CLUSTERING DOKUMEN BERITA BERBAHASA INDONESIA MENGGUNAKAN *DBSCAN*

SKRIPSI

oleh:
ELICIA JUNIAR
0910962002-96

PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2012

UNIVERSITAS BRAWIJAYA

UNIVERSITAS BRAWIJAYA

UNIVERSITAS BRAWIJAYA

CLUSTERING DOKUMEN BERITA BERBAHASA INDONESIA MENGGUNAKAN *DBSCAN*

Skripsi

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

oleh:
ELICIA JUNIAR
0910962002-96

PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2012

LEMBAR PENGESAHAN SKRIPSI

CLUSTERING DOKUMEN BERITA BERBAHASA INDONESIA MENGGUNAKAN _DBSCAN_

oleh:
ELICIA JUNIAR
0910962002-96

Setelah dipertahankan di depan Majelis Penguji
pada tanggal 9 Januari 2012
dan dinyatakan memenuhi syarat untuk memperoleh gelar
Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I,	Pembimbing II,
Dian Eka Ratnawati, S.si., Mkom	**Drs. Achmad Ridok, M.Kom**
NIP. 197306192002122001	**196808251994031002**

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, MSc
NIP 19670907 199203 1 001


UNIVERSITAS BRAWIJAYA

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama	**: Elicia Juniar**
NIM	**: 0910962002-96**
Jurusan	**: Matematika**
Program Studi	**: Ilmu Komputer**
Penulis tugas akhir berjudul	**: *Clustering* Dokumen Berita Berbahasa Indonesia Menggunakan *DBSCAN***

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, 9 Januari 2012
Yang menyatakan,

Elicia Juniar
NIM. 0910962002


UNIVERSITAS BRAWIJAYA

CLUSTERING DOKUMEN BERITA BERBAHASA INDONESIA MENGGUNAKAN *DBSCAN*

ABSTRAK

Ilmu pengetahuan dan teknologi berkembang sangat pesat, hal ini ditandai dengan banyaknya volume dokumen elektronik, terutama dokumen berita berbahasa Indonesia. Besarnya volume ini tentunya akan menimbulkan masalah dalam hal pencarian. Salah satu cara untuk mempermudah pencarian yaitu dengan *clustering* dan salah satu metode *clustering* yang dikembangkan adalah *DBSCAN (Density-Based Spatial Clustering of Application with Noise)*. *DBSCAN* membentuk *cluster* dengan ukuran dan *density* yang minimal. *Density* didefinisikan sebagai minimal banyaknya *item* dalam suatu jarak tertentu dari *item* lainnya. Dengan algoritma ini, *outlier* akan tereliminasi karena tidak memiliki *density* yang cukup untuk membentuk *cluster*.

Pada penelitian ini, akan dibuat aplikasi menggunakan *DBSCAN* untuk *clustering* dokumen berita berbahasa Indonesia. Jumlah *cluster* yang dihasilkan ditentukan oleh sistem. Pada proses *clustering*, akan digunakan parameter minimal *points (minPts)* dan parameter *epsilon (eps)*. Hasil yang didapatkan dari penelitian adalah nilai *F-measure* sebesar 0,7099 atau 70,99% pada nilai parameter *eps* = 0,04 dan *minPts* = 30.

INDONESIAN NEWS CLUSTERING USING DBSCAN

ABSTRACT

Science and technology are developing rapidly, it shown by great numbers of electronic documents, especially Indonesian news. These increasing will certainly cause problems especially in data search. One of the ways to facilitate data search is with clustering and one of the method called DBSCAN (Density-Based Spatial Clustering of Application with Noise). DBSCAN is able to form cluster with minimal size and density. Density is defined as the minimal of items to others in a certain distance. The outlier can be eliminated with this algorithm because it has low density to form cluster.

In this research, an using DBSCAN for Indonesian news clustering were conducted. The amount of cluster yielded is determined by the system. The parameters used in the clustering process are minimal points (minPts) and epsilon (eps) parameters. The trial result of its process is F-measured value of 0,7099 or 70,99%, the epsilon parameter (eps) value of 0,04 and the minimal points parameter (minPts) of 30.

KATA PENGANTAR

Alhamdulillah, segala puji syukur bagi Allah ﷻ, Tuhan Semesta Alam yang telah melimpahkan segala rahmat dan kasih-Nya kepada seluruh makhluk-Nya di bumi. Selawat dan salam terhatur kepada nabi kita Rasulullah Muhammad ﷺ, semoga kita selalu istiqamah dalam meneladaninya. Karena hanya dengan pertolongan Allah ﷻ semata, yang berupa nikmat kesempatan, kemudahan, kesehatan dan rizki, penulis akhirnya dapat menyelesaikan skripsi yang berjudul "***CLUSTERING* DOKUMEN BERITA BERBAHASA INDONESIA MENGGUNAKAN *DBSCAN*".**

Tidak dapat dipungkiri pula bahwa tidak mungkin penulis dapat menyelesaikan skripsi ini tanpa bantuan dan dukungan dari banyak pihak. Untuk itu, dengan ketulusan dan rendah hati penulis menyampaikan ucapan terima kasih kepada:

1. Ibu Dian Eka Ratnawati, S.Si, M.Kom dan Bapak Drs. Achmad Ridok, M.Kom selaku pembimbing yang telah meluangkan waktu untuk memberikan pengarahan dan masukannya kepada penulis, sejak penyusunan usulan penelitian sampai dengan selesainya laporan skripsi ini.
2. Bapak Dr. Abdul Rouf Alghofari, M.Sc., selaku Ketua Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.
3. Bapak Marji, M.T., selaku ketua program studi Ilmu Komputer, jurusan Matematika, FMIPA Universitas Brawijaya.
4. Bapak Drs. Achmad Ridok, M.Kom, selaku dosen pembimbing akademis serta segenap Bapak dan Ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.
5. Segenap staf dan karyawan Jurusan Matematika Universitas Brawijaya yang telah membantu penyusunan skripsi ini.
6. Ayah, Ibu, dan saudara-saudara, serta keluarga, terima kasih atas curahan doa, kasih sayang yang tulus serta dukungan yang telah diberikan.
7. Ario Letto Hidayatulloh yang telah memberi banyak perhatian dan dukungan tiada henti, terima kasih atas segala masukan, kritikan, dan bantuan yang diberikan kepada penulis selama ini.

8. Keluarga besar Ieka Mira Gajayana 107A, tempat penulis bernaung selama menjalani perantauan untuk menuntut ilmu, terima kasih telah menjadi keluarga kedua untuk penulis.
9. Teman-teman SAP Ilmu Komputer angkatan 2008, 2009, dan 2010, teman-teman Prodi Ilmu Komputer serta teman-teman lain yang selalu memberi dukungan dan doanya.
10. Semua pihak yang telah memberikan bantuan dan dukungannya sehingga skripsi ini selesai.

Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca dan bisa diambil manfaatnya, baik oleh penulis selaku mahasiswa maupun pihak-pihak lain yang tertarik untuk menekuni pengembangan aplikasi *clustering* dokumen berita berbahasa Indonesia khususnya menggunakan metode *density-based spatial clustering of application with noise (DBSCAN).*

Penulis menyadari bahwa masih banyak kekurangan dalam laporan ini disebabkan oleh keterbatasan kemampuan dan pengalaman. Oleh karena itu penulis sangat menghargai saran dan kritik yang sifatnya membangun demi perbaikan penulisan dan mutu isi skripsi ini untuk pengembangan selanjutnya.

Malang, Januari 2012

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR *SOURCECODE*

DAFTAR TABEL

UNIVERSITAS BRAWIJAYA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Tingginya penggunaan internet telah memacu pesatnya pertumbuhan dan pertukaran informasi. Tidak hanya dalam dunia maya, tetapi jumlah informasi dalam bentuk teks juga semakin banyak digunakan di berbagai institusi dan perusahaan. Jumlah dokumen elektronik berbahasa Indonesia yang semakin besar merupakan sumber informasi yang berharga.

Salah satu cara untuk mempermudah dalam pencarian informasi pada dokumen elektronik yang begitu banyak adalah dengan menggunakan *text mining. Text mining* adalah penemuan baru dimana informasi yang sebelumnya tidak diketahui secara otomatis akan mengekstrak informasi dari sumber-sumber tertulis yang berbeda. Salah satu bagian dari *text mining* adalah pengelompokan teks (*text clustering*). Pada dasarnya, sebuah berita merupakan sebuah teks, jadi pengertian pengelompokan berita juga dapat disamaartikan dengan pengelompokan dokumen (teks). *Clustering* merupakan salah satu metode yang dapat digunakan untuk menemukan keterkaitan antar dokumen. Tujuan *clustering* adalah untuk memisahkan kumpulan dokumen ke dalam beberapa *group* atau *cluster* dengan memiliki kemiripan antar dokumen dari segi konten (Ernawati dkk, 2009).

Proses *clustering* bertujuan untuk membantu pengguna agar memiliki akses yang lebih baik untuk dokumen sehingga memberikan kepuasan dalam kebutuhan akan informasi. Kebutuhan akan informasi tersebut dapat didefinisikan sebagai proses untuk menemukan atau memperoleh informasi baru yang berasal dari dokumen, untuk memisahkan informasi yang diinginkan dari sebuah dokumen (Shaban, 2009).

Beberapa algoritma *clustering* yang telah dikembangkan diantaranya adalah *k-means, k-median,* DBSCAN, CLARANS dan DENCLUE (Grira dkk, 2005). Tentunya setiap algoritma tersebut memiliki kelebihan dan kekurangan masing-masing. Banyaknya algoritma *clustering* menyulitkan dalam menentukan ukuran kualitas yang universal. Beberapa hal yang perlu diperhatikan adalah parameter *input* yang tidak menyulitkan *user*, *cluster* hasil yang

dapat dianalisis, dan skalabilitas terhadap penambahan ukuran *dataset*.

DBSCAN (Density-Based Spatial Clustering of Application with Noise) merupakan algoritma untuk membentuk *cluster* dengan ukuran dan *density* yang minimal. *Density* didefinisikan sebagai minimal banyaknya *item* dalam suatu jarak tertentu dari *item* lainnya. Dengan algoritma ini, *outlier* akan tereliminasi karena tidak memiliki *density* yang cukup untuk membentuk *cluster*. *Cluster* dengan bentuk yang tidak bulat juga dapat ditemukan dengan algoritma ini. *DBSCAN* menentukan sendiri jumlah *cluster* yang akan dihasilkan dan memerlukan dua inputan lain, yaitu minimal *points* (*minPts)* dan *epsilon (eps)*. Parameter *minPts* membatasi minimal banyak *item* dalam suatu *cluster* sedangkan parameter *eps* digunakan sebagai nilai *treshold* untuk jarak antar-*item* yang menjadi dasar pembentukan *neighbourhood* dari suatu titik *item*.

Adapun penelitian yang sudah dilakukan adalah pengelompokan terhadap bintik matahari menggunakan algoritma *DBSCAN*. Secara teoritis algoritma ini cukup tepat digunakan, karena secara manual bintik matahari dikelompokkan berdasarkan kedekatan jaraknya dengan bintik matahari lainnya (Satia dkk, 2011). Penelitian lain, yaitu *clustering* bunga iris. Tingkat akurasi dari algoritma ini adalah 78% (Dehuri dkk, 2006).

Berdasarkan latar belakang yang telah dipaparkan tersebut maka memungkinkan jika algoritma *DBSCAN* dicoba untuk diterapkan pada *text mining* khususnya *clustering* dokumen. Sehingga judul yang diambil dalam skripsi ini adalah **"*CLUSTERING DOKUMEN BERITA BERBAHASA INDONESIA MENGGUNAKAN DBSCAN*"**

1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan, rumusan masalah dalam skripsi ini adalah :

1. Bagaimana perancangan dan implementasi untuk *clustering* dokumen berita berbahasa Indonesia dengan metode *DBSCAN*
2. Bagaimana tingkat akurasi yang dihasilkan dari *clustering* dokumen berita berbahasa Indonesia menggunakan *DBSCAN*

1.3 Batasan Masalah

Dari permasalahan tersebut, berikut ini diberikan batasan-batasan masalah untuk menghindari melebarnya masalah yang akan diselesaikan:

1. Hanya menangani dokumen berita berbahasa Indonesia.
2. Dokumen berita berbahasa Indonesia yang akan digunakan ini berupa dokumen dalam format **.txt*.
3. Dokumen yang akan diolah diambil dari situs berita *online*, yaitu www.VIVAnews.com.
4. Metode similaritas yang digunakan adalah *cosine similarity*.
5. Jumlah dokumen yang digunakan pada tiap kategori jumlahnya harus sama.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam skripsi ini adalah :

1. Merancang dan mengimplementasikan metode *DBSCAN* untuk *clustering* dokumen berita berbahasa Indonesia.
2. Mengetahui tingkat akurasi dari penerapan *DBSCAN*.

1.5 Manfaat Penelitian

Manfaat yang ingin dicapai dalam penulisan skripsi ini adalah menyediakan aplikasi yang menerapkan algoritma *DBSCAN* yang mampu menyelesaikan masalah *clustering* dokumen berita berbahasa Indonesia, sehingga dapat mempermudah pencarian dokumen berita yang saling berkaitan.

1.6 Sistematika Penulisan

Dalam penulisan skripsi ini, sistematika penulisan dibagi menjadi lima bab, yaitu :

1. **BAB I PENDAHULUAN**

   Berisi penjelasan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika penulisan yang digunakan untuk menyusun laporan.

2. **BAB II TINJAUAN PUSTAKA**

   Berisi teori- teori yang digunakan sebagai dasar penelitian dan penulisan penelitian ini serta referensi-referensi lain yang mendukung penelitian ini baik dari buku maupun dari internet.

3. **BAB III METODE PENELITIAN**

   Menjabarkan metode-metode yang digunakan dalam penelitian dan berisi perancangan model sistem yang akan diimplementasikan.

4. **BAB IV HASIL DAN PEMBAHASAN**

   Berisi implementasi metode, rancangan dan analisa sistem yang dikembangkan, yaitu apakah hasil pencarian oleh sistem memberi hasil yang dapat membantu menyelesaikan permasalah yang dihadapi.

5. **BAB V PENUTUP**

   Berisi kesimpulan dari penelitian yang telah dilakukan dan saran-saran mengenai hasil penelitian yang perlu dikembangkan.

BAB II
TINJAUAN PUSTAKA

2.1 Berita

Menurut kamus besar bahasa Indonesia tahun 2008, berita adalah cerita atau keterangan mengenai kejadian atau peristiwa yang hangat. Sedangkan menurut Wahyudi (2002), berita adalah laporan tentang peristiwa atau pendapat yang memiliki nilai penting, menarik bagi sebagian khalayak, masih baru dan dipublikasikan melalui media massa periodik.

Menurut Budiman (2005) berita terdiri dari :

1. *Headline*
   Disebut dengan judul. Sering dilengkapi dengan anak judul. *Headline* berguna untuk: (1) menolong pembaca agar segera mengetahui peristiwa yang akan diberitakan; (2) menonjolkan satu berita dengan dukungan teknik grafika.
2. *Dateline*
   Ada yang terdiri atas nama media massa, tempat kejadian dan tangal kejadian. Tujuannya adalah untuk menunjukkan tempat kejadian dan inisial media.
3. *Lead*
   Lazim disebut dengan teras berita. Biasanya ditulis pada paragraf pertama sebuah berita. Ia merupakan unsur yang paling penting dari sebuah berita, yang menentukan apakah isi berita akan dibaca atau tidak. Ia merupakan sari pati sebuah berita, yang melukiskan seluruh berita secara singkat.
4. *Body*
   Atau tubuh berita. Isinya menceritakan peristiwa yang dilaporkan dengan bahasa yang singkat, padat, dan jelas. Dengan demikian, *body* merupakan perkembangan berita.

Di dalam suatu berita juga terdapat unsur-unsur berita. Unsur-unsur tersebut sering dikenal dengan 5W 1H (Budiman, 2005). Yang dimaksud dengan 5W 1H yaitu:

a. *What* – apa yang terjadi di dalam suatu peristiwa?
b. *Who* – siapa yang terlibat di dalamnya?
c. *Where* – di mana terjadinya peristiwa itu?

d. *When* – kapan terjadinya?
e. *Why* – mengapa peristiwa itu terjadi?
f. *How* – bagaimana terjadinya?

2.2 *Text Mining*

Text mining memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen (Mooney, 2006).

Permasalahan yang dihadapi pada *text mining* sama dengan permasalahan data mining yaitu jumlah data yang besar, dimensi yang tinggi, data dan struktur yang terus berubah, serta adanya data *noise*. Sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Pola pada *text mining* diperoleh dari bahasa alami. Perbedaan lainnya yaitu, database dirancang untuk diproses secara otomatis oleh program, sedangkan teks ditulis untuk dibaca oleh manusia. Hal ini menyebabkan adanya permasalahan baru yaitu struktur teks yang kompleks dan tidak lengkap, arti yang tidak jelas dan tidak standar, bahasa yang berbeda, serta translasi yang tidak akurat (Triawati, 2009).

Adapun tugas khusus dari *text mining* antara lain pengkategorian teks *(text categorization)* dan pengelompokan teks *(text clustering). Text categorization* merupakan bentuk dari *supervised learning*. Pada teknik ini diberikan sekumpulan data latih *(training set)* yang masing-masing memiliki atribut dan label kelas. Kemudian dibentuk model untuk masing-masing kelas sesuai dengan fungsi atribut yang dimilikinya. Selanjutnya diberikan sekumpulan data uji *(testing set)* yang digunakan untuk menentukan akurasi dari model tersebut. Tujuan dari teknik ini adalah untuk mengklasifikasikan data-data yang baru ke dalam kategori-kategori yang diberikan menggunakan model yang dihasilkan sebelumnya sebagai dasar pengetahuan (Even dan Zohar, 2002). Sedangkan *text custering* adalah bentuk dari *unsupervised learning*. Pada teknik ini dokumen-dokumen langsung dikelompok-kelompokkan berdasarkan tingkat kemiripan antara dokumen satu dengan yang lain tanpa dilakukan supervisi.

Tujuannya adalah untuk menghasilkan *cluster-cluster* teks yang tepat.

Umumnya, dalam melakukan implementasi *text mining* terdiri dari dua tahap, *text preprocessing* dan *process*. Tahap *text preprocessing* adalah tahap dimana aplikasi melakukan seleksi data yang akan diproses pada setiap dokumen, ada beberapa hal yang perlu dilakukan pada tahap ini, yaitu *case folding*, *tokenizing*, penghilangan *stopwords*, dan *stemming*. Tujuan dilakukan *text preprocessing* adalah memilih setiap kata dari dokumen dan merubahnya menjadi kata dasar yang memiliki arti lebih sempit. Tahap yang kedua adalah *process*. Tahap ini merupakan tahap inti dimana setiap kata akan diolah dengan algoritma tertentu.

2.2.1 *Preprocessing Text*

Text preprocessing bertujuan mempersiapkan teks menjadi data yang akan mengalami pengolahan pada tahapan berikutnya. Pada skripsi ini, hal- hal yang dilakukan pada tahap preposes meliputi *case folding*, *tokenizing*, penghapusan *stopwords*, *stemming*, dan perhitungan bobot.

2.2.1.1 *Case Folding*

Case folding adalah tahap dimana semua huruf dalam dokumen diubah menjadi huruf kecil untuk memudahkan proses-proses selanjutnya. Hanya huruf ‘a’ sampai ‘z’ yang diterima, sedangkan karakter selain huruf, seperti tanda baca dan angka dihilangkan dan dianggap sebagai *delimiter*.

2.2.1.2 *Tokenizing*

Tokenizing adalah proses untuk mengambil kata dan istilah sederhana dari sebuah dokumen. Kata dan istilah sederhana itu berupa potongan-potongan kata tunggal yang menyusun suatu dokumen. Pada tahap ini, dilakukan pemotongan *(parsing)* terhadap kata-kata tunggal tersebut menjadi kumpulan *token.*

2.2.1.3 Penghapusan *Stopwords*

Stopwords adalah kata-kata yang sering muncul atau memiliki frekuensi yang tinggi dalam sebuah teks dokumen. *Stopwords* tidak bermanfaat dalam proses perolehan informasi karena seringnya

muncul dalam sebuah koleksi dokumen, *stopwords* tidak dapat dijadikan kata-kata pembeda antar dokumen. Oleh karena itu *stopwords* tidak ada gunanya jika dimasukkan ke dalam indeks dari koleksi dokumen. Kata-kata yang merupakan *stopwords* adalah kata-kata yang tergolong kata sambung, preposisi, dan konjungsi *(*Ricardo dan Berthier, 1999*)*. Dengan menghilangkan *stopwords* ada beberapa keuntungan yang bisa diperoleh, yaitu mengurangi kata yang harus diindeks, mengurangi ukuran indeks, dan memperbesar perbedaan antara dokumen-dokumen.

Stoplist berisi sekumpulan kata tidak relevan, namun sering kali muncul dalam sebuah dokumen. Dengan kata lain *stoplist* berisi sekumpulan *stopwords* (Jiawei, 2001).

2.2.1.4 *Stemming* pada Bahasa Indonesia

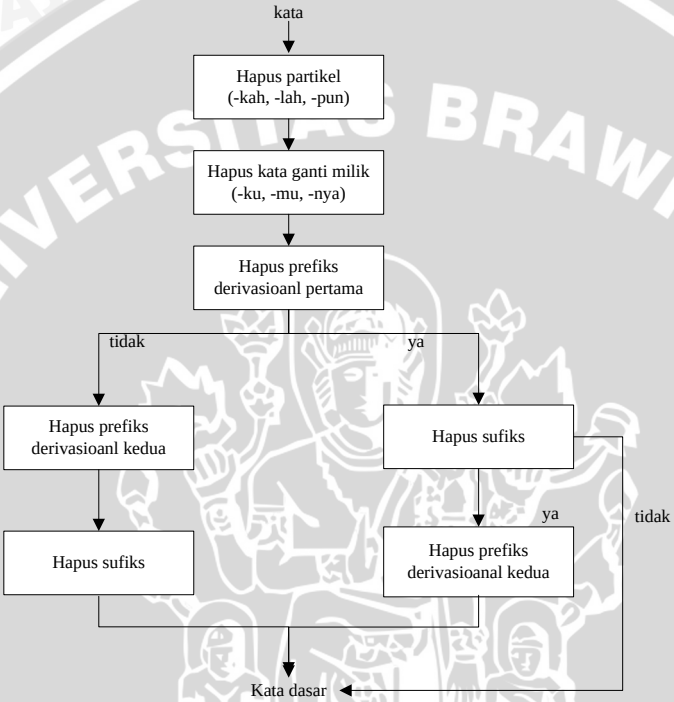
*Stemming merupakan sebuah teknik yang bertujuan untuk mereduksi bentuk dari sebuah kata berimbuhan menjadi bentuk gramatikalnya atau dasarnya (*Ricardo dan Berthier, 1999). *Stemming* yang digunakan adalah *Porter Stemmer* yang disesuaikan dengan aturan bahasa Indonesia. Sebagai contoh untuk *stemming* dalam bahasa Indonesia pemotongan imbuhan dapat dilakukan pada awalan, akhiran, ataupun sisipan, seperti kata "berbelanja" memiliki bentuk dasar "belanja", kata "menjalankan" memiliki bentuk dasar "jalan", dan yang lainnya.

Dengan melakukan *stemming*, ukuran indeks dapat diperkecil karena beberapa kata berimbuhan yang memiliki kata dasar yang sama akan diindeks menjadi satu.

Algoritma *stemming* yang sudah luas diterapkan adalah algoritma yang dibangun oleh Porter yang biasa disebut dengan *Porter Stemmer.* Algoritma ini telah dimodifikasi ke dalam berbagai bahasa. Modifikasi untuk bahasa Indonesia dilakukan oleh Fadillah Z. Tala pada tahun 2003.

Menurut Tala (2003), *stemming* adalah suatu proses yang menyediakan suatu pemetaan antara berbagai kata dengan morfologi yang berbeda menjadi satu bentuk dasar (*stem*). Hal ini bisa dilakukan dengan cara menghilangkan akhiran atau awalan dari sebuah kata. Karena *stemming* menghilangkan imbuhan dari sebuah kata dan tiap bahasa memiliki cara tersendiri dalam menambahkan imbuhan di dalamnya maka algoritma *stemming* pun yang dipakai harus sesuai dengan bahasa dari artikel atau dokumen yang akan

diproses. Dengan proses *stemming*, jumlah ragam kata yang ada di dalam artikel ataupun dokumen dapat berkurang dan dapat mengoptimalkan proses *text mining* dan dapat mempengaruhi hasil *text mining* menjadi lebih memuaskan.



Gambar 2.1 *Porter Stemmer* untuk Bahasa Indonesia
(Sumber : Tala, Fadillah Z., 2003)

2.2.1.4.1 Struktur Morfologi Kata Bahasa Indonesia

Morfologi adalah bagian dari ilmu bahasa yang membicarakan atau yang mempelajari seluk beluk kata serta pengaruh perubahanperubahan bentuk kata terhadap golongan dan arti kata, atau dengan kata lain dapat dikatakan bahwa morfologi mempelajari seluk-beluk bentuk kata serta fungsi perubahan-perubahan bentuk kata itu (Ramlan, 1995).

Morfologi kata bahasa Indonesia dibagi menjadi dua struktur, yaitu *infleksional* dan *derivasional*. *Infleksional* adalah struktur paling sederhana yang dinyatakan dalam penambahan sufiks, yang tidak menyebabkan perubahan arti pada kata dasar yang dilekatinya (Tala, 2003). Sufiks *infleksional* dibagi menjadi dua, yaitu :

1. Sufiks *-lah, -kah, -pun, -tah*. Sufiks ini sebenarnya merupakan partikel yang tidak mempunyai arti. Keberadaannya pada suatu kata adalah sebagai penegas. Contoh :

   dia + kah → diakah
   duduk + lah → duduklah
2. Sufiks *-ku, -mu, -nya*. Sufiks ini dilekatkan pada kata sebagai kata ganti kepunyaan. Contoh :

   tas + ku → tasku
   buku + mu → bukumu

Masing-masing sufiks di atas dapat melekat pada suatu kata dasar secara bersama-sama. Ketika kedua sufiks tersebut muncul bersamaan, maka sufiks jenis kedua selalu diletakkan sebelum sufiks jenis pertama. Struktur morfologi pada kata *infleksional* adalah sebagai berikut:

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

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

Struktur *derivasional* dalam bahasa Indonesia terdiri dari prefiks, sufiks dan kombinasi dari keduanya. Prefiks yang sering dipakai adalah : *ber-*, *di-*, *ke-*, *meng-*, *peng-*, *per-*, *ter-*. Berikut contoh dari masing- masing prefiks :

ber + lari → berlari
di + makan → dimakan
ke + kasih → kekasih
meng + ambil → mengambil
peng + atur → pengatur
per + lebar → perlebar
ter + baca → terbaca

Beberapa prefiks seperti *ber-*, *meng-*, *peng-*, *per-*, *ter-* mungkin akan muncul dalam bentuk yang berbeda. Bentuk dari setiap prefiks ini bergantung pada karakter pertama dari kata dasar yang dilekatinya. Tidak seperti struktur *infleksional*, pengejaan kata pada struktur *derivasional* mungkin berubah setelah penambahan prefiks. Contohnya adalah kata menyapu yang terdiri dari prefiks meng- berubah menjadi meny- dan kata dasar sapu. Prefiks meng- berubah menjadi meny- dan karakter pertama dari kata dasar mengalami peleburan.

Sufiks derivasional adalah *-i, -kan, -an* (Tala, 2003). Contoh penggunaan sufiks *derivasional* adalah :

gula + i → gulai
makan + an → makanan
sampai + kan → sampaikan

Berbeda dengan penambahan prefiks, penambahan sufiks tidak akan merubah ejaan atau bentuk dasar dari suatu kata (Tala, 2003).

Seperti yang telah disebutkan sebelumnya, struktur derivasional juga terdiri dari konfiks yang merupakan kombinasi atau gabungan dari prefiks dan sufiks yang melekat secara bersama-sama pada suatu kata. Contoh :

per + main + an → permainan
ke + kalah + an → kekalahan
ber + jatuh + an → berjatuhan
meng + ambil + i → mengambili

Tidak semua kombinasi atau penggabungan prefiks dan sufiks dapat dikombinasikan menjadi sebuah konfiks. Ada beberapa kombinasi prefiks dan sufiks yang tidak diperbolehkan. Kombinasi tersebut dapat dilihat pada tabel 2.1.

Tabel 2.1 Pasangan Konfiks yang tidak diperbolehkan (Tala, 2003).

Prefiks	Sufiks
be-	-i
di	an
ke	i | kan
meng	an
peng	i | kan
ter	an

Prefiks/konfiks dapat ditambahkan pada suatu kata yang sebelumnya telah dilekati konfiks/prefiks, sehingga menghasilkan struktur prefiks ganda. Tidak semua prefiks/konfiks dapat ditambahkan pada kata yang telah mendapatkan prefiks/konfiks. Ada beberapa aturan dalam urutan pembentukan prefiks ganda. Aturan-aturan tersebut dapat dilihat pada tabel 2.2.

Tabel 2.2 Urutan Prefiks Ganda (Tala, 2003)

Prefiks 1	Prefiks 2
meng- di- ter- ke-	per- ber-

Struktur morfologi pada kata *derivasional* adalah:

Derivasional = (prefiks + kata dasar) | (kata dasar + sufiks) | (prefiks+ kata dasar + sufiks) | (prefiks1 + prefiks2 + kata dasar) | (prefiks1 + prefiks2 + kata dasar + sufiks).

Struktur lain yang mungkin terjadi dalam morfologi bahasa Indonesia adalah penambahan sufiks *infleksional* pada suatu kata yang telah mendapat prefiks, sufiks, konfiks, maupun prefiks ganda. Hal ini dinamakan multiple sufiks. Sehingga dapat disimpulkan bahwa struktur morfologi kata bahasa Indonesia secara umum adalah sebagai berikut (Tala, 2003):

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

dimana [....] menunjukkan pilihan.

2.2.1.4.2 Proses *Stemming* Bahasa Indonesia

Berdasarkan analisa morfologi yang telah dijelaskan pada sub bab 2.2.1.4.1, didapatkan lima kelompok aturan proses *stemming* pada bahasa Indonesia (Tala, 2003). Kelima aturan tersebut adalah menangani partikel infleksional, urutan prefiks derivasional pertama dan kedua, serta sufiks derivasional, sebagaimana akan ditunjukkan Tabel 2.3, 2.4, 2.5, 2.6, dan 2.7 (Tala, 2003).

Tabel 2.3 Aturan I Penanganan terhadap Partikel Infleksional

Sufiks	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
-kah	NULL	2	NULL	diakah→ dia
-lah	NULL	2	NULL	dialah → dia
-tah*	NULL	2	NULL	apatah → apa
-pun**	NULL	2	NULL	buku pun→buku

*tah → improduktif

**pun → menurut EYD terpisah dengan kata yang mengikutinya

Tabel 2.4 Aturan II Penanganan terhadap Kata Ganti Infleksional

Sufiks	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
-ku	NULL	2	NULL	bukuku→buku
-mu	NULL	2	NULL	bukumu→buku
-nya	NULL	2	NULL	bukunya→buku

Tabel 2.5 Aturan III Penanganan terhadap Prefiks Derivasional Pertama

Prefiks	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
meng-	NULL	2	NULL	mengukur→ukur
meny-	s	2	V...*	menyapu→sapu
men-	t	2	V...	menuduh→ tuduh
men-	NULL	2	NULL	menduga→duga
mem-	p	2	V...	memukul→pukul
mem-	NULL	2	NULL	membakar→bakar
me-	NULL	2	NULL	merusak→rusak
peng-	NULL	2	NULL	pengukur→ukur
peny-	s	2	V...	penyelam→selam
pen-	t	2	V...	penari-tari
pen-	NULL	2	NULL	penduga→duga
pem-	p	2	V...	pemandu→pandu
pem-	NULL	2	NULL	pembaca→baca
di-	NULL	2	NULL	diukur→ukur

ter-	NULL	2	NULL	tersipu→sipu
ke-	NULL	2	NULL	kekasih→kasih

* kata dasar dimulai dengan huruf vokal

Tabel 2.6 Aturan IV Penanganan terhadap Prefiks Derivasional Kedua

Prefiks	Pengganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
ber-	NULL	2	NULL	berlari→lari
bel-	NULL	2	ajar	belajar→ajar
be-	NULL	2	B*er	bekerja→kerja
per-	NULL	2	NULL	perjelas→jelas
pel-	NULL	2	ajar	pelajar→ajar
pe-	NULL	2	NULL	pekerja→kerja

* tanda ini menunjukkan bahwa *stem* dimulai dengan konsonan

Tabel 2.7 Aturan V Penanganan terhadap Sufiks Derivasional

Sufiks	Peng ganti	Kondisi Ukuran	Kondisi Tambahan	Contoh
-kan	NULL	2	Prefix bukan anggota {ke,peng}	tarikkan→tarik (meng)ambilkan→ ambil
-an	NULL	2	Prefix bukan anggota {di,meng,ter}	makanan→makan (per)janjian→janji
-i	NULL	2	Prefix bukan anggota {ber, ke, peng}	tandai→tanda (men)dapati→ dapat

Kondisi ukuran adalah jumlah minimum suku kata dalam sebuah kata. Karena dalam bahasa Indonesia, kata dasar setidaknya mempunyai 2 suku kata. Maka kondisi ukuran dalam proses *stemming* bahasa Indonesia adalah dua. Adapun suku kata didefinisikan memiliki satu vokal.

