PERINGKASAN MULTI DOKUMEN BERBAHASA INGGRIS BERBASIS KONTEN MENGGUNAKAN *SINGLE PASS CLUSTERING* DAN PERANGKINGAN BERBASIS ALGORITMA GENETIKA

SKRIPSI

Oleh :

Tiara Arintadewi
0610963064 - 96

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

PERINGKASAN MULTI DOKUMEN BERBAHASA INGGRIS BERBASIS KONTEN MENGGUNAKAN *SINGLE PASS CLUSTERING* DAN PERANGKINGAN BERBASIS ALGORITMA GENETIKA

SKRIPSI

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

Oleh :

Tiara Arintadewi
0610963064 - 96

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

2012

LEMBAR PENGESAHAN SKRIPSI

PERINGKASAN MULTI DOKUMEN BERBAHASA INGGRIS BERBASIS KONTEN MENGGUNAKAN *SINGLE PASS CLUSTERING* DAN PERANGKINGAN BERBASIS ALGORITMA GENETIKA

Oleh:
TIARA ARINTADEWI
0610963064-96

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

Pembimbing I	Pembimbing II
Dewi Yanti L., S.Kom., M.Kom.	Edy Santoso, S.Si., M.Kom.
NIP. 198111162005012004	NIP. 197404142003121004

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, M.Sc.
NIP. 196709071992031001

UNIVERSITAS BRAWIJAYA

LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama	:	Tiara Arintadewi
NIM	:	0610963064-96
Jurusan	:	Matematika
Program Studi	:	Ilmu Komputer
Penulis Skripsi berjudul	:	Peringkasan Multi Dokumen Berbahasa Inggris Berbasis Konten Menggunakan *Single Pass Clustering* dan Perangkingan Berbasis Algoritma Genetika

Dengan ini menyatakan bahwa :

1. **Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub dalam isi dan tertulis pada daftar pustaka dalam Skripsi ini.**
2. **Apabila dikemudian hari ternyata Skripsi 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 Februari 2012
Yang menyatakan,

(Tiara Arintadewi)
NIM. 0610963064-96

UNIVERSITAS BRAWIJAYA

PERINGKASAN MULTI DOKUMEN BERBAHASA INGGRIS BERBASIS KONTEN MENGGUNAKAN *SINGLE PASS CLUSTERING* DAN PERANGKINGAN BERBASIS ALGORITMA GENETIKA

ABSTRAK

Saat ini perolehan informasi dan pertukaran data berupa teks melibatkan banyak sumber informasi. Tidak terbatasnya informasi digital ini memerlukan suatu cara yang mudah untuk mengetahui informasi dari dokumen, salah satunya dengan peringkasan dokumen. Metode peringkasan dokumen semula ditujukan untuk membuat ringkasan dari dokumen tunggal, namun sekarang menjadi metode yang menghasilkan ringkasan dari multi dokumen sehingga pengguna dapat memahami teks dari banyak sumber dengan mudah.

Berbagai macam dokumen masukan tidak dapat diringkas menjadi satu, hanya dokumen-dokumen yang memiliki kesamaan konten yang dapat menghasilkan satu ringkasan. Untuk itu sebelum diringkas, dokumen-dokumen tersebut dikelompokkan terlebih dahulu menggunakan *Single Pass Clustering*. Peringkasan dilakukan untuk tiap dokumen dengan ekstraksi kalimat dan perangkingan berbasis algoritma genetika. Hasil dari ringksan tiap dokumen dalam satu *cluster* digabung menjadi satu ringkasan. Ekstraksi kalimat berfungsi untuk mengidentifikasi kalimat-kalimat penting berdasarkan fitur-fitur yang ditentukan dan dari nilai fitur-fitur tersebut kalimat akan dirangking secara optimal dengan algoritma genetika.

Penelitian ini diujikan pada beberapa *cluster* dokumen dengan ukuran ringkasan 25%, 50%, dan 75%. Berdasarkan pengujian yang dilakukan dihasilkan rata-rata precision 0.710491, rata-rata recall 0.70388, dan rata-rata F-measure 0.7069. Nilai F-measure merepresentasikan akurasi sistem. Dengan tingkat akurasi yang dihasilkan maka metode ini cukup membantu dalam memperoleh infomasi dengan efisien.

Kata Kunci : Algoritma Genetika, Peringkasan Multi Dokumen, *Single Pass Clustering*, Ekstraksi Kalimat.

UNIVERSITAS BRAWIJAYA

CONTENT BASED ENGLISH MULTI DOCUMENT SUMMARIZATION USING SINGLE PASS CLUSTERING AND RANKED BASED GENETIC ALGORITHM

ABSTRACT

Information retrieval and textual data exchange involve many sources recently. These unlimited digital information need to be extracted from many documents with an easy and a fast way. One of those methods to do it is document summarization. To help user in obtaining document information from many sources, the single document summarization research has been extended to multi document summarization.

The input of multiple documents cannot be summarized into one, only documents with similar content that can be. Therefore, these documents must be grouped using Single Pass Clustering before summarized. Each document is summarized using sentence extraction and ranked with genetic algorithm. Then, the results of summarization for each document that existed in a same cluster are merge into one. Sentence extraction is a phase to identify important sentences based on determined features. By the value of those features, sentences are ranked with genetic algorithm in an optimal way.

This research has been tested on several document clusters with 25%, 50%, and 75% summary measure. This system performs precision average 0.70491, recall average 0.70388, and F-measure average 0.7069. F-measure value represents accuracy of the system. According to the accuration result, this method is quite helpfull in gaining information efficiently.

Keywords : Genetic Algorithm, Multi Document Summarization, Single Pass Clustering, Sentence Extraction.

