf ke dalam huruf kecil, sedangkan *parsing* sederhana memecah dokumen text menjadi per unit kata, sehingga hasil dari parsing ini adalah berupa kumpulan kata. Kata hasil dari *parsing* akan melewati proses *stemming*. Selanjutnya hasil *stemming* akan melewati proses *stopword removal*, sehingga akan sangat mengurangi kata – kata yang akan diproses pada tahapan selanjutnya.

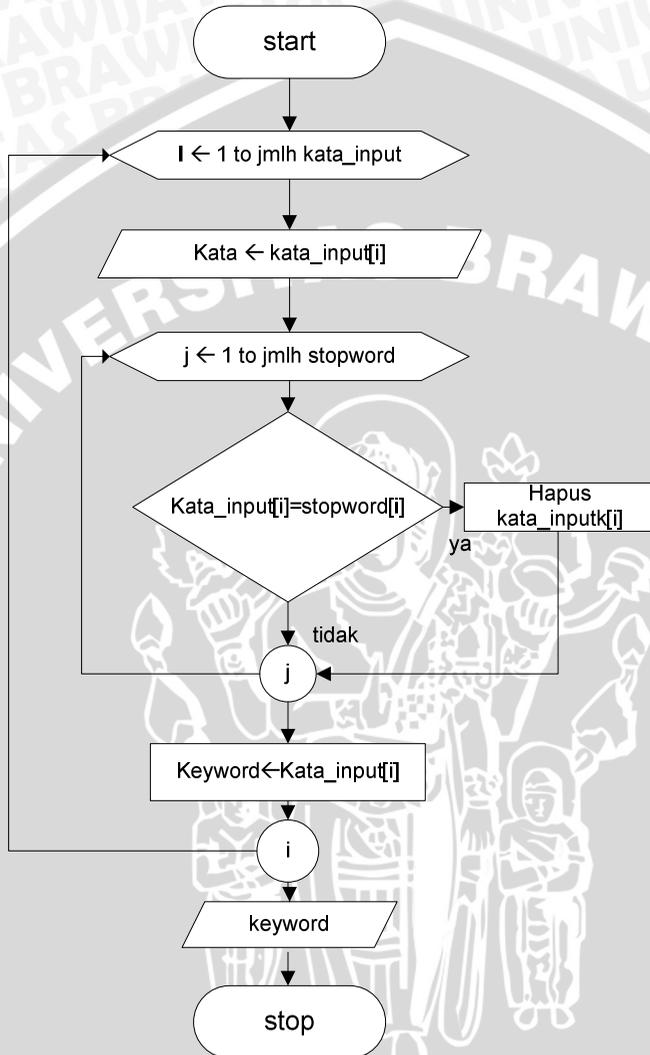
*Flowchart parsing*, *stemming*, dan *stopword removal*, dapat dilihat pada gambar 3.3, gambar 3.4, dan gambar 3.5.



**Gambar 3.3** Flowchart parsing



Gambar 3.4 Flowchart proses stemming



**Gambar 3.5** Flowchart penghilangan stopwords

Aturan *stemming* dilakukan secara sederhana dengan membagi penanganan terhadap partikel infleksional dan derivasional.

Proses-proses yang ada dalam proses *stemming* secara garis besar hanya dibagi menjadi 4 macam, yaitu sufiks infleksional, peleburan

derivasional, prefiks derivasional, dan sufiks derivasional. Proses *stemming* yang digunakan dibagi dalam 7 proses, yaitu:

1. Proses partikel sufiks infleksional menangani partikel sufiks infleksional meliputi “-kah”, “-lah”, “-pun”.
2. Proses kata ganti infleksional menangani kata ganti sufiks infleksional meliputi “-ku”, “-mu”, “-nya”.
3. Proses imbuhan prefiks derivasional pertama menangani prefiks derivasional pertama secara langsung meliputi ‘meng-’, ‘men-’, ‘mem-’, ‘me-’, ‘peng-’, ‘pen-’, ‘pem-’, ‘di-’, ‘ter-’, ‘ke’ dan prefix derivasional yang mengalami peleburan, meliputi sebagai berikut:

**Tabel 3.1** Aturan peleburan

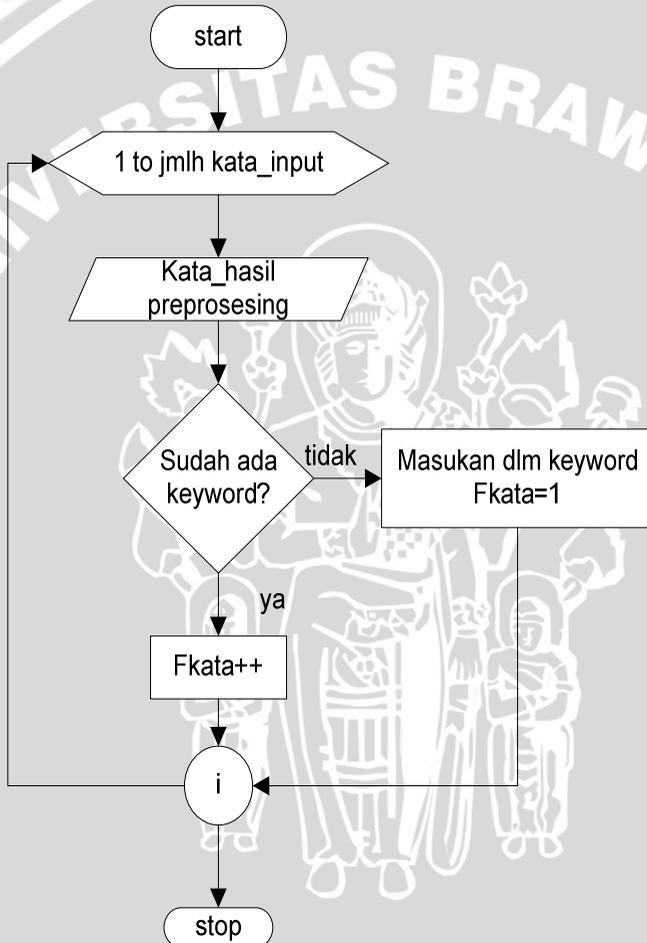
NO	Prefiks	Kondisi	Lebur	Contoh
1	mem pem	+vokal	P	memaku→paku
2	meny peny	+vokal	S	menyapu→sapu
3	men pen	+vokal	T	menari→tari

4. Proses sufiks derivasional menangani penghilangan seluruh sufiks, derivasional meliputi “-kan”, “-i”, “-an”. Proses penghilangan sufiks derivasional ini dilakukan jika terdapat proses sebelumnya (penghilangan prefiks derivasional).
5. Proses prefiks derivasional kedua menangani prefiks derivasional kedua meliputi ‘ber-’, ‘bel-’, ‘be-’, ‘per-’, ‘pel-’, ‘pe-’. Proses penghilangan prefiks derivasional kedua ini dilakukan jika terdapat proses penghilangan sufiks derivasional.
6. Proses prefiks derivasional kedua secara langsung menangani prefiks derivasional kedua meliputi ‘ber-’, ‘bel-’, ‘be-’, ‘per-’, ‘pel-’, dan ‘pe-’.
7. Proses sufiks derivasional menangani penghilangan seluruh sufiks, derivasional meliputi “-kan”, “-i”, “-an”. Proses penghilangan sufiks derivasional ini dilakukan jika terdapat proses sebelumnya (penghilangan prefiks derivasional kedua secara langsung).

### 3.2.2 Pemrosesan dengan *Agglomerative Single Linkage Hierarchical Clustering*

Pada tahap ini dilakukan langkah – langkah untuk memproses data hasil *preprocessing*. Hasil proses *preprocessing* yang berupa kata

per kata unik dihitung frekuensinya tiap dokumen, penghitungan frekuensi merupakan proses utama dalam perhitungan hal ini dikarenakan nilai frekuensi inilah yang akan diproses sehingga didapat perhitungan untuk mengklasifikasikan dokumen – dokumen tersebut. Flowchart penghitungan frekuensi dapat dilihat pada gambar 3.6.



**Gambar 3.6** Flowchart penghitungan frekuensi

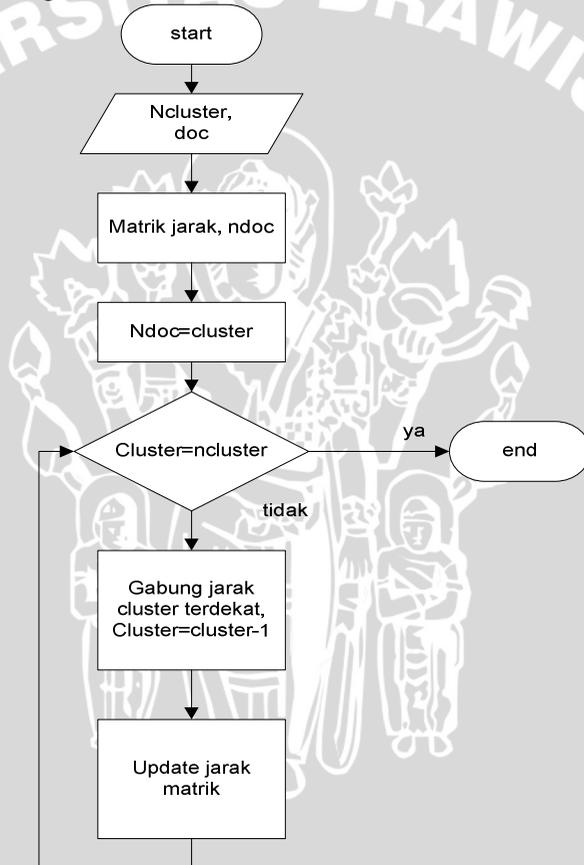
Proses dasar dari *Agglomerative Single Linkage hierarchical clustering* adalah:

1. Setiap item didefinisikan sebagai *cluster*, jadi data yang ada sejumlah  $N$  *items*, maka *cluster* yang ada sejumlah  $N$  item

tersebut. Jarak antara *clusters* sama dengan jarak antara item yang ada.

2. Selanjutnya, mencari pasangan *cluster* yang paling dekat dan menggabungkan keduanya dalam satu *cluster*.
3. Hitung jarak antara *cluster* yang baru dibentuk dengan *cluster* yang sudah ada sebelumnya. Ulangi tahap 2 dan 3 sampai *cluster* membentuk  $n$  *cluster* yang diinginkan.

*Flowchart Agglomerative Single Linkage hierarchical clustering* dapat dilihat pada gambar 3.7.



**Gambar 3.7** *Flowchart Agglomerative Single Linkage hierarchical clustering*

### 3.4 Implementasi Perhitungan

Sebagai contoh implementasi, terdapat 3 dokumen yang akan diklasifikasikan. 3 dokumen ini akan dibentuk sebanyak 2 *cluster*, maka  $n=2$ .

#### K Antisipasi.txt

Antisipasi Flu Babi, WNA Diperiksa

**BALIKPAPAN, KOMPAS.com** – Sebanyak 46 penumpang pesawat Silk Air dengan nomor penerbangan MI 284 yang datang dari Singapura, pada hari Selasa (28/4), setiba di Bandara Sepinggan, Balikpapan, Kalimantan Timur, langsung menjalani pemeriksaan kesehatan. Pemeriksaan penumpang yang dilakukan oleh petugas kesehatan pelabuhan di bandara internasional tersebut sebagai antisipasi terkait serangan flu babi dan flu singapura.

Pemeriksaan di bandara itu dilakukan dengan menggunakan alat pengukur suhu badan yang dimasukkan lubang telinga setiap penumpang. Hasil dari pemeriksaan itu diketahui tidak ada penumpang yang bersuhu badan di atas 39 derajat celsius. "Pemeriksaan ini sesuai dengan perintah dalam surat edaran departemen kesehatan terkait dengan adanya serangan flu babi dan flu singapura," ujar Kristanto Sutopo, Surveilans Epimeologi Kantor Kesehatan Pelabuhan Bandara Sepinggan.

Menurut Kristanto, pemeriksaan ini akan berlangsung sampai ada ada pemberitahuan dari pemerintah pusat bahwa serangan penyakit tidak terjadi di Indonesia. Di Bandara Sepinggan, pemeriksaan hanya dilakukan terhadap warga yang datang dari luar negeri. "Sedangkan penumpang dalam negeri tidak dilakukan," katanya.

#### KAir susu.txt

#### Laporan wartawan KOMPAS Evy Rachmawati

**KOMPAS.com** - Air Susu Ibu (ASI) kaya akan berbagai nutrisi penting juga faktor-faktor yang memperkuat imun tubuh yang dibutuhkan . Pemberian air susu ibu secara eksklusif yang diberikan enam bulan pertama menyusui menurut dokter spesialis anak dr Jeanne Roos Tikoalu akan mempengaruhi pertumbuhan dan perkembangan optimal bayi di masa berikutnya.

Selain memberikan perlindungan terhadap infeksi dan alergi, ASI juga akan merangsang pertumbuhan sistem kekebalan tubuh bayi, mengandung komponen anti inflamasi dan antioksidan. Karena itu, dengan memberikan ASI, ibu memberikan yang terbaik untuk kekebalan dan kesehatan si kecil.

B85 investor.txt

**JAKARTA, KOMPAS.com** - Dari 85 investor yang mendirikan perusahaan di kawasan industri Karawang International Industrial City (KIIC), Kabupaten Karawang, Jawa Barat, sebanyak 95 persen merupakan penanaman modal asing (PMA) Jepang.

Jumlah investor itu dipastikan akan terus bertambah pada beberapa tahun mendatang, karena masih banyak lahan kosong yang merupakan kavling industri di areal sekitar KIIC tersebut.

"Sampai sekarang, kami masih menawarkan kavling industri ke setiap investor," kata Humas PT Maligi, Pengelola kawasan industri KIIC, Bambang Sugeng, seperti dikutip Antara di Karawang, Minggu (17/5).

Dikatakannya, dari 800 hektar kavling industri siap pakai di sekitar kawasan industri KIIC, sampai saat ini baru 70 persen yang terjual. Sisanya hingga kini masih menunggu investor.

Selain itu, masih ada areal seluas 400 hektar di kawasan industri KIIC yang belum dibangun sebagai kavling industri. Rencananya, luas lahan itu akan dibangun kavling industri oleh PT Karawang Tata Bina Industrial Estate.

"Untuk 800 hektar kavling industri yang sudah siap pakai, sebelumnya dikembangkan oleh PT Maligi Permata Industrial Estate dan PT Harapan Anang Bakrie Ensang. Jadi, total luas lahan di KIIC itu 1.200 hektar," katanya.

Rata-rata, jenis industri di kawasan industri KIIC Karawang itu bergerak dibidang otomotif dan elektronik. Di Karawang sendiri, terdapat 158 perusahaan PMA, atau mencapai sekitar 40 persen dari total 396 industri.

d1= K Antisipasi.txt

d2= KAir susu.txt

d3= B85 investor.txt

Setelah ketiga dokumen tersebut dilakukan text *preprocessing* dan penghitungan frekuensi kata didapat hasil seperti tabel 3.9.

**Tabel 3.2** Hasil *preprocessing* dokumen

	fd0	fd1	fd2	$f(d0-d1)^2$	$f(d0-d1)^2$	$f(d1-d2)^2$
air	0	1	2	1	4	1
alat	0	1	0	1	0	1
alergi	0	0	1	0	1	1
anak	0	0	1	0	1	1

anang	1	0	0	1	1	0
andung	0	0	1	0	1	1
anti	0	0	1	0	1	1
antioksidan	0	0	1	0	1	1
antisipasi	0	2	0	4	0	4
areal	2	0	0	4	4	0
asi	0	0	3	0	9	9
asing	1	0	0	1	1	0
babi	0	3	0	9	0	9
badan	0	2	0	4	0	4
baik	0	0	1	0	1	1
bakrie	1	0	0	1	1	0
balikpapan	0	2	0	4	0	4
bambang	1	0	0	1	1	0
bandara	0	5	0	25	0	25
bangun	2	0	0	4	4	0
bayi	0	0	3	0	9	9
beri	0	0	1	0	1	1
bidang	1	0	0	1	1	0
bina	1	0	0	1	1	0
butuh	0	0	1	0	1	1
capa	1	0	0	1	1	0
celsius	0	1	0	1	0	1
city	1	0	0	1	1	0
departemen	0	1	0	1	0	1
derajat	0	1	0	1	0	1
diri	1	0	0	1	1	0
dokter	0	0	1	0	1	1
dr	0	0	1	0	1	1
edaran	0	1	0	1	0	1
eksklusif	0	0	1	0	1	1
elektronik	1	0	0	1	1	0

elola	1	0	0	1	1	0
enam	0	0	1	0	1	1
ensang	1	0	0	1	1	0
epimeologi	0	1	0	1	0	1
estate	2	0	0	4	4	0
evy	0	0	1	0	1	1
faktor	0	0	2	0	4	4
flu	0	5	0	25	0	25
gerak	1	0	0	1	1	0
harapan	1	0	0	1	1	0
hektar	4	0	0	16	16	0
humas	1	0	0	1	1	0
iksa	0	7	0	49	0	49
imun	0	0	1	0	1	1
industri	13	0	1	169	144	1
industrial	3	0	0	9	9	0
infeksi	0	0	1	0	1	1
inflamasi	0	0	1	0	1	1
intah	0	1	0	1	0	1
internasional	0	1	0	1	0	1
international	1	0	0	1	1	0
investor	4	0	1	16	9	1
itahu	0	1	0	1	0	1
jalan	0	1	0	1	0	1
jawa	1	0	0	1	1	0
jeanne	0	0	1	0	1	1
jenis	1	0	0	1	1	0
jepang	1	0	1	1	0	1
jual	1	0	0	1	1	0
kabupaten	1	0	0	1	1	0

kait	0	2	0	4	0	4
kalimantan	0	1	0	1	0	1
kantor	0	1	0	1	0	1
karawang	6	0	1	36	25	1
kavling	6	0	0	36	36	0
kawasan	5	0	1	25	16	1
kaya	0	0	1	0	1	1
kebal	0	0	2	0	4	4
kembang	1	0	2	1	1	4
kiic	7	0	0	49	49	0
komponen	0	0	1	0	1	1
kosong	1	0	0	1	1	0
kristanto	0	2	0	4	0	4
kutip	1	0	0	1	1	0
labuh	0	2	0	4	0	4
lahan	3	0	0	9	9	0
laporan	0	0	1	0	1	1
lindung	0	0	1	0	1	1
luas	2	0	0	4	4	0
lubang	0	1	0	1	0	1
maligi	2	0	0	4	4	0
masuk	0	1	0	1	0	1
mata	1	0	0	1	1	0
mi	0	1	0	1	0	1
modal	1	0	0	1	1	0
negeri	0	2	0	4	0	4
ngaruh	0	0	1	0	1	1
nomor	0	1	0	1	0	1
nutrisi	0	0	1	0	1	1
optimal	0	0	1	0	1	1
otomotif	1	0	0	1	1	0
pakai	2	0	0	4	4	0

periksa	0	1	0	1	0	1
perintah	0	1	0	1	0	1
perkuat	0	0	1	0	1	1
persen	3	0	1	9	4	1
pma	2	0	0	4	4	0
pt	4	0	0	16	16	0
pusat	0	1	0	1	0	1
rachmawati	0	0	1	0	1	1
rangsang	0	0	1	0	1	1
rencana	0	0	1	0	1	1
roos	1	0	0	1	1	0
sakit	0	1	0	1	0	1
sawat	0	1	0	1	0	1
sehat	0	4	1	16	1	9
seluas	1	0	0	1	1	0
sepinggan	0	3	0	9	0	9
serangan	0	3	0	9	0	9
si	0	0	1	0	1	1
silk	0	1	0	1	0	1
singapura	0	3	0	9	0	9
sisa	1	0	0	1	1	0
sistem	0	0	1	0	1	1
spesialis	0	0	1	0	1	1
sugeng	1	0	0	1	1	0
suhu	0	2	0	4	0	4
surat	0	1	0	1	0	1
survailan	0	1	0	1	0	1
susu	0	0	3	0	9	9
sutopo	0	1	0	1	0	1
tambah	1	0	0	1	1	0

tanam	1	0	0	1	1	0
tata	1	0	0	1	1	0
tawar	1	0	0	1	1	0
telinga	0	1	0	1	0	1
terbang	0	1	0	1	0	1
tikoalu	0	0	1	0	1	1
total	2	0	0	4	4	0
tubuh	0	0	2	0	4	4
tugas	0	1	0	1	0	1
tumbuh	0	0	3	0	9	9
tumpang	0	5	0	25	0	25
tunggu	1	0	0	1	1	0
ukur	0	1	0	1	0	1
usaha	2	0	0	4	4	0
warga	0	1	0	1	0	1
wartawan	0	0	1	0	1	1
wna	0	1	0	1	0	1

Setiap dokumen yang diinput merupakan anggota dari sebuah *N cluster*. Selanjutnya dari data diatas akan dibentuk matriks dengan menghitung jarak antara dokumen yang ada. Perhitungan jarak dengan menggunakan fungsi *Euclidean Distance* pada rumus 2.1.

$$d(d_0,d_1) = \sqrt{702} \\ = 26.4953$$

$$d(d_1,d_3) = \sqrt{490} \\ = 22.1359$$

$$d(d_2,d_3) = \sqrt{324} \\ = 18$$

Dari hasil perhitungan diatas maka akan dibentuk matriks, jika perhitungan  $d(i,i)$  maka nilai matrik diisi dengan 0. Setelah didapatkan hasil jarak, maka nilai yang minimum antara 2 dokumen akan digabung menjadi 1 *cluster*.