2.2.2 *Term Frequency-Inverse Document Frequency*

Fungsi pembobotan fitur bermula dari bidang *Information Retrieval*. Model paling populer adalah *tf-idf*. Dalam buku "*An Introduction to Information Retrieval*" (Manning, dkk., 2009) diterangkan bahwa skema pembobotan *tf-idf* terdiri dari dua faktor, yaitu bobot lokal dan bobot global. Bobot lokal, disimbolkan *tf*, merupakan bobot *term* t_i dalam sebuah dokumen tertentu d_j, yang diestimasi berdasarkan frekuensi t_i dalam dokumen. Sedangkan bobot global, dinotasikan *idf*, berdasarkan perhitungan jumlah dokumen yang mengandung *term* t_i dalam koleksi.

$$w_{i,j}=tf_{i,j}\times log\ (\frac{N}{dfi}) \quad (2.1)$$

$w_{i,j}$ adalah bobot *term i* dalam dokumen *j*; tf_{ij} adalah frekuensi kemunculan *term i* dalam *dokumen j*; *N* adalah total jumlah dokumen; dan df_i adalah jumlah dokumen yang mengandung *term i*.

2.3 *Cosine Similarity*

Pada aplikasi seperti *information retrieval* dan pengelompokan teks dokumen, data yang dibandingkan berupa vektor objek yang kompleks berisi sejumlah besar entitas (yang mewakili isi teks). Untuk mengukur jarak antara objek yang kompleks, fungsi kesamaan non-metrik lebih umum digunakan dibanding perhitungan jarak metrik tradisional seperti *Euclidian Distance* (Manning, dkk., 2009).

Ada beberapa cara untuk merumuskan fungsi kesamaan dua vektor U dan V. Metode yang paling baik untuk membandingkan kesamaan antara dua vektor dalam hal *text mining* adalah fungsi *Cosine Similarity* (Manning, dkk., 2009).

$$Cosine(\vec{U},\vec{V})=\frac{\vec{U}\cdot\vec{V}}{|\vec{U}|\times|\vec{V}|}=\frac{\sum_{i=1}^{n} u_i\times v_i}{\left(\sqrt{\sum_{i=1}^{n}(u_i)^2}\right)\times\left(\sqrt{\sum_{i=1}^{n}(v_i)^2}\right)} \quad (2.2)$$

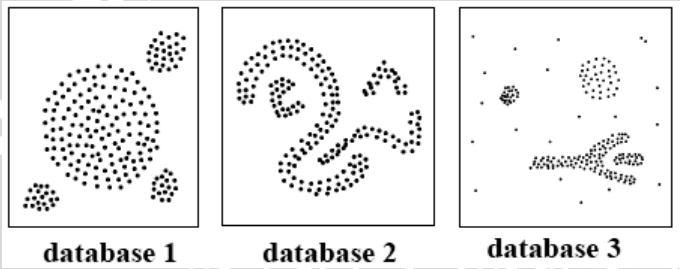
Persamaan 2.2 di atas merupakan fungsi *cosine similarity* antara vektor *U* dan *V*. Penyebutnya merupakan *dot product* (yang juga disebut *inner product*) dari vektor *U* dan *V*. Sedangkan

pembilangnya merupakan perkalian dari *Euclidian length* masing-masing vektor.

Pada persoalan *text mining* nilai bobot *term* tidak boleh negatif sehingga hasil kesamaan *cosine* dua dokumen berkisar antara 0 dan 1. Hal tersebut berarti sudut antara dua vektor dokumen tidak dapat lebih besar dari 90^o. Dengan demikian, semakin besar hasil *cosine* maka semakin besar kesamaan antar dokumen.

2.4 *Density Based Cluster*

Density Based Cluster adalah sekelompok set data yang memiliki bentuk *cluster* seperti pada gambar 2. 2.



Gambar 2.2 *Density Based Cluster* (Ester, 1996)

Alasan utama mengapa *cluster* - *cluster* pada gambar 2.1 dapat dibentuk adalah karena kepadatan *point* - *point* data pada sebuah *cluster* relatif lebih padat bila dibandingkan dengan *point* - *point* data diluar *cluster* (Ester, 1996).

2.5 *DBSCAN (Density Based Spatial Clustering of Application with Noise)*

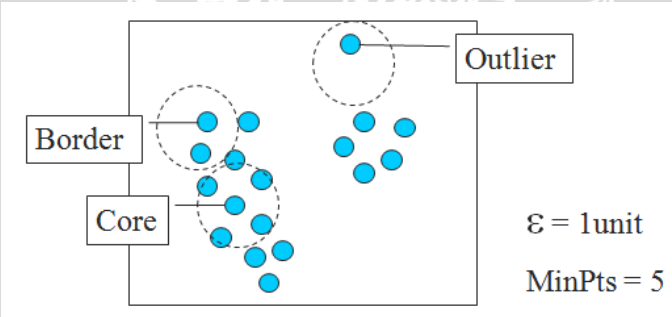
Pada dataset yang sangat besar, tidak mungkin jika data diproses sekaligus dalam memori yang tersedia. Sehingga diperlukan teknik-teknik khusus untuk melakukan *clustering* secara efisien dan juga berkualitas.

Menurut Ester (1996) algoritma *Density-based Spatial Clustering of Application with Noise (DBSCAN)* adalah membentuk *cluster* dengan ukuran dan *density* yang minimal. *Density* didefinisikan sebagai minimal banyak *point* dalam suatu jarak tertentu dari *point* lainnya. *DBSCAN* menentukan sendiri jumlah

cluster yang akan dihasilkan tapi memerlukan dua inputan lain, yaitu:

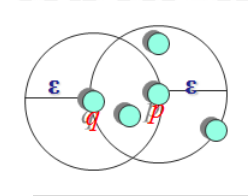
1. *MinPts* : minimal banyak *items* dalam suatu *cluster*
2. *Eps* : nilai *treshold* untuk jarak antar-*items* yang menjadi dasar pembentukan *neighbourhood* dari suatu titik *item.*

Menurut definisi, ada dua jenis titik (*points*) dalam suatu *cluster* yaitu titik di dalam *cluster* (*core points*) dan titik di tepian *cluster* (*border points*). *Point* disebut sebagai *core point* jika memiliki lebih dari jumlah tertentu *point (*dengan batasan nilai dari parameter *minPts)* dalam nilai tertentu dari parameter *eps*. *Core point* merupakan *point* yang berada di interior *cluster* (di dalam *cluster*). *Border point* memliki kurang dari nilai parameter *minPts* dengan *eps* tertentu, tetapi merupakan *neighbourhood* (di sekitar) *core point.* Sedangkan *noise* adalah setiap *point* yang bukan merupakan *core point* atau *border point.* Setiap dua *core point* yang cukup dekat (dengan batasan *eps*, jarak satu sama lain) dimasukkan dalam *cluster* yang sama. Setiap *border point* yang cukup dekat dengan *core point* dimasukkan dalam *cluster* yang sama dengan *core point.* Sedangkan *noise* diabaikan. *Core point* dan *border point* ditunjukkan pada gambar berikut 2.3.



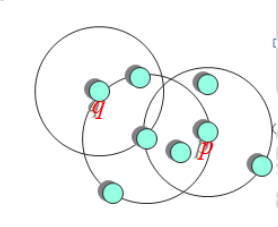
Gambar 2.3 *Border* dan *Core*

Sebuah objek *q* dikatakan *directly density-reachable* dari objek *p* jika *q* dalam *ε-Neighbourhood* dari *p* dan *p* adalah *core point.* Ditunjukkan dalam gambar 2.4.



Gambar 2.4 *Directly Density-Reachable*

Sebuah objek *p* dikatakan *density-reachable* dari objek *q* jika ada objek $p_1,...,p_n$, dimana $p_1 = q$, $p_n = p$ misal p_{i+1} *directly density-reachable dari* p_i dan nilai *minPts untuk semua* $1 <= i <= n$. Ditunjukkan dalam gambar 2.5.



Gambar 2.5 *Density-Reachable*

Pseudocode algoritma *DBSCAN* ditunjukkan dalam gambar 2.6 di bawah ini :

```
DBSCAN (SetOfPoints, Eps, MinPts)

// SetOfPoints is UNCLASSIFIED
   ClusterId := nextId (NOISE);
   FOR i FROM 1 TO  SetOfPoints.size DO
     Point := SetOfPoints.get(i);
     IF Point.ClId = UNCLASSFIED THEN
          IF  ExpandCluster(SetOfPoints,  Point,
    ClusterId, Eps,   MinPts) THEN
             ClusterId := nextId(ClusterId)
          END IF
       END IF
   END FOR
END; // DBSCAN
```

Gambar 2.6 *Pseudocode* Algoritma *DBSCAN*
(Sumber : Ester, 1996)

Sedangkan *pseudocode* untuk *expandCluster* dalam *DBSCAN* ditunjukkan dalam gambar 2.7.

```
(SetOfPoints, Point, ClId, Eps, MinPts) :
Boolean;
  Seeds := SetOfPoints.regionQuery(Points, Eps);
  IF seeds.size < MinPts THEN // no core point
     SetOfPoint.changeClId (Point, NOISE);
     RETURN False;
   ELSE    // all points in seeds are density-
           reachable from Point
     SetOfPoints.changeClIds (seeds, ClId):
     Seeds.delete (Point);
     WHILE seeds <> Empty DO
          currentP := seeds.first();
          result    :=    SetOfPoints.regionQuery
                                    (currentP
                                    , Eps);
          IF result.size >= MinPts THEN
              FOR i FROM 1 TO result.size DO
                 resultP := result.gei(i);
                 IF resultP.ClId
                         IN (UNCLASSIFIED, NOISE)
                    THEN
                    IF       resultP.ClId       =
                    UNCLUSSIFIED THEN
                          Seeds.append(resultP);
                    END IF;

                 SetOfPOints.changeClId(resultP,
                 ClId);
                 END   IF;   //   UNCLASSIFIED   or
            NOISE
              END FOR;
           END IF; // result.size >= MinPts
           Seeds.delete (currentP);
      END WHILE; // seeds <> empty
      RETURN True;
   END IF
END; Expand Cluster
```

Gambar 2.7 *Pseudocode* Algoritma *ExpandCLuster*
(Sumber : Ester, 1996)

2.6 Metode Evaluasi

Pada proses *clustering*, prosedur untuk megevaluasi hasilnya dikenal sebagai validitas *cluster* dan dapat dilakukan dengan berbagai macam pengukuran yang disebut dengan pengukuran validitas *cluster* (Petridou, 2000).

Pengukuran validitas dapat dibagi menjadi dua kategori, dimana pembagian kategori tersebut berdasarkan pada ada tidaknya referensi pengetahuan yang dapat digunakan untuk membandingkan hasil *clustering* yang dilakukan oleh sistem. Kategori yang pertama adalah mengevaluasi seberapa bagus *cluster* dengan cara membandingkan kelompok-kelompok yang dihasilkan oleh sistem dengan data kelompok yang sudah diketahui kelas-kelasnya, dimana kategori ini disebut dengan pengukuran kualitas eksternal. Sedangkan kategori yang kedua adalah membandingkan sejumlah *cluster* tanpa adanya referensi pengetahuan yang disebut dengan pengukuran kualitas internal. Contoh pengukuran kualitas eksternal adalah *entropy* dan *F-Measure* (Steinbach, 2000). Setiap *cluster* yang dihasilkan dianggap sebagai hasil retrieval dan kelompok-kelompok (kelas) dokumen yang telah diidentifikasikan sebelumnya, yang dianggap sebagai *cluster* ideal yang seharusnya dihasilkan dari pengelompokan. Secara lebih spesifik, untuk setiap *cluster* manual *i* dan *cluster* sistem *j* dijelaskan pada persamaan 2.3 dan 2.4.

$$Recall\ (i, j) = R_{ij} = n_{ij}/n_i \quad (2.3)$$

$$Precision(i, j) = P_{ij} = n_{ij}/n_j \quad (2.4)$$

Dimana n_{ij} adalah dokumen cluster manual *i* pada *cluster* sistem *j*, n_i adalah jumlah dokumen pada *cluster* manual *i*, dan n_j adalah jumlah dokumen pada *cluster* sistem *j*. Nilai *F-measure cluster* manual *i* dan *cluster* sistem *j* dinyatakan pada persamaan 2.5.

$$F_{ij} = (2 * R_{ij} * P_{ij})/(R_{ij}+P_{ij}) \quad (2.5)$$

Nilai keseluruhan *F-measure (overall F-measure)* didapatkan melalui persamaan 2.6.

$$F = \sum_i \frac{ni}{n} max\{F_{ij}\} \quad (2.6)$$

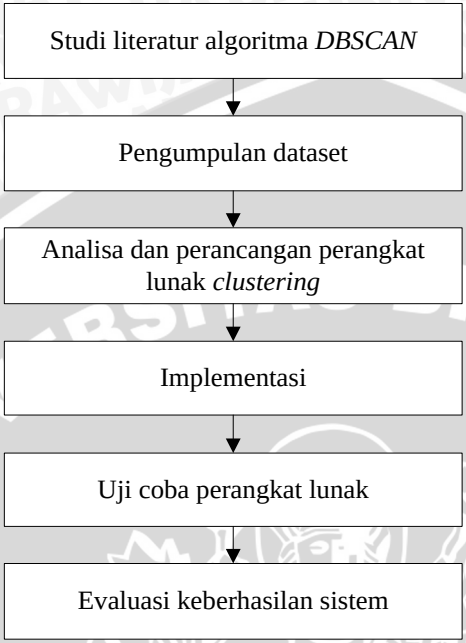
Dimana n adalah jumlah keseluruhan dokumen, n_i adalah jumlah dokumen pada *cluster* manual i, dan $max\{F_{ij}\}$ adalah nilai F_{ij} terbesar yang ditemukan pada *cluster* manual i untuk keseluruhan *cluster* sistem j.

BAB III
METODOLOGI DAN PERANCANGAN

Bab ini berisikan metodologi dari penelitian yang dilakukan dan rancangan sistem yang dibuat sebagai perangkat uji coba dalam penelitian *clustering* dokumen berita berbahasa Indonesia menggunakan algoritma *DBSCAN*. Penelitian dilakukan dengan tahapan-tahapan sebagai berikut :

1. Mencari dan mempelajari literatur-literatur mengenai algoritma *DBSCAN* yang umumnya dipakai untuk masalah *clustering*, dapat juga diaplikasikan untuk masalah *clustering dokumen* yang berasal dari buku dan sumber lain di internet.
2. Mengumpulkan dataset yang dibutuhkan untuk penelitian, yaitu dokumen berita dengan format **.txt*.
3. Menganalisis dan merancang perangkat lunak sistem *clustering* untuk dokumen berita berbahasa Indonesia yang menerapkan algoritma *DBSCAN*.
4. Mengimplementasikan rancangan sistem ke perangkat lunak berdasarkan analisis dan perancangan yang telah dilakukan.
5. Melakukan uji coba perangkat lunak yang telah dibuat menggunakan dokumen teks berita berbahasa Indonesia.
6. Melakukan evaluasi terhadap perangkat lunak yang telah dibuat menggunakan *F-measure*.

Untuk memberi gambaran visual, langkah-langkah yang dilakukan dalam penelitian dapat ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram Alir Pembuatan Perangkat Lunak

3.1 Analisis Data

Dataset yang digunakan dalam penelitian ini adalah dokumen berita berbahasa Indonesia dengan format **.txt*. Dokumen berita ini diperoleh dari situs berita *online*, yaitu www.VIVAnews.com yang telah dikelompokkan secara manual.

Pada tahap pengujian, pengelompokkan yang telah dilakukan secara manual akan dijadikan sebagai pembanding untuk melakukan evaluasi terhadap keberhasilan hasil *clustering* yang telah dilakukan oleh sistem yang telah dibuat.

Dataset dokumen berita berbahasa Indonesia yang digunakan dalam penelitian ini berjumlah 300 dokumen. Diambil dari dokumen berita yang terdiri lima kategori. Pembagian lima kategori ini merupakan pembagian yang dilakukan oleh *VIVAnews* sebagai penerbit sesuai dengan isi berita. Keterangan lebih lengkap mengenai dataset dapat dilihat pada tabel 3.1.

Tabel 3.1 Kategori dan Jumlah Dokumen Berita

No.	Kategori	Jumlah
1	Bisn (Bisnis)	60
2	Bola (Bola)	60
3	Kosm (Kosmo)	60
4	Spot (*Sport*)	60
5	Nasn (Nasional)	60
Total Dokumen Berita		300

Harapannya, setelah dilakukan *clustering* menggunakan sistem yang akan dikembangkan, dataset akan terbagi-bagi ke dalam *cluster* yang sesuai dengan *cluster* yang sebenarnya. Untuk memudahkan evaluasi, maka nama *file* disimpan dengan urutan format dimulai dari kode *cluster* diikuti dengan indeks *file* kemudian format *file*. Contoh salah satu nama *file* Bisn-001.txt, berarti dokumen tersebut termasuk ke dalam *cluster* Bisn (Bisnis) merupakan indeks atau urutan *file* ke-001 dan mempunyai format *.txt*. Perubahan nama *file* ini dilakukan secara manual.

3.2 Analisis Sistem

Pada sub bab ini, akan dibahas mengenai analisis kebutuhan sistem, batasan-batasan yang dimiliki oleh sistem, serta deskripsi umum sistem. Analisis kebutuhan sistem merinci hal-hal pokok yang harus ada dalam sistem. Batasan sistem menjelaskan ketentuan fungsional sistem. Sedangkan deskripsi umum sistem menggambarkan proses umum sistem secara bertahap.

3.2.1 Analisis Kebutuhan Sistem

Sistem dirancang berdasarkan kebutuhan mengenai perangkat otomatis yang membantu meng*cluster* dokumen berita berbahasa Indonesia. Hal-hal pokok yang harus ada dalam sistem ini adalah :

1. Sistem harus mampu meng*cluster* dokumen berita berdasarkan kemiripan (*similarity*) masing-masing dokumen.
2. Menampilkan hasil dokumen berita yang telah di*cluster*.

3.2.2 Batasan Sistem

Sistem perangkat lunak dirancang dengan ketentuan-ketentuan sebagai berikut:

1. Mengabaikan persamaan kata.
2. Dataset yang digunakan dalam bentuk teks murni (*plain text)*.
3. Berkas dataset berada di dalam media penyimpanan lokal (*local disk*) yang terstruktur pada alamat tertentu.

3.2.3 Deskripsi Sistem

Perangkat lunak *clustering* dokumen berita ini bermanfaat untuk meng*cluster* sekumpulan data yang berupa dokumen berita berbahasa Indonesia berdasarkan kategorinya. Sistem akan menerima beberapa input berupa dokumen berita berbahasa Indonesia yang kemudian akan dilakukan *preprocessing* terlebih dahulu, selanjutnya diolah menggunakan algoritma *DBSCAN* sehingga menghasilkan *cluster set*. Sistem ini menggunakan pembobotan kata dengan metode *TF-IDF* dan perhitungan similaritas menggunakan *cosine similarity*. Berikut ini merupakan alur proses yang akan dilaksanakan ketika sejumlah dokumen berita akan diproses.

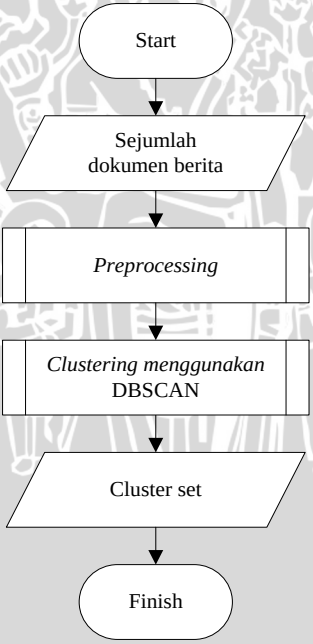
1. Pengguna memilih suatu direktori (*folder*) yang berisi sejumlah dokumen berita yang belum di*cluster* dengan format **.txt*.
2. Masing-masing dokumen berita tersebut akan dibaca dan dilakukan *preprocessing* oleh sistem, yang terdiri dari proses :
   a. *Case folding*
      Proses *case folding* adalah proses dimana semua huruf dalam dokumen diubah menjadi huruf kecil, karakter selain huruf seperti angka dan tanda baca akan dihilangkan dan dianggap sebagai *delimiter*.
   b. *Tokenizing*
      Proses *tokenizing* adalah proses memotong isi dokumen yang berupa kalimat-kalimat menjadi kata tunggal (*token*).
   c. Penghapusan *Stopwords*
      Keseluruhan kata-kata tunggal tersebut akan diperiksa, apakah termasuk kata penting atau tidak penting menggunakan daftar kata pembanding (*stopwords*).
   d. *Stemming*
      Setelah melalui proses penghapusan *stopwords*, maka kata-kata tersebut diubah ke dalam bentuk dasarnya dengan

menggunakan metode *Porter Stemmer* yang telah dimodifikasi dalam bahasa Indonesia oleh Fadillah Z.Talla.

e. Perhitungan Bobot
Sebelum melakukan perhitungan bobot, akan dicari *termlist* dari seluruh dokumen sebagai acuan untuk mengubah dokumen teks menjadi vektor. Keseluruhan *termlist* tersebut akan dihitung frekuensinya. Kemudian ditentukan bobot dari masing-masing kata unik (*term*) pada masing-masing dokumen menggunakan metode pembobotan *TF-IDF*.

3. Setelah menghitung *TF-IDF*, proses selanjutnya adalah *clustering* dokumen.
4. Dari proses *clustering* tersebut, akan diperoleh *cluster set* yang merupakan *cluster* dokumen berita yang telah diproses.

Rancangan alur dari sistem *clustering* (proses umum sistem) yang akan dikembangkan dapat dilihat pada gambar 3.2 berikut:



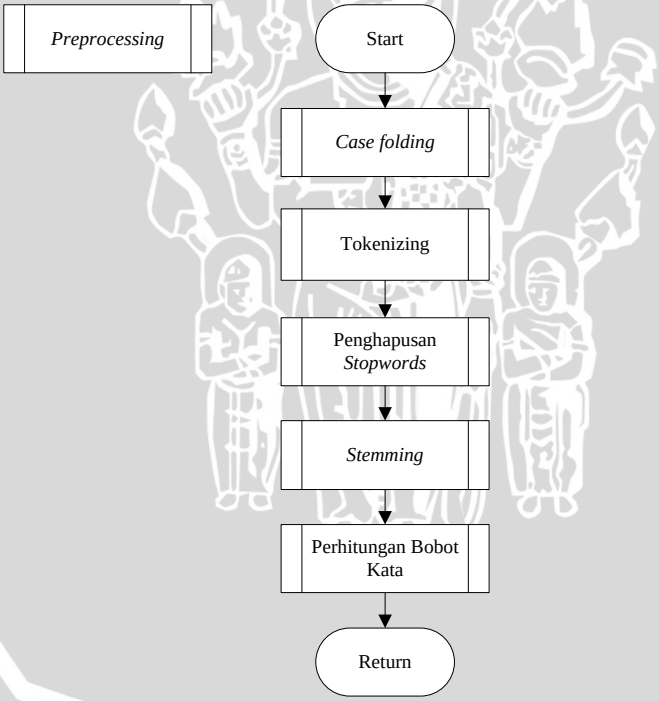
Gambar 3.2 *Flowchart* Proses Umum Sistem

3.3 Perancangan Proses

Berdasarkan analisis yang dilakukan maka dapat dilakukan perancangan proses yang terjadi dalam sistem *clustering* dokumen dan arsitektur yang akan dikembangkan. Dalam poses *clustering* dokumen terdapat dua tahapan yang dilakukan, yaitu *preprocessing* dan proses *clustering* menggunakan algoritma *DBSCAN*.

3.3.1 *Preprocessing*

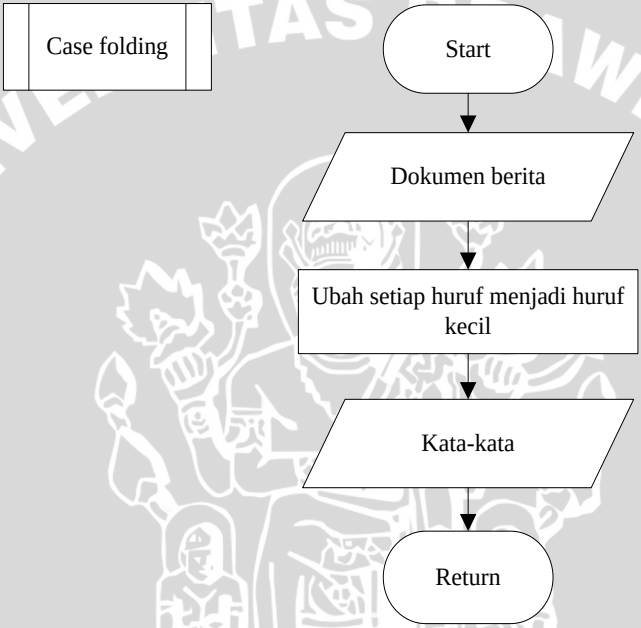
Preprocessing bertujuan untuk mempersiapkan data dokumen agar bisa menjadi data terstruktur yang berupa nilai numerik sehingga dapat dijadikan sumber data yang dapat diolah lebih lanjut. Dalam *preprocessing*, terbagi menjadi sub bagian proses yang terdiri dari *case folding*, *tokenizing*, penghapusan *stopwords*, *stemming*, dan perhitungan bobot kata (*TF-IDF*). *Flowchart* dari *preprocessing* dapat dilihat pada gambar 3.3.



Gambar 3.3 *Flowchart Preprocesing*

3.3.1.1 *Case Folding*

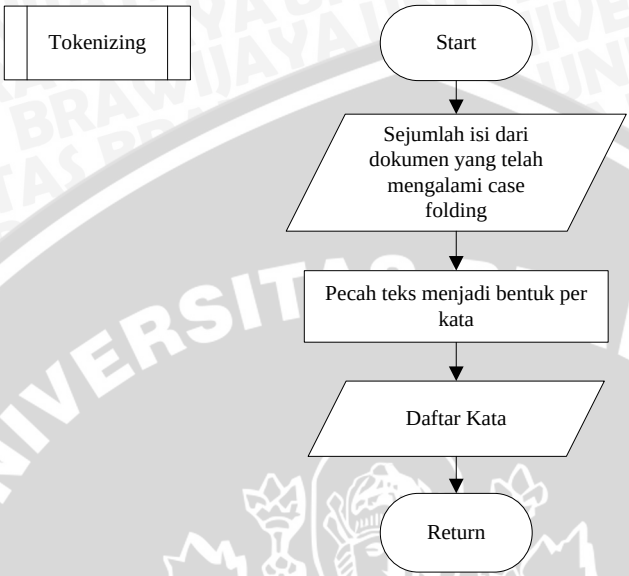
Urutan dari tahap *case folding* ini adalah merubah semua karakter menjadi karakter huruf kecil semua. Karakter angka, tanda baca dan simbol dianggap sebagai pemisah atau *delimiter* (spasi). Hasil dari proses *case folding* adalah dokumen yang hanya berisi karakter huruf kecil dari a-z dan hanya spasi tanda bacanya. *Flowchart* dari proses *case folding* dapat dilihat pada gambar 3.4.



Gambar 3.4 *Flowchart* Proses *Case folding*

3.3.1.2 *Tokenizing*

Setelah melaui proses *case folding*, tahap berikutnya adalah *tokenizing*, yaitu tahap memecah dokumen berita yang telah mengalami *case folding* menurut kata-kata yang menyusunnya dan membentuknya menjadi sebuah daftar atau *list kata*. *Flowchart* dari proses *tokenizing* dapat dilihat pada gambar 3.5.

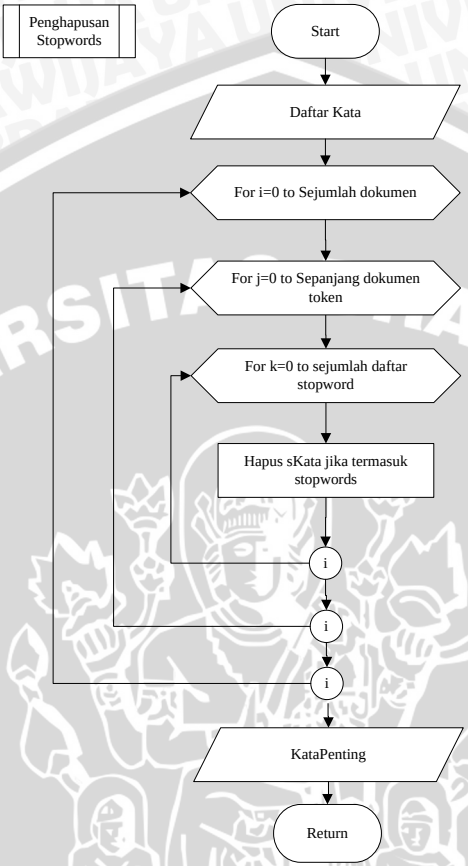


Gambar 3.5 *Flowchart* Proses *Tokenizing*

3.3.1.3 Penghapusan *Stopwords*

Hasil dari proses *tokenizing*, yang berupa daftar kata yang menyusun dokumen akan dilakukan proses penghapusan *stopwords*, yaitu menyeleksi kata yang dianggap penting dari dokumen, sedangkan kata yang tidak penting akan dihapus. *Stopwords* berisi daftar kata yang tidak penting seperti kata sambung dan kata tanya. Daftar *stopwords* yang digunakan dalam penelitian ini diambil dari penelitian Fadilla Z.Tala (Tala, 2003). Hasil dari proses ini adalah daftar kata penting yang menyusun dokumen.