UNIVERSITAS BRAWIJAYA

Kata Pengantar

Puji syukur kehadirat Allah SWT, karena hanya dengan rahmat dan karuniaNya penulis dapat menyelesaikan skripsi yang berjudul "Peringkasan Multi Dokumen berbahasa Inggris Berbasis Konten Menggunakan *Single Pass Clustering* dan Perangkingan Berbasis Algoritma Genetika"

Skripsi ini disusun dan diajukan sebagai syarat untuk memperoleh gelar sarjana pada program studi Ilmu Komputer, Jurusan Matematika, Fakultas MIPA, Universitas Brawijaya.

Dalam penyelesaian tugas akhir ini, penulis mendapat banyak bantuan baik moral maupun materiil dari banyak pihak. Atas bantuan yang telah diberikan, penulis menyampaikan penghargaan dan ucapan terima kasih kepada:

1. Dewi Yanti Liliana., S.Kom., M.Kom. dan Edy Santoso, S.Si., M.Kom., selaku dosen pembimbing, terima kasih atas semua saran, bantuan, kritikan, waktu, dorongan semangat dan bimbingannya.
2. Drs. Marji, MT., selaku Ketua Program Studi Ilmu Komputer Universitas Brawijaya Malang.
3. Dr. Abdul Rouf Alghofari, M.Sc., selaku Ketua Jurusan Matematika Fakultas MIPA Universitas Brawijaya Malang.
4. Djoko Pramono, ST selaku dosen Penasihat Akademik.
5. Dewi Widyastuti S.Pd., M.Hum., selaku pakar bahasa inggris yang telah memberikan waktu luang untuk meringkas teks berbahasa inggris.
6. 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.
7. Segenap staf dan karyawan di Jurusan Matematika FMIPA Universitas Brawijaya.
8. Keluargaku tersayang terima kasih atas semua cinta, kasih sayang, doa, dan dukungan yang tiada henti.
9. Teman-temanku semua terima kasih dukungannya.
10. Pihak lain yang telah membantu terselesaikannnya skripsi ini yang tidak bisa penulis sebutkan satu-persatu.

Semoga skripsi ini dapat memberikan manfaat kepada pembaca dan bisa diambil manfaatnya untuk pengembangan selanjutnya. Penulis menyadari skripsi ini masih jauh dari sempurna maka penulis sangat menghargai saran dan kritik yang membangun demi perbaikan penulisan dan mutu isi penelitian ini

Malang, 9 Februari 2012

Penulis

DAFTAR ISI

UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

DAFTAR *SOURCE CODE*

UNIVERSITAS BRAWIJAYA

DAFTAR LAMPIRAN

UNIVERSITAS BRAWIJAYA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan informasi *online* menjadi media yang semakin penting untuk menemukan dan merepresentasikan informasi tekstual. Proses untuk mengambil informasi ini disebut dengan *information retrieval*. *Information retrieval* merupakan suatu proses pencarian informasi pada dokumen yang didasarkan pada suatu *query* (inputan *user*) dari kumpulan dokumen yang ada yang diharapkan dapat memenuhi keinginan *user*.

Sebuah perusahaan biasanya memantau perilaku pesaingnya untuk mempertahankan keunggulan dan menemukan inovasi baru agar memperoleh strategi bisnis yang baik. Memantau perilaku perusahaan lain dilakukan dengan berburu informasi melalui berita. Mengingat berita sekarang ini begitu banyak, maka diperlukan suatu cara yang mudah untuk mendapatkan informasi yang dibutuhkan (Chong L and Chen Y, 2009). Salah satu cara yang dapat ditempuh adalah dengan meringkas dokumen. Peringkasan dokumen merupakan proses mengambil dokumen tekstual, mengekstraksi isinya, dan menyajikan konten yang paling penting untuk pengguna dalam bentuk yang lebih padat dan sesuai dengan kebutuhan pengguna (Kogilavani dan Balasubramami, 2010).

Tersedianya sumber informasi yang tidak terbatas mengakibatkan perolehan sumber informasi dan pertukaran data berupa teks melibatkan banyak sumber informasi sehingga memicu penelitian mengenai metode peringkasan dokumen yang semula ditujukan untuk membuat sebuah ringkasan dari dokumen tunggal menjadi metode peringkasan multi dokumen.

Peringkasan multi dokumen adalah proses untuk menghasilkan ringkasan tunggal dari sekumpulan dokumen terkait (Kogilavani dan Balasubramami, 2010). Pekerjaan meringkas berita merupakan pekerjaan rutin untuk editor media dan reporter, terutama untuk *breaking news*. Karena sumber berita sekarang ini banyak dan dalam bentuk elektronik, maka salah satu cara untuk membantunya adalah dengan peringkas dokumen otomatis menggunakan komputer (Zhang, Maiwen, 2005). Dari banyak sumber dapat dihasilkan satu

ringkasan yang dapat mewakili dokumen-dokumen tersebut.

Dokumen-dokumen yang akan diringkas menjadi satu harus memiliki keterkaitan topik atau konten. Untuk itu, sebelum diringkas dokumen-dokumen tersebut dikelompokkan terlebih dahulu sesuai kesamaan konten. Pengelompokkan (*clustering*) dilakukan secara otomatis, sebelum dokumen-dokumen diringkas.

Pada penelitian ini dokumen akan dikelompokkan menggunakan *single pass clustering*. *Single pass clustering* dinilai sederhana dan cukup memadai untuk peringkasan multi dokumen yang melakukan pendekatan *clustering* berbasis konten. Setelah itu dokumen diekstrak menggunakan enam fitur. Hasil dari ekstraksi kalimat akan dirangking untuk menentukan kalimat-kalimat penting. Perangkingan kalimat dilakukan dengan pendekatan algoritma genetika. Nilai enam fitur tersebut bisa bertentangan dalam satu kalimat (ada yang kecil sekaligus ada yang besar untuk satu kalimat), sehingga diperlukan suatu algoritma untuk mengolah keenam nilai tersebut agar menjadi satu nilai tunggal dimana semua nilai fitur dapat terpenuhi bersamaan dengan optimal. Pada penelitian ini algoritma genetika digunakan untuk menyelesaikan permasalahan tersebut. Dengan nilai tunggal yang dihasilkan, kalimat dapat diurutkan dari yang paling besar nilainya. Nilai mengindikasikan tingkat kepentingan kalimat.