**Tabel 3.3** Matrik dokumen

	d(d0)	d(d0)	d(d1)
d(d0)	0	26.4953	22.1359
d(d1)		0	18
d(d2)			0

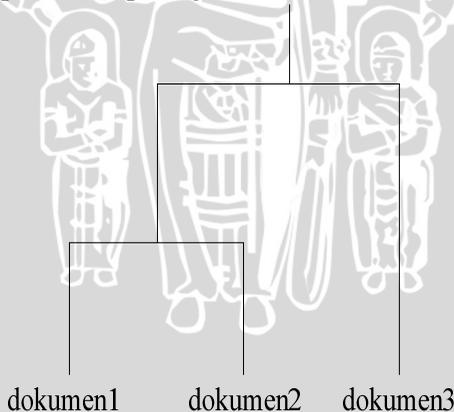
$$d(1,2)3 = \min\{d13,d23\} = d23 = 18$$

Pada iterasi pertama ini, didapatkan hasil minimum merupakan hasil perhitungan antara d1 dan d2. Keduanya digabung dalam satu *cluster*, maka didapatkan matriks yang baru. Dengan demikian kedua dokumen digabung dalam satu *cluster*, sehingga didapatkan hasil akhir sebagai berikut :

**Tabel 3.4** Matrik iterasi 1

	d(d0)	d(d1,2)
d(d0)	0	22.1359
d(d1,2)		0

Dapat diketahui dari hasil akhir n=2 maka iterasi selesai. Hasil dari *clustering* dapat dilihat pada gambar 3.8.



**Gambar 3.8** dendogram hasil klasifikasi

### 3.4.1 Ukuran kualitas Dokumen *Clustering*

Dari hasil perhitungan di atas diketahui :

$$a=2; b=0; c=0; d=1$$

Dapat dihitung nilai *recall*nya :

$$r = \frac{2}{2+0} \\ = 1$$

Dapat dihitung nilai *precision*nya :

$$p = \frac{2}{(2+0)} \\ = 1$$

Dapat dihitung nilai  $F_1$  *measure*nya :

$$= \frac{2 \times 1 \times 1}{1+1} \\ = 1$$

Dari hasil perhitungan di atas didapat nilai *recall*, *precision*, dan  $F_1$  *measure*nya 1 (100%).

### 3.5 Rancangan Uji Coba

Evaluasi hasil ini dilakukan terhadap dokumen berita yang diambil dari [www.kompas.com](http://www.kompas.com), yang terdiri dari 120 dokumen. Dokumen ini diambil dari 8 kategori yang berbeda, dokumen tersebut terdiri dari 15 dokumen ekonomi, 15 dokumen hiburan, 15 dokumen kesehatan, 15 dokumen olahraga, 15 dokumen otomotif, 15 dokumen pendidikan, 15 dokumen politik, dan 15 dokumen teknologi. Masing – masing nama dokumen ditandai berdasarkan kategorinya sehingga memudahkan untuk evaluasi efektifitas hasil sistem yang dihasilkan. Untuk mengetahui nilai efektivitas hasil dari sistem tersebut dihitung dengan menggunakan rumusan *recall*, *precision*, dan  $F_1$  *measure* (Bab 2.6). penghitungan *recall*, *precision*, dan  $F_1$  *measure* beberapa kali dengan merubah – ubah jumlah dokumen yang diklasifikasikan, yaitu 8 dokumen, 16 dokumen, 24 dokumen, 32 dokumen, 40 dokumen, 48 dokumen, 56 dokumen, 64 dokumen, 72 dokumen, 80 dokumen, 88 dokumen, 96 dokumen, 104 dokumen, 112 dokumen dan 120 dokumen. Berdasarkan *cluster* yang terbentuk, hasilnya akan dapat dilihat pada tabel 3.5.

**Tabel 3.5** Hasil uji coba

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>						
<i>cluster2</i>						
<i>cluster3</i>						
<i>cluster4</i>						
<i>cluster5</i>						
<i>cluster6</i>						
<i>cluster7</i>						
<i>cluster8</i>						



UNIVERSITAS BRAWIJAYA



## BAB IV IMPLEMENTASI DAN PEMBAHASAN

### 4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan dalam subbab ini adalah lingkungan implementasi perangkat keras dan perangkat lunak.

#### 4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem klasifikasi dengan algoritma *Agglomerative Single Linkage Hierarchical Clustering* ini adalah sebagai berikut:

1. Laptop dengan prosesor Intel® Core™ 2 Duo Prosesor T6400 @ 2.00GHz
2. Memori 2 GB
3. Harddisk dengan kapasitas 250 GB

#### 4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam sistem klasifikasi dengan algoritma *Agglomerative Single Linkage Hierarchical Clustering* ini adalah :

1. Sistem Operasi Windows XP SP 2
2. Borland Delphi 7
3. Text editor notepad

### 4.2 Struktur Data

Struktur data yang digunakan untuk menyimpan hasil *preprocessing* dokumen adalah berupa array of record, sebagai berikut :

```
type arrDoc = array of document;
```

**arrDoc** merupakan array dari record document. arrDoc inilah yang digunakan untuk menyimpan hasil dari *preprocessing* sistem.

```
type document = record  
    listWord : arrWord;  
    wordCount : integer;  
    docName : string;  
end;
```

**document** merupakan record yang terdiri dari listword yang bertipe arrWord, wordCount yang digunakan untuk menyimpan jumlah kata, dan docName yang digunakan untuk menyimpan nama setiap dokumen.

```
type arrWord = array of singleWord;
```

**arrWord** merupakan array dari record singleword.

```
type singleWord = record
  kata : string;
  frekuensi : integer;
end;
```

**singleword** merupakan record yang terdiri dari kata yang digunakan untuk menyimpan kata-kata dalam dokumen, dan frekuensi yang digunakan untuk menyimpan frekuensi setiap kata dalam dokumen.

Untuk menyimpan proses perhitungan pada sistem digunakan juga record. Ditunjukkan pada struktur data berikut:

```
type tabelDistance = record
  kata : TStringList;
  frekuensi : arr2DInt;
  deltaFrek : arr2DInt;
  totalDistance : arrInt;
  euclidDistance : arrReal;
end;
type arrInt = array of integer;
type arr2DInt = array of array of integer;
type arrReal = array of real;
```

Dari struktur data diatas dapat diketahui bahwa **tabelDistance** merupakan sebuah record yang mempunyai variabel kata yang bertipe TStringList, frekuensi yang bertipe bertipe **arr2DInt**, **arr2DInt** merupakan array dua dimensi bertipe integer, deltaFrek yang bertipe **arr2DInt**, **arr2DInt** merupakan array dua dimensi bertipe integer, totalDistance yang bertipe arrInt(array of integer), euclidDistance yang bertipe arrReal(array of real). Record ini digunakan untuk menampung perhitungan kata dan frekuensinya beserta hasil perhitungan dari kata tersebut yang berupa delta frekuensi antar dokumen, total jarak dari delta frekuensi antar dokumen, dan hasil perhitungan Euclidean distancenya.

```

type cellCluster = record
  rowCell : integer;
  colCell : integer;
  cluster : TStringList;
  euclidDist : real;
end;

```

Dari struktur data diatas dapat diketahui bahwa cellCluster merupakan sebuah record yang mempunyai variabel rowCell yang bertipe integer, colCell yang bertipe integer, cluster yang bertipe TStringList dan euclidDist bertipe real. Record ini digunakan untuk menampung nilai euclidDist antar cluster dan koordinatnya yang nantinya akan ditampilkan berupa tabel matrik *distance*.

```

type arrCell = array of cellCluster;
type tabelCluster = record
  dokCount : integer;
  dataCell : arrCell;
end;

```

Dari struktur data diatas dapat diketahui bahwa tabelCluster merupakan sebuah record yang mempunyai variabel docCount yang bertipe integer, dan dataCell yang bertipe ArrCell. Record ini digunakan untuk menampung matrik cluster beserta nilai - nilainya.

### 4.3 Implementasi Algoritma *Agglomerative Single Linkage Hierarchical Clustering*

Berdasarkan analisa dan perancangan proses sistem klasifikasi dengan algoritma *Hierarchical Clustering* dibagi menjadi dua bagian yaitu tahap *preprocessing* dan tahap klasifikasi itu sendiri.

#### 4.3.1 Implementasi *Preprocessing*

#### 4.3.2 Implementasi *Parsing dan Case Folding*

Tahap awal dari sistem pengklasifikasian ini adalah melakukan *parsing*. Berikut ini adalah fungsi untuk melakukan tahapan *parsing*.

```

function parseFile(text : string) : string;
var
  s : string;
  i, j : integer;
begin
  s := '';
  j := 0;
  for i := 1 to length(text) do

```

```

begin
  if (((ord(text[i])>=65) and (ord(text[i])<=90))
    or ((ord(text[i])>=97) and (ord(text[i])<=122))) then
    begin
      s := s + LowerCase(text[i]);
      j := 0;
    end
  else if (ord(text[i])=32) then
    begin
      j := j + 1;
      if j=1 then
        s := s + text[i];
      end
    else
      begin
        j := j + 1;
        if j=1 then
          s := s + ' ';
        end;
      end;
    end;

  Result := s;
end;

```

```

Strings := TStringList.Create;
SetLength(listWord,0);

Strings.Clear;
//diambil per kata
ExtractStrings(sep,wht,PChar(content), Strings);

```

Fungsi *parsing* di atas sekaligus merupakan fungsi *casefolding*, sehingga hasil dari fungsi *parsing* adalah berupa kata - kata yang terdiri dari kumpulan huruf saja dan dalam bentuk huruf kecil semua.

### 4.3.3 Implementasi *Stemming*

Setelah dilakukan proses *parsing*, maka setiap kata yang dihasilkan akan dicari bentuk dasarnya (*stemming*). Seperti telah dijelaskan pada subbab 3.2.1, ada 7 tahapan dalam proses *stemming*. Berikut adalah implementasi dari proses *stemming* sufiks infleksional:

```

//penghilangan sufiks infleksional
kata := StrLower(PChar(Strings.Strings[i]));
if ((AnsiEndsStr('lah', kata)) or
  (AnsiEndsStr('kah', kata)) or
  (AnsiEndsStr('pun', kata))) then begin
  len := StrLen(PChar(Strings.Strings[i])-3);
  if (countVocal(Copy(Strings.Strings[i], 0,len))>=2)
then
  Strings.Strings[i] := Copy (Strings.Strings[i],
0,len-3); end;

```

Potongan fungsi *stemming* diatas berfungsi untuk menangani partikel. Setiap kata yang berakhiran ‘kah’, ‘lah’, atau ‘pun’, akan melalui proses dibawah ini :

- Jika kata tersebut mengandung akhiran ‘lah’, ‘kah’, atau ‘pun’, maka fungsi *stemming* akan menghilangkan akhiran tersebut dan menghitung setelah akhiran dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya.

Berikut merupakan fungsi *stemming* kata ganti sufiks infleksional

```
//penghilangan kata ganti infleksional
kata := StrLower(Pchar(Strings.Strings[i]));
  if ((AnsiEndsStr('ku', kata)) or
      (AnsiEndsStr('mu', kata)) or
      (AnsiEndsStr('nya', kata))) then
    begin
      if (AnsiEndsStr('nya', kata)) then
len := StrLen(Pchar(Strings.Strings[i]))-3
    else
      len := StrLen(Pchar(Strings.Strings[i]))-2;
      if (countVocal(Copy (Strings.Strings[i], 0,len))>=2)
    then
      Strings.Strings[i] := Copy (Strings.Strings[i],
0,len);
    end;
```

Setiap kata yang berakhiran ‘ku’, ‘mu’, ‘nya’ akan melalui proses berikut :

- Jika kata tersebut mengandung akhiran ‘ku’, ‘mu’, atau ‘nya’ maka fungsi *stemming* akan menghilangkan akhiran tersebut dan menghitung setelah akhiran dihilangkan apakah jumlah huruf vocal lebih dari dua atau tidak, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya.

Berikut merupakan fungsi *stemming* prefiks derivasional tahap pertama :

Berikut merupakan fungsi *stemming* prefiks derivasional tahap pertama :

```
//meng peng
kata := StrLower(Pchar(input));
  if ((AnsiStartsStr('meng', kata)) or
      (AnsiStartsStr('peng', kata))) then
```

```

begin
  len := StrLen(Pchar(kata))-4;
  if (countVocal(Copy (kata, 5,len))>=2) then
  begin
    output := Copy (kata, 5,len);
    isfired := true;
  end;
end

```

Setiap kata yang berawalan 'meng', atau 'peng' akan melalui proses berikut :

- Jika kata tersebut berawalan 'meng', atau 'peng', maka fungsi *stemming* akan menghilangkan awalan tersebut dan menghitung setelah awalan dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya.

```

if ((AnsiStartsStr('meny', kata)) or
    (AnsiStartsStr('peny', kata))) then
begin
  len := StrLen(Pchar(kata))-3;
  if (countVocal(Copy (kata, 4,len))>=2) then
  begin
    kata := Copy (kata, 4,len);
    temp := Pchar(kata);
    temp[1] := 's';
    output := temp;
    isfired := true;
  end;
end;

```

Setiap kata yang berawalan 'meny' atau 'peny' akan melalui proses berikut :

- Jika kata tersebut berawalan 'meny', atau 'peny', maka fungsi *stemming* akan menghilangkan awalan tersebut dan menghitung setelah awalan dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya dan menambah huruf 's' pada awal kata tersebut.

```

if ((AnsiStartsStr('men', kata)) or
    (AnsiStartsStr('pen', kata))) then
begin
  len := StrLen(Pchar(kata))-2;
  if (countVocal(Copy (kata, 4,len))>=2) then
begin
  temp := Copy (kata, 3,len);
  if (temp[2] in vocal) then
    temp[1] := 't'

```

```

else
  temp := Copy(temp, 2, strlen(Pchar(temp)));
  output := temp;
  isfired := true;
end;

```

Setiap kata yang berawalan ‘men’ atau ‘pen’ akan melalui proses berikut :

- Jika kata tersebut berawalan ‘men’, atau ‘pen’, maka fungsi *stemming* akan menghilangkan awalan tersebut dan menghitung setelah awalan dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya dan menambah huruf ‘t’ pada awal kata tersebut.

```

//mem pem
if ((AnsiStartsStr('mem', kata)) or
    (AnsiStartsStr('pem', kata))) then
begin
  len := StrLen(Pchar(kata))-2;
  if (countVocal(Copy (kata, 4,len))>=2) then
  begin
    temp := Copy (kata, 3,len);
    if (temp[2] in vocal) then
      temp[1] := 'p'
    else
      temp := Copy(temp, 2, strlen(Pchar(temp)));
    output := temp;
    isfired := true;
  end;
end;

```

Setiap kata yang berawalan ‘mem’ atau ‘pem’ akan melalui proses berikut :

- Jika kata tersebut berawalan ‘mem’, atau ‘pem’, maka fungsi *stemming* akan menghilangkan awalan tersebut dan menghitung setelah awalan dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya dan jika setelah awalan dihilangkan huruf pertama merupakan huruf vocal maka ditambahkan huruf ‘p’ pada awal kata tersebut.

```

//me
if (AnsiStartsStr('me', kata)) then
begin
  len := StrLen(Pchar(kata))-2;
  if (countVocal(Copy (kata, 3,len))>=2) then

```

```

begin
    output := Copy (kata, 3,len);
    isfired := true;
end;
end
else
//di ter ke
    if ((AnsiStartsStr('di', kata)) or
        (AnsiStartsStr('ter', kata)) or
        (AnsiStartsStr('ke', kata))) then
        begin
            if (AnsiStartsStr('ter', kata)) then
                begin
                    len := StrLen(Pchar(kata))-3;
                    index := 4;
                end
            if (countVocal(Copy (kata, 0,len))>=2) then
                begin
                    output := Copy (kata, 0,len);
                    //ShowMessage('akhiran an ' + kata);
                    isfired := true;
                end;
            end;
        end;
    end;
end;

```

Setiap kata yang berawalan ‘me’, ‘di’, ‘ter’, atau ‘ke’ akan melalui proses berikut:

- Jika kata tersebut berawalan ‘me’, ‘di’, ‘ter’, atau ‘ke’, maka fungsi *stemming* akan menghilangkan awalan tersebut dan menghitung setelah awalan dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya

Berikut merupakan fungsi *stemming* suffiks derivasional:

```

kata := StrLower(Pchar(input));
if (AnsiEndsStr('kan', kata)) then
    begin
        len := StrLen(Pchar(kata))-3;
        if (countVocal(Copy (kata, 0,len))>=2) then
            begin
                output := Copy (kata, 0,len);
                isfired := true;
            end;
        end;
    end
else
if ((AnsiEndsStr('an', kata)) or
    (AnsiEndsStr('i', kata))) then
    begin

```

```

if (AnsiEndsStr('an', kata)) then
len := StrLen(Pchar(kata))-2
else
len := StrLen(Pchar(kata))-1;
if (countVocal(Copy (kata, 0,len))>=2) then
begin
output := Copy (kata, 0,len);
isfired:=true;
end;

```

Pada fungsi diatas dilakukan jika terdapat proses *stemming* prefiks derivasional tahap pertama. Setiap kata yang berakhiran 'kan', 'an', atau 'i' akan melalui proses berikut :