Gambar 3.6 berikut merupakan *flowchart* proses penghapusan *stopwords:*

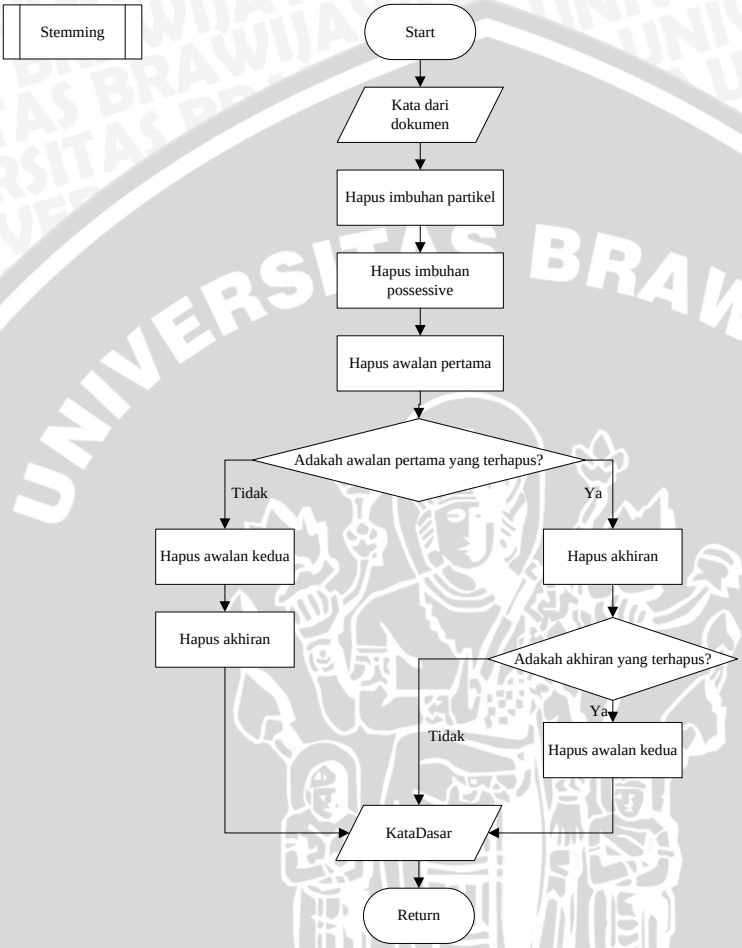


Gambar 3.6 *Flowchart* Proses Penghapusan *Stopwords*

3.3.1.4 *Stemming*

Setelah melalui proses penghapusan *stopwords*, proses berikutnya adalah mendapatkan bentuk dasar dari masing-masing kata penyusun dokumen yang disebut dengan *stemming*. Penguraian dilakukan dengan menerapkan berbagai macam aturan tertentu. Acuan algoritma *stemming* yang digunakan adalah algoritma *Porter Stemmer* yang telah dimodifikasi untuk Bahasa Indonesia berdasarkan penelitian Fadilla Z.Tala (Tala, 2003)

Gambar 3.7 berikut merupakan *flowchart* proses *stemming* untuk bahasa Indonesia:



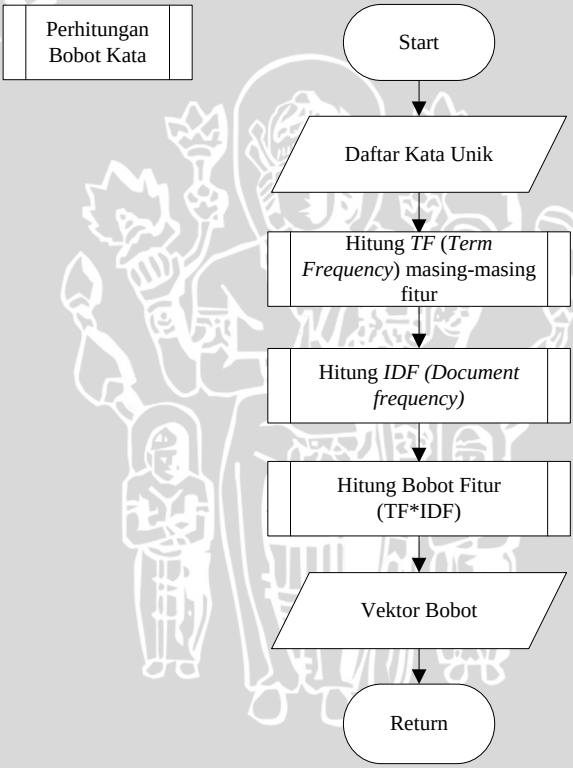
Gambar 3.7 *Flowchart* Proses *Stemming*

3.3.1.5 Perhitungan Bobot

Setelah melalui proses *stemming*, proses selanjutnya adalah menghitung bobot kata. Dalam perhitungan bobot, akan dibentuk *termlist* dari seluruh dokumen sebagai acuan untuk mengubah dokumen teks menjadi vektor. Keseluruhan *termlist* tersebut akan dihitung frekuensinya. Kemudian ditentukan bobot dari masing-masing *term* pada masing-masing dokumen menggunakan metode

TF-IDF (Term Frequency-Inverse Document Frequency). Proses awal dari *TF-IDF* ini, *Term Frequency (TF)* yaitu menghitung jumlah kemunculan tiap-tiap kata yang menyusun masing-masing dokumen. Setelah diperoleh nilai *TF* masing-masing kata dari masing-masing dokumen dan jumlah dokumen (*Document Frequency*), tahap berikutnya adalah menghitung nilai *Inverse Document Frequency (IDF)*. Pembobotan kata (*weighting*) merupakan hasil perkalian antara *TF* dan *IDF* dari pembobotan kata.

Gambar 3.8 berikut merupakan diagram *flowchart* proses pembobotan kata*:*



Gambar 3.8 *Flowchart* Proses Perhitungan Bobot

3.3.2 *Clustering* Dokumen

Dalam penelitian ini, proses *clustering* dokumen berita akan dilakukan menggunakan algoritma *DBSCAN (Density-Based Spatial Clustering of Application with Noise). DBSCAN* menentukan sendiri jumlah *cluster* yang akan dihasilkan. *DBSCAN* memerlukan dua inputan lain, yaitu parameter *minPts* yang membatasi minimal banyak *item* dalam suatu *cluster* dan parameter *eps* yang didefinisikan sebagai nilai *treshold* untuk jarak antar-*item* yang menjadi dasar pembentukan *neighbourhood* dari suatu titik *item.*

Berikut ini langkah-langkah dari proses *clustering* dokumen berita berbahasa Indonesia menggunakan algoritma *DBSCAN (Density-Based Spatial Clustering of Application with Noise)*:

1. Menentukan nilai parameter yang akan digunakan pada proses *clustering*, yaitu *set of points*, *eps*, dan *minPts.*
2. Menentukan inisialisasi *cluster id*
3. Pilihlah *point p* dari *set of points*
4. Kemudian tentukan apakah *point* belum memiliki *cluster* (belum masuk *cluster* manapun). Tentukan apakah *expandCluster* (parameter yang digunakan yaitu, *set of points*, *point*, *cluster id*, *eps*, dan *minPts*). Apabila *cluster* ter*expand*, maka *cluster id* selanjutnya meningkat. Dan jika tidak ter*expand*, maka langsung ke langkah selanjutnya.
5. Mengulang kembali langkah ketiga sehingga syarat pemberhentian terpenuhi yaitu ketika semua *point* sudah dianalisis (*next point to step* 5).

Berikut ini akan dijelaskan langkah-langkah untuk *expand cluster* dari proses:

1. Parameter *expand cluster* yang digunakan yaitu *set of points, point, cluster id, eps,* dan *minPts*.
2. Mencari *region query* dari *point* berdasarkan *epsilon (eps)* menggunakan perhitungan *cosine similarity* (persamaan 2.2) dan dari pencarian *region query* tersebut *point* dan hasilnya disimpan dalam *seeds.*
3. Membandingkan jumlah *seeds* dengan nilai *minPts* (dengan kata lain apakah *seeds size* lebih kecil dari nilai *minPts*).

4. Jika jumlah *seed* kurang dari nilai *minPts* (tidak adanya *core point*) maka *point* tersebut dianggap sebagai *noise* dan proses berhenti. Jika tidak, maka lakukan lakukan langkah 5 berikut:
5. Anggota dari *set of points* yang merupakan *seeds* maka *cluster id*-nya berubah.
6. Kemudian, hapus *point* dari *seeds.*
7. Selama *seeds* tidak kosong (*seeds* > 0), maka lakukan langkah 8
8. *Point* pertama dalam *seeds*, simpan dalam *currentP*
9. Mencari *region query* dari *point* berdasarkan *epsilon (eps)* menggunakan perhitungan *cosine similarity* (persamaan 2.2) dan dari pencarian *region query* tersebut maka *point* dan hasilnya disimpan dalam *result.*
10. Membandingkan jumlah *result* dengan nilai *minPts* (dengan kata lain apakah *result size* lebih besar sama dengan nilai *minPts*).
11. Untuk sejumlah *result* akan dilakukan iterasi dan hasilnya disimpan dalam *resultP*.
12. *ClusterId* untuk *resultP* merupakan *clusterId* sekarang.
13. Kemudian, hapus *currentP* dari *seeds*. Ulangi kembali langkah 7 sampai *seeds* kosong.
14. Proses *expand cluster* berhenti.

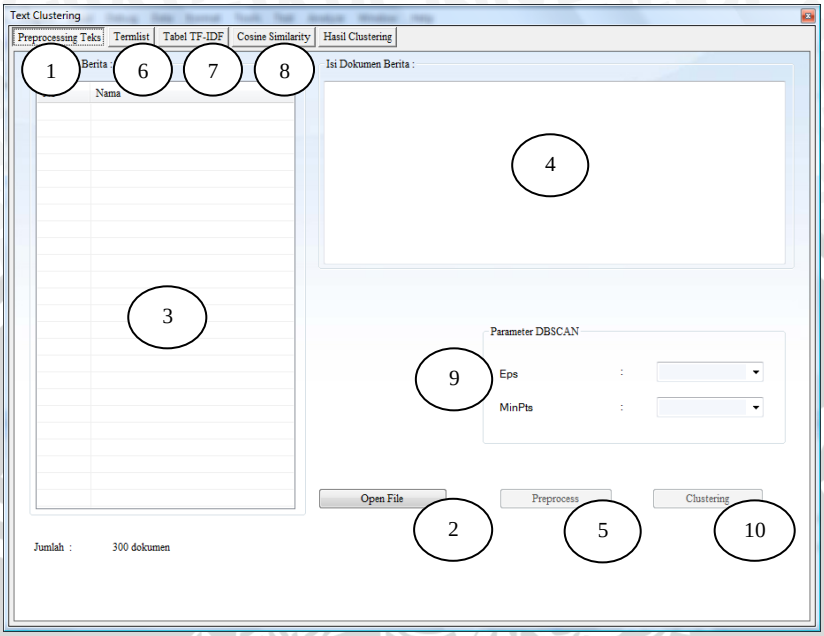
3.4 Perancangan Antar Muka

Pada sub bab ini akan dijelaskan mengenai perancangan antar muka yang digunakan dalam sistem *clustering* dokumen berita berbahasa Indonesia menggunakan algoritma *DBSCAN (Density Based Spatial Clustering of Application With Noise).* Perancangan antar muka terdiri dari dua *form* yaitu *form* awal dan *form clustering. Form* awal hanya berfungsi sebagai tampilan saja pada saat pertama kali menjalankan sistem, sedangkan *form clustering* terdiri dari lima tab yaitu, yang pertama *tab Preprocessing* Teks, yang kedua adalah *tab Termlist*, yang ketiga adalah *tab TF-IDF*, yang keempat adalah *tab Cosine Similarity*, dan yang kelima adalah *tab* Hasil *Clustering*. *Form clustering* bertujuan untuk menampilkan informasi-informasi mengenai jalannya proses dan untuk menampilkan hasil dari proses *clustering*.

Gambar 3.9 berikut menunjukkan perancangan antar muka untuk *form* Awal:

Gambar 3.9 *Form* Awal

Sedangkan perancangan antar muka untuk *form Clustering* yang akan dikembangkan, ditunjukkan pada gambar 3.10.

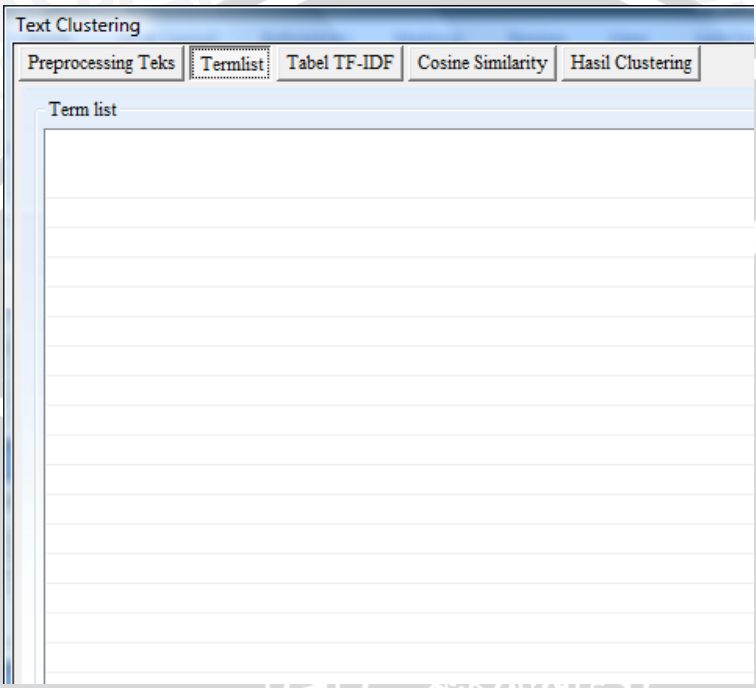


Gambar 3.10 *Form Clustering*

Adapun penjelasan dari bagian-bagian antar muka sistem adalah pada *form clustering* adalah sebagai berikut:

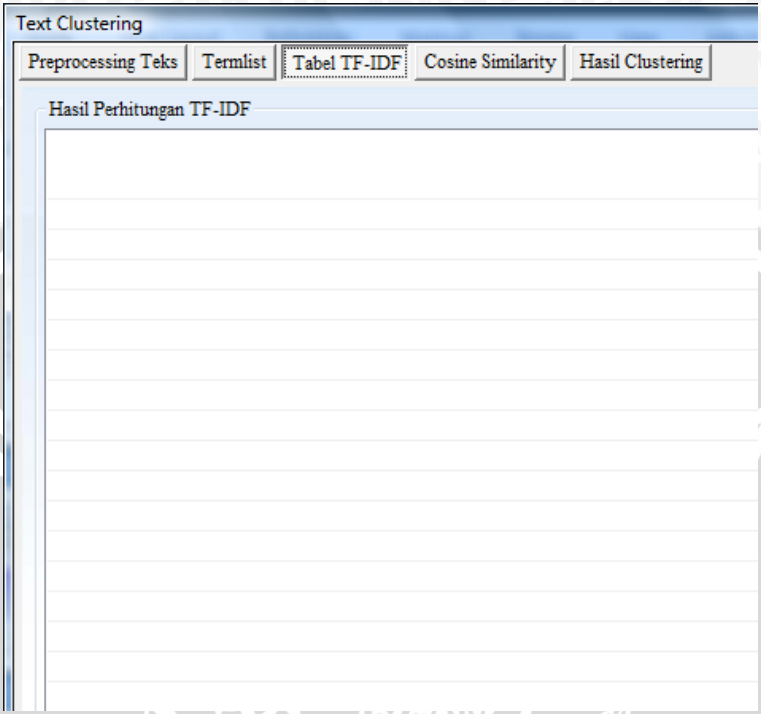
1. Merupakan tab *Preprocessing* Teks, berfungsi untuk menampilkan dokumen-dokumen yang akan digunakan dalam proses *clustering.*
2. Tombol *Open File*, berfungsi untuk memilih dokumen-dokumen yang akan digunakan dalam proses.
3. Dokumen Berita, untuk menampilkan nama dokumen dan lokasi dari dokumen yang akan dilakukan dalam proses.
4. Isi Dokumen Berita, bagian ini untuk menampilkan isi berita dari masing-masing dokumen pada *textbox.*
5. Tombol *Preprocess*, merupakan tombol yang digunakan untuk memulai *preprocessing.* Hasil yang akan ditampilkan berupa *termlist*, nilai bobot kata dan nilai similaritas dokumen yang dapat dilihat pada *tab Termlist*, *tab* Tabel *TF-IDF*, dan *tab Cosine Similarity.*

6. Merupakan *tab Termlist*, yang berfungsi menampilkan daftar kata unik dan frekuensi masing-masing kata. *Tab Termlist* ditunjukkan pada gambar 3.11.



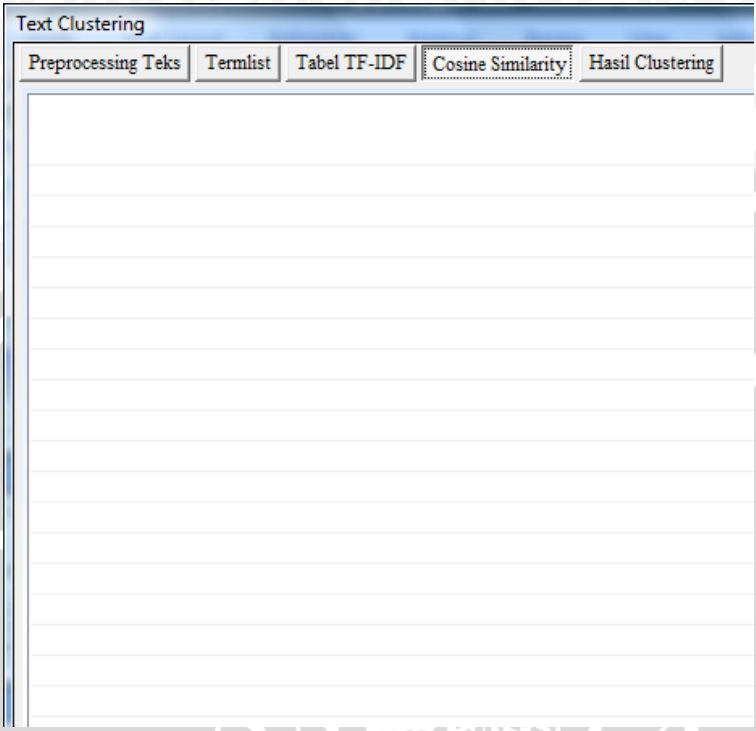
Gambar 3.11 Perancangan Antar Muka *Tab Termlist*

7. Merupakan *tab* Tabel *TF-IDF*, bagian ini untuk menampilkan nilai bobot kata dari hasil perhitungan *TF-IDF* dalam bentuk vektor. *Tab* Tabel *TF-IDF* dapat dilihat pada Gambar 3.12 berikut:



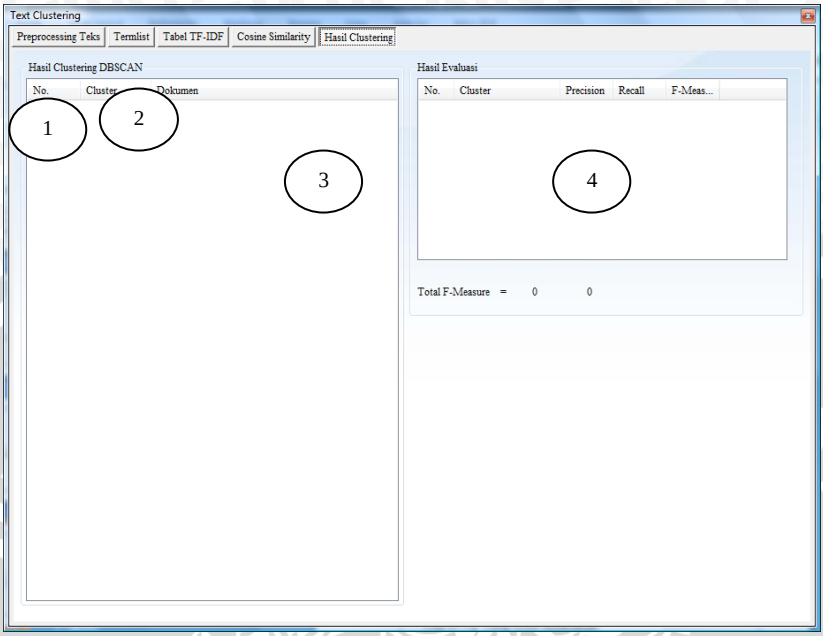
Gambar 3.12 Perancangan antar muka *tab* Tabel *TF-IDF*

8. Merupakan *tab Cosine similarity*, bagian ini untuk menampilkan hasil perhitungan *cosine similarity* (persamaan 2.2) yaitu nilai similaritas antar dokumen. *Tab Cosine similarity* dapat dilihat pada Gambar 3.13 berikut:



Gambar 3.13 Perancangan antar muka *tab Cosine Similarity*

9. Parameter *DBSCAN*, bagian ini untuk mengisi nilai parameter *eps* dan parameter *minPts* sebelum memulai proses.
10. Tombol *Clustering*, merupakan tombol yang digunakan untuk memulai proses *clustering*. Hasilnya akan dijelaskan pada Gambar 3.14 pada *tab* Hasil *Clustering.* Gambar 3.14 berikut menunjukkan *tab* Hasil *Clustering.*



Gambar 3.14 Perancangan Antar Muka *tab* Hasil *Clustering*

Adapun penjelasan dari bagian-bagian *tab* Hasil *Clustering* pada *listview* 1 dan *listview* 2 yang terdapat pada gambar 3.14 adalah:

1. No., bagian untuk menampilkan urutan dokumen yang akan ditampilkan dalam proses.
2. *Cluster*, bagian untuk menampilkan *id cluster* dari hasil proses *clustering* yang telah dilakukan.
3. Dokumen, bagian ini untuk menampilkan nama dan lokasi dokumen yang telah mengalami proses *clustering*.
4. Evaluasi, bagian ini untuk menampilkan detail evaluasi dari hasil *clustering* menggunakan *DBSCAN* berupa *precision, recall,* dan *F-Measure*.

3.5 Contoh Perhitungan Manual

Pada perhitungan manual menunjukkan proses keseluruhan secara umum. Data untuk perhitungan manual terdiri dari sepuluh dokumen berita berbahasa Indonesia. Tahap perhitungan dimulai dari *preprocessing* sampai proses *clustering.*

3.5.1 Contoh Proses *Case Folding*

Proses *case folding* merubah semua karakter menjadi karakter huruf kecil semua. Kemudian setelah diubah menjadi karakter huruf kecil, tahap selanjutnya membuang karakter angka, tanda baca dan simbol yang dianggap sebagai pemisah atau *delimiter* (spasi). Hasil dari proses *case folding* adalah dokumen yang hanya berisi karakter huruf kecil dari a-z dan hanya spasi tanda bacanya.

Misalkan dokumen d_1, sebuah dokumen berita dengan isi berita sebagai berikut:

> Harga emas yang melonjak membuat pembeli belum memburu perhiasan emas di bulan puasa.

Berikut ini adalah hasil perubahan dokumen d_1 (kata yang bercetak tebal merupakan kata yang sebelumnya mengandung huruf kapital).

> **harga** emas yang melonjak membuat pembeli belum memburu perhiasan emas di bulan puasa

3.5.2 Contoh Proses *Tokenizing*

Setelah melalui proses *case folding*, tahap berikutnya adalah *tokenizing*, yaitu tahap memecah dokumen menurut kata-kata yang menyusunnya dan membentuknya menjadi sebuah daftar atau *list* kata. Pemisahan kata ini berdasarkan karakter spasi. Hasil dari proses *tokenizing* adalah daftar kata yang menyusun dokumen tersebut.

Berikut ini merupakan contoh hasil proses *tokenizing* untuk dokumen pada contoh *case folding*:

harga
emas
yang
melonjak
membuat
pembeli
belum
memburu
perhiasan
emas
di
bulan
puasa

3.5.3 Contoh Proses Penghapusan *Stopwords*

Hasil dari proses *tokenizing* yaitu berupa daftar kata yang menyusun dokumen, kemudian akan disaring, kata mana yang penting dan mana kata yang tidak penting yang bisa dihapus. Proses ini disebut dengan penghapusan *stopwords*. Proses penghapusan *stopwords* adalah memandingkan kata-kata hasil *tokenizing* dengan daftar kata yang tidak penting yang ada pada *stopwords* (daftar kata tidak penting yang ada *stopwords* ditunjukkan pada Lampiran I).

Tujuan dari proses penghapusan *stopwords* ini adalah untuk mendapatkan kata-kata yang penting yang merepresentasikan isi dokumen. Hasil dari proses penghapusan *stopwords* ini adalah daftar kata penting yang menyusun dokumen. Berikut ini contoh hasil proses penghapusan *stopwords* dari hasil proses *tokenizing* pada contoh sebelumnya.

harga
emas
melonjak
pembeli
memburu
perhiasan
emas
puasa

3.5.4 Contoh Proses *Stemming*

Proses mengubah suatu kata menjadi bentuk kata dasarnya. Berikut ini contoh proses *stemming* pada dokumen hasil proses penghapusan *stopwords* pada contoh sebelumnya:

No.	Kata	Kata Dasar
1	harga	harga
2	emas	emas
3	melonjak	lonjak
4	pembeli	beli
5	memburu	buru
6	perhiasan	hias
7	emas	emas
8	puasa	puasa

3.5.5 Contoh Perhitungan Bobot

Setelah melalui proses *stemming*, proses selanjutnya adalah perhitungan bobot. Misalkan terdapat sepuluh dokumen berita berbahasa Indonesia. Dokumen pertama merupakan dokumen yang telah digunakan pada contoh sebelumnya, yaitu:

Dokumen d_1 :

> Harga emas yang melonjak membuat pembeli belum memburu perhiasan emas di bulan puasa.

Dokumen d_2 :

> Harga emas terus mencatat rekor seiring gejolak pasar saham akibat penurunan rating surat utang Amerika.

Dokumen d_3 :

> Harga emas terus meroket hampir menyentuh angka Rp500.000 per gram.

Dokumen d_4 :

Terus meroketnya harga emas hingga menembus rekor baru, US$1.750 troy ounces membantu kinerja aliran keuangan PT Newmont Nusa Tenggara

Dokumen d_5 :

Melani juga memberikan dua buah buku Kumpulan Dzikir dan Doa kepada Herawati

Dokumen d_6 :

Gempa bumi ternyata masih sering mengguncang wilayah negara Asia

Dokumen d_7 :

Para ilmuwan mengingatkan, wanita perokok memiliki berisiko dua kali lipat terkena serangan kanker paru-paru dibandingkan dengan pria perokok.

Dokumen d_8 :

Mendapati putra mereka, Dugan Smith didiagnosis kanker tulang di lutut dan harus diamputasi di usia 10, membuat kedua orang tua bocah Smith berpikir ribuan kali.

Dokumen d_9 :

Macmillan Cancer Support pun menemukan, bahwa lebih dari setengah dokter, perawat, dan perawat onkologi kanker tidak pernah menginformasikan tentang manfaat olahraga untuk para pasien kanker.

Dokumen d_{10} :

> Sebuah riset menemukan, orang yang merokok setelah bangun tidur berisiko hampir dua kali lipat terkena kanker ketimbang mereka yang menunggu satu jam dan sarapan lebih dulu.

Kemudian akan dikumpulkan kata (*term*) dari semua dokumen ke dalam daftar kata (*termlist*) dan menyimpan data kemunculan setiap *termlist* beserta frekuensinya. *Termlist* dari contoh dokumen 1 sampai dokumen 5 ditunjukkkan pada Tabel 3.2. Sedangkan *termlist* dari contoh dokumen 6 sampai dokumen 10 ditunjukkkan pada Tabel 3.3.

Tabel 3.2 Contoh *Termlist* (d_1-d_5)

Termlist	d_1	d_2	d_3	d_4	d_5
harga	1	1	1	1	0
emas	1	1	1	1	0
lonjak	1	0	0	0	0
beli	1	0	0	0	0
buru	1	0	0	0	0
hias	1	0	0	0	0
puasa	1	0	0	0	0
mencatat	0	1	0	0	0
rekor	0	1	0	1	0
seiring	0	1	0	0	0
gejolak	0	1	0	0	0
pasar	0	1	0	0	0
saham	0	1	0	0	0
akibat	0	1	0	0	0
turun	0	1	0	0	0
rating	0	1	0	0	0
surat	0	1	0	0	0

utang	0	1	0	0	0
amerika	0	1	0	0	0
roket	0	0	1	1	0
sentuh	0	0	1	0	0
angka	0	0	1	0	0
tembus	0	0	0	1	0
lani	0	0	0	0	1
buah	0	0	0	0	1
buku	0	0	0	0	1
kumpul	0	0	0	0	1
dzikir	0	0	0	0	1
doa	0	0	0	0	1
herawati	0	0	0	0	1
gempa	0	0	0	0	0
bumi	0	0	0	0	0
goncang	0	0	0	0	0
wilayah	0	0	0	0	0
negara	0	0	0	0	0
asia	0	0	0	0	0
ilmuw	0	0	0	0	0
wanita	0	0	0	0	0
okok	0	0	0	0	0
piliki	0	0	0	0	0
isiko	0	0	0	0	0
kali	0	0	0	0	0
lipat	0	0	0	0	0
kena	0	0	0	0	0
serang	0	0	0	0	0
kanker	0	0	0	0	0
paru	0	0	0	0	0

banding	0	0	0	0	0
pria	0	0	0	0	0
mendapati	0	0	0	0	0
putra	0	0	0	0	0
dugan	0	0	0	0	0
smith	0	0	0	0	0
diagnosis	0	0	0	0	0
tulang	0	0	0	0	0
lutut	0	0	0	0	0
amputasi	0	0	0	0	0
usia	0	0	0	0	0
orang	0	0	0	0	0
tua	0	0	0	0	0
bocah	0	0	0	0	0
pikir	0	0	0	0	0
ribu	0	0	0	0	0
macmill	0	0	0	0	0
cancer	0	0	0	0	0
support	0	0	0	0	0
temu	0	0	0	0	0
dokter	0	0	0	0	0
awat	0	0	0	0	0
onkologi	0	0	0	0	0
informasi	0	0	0	0	0
manfaat	0	0	0	0	0
olahraga	0	0	0	0	0
pasien	0	0	0	0	0
riset	0	0	0	0	0
rokok	0	0	0	0	0
bangun	0	0	0	0	0
tidur	0	0	0	0	0

timbang	0	0	0	0	0
tunggu	0	0	0	0	0
jam	0	0	0	0	0
sarap	0	0	0	0	0

Tabel 3.3 Contoh *Termlist* (d_6-d_{10})

Termlist	d_6	d_7	d_8	d_9	d_{10}
harga	0	0	0	0	0
emas	0	0	0	0	0
lonjak	0	0	0	0	0
beli	0	0	0	0	0
buru	0	0	0	0	0
hias	0	0	0	0	0
puasa	0	0	0	0	0
mencatat	0	0	0	0	0
rekor	0	0	0	0	0
seiring	0	0	0	0	0
gejolak	0	0	0	0	0
pasar	0	0	0	0	0
saham	0	0	0	0	0
akibat	0	0	0	0	0
turun	0	0	0	0	0
rating	0	0	0	0	0
surat	0	0	0	0	0
utang	0	0	0	0	0
amerika	0	0	0	0	0
roket	0	0	0	0	0
sentuh	0	0	0	0	0
angka	0	0	0	0	0
tembus	0	0	0	0	0

lani	0	0	0	0	0
buah	0	0	0	0	0
buku	0	0	0	0	0
kumpul	0	0	0	0	0
dzikir	0	0	0	0	0
doa	0	0	0	0	0
herawati	0	0	0	0	0
gempa	1	0	0	0	0
bumi	1	0	0	0	0
goncang	1	0	0	0	0
wilayah	1	0	0	0	0
negara	1	0	0	0	0
asia	1	0	0	0	0
ilmuw	0	1	0	0	0
wanita	0	1	0	0	0
okok	0	2	0	0	0
piliki	0	1	0	0	0
isiko	0	1	0	0	1
kali	0	1	1	0	0
lipat	0	1	0	0	1
kena	0	1	0	0	1
serang	0	1	0	0	0
kanker	0	1	1	2	1
paru	0	2	0	0	0
banding	0	1	0	0	0
pria	0	0	1	0	0
mendapati	0	0	1	0	0
putra	0	0	1	0	0
dugan	0	0	1	0	0
smith	0	0	2	0	0
diagnosis	0	0	1	0	0

tulang	0	0	1	0	0
lutut	0	0	1	0	0
amputasi	0	0	1	0	0
usia	0	0	1	0	0
orang	0	0	1	0	1
tua	0	0	1	0	0
bocah	0	0	1	0	0
pikir	0	0	1	0	0
ribu	0	0	1	0	0
macmill	0	0	0	1	0
cancer	0	0	0	1	0
support	0	0	0	1	0
temu	0	0	0	1	1
dokter	0	0	0	1	0
awat	0	0	0	2	0
onkologi	0	0	0	1	0
informasi	0	0	0	1	0
manfaat	0	0	0	1	0
olahraga	0	0	0	1	0
pasien	0	0	0	1	0
riset	0	0	0	0	1
rokok	0	0	0	0	1
bangun	0	0	0	0	1
tidur	0	0	0	0	1
timbang	0	0	0	0	1
tunggu	0	0	0	0	1
jam	0	0	0	0	1
sarap	0	0	0	0	1

Dari tabel 3.2 dan tabel 3.3, dapat diketahui frekuensi setiap kata dalam dokumen. Kemudian menghitung nilai bobot kata dengan persamaan persamaan 2.1. Contoh perhitungan bobot masing-masing kata dari dokumen 1 sampai dokumen 5 dapat dilihat pada tabel 3.4.