Berbeda dengan metode lainnya yang hanya menggunakan satu solusi untuk mengevaluasi solusi berikutnya, algoritma genetika bekerja pada kumpulan kandidat solusi, sehingga didapatkan ruang pencarian (*search space*) yang lebih luas. Pada proses perangkingan dengan pendekatan algoritma genetika, dilakukan pembobotan keenam nilai fitur. Pembobotan bisa dilakukan dengan fungsi objektif Fan (Kusumadewi, 2005). Fungsi ini digunakan dalam perhitungan fungsi fitness. Hasil dari perangkingan berbasis algoritma genetika adalah nilai tunggal yang optimal untuk masing-masing kalimat.

Dengan adanya ringkasan, diharapkan pengguna dapat dengan mudah memahami makna sebuah teks tanpa harus membaca keseluruhan teks. Pengguna dapat menghindari pembacaan teks yang tidak relevan dengan informasi yang diharapkan oleh pengguna, sehingga didapatkan cara untuk mendapatkan inti dari informasi-informasi tersebut tanpa perlu bersusah payah. Jika peringkasan dilakukan secara manual tentunya diperlukan usaha yang lebih, untuk itu peringkas otomatis menggunakan komputer sangat berguna dalam

menyalin unit-unit teks yang paling penting atau paling informatif dari teks sumber.

Berdasarkan latar belakang yang telah dipaparkan maka dilakukan penelitian yang berjudul **"Peringkasan Multi Dokumen Berbahasa Inggris Berbasis Konten Menggunakan *Single Pass Clustering* dan Perangkingan Berbasis Algoritma Genetika"** agar didapatkan sistem yang dapat meringkas sekumpulan dokumen berbahasa Inggris.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka dapat diuraikan permasalahan pada penelitian ini yaitu sebagai berikut:

1. Bagaimana merancang sebuah sistem yang otomatis meringkas multi dokumen berbahasa Inggris menggunakan *single pass clustering* dan pembobotan dengan algoritma genetika.
2. Bagaimana hasil evaluasi ringkasan multi dokumen berbahasa Inggris dengan tiga parameter yaitu *precision*, *recall*, dan *F-measure*.

1.3 Tujuan Penelitian

Tujuan yang ingin dicapai pada penelitian ini adalah:

1. Merancang sebuah sistem untuk meringkas multi dokumen berbahasa Inggris otomatis menggunakan *single pass clustering* dan perangkingan berbasis algoritma genetika.
2. Menganalisa kinerja sistem peringkas multi dokumen berbahasa Inggris dengan mengevaluasinya menggunakan tiga parameter yaitu *precision*, *recall*, dan *F-measure*.

1.4 Batasan Masalah

Penelitian ini memiliki batasan agar dapat terfokus pada tujuan dan menghindari meluasnya permasalahan. Adapun faktor-faktor yang membatasi penelitian ini yaitu:

1. Sistem ini digunakan untuk dokumen berbahasa Inggris.
2. Inputan dalam sistem ini berupa file dokumen berekstensi .txt yang berisi teks.
3. Tiap file berisi satu dokumen (satu judul) dan minimal terdiri dari satu paragraf.
4. Pada *preprocessing* menggunakan *stemming* Porter.

5. Maksimum jumlah dokumen inputan sebanyak 10 dokumen.
6. Diasumsikan hasil klastering mengikuti teknik single pass, dimana data akan mengklaster sendiri berdasarkan *similarity* dari konten yang dimiliki.
7. Dokumen uji yang tersedia sebanyak 9 dokumen.
8. Tidak boleh ada dokumen yang isinya sama.

1.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini maka akan diperoleh sebuah sistem yang dapat meringkas multi dokumen berbahasa Inggris secara otomatis untuk membantu mendapatkan informasi penting dari dokumen-dokumen tersebut menggunakan *single pass clustering* dan perangkingan berbasis algoritma genetika.

1.6 Metodologi Penelitian

Dalam melakukan penelitian ini, beberapa metode yang digunakan antara lain :

1. Studi Literatur
   Mempelajari berbagai macam literatur tentang *text summarization, multi document summarization,* algoritma genetika, clustering dokumen/teks, dan teori lain yang berkaitan.
2. Pendefinisian dan Analisis Masalah
   Mendefinisikan masalah dan metode-metode yang digunakan juga menganalisisnya kemudian mengimplementasikan metode-metode tersebut dalam masalah yang ada.
3. Implementasi Sistem
   Membuat aplikasi yang dapat meringkas multi dokumen berbahasa Inggris secara otomatis dengan *single pass clustering,* ekstraksi kalimat dan algortima genetika.
4. Uji Coba dan Analisis Hasil Implementasi
   Menguji coba perangkat lunak tersebut dan menganalisa hasil dari implementasi kemudian dievaluasi dengan tiga parameter yaitu *precision*, *recall*, dan *F-measure*.

1.7 Sistematika Penulisan

Sistematika penulisan dari penelitian ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab pendahuluan ini terdiri dari latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Dalam bab ini dibahas mengenai pustaka yang digunakan dalam penelitian ini. Teori-teori yang menjelaskn tentang *text summarization* secara umum, *multi document summarization*, algoritma genetika dan teori-teori lain yang mendukung penyelesaian peringkasan dokumen berbahasa Inggris dengan algoritma genetika.