- Jika kata tersebut berakhiran 'kan', 'an', atau 'i', maka fungsi *stemming* akan menghilangkan akhiran tersebut dan menghitung setelah akhiran dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya

Berikut merupakan fungsi *stemming* suffiks derivasional:

```

isfired := false;
ajar := 'ajar';
kata := StrLower(Pchar(input));
if ((AnsiStartsStr('ber', kata)) or
(AnsiStartsStr('per', kata))) then
begin
len := StrLen(Pchar(kata))-3;
if (countVocal(Copy (kata, 4,len))>=2) then
begin
output := Copy (kata, 4,len);
isfired := true;
end; end

```

Pada fungsi diatas dilakukan jika terdapat proses *stemming* suffiks ('kan', 'an', atau 'i'). Setiap kata yang berawalan 'ber', atau 'per' akan melalui proses berikut :

- Jika kata tersebut berawalan 'ber', atau 'per' dan terdapat kata 'ajar' maka kata hasil adalah ajar jika bukan maka fungsi *stemming* akan menghilangkan awalan tersebut dan menghitung setelah awalan dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya.

```

//bel pel
  if ((AnsiStartsStr('bel', kata)) or
      (AnsiStartsStr('pel', kata))) then
  begin
    if ((StrPos(Pchar(kata),Pchar(ajar))<> nil)) then
    begin
      output := 'ajar';
    end
    else
    begin
      len := StrLen(Pchar(kata))-2;
      temp := Copy (kata, 3,len);
      if (countVocal(Copy (kata, 2,len))>=2)
then
      begin
        output := Copy (kata, 3,len);
      end;
    end;
  end;

```

Pada fungsi diatas dilakukan jika terdapat proses *stemming* suffiks ('kan', 'an', atau 'i'). Setiap kata yang berawalan 'bel' , 'pel' akan melalui proses berikut :

- Jika kata tersebut berawalan berawalan 'bel' atau 'pel' dan kata berikutnya adalah kata 'ajar' maka fungsi *stemming* akan menghilangkan awalan tersebut dan melihat setelah awalan dihilangkan apakah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya.

```

if (AnsiStartsStr('be', kata)) then
  begin
    len := StrLen(Pchar(kata))-2;
    if (countVocal(Copy (kata, 3,len))>=2) then
    begin
      temp := Copy (kata, 3,len);
      if ((temp[2]='e') and (temp[3]='r')) then
      begin
        output := temp;
        isfired := true;
      end
    end;
  end
end

```

Pada fungsi diatas dilakukan jika terdapat proses *stemming* suffiks ('kan', 'an', atau 'i'). Setiap kata yang berawalan 'be' akan melalui proses berikut :

- Jika kata tersebut berawalan 'be' maka fungsi *stemming* akan menghilangkan awalan tersebut dan menghitung setelah awalan dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya.

```

//pe
if (AnsiStartsStr('pe', kata)) then
begin
len := StrLen(Pchar(kata))-2;
if (countVocal(Copy (kata, 3,len))>=2) then
begin
output := Copy (kata, 3,len);
isfired := true;
end;
end ;

```

Pada fungsi di atas dilakukan jika terdapat proses *stemming* suffiks ('kan', 'an', atau 'i'). Setiap kata yang berawalan 'pe' akan melalui proses berikut :

- Jika kata tersebut berawalan berawalan 'pe' maka fungsi *stemming* akan menghilangkan awalan tersebut dan menghitung setelah awalan dihilangkan apakah jumlah huruf vocal lebih dari dua, jika iya maka kata hasil penghilangan tersebut merupakan hasilnya.

#### 4.3.4 Implementasi *Stopword Removal*

```

function cekStopWord(kata : string; stopWord : arrString) :
boolean;
var
isStopWord : boolean;
i : integer;
begin
isStopWord := false;
for i:=low(stopWord) to high(stopWord) do
if (kata = stopWord[i]) then
begin
isStopWord := true;
end;
Result := isStopWord;
end;

```

Pada fungsi di atas, setiap kata yang dihasilkan dari proses *stemming* akan dicek apakah kata tersebut termasuk dalam daftar *stopword* bukan. Jika termasuk dalam daftar, maka kata tersebut akan dihilangkan, dan jika termasuk, maka kata tersebut akan diproses.

#### 4.3.5 Implementasi Penghitungan *Agglomerative Single Linkage Hierarchical Clustering*

Langkah pertama yang dilakukan pada proses perhitungan adalah proses penghitungan frekuensi kata.

Berikut merupakan fungsi untuk menghitung frekuensi :

```
if find=false then
  begin
    SetLength(bufList, length(bufList)+1);
    bufList[high(bufList)].kata := kata;
    bufList[high(bufList)].frekuensi := 1;
  end;
end;
Result := bufList;
```

Prosedur diatas digunakan untuk menghitung frekuensi kata tiap dokumen dan menyimpannya dalam array buflist, yang sebelumnya telah didefinisikan buflist merupakan **arrWord**. Frekuensi yang terhitung digunakan untuk menghitung jarak antar dokumen dengan memakai rumus *Euchclidean Distance*. Berikut fungsi untuk menghitung *Ecludian Distance* tersebut :

```
dokCount := length(listDokumen);
combineCount := totalCombine(dokCount);
try
  bufStringList := TStringList.Create;
  for i := 0 to length(listDokumen[0].listWord)-1 do
  begin
    bufStringList.Add(listDokumen[0].listWord[i].kata);
  end;
  for i := 1 to dokCount-1 do
  begin
    for j := 0 to length(listDokumen[i].listWord)-1 do
    begin
      find := false;
      for k := 0 to bufStringList.Count-1 do
      begin
        if bufStringList[k] = listDokumen[i].listWord[j].kata
        then
          find := true;
        end;
      end;
      if find = false then
        bufStringList.Add(listDokumen[i].listWord[j].kata);
      end; end;

    bufStringList.Sort;
    SetLength(bufTabel.kata, bufStringList.Count);
    for i := 0 to bufStringList.Count-1 do
    begin
      bufTabel.kata[i] := copy(bufStringList[i], 0,
        length(bufStringList[i]));
    end;
  end;
end;
```

```

SetLength(bufTabel.frekuensi, bufStringList.Count);
SetLength(bufTabel.deltaFrek, bufStringList.Count);
SetLength(bufTabel.totalDistance, combineCount);
SetLength(bufTabel.euclidDistance, combineCount);
ProgressBar.Position := 0;
ProgressBar.Max := length(bufTabel.frekuensi)-1;
for i := 0 to length(bufTabel.frekuensi)-1 do
begin
SetLength(bufTabel.frekuensi[i], dokCount);
for j:= 0 to dokCount-1 do
begin
bufTabel.frekuensi[i][j] := 0;
k := 0;
while ((k<Length(listDokumen[j].listWord)-1) and
(bufTabel.kata[i]<>listDokumen[j].listWord[k].kata))
do
begin
k := k + 1;
end;
if (bufTabel.kata[i]=listDokumen[j].listWord[k].kata)
then
begin
bufTabel.frekuensi[i][j] :=
listDokumen[j].listWord[k].frekuensi;
end;
end;
SetLength(bufTabel.deltaFrek[i], combineCount);
l := 0;
for m := 0 to dokCount-2 do
for n := m+1 to dokCount-1 do
begin
bufTabel.deltaFrek[i][l] :=
sqr(bufTabel.frekuensi[i][m] -
bufTabel.frekuensi[i][n]);
bufTabel.totalDistance[l] :=
bufTabel.totalDistance[l]
+ bufTabel.deltaFrek[i][l];
l := l + 1;
end;
ProgressBar.Position := i;
end;
for i := 0 to length(bufTabel.totalDistance)-1 do
bufTabel.euclidDistance[i] :=
sqrt(bufTabel.totalDistance[i]);

```

Fungsi createTable di atas digunakan untuk menghitung *Euclidean Distance* antar dokumen, mendaftarkan kata – kata pada dokumen – dokumen yang diproses beserta frekuensinya pada masing – masing dokumen. *Euclidean Distance* yang terbentuk

disimpan dalam bufTabel, bufTabel adalah record **tableDistance** yang telah ddefiniskan.

Setelah diketahui nilai *Euclidean Distance*, akan dibentuk inisialisi pembentukan cluster awal, berikut fungsi pembentukan awal cluster tersebut:

```
n := 0;
for i := 0 to dokCount-2 do
  for j := i+1 to dokCount-1 do
    begin
      bufCluster.dataCell[n].rowCell := i;
      bufCluster.dataCell[n].colCell := j;
      bufCluster.dataCell[n].euclidDist := dataDistance[n];
      ProgrssBar.Position := n;
      n := n + 1;
    end;
  bufCluster.clusterClass := TStringList.Create;
  for i := 0 to bufCluster.clusterCount-1 do
    bufCluster.clusterClass.Add(IntToStr(i));
  Result := bufCluster;
end;
begin
  newClusters.dataCell[i].euclidDist := buf1[index1];
  index1 := index1 + 1;
end;
function findMinDataCluster(cluster : tabelCluster) :
integer;
var
  i, j : integer;
  cellMin : cellCluster;
begin
  j := 0;
  cellMin.euclidDist := Cluster.dataCell[0].euclidDist;
  for i:= 1 to length(Cluster.dataCell)-1 do
    begin
      if cellMin.euclidDist > Cluster.dataCell[i].euclidDist
then
begin
      cellMin.euclidDist := Cluster.dataCell[i].euclidDist;
      j := i;
    end;
  end;
  Result := j;
```

Fungsi create cluster diatas digunakan untuk menginisialisasi nilai Euclidean distance pada tahap awal pembentukan cluster, hasil dari fungsi diatas nantinya dapat ditampilkan dalam bentuk matrik. Tabel matrik yang terbentuk berisi nilai Eucidean distance masing – masing dokumen hasil dari perhitungan fungsi createTabel.

```

newClusters.clusterCount := bufClusters.clusterCount-1;
SetLength(newClusters.dataCell,
totalCombine(newClusters.clusterCount));
n := 0;
for i := 0 to newClusters.clusterCount-2 do
begin
for j := i+1 to newClusters.clusterCount-1 do
begin
newClusters.dataCell[n].rowCell := i;
newClusters.dataCell[n].colCell := j;
n := n+1;
end;
end;
index1 := 0; index2 := 0;
for i := 0 to length(newClusters.dataCell)-1 do
begin
if newClusters.dataCell[i].colCell =
newClusters.clusterCount-1 then
begin
newClusters.dataCell[i].euclidDist := buf2[index2];
index2 := index2 + 1;
end
Else
Begin
newClusters.dataCell[i].euclidDist:=buf1[index1];
index1:=index1+1;
end
end;
newClusters.clusterClass := TStringList.Create;
newClusters.clusterClass := strNotJoin;
newClusters.clusterClass.Add(strJoin);
str := '';

```

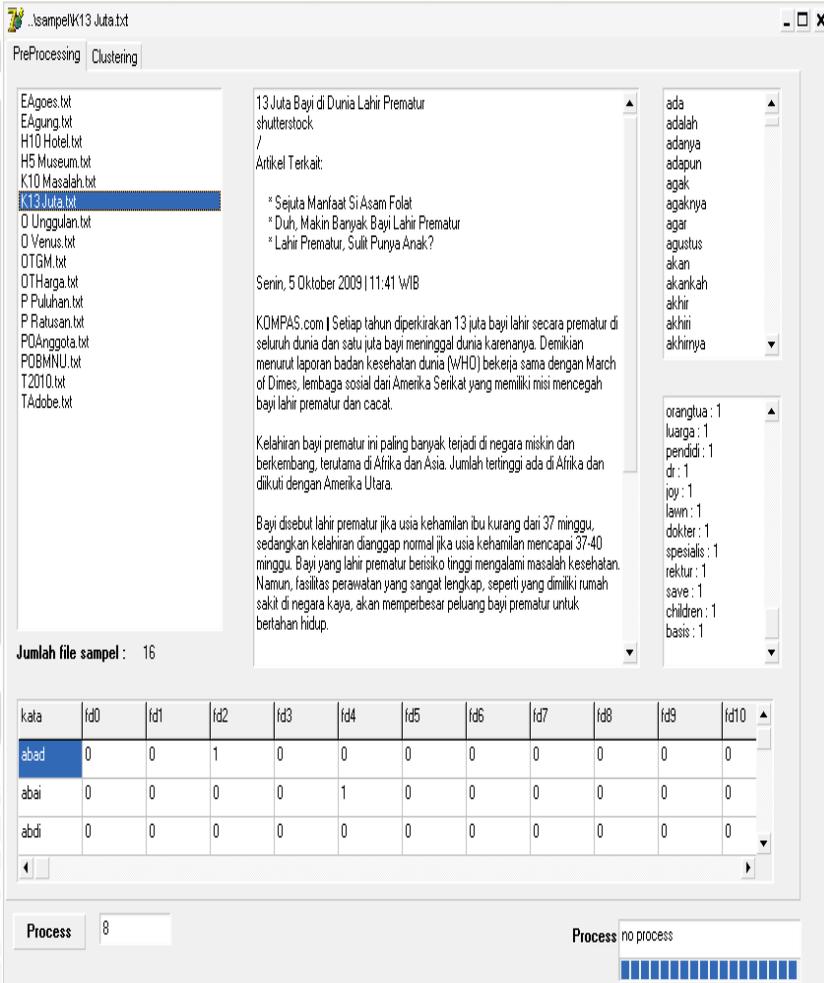
Fungsi diatas digunakan untuk pembentukan cluster baru setelah dilakukan penghitungan jarak minimal antar cluster. Custer yang mempunyai jarak terdekat digabungkan menjadi satu cluster baru. Proses ini berlangsung sampai terbentuk jumlah cluster yang diinginkan.

#### 4.4 Implementasi Antar Muka

Implementasi antar muka *preprocessing* dapat dilihat pada gambar 4.1. Dapat dilihat pada gambar terdapat daftar nama – nama dokumen, *window* untuk menampilkan isi dari dokumen tersebut jika nama dokumen di *double* klik. Daftar stopword, kata hasil *preprocessing* dan frekuensinya masing – masing dokumen. Kata dan frekuensi masing – masing dokumen ini akan terlihat jika *user*

melakukan *double* klik pada *window* nama – nama dokumen. Jumlah file sampel menunjukkan banyaknya dokumen yang akan diklasifikasikan.

Ketika sistem akan dijalankan, ditentukan jumlah *cluster* yang akan terbentuk terlebih dahulu. Untuk menjalankan sistem menggunakan tombol proses, selama proses sistem berjalan dapat dilihat pada *progress bar* proses yang dilakukan oleh sistem. Setelah sistem selesai melakukan klasifikasi akan muncul daftar semua kata, frekuensi, dan jarak masing – masing dokumen pada tabel.



Gambar 4.1 Antarmuka tahap *preprocessing*

Implementasi proses *clustering* dapat dilihat pada gambar 4.2. Dapat dilihat pada gambar, terdapat *window* untuk menampilkan hasil *clustering* dokumen dalam bentuk hierarki, *window* untuk menampilkan isi dokumen hasil *clustering*, perhitungan jarak antar *cluster*, perhitungan antar *cell* matrik yang terbentuk, dan hasil akhir matrik jarak yang terbentuk. Isi dari menu *clustering* ini terlihat setelah proses *clustering* selesai dilakukan.

The screenshot shows a software window titled "PreProcessing Clustering" with a file explorer on the left, a text preview in the center, and a distance matrix at the bottom.

**File Hierarchy:**

- 1: EAgung.txt
- 2: H10 Hotel.txt
- 3: H5 Museum.txt
- 5: K13 Jula.txt
- 10: P Puluhan.txt
- 12: PQ Anggota.txt
- 13: POBMNU.txt
  - 8: OTGM.txt
    - 14: T2010.txt
      - 11: P Ratusan.txt
      - 15: TAdobe.txt
    - 6: O Unggulan.txt
    - 7: O Venus.txt

**Text Preview:**

Puluhan Siswa SD Cisarua Keracunan Coklat

SUKABUMI|Sebanyak 25 orang pelajar Sekolah Dasar Negeri (SDN) Cisarua 4, Desa Cisarua, Kecamatan Sukabumi, Kabupaten Sukabumi keracunan makanan bakso coklat (Baslok). Dampaknya para pelajar tersebut selama satu hari tidak bisa sekolah seperti biasanya.

Dari 25 orang siswa, 16 orang diantaranya harus dibawa ke Puskesmas Sukabumi karena kondisinya cukup parah. Sementara sembilan pelajar lainnya hanya mendapatkan perawatan di Unit Kesehatan Sekolah (UKS) SDN 4 Cisarua.

"Para pelajar tersebut kini sudah sehat kembali namun bekas keracunannya masih terlihat di bagian

0:(0)  
1:(1)  
2:(2)  
3:(3)  
4:(4)  
5:(5)  
6:(6)  
7:(7)  
8:(8)  
9:(9)  
10:(10)  
11:(11)  
12:(12)  
13:(13)  
14:(14)  
15:(15)  
16:(16)

isi cell cluster awal :

0, 1 = 72.9520  
0, 2 = 90.0722  
0, 3 = 86.4292  
0, 4 = 70.7531  
0, 5 = 73.5595  
0, 6 = 71.5122  
0, 7 = 71.8749  
0, 8 = 71.2180  
0, 9 = 71.5472  
0, 10 = 72.6154  
0, 11 = 70.4557  
0, 12 = 74.9667  
0, 13 = 72.9246  
0, 14 = 70.3349  
0, 15 = 71.1337  
1, 3 = 63.2535  
1, 3 = 58.8048  
1, 4 = 30.4631  
1, 5 = 36.0139  
1, 6 = 29.5635  
1, 7 = 30.7896  
1, 8 = 30.1993  
1, 9 = 30.5450  
1, 10 = 38.4057

**Distance Matrix:**

0	72.9520	90.0722	86.4292	73.5595	72.6154	74.9667	72.9246	70.3349
0		63.2535	58.8048	36.0139	38.4057	36.3593	31.9374	28.3549
			78.0705	63.9062	65.5134	65.1537	62.7615	60.4897
				59.0847	60.6548	60.7454	58.2580	55.6956
					38.8587	39.0768	34.6554	30.6431
						41.3884	37.5899	30.3150

Process: 8

Gambar 4.2 Antarmuka proses *clustering*

## 4.5 Implementasi Uji Coba

### 4.5.1 Skenario Evaluasi

Pada pengujian sistem klasifikasi ini, menggunakan 120 buah dokumen yang sebelumnya telah diketahui berasal dari 8 kategori yang berbeda.

Untuk mempelajari pengaruh hasil *clustering* yang terbentuk maka dilakukan uji coba. *Cluster – cluster* yang terbentuk akan dievaluasi dengan menggunakan *recall*, *precisian*, dan  $F_1$  *measure*.

### 4.5.2 Hasil Evaluasi

Beikut ini adalah hasil evaluasi uji coba yang dilakukan sebanyak 15 kali uji coba.

Uji coba pertama digunakan 8 dokumen, dapat dilihat pada tabel 4.1.