Tabel 3.4 Tabel Perhitungan Bobot Kata (d_1-d_5)

Term List	*df*	*idf = log (N/df)*	*w = tf * idf*				
			d_1	d_2	d_3	d_4	d_5
harga	4	0,40	0,40	0,40	0,40	0,40	0,00
emas	4	0,40	0,80	0,40	0,40	0,40	0,00
lonjak	1	1,00	1,00	0,00	0,00	0,00	0,00
beli	1	1,00	1,00	0,00	0,00	0,00	0,00
buru	1	1,00	1,00	0,00	0,00	0,00	0,00
hias	1	1,00	1,00	0,00	0,00	0,00	0,00
puasa	1	1,00	1,00	0,00	0,00	0,00	0,00
mencatat	1	1,00	0,00	1,00	0,00	0,00	0,00
rekor	2	0,70	0,00	0,70	0,00	0,70	0,00
seiring	1	1,00	0,00	1,00	0,00	0,00	0,00
gejolak	1	1,00	0,00	1,00	0,00	0,00	0,00
pasar	1	1,00	0,00	1,00	0,00	0,00	0,00
saham	1	1,00	0,00	1,00	0,00	0,00	0,00
akibat	1	1,00	0,00	1,00	0,00	0,00	0,00
turun	1	1,00	0,00	1,00	0,00	0,00	0,00
rating	1	1,00	0,00	1,00	0,00	0,00	0,00
surat	1	1,00	0,00	1,00	0,00	0,00	0,00
utang	1	1,00	0,00	1,00	0,00	0,00	0,00
amerika	1	1,00	0,00	1,00	0,00	0,70	0,00
roket	2	0,70	0,00	0,00	0,70	0,70	0,00
sentuh	1	1,00	0,00	0,00	1,00	0,00	0,00
angka	1	1,00	0,00	0,00	1,00	0,00	0,00
tembus	1	1,00	0,00	0,00	0,00	1,00	0,00

lani	1	1,00	0,00	0,00	0,00	0,00	1,00
buah	1	1,00	0,00	0,00	0,00	0,00	1,00
buku	1	1,00	0,00	0,00	0,00	0,00	1,00
kumpul	1	1,00	0,00	0,00	0,00	0,00	1,00
dzikir	1	1,00	0,00	0,00	0,00	0,00	1,00
doa	1	1,00	0,00	0,00	0,00	0,00	1,00
herawati	1	1,00	0,00	0,00	0,00	0,00	1,00
gempa	1	1,00	0,00	0,00	0,00	0,00	0,00
bumi	1	1,00	0,00	0,00	0,00	0,00	0,00
goncang	1	1,00	0,00	0,00	0,00	0,00	0,00
wilayah	1	1,00	0,00	0,00	0,00	0,00	0,00
negara	1	1,00	0,00	0,00	0,00	0,00	0,00
asia	1	1,00	0,00	0,00	0,00	0,00	0,00
ilmuw	1	1,00	0,00	0,00	0,00	0,00	0,00
wanita	1	1,00	0,00	0,00	0,00	0,00	0,00
okok	1	1,00	0,00	0,00	0,00	0,00	0,00
piliki	1	1,00	0,00	0,00	0,00	0,00	0,00
isiko	2	0,70	0,00	0,00	0,00	0,00	0,00
kali	3	0,52	0,00	0,00	0,00	0,00	0,00
lipat	2	0,70	0,00	0,00	0,00	0,00	0,00
kena	2	0,70	0,00	0,00	0,00	0,00	0,00
serang	1	1,00	0,00	0,00	0,00	0,00	0,00
kanker	4	0,40	0,00	0,00	0,00	0,00	0,00
paru	1	1,00	0,00	0,00	0,00	0,00	0,00
banding	1	1,00	0,00	0,00	0,00	0,00	0,00
pria	1	1,00	0,00	0,00	0,00	0,00	0,00
mendapati	1	1,00	0,00	0,00	0,00	0,00	0,00
putra	1	1,00	0,00	0,00	0,00	0,00	0,00
dugan	1	1,00	0,00	0,00	0,00	0,00	0,00
smith	1	1,00	0,00	0,00	0,00	0,00	0,00

diagnosis	1	1,00	0,00	0,00	0,00	0,00	0,00
tulang	1	1,00	0,00	0,00	0,00	0,00	0,00

Sedangkan contoh perhitungan bobot masing-masing kata dari dokumen 6 sampai dokumen 10 dapat dilihat pada tabel 3.5.

Tabel 3.5 Tabel Perhitungan Bobot Kata (d_6-d_{10})

Term List	*df*	*idf = log (N/df)*	*w = tf * idf*				
			d_6	d_7	d_8	d_9	d_{10}
harga	4	0,40	0,00	0,00	0,00	0,00	0,00
emas	4	0,40	0,00	0,00	0,00	0,00	0,00
lonjak	1	1,00	0,00	0,00	0,00	0,00	0,00
beli	1	1,00	0,00	0,00	0,00	0,00	0,00
buru	1	1,00	0,00	0,00	0,00	0,00	0,00
hias	1	1,00	0,00	0,00	0,00	0,00	0,00
puasa	1	1,00	0,00	0,00	0,00	0,00	0,00
mencatat	1	1,00	0,00	0,00	0,00	0,00	0,00
rekor	2	0,70	0,00	0,00	0,00	0,00	0,00
seiring	1	1,00	0,00	0,00	0,00	0,00	0,00
gejolak	1	1,00	0,00	0,00	0,00	0,00	0,00
pasar	1	1,00	0,00	0,00	0,00	0,00	0,00
saham	1	1,00	0,00	0,00	0,00	0,00	0,00
akibat	1	1,00	0,00	0,00	0,00	0,00	0,00
turun	1	1,00	0,00	0,00	0,00	0,00	0,00
rating	1	1,00	0,00	0,00	0,00	0,00	0,00
surat	1	1,00	0,00	0,00	0,00	0,00	0,00
utang	1	1,00	0,00	0,00	0,00	0,00	0,00
amerika	1	1,00	0,00	0,00	0,00	0,00	0,00
roket	2	0,70	0,00	0,00	0,00	0,00	0,00
sentuh	1	1,00	0,00	0,00	0,00	0,00	0,00

angka	1	1,00	0,00	0,00	0,00	0,00	0,00
tembus	1	1,00	0,00	0,00	0,00	0,00	0,00
lani	1	1,00	0,00	0,00	0,00	0,00	0,00
buah	1	1,00	0,00	0,00	0,00	0,00	0,00
buku	1	1,00	0,00	0,00	0,00	0,00	0,00
kumpul	1	1,00	0,00	0,00	0,00	0,00	0,00
dzikir	1	1,00	0,00	0,00	0,00	0,00	0,00
doa	1	1,00	0,00	0,00	0,00	0,00	0,00
herawati	1	1,00	0,00	0,00	0,00	0,00	0,00
gempa	1	1,00	1,00	0,00	0,00	0,00	0,00
bumi	1	1,00	1,00	0,00	0,00	0,00	0,00
goncang	1	1,00	1,00	0,00	0,00	0,00	0,00
wilayah	1	1,00	1,00	0,00	0,00	0,00	0,00
negara	1	1,00	1,00	0,00	0,00	0,00	0,00
asia	1	1,00	1,00	0,00	0,00	0,00	0,00
ilmuw	1	1,00	0,00	1,00	0,00	0,00	0,00
wanita	1	1,00	0,00	1,00	0,00	0,00	0,00
okok	1	1,00	0,00	2,00	0,00	0,00	0,00
piliki	1	1,00	0,00	1,00	0,00	0,00	0,00
isiko	2	0,70	0,00	0,70	0,00	0,00	0,00
kali	3	0,52	0,00	0,52	0,52	0,00	0,52
lipat	2	0,70	0,00	0,70	0,00	0,00	0,70
kena	2	0,70	0,00	0,70	0,00	0,00	0,70
serang	1	1,00	0,00	1,00	0,00	0,00	0,00
kanker	4	0,40	0,00	0,40	0,40	0,80	0,40
paru	1	1,00	0,00	2,00	0,00	0,00	0,00
banding	1	1,00	0,00	1,00	0,00	0,00	0,00
pria	1	1,00	0,00	1,00	0,00	0,00	0,00
mendapati	1	1,00	0,00	0,00	1,00	0,00	0,00
putra	1	1,00	0,00	0,00	1,00	0,00	0,00

dugan	1	1,00	0,00	0,00	1,00	0,00	0,00
smith	1	1,00	0,00	0,00	2,00	0,00	0,00
diagnosis	1	1,00	0,00	0,00	1,00	0,00	0,00
tulang	1	1,00	0,00	0,00	1,00	0,00	0,00

3.5.6 Contoh Proses *Clustering* Dokumen

Misal terdapat sepuluh dokumen berita berbahasa Indonesia. Kesepuluh dokumen tersebut telah disimpan dalam satu folder. Tabel 3.6 berikut menunjukkan vektor dokumen dari kesepuluh dokumen berita berbahasa Indonesia yang telah mengalami *preprocessing*.

Tabel 3.6 Contoh Vektor Dokumen

	w_1	w_2	w_3	w_4	w_5	$w_{....}$	w_{82}
d_1	0,40	0,80	1,00	1,00	1,00	...	0,00
d_2	0,40	0,40	0,00	0,00	0,00	...	0,00
d_3	0,40	0,40	0,00	0,00	0,00	...	0,00
d_4	0,40	0,40	0,00	0,00	0,00	...	0,00
d_5	0,00	0,00	0,00	0,00	0,00	...	0,00
d_6	0,00	0,00	0,00	0,00	0,00	...	0,00
d_7	0,00	0,00	0,00	0,00	0,00	...	0,00
d_8	0,00	0,00	0,00	0,00	0,00	...	0,00
d_9	0,00	0,00	0,00	0,00	0,00	...	0,00
d_{10}	0,00	0,00	0,00	0,00	0,00	...	1,00

Hasil dari *preprocessing* yang berupa vektor dokumen ini akan digunakan untuk proses selanjutnya, yaitu *clustering* menggunakan *DBSCAN*. Berikut ini adalah contoh perhitungan manual proses *clustering* dokumen berita berbahasa Indonesia menggunakan algoritma *DBSCAN*:

1. Menentukan parameter yang digunakan dalam *DBSCAN* (*setOfPoints* : jumlah *point* yang diproses, *eps* : nilai *treshold* untuk jarak antar *point* yang menjadi dasar pembentukan *neighbourhood* dari suatu titik *point,* dan *minPts* : minimal banyak *points* dalam suatu *cluster):*

   SetOfPoints : $d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}$
   Eps : 0,02
   MinPts : 4

2. Inisialisasi *cluster id* awal
3. Iterasi pertama, diambil *point* ke-i dari *SetOfPoints*. Misal. *point* ← d_1
4. Berikutnya, lakukan *expandCLuster* pada *point* (d_1).

Langkah-langkah *expandCLuster* untuk *point* (d_1) akan dijelaskan sebagai berikut:

1. Parameter *expandCluster* yang digunakan, yaitu:

   SetOfPoints : $d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}$
   Point : d_1
   Cluster Id : 1
   Eps : 0,02
   MinPts : 4

2. Mencari *regionQuery* dari *point* (d_1) berdasarkan *epsilon* kemudian simpan dalam s*eeds*. *Region query* diperoleh dari perbandingan hasil perhitungan *cosine similarity* (persamaan 2.4) dan nilai *eps*. Tabel 3.7 berikut merupakan hasil perhitungan *cosine similarity* untuk d_1 sampai d_5.

Tabel 3.7 Nilai Similaritas (d_1-d_5)

	d_1	d_2	d_3	d_4	d_5
d_1	1	0,057	0,117	0,130	0
d_2	0,057	1	0,055	0,154	0
d_3	0,117	0,055	1	0,317	0
d_4	0,130	0,154	0,317	1	0
d_5	0	0	0	0	1
d_6	0	0	0	0	0
d_7	0	0	0	0	0
d_8	0	0	0	0	0
d_9	0	0	0	0	0
d_{10}	0	0	0	0	0

Sedangkan tabel 3.8 berikut merupakan hasil perhitungan *cosine similarity* untuk d_6 sampai d_5.

Tabel 3.8 Nilai Similaritas (d_6-d_{10})

	d_6	d_7	d_8	d_9	d_{10}
d_1	0	0	0	0	0
d_2	0	0	0	0	0
d_3	0	0	0	0	0
d_4	0	0	0	0	0
d_5	0	0	0	0	0
d_6	1	0	0	0	0
d_7	0	1	0,026	0,021	0,144
d_8	0	0,026	1	0,020	0,067
d_9	0	0,021	0,020	1	0,064
d_{10}	0	0,144	0,067	0,064	1

Dari tabel 3.7 dan tabel 3.8 dilakukan pengecekan satu per satu dari *point* d_2 sampai dengan *point* d_{10} untuk mendapatkan *region query* dari *point* d_1. Hasil dari perhitungan tersebut diperoleh bahwa *region query* dari *point* d_1 adalah d_2, d_3, d_4. Maka *seeds* = d_1, d_2, d_3, d_4.

Contoh perhitungan nilai similaritas antara *point* d_1 dan *point* d_2 menggunakan persamaan 2.2 adalah sebagai berikut.

d_1 = (0,40; 0,80; 1,00; 1,00; 1,00; 1,00; 1,00; 0,00; 0,00; 0,00; 0,00; 0,00; 0,00; 0,00; 0,00; 0,00; 0,00; 0,00; 0,00)
d_2 = (0,40; 0,40; 0,00; 0,00; 0,00; 0,00; 0,00; 1,00; 0,70; 1,00; 1,00; 1,00; 1,00; 1,00; 1,00; 1,00; 1,00; 1,00; 1,00)

$$cosine(d_1, d_2) = \frac{\sum_{i=1}^{n} d_1 . d_2}{\sqrt{\sum_{i=1}^{n}(d_1)^2} . \sqrt{\sum_{i=1}^{n}(d_2)^2}}$$

Jika persamaan tersebut dijabarkan dan dihitung satu per satu maka diperoleh perhitungan = $\sum_{i=1}^{n} d_1 . d_2$

= (0,40 x 0,40) + (0,80 x 0,40) + (1,00 x 0,00) + (1,00 x 0,00) + (1,00 x 0,00) + (1,00 x 0,00) + (1,00 x 0,00) + (0,00 x 1,00) + (0,00 x 0,70) + (0,00 x 1,00) + (0,00 x 1,00) + (0,00 x 1,00) + (0,00 x 1,00) + (0,00 x 1,00) + (0,00 x 1,00) + (0,00 x 1,00) + (0,00 x 1,00)
= 0,16 + 0,32 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0
= 0,48

Kemudian, menghitung panjang masing-masing vektor. Pertama, panjang vektor *point* d_1 $= \sqrt{\sum_{j=1}^{n}(d_1)^2}$

$$= \sqrt{\left((0{,}40)^2 + (0{,}80)^2 + (1)^2 + (1)^2 + (1)^2 + (1)^2 + (1)^2 + 0\right)}$$
$$= \sqrt{0{,}16 + 0{,}64 + 1 + 1 + 1 + 1 + 1}$$
$$= \sqrt{5{,}8}$$
$$= 2{,}408$$

Yang kedua, panjang vektor *point* d_2 $= \sqrt{\sum_{j=1}^{n}(c_1)^2}$

$$= \sqrt{\left((0{,}40)^2 + (0{,}40)^2 + (0)^2 + (1)^2 + (0{,}7)^2 + (10)\right)}$$
$$= \sqrt{0{,}16 + 0{,}16 + 0 + 1 + 0{,}49 + 10}$$
$$= \sqrt{11{,}81}$$
$$= 3{,}436$$

Sehingga nilai dari perhitungan similaritas antara *point* d_1 dan *point* d_2 adalah:

$$cosine(d_1, d_2) = \frac{\sum_{i=1}^{n} d_1 . d_2}{\sqrt{\sum_{i=1}^{n}(d_1)^2} . \sqrt{\sum_{i=1}^{n}(d_2)^2}}$$

$$= \frac{0{,}48}{2{,}408 * 3{,}436} = \frac{0{,}48}{8{,}273}$$

$$= 0{,}057$$

Jadi, nilai similaritas antara *point* d_1 dan *point* d_2 adalah 0,057.

3. Berikutnya, membandingkan jumlahnya *seeds* dengan nilai *minPts*.
   jumlah *seeds* = 4
   nilai *minPts* = 4
   a. 4 < 4 , tidak benar. Maka, lakukan langkah 5.
4. Jika jumlahnya *seeds* kurang dari nilai *minPts*, maka *point* dianggap sebagai *noise.*
5. Anggota dari *set of points* yang merupakan *seeds* maka *cluster id*- nya berubah.
6. Hapus *point* d_1 dari *seeds*.
7. Selama *seeds* tidak kosong (*seeds* > 0), maka lakukan langkah berikutnya, yaitu langkah 7.
8. *Point* d_2 yang merupakan isi pertama dari *seeds* akan disimpan dalam *currentP*.
   currentP ← *point* d_2
9. Mencari *region query* dari d_2. Dari tabel 3.7 dan tabel 3.8 dilakukan pengecekan satu per satu dari tiap *point* untuk mendapatkan *region query* dari *point* d_2. Hasil dari perhitungan tersebut diperoleh bahwa *region query* dari *point* d_2 adalah d_1, d_3, d_4. Maka *result* = d_2, d_1, d_3, d_4.
10. Membandingkan jumlahnya *result* dengan nilai *minPts.*
    jumlah *result* = 4
    nilai *minPts* = 4
    b. 4 >= 4, benar. Maka, lakukan langkah 11.
11. Lakukan iterasi untuk sejumlah *result* (d_2, d_1, d_3, d_4). *Result* ke-i simpan dalam *resultP*.
    resultP ← d_2 (sudah ter*cluster*)
    resultP ← d_1 (sudah ter*cluster*)
    resultP ← d_3 (sudah ter*cluster*)
    resultP ← d_4 (sudah ter*cluster*)
12. Semua yang ada dalam *resultP* adalah *cluster* 1.
13. Hapus *currentP* (d_2) dari *seeds*. Kembali ke langkah 7, sampai *seeds* kosong (*seeds* = 0).
    Iterasi pertama berhenti, diperoleh C_1 = d_1, d_2, d_3, d_4.

Untuk *point-point* berikutnya, akan dilakukan proses yang sama dari langkah awal *DBSCAN*, *expandCluster* dari langkah 1 sampai langkah 13 (sampai terbentuk *cluster*). Setelah melalui iterasi tertentu pada proses *clustering* maka terbentuklah 2 *cluster* dan munculnya *noise* dimana distribusi dokumen didalamnya ditunjukkan pada tabel 3.9.

Tabel 3.9 Hasil *Clustering DBSCAN*

C_1	C_2	*Noise*
d_1, d_2, d_3, d_4	d_7, d_8, d_9, d_{10}	d_5, d_6

3.6 Perancangan Uji Coba

Setelah perangkat lunak telah disusun, langkah selanjutnya adalah melakukan pengujian dan juga evaluasi sistem. Uji coba dilakukan untuk mengevaluasi keakuratan dalam hasil *clustering* yang dilakukan oleh sistem. Tingkat akurasi sistem diukur dengan standar evaluasi *F-measure*.

3.6.1 Skenario Evaluasi

Pada pengujian sistem, data yang diambil dari *website* www.VIVAnews.com akan dijadikan sebagai data uji sekaligus data pembanding. Dokumen berita berbahasa Indonesia yang digunakan sebagai data uji dikumpulkan dan diolah dari dokumen berita berbahasa Indonesia yang diambil dari *website* pada kisaran tanggal 4 Agustus 2011 hingga 7 Oktober 2011 dengan jumlah total 300 dokumen yang di*cluster* secara manual ke dalam 5 *cluster* yang berbeda. Masing-masig *cluster* tersebut terdiri dari 60 dokumen berita. Hasil *clustering* secara manual tersebut yang akan digunakan sebagai *cluster* pembanding untuk evaluasi sistem.

Masukan berupa nilai parameter (*eps* dan *minPts*) yang berbeda-beda akan diujikan untuk mengetahui nilai *F-measure* terbaik. Kemudian hasil dari *cluster* yang terbentuk dengan menggunakan algoritma *DBSCAN* akan dibandingkan dengan *cluster* yang telah disusun secara manual.

3.6.2 Hasil Evaluasi

Untuk mengetahui tentang keakuratan sistem, hasil proses *clustering* yang dilakukan oleh sistem akan digunakan dalam perhitungan nilai *F-measure* (pada persamaan 2.5). Pengujian akan dilakukan beberapa kali dengan jumlah dokumen uji yang sama yaitu sebanyak 300 dokumen berita berbahasa Indonesia tentunya dengan perbedaan input parameter. Parameter yang digunakan adalah parameter *epsilon (Eps)* dan parameter *minPts*, keduanya sangat berpengaruh pada hasil *clustering* karena menentukan besarnya nilai akurasi. Hasil evaluasi *F-measure* yang diperoleh secara keseluruhan ditunjukkan pada tabel 3.10, tabel 3.11, dan tabel 3.12.

Tabel 3.10 Tabel Skenario 1

Percobaan	**Parameter**		**F-measure (%)**
	minPts	***eps***	
1			
2			
3			
4			
5			
...			
n			

Tabel 3.11 Tabel Skenario 2

Percobaan	**Parameter**		**F-measure (%)**
	minPts	***eps***	
1			
2			
3			
4			
5			
...			
n			

Tabel 3.12 Tabel Skenario *n*

Percobaan	**Parameter**		**F-measure (%)**
	minPts	***eps***	
1			
2			
3			
4			
5			
...			
n			

dimana:
skenario 1 : *minPts* (...) dan *eps* (...)
skenario 2 : *minPts* (...) dan *eps* (...)
skenario *n* : *minPts* (...) dan *eps* (...)

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

4.1 Lingkungan Implementasi

Perangkat lunak ini akan digunakan untuk proses *clustering* dokumen berita berbahasa Indonesia dengan menggunakan algoritma *DBSCAN (Density-Based Spatial Clustering of Application With Noise).* Inputan yang digunakan adalah berupa dokumen berita yang sudah diformat dalam bentuk *file* teks. Perangkat lunak ini membutuhkan dua parameter lain, yaitu parameter *epsilon* dan parameter *minPts*.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem *clustering* dokumen berita berbahasa Indonesia dengan metode *DBSCAN* adalah sebagai berikut:

1. *Processor* Intel Core™2 Duo T5800.
2. *Memory* 4GB 800MHz SDRAM.
3. *Harddisk* 250GB.
4. Monitor 14.1" diagonal widescreen TruBrite display.
5. *Keyboard.*
6. *Mouse.*

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan untuk pengembangan sistem *clustering* dokumen berita berbahasa Indonesia dengan algoritma *DBSCAN* adalah sebagai berikut:

1. Sistem operasi Windows Vista Home Bussiness.
2. Microsoft Visual Studio 2008 bahasa pemrograman *C#.*
3. Text editor Notepad.

4.2 Implementasi Program

Berdasarkan perancangan pada bab 3, maka pada bab ini akan dibahas mengenai implementasi dari perancangan tersebut. Secara garis besar proses terbagi menjadi dua bagian utama, yaitu *preprocessing* dan *clustering DBSCAN*.

4.2.1 Struktur Data

Struktur data yang digunakan dalam aplikasi akan ditunjukkan pada *SourceCode* 4.1.

```
public partial class Clustering : Form
{
    string[,] dok;
    string[][] dokToken;
    ArrayList daftarStopwords;
    string[][] dokStopwords;
    string[][] dokStemming;
    ArrayList termList;
    string[] termArray;

    string vocal = "aiueo";

    string[][][]TF;
    string[][] IDF;
    string[][][] Bobot;

    ArrayList nilaiPoints;
    double[] d;
    int csize;
```

```
class MyPoint
{
    public Double[] Vektor;
    public int ClusterId;
    public string NamaDok;
    public MyPoint(Double[] Vektor)
    {
        this.Vektor = Vektor;
    }
}
```

Sourcecode 4.1 Struktur Data

Struktur data yang digunakan dalam aplikasi ini digunakan sebagai fungsi penyimpanan dan untuk melakukan perhitungan mulai dari preproses sampai proses *clustering.*

a. `dok`, variabel ini berfungsi untuk menyimpan sejumlah dokumen bertipe `string` dengan *array* dua dimensi. Dimensi pertama berisi nama dokumen dan dimensi kedua merupakan isi dari masing-masing dokumen.
b. `dokToken`, variabel ini memiliki tipe `string` dengan *jagged array*. Berfungsi untuk menyimpan dokumen yang isinya *token-token.* Ukuran dimensi *jagged array* yang pertama adalah sejumlah dokumen, dan ukuran dimensi *jagged array* yang kedua adalah *token-token.*
c. `daftarStopwords`, variabel ini memiliki tipe `arraylist`. Berfungsi untuk menyimpan kata-kata tidak penting yang masuk dalam daftar *stopwords.*
d. `dokStopwords`, variabel bertipe `string` dengan *jagged array* ini berfungsi untuk menyimpan kata-kata penting hasil dari proses penghapusan *stopword* (menyimpan kata-kata yang tidak ada dalam daftar *stopword*). Ukuran dimensi *jagged array* yang pertama adalah sejumlah dokumen, dan ukuran dimensi *jagged array* yang kedua sebanyak jumlah *token.*
e. `dokStemming`, variabel bertipe `string` dengan *jagged array* ini berfungsi untuk menyimpan kata-kata hasil dari proses *stemming.* Ukuran dimensi *jagged array* yang pertama adalah sejumlah dokumen, dan ukuran dimensi *jagged array* yang kedua sebanyak jumlah *token* yang telah mengalami proses *stemming.*
f. `termList`, variabel bertipe `arraylist`. Berfungsi untuk menyimpan daftar semua *term* dari keseluruhan dokumen.
g. `termArray`, variabel bertipe `string` dengan *array* satu dimensi ini merupakan konversi dari `termList`, berfungsi untuk menyimpan daftar semua *term* dari `termList`.
h. `vocal`, variabel bertipe `string` ini digunakan dalam proses *stemming* (untuk mengenali jenis huruf vokal).
i. `TF`, variabel bertipe `string` dengan *jagged array* ini berfungsi untuk menyimpan *term* beserta frekuensi kemunculannya dalam satu dokumen (*term frequency*). Ukuran dimensi *jagged array* yang pertama adalah sejumlah `termList`, ukuran dimensi *jagged array* yang kedua sebanyak jumlah `dok`, dan ukuran dimensi *jagged array* yang ketiga adalah `temp` dari sejumlah `dok` yang berisi jumlah frekuensi kemunculan *term.*

j. `IDF`, variabel bertipe `string` dengan *jagged array* ini berfungsi menyimpan hasil dari proses perhitungan *IDF*.
k. `Bobot`, variabel bertipe `string` dengan *jagged array* befungsi menyimpan hasil dari proses perhitungan bobot kata.
l. `nilaiPoints`, variabel bertipe `arraylist` ini berfungsi untuk menyimpan *point-point* yang berisi dokumen-dokumen untuk proses *clustering.*
m. `d`, variabel bertipe `double array` yang berfungsi untuk memasukkan fitur.
n. `csize`, variabel bertipe `int` yang berfungsi untuk menyimpan ukuran *cluster* dari sistem.
o. `class MyPoint`, merupakan representasi objek-objek yang digunakan dalam proses *clustering*, `class MyPoint` berisi `public Double[] Vektor`, `public int ClusterId`, `public string NamaDok`, dan `public MyPoint`.

4.2.2 Implementasi *Preprocessing*

Preprocessing merupakan tahap awal dari proses, dalam preproses ini terdapat proses *case folding, tokenizing*, penghapusan *stopwords*, dan *stemming.*

4.2.2.1 *Input* Dokumen

Pada proses ini, sejumlah 300 dokumen yang telah disimpan dalam format **.txt* akan di*input*kan secara bersamaan dan dilakukan pembacaan setiap dokumen. Pada dokumen akan dilakukan *preprocessing* sehingga membentuk *termlist* beserta frekuensi kemunculan kata dari masing-masing dokumen, *array* yang merupakan hasil perhitungan *cosine similarity* dan juga perhitungan bobot.

Untuk melakukan *input* data, digunakan komponen `OpenFileDialog`. Karena dokumen teks yang akan diproses terdiri dari banyak dokumen, maka proses *input* berasal dari satu *folder*. Proses *input* dokumen akan ditunjukkan pada *sourcecode* 4.2.

```
private void btnOpen_Click(object sender, EventArgs
e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Multiselect = true;
    ofd.ShowDialog();
    string[] filename = ofd.FileNames;

    ListViewItem lsv;
    listViewDok.Items.Clear();
    listViewDok.Columns.Clear();
    listViewDok.Columns.Add("No.", 40);
    listViewDok.Columns.Add("Nama", 250);
    listViewDok.Columns.Add("", 1);

    dok = new string[filename.Length, 2];
    for (int i = 0; i < filename.Length; i++)
    {
        lsv = new ListViewItem((i + 1).ToString());
        lsv.SubItems.Add(filename[i]);
        listViewDok.Items.Add(lsv);
```

Sourcecode 4.2 *Input* Dokumen

Fungsi tersebut melakukan pembacaan setiap judul *file* yang ada dalam folder yang dipilih dan menampilkannya dalam sebuah `listView` yang diberi nama `listViewDok`. Isi dokumen berita pada setiap judulnya akan ditampilkan ke dalam `textbox`, ditunjukkan pada *sourcecode* 4.3 yaitu pada *event* `listViewDok_SelectedIndexChanged`.

```
private void listViewDok_SelectedIndexChanged(object
sender, EventArgs e)
{
    if (listViewDok.SelectedItems.Count > 0)
    {
        string filename =
        listViewDok.Items[listViewDok.SelectedIndices
        [0]].SubItems[1].Text;
        FileStream fs = new FileStream(filename,
        FileMode.Open);
        StreamReader str = new StreamReader(fs);
```

```
//Lanjutan..
        string data = str.ReadToEnd();
        txtIsiDok.Text = data;
        str.Close();
        fs.Close();
    }
}
```

Sourcecode 4.3 Fungsi Menampilkan Isi Dokumen

Pada *sourcecode* 4.3 `string` data digunakan sebagai variabel, untuk membaca dokumen digunakan fungsi `ReadToEnd()`. Hasil pembacaan dokumen berita akan ditampilkan dalam sebuah `textbox`, yaitu dengan nama `txtIsiDok`.

4.2.2.2 *Case Folding*

Tahap awal dari sistem *clustering* dokumen berita berbahasa Indonesia ini adalah melakukan *preprocessing*. Tahap awal dari *preprocessing* adalah *case folding*. Potongan *sourcecode* pada tahap *case folding* dapat ditunjukkan pada *Sourcecode* 4.4.

```
        dok[i, 0] = filename[i];
        FileStream fs = new FileStream(filename[i],
        FileMode.Open);
        StreamReader str = new StreamReader(fs);
        string data = str.ReadToEnd();

        str.Close();
        fs.Close();
        dok[i, 1] = data.ToLower();

        //Buang angka dan karakter
        string[] spasi = new string[] { ",", "\"",
        "?", ")", "`", "'","(", ".", "/", "-", "\r",
        "\n", ":", "\\", "!", "{", "}", "+", "*",
        "$", ">", "<",
                "0", "1", "2", "3", "4", "5", "6",
        "7","8", "9"};
```

```
//Lanjutan...
        for (int j = 0; j < spasi.Length; j++)
        {
            dok[i, 1] = dok[i, 1].Replace(spasi[j], "
            ");
        }
        btnPreproses.Enabled = true;
        progBarPreproses.Visible = true;
        this.Refresh();
}
```

Sourcecode 4.4 Tahap *Case Folding*

Pada tahap *case folding* digunakan fungsi `ToLower()` untuk mengubah setiap huruf dari `data` yang bertipe `string` menjadi huruf kecil (*lower case*) yang akan disimpan dalam `dok[i, 1]`. Kemudian dilanjutkan dengan menghilangkan angka dan karakter lain selain huruf (yang disimpan dalam variable bernama `spasi`) sehingga suatu dokumen hanya terdiri dari kata dan istilah sederhana (*clear word*).