BAB III METODE DAN PERANCANGAN

Pada bab ini, dijelaskan tahap-tahap meringkas multi dokumen berbahasa Inggris dengan mengimplementasikan algoritma genetika, seperti deskripsi umum sistem, mekanisme kerja sistem dalam deskripsi proses, perancangan uji coba dan evaluasi hasil.

BAB IV PEMBAHASAN DAN ANALISA HASIL

Bab ini menjelaskan proses implementasi sistem dari rancangan penelitian yang dijelaskan pada BAB III, menjelaskan penerapan aplikasi, uji coba, dan analisa hasil percobaan.

BAB V KESIMPULAN DAN SARAN

BAB V berisi kesimpulan dari pembahasan dan saran yang diharapkan bermanfaat untuk pengembangan tugas akhir selanjutnya maupun bagi pihak-pihak yang memerlukan.


UNIVERSITAS BRAWIJAYA

BAB II
TINJAUAN PUSTAKA

Bab ini akan membahas tentang teori-teori yang mendukung penelitian ini. Teori-teori tersebut antara lain penjelasan tentang teks dokumen, *information retrieval* dan *text summarization* beserta teori-teori yang mendukung untuk memperoses teks seperti *preprocessing*, *similarity*, dan pembobotan kata (*term weight*). Selain itu juga akan dijelaskan tentang *single pass clustering* dan algoritma genetika.

2.1 Dokumen Teks

Teks merupakan salah satu sarana untuk menyalurkan informasi. Menurut Luxemburg, dkk, 1989 teks adalah ungkapan bahasa yang menurut isi, sintaksis, dan pragmatik merupakan satu kesatuan. Dari pengertian tersebut dapat diartikan teks merupakan suatu kesatuan bahasa yang memiliki isi dan bentuk untuk menyampaikan pesan tertentu. Teks memiliki isi berupa ide-ide atau amanat yang hendak disampaikan dan bentuk berupa cerita dalam teks yang dapat dibaca dan dipelajari. Teks dalam filologi diartikan sebagai 'tenunan kata-kata', yakni serangkaian kata-kata yang berinteraksi membentuk satu kesatuan makna yang utuh (Permadi, Tedi, 2011).

Elemen-elemen pembentuk teks yaitu :

1. Kata

   Kata merupakan satuan terkecil dari sebuah bahasa, baik bahasa tertulis maupun bahasa lisan yang belum mengungkapkan pikiran yang utuh. Teks termasuk dalam bahasa tertulis. Kata bisa berupa kata dasar/bentuk asal dan kata turunan.

   Setiap kata memiliki dua aspek, yaitu bentuk dan makna. Bentuk merupakan sesuatu yang dapat diinderai seperti dilihat atau didengar, sedangkan makna merupakan sesuatu yang dapat menimbulkan reaksi dalam pikiran kita karena rangsangan bentuk. (Sofyan A.N, Eni K, Wahya, Kostaman J, Yudi P, 2007).
2. Kalimat

   Kalimat merupakan kumpulan kata yang mengungkapkan pikiran yang utuh. Menurut Sofyan A.N, 2007 kalimat adalah bagian dari ujaran yang bisa berdiri sendiri, bermakna, dan

diakhiri oleh intonasi akhir. Sebuah kalimat sekurang-kurangnya memiliki subjek dan predikat.

Subjek merupakan pelaku dalam kalimat yang menjadi pokok pembicaraan dan biasanya berupa kata benda. Sedangkan predikat yaitu tindakan yang dilakukan dalam kalimat tersebut dan biasanya berupa kata kerja. Selain subjek dan predikat, kalimat juga dapat tersusun atas objek dan keterangan. Objek berupa kata benda yang dikenai tindakan oleh subjek, sedangkan keterangan berupa keterangan tempat dan keterangan waktu.

3. Paragraf

Paragraf atau alinea merupakan inti penuangan buah pikiran dalam sebuah karangan. Paragraf terdiri dari beberapa kalimat yang memiliki kalimat utama sebagai ide pokok yang didukung kalimat lainnya dalam paragraf. Kalimat-kalimat ini saling bertalian dalam suatu rangkaian untuk membentuk sebuah gagasan. (Sofyan A.N., Eni K., Wahya, Kostaman J., Yudi P., 2007).

2.2 *Information Retrieval (IR)*

Berikut ini akan dijelaskan mengenai hal-hal yang berkaitan dengan *Information Retrieval* seperti definisi, sistem *Information Retrieval*, dan model *Information Retrieval.*

2.2.1 Definisi *Information Retrieval (IR)*

Pada dasarnya *information retrieval* merupakan proses untuk menentukan dokumen dalam koleksi yang harus ditemubalikkan untuk memenuhi keinginan pengguna akan informasi. (Yogatama, Dani, 2008). Hans Peter Luhn, penemu IBM (*International Business Machines*) adalah perintis dalam menggunakan komputer untuk *information retrieval*. *Information retrieval* yang dalam bahasa indonesia bisa berarti menemubalikkan informasi atau mengambil informasi, dianggap sebagai suatu proses mengambil informasi yang relevan dari penyimpanan informasi. Tujuan dari *Information retrieval* adalah untuk membantu pengguna (*user*) menemukan informasi dengan cepat dari dokumen.

Dalam presentasi seminarnya Dr. Glockner menguraikan beberapa pengertian information retrieval menurut para ahli, diantaranya :

a. Menurut G. Salton information retrieval adalah bidang yang berkaitan dengan struktur, analisis, organisasi, penyimpanan, pencarian dan pengambilan informasi.
b. Menurut G. Klir et al. secara umum information retrival dapat didefinisikan sebagai masalah seleksi informasi dokumenter dari penyimpanan dalam menanggapi pertanyaan-pertanyaan pencarian.
c. Menurut G. Bordogna et al. sistem *information retrieval* memproses informasi permintaan pengguna berdasarkan analisis dari sumber informasi (misalnya dokumen) yang tersimpan dalam arsip. Konten dokumen harus direpresentasikan dalam bentuk yang otomatis terproses.