**Tabel 4.1** Jumlah dokumen setiap kategori pada uji coba pertama

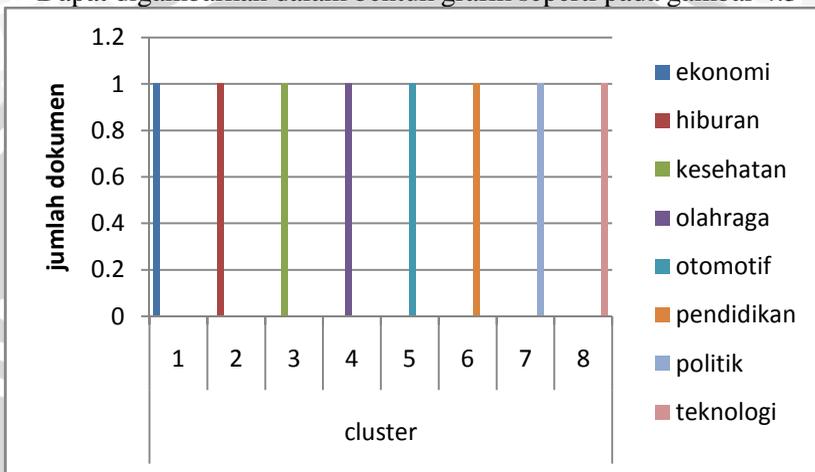
kategori	jumlah
ekonomi	1
hiburan	1
kesehatan	1
olahraga	1
otomotif	1
pendidikan	1
politik	1
teknologi	1

Berikut merupakan hasil *clustering* pada uji coba pertama pada sistem, anggota *clusternya* terdiri dari:

- *Cluster* 1 terdiri : 1 dokumen ekonomi
- *Cluster* 2 terdiri : 1 dokumen ekonomi
- *Cluster* 3 terdiri : 1 dokumen hiburan
- *Cluster* 4 terdiri : 1 dokumen hiburan
- *Cluster* 5 terdiri : 1 dokumen hiburan
- *Cluster* 6 terdiri : 1 dokumen kesehatan
- *Cluster* 7 terdiri : 1 dokumen pendidikan
- *Cluster* 8 terdiri : 1 dokumen ekonomi

Pada uji coba 8 dokumen, diperoleh hasil pengclusteran sistem menunjukkan setiap *cluster* terdiri dari 1 dokumen, tiap *cluster* beranggotakan dokumen yang berbeda.

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.3



**Gambar 4.3** grafik jumlah dokumen hasil *clustering* sistem pada uji coba pertama

**Tabel 4.2** Evaluasi efektivitas uji coba pertama (8 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	0	0	1	1	1
<i>cluster2</i>	1	0	0	1	1	1
<i>cluster3</i>	1	0	0	1	1	1
<i>cluster4</i>	1	0	0	1	1	1
<i>cluster5</i>	1	0	0	1	1	1
<i>cluster6</i>	1	0	0	1	1	1
<i>cluster7</i>	1	0	0	1	1	1
<i>cluster8</i>	1	0	0	1	1	1

Keterangan :

a : Pengklasifikasian sebenarnya ya dan pengklasifikasian oleh sistem ya.

b : Pengklasifikasian sebenarnya ya dan pengklasifikasian oleh sistem bukan.

c : Pengklasifikasian sebenarnya bukan dan pengklasifikasian oleh sistem ya.

Dari perhitungan *recall*, *precision*, dan *F<sub>1</sub> measure* pada tabel 4.2 diperoleh hasil semuanya bernilai satu, karena semua dokumen dari kategori berbeda yang diuji cobakan dapat masuk ke semua *cluster*, *cluster-cluster* yang terbentuk diasumsikan sebagai kategori yang berbeda.

Uji coba kedua digunakan 16 dokumen, terlihat pada tabel 4.3.

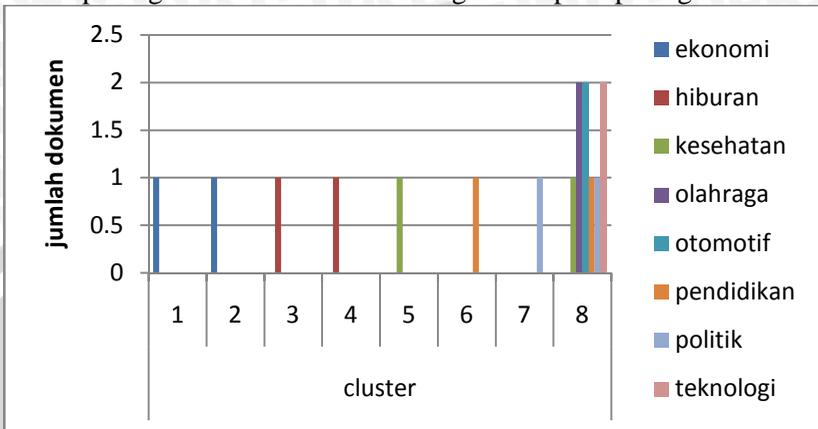
**Tabel 4.3** jumlah dokumen setiap kategori pada uji coba kedua

kategori	jumlah
ekonomi	2
hiburan	2
kesehatan	2
olahraga	2
otomotif	2
pendidikan	2
politik	2
teknologi	2

Berikut merupakan hasil *clustering* pada uji coba kedua pada sistem, anggota *clusternya* terdiri dari:

- *Cluster 1* terdiri : 1 dokumen ekonomi
- *Cluster 2* terdiri : 1 dokumen ekonomi
- *Cluster 3* terdiri : 1 dokumen hiburan
- *Cluster 4* terdiri : 1 dokumen hiburan
- *Cluster 5* terdiri : 1 dokumen kesehatan
- *Cluster 6* terdiri : 1 dokumen pendidikan
- *Cluster 7* terdiri : 1 dokumen politik
- *Cluster 8* terdiri : 1 dokumen kesehatan  
2 dokumen olahraga  
2 dokumen otomotif  
2 dokumen pendidikan  
1 dokumen politik  
2 dokumen teknologi

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.4.



**Gambar 4.4** grafik jumlah dokumen hasil *clustering* sistem pada uji coba kedua

Pada uji coba 16 dokumen, diperoleh hasil peng*clustering* sistem dokumen mengumpul disalah satu *cluster*(*cluster* 8) sebanyak 9 dokumen mengumpul di *cluster* ini, sedangkan *cluster* 1 sampai dengan *cluster* 7 beranggotakan masing – masing 1 dokumen.

**Tabel 4.4** Evaluasi efektivitas uji coba kedua (16 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	1	0	1	0.5	0.666667
<i>cluster2</i>	1	1	0	1	0.5	0.666667
<i>cluster3</i>	1	1	0	1	0.5	0.666667
<i>cluster4</i>	1	1	0	1	0.5	0.666667
<i>cluster5</i>	1	1	0	1	0.5	0.666667
<i>cluster6</i>	1	1	0	1	0.5	0.666667
<i>cluster7</i>	1	1	0	1	0.5	0.666667
<i>cluster8</i>	2	0	7	0.222222	1	0.363636

Pada tabel 4.4 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang

berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recall*nya kecil yaitu 0.222222, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan oleh salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.5 karena dari dua dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang ter*cluster*. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai *F<sub>1</sub> measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama(0.666667) dan lebih besar daripada nilai *F<sub>1</sub> measure* pada *cluster* terakhir(0.363636) dikarenakan pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai *F<sub>1</sub> measure*.

Uji coba ketiga digunakan 24 dokumen, dapat dilihat pada tabel 4.5.

**Tabel 4.5** jumlah dokumen setiap kategori pada uji coba ketiga

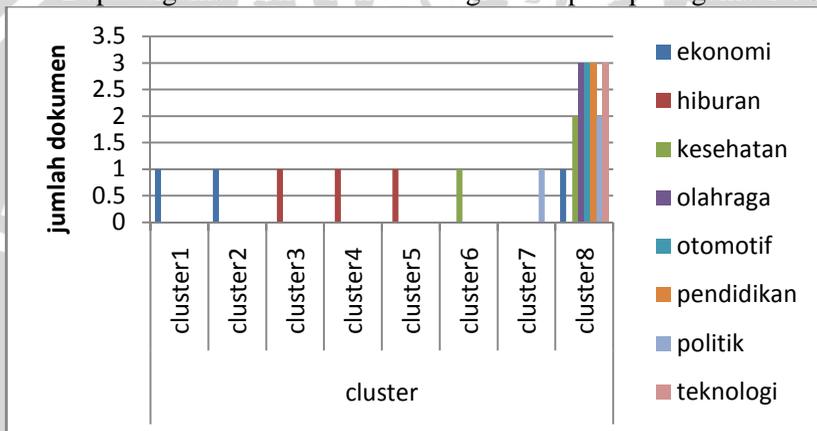
kategori	jumlah
ekonomi	3
hiburan	3
kesehatan	3
olahraga	3
otomotif	3
pendidikan	3
politik	3
teknologi	3

Berikut merupakan hasil *clustering* pada uji coba ketiga pada sistem, anggota *clusternya* terdiri dari:

- *Cluster* 1 terdiri : 1 dokumen ekonomi
- *Cluster* 2 terdiri : 1 dokumen ekonomi
- *Cluster* 3 terdiri : 1 dokumen hiburan
- *Cluster* 4 terdiri : 1 dokumen hiburan
- *Cluster* 5 terdiri : 1 dokumen hiburan
- *Cluster* 6 terdiri : 1 dokumen kesehatan

- *Cluster 7* terdiri : 1 dokumen politik
- *Cluster 8* terdiri : 1 dokumen ekonomi  
2 dokumen kesehatan  
3 dokumen olahraga  
3 dokumen otomotif  
3 dokumen pendidikan  
2 dokumen politik  
3 dokumen teknologi

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.5.



**Gambar 4.5** grafik jumlah dokumen hasil *clustering* sistem pada uji coba ketiga

Pada uji coba 24 dokumen, diperoleh hasil peng*clustering* sistem dokumen mengumpul disalah satu *cluster*(*cluster 8*) sebanyak 17 dokumen mengumpul di *cluster* ini, sedangkan *cluster 1* sampai dengan *cluster 7* beranggotakan masing – masing 1 dokumen.

**Tabel 4.6** Evaluasi efektivitas uji coba ketiga (24 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	2	0	1	0.333333	0.5
<i>cluster2</i>	1	2	0	1	0.333333	0.5
<i>cluster3</i>	1	2	0	1	0.333333	0.5
<i>cluster4</i>	1	2	0	1	0.333333	0.5

<i>cluster5</i>	1	2	0	1	0.333333	0.5
<i>cluster6</i>	1	2	0	1	0.333333	0.5
<i>cluster7</i>	1	2	0	1	0.333333	0.5
<i>cluster8</i>	3	0	14	0.176471	1	0.3

Pada tabel 4.6 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recall*nya kecil yaitu sebesar 0.176471, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan oleh salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.333333 karena dari tiga dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang tercluster. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama(0.5) dan lebih besar daripada nilai  $F_1$  *measure* pada *cluster* terakhir (0.3) dikarenakan pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba keempat digunakan 32 dokumen, dapat dilihat pada tabel 4.7.

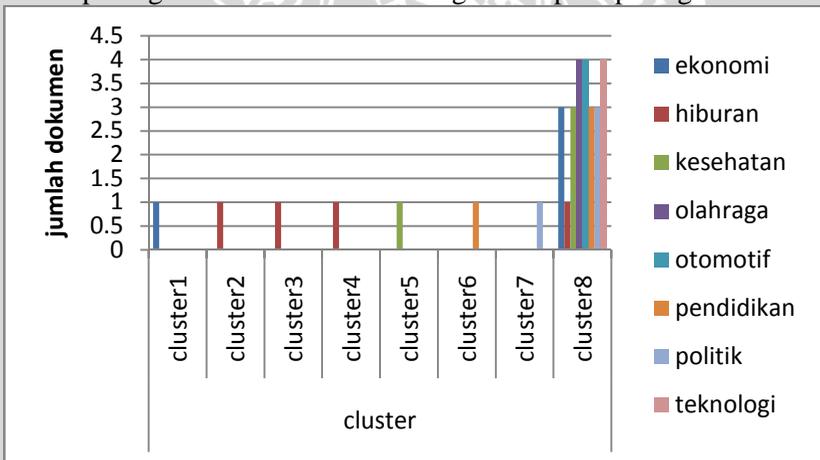
**Tabel 4.7** jumlah dokumen setiap kategori pada uji coba keempat

kategori	jumlah
ekonomi	4
hiburan	4
kesehatan	4
olahraga	4
otomotif	4
pendidikan	4
politik	4
teknologi	4

Berikut merupakan hasil *clustering* pada uji coba keempat pada sistem, anggota *clusternya* terdiri dari:

- *Cluster 1* terdiri : 1 dokumen ekonomi
- *Cluster 2* terdiri : 1 dokumen hiburan
- *Cluster 3* terdiri : 1 dokumen hiburan
- *Cluster 4* terdiri : 1 dokumen hiburan
- *Cluster 5* terdiri : 1 dokumen kesehatan
- *Cluster 6* terdiri : 1 dokumen pendidikan
- *Cluster 7* terdiri : 1 dokumen politik
- *Cluster 8* terdiri : 3 dokumen ekonomi  
1 dokumen hiburan  
3 dokumen kesehatan  
4 dokumen olahraga  
4 dokumen otomotif  
3 dokumen pendidikan  
3 dokumen politik  
4 dokumen teknologi

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.6:



**Gambar 4.6** grafik jumlah dokumen hasil *clustering* sistem pada uji coba keempat

Pada uji coba 32 dokumen, diperoleh hasil peng*clustering* sistem dokumen mengumpul disalah satu *cluster(cluster 8)* sebanyak 25 dokumen mengumpul di *cluster* ini, sedangkan *cluster 1* sampai dengan *cluster 7* beranggotakan masing – masing 1 dokumen.

**Tabel 4.8** Evaluasi efektivitas uji coba keempat (32 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	3	0	1	0.25	0.4
<i>cluster2</i>	1	3	0	1	0.25	0.4
<i>cluster3</i>	1	3	0	1	0.25	0.4
<i>cluster4</i>	1	3	0	1	0.25	0.4
<i>cluster5</i>	1	3	0	1	0.25	0.4
<i>cluster6</i>	1	3	0	1	0.25	0.4
<i>cluster7</i>	1	3	0	1	0.25	0.4
<i>cluster8</i>	4	0	21	0.16	1	0.275862

Pada tabel 4.8 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recall*nya kecil yaitu sebesar 0.16, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.25 karena dari 4 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang ter*cluster*. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.4) dan lebih besar daripada nilai  $F_1$  *measure* pada *cluster* terakhir (0.275862) dikarenakan pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba kelima digunakan 40 dokumen, dapat dilihat pada tabel 4.9.

**Tabel 4.9** jumlah dokumen setiap kategori pada uji coba kelima

kategori	jumlah
ekonomi	5
hiburan	5

kesehatan	5
olahraga	5
otomotif	5
pendidikan	5
politik	5
teknologi	5

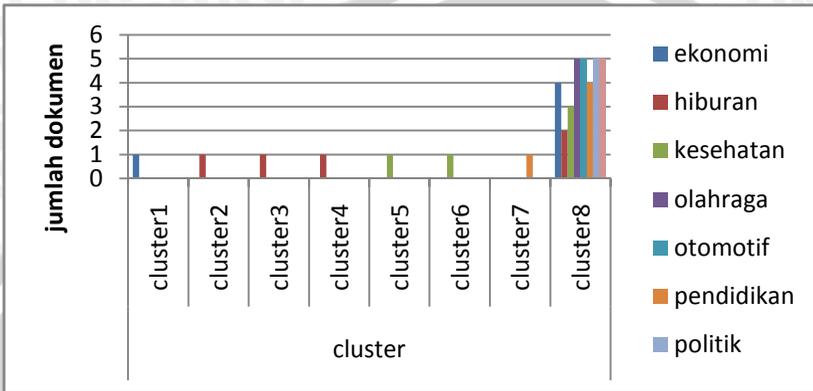
Berikut merupakan hasil *clustering* pada uji coba kelima pada sistem, anggota *clusternya* terdiri dari:

- *Cluster 1* terdiri : 1 dokumen ekonomi
- *Cluster 2* terdiri : 1 dokumen hiburan
- *Cluster 3* terdiri : 1 dokumen hiburan
- *Cluster 4* terdiri : 1 dokumen hiburan
- *Cluster 5* terdiri : 1 dokumen kesehatan
- *Cluster 6* terdiri : 1 dokumen kesehatan
- *Cluster 7* terdiri : 1 dokumen pendidikan
- *Cluster 8* terdiri : 4 dokumen ekonomi  
2 dokumen hiburan  
3 dokumen kesehatan  
5 dokumen olahraga  
5 dokumen otomotif  
4 dokumen pendidikan  
5 dokumen politik  
5 dokumen teknologi

Pada uji coba40 dokumen, diperoleh hasil peng*clusteran* sistem dokumen mengumpul disalah satu *cluster(cluster 8)* sebanyak 33 dokumen mengumpul di *cluster* ini, sedangkan *cluster 1* sampai dengan *cluster 7* beranggotakan masing – masing 1 dokumen. Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.7.

Pada tabel 4.10 dapat diketahui bahwa nilai *recall* pada *cluster 1* sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster 8*, nilai *recallnya* kecil yaitu sebesar 0.131579, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall*

dihitung berdasarkan salah satu kategori yang jumlahnya mengumpul banyak di *cluster* ini.



**Gambar 4.7** grafik jumlah dokumen hasil *clustering* sistem pada uji coba kelima

**Tabel 4.10** Evaluasi efektivitas uji coba kelima (40 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	4	0	1	0.2	0.333333
<i>cluster2</i>	1	4	0	1	0.2	0.333333
<i>cluster3</i>	1	4	0	1	0.2	0.333333
<i>cluster4</i>	1	4	0	1	0.2	0.333333
<i>cluster5</i>	1	4	0	1	0.2	0.333333
<i>cluster6</i>	1	4	0	1	0.2	0.333333
<i>cluster7</i>	1	4	0	1	0.2	0.333333
<i>cluster8</i>	5	0	33	0.131579	1	0.232558

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.2 karena dari 5 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang ter*cluster*. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.333333) dan lebih besar daripada nilai  $F_1$  *measure* pada *cluster* terakhir (0.232558) dikarenakan pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba keenam digunakan 48 dokumen, dapat dilihat pada tabel 4.11.

**Tabel 4.11** jumlah dokumen setiap kategori pada uji coba keenam

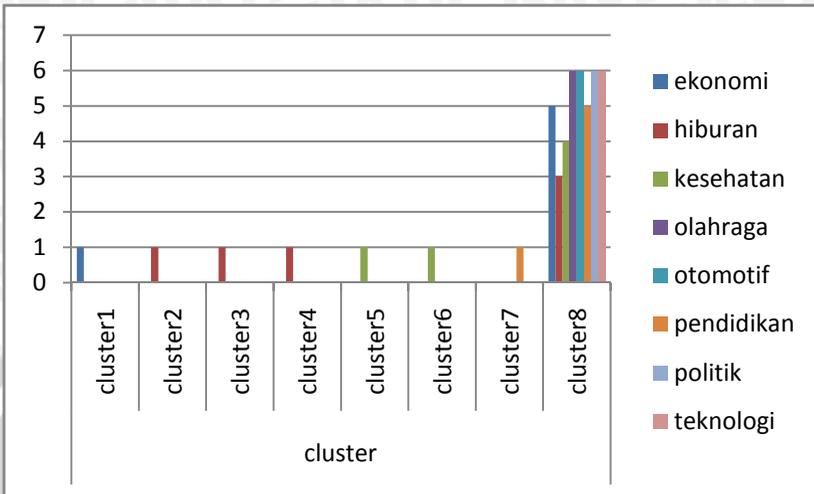
kategori	jumlah
ekonomi	6
hiburan	6
kesehatan	6
olahraga	6
otomotif	6
pendidikan	6
politik	6
teknologi	6

Berikut merupakan hasil *clustering* pada uji coba keenam pada sistem, anggota *clusternya* terdiri dari:

- *Cluster 1* terdiri : 1 dokumen ekonomi
- *Cluster 2* terdiri : 1 dokumen hiburan
- *Cluster 3* terdiri : 1 dokumen hiburan
- *Cluster 4* terdiri : 1 dokumen hiburan
- *Cluster 5* terdiri : 1 dokumen kesehatan
- *Cluster 6* terdiri : 1 dokumen kesehatan
- *Cluster 7* terdiri : 1 dokumen pendidikan
- *Cluster 8* terdiri : 5 dokumen ekonomi  
3 dokumen hiburan  
4 dokumen kesehatan  
6 dokumen olahraga  
6 dokumen otomotif  
5 dokumen pendidikan  
6 dokumen politik  
6 dokumen teknologi

Pada uji coba 48 dokumen, diperoleh hasil pengclusteran sistem dokumen mengumpul disalah satu *cluster*(*cluster 8*) sebanyak 41 dokumen mengumpul di *cluster* ini, sedangkan *cluster 1* sampai dengan *cluster 7* beranggotakan masing – masing 1 dokumen.