4.2.2.3 *Tokenizing*

Tahap selanjutnya adalah *tokenizing*, ditunjukkan pada *Sourcecode* 4.5.

```
private void Tokenizing()
{
    char[] delimiter = {' '};
    dokToken = new string[dok.GetLength(0)][];
    for (int i = 0; i < dok.GetLength(0); i++)
    {
        string[] token = dok[i,1].Split(delimiter,
        StringSplitOptions.RemoveEmptyEntries);
        dokToken[i] = new string[token.Length];
        for(int j=0; j < token.Length; j++)
        {
            dokToken[i][j] = token[j];
        }
    }
}
```

Sourcecode 4.5 Fungsi *Tokenizing*

Pada tahap *tokenizing*, pertama *variabel array* `char` yang bernama `delimiter` digunakan untuk meyimpan karakter spasi. Kemudian ditentukan ukuran dimensi *jagged* yang pertama dari `dokToken` yaitu sejumlah `dok`. Selanjutnya, adalah memecah-mecah isi dokumen menjadi potongan-potongan kata yang membentuknya. Proses ini menggunakan fungsi `Split`() dengan karakter spasi sebagai pemisah (*delimiter*) untuk mengurai suatu teks menjadi bentuk per kata. Artinya, ketika ditemukan karakter spasi maka akan langsung dilakukan pemotongan terhadap *string* tersebut. Variabel *array string* bernama `token` digunakan untuk menyimpan bagian-bagian kata yang telah dipisahkan oleh fungsi `Split()`. Setelah itu, ditentukan ukuran *jagged array* yang kedua dari `dokToken` yaitu sejumlah `token`, dan untuk sejumlah `token`, `token ke-j` akan disimpan dalam `dokToken[i][j]`.

4.2.2.4 Penghapusan *Stopwords*

Penghapusan *stopwords* bertujuan untuk memperoleh kata-kata penting dengan menghilangkan kata-kata yang tidak relevan dan tidak merefleksikan isi dokumen. *Stopword* tidak bermanfaat dalam proses perolehan informasi karena seringnya muncul dalam sebuah koleksi dokumen, *stopword* tidak dapat dijadikan kata-kata pembeda antar dokumen. Untuk itu harus dilakukan penghapusan *stopword*, hal ini dilakukan untuk memilih kata yang dianggap penting dan menggambarkan isi dokumen.

Proses ini diawali dengan membentuk sebuah daftar kata yang berisi *stopword*. Kemudian dilanjutkan dengan melakukan pengecekan apakah kata ada dalam `daftarStopwords`, setelah itu dilanjutkan dengan proses pemampatan kata. Proses penghapusan *stopwords* ditunjukkan pada *Sourcecode* 4.6.

```
private void Stopword()
{
    string filename = "stopword.txt";
    FileStream fs = new FileStream(filename,
    FileMode.Open);
    StreamReader str = new StreamReader(fs);
    daftarStopwords = new ArrayList();
    while (!str.EndOfStream)
    {
        string s = str.ReadLine();
        daftarStopwords.Add(s);
    }
    str.Close();
    fs.Close();

    //Buang Stopwords
    for (int i = 0; i < dok.GetLength(0); i++)
    {
        for (int j=0; j < dokToken[i].Length; j++)
        {
            string sKata = dokToken[i][j];
            for (int k = 0; k <
            daftarStopwords.Count; k++)
            {
                if (sKata == (string)
                daftarStopwords[k])
                {
                    dokToken[i][j] = "";
                    break;
                }
            }
         }
     }

//Pemampatan kata
     dokStopword = new string[dok.GetLength(0)][];
     for (int i = 0; i < dok.GetLength(0); i++)
     {
          int jumlah = 0;
          for (int j = 0; j < dokToken[i].Length; j++)
          {
              if (dokToken[i][j] != "")
              {
                  jumlah = jumlah + 1;
          }
      }
```

```
//Lanjutan...
        dokStopword[i] = new string[jumlah];
        int index = 0;
        for (int k = 0; k < dokToken[i].Length; k++)
        {
            if (dokToken[i][k] != "")
            {
                dokStopword[i][index] =
                dokToken[i][k];
                index++;
            }
        }
    }
}
```

Sourcecode 4.6 Fungsi *Stopwords*

Pada *Sourcecode* 4.6, `stopword.txt` disimpan dalam variabel `filename` bertipe *string*. Kemudian, dibaca tiap baris dan disimpan dalam variabel *string* bernama `s`, `s` ditambahkan dalam variabel dengan tipe `arrayList` bernama `daftarStopwords`.

Selanjutnya, dilakukan perulangan untuk sejumlah `dok` dan untuk sejumlah `dokToken`, ambil dari `dokToken[i][j]` dan simpan dalam variabel `string` bernama `sKata`. Selanjutnya, lakukan perulangan untuk sejumlah `daftarStopwords`. Jika ditemukan *string* yang sama maka isi dari `dokToken` akan diganti menjadi *string* kosong dengan perintah `dokToken[i][j] = ""` dimana *i* dan *j* merupakan indeks dari *string* yang akan dijadikan *string* kosong. Setelah itu, akan dilakukan pemampatan kata.

4.2.2.5 *Stemming*

Proses *stemming* bertujuan untuk mencari *root* kata dari tiap kata hasil dari penghapusan *stopwords.* Proses ini akan menghilangkan imbuhan dalam bahasa Indonesia yang terdiri dari awalan *(prefiks)* dan akhiran *(sufiks).* Tahap pertama adalah menangani partikel kata yang berakhiran kah, lah, dan tah. Potongan *sourcecode* untuk menangani akhiran kah, lah, dan tah ditunjukkan pada Sourcecode 4.7.

```
//Stemming untuk akhiran kah, tah, dan lah
if (bufStr.EndsWith("kah") || bufStr.EndsWith("tah")
|| bufStr.EndsWith("lah"))
{
    if (countVocal(bufStr) > 2 )
    {
        if (bufStr != "pemerintah")
        {
            lenStr = bufStr.Length - 3;
            s = bufStr.Substring(0, lenStr);
            kata = s;
        }
        else kata = bufStr;
     }
}
bufStr = kata;
```

Sourcecode 4.7 Kode Program lah, kah, dan tah

Tahap kedua adalah menangani partikel kata yang berakhiran ku, mu, nya. Potongan *sourcecode* untuk menangani akhiran ku, mu, an nya ditunjukkan pada *sourcecode* 4.8.

```
//stemming ku, mu
if (bufStr.EndsWith("ku") || bufStr.EndsWith("mu"))
{
    if (countVocal(bufStr) > 2)
    {
        lenStr = bufStr.Length - 2;
        s = bufStr.Substring(0, lenStr);
        kata = s;
    }
}
```

```
//Lanjutan...
//stemming -nya
if (bufStr.EndsWith("nya"))
{
    lenStr = bufStr.Length - 3;
    s = bufStr.Substring(0, lenStr);
    kata = s;
}
```

Sourcecode 4.8 Kode Program ku, mu, dan nya

Tahap ketiga adalah menangani awalan me, pen, dan pem serta di, ter, ke. Potongan *sourcecode* yang ditampilkan hanya untuk menangani kata yang berawalan me, potongan ditunjukkan pada *sourcecode* 4.9.

```
//stemming kata berawalan m-
if (bufStr.StartsWith("m"))
{
    if (bufStr.StartsWith("meny"))
    {
        lenStr = bufStr.Length - 4;
        s = bufStr.Substring(4, lenStr);
        s = 's' + bufStr.Substring(4, lenStr);
        kata = s;
    }
    else if (bufStr.StartsWith("meng"))
    {
        lenStr = bufStr.Length - 4;
        s = bufStr.Substring(4, lenStr);
        kata = s;
    }
    else if (bufStr.StartsWith("men"))
    {
        lenStr = bufStr.Length - 3;
        s = bufStr.Substring(3, lenStr);
        bool ada = true;
        for (int j = 0; j < vocal.Length; j++)
        {
            if (s[0] == vocal[j])
            ada = false;
        }
```

```
//Lanjutan stemming kata berawalan m
        if (ada) s = bufStr;
        else s = 't' + bufStr.Substring(3,
        lenStr);
        kata = s;
    }
    else if (bufStr.StartsWith("mem"))
    {
        lenStr = bufStr.Length - 3;
        s = bufStr.Substring(3, lenStr);
        for (int j = 0; j < vocal.Length; j++)
        {
            if (s[0] == vocal[j])
                s = 'p' + bufStr.Substring(3,
                lenStr);
    }
    else if (bufStr.StartsWith("me"))
    {
        lenStr = bufStr.Length - 2;
        s = bufStr.Substring(2, lenStr);
        kata = s;
    }
}
```

Sourcecode 4.9 Kode Program Awalan Kata me

Tahap keempat adalah menangani awalan be dan pe. Potongan *sourcecode* untuk menangani awalan be ditunjukkan pada *sourcecode* 4.10.

```
//stemming kata berawalan be-
if (bufStr.StartsWith("b"))
{
    if (bufStr.StartsWith("ber"))
    {
        lenStr = bufStr.Length - 3;
        s = bufStr.Substring(3, lenStr);
        kata = s;
    }
    else if (bufStr.IndexOf("ajar") >= 0)
    {
        lenStr = bufStr.Length - 3;
        s = bufStr.Substring(3, lenStr);
        kata = s;
    }
```

```
//lanjutan...
    else if (bufStr.IndexOf("ajar") >= 0)
    {
        lenStr = bufStr.Length - 3;
        s = bufStr.Substring(3, lenStr);
        kata = s;
    }
    else if (bufStr.IndexOf("kerja") >= 0)
    {
        lenStr = bufStr.Length - 2;
        s = bufStr.Substring(2, lenStr);
        kata = s;
    }
}
```

Sourcecode 4.10 Kode Program Awalan Kata be

Tahap kelima adalah menangani akhiran kan, an, dan i. Potongan *sourcecode* untuk menangani akhiran kan, an, dan i ditunjukkan pada *sourcecode* 4.11.

```
//stemming kata berakhiran kan, an dan i
if ((bufStr.EndsWith("kan") &&
(!(bufStr.StartsWith("ke")) ||
!(bufStr.StartsWith("peng")))))
{
    lenStr = bufStr.Length - 3;
    s = bufStr.Substring(0, lenStr);
    kata = s;
}
else if (bufStr.EndsWith("an") &&
(!(bufStr.StartsWith("di")) ||
!(bufStr.StartsWith("meng"))
                    || !(bufStr.StartsWith("ter"))))
{
    lenStr = bufStr.Length - 2;
    s = bufStr.Substring(0, lenStr);
    kata = s;
}
```

```
//Lanjutan...
else if (bufStr.EndsWith("i") &&
((bufStr.StartsWith("ber")) ||
(bufStr.StartsWith("ke"))
                    || (bufStr.StartsWith("meng"))))
{
    lenStr = bufStr.Length - 1;
    s = bufStr.Substring(0, lenStr);
    kata = s;
}
```

Sourcecode 4.11 Kode Program Akhiran kan, an, dan i

4.2.2.6 Implementasi Perhitungan Bobot

Proses perhitungan bobot merupakan tahap *preprocessing* sebelum masuk tahap *clustering*. Adapun rumus dari *TF-IDF* sudah dijelaskan pada bab 2 dengan menggunakan persamaan 2.1.

Awalnya, akan dilakukan proses perhitungan atau menghitung jumlah kemunculan tiap kata yang terkandung dari setiap dokumen. Daftar kata dasar yang diperoleh sebelumnya dari proses *stemming* digunakan sebagai parameter input. Untuk membentuk *termlist*, akan ditunjukkan pada sourcecode 4.2.

```
private void DaftarKata()
{
    termList = new ArrayList();
    for (int i = 0; i < dokStemming.GetLength(0);
    i++ )
    {
        for (int j = 0; j < dokStemming[i].Length;
        j++)
        {
            string Term = dokStemming[i][j];
            if (Term.Length < 3)
            {
                break;
            }
            bool ada = true;
```

```
//Lanjutan...
            for (int k = 0; k < termList.Count; k++)
            {
                string s = (string)termList[k];
                if (s == Term)
                {
                    ada = false;
                    break;
                }
            }
            if (ada)
            {
                termList.Add(Term);
            }
        }
    }
}
```

Sourcecode 4.12 Fungsi DaftarKata

Sourcecode 4.12 menunjukkan fungsi yang bernama `DaftarKata`, yang berfungsi untuk menyimpan `termlist` (daftar kata) dari dokumen. Hasil akhir dari proses ini adalah sebuah daftar kata. Setelah menyusun daftar kata, maka akan dilakukan perhitungan frekuensi kemunculan kata dalam dokumen. Sedangkan untuk mengitung frekuensi kemunculan kata yang menyusun dokumen akan ditunjukkan pada *sourcecode* 4.13.

```
private void TermFrek()
{
    termArray = new string[termList.Count];
    TF = new string[termList.Count][][];
    for (int i = 0; i < termList.Count; i++)
    {
        termArray[i] = (string)termList[i];
    }
    for (int j = 0; j < termList.Count; j++)
    {
        string[][] temp;
        temp = new string[dok.GetLength(0)][];
```

```
//Lanjutan...
        for (int k = 0; k < dok.GetLength(0); k++)
        {
            temp[k] = new string[3];
            int jum = 0;
            for (int l = 0; l <
            dokStemming[k].Length; l++)
            {
                if (dokStemming[k][l] ==
                termArray[j])
                {
                    jum++;
                }
            }
            temp[k][0] = termArray[j];
            temp[k][1] = dok[k,0];
            temp[k][2] = jum.ToString();
    }
    TF[j] = temp;
}
```

Sourcecode 4.13 Fungsi *Term*Frek

Sourcecode 4.13 menunjukkan fungsi yang bernama `TermFrek`, yaitu untuk menampilkan kata-kata yang menyusun dokumen beserta frekuensinya. Kata-kata beserta frekuensinya diimplementasikan dalam bentuk variabel `TF`.

Selanjutnya, dilakukan perhitungan *IDF* dan perhitungan nilai bobot *(TF-IDF). Sourcecode* untuk perhitungan *IDF* ditunjukkan pada *sourcecode* 4.14.

```
private void HitungIDF()
{
    IDF = new string[termArray.Length][];
    for (int i = 0; i < termArray.Length; i++)
    {
        IDF[i] = new string[3];
        int jum = 0;
```

```
//Lanjutan...
        for (int j = 0; j < dok.GetLength(0); j++)
        {
            if (Convert.ToInt32(TF[i][j][2]) > 0)
            {
                jum++;
            }
        }
        IDF[i][0] = termArray[i];
        IDF[i][1] = jum.ToString();
        double HitIDF =
        Math.Log((Convert.ToDouble(dok.GetLength(0))
        / Convert.ToDouble(IDF[i][1])), 10);
        IDF[i][2] = HitIDF.ToString();
    }
}
```

Sourcecode 4.14 Fungsi Hitung*IDF*

Pada *Sourcecode* 4.14, pertama akan dilakukan instanisasi `IDF` dalam sejumlah `termArray`. Kemudian dilakukan perulangan untuk sejumlah `termArray` dan selanjutnya sejumlah `dok`, apabila jumlah frekuensi lebih dari 0 maka `jum++` (jumlah meningkat). Berikutnya, mengisi dimensi *jagged array* `IDF` ke-i yang ke-0 dengan `termArray i` dan dimensi *jagged array* `IDF` ke-i yang ke-1 dengan `jum` (yang telah dikonversi ke *string*). Untuk menghitung `IDF` digunakan fungsi `Math.Log()`. Selanjutnya, mengisi dimensi *jagged array* `IDF` ke-i yang ke-2 dengan nilai hasil perhitungan `HitIDF` (yang telah dikonversi ke *string*).

Berikutnya, *sourcecode* untuk menghitung bobot ditunjukkan pada *sourcecode* 4.15.

```
private void HitungBobot()
{
    Bobot = new string[termList.Count][][];
    for (int i = 0; i < termList.Count; i++)
    {
        Bobot[i] = new string[dok.GetLength(0)][];
        for (int j = 0; j < dok.GetLength(0); j++)
        {
            Bobot[i][j] = new string[3];
            Bobot[i][j][0] = termArray[i];
            Bobot[i][j][1] = dok[j, 0];
            double bobotTfIdf =
            Convert.ToDouble(TF[i][j][2]) *
            Convert.ToDouble(IDF[i][2]);
            Bobot[i][j][2] = bobotTfIdf.ToString();
        }
    }
}
```

Sourcecode 4.15 Fungsi HitungBobot

Pada *Sourcecode* 4.15, pertama akan dilakukan instanisasi `Bobot` untuk sejumlah `termList`. Kemudian dilakukan perulangan untuk sejumlah `termList`, instanisasi `Bobot` ke `i` untuk sejumlah `dok` dan lakukan perulangan sejumlah `dok`. Berikutnya, instanisasi `Bobot` dimensi ke-3, untuk dimensi yang ke-0 akan diisi dengan `termArray` ke `i`, untuk dimensi yang ke-1 akan diisi dengan `dok` ke `j,0` dan untuk dimensi yang ke-2 dari `Bobot` akan diisi dengan hasil perhitungan `Bobot` (yang telah dikonversi ke *string*).

4.2.3 Implementasi *Clustering* menggunakan *DBSCAN*

Tujuan dari proses ini adalah untuk membentuk *cluster* dengan ukuran dan *density* yang minimal. Tahap *DBSCAN*, dimulai dengan deklarasi `nilaiPoints (sejumlah point)` yang akan digunakan dalam proses. Fungsi `SetOfPoints()` akan digunakan untuk menampung `nilaiPoints` yang akan diproses. *Sourcecode*nya akan ditunjukkan pada *sourcecode* 4.16.

```
private void SetOfPoints()
{
    nilaiPoints = new ArrayList();
    for (int i = 0; i < dok.GetLength(0); i++)
    {
        d = new double[termList.Count];
        for (int j = 0; j < termList.Count; j++)
        {
            d[j] = Convert.ToDouble(Bobot[j][i][2]);
        }
        MyPoint point = new MyPoint(d);
        point.NamaDok =
        System.IO.Path.GetFileName(dok[i,
        0]).Replace(".txt", "");
        nilaiPoints.Add(point);
     }
}
```

Sourcecode 4.16 Fungsi *SetOfPoints*

Pada sourcecode 4.16, awalnya akan dideklarasikan tipe data dari `nilaiPoints`. Lakukan perulangan untuk sejumlah `dok`, maka `d` bertipe `double` (ukurannya sejumlah `termlist`). Kemudian dilakukan perulangan untuk sejumlah `temlist`, maka simpan nilai `Bobot` dalam variabel `d[j]`, `point` dalam `MyPoint` diisi dengan nilai `d`. Untuk `point`, nama dokumen akan diambil nama *file*nya saja, sedangkan tipe file **.txt* akan diganti dengan *string* kosong. Selanjutnya, `point` akan ditambahkan dalam `nilaiPoints`.

Untuk tahap *clustering DBSCAN* ditunjukkan pada *sourcecode* 4.17.

```
private int DBSCAN(ArrayList point, double Eps, int
MinPts)
{
    int ClusterId = 1;
    for (int i = 0; i < point.Count; i++)
    {
        MyPoint p = (MyPoint)point[i];
        if (p.ClusterId == 0)
        {
            if (ExpandCluster(point, p, ClusterId,
            Eps, MinPts))
            {
                ClusterId++;
            }
        }
     }
     return ClusterId;
}
```

Sourcecode 4.17 Fungsi *DBSCAN*

Pada *sourcecode* 4.17, digunakan parameter `point` bertipe `ArrayList`, `Eps` bertipe `double`, dan `MinPts` bertipe `int`. Awalnya `ClusterId` diinisialisasi dengan nilai 1. Lakukan perulangan untuk sejumlah `point`. Simpan `point` ke i dari `MyPoint` dengan nama `p`. Kemudian lakukan pengecekan apakah `p` dalam `ClusterId` bukan nol (belum ter*cluster*), kemudian apakah `ExpandCluster` maka `ClusterId` akan meningkat.

Selanjutnya, fungsi yang digunakan dalam *DBSCAN*, yaitu fungsi *expandCluster*. *Sourcecode* untuk *expandCluster* akan ditunjukkan pada *sourcecode* 4.18.

```
private bool ExpandCluster(ArrayList point, MyPoint
P, int ClusterId, double Eps, int MinPts)
{
    ArrayList seeds = regionQuery(point, P, Eps);
    ArrayList tempSeeds = new ArrayList();
    if (seeds.Count < MinPts)
    {
        P.ClusterId = -1;
        return false;
    }
```

```
//Lanjutan...
    else
    {
        for (int i = 0; i < seeds.Count; i++)
        {
            MyPoint tempPoint = (MyPoint)seeds[i];
            tempPoint.ClusterId = ClusterId;
            tempSeeds.Add(tempPoint);
        }
        seeds = tempSeeds;
        seeds.Remove(P);
        while (seeds.Count > 0)
        {
            MyPoint currentP = (MyPoint)seeds[0];
            ArrayList result = regionQuery(point,
            currentP, Eps);
            int corePoint = 0;
            for (int i = 0; i < result.Count; i++)
            {
                MyPoint mp = (MyPoint)result[i];
                for (int j = 0; j < tempSeeds.Count;
                j++)
                {
                    MyPoint mpCore =
                    (MyPoint)tempSeeds[j];
                    if (mp.NamaDok == mpCore.NamaDok)
                    {
                        corePoint++;
                    }
                }
            }
            if (corePoint >= MinPts)
            {
                for (int i = 0; i < result.Count;
                i++)
                {
                    MyPoint tempP =
                    (MyPoint)result[i];
                    if (tempP.ClusterId == 0 ||
                    tempP.ClusterId == -1)
                    {
                        if (tempP.ClusterId == 0)
                        {
                           tempP.ClusterId =
                           ClusterId;
                           seeds.Add(tempP);
                        }
```

```
//Lanjutan...
                        }
                    }
                }
                seeds.Remove(currentP);
            }
            return true;
        }
    }
```

Sourcecode 4.18 Fungsi *ExpandCluster*

Pada *sourcecode* 4.18, digunakan parameter `point` bertipe `ArrayList`, `P` dari `MyPoint`, `ClusterId` bertipe `int`, `Eps` bertipe `double`, dan `MinPts` bertipe `int`. Fungsi `return false` terjadi apabila jumlahnya `seeds` kurang dari `minPts` sehingga `P` dalam `ClusterId` bernilai -1 (atau dianggap sebagai *noise*). *Neighbour* dari tiap *point* akan dihitung. Apabila tidak ada *neighbour*, maka fungsi berakhir.

Region query merupakan proses dalam pencarian *neighbour* dari tiap *point*. *Region query* dicari berdasarkan nilai similaritas antar *point*. *Sourcecode* untuk *region query* ditunjukkan pada *sourcecode* 4.19.

```
private ArrayList regionQuery(ArrayList point,
MyPoint P, double Eps)
{
        ArrayList seeds = new ArrayList();
        for (int i = 0; i < nilaiPoints.Count; i++)
        {
                MyPoint tempPoint =
                (MyPoint)nilaiPoints[i];
                Double tempCosine =
                cosine(tempPoint.Vektor, P.Vektor);
                if ((tempCosine >= Eps))
                {
                        seeds.Add(tempPoint);
                }
        }
        return seeds;
}
```

Sourcecode 4.19 Fungsi *regionQuery*

Pada *Sourcecode* 4.19, dapat dilihat bahwa `tempPoint` diperoleh dari `nilaiPoints` ke `i`, sedangkan `tempCosine` didapat dari hasil `cosine` yang berupa vektor. Apakah nilai dari `tempCosine` lebih besar sama dengan nilai parameter `eps` maka `tempPoint` akan ditambahkan dalam `seeds`.

Untuk tahap perhitungan nilai similaritas menggunakan *cosine similarity* (persamaan 2.2), akan ditunjukkkan pada *sourcecode* 4.20 berikut.

```
private Double cosine(Double[] a, Double[] b)
{
        double dp = 0, aa = 0, bb = 0;
        for (int i = 0; i < a.Length; i++)
        {
                dp += a[i] * b[i];
                aa += a[i] * a[i];
                bb += b[i] * b[i];
        }
        aa = Math.Sqrt(aa);
        bb = Math.Sqrt(bb);
        return dp / (aa * bb);
}
```

Sourcecode 4.20 Fungsi *cosine*

Perhitungan jarak antar dokumen dilakukan dengan menggunakan perhitungan *cosine similarity* disesuaikan dengan rumus yang telah dijelaskan pada bab 2. Keluarannya adalah nilai similaritas antar dokumen. Jadi, jika terdapat sepuluh dokumen masukan maka akan menghasilkan sepuluh nilai kemiripan.

4.3 Implementasi Antar Muka

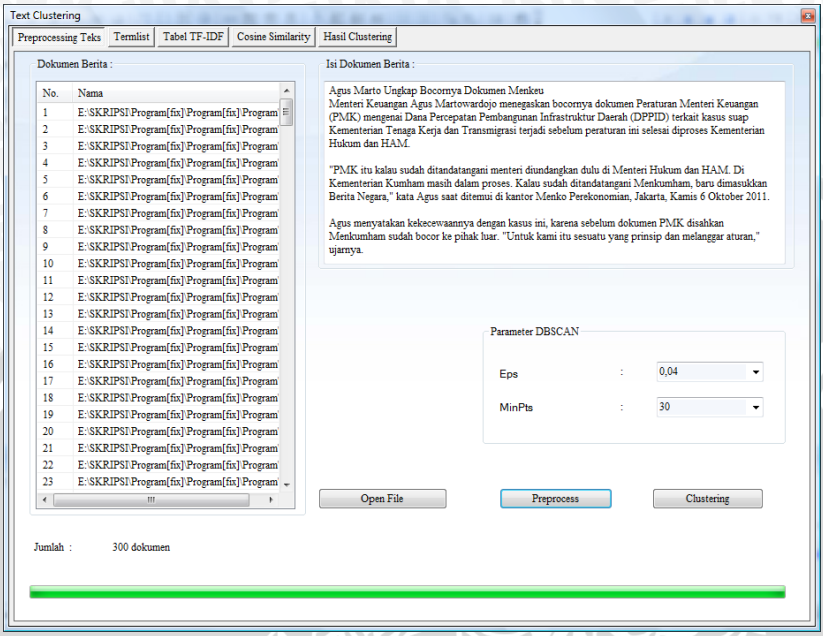
Berdasarkan perancangan antar muka pada sub bab 3.4, maka dihasilkan 2 *form* yaitu *form* awal dan *form clustering*. *Form* awal hanya berfungsi sebagai tampilan saja dan ditunjukkan pada gambar 4.1. *Form clustering* terdiri dari 5 *tab* yaitu *tab Preprocessing*, *tab Termlist*, *tab Tabel TF-IDF*, *tab Cosine Similarity*, dan *tab Hasil Clustering*.

Gambar 4.1 *Form* Awal

Pada *form clustering*, terdapat tab pertama yaitu *tab Preprocessing*, berfungsi untuk memilih dokumen-dokumen yang akan digunakan dalam proses. *Tab Preprocessing* ditunjukkan pada gambar 4.2.

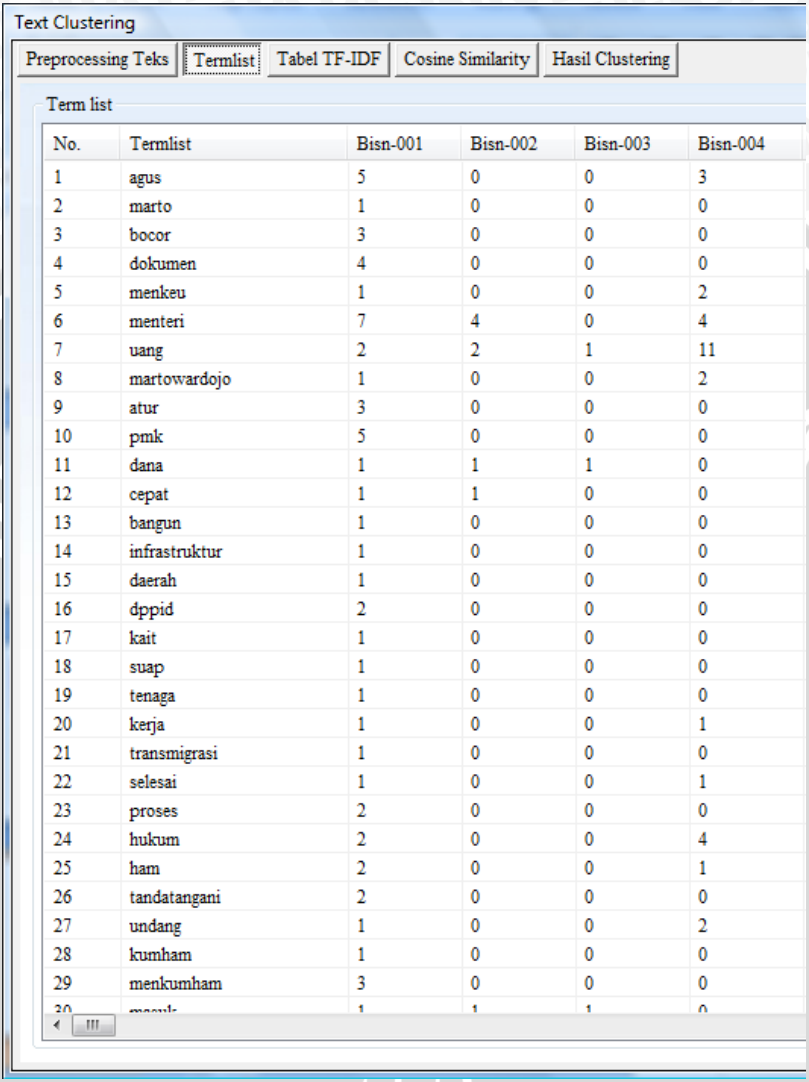
Pada gambar 4.2 terdapat *button Open File* berfungsi untuk menampilkan *folder* yang berisi dokumen berita. *Listview* berisi dokumen *input* dan isi dari dokumen akan ditampilkan pada *textbox* isi dokumen. *Button preprocess* digunakan untuk mulai melakukan preproses pada dokumen *input*. Hasilnya berupa *termlist* beserta frekuensinya, hasil perhitungan bobot, dan nilai similaritas yang masing-masing ditunjukkan pada *tab Termlist*, *tab* Tabel *TF-IDF*, dan *tab Cosine Similarity*.

Tab Termlist, dijelaskan pada Gambar 4.3 dimana *tab* tersebut akan menampilkan *term* (kata) dan frekuensi dari masing-masing kata. Untuk *tab* Tabel *TF-IDF* akan dijelaskan pada Gambar 4.4 dimana bagian ini berfungsi untuk menampilkan hasil perhitungan bobot kata dalam bentuk vektor. Sedangkan *tab Cosine similarity*, ditunjukkan pada Gambar 4.5 yang berfungsi untuk menampilkan hasil perhitungan dari nilai similaritas antar dokumen.



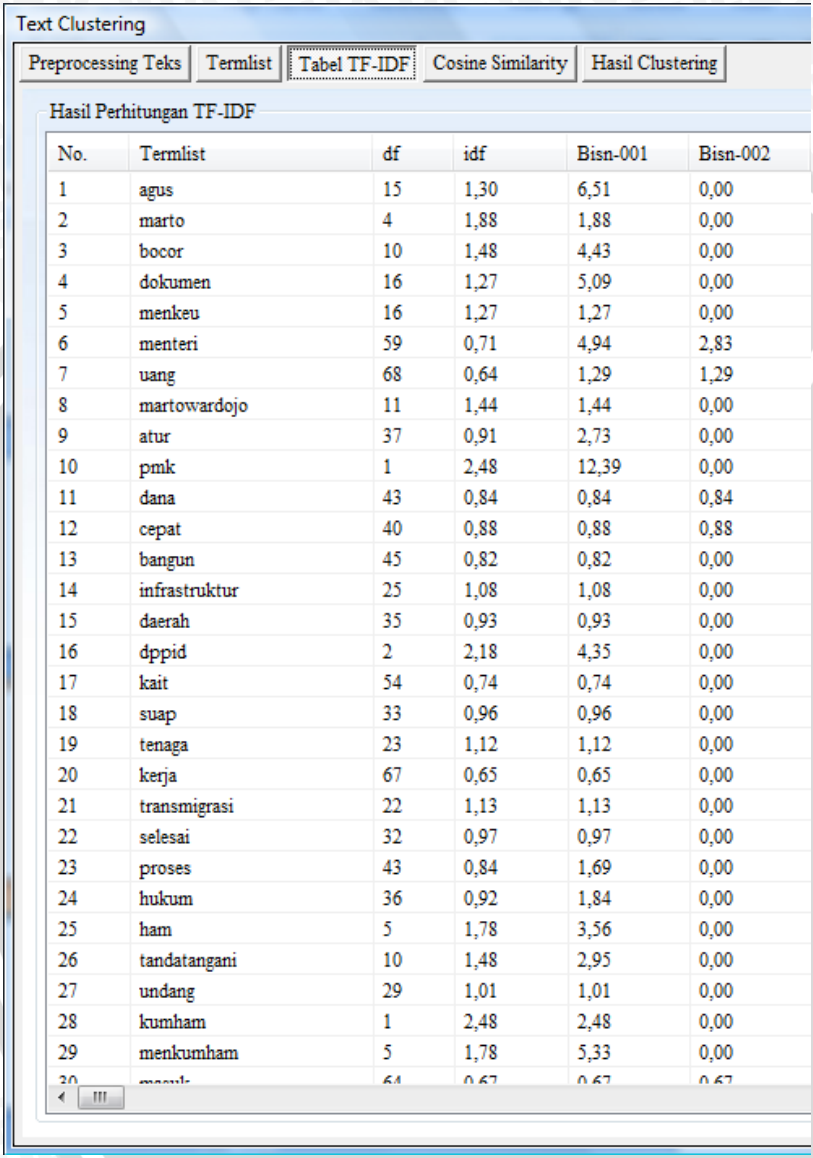
Gambar 4.2 Tab *Preprocessing* Teks

Group box yang bernama parameter DBSCAN pada gambar 4.2, digunakan untuk mengisi nilai parameter *eps* dan parameter *minPts.* Nilai dari parameter *eps* ditampilkan pada *combobox* pertama dengan nama cb_eps, sedangkan nilai dari parameter *minPts* ditampilkan pada *combobox* kedua dengan nama cb_minPts.