2.2.2 *Information Retrieval System (IRS)*

Area *Information Retrieval* (IR) dalam cabang Ilmu Komputer meliputi representasi, penyimpanan, pengorganisasian dan pengaksesan informasi dalam data dalam jumlah besar. Representasi dan organisasi informasi harus memberikan kemudahan akses bagi pengguna yang ingin memperoleh informasi. Informasi yang diinginkan pengguna direpresentasikan dalam bentuk *query* yang mengandung *term* (kata) yang digunakan untuk pencarian. *Information Retrieval System* (IRS) adalah sebuah perangkat lunak untuk representasi data, penyimpanan dan pencarian informasi. Sebuah permintaan pencarian kemudian dievaluasi terhadap representasi dokumen internal dan sistem memutuskan apakah dokumen tersebut relevan dan berapa banyak dokumen yang relevan dengan *query*.

Untuk pengguna biasa IRS menyediakan dua fungsi utama yaitu penyimpanan data dan pencarian informasi dalam rangka untuk memenuhi kebutuhan informasi pengguna. Jika IRS yang akan digunakan adalah IRS untuk pencarian informasi, kebutuhan informasi yang yang dituntut harus dinyatakan dalam bahasa tertentu dalam proses disebut *query*. Sistem pencarian merupakan upaya untuk menemukan informasi yang dicari dalam koleksi dokumenter yang relevan dengan *query*. Pada sistem pencarian IRS akan memerintahkan sekumpulan dokumen diambil kemudian ditawarkan pada pengguna. Dokumen yang diperoleh adalah subset dari koleksi dokumenter yang dianggap relevan dengan *query* pengguna oleh sistem aplikasi. Ini berarti sistem harus dapat memaksimalkan subset

dokumen relevan yang diambil dan meminimalkan subset dokumen tidak relevan yang diambil.

Kebanyakan IRS menggunakan *keyword* untuk mengambil dokumen. Pertama, sistem mengekstrak *keyword* dari dokumen, kemudian memberikan bobot pada *keyword* dengan menggunakan berbagai macam pendekatan. Sistem mempunyai dua masalah utama. salah satunya adalah bagaimana mengekstrak *keyword* dengan tepat dan yang lainnya adalah bagaimana menentukan bobot *keyword* tersebut. sebuah IRS pada dasarnya dibentuk oleh tiga komponen utama yaitu sebagai berikut:

1. *The Documentry Database* (database dokumenter).

   Komponen ini menyimpan dokumen dan representasi informasinya. Hal ini berkaitan dengan modul pengindeksan yang secara otomatis menghasilkan representasi setiap dokumen dengan mengestrak isi dokumen. representasi dokumen tekstual biasanya didasarkan pada *index terms* (yang dapat berupa *single term* atau urutan), dimana konten untuk mengidentifikasi dokumen.
2. *The Query Subsystem* (subsistem query).

   Subsistem *query* memungkinkan pengguna untuk merumuskan kebutuhan informasi mereka sehingga sistem dapat menyajikan dokumen yang relevan sesuai kebutuhan pengguna. Untuk melakukan itu, bahasa *query* diproses melalui sekumpulan aturan yang sesuai prosedur untuk mendapatkan dokumen yang relevan.
3. *The Matching Mechanism* (mekanisme yang cocok)

   Mekasnisme yang cocok mengevaluasi sejauh mana representasi suatu dokumen memenuhi persyaratan yang dinyatakan dalam *query*. Selain itu juga mengevaluasi Retrieval Status Value (RSV) atau nilai Status Retrieval dan mengambil dokumen yang dianggap relevan.

Sebuah sistem IR dirancang untuk mendukung kebutuhan informasi dan bertujuan untuk memberikan daftar hasil pemrosesan dokumen yang menjawab kebutuhan informasi pengguna melalui *query* (informasi input). Dalam sistem IR, kebutuhan informasi pengguna dalam dokumen harus diterjemahkan terlebih dahulu dari suatu koleksi/kumpulan *keyword* yang kemudian dilakukan urutan rangking relevansi sesuai query (input *keyword*) pengguna. Dalam

suatu sistem *retrieval* yang efektif, terjalin hubungan yang bagus antara *user task* (tugas pengguna) dengan *logical view* dari dokumen seperti yang dijelaskan berikut ini:

1. User Task

   User task merupakan proses translasi informasi yang dibutuhkan pengguna menjadi suatu bentuk *query* (Yogatama, Dani, 2008). Pengguna harus menerjemahkan kebutuhan informasinya ke dalam bentuk bahasa *query* yang disediakan oleh sistem. Dalam sistem *information retrieval*, secara normal akan menentukan serangkaian kata-kata yang memberikan arti/maksud dari kebutuhan informasi pengguna. Terdapat dua jenis *task* yang dapat dilakukan pengguna RIS, yaitu menggunakan *information retrieval* dan browsing.

2. *Logical View* pada dokumen

   Dokumen yang ditampilkan dari database, dapat diproses melalui *index* atau *keyword*. Dalam menentukan *index* atau *keyword* suatu dokumen dapat diekstrak secara langsung dari teks dalam dokumen atau menspesifikasikan subjek-subjek dalam dokumen oleh pengguna. Kemudian sistem akan memberikan sekumpulan dokumen yang relevan. Dokumen dalam koleksi biasanya direpresentasikan melalui suatu set *term* indeks atau *keyword*. *Term* indeks tersebut menyediakan suatu *logical view* dari dokumen. Proses *logical view* dokumen merupakan proses sistem untuk memberikan dokumen yang dibutuhkan pengguna mulai dari menerjemahkan kebutuhan pengguna, *preprocessing*, mencari melalui indeks atau keyword, mengoleksi dokumen yang relevan, mengurutkan dokumen sesuai relevansinya hingga menampilkan dokumen yang relevan. Untuk dokumen koleksi yang besar, maka sistem mungkin harus mengurangi jumlah set *term* indeks. Hal ini dapat dilakukan dengan membuang *stopwords*, menggunakan *stemming*, ataupun yang lain.