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.8.



**Gambar 4.8** grafik jumlah dokumen hasil *clustering* sistem pada uji coba keenam

**Tabel 4.12** Evaluasi efektivitas uji coba keenam (48 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	5	0	1	0.166667	0.285714
<i>cluster2</i>	1	5	0	1	0.166667	0.285714
<i>cluster3</i>	1	5	0	1	0.166667	0.285714
<i>cluster4</i>	1	5	0	1	0.166667	0.285714
<i>cluster5</i>	1	5	0	1	0.166667	0.285714
<i>cluster6</i>	1	5	0	1	0.166667	0.285714
<i>cluster7</i>	1	5	0	1	0.166667	0.285714
<i>cluster8</i>	6	0	35	0.146341	1	0.255319

Pada tabel 4.12 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recall*nya kecil yaitu sebesar 0.146341, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.166667 karena dari 6 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang tercluster. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.285714) dan lebih besar daripada nilai  $F_1$  *measure* pada *cluster* terakhir (0.255319) dikarenakan pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba ketujuh digunakan 56 dokumen, dapat dilihat pada tabel 4.13 :

**Tabel 4.13** jumlah dokumen setiap kategori pada uji coba ketujuh

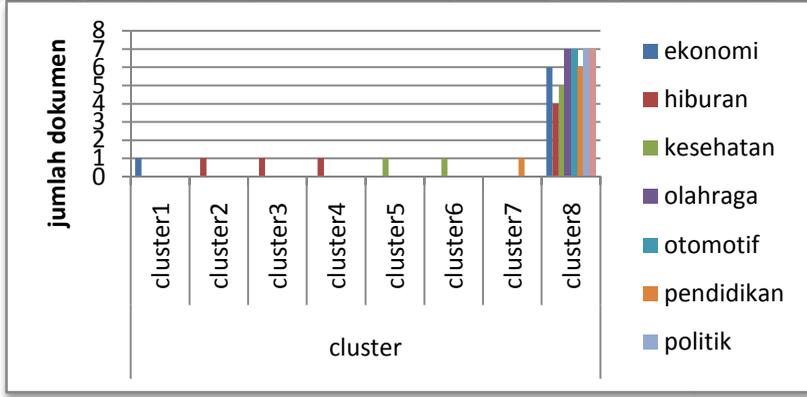
kategori	jumlah
ekonomi	7
hiburan	7
kesehatan	7
olahraga	7
otomotif	7
pendidikan	7
politik	7
teknologi	7

Berikut merupakan hasil *clustering* pada uji coba ketujuh pada sistem, anggota *clusternya* terdiri dari:

- *Cluster* 1 terdiri : 1 dokumen ekonomi
- *Cluster* 2 terdiri : 1 dokumen hiburan
- *Cluster* 3 terdiri : 1 dokumen hiburan
- *Cluster* 4 terdiri : 1 dokumen hiburan
- *Cluster* 5 terdiri : 1 dokumen kesehatan
- *Cluster* 6 terdiri : 1 dokumen kesehatan
- *Cluster* 7 terdiri : 1 dokumen pendidikan
- *Cluster* 8 terdiri : 6 dokumen ekonomi  
6 dokumen hiburan  
6 dokumen kesehatan  
7 dokumen olahraga

7 dokumen otomotif  
 6 dokumen pendidikan  
 6 dokumen politik  
 7 dokumen teknologi

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.9.



**Gambar 4.9** grafik jumlah dokumen hasil *clustering* sistem pada uji coba ketujuh

Pada uji coba 56 dokumen, diperoleh hasil peng*clustering* sistem dokumen mengumpul disalah satu *cluster*(*cluster* 8) sebanyak 48 dokumen mengumpul di *cluster* ini, sedangkan *cluster* 1 sampai dengan *cluster* 7 beranggotakan masing – masing 1 dokumen.

**Tabel 4.14** Evaluasi efektivitas uji coba ketujuh (56 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	<i>F<sub>1</sub> measure</i>
<i>cluster1</i>	1	6	0	1	0.142857	0.25
<i>cluster2</i>	1	6	0	1	0.142857	0.25
<i>cluster3</i>	1	6	0	1	0.142857	0.25
<i>cluster4</i>	1	6	0	1	0.142857	0.25
<i>cluster5</i>	1	6	0	1	0.142857	0.25
<i>cluster6</i>	1	6	0	1	0.142857	0.25
<i>cluster7</i>	1	6	0	1	0.142857	0.25
<i>cluster8</i>	7	0	41	0.145833	1	0.254545

Pada tabel 4.14 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang

berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recall*-nya kecil yaitu sebesar 0.145833, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.142857 karena dari 7 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang ter*cluster*. Pada *cluster* terakhir nilai *precision*-nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.25) dan  $F_1$  *measure* pada *cluster* terakhir adalah 0.254545. Pada *cluster* terakhir nilai *recall*-nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba kedelapan digunakan 64 dokumen, dapat dilihat pada tabel 4.15.

**Tabel 4.15** jumlah dokumen setiap kategori pada uji coba kedelapan

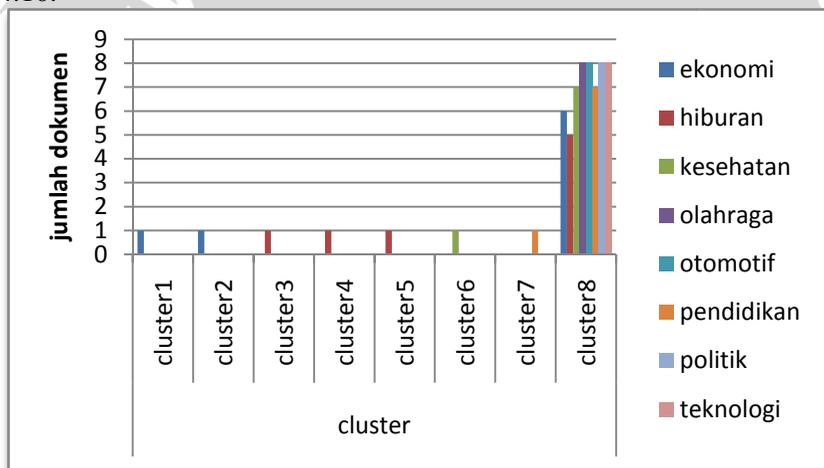
kategori	jumlah
ekonomi	8
hiburan	8
kesehatan	8
olahraga	8
otomotif	8
pendidikan	8
politik	8
teknologi	8

Berikut merupakan hasil *clustering* pada uji coba kedelapan pada sistem, anggota *clusternya* terdiri dari:

- *Cluster* 1 terdiri : 1 dokumen ekonomi
- *Cluster* 2 terdiri : 1 dokumen ekonomi
- *Cluster* 3 terdiri : 1 dokumen hiburan
- *Cluster* 4 terdiri : 1 dokumen hiburan
- *Cluster* 5 terdiri : 1 dokumen hiburan

- Cluster 6 terdiri : 1 dokumen kesehatan
- Cluster 7 terdiri : 1 dokumen pendidikan
- Cluster 8 terdiri : 6 dokumen ekonomi  
5 dokumen hiburan  
7 dokumen kesehatan  
8 dokumen olahraga  
8 dokumen otomotif  
7 dokumen pendidikan  
8 dokumen politik  
8 dokumen teknologi

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.10.



**Gambar 4.10** grafik jumlah dokumen hasil *clustering* sistem pada uji coba kedelapan

Pada uji coba 64 dokumen, diperoleh hasil peng*clustering* sistem dokumen mengumpul disalah satu *cluster*(*cluster 8*) sebanyak 57 dokumen mengumpul di *cluster* ini, sedangkan *cluster 1* sampai dengan *cluster 7* beranggotakan masing – masing 1 dokumen.

**Tabel 4.16** Evaluasi efektivitas uji coba kedelapan (64 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	7	0	1	0.125	0.222222
<i>cluster2</i>	1	7	0	1	0.125	0.222222
<i>cluster3</i>	1	7	0	1	0.125	0.222222

<i>cluster4</i>	1	7	0	1	0.125	0.222222
<i>cluster5</i>	1	7	0	1	0.125	0.222222
<i>cluster6</i>	1	7	0	1	0.125	0.222222
<i>cluster7</i>	1	7	0	1	0.125	0.222222
<i>cluster8</i>	8	0	49	0.140351	1	0.246154

Pada tabel 4.16 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recall*nya kecil yaitu sebesar 0.140351, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.125 karena dari 8 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang ter*cluster*. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.222222) dan nilai  $F_1$  *measure* pada *cluster* terakhir adalah 0.246154. Pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba kesembilan digunakan 72 dokumen, dapat dilihat pada tabel 4.17.

**Tabel 4.17** jumlah dokumen setiap kategori pada uji coba kesembilan

kategori	jumlah
ekonomi	9
hiburan	9
kesehatan	9
olahraga	9
otomotif	9
pendidikan	9

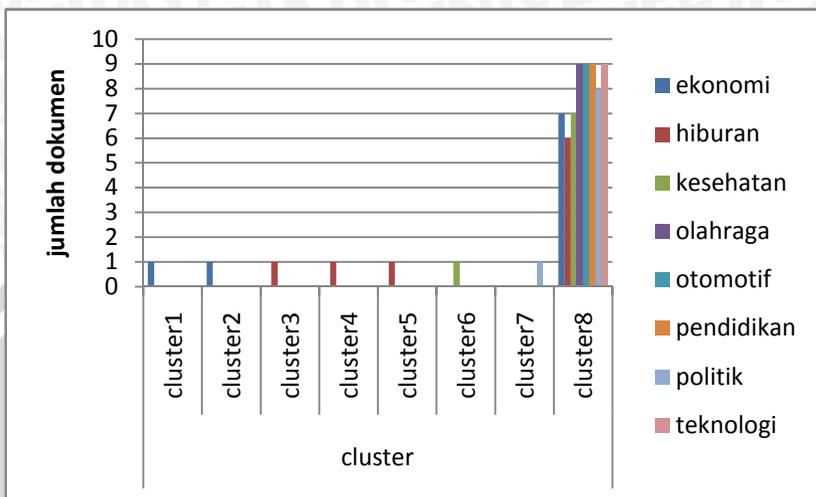
politik	9
teknologi	9

Berikut merupakan hasil *clustering* pada uji coba kesembilan pada sistem, anggota *clusternya* terdiri dari:

- *Cluster 1* terdiri : 1 dokumen ekonomi
- *Cluster 2* terdiri : 1 dokumen ekonomi
- *Cluster 3* terdiri : 1 dokumen hiburan
- *Cluster 4* terdiri : 1 dokumen hiburan
- *Cluster 5* terdiri : 1 dokumen hiburan
- *Cluster 6* terdiri : 1 dokumen kesehatan
- *Cluster 7* terdiri : 1 dokumen politik
- *Cluster 8* terdiri : 7 dokumen ekonomi  
6 dokumen hiburan  
8 dokumen kesehatan  
9 dokumen olahraga  
9 dokumen otomotif  
9 dokumen pendidikan  
8 dokumen politik  
9 dokumen teknologi

Pada uji coba 72 dokumen, diperoleh hasil pengclusteran sistem dokumen mengumpul disalah satu *cluster*(*cluster 8*) sebanyak 65 dokumen mengumpul di *cluster* ini, sedangkan *cluster 1* sampai dengan *cluster 7* beranggotakan masing – masing 1 dokumen. Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.11.

Pada tabel 4.18 dapat diketahui bahwa nilai *recall* pada *cluster 1* sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster 8*, nilai *recallnya* kecil yaitu sebesar 0.138462, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.



**Gambar 4.11** grafik jumlah dokumen hasil *clustering* sistem pada uji coba kesembilan

**Tabel 4.18** Evaluasi efektivitas uji coba kesembilan (72 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	8	0	1	0.111111	0.2
<i>cluster2</i>	1	8	0	1	0.111111	0.2
<i>cluster3</i>	1	8	0	1	0.111111	0.2
<i>cluster4</i>	1	8	0	1	0.111111	0.2
<i>cluster5</i>	1	8	0	1	0.111111	0.2
<i>cluster6</i>	1	8	0	1	0.111111	0.2
<i>cluster7</i>	1	8	0	1	0.111111	0.2
<i>cluster8</i>	9	0	56	0.138462	1	0.243243

Nilai *precision* pada *cluster 1* sampai dengan *cluster 7* adalah 0,111111 karena dari 9 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang tercluster. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster 1* sampai dengan *cluster 7* nilainya sama (0,2) dan nilai  $F_1$  *measure* pada *cluster* terakhir adalah

0.243243. Pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba kesepuluh digunakan 80 dokumen, dapat dilihat pada tabel 4.19.

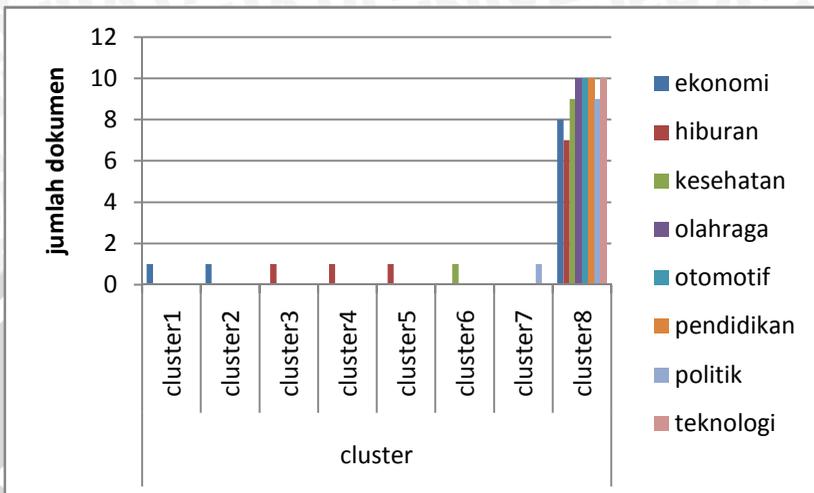
**Tabel 4.19** jumlah dokumen setiap kategori pada uji coba kesepuluh

kategori	jumlah
ekonomi	10
hiburan	10
kesehatan	10
olahraga	10
otomotif	10
pendidikan	10
politik	10
teknologi	10

Berikut merupakan hasil *clustering* pada uji coba kesepuluh pada sistem, anggota *clusternya* terdiri dari:

- *Cluster 1* terdiri : 1 dokumen ekonomi
- *Cluster 2* terdiri : 1 dokumen ekonomi
- *Cluster 3* terdiri : 1 dokumen hiburan
- *Cluster 4* terdiri : 1 dokumen hiburan
- *Cluster 5* terdiri : 1 dokumen hiburan
- *Cluster 6* terdiri : 1 dokumen kesehatan
- *Cluster 7* terdiri : 1 dokumen politik
- *Cluster 8* terdiri : 8 dokumen ekonomi  
7 dokumen hiburan  
9 dokumen kesehatan  
10 dokumen olahraga  
10 dokumen otomotif  
10 dokumen pendidikan  
9 dokumen politik  
10 dokumen teknologi

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.12.



**Gambar 4.12** grafik jumlah dokumen hasil *clustering* sistem pada uji coba kesepuluh

Pada uji coba 80 dokumen, diperoleh hasil peng*clustering* sistem dokumen mengumpul disalah satu *cluster*(*cluster* 8) sebanyak 73 dokumen mengumpul di *cluster* ini, sedangkan *cluster* 1 sampai dengan *cluster* 7 beranggotakan masing – masing 1 dokumen.

**Tabel 4.20** Evaluasi efektivitas uji coba kesepuluh (80 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	9	0	1	0.1	0.181818
<i>cluster2</i>	1	9	0	1	0.1	0.181818
<i>cluster3</i>	1	9	0	1	0.1	0.181818
<i>cluster4</i>	1	9	0	1	0.1	0.181818
<i>cluster5</i>	1	9	0	1	0.1	0.181818
<i>cluster6</i>	1	9	0	1	0.1	0.181818
<i>cluster7</i>	1	9	0	1	0.1	0.181818
<i>cluster8</i>	10	0	63	0.136986	1	0.240964

Pada tabel 4.20 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk

diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recall*-nya kecil yaitu sebesar 0.136986, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.1 karena dari 10 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang ter*cluster*. Pada *cluster* terakhir nilai *precision*-nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.181818) dan nilai  $F_1$  *measure* pada *cluster* terakhir adalah 0.240964. Pada *cluster* terakhir nilai *recall*-nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba kesebelas digunakan 88 dokumen, dapat dilihat pada tabel 4.21.

**Tabel 4.21** jumlah dokumen setiap kategori pada uji coba kesebelas

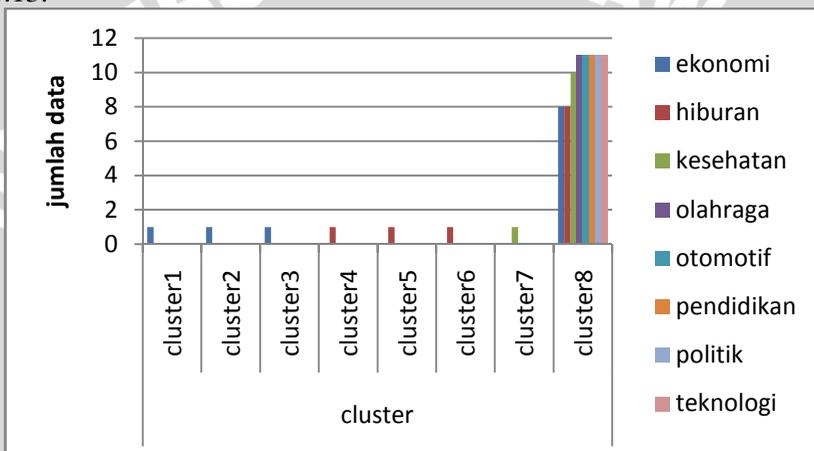
kategori	jumlah
ekonomi	11
hiburan	11
kesehatan	11
olahraga	11
otomotif	11
pendidikan	11
politik	11
teknologi	11

Berikut merupakan hasil *clustering* pada uji coba kesebelas pada sistem, anggota *clusternya* terdiri dari:

- *Cluster* 1 terdiri : 1 dokumen ekonomi
- *Cluster* 2 terdiri : 1 dokumen ekonomi
- *Cluster* 3 terdiri : 1 dokumen ekonomi
- *Cluster* 4 terdiri : 1 dokumen hiburan
- *Cluster* 5 terdiri : 1 dokumen hiburan
- *Cluster* 6 terdiri : 1 dokumen hiburan
- *Cluster* 7 terdiri : 1 dokumen kesehatan

- *Cluster* 8 terdiri : 8 dokumen ekonomi  
 8 dokumen hiburan  
 10 dokumen kesehatan  
 11 dokumen olahraga  
 11 dokumen otomotif  
 11 dokumen pendidikan  
 11 dokumen politik  
 11 dokumen teknologi

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.13.