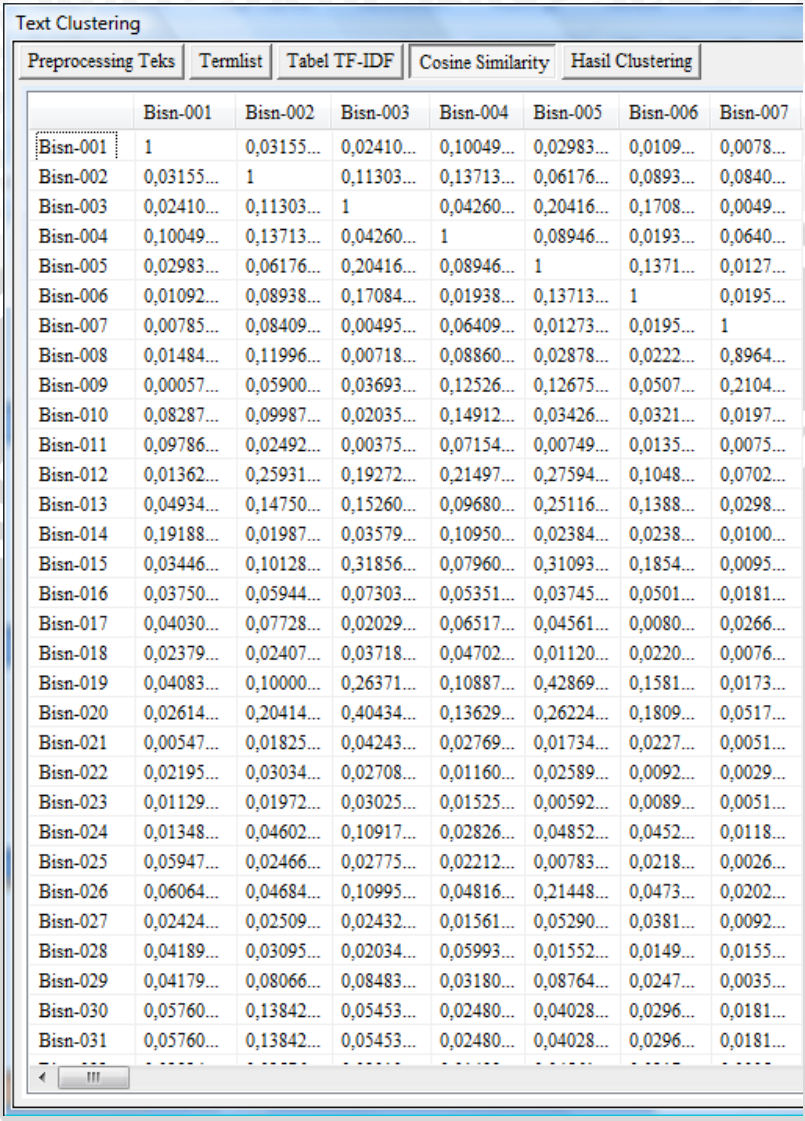
No.	Termlist	Bisn-001	Bisn-002	Bisn-003	Bisn-004
1	agus	5	0	0	3
2	marto	1	0	0	0
3	bocor	3	0	0	0
4	dokumen	4	0	0	0
5	menkeu	1	0	0	2
6	menteri	7	4	0	4
7	uang	2	2	1	11
8	martowardojo	1	0	0	2
9	atur	3	0	0	0
10	pmk	5	0	0	0
11	dana	1	1	1	0
12	cepat	1	1	0	0
13	bangun	1	0	0	0
14	infrastruktur	1	0	0	0
15	daerah	1	0	0	0
16	dppid	2	0	0	0
17	kait	1	0	0	0
18	suap	1	0	0	0
19	tenaga	1	0	0	0
20	kerja	1	0	0	1
21	transmigrasi	1	0	0	0
22	selesai	1	0	0	1
23	proses	2	0	0	0
24	hukum	2	0	0	4
25	ham	2	0	0	1
26	tandatangani	2	0	0	0
27	undang	1	0	0	2
28	kumham	1	0	0	0
29	menkumham	3	0	0	0
30	masuk	1	1	1	0

Gambar 4.3 *Tab Termlist*



No.	Termlist	df	idf	Bisn-001	Bisn-002
1	agus	15	1,30	6,51	0,00
2	marto	4	1,88	1,88	0,00
3	bocor	10	1,48	4,43	0,00
4	dokumen	16	1,27	5,09	0,00
5	menkeu	16	1,27	1,27	0,00
6	menteri	59	0,71	4,94	2,83
7	uang	68	0,64	1,29	1,29
8	martowardojo	11	1,44	1,44	0,00
9	atur	37	0,91	2,73	0,00
10	pmk	1	2,48	12,39	0,00
11	dana	43	0,84	0,84	0,84
12	cepat	40	0,88	0,88	0,88
13	bangun	45	0,82	0,82	0,00
14	infrastruktur	25	1,08	1,08	0,00
15	daerah	35	0,93	0,93	0,00
16	dppid	2	2,18	4,35	0,00
17	kait	54	0,74	0,74	0,00
18	suap	33	0,96	0,96	0,00
19	tenaga	23	1,12	1,12	0,00
20	kerja	67	0,65	0,65	0,00
21	transmigrasi	22	1,13	1,13	0,00
22	selesai	32	0,97	0,97	0,00
23	proses	43	0,84	1,69	0,00
24	hukum	36	0,92	1,84	0,00
25	ham	5	1,78	3,56	0,00
26	tandatangani	10	1,48	2,95	0,00
27	undang	29	1,01	1,01	0,00
28	kumham	1	2,48	2,48	0,00
29	menkumham	5	1,78	5,33	0,00
30	masuk	64	0,67	0,67	0,67

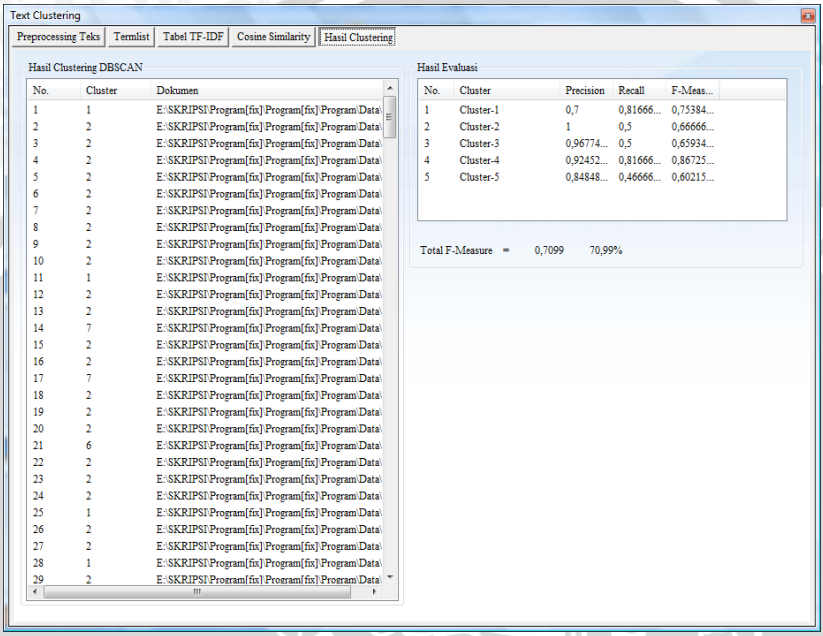
Gambar 4.4 *Tab* Tabel *TF-IDF*



	Bisn-001	Bisn-002	Bisn-003	Bisn-004	Bisn-005	Bisn-006	Bisn-007
Bisn-001	1	0,03155...	0,02410...	0,10049...	0,02983...	0,0109...	0,0078...
Bisn-002	0,03155...	1	0,11303...	0,13713...	0,06176...	0,0893...	0,0840...
Bisn-003	0,02410...	0,11303...	1	0,04260...	0,20416...	0,1708...	0,0049...
Bisn-004	0,10049...	0,13713...	0,04260...	1	0,08946...	0,0193...	0,0640...
Bisn-005	0,02983...	0,06176...	0,20416...	0,08946...	1	0,1371...	0,0127...
Bisn-006	0,01092...	0,08938...	0,17084...	0,01938...	0,13713...	1	0,0195...
Bisn-007	0,00785...	0,08409...	0,00495...	0,06409...	0,01273...	0,0195...	1
Bisn-008	0,01484...	0,11996...	0,00718...	0,08860...	0,02878...	0,0222...	0,8964...
Bisn-009	0,00057...	0,05900...	0,03693...	0,12526...	0,12675...	0,0507...	0,2104...
Bisn-010	0,08287...	0,09987...	0,02035...	0,14912...	0,03426...	0,0321...	0,0197...
Bisn-011	0,09786...	0,02492...	0,00375...	0,07154...	0,00749...	0,0135...	0,0075...
Bisn-012	0,01362...	0,25931...	0,19272...	0,21497...	0,27594...	0,1048...	0,0702...
Bisn-013	0,04934...	0,14750...	0,15260...	0,09680...	0,25116...	0,1388...	0,0298...
Bisn-014	0,19188...	0,01987...	0,03579...	0,10950...	0,02384...	0,0238...	0,0100...
Bisn-015	0,03446...	0,10128...	0,31856...	0,07960...	0,31093...	0,1854...	0,0095...
Bisn-016	0,03750...	0,05944...	0,07303...	0,05351...	0,03745...	0,0501...	0,0181...
Bisn-017	0,04030...	0,07728...	0,02029...	0,06517...	0,04561...	0,0080...	0,0266...
Bisn-018	0,02379...	0,02407...	0,03718...	0,04702...	0,01120...	0,0220...	0,0076...
Bisn-019	0,04083...	0,10000...	0,26371...	0,10887...	0,42869...	0,1581...	0,0173...
Bisn-020	0,02614...	0,20414...	0,40434...	0,13629...	0,26224...	0,1809...	0,0517...
Bisn-021	0,00547...	0,01825...	0,04243...	0,02769...	0,01734...	0,0227...	0,0051...
Bisn-022	0,02195...	0,03034...	0,02708...	0,01160...	0,02589...	0,0092...	0,0029...
Bisn-023	0,01129...	0,01972...	0,03025...	0,01525...	0,00592...	0,0089...	0,0051...
Bisn-024	0,01348...	0,04602...	0,10917...	0,02826...	0,04852...	0,0452...	0,0118...
Bisn-025	0,05947...	0,02466...	0,02775...	0,02212...	0,00783...	0,0218...	0,0026...
Bisn-026	0,06064...	0,04684...	0,10995...	0,04816...	0,21448...	0,0473...	0,0202...
Bisn-027	0,02424...	0,02509...	0,02432...	0,01561...	0,05290...	0,0381...	0,0092...
Bisn-028	0,04189...	0,03095...	0,02034...	0,05993...	0,01552...	0,0149...	0,0155...
Bisn-029	0,04179...	0,08066...	0,08483...	0,03180...	0,08764...	0,0247...	0,0035...
Bisn-030	0,05760...	0,13842...	0,05453...	0,02480...	0,04028...	0,0296...	0,0181...
Bisn-031	0,05760...	0,13842...	0,05453...	0,02480...	0,04028...	0,0296...	0,0181...

Gambar 4.5 *Tab Cosine Similarity*

Selanjutnya *button Clustering* pada Gambar 4.2, merupakan tombol yang digunakan untuk memulai proses *clustering*. Hasilnya ditunjukkan pada Gambar 4.6 pada *tab* Hasil *Clustering.*



Gambar 4.6 *Tab* Hasil *Clustering*

Hasil *clustering* dapat dilihat pada *listview*, yang menampilkan No., yaitu bagian untuk menampilkan urutan dokumen yang akan ditampilkan dalam proses. *Cluster*, bagian untuk menampilkan id *cluster* dari hasil proses *clustering* yang telah dilakukan. Dokumen, bagian ini untuk menampilkan nama dan lokasi dokumen hasil proses. Sedangkan pada *listview* Hasil Evaluasi, bagian ini untuk menampilkan detail evaluasi (hasil *clustering* menggunakan algoritma *DBSCAN*) menggunakan *F-measure.*

4.4 Implementasi Uji Coba

Pada sub bab ini akan dibahas mengenai implementasi dari metode pengujian yang telah dilakukan oleh sistem dan hasil dari pengujian tersebut.

4.4.1 Skenario Evaluasi

Pada pegujian sistem *clustering* dokumen berita berbahasa Indonesia berdasarkan kategori ini, dokumen berita yang akan di*cluster* diambil dari sebuah situs berita *online* yaitu www.VIVAnews.com sebanyak 300 dokumen. Dokumen berita yang diperoleh kemudian disimpan dengan format **.txt*. Kategori yang diperoleh dari www.VIVAnews.com sebanyak lima macam, yaitu bisnis, bola, kosmo, *sport*, dan nasional. Banyaknya dokumen yang digunakan dalam proses *clustering* ditunjukkan pada tabel 4.1.

Tabel 4.1 Dokumen Percobaan

Kategori	Jumlah Data Uji
Bisnis	60
Bola	60
Kosmo	60
Sport	60
Nasional	60
Jumlah dokumen	**300**

Pada penelitian ini akan dilakukan beberapa kali percobaan terhadap dokumen berita dengan jumlah yang sama akan tetapi menggunakan nilai parameter (*eps* dan *minPts*) yang berbeda-beda. Dari nilai parameter *epsilon* dan *minPts* yang berbeda-beda tersebut akan dibandingkan hasil *F-measure* yang dihasilkan. Nilai parameter yang digunakan adalah :

- *Eps* : 0,01 - 0,09 dengan interval 0,01
- *MinPts* : 10 – 50 dengan interval 10

Karena algoritma *DBSCAN* memerlukan dua input parameter dalam perhitungannya, yaitu *epsilon* (nilai *treshol*d untuk jarak antar *item* yang menjadi dasar pembentukan *neighbourhood* dari suatu titik *item*) dan *minPts* (minimal banyak *item* dalam suatu *cluster*) maka daftar pengujian berdasarkan perubahan parameter *eps* dan *minPts* diatur dengan kombinasi dari kedua parameter tersebut.

4.4.2 Uji Coba dan Analisis

Uji coba terhadap sistem dilakukan untuk mengetahui kinerja sistem yang dibangun. Evaluasi tersebut dilakukan dengan membandingkan hasil *clustering* yang dilakukan oleh sistem dengan kategori berita yang diperoleh dari pengelompokan secara manual.

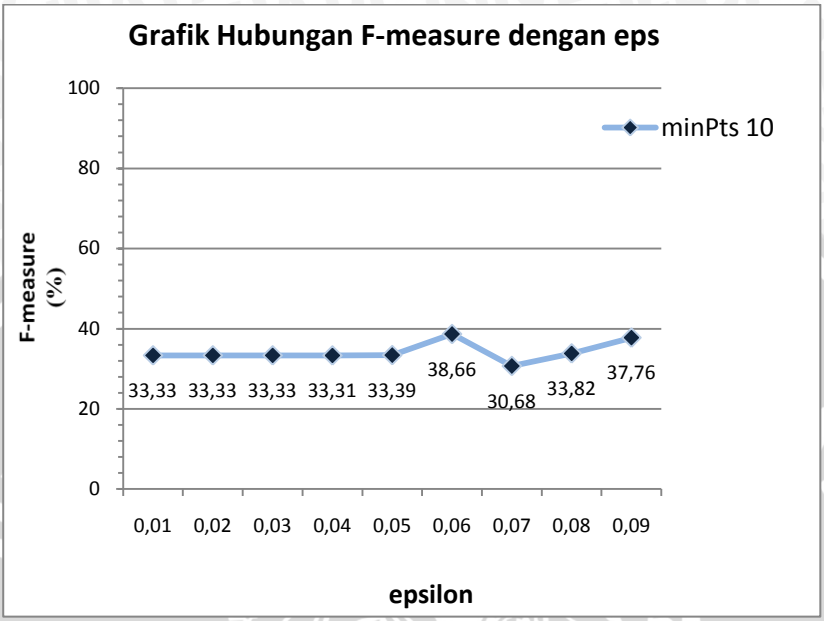
Ada 5 skenario pengujian yang dilakukan untuk mendapatkan nilai *F-Measure* dari parameter yang diberikan. Nilai parameter *eps* yang digunakan adalah sebesar 0,01-0,09 dengan nilai parameter *minPts* seperti yang dijelaskan pada tiap skenario pengujian berikut:

- Skenario 1, dengan nilai *minPts* 10
- Skenario 2, dengan nilai *minPts* 20
- Skenario 3, dengan nilai *minPts* 30
- Skenario 4, dengan nilai *minPts* 40
- Skenario 5, dengan nilai *minPts* 50

Hasil pengujian pada skenario 1 ditunjukkan pada tabel 4.2 dan gambar 4.7.

Tabel 4.2 Hasil Pengujian Skenario 1

Percobaan ke-	Parameter		F-measure (%)
	MinPts	*Eps*	
1	10	0,01	33,33
2	10	0,02	33,33
3	10	0,03	33,33
4	10	0,04	33,31
5	10	0,05	33,39
6	**10**	**0,06**	**38,66**
7	10	0,07	30,68
8	10	0,08	33,82
9	10	0,09	37,76



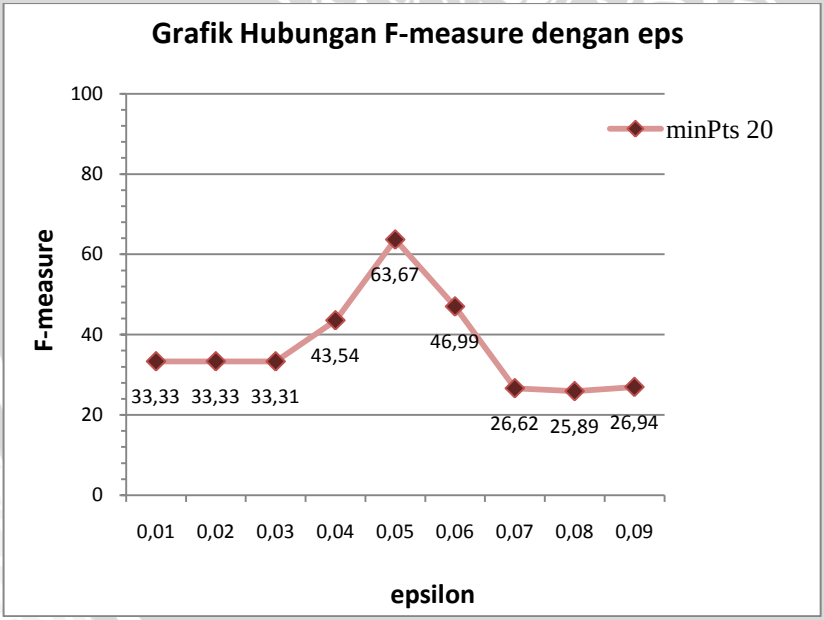
Gambar 4.7 Grafik Hasil Pengujian Skenario 1

Dari tabel 4.2 dan gambar 4.7 dapat dilihat bahwa nilai *overall F-measure* awalnya cenderung stabil, pada *eps* 0,06 mulai meningkat dan diperoleh nilai *overall F-measure* terbaik. Dari pengujian yang dilakukan tersebut, maka didapatkan perubahan nilai *overall F-measure* terhadap perubahan nilai parameter *eps* yang diberikan. Sehingga, setiap nilai parameter dapat mengakibatkan perubahan nilai *overall F-measure* pada hasil pengujian. Pada tahap pencarian *region query* (*eps-neighbourhood*), apabila nilai similaritas dokumen lebih besar sama dengan nilai *eps* maka setiap dua *core point* yang cukup dekat jaraknya satu sama lain akan dimasukkan dalam satu *cluster*. Parameter *eps* memiliki peran yang penting dalam menetapkan jarak minimum dari *point* dalam *cluster* yang sama. Untuk mendapatkan hasil *F-measure* terbaik, maka pemilihan yang benar dan tepat dari nilai parameter perlu memperhatikan nilai similaritas antar *point* (dalam hal ini dokumen). Nilai hasil perhitungan *cosine similarity* dapat dilihat pada lampiran 2.

Selanjutnya, hasil pengujian pada skenario 2 ditunjukkan pada tabel 4.3 dan gambar 4.8.

Tabel 4.3 Hasil Pengujian Skenario 2

Percobaan ke-	Parameter		F-measure (%)
	MinPts	*Eps*	
1	20	0,01	33,33
2	20	0,02	33,33
3	20	0,03	33,31
4	20	0,04	43,54
5	**20**	**0,05**	**63,67**
6	20	0,06	46,99
7	20	0,07	26,62
8	20	0,08	25,89
9	20	0,09	26,94



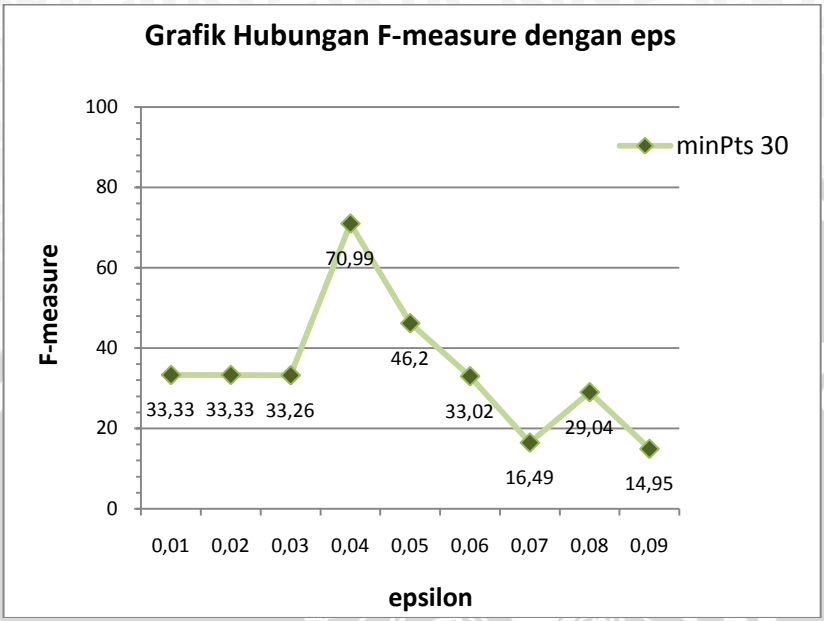
Gambar 4.8 Grafik Hasil Pengujian Skenario 2

Dari tabel 4.3 dan gambar 4.8 dapat dilihat bahwa nilai *overall F-measure* awalnya cenderung stabil, pada *eps* 0,04 mulai meningkat hingga nilai *overall F-measure* terbaik diperoleh ketika *eps* sama dengan 0,05. Kemudian, nilai setelah nilai tertinggi akan turun. Untuk mendapatkan hasil *F-measure* terbaik, pemilihan yang benar dan tepat dari nilai parameter perlu memperhatikan nilai similaritas antar *point* (dalam hal ini dokumen). Nilai hasil perhitungan *cosine similarity* dapat dilihat pada lampiran 2.

Hasil pengujian pada skenario 3 akan ditunjukkan pada tabel 4.4 dan grafik 4.9.

Tabel 4.4 Hasil Pengujian Skenario 3

Percobaan ke-	Parameter		F-measure (%)
	MinPts	*Eps*	
1	30	0,01	33,33
2	30	0,02	33,33
3	30	0,03	33,26
4	**30**	**0,04**	**70,99**
5	30	0,05	46,20
6	30	0,06	33,02
7	30	0,07	16,49
8	30	0,08	29,04
9	30	0,09	14,95



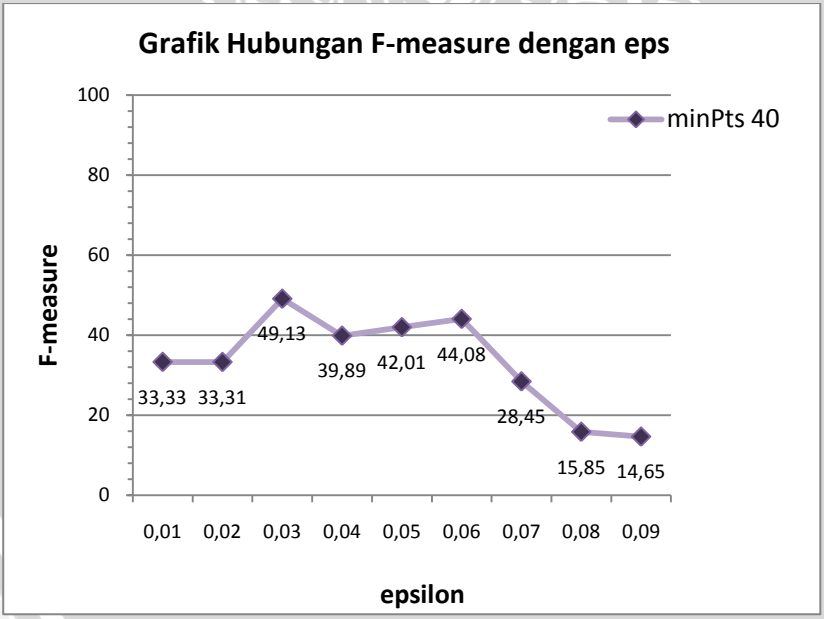
Gambar 4.9 Grafik Hasil Pengujian Skenario 3

Dari tabel 4.4 dan gambar 4.9 dapat dilihat bahwa nilai *overall F-measure* awalnya cenderung stabil, pada *eps* 0,04 naik secara signifikan dan diperoleh nilai *overall F-measure* terbaik atau mencapai puncak. Kemudian, nilai setelah nilai tertinggi akan turun. Untuk mendapatkan hasil *F-measure* terbaik, pemilihan yang benar dan tepat dari nilai parameter perlu memperhatikan nilai similaritas antar *point* (dalam hal ini dokumen). Nilai hasil perhitungan *cosine similarity* dapat dilihat pada lampiran 2.

Selanjutnya, hasil pengujian pada skenario 3 akan ditunjukkan pada tabel 4.5 dan gambar 4.10.

Tabel 4.5 Hasil Pengujian Skenario 4

Percobaan ke-	Parameter		F-measure (%)
	MinPts	*Eps*	
1	40	0,01	33,33
2	40	0,02	33,31
3	**40**	**0,03**	**49,13**
4	40	0,04	39,89
5	40	0,05	42,01
6	40	0,06	44,08
7	40	0,07	28,45
8	40	0,08	15,85
9	40	0,09	14,65



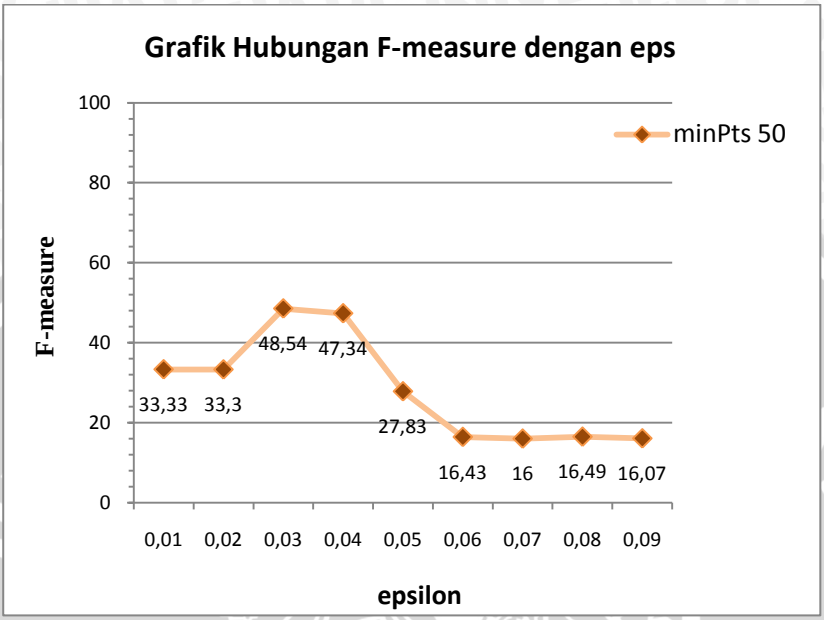
Gambar 4.10 Grafik Hasil Pengujian Skenario 4

Dari tabel 4.5 dan gambar 4.10 dapat dilihat bahwa nilai *overall F-measure* mulai mengalami kenaikan atau meningkat dan didapat nilai *overall F-measure* terbaik atau mencapai puncak pada *eps* 0,03. Kemudian, nilai setelah nilai tertinggi akan turun kemudian naik dan setelah itu turun terus-menerus. Untuk mendapatkan hasil *F-measure* terbaik, pemilihan yang benar dan tepat dari nilai parameter perlu memperhatikan nilai similaritas antar *point* (dalam hal ini dokumen). Nilai hasil perhitungan *cosine similarity* dapat dilihat pada lampiran 2.

Berikutnya, hasil pengujian pada skenario 5 akan ditunjukkan pada tabel 4.6 dan grafik 4.11.

Tabel 4.6 Hasil Pengujian Skenario 5

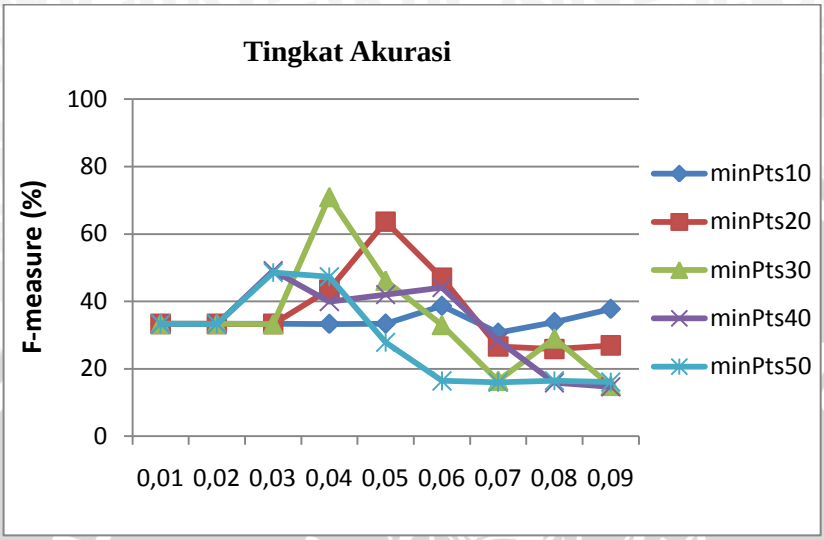
Percobaan ke-	Parameter		F-measure (%)
	MinPts	*Eps*	
1	50	0,01	33,33
2	50	0,02	33,30
3	**50**	**0,03**	**48,54**
4	50	0,04	47,34
5	50	0,05	27,83
6	50	0,06	16,43
7	50	0,07	16,00
8	50	0,08	16,49
9	50	0,09	16,07



Gambar 4.11 Grafik Hasil Pengujian 5

Dari tabel 4.6 dan gambar 4.10 dapat dilihat bahwa nilai *overall F-measure* mulai meningkat dan merupakan *overall F-measure* terbaik pada *eps* 0,03. Selanjutnya, nilai setelah nilai tertinggi akan turun. Untuk mendapatkan hasil *F-measure* terbaik, pemilihan yang benar dan tepat dari nilai parameter perlu memperhatikan nilai similaritas antar *point* (dalam hal ini dokumen). Nilai hasil perhitungan *cosine similarity* dapat dilihat pada lampiran 2.