2.2.3 *Information Retrieval Models*

Beberapa model *retrieve* (pengambilan informasi) telah diteliti dan dikembangkan pada area IR. Ada beberapa model IR, diantaranya sebagai berikut:

1. Boolean Model

Dalam model pengambilan Boolean, modul pengindeks melakukan pengindeksan biner dalam arti bahwa suatu *term*

dalam dokumen signifikan atau tidak. Yang dimaksud signifikan yaitu *term* tersebut muncul setidaknya sekali dalam dokumen. *query-query* pengguna dalam model ini diekspresikan menggunakan bahasa query berdasarkan *term-term* dan memungkinkan kombinasi dari kebutuhan pengguna yang sederhana dengan operator logika AND, OR, dan NOT. Hasil yang diperoleh dari pengolahan *query* adalah satu set dokumen yang benar-benar cocok, seperti hanya ada dua kemungkinan yang dipertimbangkan untuk setiap dokumen yaitu relevan atau tidak relevan dengan kebutuhan pengguna yang diwakili oleh *query* pengguna.

2. *Vector space model* (Model Ruang Vektor)

   Dalam model ini, dokumen dipandang sebagai vektor berdimensi n dalam ruang dokumen dan masing-masing *term* mewakili satu dimensi dalam ruang dokumen, dimana n adalah jumlah term khusus yang digunakan untuk menggambarkan isi dari dokumen dalam koleksi. Sebuah *query* juga diperlakukan dengan cara yang sama dan dibangun dari *term-term* dan bobot yang disediakan dalam permintaan pengguna. Pengambilan dokumen didasarkan pada pengukuran *similarity* antara *query* dan dokumen. Ini berari bahwa dokumen yang *similarity*nya dengan *query* lebih tinggi dinilai lebih relevan sehingga harus diambil oleh IRS (*Information Retrieval System*) dalam posisi yang lebih tinggi dalam daftar dokumen yang diambil. Dalam metode ini dokumen yang diambil direpresentasikan pada pengguna dengan teratur sesuai dengan relevansi terhadap *query*.

3. *Probabilistic Model* (Model Probabilistik)

   Model ini mencoba menggunakan teori probabilitas untuk membangun fungsi pencarian dan cara operasinya. Informasi yang digunakan untuk menyusun fungsi pencarian diperoleh dari distribusi *term* indeks seluruh koleksi dokumen atau subsetnya. Informasi ini digunakan untuk mengatur nilai dari beberapa parameter fungsi pencarian yang terdiri dari satu set bobot yang berhubungan dengan *term* indeks.

Menurut Dr. Glockner sebuah sistem IR yang baik harus mampu menerima permintaan (query) pengguna, mengerti kebutuhan pengguna dari *query* input pengguna, mancari ke dalam database

untuk dokumen atau informasi yang relevan saja, mengambil dokumen atau informasi untuk diberikan pada pengguna, merangking dokumen atau informasi berdasarkan urutan relevansinya.

Jadi proses *retrieval* merupakan proses yang kompleks dan dapat dibagi menjadi banyak subproses. Proses pertama adalah pendefinisian database teks. Hal ini dilakukan dengan mengidentifikasi dokumen-dokumen yang akan digunakan, operasi yang akan dilakukan terhadap teks, dan model teks. Model teks yang dimaksud yaitu struktur teks dan elemen mana saja dari teks yang dapat ditemubalikkan. Setelah itu dilakukan *text operations* atau yang lebih dikenal dengan *preprocessing* untuk mentransformasikan dokumen asli menjadi *logical view* dokumen tersebut. Setelah *logical view* dokumen diperoleh, dilakukan pembuatan indeks untuk mempercepat proses pencarian terhadap jumlah data yang besar. Setelah database dokumen diindeks, maka proses kedua dilakukan yaitu inisiasi proses temu balik. Pengguna mendefinisikan kebutuhannya yang kemudian akan ditransformasikan oleh *text operations* yang sama dengan yang digunakan pada koleksi dokumen. *Query* yang diperoleh dari hasil transformasi inilah yang akan diproses untuk mendapatkan dokumen temu balik. Pemrosesan *query* dengan cepat dimungkinkan oleh struktur indeks yang telah dibuat sebelumnya. Setelah itu, dokumen temu balik diurutkan berdasarkan kemungkinan relevansinya.

2.3 Text Summarization (Peringkasan Teks)

Untuk mengambil informasi dari dokumen bisa dilakukan dengan meringkas dokumen. Peringkasan dokumen berperan penting dalam perkembangan *Information Retrieval System* (IRS) yang efektif dan efisien. *Text summarization* atau teks ringkasan merupakan turunan sumber teks yang dimampatkan dengan menyeleksi dan/atau menggeneralisasi konten penting (Bubenhofer, 2002). Sedangkan menurut Okumura, *text summarization* adalah proses untuk mengurangi panjang atau kompleksitas dari text asli, tanpa menghilangkan informasi penting/utama. Dokumen akan di preprocessing terlebih dahulu kemudian direpresentasikan sebagai vektor yang memiliki *term*/bobot dengan nilai tertentu.

Menurut Llorent, Elena berdasarkan bentuknya peringkasan dokumen dibagi menjadi dua yaitu :

1. Extracts
   Extracts merupakan ringkasan yang sepenuhnya terdiri dari kalimat atau urutan kata yang ada dalam dokumen asli. Selain kalimat lengkap, ekstrak dapat berisi frasa dan paragraf.
2. Abstracts
   Abstracts terdiri dari urutan kata yang tidak ada dalam dokumen aslinya. Biasanya dibangun dari konten yang ada namun dengan menggunakan metode canggih. Hal ini umumnya sulit bagi komputer untuk berhasil memecahkan persyaratan pendekatan tersebut karena banyak keterbatasan, termasuk seni dalam generasi bahasa dan kompleksitas bahasa manusia.