**Gambar 4.13** grafik jumlah dokumen hasil *clustering* sistem pada uji coba kesebelas

Pada uji coba 88 dokumen, diperoleh hasil peng*clustering* sistem dokumen mengumpul disalah satu *cluster*(*cluster* 8) sebanyak 81 dokumen mengumpul di *cluster* ini, sedangkan *cluster* 1 sampai dengan *cluster* 7 beranggotakan masing – masing 1 dokumen.

**Tabel 4.22** Evaluasi efektivitas uji coba kesebelas (88 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster</i> 1	1	10	0	1	0.090909	0.166667
<i>cluster</i> 2	1	10	0	1	0.090909	0.166667
<i>cluster</i> 3	1	10	0	1	0.090909	0.166667
<i>cluster</i> 4	1	10	0	1	0.090909	0.166667

<i>cluster5</i>	1	10	0	1	0.090909	0.166667
<i>cluster6</i>	1	10	0	1	0.090909	0.166667
<i>cluster7</i>	1	10	0	1	0.090909	0.166667
<i>cluster8</i>	11	0	70	0.135802	1	0.23913

Pada tabel 4.22 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recall*nya kecil yaitu sebesar 0.135802, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.090909 karena dari 11 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang tercluster. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.166667) dan nilai  $F_1$  *measure* pada *cluster* terakhir adalah 0.23913. Pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba keduabelas digunakan 96 dokumen, dapat dilihat pada tabel 4.23.

**Tabel 4.23** jumlah dokumen setiap kategori pada uji coba kesebelas

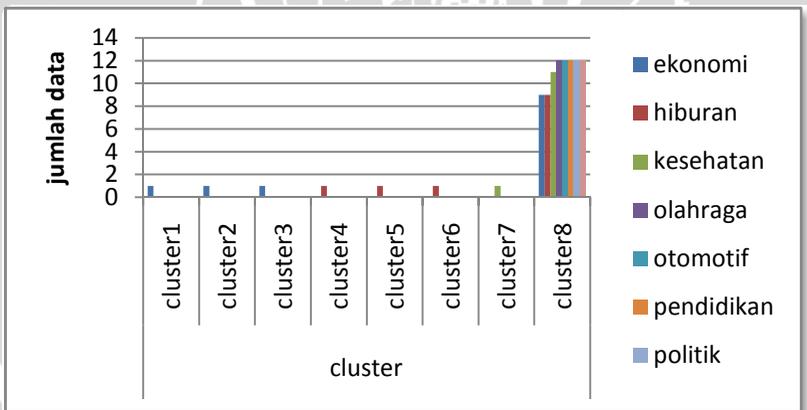
kategori	jumlah
ekonomi	12
hiburan	12
kesehatan	12
olahraga	12
otomotif	12
pendidikan	12
politik	12
teknologi	12

Berikut merupakan hasil *clustering* pada uji coba kesebelas pada sistem, anggota *clusternya* terdiri dari:

- *Cluster 1* terdiri : 1 dokumen ekonomi
- *Cluster 2* terdiri : 1 dokumen ekonomi
- *Cluster 3* terdiri : 1 dokumen ekonomi
- *Cluster 4* terdiri : 1 dokumen hiburan
- *Cluster 5* terdiri : 1 dokumen hiburan
- *Cluster 6* terdiri : 1 dokumen hiburan
- *Cluster 7* terdiri : 1 dokumen kesehatan
- *Cluster 8* terdiri : 9 dokumen ekonomi  
9 dokumen hiburan  
11 dokumen kesehatan  
12 dokumen olahraga  
12 dokumen otomotif  
12 dokumen pendidikan  
12 dokumen politik

Pada uji coba 96 dokumen, diperoleh hasil pengclusteran sistem dokumen mengumpul disalah satu *cluster* (*cluster 8*) sebanyak 89 dokumen mengumpul di *cluster* ini, sedangkan *cluster 1* sampai dengan *cluster 7* beranggotakan masing – masing 1 dokumen.

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.14.



**Gambar 4.14** grafik jumlah dokumen hasil *clustering* sistem pada uji coba keduabelas

**Tabel 4.24** Evaluasi efektivitas uji coba keduabelas (96 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	11	0	1	0.083333	0.153846
<i>cluster2</i>	1	11	0	1	0.083333	0.153846
<i>cluster3</i>	1	11	0	1	0.083333	0.153846
<i>cluster4</i>	1	11	0	1	0.083333	0.153846
<i>cluster5</i>	1	11	0	1	0.083333	0.153846
<i>cluster6</i>	1	11	0	1	0.083333	0.153846
<i>cluster7</i>	1	11	0	1	0.083333	0.153846
<i>cluster8</i>	12	0	77	0.134831	1	0.237624

Dapat dilihat pada tabel 4.24 diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recall*nya kecil yaitu sebesar 0.134831, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.083333 karena dari 12 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang tercluster. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.153846) dan nilai  $F_1$  *measure* pada *cluster* terakhir adalah 0.237624. Pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba ketigabelas digunakan 104 dokumen, dapat dilihat pada tabel 4.25.

**Tabel 4.25** jumlah dokumen setiap kategori pada uji coba ketigabelas

kategori	jumlah
ekonomi	13
hiburan	13
kesehatan	13

olahraga	13
otomotif	13
pendidikan	13
politik	13
teknologi	13

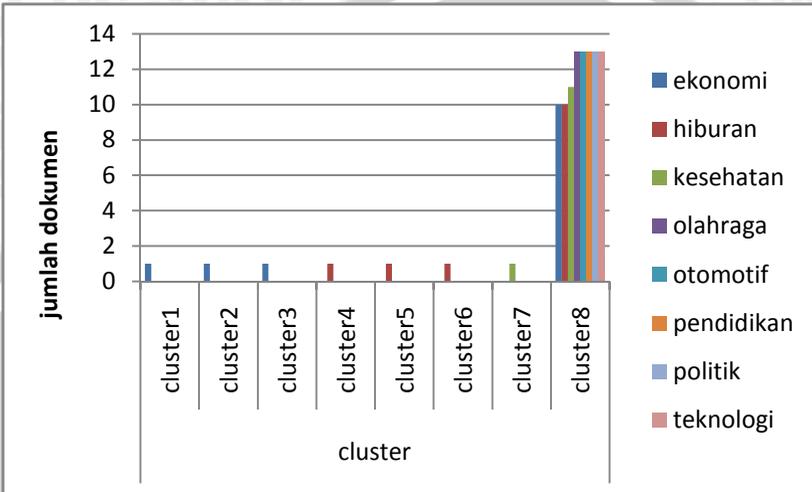
Berikut merupakan hasil *clustering* pada uji coba ketigabelas pada sistem, anggota *clusternya* terdiri dari:

- *Cluster* 1 terdiri : 1 dokumen ekonomi
- *Cluster* 2 terdiri : 1 dokumen ekonomi
- *Cluster* 3 terdiri : 1 dokumen ekonomi
- *Cluster* 4 terdiri : 1 dokumen hiburan
- *Cluster* 5 terdiri : 1 dokumen hiburan
- *Cluster* 6 terdiri : 1 dokumen hiburan
- *Cluster* 7 terdiri : 1 dokumen kesehatan
- *Cluster* 8 terdiri : 10 dokumen ekonomi  
10 dokumen hiburan  
12 dokumen kesehatan  
13 dokumen olahraga  
13 dokumen otomotif  
13 dokumen pendidikan  
13 dokumen politik  
13 dokumen teknologi

Pada uji coba 104 dokumen, diperoleh hasil peng*clusteran* sistem dokumen mengumpul disalah satu *cluster(cluster* 8) sebanyak 97 dokumen mengumpul di *cluster* ini, sedangkan *cluster* 1 sampai dengan *cluster* 7 beranggotakan masing – masing 1 dokumen. Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.15.

Pada tabel 4.26 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai *recallnya* kecil yaitu sebesar 0.134021, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall*

dihitung berdasarkan salah satu kategori yang jumlahnya mengumpul banyak di *cluster* ini.



**Gambar 4.15** grafik jumlah dokumen hasil *clustering* sistem pada uji coba ketigabelas

**Tabel 4.26** Evaluasi efektivitas uji coba ketigabelas (104 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	12	0	1	0.076923077	0.142857
<i>cluster2</i>	1	12	0	1	0.076923077	0.142857
<i>cluster3</i>	1	12	0	1	0.076923077	0.142857
<i>cluster4</i>	1	12	0	1	0.076923077	0.142857
<i>cluster5</i>	1	12	0	1	0.076923077	0.142857
<i>cluster6</i>	1	12	0	1	0.076923077	0.142857
<i>cluster7</i>	1	12	0	1	0.076923077	0.142857
<i>cluster8</i>	13	0	84	0.134021	1	0.236364

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.076923077 karena dari 13 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang digunakan yang tercluster. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.142857) dan nilai  $F_1$  *measure* pada *cluster* terakhir adalah 0.236364. Pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba keempatbelas digunakan 112 dokumen, dapat dilihat pada tabel 2.7.

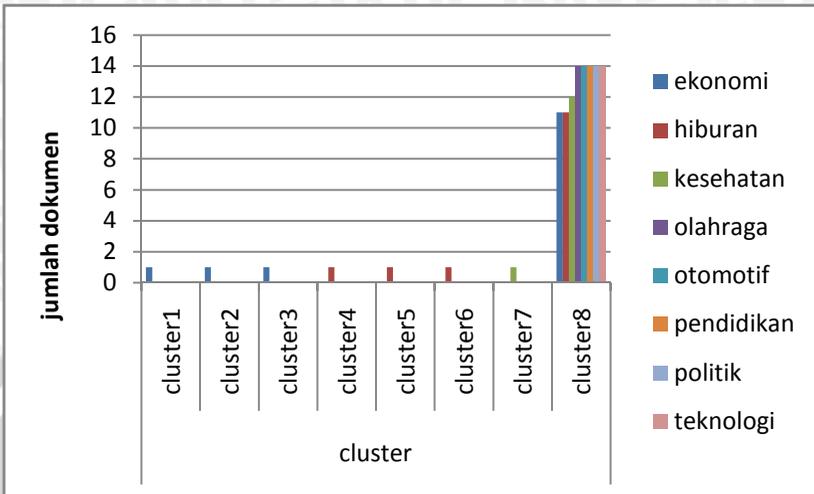
**Tabel 4.27** jumlah dokumen setiap kategori pada uji coba keempatbelas

kategori	jumlah
ekonomi	14
hiburan	14
kesehatan	14
olahraga	14
otomotif	14
pendidikan	14
politik	14
teknologi	14

Berikut merupakan hasil *clustering* pada uji coba keempatbelas pada sistem, anggota *clusternya* terdiri dari:

- *Cluster 1* terdiri : 1 dokumen ekonomi
- *Cluster 2* terdiri : 1 dokumen ekonomi
- *Cluster 3* terdiri : 1 dokumen ekonomi
- *Cluster 4* terdiri : 1 dokumen hiburan
- *Cluster 5* terdiri : 1 dokumen hiburan
- *Cluster 6* terdiri : 1 dokumen hiburan
- *Cluster 7* terdiri : 1 dokumen kesehatan
- *Cluster 8* terdiri : 11 dokumen ekonomi  
11 dokumen hiburan  
13 dokumen kesehatan  
14 dokumen olahraga  
14 dokumen otomotif  
14 dokumen pendidikan  
14 dokumen politik  
14 dokumen teknologi

Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.16.



**Gambar 4.16** grafik jumlah dokumen hasil *clustering* sistem pada uji coba keempatbelas

Pada uji coba 112 dokumen, diperoleh hasil peng*clustering* sistem dokumen mengumpul disalah satu *cluster*(*cluster* 8) sebanyak 105 dokumen mengumpul di *cluster* ini, sedangkan *cluster* 1 sampai dengan *cluster* 7 beranggotakan masing – masing 1 dokumen.

**Tabel 4.28** Evaluasi efektivitas uji coba keempatbelas (112 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster1</i>	1	13	0	1	0.071429	0.133333333
<i>cluster2</i>	1	13	0	1	0.071429	0.133333333
<i>cluster3</i>	1	13	0	1	0.071429	0.133333333
<i>cluster4</i>	1	13	0	1	0.071429	0.133333333
<i>cluster5</i>	1	13	0	1	0.071429	0.133333333
<i>cluster6</i>	1	13	0	1	0.071429	0.133333333
<i>cluster7</i>	1	13	0	1	0.071429	0.133333333
<i>cluster8</i>	14	0	91	0.133333	1	0.235294118

Pada tabel 4.28 dapat diketahui bahwa nilai *recall* pada *cluster* 1 sampai dengan 7 nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster* 8, nilai

*recall*nya kecil yaitu sebesar 0.133333, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster* 1 sampai dengan *cluster* 7 adalah 0.071429 karena dari 14 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang tercluster. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster* 1 sampai dengan *cluster* 7 nilainya sama (0.133333333) dan nilai  $F_1$  *measure* pada *cluster* terakhir adalah 0.235294118. Pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

Uji coba kelimabelas digunakan 120 dokumen, dapat dilihat pada tabel 4.29.

**Tabel 4.29** jumlah dokumen setiap kategori pada uji coba kelimabelas

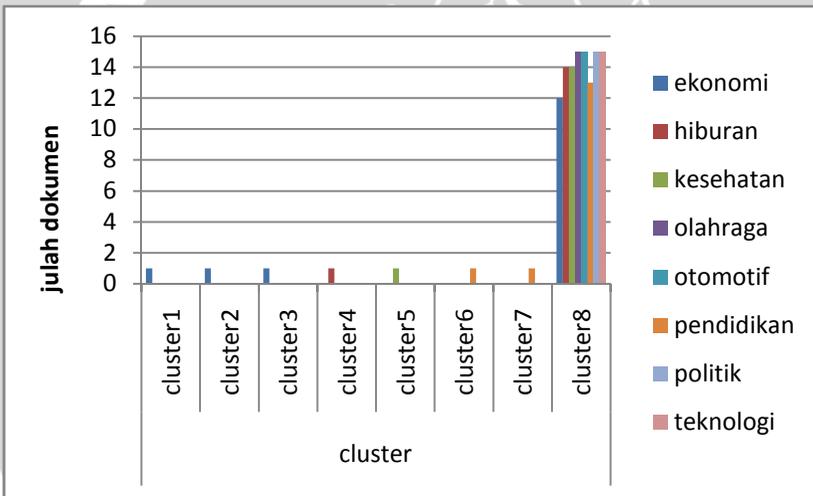
kategori	jumlah
ekonomi	15
hiburan	15
kesehatan	15
olahraga	15
otomotif	15
pendidikan	15
politik	15
teknologi	15

Berikut merupakan hasil *clustering* pada uji coba kelimabelas pada sistem, anggota *clusternya* terdiri dari:

- *Cluster* 1 terdiri : 1 dokumen ekonomi
- *Cluster* 2 terdiri : 1 dokumen ekonomi
- *Cluster* 3 terdiri : 1 dokumen ekonomi
- *Cluster* 4 terdiri : 1 dokumen hiburan
- *Cluster* 5 terdiri : 1 dokumen kesehatan
- *Cluster* 6 terdiri : 1 dokumen pendidikan
- *Cluster* 7 terdiri : 1 dokumen pendidikan

- *Cluster* 8 terdiri : 12 dokumen ekonomi  
 14 dokumen hiburan  
 14 dokumen kesehatan  
 15 dokumen olahraga  
 15 dokumen otomotif  
 13 dokumen pendidikan  
 15 dokumen politik  
 15 dokumen teknologi

Pada uji coba 120 dokumen, diperoleh hasil pengclusteran sistem dokumen mengumpul disalah satu *cluster*(*cluster* 8) sebanyak 113 dokumen mengumpul di *cluster* ini, sedangkan *cluster* 1 sampai dengan *cluster* 7 beranggotakan masing – masing 1 dokumen. Dapat digambarkan dalam bentuk grafik seperti pada gambar 4.17.



**Gambar 4.17** grafik jumlah dokumen hasil *clustering* sistem pada uji coba kelimabelas

**Tabel 4.30** Evaluasi efektivitas uji coba kelimabelas (120 dokumen)

	a	b	c	<i>recall</i>	<i>precision</i>	$F_1$ <i>measure</i>
<i>cluster</i> 1	1	14	0	1	0.066667	0.125
<i>cluster</i> 2	1	14	0	1	0.066667	0.125
<i>cluster</i> 3	1	14	0	1	0.066667	0.125
<i>cluster</i> 4	1	14	0	1	0.066667	0.125

<i>cluster5</i>	1	14	0	1	0.066667	0.125
<i>cluster6</i>	1	14	0	1	0.066667	0.125
<i>cluster7</i>	1	14	0	1	0.066667	0.125
<i>cluster8</i>	15	0	98	0.132743	1	0.234375

Pada tabel 4.30 dapat diketahui bahwa nilai *recall* pada *cluster 1* sampai dengan *7* nilainya adalah 1, karena hanya 1 dokumen yang berada didalam *cluster* tersebut dan *cluster* yang terbentuk diasumsikan sebagai kategori dari hasil klasifikasi dokumen yang berada dalam *cluster* tersebut. Sedangkan pada *cluster 8*, nilai *recall*nya kecil yaitu sebesar 0.132743, karena pada *cluster* ini terdapat sejumlah dokumen dari kategori yang berbeda, nilai *recall* dihitung berdasarkan salah satu kategori yang jumlah dokumennya mengumpul banyak di *cluster* ini.

Nilai *precision* pada *cluster 1* sampai dengan *cluster 7* adalah 0.066667 karena dari 15 dokumen yang digunakan pada masing-masing kategori hanya 1 dokumen yang tercluster. Pada *cluster* terakhir nilai *precision*nya adalah 1.

Nilai  $F_1$  *measure* pada *cluster 1* sampai dengan *cluster 7* nilainya sama (0.125) dan nilai  $F_1$  *measure* pada *cluster* terakhir adalah 0.234375. Pada *cluster* terakhir nilai *recall*nya kecil sehingga mempengaruhi nilai  $F_1$  *measure*.

#### 4.5.3 Analisa hasil

Dari hasil evaluasi efektivitas, didapatkan nilai *recall*, *precision*, dan  $F_1$  *measure* berbeda – beda untuk jumlah dokumen yang berbeda (tiap uji coba). Dengan semakin banyak jumlah dokumen maka nilai *recall*, *precision*, dan  $F_1$  *measure* menjadi menurun. Kecuali pada uji coba keenam(48 dokumen), pada uji coba ini nilai *recall* naik tetapi kemudian turun lagi, naik dan turunnya nilai *recall* ini mempengaruhi nilai  $F_1$  *measure* yang terbentuk.