Dari data hasil pengujian yang dilakukan pada skenario 1 sampai dengan skenario 5, masing-masing nilai *overall F-measure* terbaik dapat dilihat pada gambar 4.12.



Gambar 4.12 Grafik Tingkat Akurasi

Dari grafik 4.12 dapat dilihat bahwa pada skenario tertentu terdapat nilai *overall F-measure* yang tinggi atau mencapai puncak dan nilai *overall F-measure* setelah nilai tertinggi akan turun. Nilai *overall F-measure* terbaik dari skenario pengujian yaitu 0,7099 atau 70.99%. Nilai *overall F-measure* terbaik terjadi saat skenario pengujian ke-3 dengan nilai parameter *eps* 0,04 dan nilai parameter *minPts* 30.

Pada hasil pengujian, terdapat nilai-nilai yang paling berpengaruh terhadap perubahan nilai *F-measure* yaitu paramater *eps* dan *minPts* . Hal ini disebabkan karena parameter *minPts* membatasi ukuran (jumlah) *cluster* dari nilai minimal *cluster* yang sebelumnya. Semakin kecil nilai parameter *minPts* yang digunakan, maka jumlah *cluster* yang dihasilkan akan semakin banyak. *Point* yang memiliki intensitas kecil (atau jaraknya terlalu jauh) akan dieliminasi dari *result*, sehingga hal ini bisa jadi menimbulkan masalah jika dilakukan *clustering* pada data yang terlalu kecil. Sedangkan pada proses *expand cluster*, yaitu pada tahap pencarian *region query* (*eps-neighbourhood* atau dengan kata lain pencarian tetangga terdekat), apabila nilai similaritas dokumen lebih besar sama dengan nilai parameter *eps* maka setiap dua *core point* yang cukup dekat jaraknya satu sama lain akan dimasukkan dalam satu *cluster*. Oleh karena itu,

dalam pengujian sistem *clustering* menggunakan algoritma *DBSCAN*, harus digunakan nilai parameter *minPts* yang tepat dan nilai parameter *eps* yang sesuai dalam menetapkan jarak minimum dari *point* dalam *cluster* yang sama.

BAB V
PENUTUP

a. Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah sebagai berikut :

1. Sistem *clustering* dokumen berita berbahasa Indonesia menggunakan *DBSCAN (Density-Based Spatial Clustering of Application with Noise)* diimplementasikan dengan dua tahap utama, yaitu *preprocessing* dan *clustering*. *Preprocessing* merupakan tahapan untuk mengubah *file* teks berita menjadi vektor dokumen yang berupa hasil similaritas antar dokumen. Pada tahap *clustering DBSCAN* , jumlah *cluster* yang dihasilkan ditentukan oleh sistem dan memperhitungkan *density* tiap *point* terhadap *point* lainnya. Parameter *minPts* membatasi ukuran (jumlah) *point* dalam *cluster* dari nilai minimal kelas. Parameter *minPts* akan menghilangkan *cluster-cluster* yang tidak relevan dan dianggap sebagai *noise*. Sedangkan parameter *eps* juga memiliki peran yang penting dalam menetapkan jarak minimum dari *point* dalam *cluster* yang sama dengan memperhatikan nilai similaritas. Oleh karena itu, perlu menentukan nilai parameter yang tepat untuk mendapatkan hasil terbaik.
2. Tingkat akurasi terbaik dari *clustering* dokumen berita menggunakan algoritma *DBSCAN* adalah sebesar 0,7099 atau 70,99% pada pengujian terhadap 300 dokumen dengan nilai parameter *eps* 0,04 dan *minPts* 30.

b. Saran

Dalam penelitian ini masih banyak hal-hal yang dapat dikembangkan untuk penelitian lebih lanjut, diantaranya :

1. Dalam penelitian hanya menggunakan dataset yang berasal dari satu sumber, manfaatnya akan lebih besar jika digunakan lebih dari satu sumber berita.
2. Algoritma *DBSCAN* memerlukan parameter *eps* dan *minPts* dalam perhitungannya. Nilai parameter ditentukan oleh *user*, sesuai dengan dataset yang digunakan. Inisialisasi awal untuk *eps* dapat dicoba dengan menggunakan *k-dist graph.*

DAFTAR PUSTAKA

Baeza-Yates, Ricardo and Bertheir Ribeiro-Neto. 1999. *Modern Information Retrieval* . Addison-Wesley.

Budiman, K. 2005. *Dasar-Dasar Jurnalistik***.** Pelatihan Jurnalistik – info jawa 12-15 desember 2005. www.infojawa.org. tanggal akses: 19 April 2011.

Dehuri, S., Mohapatra, C., Ghosh, A., dan Mall, 2006. *A Comparative Study of Clustering Algorithm.* http://docsdrive.com/pdfs/ansinet/itj/2006/551-559.pdf. tanggal akses: 21 April 2011

Ernawati, Sari dan Arie Ardiyanti. 2009. Klusterisasi DokumenBerita Berbahasa Indonesia Menggunakan Document IndexGraph.http://journal.uii.ac.id/index.php/Snati/article/viewFile/1286/1095. tanggal akses: 21 April 2011

Even , Yahir dan Zohar. 2002. *Introduction to Text Mining*. Automated Learning Group National Center For Supercomputing Applications. University of illionis. http://algdocs.ncsa.uiuc.edu/PR-20021116-2.ppt. tanggal akses: 20 April 2011

Grira, N., Crucianu M., Boujemaa N, 2005. *Unsupervised and SemiSupervised Clustering: a Brief Survey*. In: 7th ACM SIGMM international workshop on multimedia information retrieval, pp 9-16. http://www-rocq.inria.fr/~crucianu/src/BriefSurveyClustering.pdf. tanggal akses: 21 April 2011

Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K. 2004. *An KNN model-Based Approach and Its Application in Text Categorization*. Laecture Notes in Computer Science Volume 2954, 2004.

Han, Jiawei and Micheline Kamber. 2001. *Data mining : Concepts and Techniqes*. Morgan Kaufmann Publisher: San Fransisco.

Kamus Pusat Bahasa. 2008. *Kamus Bahasa Indonesia.* http://lms.bpkp.go.id/file.php/1/Kamus_Bahasa_Indonesia.pdf. diakses tanggal akses: 17 Mei 2011.

Manning, Christoper D Prabhakar Rasghawan dan Hinrich Schutze. 2009. *An Introduction to Information Retrieval*. Cambridge University Press. http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf. tanggal akses: 24 April 2011

M. Ester, H-P. Kriegel, J. Sander, X. Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, 1996. tanggal akses: 14 Mei 2011.

Mooney, Raymond J. 2006. *Machine Learning Text categorization*. University of Texas: Texas. http://lecturer.eepis-its.edu/~iwanarif/kuliah/dm/6Text%20Mining.pdf. tanggal akses: 14 Mei 2011.

Petridou, Koutsonikola, Vakali dan Papadimtriou. 2000. *A Divergence-Oriented Approach for Web Users Clustering*. Departement of Informatics Aristotle Univeristy:Greece. http://oswinds.csd.auth.gr/papers/2006/iccsa06.pdf. tanggal akses: 14 Mei 2011.

Porter, M. 2006. *The Porter Stemming Algorithm* http://www.comp.lancs.ac.uk/computing/research/stemming/general/porter.htm. tanggal akses: 28 Mei 2011

Ramlan, M. 1995. *Ilmu Bahasa Indonesia : Morfologi Suatu Tinjauan Deskriptif*. CV. Karyono. Yogyakarta.

Satia Budhi, G., Adipranata, R., Sugiarto, M., Anwar, B., Setiahadi, B., 2011. *Pengelompokan Sunspot pada Citra Digital Matahari Menggunakan Metode Clustering DBSCAN*. http://www.google.co.id/PengelompokanBintikMatahariMenggunakanDBSCAN(Greg-Rudy).doc. tanggal akses: 21 April 2011

Shaban, Khaled B. 2009. *A Semantic Approach for Document Clustering. Departement of Computer Science and Engineering, Qatar University.*

Steinbach, M., Karypis, G., dan Kumar, V., 2000. *A Comparison of Document Clustering Techniques*. Technical Report. Departement of Computer Science and Engineering:University of Minnesota.

Tala, Fadillah Z, 2003. *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*. Master of Logic Project. Institute for Logic, Language and Computation. Universiteit van Amsterdam. The Netherlands. www.illc.uva.nl/Publications/ResearchReports/MoL-200302.text.pdf. tanggal akses: 5 April 2011

Triawati, Candra. 2009. *Metode Pembobotan Statistical Concept Based untuk Klastering dan Kategorisasi Dokumen Berbahasa Indonesia*. Institut Teknologi Telkom. Bandung.

Wahyudi, JB. 2002. *Dasar-dasar jurnalistik radio dan televise.* Perpustakaan Utan Kayu. Jakarta.

Lampiran 1
Daftar *Stopwords* Bahasa Indonesia

ada	awal	benar
adalah	awalnya	benarkah
adanya	bagai	benarlah
adapun	bagaikan	berada
agak	bagaimana	berakhir
agaknya	bagaimanakah	berakhirlah
agar	bagaimanapun	berakhirnya
akan	bagi	berapa
akankah	bagian	berapakah
akhir	bahkan	berapalah
akhiri	bahwa	berapapun
akhirnya	bahwasanya	berarti
aku	baik	berawal
akulah	bakal	berbagai
amat	bakalan	berdatangan
amatlah	balik	beri
anda	banyak	berikan
andalaha	bapak	berikut
antar	baru	berikutnya
antara	bawah	berjumlah
antaranya	beberapa	berkali
apa	begini	berkata
apaan	beginian	berkehendak
apabila	beginikah	berkeinginan
apakah	beginilah	berkenaan
apalagi	begitu	berlainan
apatah	begitukah	berlalu
artinya	begitulah	berlangsung
asal	begitupun	berlebihan
asalkan	bekerja	bermaksud
atas	belakang	bermula
atau	belakangan	bersama
ataukah	belum	bersiap
ataupun	belumlah	bertanya

berturut	demi	dilihat
bertutur	demikian	dimaksud
berujar	demikianlah	dimaksudkan
berupa	dengan	dimaksudkannya
besar	depan	dimaksudnya
betul	di	diminta
betulkah	dia	dimintai
biasa	diakhiri	dimisalkan
biasanya	diakhirinya	dimulai
bila	dialah	dimulailah
bilakah	diantara	dimulainya
bisa	diantaranya	dimungkinkan
bisakah	diberi	dini
boleh	diberikan	dipastikan
bolehkah	diberikannya	diperbuat
bolehlah	dibuat	diperbuatnya
buat	dibuatnya	dipergunakan
bukan	didapat	diperkirakan
bukankah	didatangkan	diperlihatkan
bukanlah	digunakan	diperlukan
bukannya	diibaratkan	diperlukannya
bulan	diibaratkannya	dipersoalkan
bung	diingat	dipertanyakan
cara	diingatkan	dipunyai
caranya	diinginkan	diri
cukup	dijawab	dirinya
cukupkah	dijelaskan	disampaikan
cukuplah	dijelaskannya	disebut
cuma	dikarenakan	disebutkan
dahulu	dikatakan	disebutkannya
dalam	dikatakannya	disini
dan	dikerjakan	disinilah
dapat	diketahui	ditambahkan
dari	diketahuinya	ditandaskan
daripada	dikira	ditanya
datang	dilakukan	ditanyai
dekat	dilalui	ditanyakan
ditegaskan	ibarat	kalaupun

ditujukan	ibaratkan	kalian
ditunjuk	ibaratnya	kami
ditunjuki	ibu	kamilah
ditunjukkan	ikut	kamu
ditunjukkannya	ingat	kamulah
ditunjuknya	ingin	kan
dituturkan	inginkah	kapan
dituturkannya	ini	kapankah
diucapkan	inikah	kapanpun
diucapkannya	inilah	karena
diungkapkan	itu	karenanya
dong	itukah	kasus
dua	itulah	kata
dulu	jadi	katakan
empat	jadilah	katakanlah
enggak	jadinya	katanya
enggaknya	jangan	ke
entah	jangankan	keadaan
entahlah	janganlah	kebetulan
guna	jauh	kecil
gunakan	jawab	kedua
hal	jawaban	keduanya
hampir	jawabnya	keinginan
hanya	jelas	kelamaan
hanyalah	jelaskan	kelihatan
hari	jelaslah	kelihatannya
harus	jelasnya	kelima
haruslah	jika	keluar
harusnya	jikalau	kembali
hendak	juga	kemudian
hendaklah	jumlah	kemungkinan
hendaknya	jumlahnya	kemungkinannya
hingga	justru	kenapa
ia	kala	kepada
ialah	kalau	kepadanya
ibarat	kalaulah	kesampaian
keseluruhan	manalagi	menanti

keseluruhannya	masa	menantikan
keterlaluan	masalah	menanya
ketika	masalahnya	menanyai
khususnya	masih	menanyakan
kini	masihkah	mendapat
kinilah	masing	mendapatkan
kira	mau	mendatang
kiranya	maupun	mendatangi
kita	melainkan	mendatangkan
kitalah	melakukan	menegaskan
kok	melalui	mengakhiri
kurang	melihat	mengapa
lagi	melihatnya	mengatakan
lagian	memang	mengatakannya
lah	memastikan	mengenai
lain	memberi	mengerjakan
lainnya	memberikan	mengetahui
lalu	membuat	menggunakan
lama	memerlukan	menghendaki
lamanya	memihak	mengibaratkan
lanjut	meminta	mengibaratkannya
lanjutnya	memintakan	mengingat
lebih	memisalkan	mengingatkan
lewat	memperbuat	menginginkan
lima	mempergunakan	mengira
luar	memperkirakan	mengucapkan
macam	memperlihatkan	mengucapkannya
maka	mempersiapkan	mengungkapkannya
makanya	mempersoalkan	menjadi
makin	mempertanyakan	menjawab
malah	mempunyai	menjelaskan
malahan	memulai	menuju
mampu	memungkinkan	menunjuk
mampukah	menaiki	menunjuki
mana	menambahkan	menunjukkan
manakala	menandaskan	menunjuknya
menurut	padahal	sambil
menuturkan	padanya	sampai

menyampaikan	pak	sampaikan
menyangkut	paling	sama
menyatakan	panjang	sangat
menyebutkan	pantas	sangatlah
menyeluruh	para	satu
menyiapkan	pasti	saya
merasa	pastilah	sayalah
mereka	penting	se
merekalah	pentingnya	sebab
merupakan	per	sebabnya
meski	percuma	sebagai
meskipun	perlu	sebagaimana
meyakini	perlukah	sebagainya
meyakinkan	perlunya	sebagian
minta	pernah	sebaik
mirip	persoalan	sebaiknya
misal	pertama	sebaliknya
misalkan	pertanyaan	sebanyak
misalnya	pertanyakan	sebegini
mula	pihak	sebegitu
mulai	pihaknya	sebelum
mulailah	pukul	sebelumnya
mulanya	pula	sebenarnya
mungkin	pun	seberapa
mungkinkah	punya	sebesar
nah	rasa	sebetulnya
naik	rasanya	sebisanya
namun	rata	sebuah
nanti	rupanya	sebut
nantinya	saat	sebutlah
nyaris	saatnya	sebutnya
nyatanya	saja	secara
oleh	sajalah	secukupnya
olehnya	saling	sedang
pada	sama	sedangkan
sedemikian	semakin	seseorang
sedikit	semampu	sesuatu

sedikitnya	semampunya	sesuatunya
seenaknya	semasa	sesudah
segala	semasih	sesudahnya
segalanya	semata	setelah
segera	semata-mata	setempat
seharusnya	semaunya	setengah
sehingga	sementara	seterusnya
seingat	semisal	setiap
sejak	semisalnya	setiba
sejauh	sempat	setibanya
sejenak	semua	setidaknya
sejumlah	semuanya	setinggi
sekadar	semula	seusai
sekadarnya	sendiri	sewaktu
sekali	sendirian	siap
sekalian	sendirinya	siapa
sekaligus	seolah	siapakah
sekalipun	seolah-olah	siapapun
sekarang	seorang	sini
sekecil	sepanjang	sinilah
seketika	sepantasnya	soal
sekiranya	sepantasnyalah	soalnya
sekitar	seperlunya	suatu
sekitarnya	seperti	sudah
sekurangnya	sepertinya	sudahkah
sela	sepihak	sudahlah
selain	sering	supaya
selaku	seringnya	tadi
selalu	serta	tadinya
selama	serupa	tahu
selamanya	sesaat	tahun
selanjutnya	sesama	tak
seluruh	sesampai	tambah
seluruhnya	sesegera	tambahnya
semacam	sesekali	tampak
tampaknya	tersebut	walaupun
tandas	tersebutlah	wong
tandasnya	tertentu	yaitu

tanpa	tertuju	yakin
tanya	terus	yakni
tanyakan	terutama	yang
tanyanya	tetap	
tapi	tetapi	
tegas	tiap	
tegasnya	tiba	
telah	tidak	
tempat	tidakkah	
tengah	tidaklah	
tentang	tiga	
tentu	tinggi	
tentulah	toh	
tentunya	tunjuk	
tepat	turut	
terakhir	tutur	
terasa	tuturnya	
terbanyak	ucap	
terdahulu	ucapnya	
terdapat	ujar	
terdiri	ujarnya	
terhadap	umum	
terhadapnya	umumnya	
teringat	ungkap	
terjadi	ungkapnya	
terjadilah	untuk	
terjadinya	usah	
terkira	usai	
terlalu	waduh	
terlebih	wah	
terlihat	wahai	
termasuk	waktu	
ternyata	waktunya	
tersampaikan	walau	

UNIVERSITAS BRAWIJAYA

Lampiran 2
Nilai *Cosine Similarity*

Text Clustering

Preprocessing Teks | Termlist | Tabel TF-IDF | Cosine Similarity | Hasil Clustering

	Bisn-001	Bisn-002	Bisn-003	Bisn-004	Bisn-005	Bisn-006	Bisn-007	Bisn-008	Bisn-009	Bisn-010
Bisn-002	0,0315...	1	0,11303...	0,13713...	0,06176...	0,0893...	0,08409...	0,11996...	0,0590...	0,0998...
Bisn-003	0,0241...	0,11303...	1	0,04260...	0,20416...	0,1708...	0,00495...	0,00718...	0,0369...	0,0203...
Bisn-004	0,1004...	0,13713...	0,04260...	1	0,08946...	0,0193...	0,06409...	0,08860...	0,1252...	0,1491...
Bisn-005	0,0298...	0,06176...	0,20416...	0,08946...	1	0,1371...	0,01273...	0,02878...	0,1267...	0,0342...
Bisn-006	0,0109...	0,08938...	0,17084...	0,01938...	0,13713...	1	0,01956...	0,02221...	0,0507...	0,0321...
Bisn-007	0,0078...	0,08409...	0,00495...	0,06409...	0,01273...	0,0195...	1	0,89640...	0,2104...	0,0197...
Bisn-008	0,0148...	0,11996...	0,00718...	0,08860...	0,02878...	0,0222...	0,89640...	1	0,2469...	0,0335...
Bisn-009	0,0005...	0,05900...	0,03693...	0,12526...	0,12675...	0,0507...	0,21045...	0,24695...	1	0,0091...
Bisn-010	0,0828...	0,09987...	0,02035...	0,14912...	0,03426...	0,0321...	0,01976...	0,03354...	0,0091...	1
Bisn-011	0,0978...	0,02492...	0,00375...	0,07154...	0,00749...	0,0135...	0,00750...	0,01165...	0,0354...	0,0536...
Bisn-012	0,0136...	0,25931...	0,19272...	0,21497...	0,27594...	0,1048...	0,07026...	0,11361...	0,2285...	0,0814...
Bisn-013	0,0493...	0,14750...	0,15260...	0,09680...	0,25116...	0,1388...	0,02986...	0,03961...	0,0685...	0,0702...
Bisn-014	0,1918...	0,01987...	0,03579...	0,10950...	0,02384...	0,0238...	0,01002...	0,01164...	0,0121...	0,0728...
Bisn-015	0,0344...	0,10128...	0,31856...	0,07960...	0,31093...	0,1854...	0,00955...	0,01098...	0,0705...	0,0428...
Bisn-016	0,0375...	0,05944...	0,07303...	0,05351...	0,03745...	0,0501...	0,01810...	0,02231...	0,0075...	0,0530...
Bisn-017	0,0403...	0,07728...	0,02029...	0,06517...	0,04561...	0,0080...	0,02662...	0,03703...	0,0714...	0,0247...
Bisn-018	0,0237...	0,02407...	0,03718...	0,04702...	0,01120...	0,0220...	0,00767...	0,01114...	0,0108...	0,0453...
Bisn-019	0,0408...	0,10000...	0,26371...	0,10887...	0,42869...	0,1581...	0,01737...	0,02744...	0,1069...	0,0396...
Bisn-020	0,0261...	0,20414...	0,40434...	0,13629...	0,26224...	0,1809...	0,05179...	0,07931...	0,0841...	0,0594...
Bisn-021	0,0054...	0,01825...	0,04243...	0,02769...	0,01734...	0,0227...	0,00516...	0,00566...	0,0079...	0,0088...
Bisn-022	0,0219...	0,03034...	0,02708...	0,01160...	0,02589...	0,0092...	0,00297...	0,00510...	0,0017...	0,0074...
Bisn-023	0,0112...	0,01972...	0,03025...	0,01525...	0,00592...	0,0089...	0,00510...	0,00502...	0,0137...	0,0157...
Bisn-024	0,0134...	0,04602...	0,10917...	0,02826...	0,04852...	0,0452...	0,01181...	0,00988...	0,0188...	0,0279...
Bisn-025	0,0594...	0,02466...	0,02775...	0,02212...	0,00783...	0,0218...	0,00260...	0,00341...	0,0004...	0,0203...
Bisn-026	0,0606...	0,04684...	0,10995...	0,04816...	0,21448...	0,0473...	0,02021...	0,00809...	0,0237...	0,0769...
Bisn-027	0,0242...	0,02509...	0,02432...	0,01561...	0,05290...	0,0381...	0,00927...	0,00330...	0,0508...	0,0045...
Bisn-028	0,0418...	0,03095...	0,02034...	0,05993...	0,01552...	0,0149...	0,01557...	0,00893...	0,0020...	0,0805...
Bisn-029	0,0417...	0,08066...	0,08483...	0,03180...	0,08764...	0,0247...	0,00353...	0,00083...	0,0288...	0,0459...
Bisn-030	0,0576...	0,13842...	0,05453...	0,02480...	0,04028...	0,0296...	0,01812...	0,02308...	0,0139...	0,0220...
Bisn-031	0,0576...	0,13842...	0,05453...	0,02480...	0,04028...	0,0296...	0,01812...	0,02308...	0,0139...	0,0220...
Bisn-032	0,0282...	0,02576...	0,03010...	0,01488...	0,04561...	0,0217...	0,00052...	0,00052...	0,0154...	0,0465...
Bisn-033	0,0551...	0,05009...	0,01843...	0,02873...	0,05100...	0,0457...	0,00960...	0,00342...	0,0517...	0,0535...
Bisn-034	0,0417...	0,03129...	0,01710...	0,04411...	0,00538...	0,0108...	0,00986...	0,00837...	0,0162...	0,0695...
Bisn-035	0,0061...	0,01443...	0,03547...	0,00962...	0,08482...	0,0194...	0,00543...	0,00076...	0,0039...	0,0143...
Bisn-036	0,0617...	0,01694...	0,05198...	0,04997...	0,01780...	0,0183...	0,01692...	0,00491...	0,0030...	0,0324...
Bisn-037	0,0688...	0,11149...	0,10487...	0,06308...	0,06207...	0,0847...	0,00360...	0,00992...	0,0535...	0,0518...
Bisn-038	0,1056...	0,02388...	0,13803...	0,06496...	0,14816...	0,0656...	0,01347...	0,01363...	0,0127...	0,0362...
Bisn-039	0,0189...	0,03490...	0,04228...	0,02305...	0,04925...	0,0424...	0,01059...	0,00572...	0,0402...	0,0178...
Bisn-040	0,0026...	0,01995...	0,02453...	0,02530...	0,01436...	0,0130...	0,00425...	0,00226...	0,0078...	0,0036...
Bisn-041	0,0011...	0,01098...	0,05436...	0,00374...	0,00169...	0,0103...	0,00150...	0,00071...	0,0006...	0,0006...
Bisn-042	0,0129...	0,03080...	0,01844...	0,01302...	0,00428...	0,0307...	0,00238...	0,00128...	0,0421...	0,0039...
Bisn-043	0,0150...	0,03597...	0,03412...	0,02335...	0,01305...	0,0328...	0,00253...	0,00367...	0,0333...	0,0023...
Bisn-044	0,0035...	0,04328...	0,10444...	0,00919...	0,01972...	0,0260...	0,00255...	0,00371...	0,0433...	0,0083...
Bisn-045	0,0055...	0,12726...	0,10355...	0,07483...	0,12509...	0,1027...	0,01587...	0,02808...	0,0241...	0,0461...
Bisn-046	0,0012...	0,07349...	0,09312...	0,00149...	0,06479...	0,0464...	0,01229...	0,00695...	0,0165...	0,0073...
Bisn-047	0,0083...	0,05708...	0,07669...	0,01512...	0,09362...	0,0529...	0,00299...	0,00353...	0,0160...	0,0122...

Bisn-048	0,0052...	0,03341...	0,03298...	0,04173...	0,01902...	0,0068...	0,00115...	0	0	0,1777...
Bisn-049	0,0073...	0,12753...	0,05405...	0,10100...	0,05783...	0,0118...	0,05763...	0,07306...	0,0457...	0,0207...
Bisn-050	0,0021...	0,09504...	0,08653...	0,01694...	0,05021...	0,0632...	0,02462...	0,02303...	0,0349...	0,0060...
Bisn-051	0,0136...	0,03871...	0,12537...	0,03118...	0,18262...	0,0644...	0,00799...	0,00067...	0,0150...	0,0218...
Bisn-052	0,0118...	0,13177...	0,19079...	0,04785...	0,03468...	0,1052...	0,00556...	0,00708...	0,0567...	0,0408...
Bisn-053	0,0173...	0,04106...	0,06194...	0,04346...	0,02018...	0,0192...	0,00028...	0,00027...	0,0043...	0,0031...
Bisn-054	0,0282...	0,09963...	0,04721...	0,11016...	0,03667...	0,0396...	0,01506...	0,02348...	0,0273...	0,0343...
Bisn-055	0,0104...	0,07121...	0,04582...	0,02840...	0,01243...	0,0442...	0,00426...	0,00226...	0,0237...	0,0053...
Bisn-056	0,0061...	0,03603...	0,06864...	0,03912...	0,07032...	0,0531...	0,01759...	0,02221...	0,0244...	0,0124...
Bisn-057	0,0169...	0,06582...	0,02170...	0,02499...	0,01427...	0,0552...	0,01168...	0,00840...	0,0630...	0,0197...
Bisn-058	0,0022...	0,03571...	0,06774...	0,02814...	0,07655...	0,0355...	0,00324...	0,00390...	0,1091...	0,0127...
Bisn-059	0,0005...	0,09393...	0,04599...	0,01804...	0,00786...	0,0263...	0,00334...	0,00329...	0,0300...	0,0268...
Bisn-060	0,0461...	0,07706...	0,12744...	0,05485...	0,14541...	0,1331...	0,01528...	0,01291...	0,0271...	0,0721...
Bola-001	0,0001...	0,00343...	0,00391...	0,00449...	0,02627...	0,0049...	7,75575...	7,63032...	0,0001...	0,0024...
Bola-002	0,0010...	0,00730...	0,01099...	0,00239...	0,00297...	0,0032...	0,00016...	0,00016...	0,0021...	0,0017...
Bola-003	0,0001...	0,01342...	0,01926...	7,46803...	0,01094...	0,0635...	0,00277...	0,00272...	0,0040...	0,0005...
Bola-004	0,0013...	0,00887...	0,01126...	0,00350...	0,00018...	0,0131...	0,00012...	0,00011...	0,0052...	0,0020...
Bola-005	0,0107...	0,00497...	0,00705...	0,00180...	0,00086...	0,0144...	0,00876...	0,00738...	0,0046...	0,0045...
Bola-006	0,0026...	0,00670...	0,01817...	0,00325...	0,00985...	0,0171...	0,00776...	0,00386...	0,0070...	0,0158...
Bola-007	0,0018...	0,00912...	0,00979...	0,00346...	0,00556...	0,0014...	0,00093...	0,00091...	0,0013...	0,0104...
Bola-008	0,0030...	0,00086...	0,00406...	0,01077...	0,01247...	0,0053...	0,04833...	0,00792...	0,0078...	0,0070...
Bola-009	0	0,00334...	0,00169...	0	0,00307...	0	0	0	0	0,0013...
Bola-010	0,0013...	0,00626...	0,01090...	0,00208...	0,01470...	0,0122...	0	0	0	0,0016...
Bola-011	0,0175...	0,01774...	0,03859...	0,01076...	0,00949...	0,0214...	0,01813...	0,01614...	0,0134...	0,0263...
Bola-012	0,0047...	0,00952...	0,00489...	0,00578...	0,00307...	0,0057...	0,00208...	0,00572...	0,0119...	0,0108...
Bola-013	0,0066...	0,01394...	0,02961...	0,01433...	0,00905...	0,0152...	0,00438...	0,00299...	0,0015...	0,0172...
Bola-014	0,0114...	0,01160...	0,02355...	0,01207...	0,01465...	0,0049...	0,00497...	0,00228...	0,0017...	0,0148...
Bola-015	0,0227...	0,04338...	0,05625...	0,01132...	0,00967...	0,0204...	0,00734...	0,00795...	0,0236...	0,0384...
Bola-016	0,0041...	0,02932...	0,02534...	0,02090...	0,00843...	0,0157...	0,02242...	0,01383...	0,0033...	0,0167...
Bola-017	0,0075...	0,01355...	0,02307...	0,01107...	0,00860...	0,0781...	0,00834...	0,00480...	0,0044...	0,0244...
Bola-018	0,0032...	0,02027...	0,01323...	0,03033...	0,02254...	0,0081...	0,01389...	0,01797...	0,0085...	0,0042...
Bola-019	0,0007...	0,03107...	0,01698...	0,02078...	0,00696...	0,0067...	0,01228...	0,01117...	0,0106...	0,0140...
Bola-020	0,0024...	0,02162...	0,00330...	0,01726...	0,00391...	0,0029...	0,00503...	0,00495...	0,0014...	0,0086...
Bola-021	0,0016...	0,00498...	0,00687...	0,00121...	0,00647...	0,0019...	0,00333...	0,00328...	0,0001...	0,0030...
Bola-022	0,0085...	0,01222...	0,00880...	0	0,00518...	0,0046...	0	0	0	0,0012...
Bola-023	0	0,01322...	0,00227...	0,00517...	0,00311...	0,0061...	0	0	0	0,0024...
Bola-024	0,0027...	0,00078...	0,00078...	0,00070...	0,00990...	0,0093...	0,05853...	0,00932...	0,0071...	0,0073...
Bola-025	0,0037...	0,00108...	0,00108...	0,00098...	0,00486...	0,0041...	0,06827...	0,01296...	0,0099...	0,0088...
Bola-026	0	0,01374...	0,01058...	0,00304...	0,00374...	0,0010...	0,00493...	0,00097...	0	0,0067...
Bola-027	0,0062...	0,00516...	0,00167...	0,00640...	0,00818...	0,0025...	0,01512...	0,02163...	0,0038...	0,0068...
Bola-028	0,0013...	0,00396...	0,02052...	0,00361...	0,01533...	0,0016...	0	0	0	0,0005...
Bola-029	0,0009...	0,00720...	0,01652...	0,00416...	0,01066...	0,0020...	0,00379...	0,00373...	0,0003...	0,0005...
Bola-030	0	0,00291...	0,00526...	0	0,00368...	0,0009...	0	0	0,0008...	0
Bola-031	0,0002...	0,01083...	0,00677...	0,00988...	0,00336...	0,0187...	0,00374...	0,01026...	0,0002...	0,0141...
Bola-032	0,0090...	0,00135...	0,01141...	0,00438...	0,00364...	0	0	0	0,0036...	0,0084...
Bola-033	0,0009...	0,01032...	0,02019...	0,00258...	0,00014...	0,0028...	9,11053...	8,96318...	0,0030...	0,0049...
Bola-034	0,0008...	0,00089...	0,01005...	0,00290...	0	0	0,05248...	0,05701...	0,0010...	0,0142...
Bola-035	0	0,00053...	0,00113...	0,00362...	0	0	0,00054...	0	0,0039...	0,0054...
Bola-036	0,0052...	0,00634...	0,02199...	0,00688...	0,00165...	0,0054...	0,00949...	0,00933...	0,0012...	0,0188...
Bola-037	0,0112...	0,02493...	0,01067...	0,00132...	0,01140...	0,0123...	0,02791...	0,03051...	0	0,0152...
Bola-038	0	0,02191...	0,00616...	0,00066...	0,01836...	0,0159...	0	0	0	0,0018...
Bola-039	0,0030...	0,00621...	0,00400...	0,00189...	0,01152...	0,0005...	0,00045...	0,00044...	0,0001...	0,0119...
Bola-040	0,0032...	0,01146...	0,01502...	0,00426...	0,00459...	0,0004...	0,04180...	0,04483...	0,0029...	0,0105...
Bola-041	0,0008...	0,00282...	0,00782...	0,00560...	0,00540...	0,0032...	0	0	0,0118...	0,0089...
Bola-042	0,0014...	0,00500...	0,00709...	9,54294...	0,00312...	0,0043...	0,00010...	0,00010...	0,0014...	0,0043...