Jadi secara singkat perbedaan dari ekstrak dan abstrak adalah ringkasan ekstrak terdiri dari kalimat yang diekstraksi dari dokumen sedangkan ringkasan abstrak berisi kata-kata dan frase yang tidak ada dalam dokumen asli.

Menurut jumlah sumbernya peringkasan teks dapat dibagi menjadi dua, yaitu:

1. Peringkasan Dokumen Tunggal (*Single Document Summarization*)
   Peringkasan dokumen tunggal merupakan proses peringkasan yang sumbernya berasal dari satu dokumen.
2. Peringkasan Multi Dokumen (*Multi Document Summarization*)
   Peringkasan multi dokumen adalah proses peringkasan dokumen yang sumbernya berasal dari banyak dokumen untuk dijadikan sebuah ringkasan yang relevan dengan semua dokumen. Peringkasan multi dokumen dapat dilakukan pada dokumen-dokumen yang memiliki kesamaan topik. Jika dokumen-dokumen inputan memiliki beragam topik, maka sebelum dilakukan peringkasan dokumen-dokumen tersebut harus diklastering dahulu sehingga hasil ringkasannya sebanyak jumlah klaster yang dihasilkan.

Ada dua perbedaan utama antara peringkasan dokumen tunggal dan peringkasan banyak dokumen. Pertama, kebanyakan pendekatan untuk dokumen tunggal mencakup ekstraksi kalimat dari dokumen. Kedua, sebagian besar sistem peringkasan dokumen tunggal sampai batas tertentu, memanfaatkan struktur monolitik dokumen. Sebagai contoh, salah satu cara sederhana yang cukup efektif untuk menulis ringkasan satu dokumen adalah dengan mengambil kalimat pertama dari tiap paragraf dan menempatkannya

bersama-sama dalam urutan aslinya. Tetapi berbeda untuk peringkasan multi dokumen, struktur dari sebuah dokumen tunggal tidak dapat langsung digunakan sedemikian rupa. Dalam pengertian ini, sistem peringkasan multi dokumen biasanya tidak terlalu mengandalkan struktur dokumen. Meringkas sekumpulan dokumen yang terkait secara tematik memiliki beberapa tantangan. Dalam peringkasan multi dokumen, untuk menghindari pengulangan, maka harus mengidentifikasi dan menemukan tema yang tumpang tindih. Selain itu, juga harus memutuskan susunan kalimat ringkasan yang dihasilkan. Untuk menangani potensi ketidakkonsistenan antara dokumen dan bila perlu untuk menyusun peristiwa dari berbagai sumber berbagai metode pendekatan dilakukan untuk membuat sistem peringkasan multi dokumen. Pada tahun 1999, Radev menciptakan metode peringkasan multi dokumen dengan nama SUMMONS. Metode pendekatan ini dilakukan dengan melakukan pendekatan ekstraksi informasi. SUMMONS akan mengklaster dokumen sesuai kontennya, kemudian menerapkan aturan untuk ekstraksi informasi-informasi utama. Sedangkan Barzilay, McKeown, dan Elhadad pada tahun 1997 menciptakan metode pendekatan dengan menguraikan setiap kalimat ke dalam struktur ketergantungan sintaksis (pohon parse sederhana), menggunakan parser yang kokoh kemudian mencocokkan pohon dari seluruh dokumen menggunakan aturan parafrase untuk memproses pohon-pohon yang diperlukan.

2.3.1 *Preprocessing*

Tahap *preprocessing* merupakan tahap di mana sistem melakukan seleksi data pada setiap dokumen dan membersihkan sumber data sehingga didapatkan kata unik atau kata penting. Proses pembersihan yang dimaksud pada konteks peringkasan dokumen adalah menghilangkan kata yang tidak penting dan menghilangkan imbuhan. Setiap dokumen akan dipecah-pecah menjadi kata-kata penyusunnya yang nantinya kata-kata ini diproses hingga menjadi kata unik dan punya makna khusus sehingga bisa menjadi *keyword* yang dijadikan acuan dalam pemrosesan teks. Tahap *preprocessing* pada *information retrieval* bisa terdiri dari:

1. *Case Folding*

   Case Folding merupakan proses untuk mengubah semua huruf pada dokumen menjadi huruf kecil dan menghilangkan tanda baca

untuk mempermudah proses berikutnya sehingga bisa menyamakan makna untuk kata yang sama.

2. *Tokenizing*
   Tokenizing adalah proses memecah dokumen menjadi kata-kata penyusunnya. Tiap kata dipisahkan oleh spasi dan tanpa tanda baca.
3. *Stemming*
   Arti *stem* sendiri adalah akar kata. Jadi, *stemming* merupakan proses pemotongan *affix* (imbuhan), baik *prefix* (awalan) maupun *suffix* (akhiran) dari sebuah *term* menjadi kata dasarnya. Dalam bahasa Inggris menghilangkan imbuhan ini bisa mengubah kata kerja menjadi kata benda. Contoh *stemming* yaitu *connect* merupakan *stem* dari *connected*, *connecting*, *connection*, dan *connections*. *Stemming* dilakukan dengan asumsi bahwa kata-kata yang memiliki *stem* yang sama memiliki makna yang serupa. *Stemming* bisa mengurangi jumlah kata-kata unik dalam indeks sehingga mengurangi kebutuhan ruang penyimpanan untuk indeks dan mempercepat proses sistem juga menghindari ketidakcocokan untuk dokumen yang relevan. Teknik *stemmer* bertujuan untuk mereduksi *term* agar menjadi bentuk akarnya Ada beberapa macam teknik *stemmer*, salah satunya yang paling terkenal untuk bahasa Inggris adalah stemming Porter.
4. *Stop Word Removal*
   Stop Word merupakan daftar kata yang sering muncul dan bukan merupakan kata khusus yang bisa dijadikan patokan atau *keyword* apakah sebuah dokumen relevan atau tidak. *Stop Word* biasanya adalah kata hubung, kata tanya…. Kata-kata tersebut pada tahap selanjutnya akan diproses sehingga proses hanya dilakukan pada kata-kata penting dan tidak memproses kata-kata tidak penting untuk relevansi dokumen.
   Jadi, *Stop Word Removal* adalah proses untuk menghilangkan kata-kata yang ada pada daftar *Stop Word*, sehingga kata-kata yang dihasilkan nanti adalah kata yang spesifik untuk mempercepat dan mengoptimalkan proses *information retrieval*.