Dari pengujian sistem yang dilakukan didapat hasil *clustering* dokumen yang selalu mengumpul disalah satu *cluster* yang terbentuk sehingga menyebabkan nilai *recall*, *precision*, dan  $F_1$  *measure* yang sangat rendah. Hal ini disebabkan jarak antar kedekatan dokumen dihitung hanya berdasarkan banyaknya frekuensi kata yang muncul dalam dokumen tersebut, tanpa memperdulikan kata tersebut unik

atau hanya berupa kata umum yang sering muncul di tiap dokumen walaupun berbeda kategori. Sedangkan dokumen yang tidak mengumpul merupakan dokumen yang mempunyai kata – kata paling tidak umum dari semua kumpulan yang ada.

Kata yang dihitung setiap dokumen sangat dipengaruhi oleh hasil proses *preprocessing*. Proses *preprocessing* yang diterapkan dalam sistem ini menggunakan proses yang sederhana, khususnya proses *stemming*, sehingga dihasilkan kata – kata yang kurang mencerminkan isi dari dokumen tersebut. Sehingga proses selanjutnya menjadi kurang bermakna. Hal ini menyebabkan hasil *clustering* yang terbentuk tidak dapat mengklasifikasikan dokumen yang ada, karena dokumen yang mempunyai jarak terbesar terhadap kumpulan dokumen lainnya akan membentuk *cluster* tersendiri.

Dokumen yang mempunyai jarak terbesar berasal dari dokumen ekonomi, hiburan, kesehatan, politik, dan pendidikan. Untuk kategori olahraga, teknologi, dan otomotif, dokumen yang berasal dari kategori tersebut selalu mengumpul disalah satu *cluster*, dan sepanjang 15kali uji coba yang dilakukan, tidak satu kalipun dokumen dari kategori tersebut dapat memisah menjadi *cluster* tersendiri. Hal ini dikarenakan karakteristik dan jumlah kata pada kategori tersebut jangkauannya sangat luas sehingga mempunyai keterkaitan yang umum. Tidak seperti dokumen pada kategori ekonomi, hiburan, kesehatan, politik, dan pendidikan, yang dapat membentuk *cluster* tersendiri walaupun tidak secara keseluruhan mengelompok, hal ini menunjukkan terdapat perbedaan kata yang mencolok pada dokumen yang dapat membentuk *cluster* tersebut. Oleh karena itu karakteristik teks sangat mempengaruhi efektivitas dan ketepatan sistem dalam mengklasifikasikan dokumen berita.

## BAB V PENUTUP

### 5.1 Kesimpulan

Kesimpulan yang diperoleh dari pengerjaan Tugas Akhir ini adalah:

1. Telah dipelajari dan dibuat sistem pengklasifikasian berita berbahasa Indonesia dengan Algoritma *Agglomerative Single Linkage Hierarchical Clustering* yang datanya bersumber dari [www.kompas.com](http://www.kompas.com), sistem ini menghasilkan nilai *recall*, *precision* dan *F<sub>1</sub> measure* rendah karena dokumen cenderung mengumpul disalah satu *cluster*.
2. Efektivitas sistem cenderung menurun dengan meningkatnya jumlah dokumen yang digunakan.
3. Berdasarkan hasil dari percobaan yang telah dilakukan pada sistem, maka metode yang diterapkan pada sistem ini tidak dapat digunakan untuk klasifikasi dokumen berita berbahasa Indonesia.

### 5.2 Saran

Saran yang dapat diberikan setelah mengerjakan Tugas Akhir ini adalah :

1. Proses *preprocessing* khususnya *stemming* yang dibangun lebih kompleks lagi, dengan menggunakan kamus kata dasar misalnya agar hasil *stemming* yang diperoleh hasilnya lebih maksimal.
2. Penambahan algoritma *tf-idf* untuk menghitung bobot kata dalam kumpulan dokumen tersebut, agar kata-kata umum dalam kumpulan dokumen tersebut tidak terlalu mempengaruhi jarak antar dokumen yang didapat.

UNIVERSITAS BRAWIJAYA



## DAFTAR PUSTAKA

- Arsono, Julio. **Parsing dan Stemming**. <http://web.ipb.ac.id/~julio/webaku/isi/kom471/slides>. diakses pada tanggal 27 April 2009.
- Budhi, G.S., RAharjo, A.L., dan Taufik, H. 2008. **Hierarchical Clustering Untuk Aplikasi Automated Text Integration**. Seminar Nasional Aplikasi Teknologi Informasi. Yogyakarta.  
[http://fportfolio.petra.ac.id/user\\_files/Hierarchical](http://fportfolio.petra.ac.id/user_files/Hierarchical)  
Diakses pada tanggal 12 Maret 2009
- Even, Yahir dan Zohar. 2002. **Introduction to Text Mining. Automated Learning Group National Center For Supercomputing Applications**. University of Illionis.  
<http://aldocs.nasca.uiuc.edu/PR-2021116-2.ppt>  
Diakses pada tanggal 27 April 2009.
- G. Karypis, E.H. Han, V. Kumar. 1999. **Chameleon: a Hierarchical Clustering Algorithm Using Dynamic Modeling**. IEEE Computer: Special Issue on Data Analysis and Mining 32(8):68W5.
- Hartini, Entin. 2009. **Metode Clustering Hirarki**. Pusat Pengembangan Teknologi Informasi dan Komputasi BATAN. <http://lecturer.eepis-its.edu/entin/Clustering.pdf>  
Diakses pada tanggal 11 Juni 2009.
- Hearst, Marti. 2003. **What is text mining?**. SIMS,UC Berkeley.  
<http://www.sims.berkeley.edu/~hearst/text-mining.html>.

Diakses pada tanggal 14 Nopember 2009.

Manning, Christoper D Prabhakar Rasghavan dan Hinrich Schutze. 2006. ***An Introduction to Information Retrieval***. Cambridge University Press.

Ma'lufatul, Maisuroh.2007. **Penerapan *Text Mining* Dalam Pengklasifikasian Berita Berbahasa Indonesia dengan Algoritma *Naïve Bayes Classifier***. Skripsi Matematika dan Ilmu Pengetahuan Alam Unibraw.

Porter, M. F.. 1980. ***An Algorithm for Suffix Stripping***. <http://www.tartarus.org/~martin/PorterStemmer/def.txt>.  
Diakses pada tanggal :1 Juni 2009.

Ridho, Ali Barakbah, K. Arai. 2004. ***Determining Constraints of Moving Variance to Find Global Optimum and Make Automatic Clustering***. In. IES. Politeknik Elektronika Negeri Surabaya, ITS.

Ridho, Ali Barakbah. 2006. ***Clustering***. In Workshop *Data Mining Teknologi Informasi Politeknik Elektronika Negeri Surabaya*.  
[http://lecturer.eepis-its.edu/entin/Machine Learning/Minggu20Clustering.pdf](http://lecturer.eepis-its.edu/entin/Machine%20Learning/Minggu20Clustering.pdf)  
Diakses pada tanggal 12 Maret 2009

Suharso, Wildan. 2008. **Klasifikasi Dokumen Teks Berbahasa Indonesia Menggunakan Metode *Rocchio***. Skripsi Matematika dan Ilmu Pengetahuan Alam Unibraw.

Tala, Fadillah. 2003. *A Study of Stemming E? etc on Information Retrieval in Bahasa Indonesia*. Master of Logic Project-Institute for Logic, Language and Computation Universiteit van Amsterdam, The Netherlands.

Teknomo, Kardi. 2005. *K-Means Clustering Tutorials*. <http://people.revoledu.com/kardi/tutorial/kMean>  
Diakses pada tanggal 27 April 2009.

Wibisono, Yudi dan Masayu Leylia. 2005. *Clustering Berita Berbahasa Indonesia*. [http://fpmipa.upi.edu/staff/yudi/KNSI\\_Clustering\\_yudi\\_masayu.pdf](http://fpmipa.upi.edu/staff/yudi/KNSI_Clustering_yudi_masayu.pdf)  
Diakses pada tanggal 3 Maret 2009

Yiming Yang,Jaime G.Carbonell, Rulf D. Brown, Thomas Pierce, Brian T. Achibald, Xin Liu. *LearningApproaches for Detecting and TrackingNews Events*. IEEE Intelligent Systems ,Language Techlonolies Institute,Carbegie Mellon University, 1999

Zainal, Agus dan Ari Novan. *Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering*. Teknologi Informasi Institut Teknologi Sepuluh Nopember.  
Institut Teknologi Sepuluh Nopember(ITS).Surabaya.  
<http://agusza.its-sby.edu/hirodai/SITIAKlasifikasiEvent.pdf>.  
Diakses pada tanggal 10 Maret 2009

UNIVERSITAS BRAWIJAYA



## LAMPIRAN 1

### Daftar stopwords

1	ada	37	boleh	73	jadi
2	agak	38	buat	74	jangan
3	agar	39	bukan	75	jangan
4	akan	40	cuma	76	jika
5	akhir	41	dahulu	77	jikalau
6	aku	42	dalam	78	juga
7	amat	43	dan	79	justru
8	anda	44	dapat	80	kala
9	antar	45	dari	81	kalau
10	antara	46	daripada	82	kali
11	apa	47	dekat	83	kalian
12	apaan	48	demi	84	kami
13	apabila	49	demikian	85	kamu
14	apalagi	50	dengan	86	kan
15	apatah	51	depan	87	kapan
16	atau	52	di	88	karena
17	bagai	53	dia	89	ke
18	bagaikan	54	diantara	90	kecil
19	bagaimana	55	dini	91	khusus
20	bagi	56	diri	92	kian
21	bahkan	57	dong	93	kini
22	bahwa	58	dulu	94	kira
23	bahwas	59	enggak	95	kita
24	balik	60	entah	96	kok
25	banyak	61	hadap	97	lagi
26	beberapa	62	hai	98	lagian
27	begini	63	hampir	99	lah
28	beginian	64	hanya	100	lain
29	begitu	65	harus	101	laku
30	belum	66	hendak	102	lalu
31	benar	67	hingga	103	lama
32	berapa	68	ia	104	lebih
33	betul	69	ibarat	105	macam
34	biasa	70	ingin	106	maka
35	bila	71	ini	107	makin
36	bisa	72	itu	108	malah

109	malahan	146	saling	183	sela
110	mampu	147	sama	184	selain
111	mana	148	sambil	185	selaku
112	manakala	149	sampai	186	selalu
113	manalagi	150	sana	187	selama
114	mang	151	sangat	188	seluruh
115	masih	152	saya	189	semacam
116	masing	153	se	190	semakin
117	mau	154	sebab	191	semasih
118	meski	155	sebagai	192	semau
119	mudi	156	sebagaimana	193	sementara
120	mula	157	sebalik	194	sempat
121	mungkin	158	sebanyak	195	semua
122	nah	159	sebegini	196	semula
123	namun	160	sebegitu	197	sendiri
124	nanti	161	sebelum	198	seolah
125	napa	162	sebenar	199	seorang
126	nyaris	163	seberapa	200	sepanjang
127	olah	164	sebetul	201	sepantas
128	oleh	165	sebisa	202	seperti
129	orang	166	sebuah	203	sering
130	pada	167	sebut	204	serta
131	padahal	168	sedang	205	serupa
132	paling	169	sedemikian	206	sesaat
133	panjang	170	sedikit	207	sesama
134	pantas	171	segala	208	sesegera
135	para	172	segera	209	sesekali
136	pasti	173	seharus	210	seseorang
137	per	174	sehingga	211	sesuatu
138	pernah	175	sejak	212	sesudah
139	pula	176	sejenak	213	setelah
140	pun	177	sekali	214	seterus
141	reka	178	sekaligus	215	setiap
142	rupa	179	sekarang	216	setidak
143	rupa	180	seketika	217	sewaktu
144	saat	181	sekira	218	siapa
145	saja	182	sekitar	219	sini

220	suatu	257	f	294	oktober
221	sudah	258	g	295	november
222	supaya	259	h	296	desember
223	tadi	260	i	297	jam
224	tak	261	j	298	menit
225	tanpa	262	k	299	detik
226	tapi	263	l	300	hari
227	telah	264	m	301	bulan
228	tentang	265	n	302	tahun
229	tentu	266	o		
230	terus	267	p		
231	tetapi	268	q		
232	tiap	269	r		
233	tidak	270	s		
234	tika	271	t		
235	umpama	272	u		
236	untuk	273	v		
237	toh	274	w		
238	waduh	275	x		
239	wah	276	y		
240	wahai	277	z		
241	waktu	278	senin		
242	walau	279	selasa		
243	walaupun	280	rabu		
244	wong	281	kamis		
245	yaitu	282	jumat		
246	yakni	283	sabtu		
247	yang	284	minggu		
248	kompas	285	januari		
249	com	286	pebruari		
250	jakarta	287	maret		
251	www	288	april		
252	a	289	mei		
253	b	290	juni		
254	c	291	juli		
255	d	292	agustus		
256	e	293	september		

UNIVERSITAS BRAWIJAYA



## LAMPIRAN 2

### Contoh dokumen yang diklasifikasikan

EAgoes.txt

Agoes, dari Iseng Jadi Pengusaha Jamur

Iptek.net.id

Jamur Kayu (Ganoderma lucidum(Leyss.ex Fr.) Karst.)

/

Artikel Terkait:

- \* Tati Hartati, Inspirasi Ibu untuk Ruman Dannis
- \* Saptu, Himpun Laba dari Bisnis Narsis
- \* Eva, Kerupuk Kelempang Beromzet Puluhan Juta
- \* Martini, Dengan Rp 250.000 Raup Euro dan Dollar AS
- \* A Pramono, Dari Office Boy Jadi Miliarder

Jumat, 10 Juli 2009 | 14:32 WIB

KOMPAS.com - Agoes Poernomo, pendiri usaha jamur UD Payung Manfaat, tak pernah membayangkan bahwa ia bakal menjadi pengusaha jamur seperti sekarang. Semula, pria kelahiran Blitar ini hanya iseng mencoba membuat bibit jamur merang di sekeliling rumahnya. Apalagi, keluarganya penyuka masakan jamur. Ternyata, selain hasilnya bagus, jamurnya juga disukai tetangganya. Agoes pun mulai memperbanyak bibit jamur yang disemainya.

Namun, aktivitas yang dilakoninya sejak tahun 1976 itu hanya bertahan setahun. Sebab, informasi tentang jamur masih sangat terbatas. Ketika itu, bapak dua anak yang masih berstatus pegawai di Departemen Kesehatan di Blitar itu akhirnya kembali menekuni bisnis sampingannya yang lain, yaitu usaha bengkel motor, pembuatan mebel kayu, dan kerajinan batu.

Meski tidak lagi membuat bibit jamur, Agoes terus mengikuti perkembangan bisnis jamur di Blitar. Uniknya, setelah tak menyentuh bisnis jamur beberapa tahun, ternyata nasib membawanya kembali ke budidaya jamur.

Lantaran melihat besarnya peluang bisnis jamur di Blitar dan jumlah pemainnya yang masih sedikit, Agoes memutuskan untuk kembali menekuni budidaya jamur pada 1996. Tapi, ia fokus menggarap bisnis jamur kayu. "Pertimbangannya secara ekonomi, produksinya lebih gampang," papar Agoes.

Kali ini, Agoes serius menekuni bisnis ini. Pria berumur 59 tahun ini membangun satu rumah kecil di Desa Sumberdiren sebagai pusat usahanya. Ia memajang nama Toko Jamur. Untuk semua ini, dia mengeluarkan modal sekitar Rp 6 juta.

Karena ingin lebih fokus di bisnis jamur, Agoes mengajukan pensiun dini di tempatnya bekerja pada tahun 2000. Selanjutnya, bersama putra pertamanya, Agoes makin serius menekuni bisnis jamur. Ia mengubah nama usahanya menjadi UD Payung Manfaat.

Bisnis Agoes mengalami masa keemasan dari tahun 1998 hingga 2000. Kala itu, dia bisa memproduksi jamur kering maupun jamur segar hingga 6 kuintal setiap bulan.

Agoes pun rajin memberikan penyuluhan tentang budidaya jamur kepada para petani. Bahkan,

saat ini, di Blitar saja ia punya sekitar 300 petani plasma binaan. Omzet di tahun pertamanya yang hanya Rp 10 juta per bulan kini berkembang hingga lima kali lipat.

UD Payung Manfaat sudah bisa memproduksi bibit sendiri dan mengolah jamur menjadi berbagai jenis makanan. Dengan dua pabrik serta rumah jamur yang dimilikinya, aset usaha Agoes kini mencapai Rp 700 juta.

Belajar secara otodidak

Menariknya, Agoes mempelajari semuanya dengan otodidak. Ia tak segan melancong ke luar daerah untuk belajar lebih dalam tentang jamur dari petani jamur lain.

Namun, meski tidak mempunyai pengetahuan cukup, Agoes memiliki semangat tinggi untuk belajar. Pada masa awal usahanya, di 1996, anak keenam dari 11 bersaudara ini belajar budidaya jamur lewat berbagai literatur, termasuk buku panduan terbitan dinas pertanian dan kehutanan. “Dua tahun pertama memang berat, cari informasi ke mana-mana, juga cari bibit sampai ke luar daerah,” kenang pria 59 tahun ini.

Setelah merasa memiliki pengetahuan lumayan memadai, ia mulai menjalin kerjasama dengan Fakultas Pertanian Universitas Brawijaya Malang dan Universitas Airlangga Surabaya. Dengan kedua kampus itu, ia menggelar penelitian dan pengembangan budidaya jamur. Dari sini, ia belajar soal pemilihan media tanam.

Bersama putra pertamanya, Agung Hidayanto, Agoes juga beberapa kali melakukan studi banding ke petani jamur di wilayah Lembang, Bogor dan Yogyakarta. Tujuannya untuk mendalami budidaya jamur secara langsung.

Berbekal ilmu dari petani, dia belajar memodifikasi cara pembudidayaan mereka. Ternyata langkah-langkah itu membuahkan hasil. Bahkan, ia sukses membudidayakan jamur di wilayah Blitar, daerah dataran rendah yang sebenarnya kurang cocok untuk bertanam jamur.

Produktivitas dan kualitas tanaman jamurnya tak kalah dengan yang dihasilkan petani di dataran tinggi. Maka, dari sekedar budidaya, pada awal 1998, ia membuka toko untuk berjualan jamur.

Lini usaha Agoes kian bertambah. Kalau awalnya ia hanya memproduksi jamur kayu segar dan kering, sejak punya toko, dia mulai menjual bibit jamur dan media tanamnya. Pada tahun-tahun tersebut, dalam sebulan ia bisa menghasilkan 6 kuintal jamur segar, 4 kuintal jamur kering dan sekitar 3.000 media tanam yang sudah berbibit.

Pada 2000, Agoes melangkah lebih jauh lagi. Ia tak hanya mengolah jamur menjadi makanan seperti kripik jamur, kopi jamur, hingga kapsul jamur. Ia juga mengganti mereknya, dari semula Toko Jamur, menjadi merek Payung Manfaat. Ia memilih nama itu agar sesuai dengan visi usahanya yang ingin memberi perlindungan (payung) kepada masyarakat.

Di tahun yang sama, dia juga mendapatkan sertifikat dari Dinas Kesehatan untuk semua proses dan hasil produknya. Melihat berbagai kemajuan itu, Agoes makin jatuh hati pada jamur. Agar bisa lebih fokus mengembangkan Payung Manfaat, pria lulusan SMA ini pun mengajukan pensiun dini pada tahun 2000, enam tahun lebih cepat dari jadwal pensiunnya.