Bola-043	0,0006...	0,00726...	0,01656...	0,00182...	0,00569...	0,0028...	0,00329...	5,68052...	0,0060...	0,0027...
Bola-044	0,0046...	0,00867...	0,00814...	0,00176...	0,00838...	0,0014...	0	0	0,0090...	0,0020...
Bola-045	0	0,00171...	0,02001...	0	0,00058...	0,0021...	0,00810...	0,00797...	0,0014...	0,0030...
Bola-046	0	0,00263...	0,00357...	0	0,00665...	0,0005...	0	0	0,0014...	0,0021...
Bola-047	0,0082...	0,01245...	0,02736...	0,00247...	0,00390...	0,0180...	0	0	0,0079...	0,0034...
Bola-048	0	0,00080...	0,00192...	0,00405...	0,00356...	0	0,04915...	0,05337...	0	0,0042...
Bola-049	0,0038...	0,01011...	0,00911...	0,00565...	0,01160...	0,0021...	0,02583...	0,02819...	0,0220...	0,0166...
Bola-050	0,0037...	0,00532...	0,00308...	0,00482...	0	0,0005...	0,00429...	0	0,0014...	0,0099...
Bola-051	0,0014...	0,00441...	0,00435...	0	0,00565...	0,0035...	0,00093...	0,00091...	0,0031...	0,0028...
Bola-052	0,0037...	0,00135...	0,01153...	0,00898...	0,00101...	0,0020...	0,00583...	0,00376...	0	0,0013...
Bola-053	0,0003...	0,00195...	0,01288...	0,00107...	0,00489...	0,0050...	0,00680...	0,00151...	0,0051...	0,0060...
Bola-054	0	0,00303...	0,00224...	0,00102...	0,00030...	0,0009...	0,03491...	0,03151...	0	0,0009...
Bola-055	0,0026...	0,00118...	0,01029...	0,00126...	0,00011...	0	7,18899...	7,07272...	0,0116...	0,0021...
Bola-056	0	0,00900...	0,01512...	0	0,00163...	0,0035...	0,02249...	0,02459...	0,0075...	0
Bola-057	0,0001...	0,00479...	0,00317...	0,00392...	0,00016...	0	0,00010...	0,00010...	0,0016...	0,0039...
Bola-058	0,0111...	0,00072...	0,00407...	0,00781...	0,00417...	0,0195...	0,00798...	0,00167...	0,0002...	0,0071...
Bola-059	0	0,01426...	0,01503...	0,00920...	0,01301...	0,0016...	0,00421...	0,00414...	0,0018...	0,0007...
Bola-060	0,0019...	0,00780...	0,01619...	0,00087...	0,00258...	0	0,00208...	0	0	0,0060...
Kosm-001	0,0021...	0,01670...	0,01957...	0,01521...	0,01379...	0,0059...	0,01557...	0,01300...	0,0033...	0,0106...
Kosm-002	0,0107...	0,00802...	0,01189...	0,01064...	0,00045...	0,0123...	0	0	0	0,0055...
Kosm-003	0,0038...	0,00932...	0,00572...	0,00815...	0,01257...	0,0023...	0	0	0,0018...	0,0115...
Kosm-004	0,0013...	0,00099...	0,00796...	0,00163...	0,00388...	0,0050...	0,00847...	0,00410...	0,0029...	0,0031...
Kosm-005	0	0,00580...	0,01172...	0	0,00198...	0,0026...	0,00222...	0	0	0,0025...
Kosm-006	0	0,00782...	0,00940...	0,00239...	0,00512...	0,0032...	0	0	0,0168...	0,0236...
Kosm-007	0	0,00124...	0	0	0	0,0009...	0,00380...	0	0,0186...	0,0003...
Kosm-008	0,0023...	0,02346...	0,02543...	0,01260...	0,01004...	0,0105...	0	0	0,0133...	0,0033...
Kosm-009	0,0001...	0,00419...	0,00771...	0,01022...	0,00303...	0,0038...	0,00496...	0,00549...	0,0136...	0,0017...
Kosm-010	0,0060...	0,01050...	0,00915...	0,01075...	0,04153...	0,0085...	0,00831...	0,00791...	0,0038...	0,0059...
Kosm-011	0,0186...	0,03498...	0,04065...	0,01516...	0,02413...	0,0193...	0,01245...	0,00636...	0,0129...	0,0162...
Kosm-012	0,0072...	0,00064...	0,00679...	0,00057...	0,01170...	0,0055...	0,00234...	0	0	0,0017...
Kosm-013	0,0043...	0,02028...	0,12110...	0,01666...	0,04807...	0,0299...	0,00485...	0,00694...	0,0095...	0,0116...
Kosm-014	0,0063...	0,00430...	0,02866...	0,02384...	0,01452...	0,0083...	0,01094...	0,01046...	0,0053...	0,0072...
Kosm-015	0,0051...	0,00747...	0,00776...	0,00616...	0,00066...	0,0042...	0,00259...	0,00046...	0,0038...	0,0028...
Kosm-016	0,0042...	0,01896...	0,01350...	0,00227...	0,01884...	0,0060...	0,00214...	0,00216...	0,0060...	0,0053...
Kosm-017	0	0	0	0	0	0	0,00028...	0	0	0,0028...
Kosm-018	0,0016...	0,00381...	0,00502...	0,00057...	0,00797...	0	0,00476...	0,00062...	0	0,0074...
Kosm-019	0,0026...	0,00361...	0,01163...	0,01645...	0,00438...	0,0039...	0,00663...	0,00473...	0,0014...	0,0088...
Kosm-020	0,0023...	0,00965...	0,02834...	0,01608...	0,01284...	0,0159...	0,00300...	0,00078...	0,0123...	0,0054...
Kosm-021	0	0,00304...	0,00425...	0,00161...	0,00750...	0,0050...	0,00090...	0,00088...	0,0011...	0,0140...
Kosm-022	0,0055...	0,01848...	0,02514...	0,00232...	0,01300...	0,0318...	0,00189...	0,00133...	0	0,0053...
Kosm-023	0,0062...	0,00825...	0,03576...	0,04983...	0,03540...	0,0155...	0,00166...	0,00098...	0,0185...	0,0099...
Kosm-024	0,0034...	0,00914...	0,01647...	0	0,00819...	0,0064...	0,00067...	0,00067...	0	0,0036...
Kosm-025	0,0052...	0,01639...	0,02661...	0,00521...	0,01072...	0,0109...	0,00150...	0,00071...	0,0044...	0,0080...
Kosm-026	0,0018...	0,00307...	0,00621...	0,00961...	0,00309...	0,0053...	0	0,00323...	0	0,0018...
Kosm-027	0	0,00173...	0	0,00806...	0,00237...	0	0,00357...	0,00070...	0	0,0089...
Kosm-028	0,0075...	0,00200...	0,00782...	0,01455...	0,01043...	0,0257...	0,01079...	0,01061...	0	0,0059...
Kosm-029	0,0026...	0,00322...	0,01080...	0,01666...	0,00782...	0,0054...	0,00205...	0,00316...	0,0045...	0,0114...
Kosm-030	0,0044...	0,01658...	0,00616...	0,00322...	0,00476...	0,0093...	0,00129...	0,00093...	0,0075...	0,0048...
Kosm-031	0,0021...	0,05664...	0,00426...	0,02308...	0,00797...	0,0027...	0,02046...	0,02747...	0,0070...	0,0078...
Kosm-032	0,0027...	0,00139...	0,01623...	0,00595...	0,00310...	0,0009...	0,00919...	0,00420...	0,0007...	0,0106...
Kosm-033	0	0,00102...	0,01187...	0,00517...	0,00215...	0,0072...	0	0	0,0023...	0,0018...
Kosm-034	0	0,00959...	0,00935...	0,00436...	0,00796...	0,0075...	0	0,00063...	0,0060...	0,0018...
Kosm-035	0,0026...	0,00528...	0,02546...	0,02601...	0,04211...	0,0157...	0,00854...	0,00649...	0,0017...	0,0115...
Kosm-036	0,0010...	0,00183...	0,01246...	0,01107...	0,00713...	0,0095...	0,00025...	0,00157...	0,0054...	0,0113...
Kosm-037	0	0,00633...	0,00097...	0,00219...	0	0,0069...	0,00069...	0	0	0,0055...

Kosm-038	0,0054...	0,00498...	0,01095...	0,01659...	0,00257...	0,0017...	0,00046...	0,00046...	0,0007...	0,0060...
Kosm-039	0,0074...	0,00947...	0,02464...	0,00967...	0,00340...	0,0035...	0,00135...	0,00043...	0,0025...	0,0034...
Kosm-040	0,0028...	0,00894...	0,02088...	0,01200...	0,01687...	0,0056...	0,00394...	0,00287...	0,0022...	0,0046...
Kosm-041	0,0005...	0,01253...	0,01829...	0,01484...	0,00677...	0,0292...	0,00520...	0,00235...	0,0038...	0,0162...
Kosm-042	0,0099...	0,04348...	0,01746...	0,01381...	0,01736...	0,0263...	0,00120...	0,00087...	0,0171...	0,0125...
Kosm-043	0,0022...	0,03243...	0,01926...	0,00409...	0,00383...	0,0147...	0,00111...	0	0,0245...	0,0012...
Kosm-044	0,0024...	0,01904...	0,00933...	0,00400...	0,00312...	0,0099...	0,00348...	0,00246...	0,0025...	0,0049...
Kosm-045	0,0117...	0,01654...	0,03427...	0,00794...	0,02261...	0,0235...	0,00536...	0,00604...	0,0053...	0,0146...
Kosm-046	0,0106...	0,02822...	0,03260...	0,02340...	0,02072...	0,0084...	0,01509...	0,00994...	0,0147...	0,0114...
Kosm-047	0,0024...	0,01064...	0,00509...	0,03340...	0,00214...	0,0040...	0,01457...	0,01041...	0	0,0189...
Kosm-048	0,0086...	0,02115...	0,01055...	0,03108...	0,00207...	0,0210...	0,00844...	0,00830...	0,0015...	0,0261...
Kosm-049	0,0165...	0,00491...	0,02055...	0,00585...	0,00458...	0,0022...	0,00835...	0,00478...	0	0,0058...
Kosm-050	0,0022...	0,00742...	0,02534...	0,01805...	0,00773...	0,0134...	0,00052...	0	0,0067...	0,0087...
Kosm-051	0,0104...	0,02192...	0,04029...	0,02528...	0,02139...	0,0184...	0,02047...	0,00739...	0,0073...	0,0099...
Kosm-052	0,0047...	0,00817...	0,01785...	0,00818...	0,01365...	0,0141...	0,00032...	0	0,0043...	0,0108...
Kosm-053	0,0023...	0,03104...	0,02049...	0,01260...	0,01223...	0,0144...	0,00164...	0,00161...	0,0170...	0,0346...
Kosm-054	0,0103...	0,01198...	0,03751...	0,01857...	0,07606...	0,0613...	0,00961...	0,01060...	0,0013...	0,0080...
Kosm-055	0,0121...	0,00652...	0,01534...	0,03156...	0,00592...	0,0158...	0,03002...	0,01819...	0	0,0095...
Kosm-056	0,0138...	0,03043...	0,01127...	0,01193...	0,01075...	0,0149...	0,00588...	0,00956...	0,0243...	0,0171...
Kosm-057	0,0119...	0,02016...	0,03195...	0,01662...	0,02891...	0,0302...	0,00113...	0,00082...	0,0175...	0,0133...
Kosm-058	0,0156...	0,03651...	0,01489...	0,01674...	0,00774...	0,0152...	0,00331...	0,00695...	0,0137...	0,0137...
Kosm-059	0,0005...	0,04199...	0,03213...	0,00167...	0,00504...	0,0400...	0,01938...	0,00302...	0,0180...	0,0177...
Kosm-060	0,0251...	0,01275...	0,03193...	0,02407...	0,01237...	0,0179...	0,00597...	0,00731...	0,0136...	0,0111...
Naso-001	0,4129...	0,03917...	0,02778...	0,10821...	0,04541...	0,0245...	0,01871...	0,02444...	0,0007...	0,0744...
Naso-002	0,3218...	0,03572...	0,02682...	0,12716...	0,05213...	0,0433...	0,03621...	0,03114...	0,0079...	0,0795...
Naso-003	0,0232...	0,01556...	0,01105...	0,01994...	0,00597...	0,0122...	0,01064...	0,01423...	0,0134...	0,0547...
Naso-004	0,2703...	0,03182...	0,02040...	0,08237...	0,04051...	0,0163...	0,00911...	0,01194...	0,0006...	0,0579...
Naso-005	0,0558...	0,03506...	0,00137...	0,08604...	0,00683...	0,0007...	0,01011...	0,00995...	0,0002...	0,1240...
Naso-006	0,0336...	0,02557...	0,02598...	0,04348...	0,02777...	0,0237...	0,04401...	0,01455...	0,0042...	0,0200...
Naso-007	0,0728...	0,01795...	0,01662...	0,04265...	0,00738...	0,0153...	0,02269...	0,02881...	0,0083...	0,0275...
Naso-008	0,0320...	0,01076...	0,02914...	0,01851...	0,03150...	0,0190...	0,00840...	0,00498...	0,0184...	0,0168...
Naso-009	0,0121...	0,01279...	0,00861...	0,01393...	0,00362...	0,0029...	0,00492...	0,00414...	0,0067...	0,0271...
Naso-010	0,2739...	0,03142...	0,01130...	0,14640...	0,01548...	0,0126...	0,00847...	0,01475...	0,0006...	0,0821...
Naso-011	0,0580...	0,01601...	0,00504...	0,01649...	0,01540...	0,0109...	0,00966...	0,00810...	0,0169...	0,0097...
Naso-012	0,0226...	0,01986...	0,00112...	0,01718...	0,00210...	0,0113...	0,00312...	0,00275...	0,0022...	0,0492...
Naso-013	0,0283...	0,02042...	0,00114...	0,01536...	0,00224...	0,0080...	0,00479...	0,00383...	0,0019...	0,0443...
Naso-014	0,0236...	0,01202...	0,00800...	0,00359...	0,00032...	0,0159...	0,01574...	0,00589...	0,0101...	0,0072...
Naso-015	0,0539...	0,01297...	0,02166...	0,02654...	0,00733...	0,0107...	0,01638...	0,02580...	0,0016...	0,0455...
Naso-016	0,0318...	0,01404...	0,02492...	0,02149...	0,00870...	0,0255...	0,00546...	0,00727...	0,0014...	0,0276...
Naso-017	0,0259...	0,01595...	0,00447...	0,00856...	0,00090...	0,0046...	0,00252...	0,00274...	0,0001...	0,0174...
Naso-018	0,0320...	0,01673...	0,02628...	0,02786...	0,01675...	0,0308...	0,05377...	0,02453...	0,0023...	0,0210...
Naso-019	0,0263...	0,04589...	0,04962...	0,02972...	0,02800...	0,0152...	0,02856...	0,04823...	0,0040...	0,0158...
Naso-020	0,0488...	0,00974...	0,01069...	0,01856...	0,00099...	0,0093...	0,00835...	0,01409...	0,0037...	0,0299...
Naso-021	0,0284...	0,16201...	0,05015...	0,08406...	0,04449...	0,0220...	0,07261...	0,09352...	0,0224...	0,0604...
Naso-022	0,0099...	0,01346...	0,00234...	0,01234...	0,00602...	0,0058...	0,22632...	0,23956...	0,0026...	0,0424...
Naso-023	0,0493...	0,03097...	0,00742...	0,06888...	0,00691...	0,0001...	0,00366...	0,00273...	0,0004...	0,0946...
Naso-024	0,2488...	0,04656...	0,02748...	0,11067...	0,02037...	0,0212...	0,00531...	0,01087...	0,0071...	0,0694...
Naso-025	0,0017...	0,00322...	0,00080...	0,00542...	0,01538...	0	0,07008...	0,02558...	0	0,0081...
Naso-026	0,0501...	0,04590...	0,02770...	0,02219...	0,03279...	0,0173...	0,12228...	0,11440...	0,0409...	0,0506...
Naso-027	0,0665...	0,06149...	0,01100...	0,08119...	0,00354...	0,0156...	0,05723...	0,06595...	0,0058...	0,1487...
Naso-028	0,0393...	0,01933...	0,01336...	0,01634...	0,00042...	0,0070...	0,00174...	0,00026...	0,0382...	0,0088...
Naso-029	0,0499...	0,02416...	0,01382...	0,04584...	0,00576...	0,0149...	0,04094...	0,04398...	0,0006...	0,0454...
Naso-030	0,0834...	0,02259...	0,01360...	0,02693...	0,00728...	0,0150...	0,02919...	0,02188...	0,0006...	0,0250...
Naso-031	0,0171...	0,02375...	0,00416...	0,04351...	0,00421...	0,0240...	0,00520...	0,00512...	0,0024...	0,1017...
Naso-032	0,1654...	0,03875...	0,00869...	0,09999...	0,00932...	0,0129...	0,01185...	0,01573...	0,0021...	0,0680...

Naso-033	0,0151...	0,00920...	0,00067...	0,00833...	0,00115...	0,0029...	0,00030...	0,00030...	0,0056...	0,0587...
Naso-034	0,0243...	0,00743...	0,00296...	0,01706...	0,00034...	0,0036...	0,00606...	0,00191...	0,0015...	0,0393...
Naso-035	0,1086...	0,03063...	0,01539...	0,03796...	0,00385...	0,0152...	0,02624...	0,02242...	0,0006...	0,0326...
Naso-036	0,0622...	0,02915...	0,01141...	0,04175...	0,01101...	0,0104...	0,02209...	0,01692...	0,0080...	0,0292...
Naso-037	0,1183...	0,03165...	0,00476...	0,05288...	0,00704...	0,0046...	0,00288...	0,00284...	0,0024...	0,1236...
Naso-038	0,0318...	0,04024...	0,00047...	0,07421...	0,00289...	0,0027...	0,00192...	0,00021...	0,0015...	0,1676...
Naso-039	0,0181...	0,00775...	0,00385...	0,00545...	0,00272...	0,0023...	0,00017...	0,00017...	0,0017...	0,0265...
Naso-040	0,0588...	0,01279...	0,00132...	0,00388...	0,00349...	0,0008...	0,00240...	0,00018...	0,0068...	0,0070...
Naso-041	0,0382...	0,02787...	0,00502...	0,06261...	0,00819...	0,0483...	0,00154...	0,00193...	0,0306...	0,1118...
Naso-042	0,0211...	0,02602...	0,00244...	0,04406...	0,00754...	0,0069...	0,05269...	0,05271...	0,0033...	0,0984...
Naso-043	0,0204...	0,02349...	0,00268...	0,04499...	0,00672...	0,0144...	0,00579...	0,01143...	0,0095...	0,0920...
Naso-044	0,0112...	0,00888...	0,02424...	0,00681...	0,01181...	0,0070...	0,01163...	0,00477...	0,0021...	0,0202...
Naso-045	0,0107...	0,00629...	0,01683...	0,00509...	0,00525...	0,0108...	0,01324...	0,00543...	0,0024...	0,0248...
Naso-046	0,0175...	0,00318...	0,00777...	0,00223...	0,00929...	0,0013...	0,00546...	0,00537...	0,0009...	0,0251...
Naso-047	0,2098...	0,03706...	0,00974...	0,09621...	0,01888...	0,0122...	0,02149...	0,01499...	0,0019...	0,0829...
Naso-048	0,0285...	0,03275...	0,01195...	0,02734...	0,00620...	0,0243...	0,02158...	0,01168...	0,0229...	0,0683...
Naso-049	0,0581...	0,01415...	0,00574...	0,03720...	0,00283...	0,0145...	0,02473...	0,00890...	0,0186...	0,0478...
Naso-050	0,0083...	0,00903...	0,00231...	0,00499...	0,00068...	0,0012...	0,02223...	0,00281...	0,0016...	0,0066...
Naso-051	0,0729...	0,02211...	0,01067...	0,02688...	0,00509...	0,0078...	0,01291...	0,01145...	0,0176...	0,0311...
Naso-052	0,0653...	0,02673...	0,00480...	0,03953...	0,00529...	0,0111...	0,02715...	0,01263...	0,0171...	0,0661...
Naso-053	0,0143...	0,00283...	0,00730...	0,01653...	0,00363...	0,0180...	0,01199...	0,00532...	0,0013...	0,0066...
Naso-054	0,0122...	0,00913...	0,01153...	0,00415...	0,00616...	0,0105...	0,02405...	0,02369...	0,0349...	0,0099...
Naso-055	0,0116...	0,00938...	0,00278...	0,00695...	0,00033...	0,0034...	0,01991...	0,00667...	0,0036...	0,0032...
Naso-056	0,0073...	0,00946...	0,00207...	0,02037...	0,00474...	0,0052...	0,09266...	0,08746...	0,0041...	0,0152...
Naso-057	0,0105...	0,00768...	0,02626...	0,06305...	0,00445...	0,0132...	0,00666...	0,00991...	0,0192...	0,0162...
Naso-058	0,0184...	0,01415...	0,01120...	0,00658...	0,01007...	0,0046...	0,00214...	0,00405...	0,0068...	0,0096...
Naso-059	0,0380...	0,01821...	0,01966...	0,01360...	0,01817...	0,0229...	0,06375...	0,02827...	0,0055...	0,0097...
Naso-060	0,0326...	0,02209...	0,03852...	0,04661...	0,01990...	0,0217...	0,02709...	0,02802...	0,0088...	0,0187...
Spot-001	0,0001...	0,01656...	0,02332...	0,00907...	0,00293...	0,0030...	0,00334...	0,00139...	0,0022...	0,0111...
Spot-002	0,0096...	0,00242...	0,00941...	0,00192...	0,02156...	0,0108...	0,00045...	0,00044...	0,0092...	0,0095...
Spot-003	0,0031...	0,00091...	0,00413...	0,00741...	0,00613...	0	0,00012...	0,00011...	0,0002...	0,0058...
Spot-004	0,0008...	0,01022...	0,00904...	0,00109...	0,01229...	0,0065...	0,00050...	0,00049...	0,0002...	0,0062...
Spot-005	0,0025...	0,00113...	0,01436...	0,00266...	0,00719...	0,0140...	0,00064...	0,00063...	0,0144...	0,0055...
Spot-006	0,0043...	0,00622...	0,00488...	0,00066...	0,00109...	0,0114...	0,00252...	0,00069...	0,0062...	0,0027...
Spot-007	0,0010...	0,01265...	0,01730...	0,00490...	0,01813...	0,0007...	0,00063...	0,00062...	0,0202...	0,0217...
Spot-008	0,0134...	0,00934...	0,02548...	0,01467...	0,01504...	0,0068...	8,55089...	0,00227...	0,0036...	0,0245...
Spot-009	0,0020...	0,00732...	0,00040...	0,00364...	0,00575...	0,0010...	0,01011...	0,00841...	0,0014...	0,0078...
Spot-010	0,0009...	0,00062...	0,01041...	8,37750...	0,00013...	0,0033...	0,00558...	0,00549...	0,0026...	0,0064...
Spot-011	0,0001...	0,00682...	0,02362...	0,00845...	0,00013...	0,0088...	0,00282...	8,70227...	0,0038...	0,0019...
Spot-012	0,0015...	0,00387...	0,00336...	0,00015...	0,00335...	0,0060...	0,00016...	0,00016...	0,0024...	0,0042...
Spot-013	0,0103...	0,00252...	0,02741...	0,01241...	0,00786...	0,0115...	0,00084...	0	0,0048...	0,0051...
Spot-014	0,0007...	0,00631...	0,00359...	0,00508...	0,00791...	0,0085...	0	0	0,0021...	0
Spot-015	0,0047...	0,00889...	0,00687...	0,00955...	0,03240...	0,0030...	0,00014...	0,00389...	0,0002...	0,0204...
Spot-016	0,0012...	0,00934...	0,01623...	0,00705...	0,00902...	0,0088...	0,00236...	0,00414...	0,0010...	0,0124...
Spot-017	0,0021...	0,00129...	0,00674...	0,01589...	0	0,0071...	0,00043...	0	0,0070...	0,0449...
Spot-018	0,0192...	0,01756...	0,05466...	0,01504...	0,03011...	0,0575...	0,00437...	0,00297...	0,0531...	0,0480...
Spot-019	0,0022...	0,00222...	0,00441...	0,00332...	0,01026...	0,0068...	0,00350...	0,00344...	0,0014...	7,9349...
Spot-020	0,0001...	0,01095...	0,00791...	0,00011...	0,00526...	0,0020...	0,00228...	0,00011...	0,0018...	0,0001...
Spot-021	0,0013...	0,01668...	0,03942...	0,00131...	0,00613...	0,0143...	0,00426...	0,00026...	0,0160...	0,0029...
Spot-022	0,0009...	0,00836...	0,03910...	0,00086...	0,01441...	0,0096...	0,01466...	0,00055...	0,0016...	0,0059...
Spot-023	0,0015...	0,00290...	0,01447...	0,00020...	0,00662...	0,0036...	0,01220...	0,00021...	0,0050...	0,0068...
Spot-024	0,0001...	0,01499...	0,11759...	0,00846...	0,04458...	0,0513...	0,00011...	0,00010...	0,0090...	0,0089...
Spot-025	0,0002...	0,00793...	0,10528...	0,00363...	0,03831...	0,0361...	0,00012...	0,00012...	0,0226...	0,0058...
Spot-026	0,0027...	0,00903...	0,02202...	0,00059...	0,01512...	0,0113...	0,00521...	0,00063...	0,0019...	0,0005...
Spot-027	0,0013...	0,00086...	0,03232...	0,00010...	0,00282...	0,0024...	0,01247...	0,00011...	0,0002...	0,0023...

Spot-028	0,0032...	0,01321...	0,08577...	0,02439...	0,02757...	0,0888...	0,00336...	0,00330...	0,0104...	0,0247...
Spot-029	0,0031...	0,00311...	0,02626...	0,00165...	0,00476...	0,0036...	0,00604...	0,00379...	0,0038...	0,0052...
Spot-030	0,0004...	0,01189...	0,04373...	0,00202...	0,01759...	0,0203...	0,01344...	0,00024...	0,0088...	0,0050...
Spot-031	0,0010...	0,00162...	0,04585...	0,00057...	0,01179...	0,0014...	0,00732...	0,00060...	0,0052...	0,0005...
Spot-032	0,0007...	0,00347...	0,05189...	0,00127...	0,01064...	0,0111...	0,00454...	0,00160...	0,0043...	0,0048...
Spot-033	0,0005...	0,00327...	0,01477...	0,00062...	0,00995...	0,0071...	0,01274...	0,00033...	0,0046...	0,0012...
Spot-034	0,0005...	0,01280...	0,11104...	0,00437...	0,03379...	0,0818...	0,01053...	0,01036...	0,0226...	0,0040...
Spot-035	0	0,01885...	0,08746...	0,00910...	0,03537...	0,0918...	0,00956...	0,00940...	0,0191...	0,0030...
Spot-036	0	0,01299...	0,08557...	0,01888...	0,04997...	0,0590...	0,00335...	0,00330...	0,0432...	0,0048...
Spot-037	0	0,00507...	0,01883...	0	0,00408...	0,0039...	0	0	0,0013...	0,0122...
Spot-038	0,0001...	0,05689...	0,07757...	0,01818...	0,03904...	0,0230...	0,01646...	0,02169...	0,0123...	0,0103...
Spot-039	0,0099...	0,01759...	0,05571...	0,01153...	0,01948...	0,0378...	0,00871...	0,00647...	0,0086...	0,0155...
Spot-040	0,0001...	0,00779...	0,03299...	0,00089...	0,02853...	0,0141...	0,00845...	0,00078...	0,0126...	0,0058...
Spot-041	0,0326...	0,02264...	0,02176...	0,03598...	0,01959...	0,0087...	0,00318...	0,01046...	0,0011...	0,0475...
Spot-042	0,0021...	0,02124...	0,08579...	0,01244...	0,02605...	0,0363...	0,00149...	0,00107...	0,0102...	0,0046...
Spot-043	0,0108...	0,01151...	0,03182...	0,00329...	0,00963...	0,0121...	0,00500...	0,00496...	0,0106...	0,0100...
Spot-044	0,0052...	0,02686...	0,15041...	0,02820...	0,06200...	0,0790...	0,01311...	0,01290...	0,0132...	0,0056...
Spot-045	0,0170...	0,02340...	0,07856...	0,00657...	0,01735...	0,0350...	0,00198...	0,00195...	0,0019...	0,0139...
Spot-046	0,0079...	0,00832...	0,01676...	0,00266...	0,00431...	0,0270...	0,00436...	0,00283...	0,0040...	0,0091...
Spot-047	0,0181...	0,00453...	0,01040...	0,01438...	0,00300...	0,0008...	0,00796...	0,00504...	0,0026...	0,0117...
Spot-048	0,0072...	0,00559...	0,01268...	0,00890...	0,00688...	0,0039...	0,00215...	0,00143...	0,0018...	0,0174...
Spot-049	0,0007...	0,00617...	0,07989...	0,00741...	0,01987...	0,0237...	0	0	0,0038...	0,0044...
Spot-050	0,0009...	0,01209...	0,01965...	0,00226...	0,02295...	0,0231...	0,00200...	0	0,0039...	0,0119...
Spot-051	0	0	0	0	0	0	0	0	0	0
Spot-052	0,0055...	0,01111...	0,03396...	0,00125...	0,01340...	0,0138...	0	0	0,0097...	0,0048...
Spot-053	0,0020...	0,02033...	0,03699...	0,01514...	0,03223...	0,0154...	0,00316...	0,00311...	0,0290...	0,0024...
Spot-054	0	0,00278...	0,01504...	0,00125...	0	0,0009...	0,00109...	0	0,0138...	0,0033...
Spot-055	0,0098...	0,01053...	0,05493...	0,02760...	0,01839...	0,0168...	0,00576...	0,00536...	0,0135...	0,0097...
Spot-056	0,0059...	0,03223...	0,05399...	0,03454...	0,04103...	0,0267...	0,00206...	0,00394...	0,0063...	0,0303...
Spot-057	0	0,00067...	0,01005...	0,00043...	0,01263...	0,0043...	0,00050...	0	0,0017...	0,0079...
Spot-058	0,0141...	0,00627...	0,01154...	0,02299...	0,03031...	0,0054...	0,00499...	0,00388...	0,0010...	0,0263...
Spot-059	0,0002...	0,01264...	0,03842...	0,00140...	0,01168...	0,0209...	0,01766...	0,00268...	0,0126...	0,0023...
Spot-060	0	0	0,00419...	0,00235...	0	0,0018...	0,00321...	0,00316...	0,0012...	0,0006...