Sejak pensiun dini, Agoes kian optimistis terus mengembangkan usahanya. Dalam jangka pendek ini, dia ingin meningkatkan kapasitas produksinya. Selain itu, lelaki tamatan SMA ini terus melakukan inovasi, baik terkait bahan baku, proses pengolahan, hasil produksi, maupun pemasarannya.

Semua itu adalah bagian dari upayanya memujudkan impian besarnya. “Ekspansi ke seluruh Nusantara dengan pengembangan plasma adalah impian terbesar saya,” ujar kakek satu cucu ini.

Agoes yakin, pengembangan plasma di seluruh daerah Indonesia tidak terlalu susah. Sebab, sejak awal dia sudah melakukan pelatihan gratis kepada para petani jamur. Bukan hanya untuk petani di Blitar, Agoes juga memberikan pelatihan kepada para petani dari berbagai daerah lain. “Ini menjadi kebahagiaan tersendiri. Banyak orang bisa belajar dari pengalaman saya,” tuturnya.

Kerelaan Agoes berbagi ilmu inilah yang membuatnya sukses mengembangkan jaringan plasma selama ini. Saat ini dia sudah punya 300 petani plasma di Blitar. Dalam skema plasma ini, Agoes menyediakan bibit jamur dan menampung hasil panen. Jadi petani tinggal memelihara tanamannya saja.

Ke depan, Agoes berharap bisa merangkul ribuan orang yang pernah mendapat pelatihan darinya, khususnya yang berasal dari luar Jawa. Agoes akan memberi bimbingan dan konsultasi bagi mereka yang mau menjadi petani plasma. Dia akan memanfaatkan teknologi agar konsultasi bisa kontinyu.

Seiring usianya yang kian bertambah, Agoes mundur perlahan dari kepadatan aktivitas bisnisnya. Sejak tahun 2000, meski masih memegang kendali bisnis, dia tak lagi turun tangan langsung memberi pelatihan.

Lalu sejak 2006, dia mendelegasikan kendali bisnis kepada putra sulungnya, Agung Hidayanto. Impian besarnya pun dia titipkan ke pundak Agung yang sejak awal ikut membesarkan Payung Manfaat. “Sekarang saya hanya turun tangan untuk pembibitan saja,” tuturnya.

Sebagai pegangan bagi penerus bisnisnya, Agoes mewariskan prinsip utama yang selalu dipegangnya selama ini. “Kunci sukses berbisnis adalah ketekunan, serta tidak pelit berbagi ilmu, karena itu sebuah bentuk ibadah,” ujar Agoes.

Dia juga berharap putranya bisa mengembangkan bisnis jamurnya dengan menerapkan teknologi tepat guna. “Dan jangan lupa menjadikan pengalaman sebagai guru terbaik,” ujar Agoes membagikan nasihatnya kepada putra tertuanya itu.

HResmi.txt

Resmi Bercerai, Chintami Atmanegara Menangis

Rabu, 17 Juni 2009 - 11:27 wib

JAKARTA - Pengadilan Agama Jakarta Selatan akhirnya mengabulkan gugatan cerai yang diajukan Chintami Atmanegara terhadap suaminya, Bayu Maruto.

“Akhirnya majelis hakim memenuhi keputusan saya (bercerai),” kata Chintami usai mengikuti sidang di Pengadilan Agama Jakarta Selatan, Jakarta, Rabu (17/6/2009).

Saat ditanyakan bagaimana perasaannya setelah bercerai dengan suaminya, Chintami mengaku lega. Namun, sesaat kemudian dia menangis dan langsung dipeluk kakaknya Minati

Atmanegara yang berada di sampingnya.

Chintami mengaku meskipun keputusan bercerai ini sangat sulit bagi dirinya, bagaimana pun dia harus tetap menjalaninya. "Alhamdulillah. Kita pisahnya baik-baik. Yang saya harapkan ketemuanya baik-baik, pisah pun baik-baik," katanya.

Ketidakhadiran Bayu Maruto di dalam persidangan, tidak dipermasalahkan Chintami. Pasalnya, Bayu memang sedang memiliki urusan yang tak bisa ditinggalkan di luar kota.

Sebelumnya Chintami menggugat cerai suaminya Bayu Maruto ke Pengadilan Agama Jakarta Selatan bulan Mei lalu. Chintami tidak mau menyebutkan dengan jelas penyebab gugatan cerainya itu. Yang pasti, dia membantah tegas adanya orang ketiga dalam hubungan rumah tangga mereka.

(uky)

K10 Masalah.txt

10 Masalah Seks yang Kerap Dikeluhkan

shutterstock.com

/

Sabtu, 3 Oktober 2009 | 10:51 WIB

KOMPAS.com — Menurut para pakar seks, ada 10 problem utama yang sering melanda pasangan modern.

1. Kecanduan, yakni segala aktivitas seksual yang tak bisa dikontrol. Misalnya, melihat material pornografi, masturbasi, atau mengunjungi lokalisasi.
2. Bosan. Bertambahnya usia perkawinan sering kali membuat kehidupan seks menjadi monoton.
3. Sulit orgasme. Penelitian menunjukkan, sebanyak 35 persen perempuan belum pernah merasakan orgasme, dan 25 persen wanita sulit mencapai orgasme. Selain itu diperkirakan, 70 persen wanita tidak bisa mencapai klimaks bila hanya lewat penetrasi seks saja. Untuk mencapai orgasme, mereka memerlukan bantuan alat bantu seks.
4. Ejakulasi dini. Masalah seksual ini dikeluhkan oleh lebih dari 40 persen pria.
5. Gairah tak sama. Secara umum pria memang punya dorongan seks lebih kuat karena mereka memiliki hormon testosteron 20-40 kali lebih banyak dibanding wanita. Pada pria, kadar testosteron mereka akan tetap sama, sedangkan wanita sangat dipengaruhi oleh siklus menstruasi, kehamilan, dan usia.
6. Sulit ereksi. Masalah ini diderita oleh 40 persen pria berusia 40 tahun, dan 70 persen dialami pria berusia 70 tahun. Mayoritas disebabkan oleh stres dan gangguan kesehatan lainnya.
7. Nyeri saat berhubungan. Pada umumnya kondisi ini disebabkan oleh kurangnya rangsangan. Pasca-persalinan, menopause, infeksi, iritasi, dan masalah ginekologi lainnya juga bisa menyebabkan rasa nyeri dan kesakitan saat penetrasi.
8. Tak bergairah. Redupnya gairah bercinta bisa disebabkan oleh banyak faktor. Misalnya stres, kelelahan, sakit, atau masalah emosional dengan pasangan.

9. Kecanduan pornografi di internet. Orang yang kecanduan pornografi di internet biasanya malas melakukan interaksi fisik secara nyata. Akibatnya, pasangannya pun merasa diabaikan.

10. Punya pasangan online. Menjamurnya situs-situs pertemanan membuka kesempatan untuk "selingkuh". Tak sedikit pasangan yang ditinggalkan karena suami atau istrinya justru sibuk bermesraan di internet dengan kekasih di dunia maya.

O Unggulan.txt

Unggulan Belum Terhadap di Tenis Garuda Series

ANTARANEWS.COM

JAKARTA--Pemain unggulan masih belum menemukan lawan yang tangguh di laga BTPN Tegal Open Garuda Indonesia Tennis Series III GOR Wisanggeni Tegal, Rabu (7/10). Hanya unggulan ke-12 Anshari Nursida (DKI Jakarta) yang gagal setelah dikalahkan mantan pemain pelatnas Ayrton Wibowo dengan mudah, 0-6, 0-6.

Unggulan pertama, Christopher Rungkat (DKI) menang atas Ryan Marlo (Jabar) 6-2, 6-3. Peringkat ke-14, Eko Septim (Jateng), mengalahkan Tri Aris Munandar (Jabar) 6-7(4), 6-0, dan 6-0. Demikian pula unggulan delapan Surya Wijaya (Kaltim) menang atas Rizky Bagus Saputra (Jateng) 6-4, 7-6(2).

Pemain lain, unggulan empat Nesa Arta (DKI) menundukkan Budi Darmawan (Jateng) 6-1, 6-1 dan unggulan ke-16 Ariawan Poerbo (Sulut) menang atas Alfonso Bangun (Sumut) 6-0, 6-1. Enrico Satria (Jateng) mengalahkan Kurnia Suranto (Jateng) 6-3, 6-1, dan Andrian Raturandang (Jabar) menang atas Bangun Hartato (Jateng) 6-4, 6-2.

Demikian pula, Prima Simpatiaji (Jateng), Andery Setyawanto (Jatim), David Agung Susanto (Jateng), dan Sunu Wahyu Trijati (Jateng), tak bisa ditembus lawan. Para pemain unggulan baru bertemu Kamis (8/10) ini di babak perdelapan final untuk saling mengalahkan. lnh/eye

O TGM.txt

GM Indonesia Gandeng 100 Vendor

GMAI

GM Asia-Pasifik Nick Reilly (kiri) saat meninjau pabrik perakitan milik GM Indonesia awal tahun ini

Artikel Terkait:

Chevrolet Aveo Prime dari GM Indonesia

Dua Generasi Chevy Camaro Tandai Eksistensi New GM...

GM Indonesia Luncurkan Chevy Captiva FWD 2009

Kamis, 8/10/2009 | 09:48 WIB

JAKARTA, KOMPAS.com - Rencana pengaktifan kembali pabrik PT General Motors Autoworld Indonesia (GMAI) tahun 2012 mendatang akan menjadi berkah tersendiri bagi industri komponen nasional. Pasalnya, agen tunggal pemegang merek (ATPM) mobil Chevrolet (Chevy) ini menargetkan menggandeng 100 vendor komponen lokal.

"Paling aman 2012 pabrik mulai aktif lagi. Sampai sekarang belum ada supplier yang ditunjuk

dan semua masih proses. Mereka mulai banyak apply (mengajukan), kita inginnya kembali menggandeng 100 supplier lokal seperti waktu (Opel) Blazer dulu," ujar Direktur Operasional GM Indonesia Amir Pramadana saat berbincang dengan KOMPAS.com, baru-baru ini.

Saat ini GM Indonesia memiliki fasilitas perakitan di Pondok Ungu, Bekasi, Jawa Barat dengan kapasitas produksi hingga 20.000 unit per tahun. kendaraan yang menjadi prioritas diproduksi yang segmen pasar paling gemuk.

Berdasarkan data Gaikindo, kendaraan kategori multi purpose vehicle (MPV) menduduki peringkat teratas. Nah, GM berencana memproduksi MPV tujuh penumpang (seven-seater), menggunakan mesin berkapasitas 1.500cc. Harganya, diperkirakan Rp150 jutaan per unit.

Selain itu, GMAI juga menargetkan bisa memenuhi 60 persen konten lokal Asean untuk produknya. Memanfaatkan kesepakatan kerjasama Free Trade Asean (FTA) yang sudah bergulir saat ini.

"Saat pabrik mulai beroperasi, saya rasa kita akan bisa menyerap sekitar 400-500 orang pekerja, ditambah 100 (orang) lagi buat di office (manajemen)," lanjut Arif.

Terkait penyerapan tenaga kerja, GM Indonesia mengaku tetap akan mendahului para karyawan sebelumnya yang sempat terkena pemutusan hubungan kerja (PHK). Langkah ini dilakukan agar pabrikan bisa langsung bisa beroperasi dengan waktu yang efektif dan efisien.

#### P Puluhan.txt

SUKABUMI—Sebanyak 25 orang pelajar Sekolah Dasar Negeri (SDN) Cisarua 4, Desa Cisarua, Kecamatan Sukabumi, Kabupaten Sukabumi keracunan makanan bakso cilok (Baslok). Dampaknya para pelajar tersebut selama satu hari tidak bisa sekolah seperti biasanya.

Dari 25 orang siswa, 16 orang diantaranya harus dibawa ke Puskesmas Sukabumi karena kondisinya cukup parah. Sementara sembilan pelajar lainnya hanya mendapatkan perawatan di Unit Kesehatan Sekolah (UKS) SDN 4 Cisarua.

"Para pelajar tersebut kini sudah sehat kembali namun bekas keracunannya masih terlihat di bagian wajah," tandas Kepala Bidang Pemberantasan Penyakit Menular dan Penyehatan Lingkungan (P2MPL), Dinas Kesehatan (Dinkes) Kabupaten Sukabumi, Dadang Sucipta, Rabu (7/10). Dijelaskan, puluhan pelajar tersebut mengalami keracunan setelah memakan cilok yang terbuat dari campuran ikan dan tepung aci.

Mereka kata Dadang beramai-ramai jajan cilok pada saat jam istirahat. Hingga kini terang dia bekas keracunan berupa bintik-bintik merah pada bagian wajah masih belum hilang. Di duga para siswa SD itu alergi dengan makanan cilok yang terbuat dari ikan." Mereka mengeluhkan pusing-pusing setelah memakan cilok. Namun anak-anak itu tidak sampai muntah," imbuh dia.

Dadang menjelaskan Dinkes Kabupaten Sukabumi telah mengambil sampel cilok untuk diteliti lebih lanjut. Sampel itu sambung dia akan diteliti di laboratorium kesehatan daerah, Bandung dan hasilnya akan diperoleh dalam beberapa hari ke depan. "Namun dugaan kuat faktor keracunan berasal dari ikan," jelas Dadang. Pasalnya lanjut dia para pelajar tersebut mengalami gejala alergi pada bagian wajahnya.

Menurut Dadang, para pelajar yang terkena keracunan tersebut sudah mendapatkan penanganan medis. Pada hari Rabu (7/10) ini pun, jelas dia ke 25 pelajar tersebut mulai kembali sekolah seperti biasanya. "Penjual makanan cilok tersebut sudah dimintai keterangan

oleh pihak kepolisian," tandas Dadang. Dia menambahkan pedagang itu nantinya akan mendapatkan pembinaan agar menjual makanan secara bersih dan sehat sehingga aman dimakan oleh anak-anak.

Para pedagang makanan tandas Dadang jangan hanya memperhatikan masalah keuntungan semata. Mereka kata dia harus memperhatikan pula aspek kebersihan dari makanan yang dijualnya. Selain itu Dinkes Kabupaten Sukabumi tandas Dadang menghimbau agar para pelajar lebih berhati-hati dalam membeli makanan yang ada di sekolah. Langkah itu tutur dia diharapkan mampu menghindari pelajar dari makanan yang kurang sehat dan beracun.

POAnggota.txt

Anggota MPR Ucap Janji Bakti

Jum'at, 02 Oktober 2009 13:36

Kapanlagi.com - Anggota Majelis Permusyawaratan Rakyat (MPR) masa bakti 2009-2014 mengucapkan sumpah/janji yang dipandu oleh Ketua Mahkamah Agung Harifin A Tumpa, di Jakarta, Jumat (02/10).

Pengucapan sumpah/janji anggota MPR ini seharusnya dilaksanakan pada Kamis 1 Oktober 2009, namun tertunda hingga 2 Oktober karena molornya pemilihan pimpinan Dewan Perwakilan Daerah (DPD).

[Info untuk Anda: "Semua berita KapanLagi.com bisa dibuka di ponsel. Pastikan layanan GPRS atau 3G Anda sudah aktif, lalu buka mobile internet browser Anda, masukkan alamat: m.kapanlagi.com"]

Upacara pengucapan sumpah/janji anggota MPR terpilih itu dihadiri Wakil Presiden Jusuf Kalla. Sidang paripurna dengan agenda pengucapan sumpah/janji ini dipimpin oleh Ketua Sementara MPR Marzuki Alie.

Sebelum memasuki acara pengucapan sumpah/janji, Marzuki Alie dalam sambutannya menyampaikan belasungkawa atas musibah gempa yang terjadi di Sumatera Barat. Ia berharap agar korban gempa diberikan kekuatan dan ketabahan.

"Saat mengucapkan sumpah janji ini, saudara kita di Sumatera Barat dan Jambi menderita," kata Marzuki yang telah ditetapkan sebagai Ketua DPR.

Sementara itu, dalam pengucapan sumpah/janji, anggota MPR berjanji menjalankan kewajiban sebaik-baiknya dan seadil-adilnya, sesuai peraturan perundang-undangan dengan berpedoman pada Pancasila dan UUD 1945.

Anggota MPR berjanji dalam menjalankan kewajibannya akan bekerja dengan sungguh-sungguh demi tegaknya kehidupan demokrasi serta mengutamakan kepentingan bangsa dan negara daripada kepentingan pribadi, seseorang, dan golongan.

"Akan perjuangkan aspirasi rakyat dan daerah yang saya wakili untuk mewujudkan tujuan nasional demi kepentingan bangsa dan Negara Kesatuan RI," kata anggota MPR menirukan petikan sumpah/janji.

Setelah pengucapan sumpah/janji, anggota MPR akan melaksanakan sidang paripurna untuk memilih pimpinan MPR 2009-2014.

Sebelumnya, pada Kamis (1/10) sebanyak 560 anggota Dewan Perwakilan Rakyat (DPR) dan 128 anggota DPD terpilih periode 2009-2014 mengucapkan sumpah/janji dalam sidang paripurna.

Upacara pengucapan sumpah/janji anggota DPR dan DPD itu juga dihadiri oleh Wakil Presiden Jusuf Kalla. (kpl/bee)

T2010.txt

2010 Sultra Mulai Nikmati Jaringan 3G

Selasa, 16 Juni 2009 23:13

Kapanlagi.com - PT. Telkomsel menargetkan pembangunan jaringan 3G (generasi ketiga) dan 3.5 G di Sulawesi Tenggara (Sultra) yang akan selesai pada tahun 2010.

Branch Manager Telkomsel Kendari, Asbullah di Kendari, Senin, mengatakan, PT. Telkomsel sedang menyiapkan transmisi untuk layanan broadband atau servis internet yang memiliki kecepatan transfer yang tinggi karena memiliki jalur data yang lebih lebar.

"Kami sedang menyiapkan infrastruktur untuk layanan broadband, khususnya 3G dan 3.5 G ini. Oleh karena itu, kalau infrastruktur ini terwujud pada tahun 2010 tidak perlu menggunakan GPRS karena ada broadband untuk internet," katanya.

Menurut Asbullah, masyarakat Sultra sudah sewajarnya menggunakan layanan broadband, sebab sudah banyak pengguna smart phone atau handphone yang menyediakan layanan 3G dan 3.5G yang bisa dimanfaatkan untuk video call maupun transfer data dengan kecepatan hingga 3,2 mbps (megabyte per second) .

"Di Sultra sudah banyak yang menggunakan smart phone, tapi sayangnya mereka belum bisa mengakses layanan broadband ini. Oleh karena itu, kami akan upayakan pembangunan infrastruktur itu secepatnya," jelasnya.

Selain dari banyaknya pengguna smart phone, juga indikator kebutuhan masyarakat atas layanan teknologi transfer data melalui layanan internet di Sultra terlihat semakin menjamurnya jumlah warnet di berbagai kota utama seperti Kendari, Kolaka dan Bau-bau.

"Kalau kita lihat warnet sekarang menjamur di Sultra, jika layanan broadband sudah ada, masyarakat tidak perlu ke warnet lagi, mereka bisa menggunakan handphone saja," ujarnya.

Menurut dia, setelah layanan itu bisa diakses, PT. Telkomsel akan meluncurkan iPhone 3G yang merupakan kombinasi tiga produk dalam satu ponsel revolusioner produksi apple dengan layar lebar dan menyediakan sekitar 160 fitur. (kpl/roc)