2.3.2 Porter *Stemmer*

Stemming Porter atau Porter *Stemmer* adalah teknik *stemming* yang paling umum digunakan untuk dokumen bahasa Inggris. Algoritma Porter diciptakan oelh Martin Porter pada tahun 1980-an.

Dalam konteks *information retrieval*, *stemming* digunakan untuk mereduksi keanekaragaman bentuk *term* agar dapat menghindari ketidakcocokan yang dapat menurunkan nilai *recall*. Teknik ini terdiri dari lima tahap reduksi kata secara berurutan dan hanya dilakukan sekali untuk setiap *term* (kata). Setiap tahap terdapat berbagai aturan yang harus dipilih sesuai kebutuhan suatu *term*. Aturan-aturan Porter *Stemmer* didapatkan dari alamat website http://snowball.tartarus.org/algorithms/porter/stemmer.html. Aturan-aturan Porter *Stemmer* tersebut sebagai berikut:

1. Step 1a : remove plural suffixation
2. Step 1b : remove verbal inflection
3. Step 1b1 : continued for -ed and -ing rules
4. Step 1c : y and i
5. Step 3
6. Step 4 : delete last suffix 4
7. Step 5a : remove e
8. Step 5b : reduction

Beberapa definisi penting untuk algoritma Porter ini adalah:

- Konsonan (c) adalah huruf-huruf selain A, I, U, E, atau O, Konsonan adalah huruf-huruf selain Y yang diawali oleh suatu konsonan. C adalah notasi untuk c>0.
- Vokal (v) adalah huruf-huruf selain konsonan. V adalah notasi untuk v>0.

Notasi penting untuk aturan-aturan pada Porter Stemmer adalah:

- m digunakan untuk menyatakan perulangan huruf atau banyaknya suatu huruf.
- *S adalah *stem* yang berakhir dengan huruf S (notasi * bisa juga digunakan untuk huruf-huruf lain).
- *v* adalah *stem* yang mengandung huruf vokal.
- *d adalah *stem* yang berakhir dengan *double consonant.*
- *o adalah *stem* yang berakhir dengan cvc, dengan c yang kedua bukan W, X, atau Y.
- Aturan dalam tahap-tahap *Porter Stemmer* dinotasikan dengan: (kondisi) S1 → S2

Jika terdapat beberapa kondisi yang dipenuhi dalam satu set aturan, hanya satu aturan yang akan diaktifkan, yaitu aturan dengan S1 paling panjang untuk kata tersebut.

Penjelasan untuk tahapan di atas sebagai berikut :

1) Tahap 1
   a) Tahap ini bertujuan untuk mereduksi *term-term* jamank menjadi bentuk tunggalnya. Aturannya sebagai berikut:
      1. sses → ss
      2. ies → i
      3. ss → ss
      4. s → (hilang)
   b) Tahap ini bertujuan untuk mereduksi *term-term continuous* atau *participle* ke *term* dasar. Aturan yang digunakan sebagai berikut:
      1. Jika (m>0) maka EED → EE
      2. Jika (*v*) maka ED → (hilang)
      3. Jika (*v*) maka ING → (hilang)

      Jika aturan 2 dan 3 pada 1b dijalankan maka langkah berikutnya yaitu aturan-aturan berikut:
      1. AT → ATE
      2. BL → BLE
      3. IZ → IZE
      4. Jika (*d dan bukan (*L or *S or *Z)) → single letter
      5. Jika (m-1) dan (*o) → E
   c) Tahap ini bertujuan untuk mengganti '-Y' menjadi '-I' jika ada huruf vokal lain dalam *term*. Aturan yang digunakan:
      1. Jika (*v) Y maka → I

2) Tahap 2

   Tahap ini digunakan untuk mereduksi *term* yang berakhiran ganda agar menjadi *term* dengan akhiran tunggal. Aturan-aturan yang dipakai sebagai berikut:

   Semua digunakan untuk (m>0)

   1. ATIONAL → ATE
   2. TIONAL → TION
   3. ENCI → ENCE
   4. ANCI → ANCE
   5. IZER → IZE
   6. ABLI →ABLE
   7. ALLI → AL
   8. ENTLI → ENT
   9. ELI → E
   10. OUSLI → OUS
   11. IZATION → IZE
   12. ATION → ATE
   13. ATOR → ATE
   14. ALISM → AL
   15. IVENESS → IVE
   16. FULNESS → FUL
   17. OUSNESS → OUS
   18. ALITI → AL
   19. IVITY → IVE
   20. BILITY → BLE

3) Tahap 3

Aturan-aturan yang digunakan pada tahap ini sebagai berikut:

1. (m>0) ICTATE → IC
2. (m>0) ATIVE → (hilang)
3. (m>0) ALIZE → AZ
4. (m>0) ICITE → IC
5. (m>0) ICAL → IC
6. (m>0) FUL → (hilang)
7. (m>0) NESS → (hilang)

4) Tahap 4

Pada tahap ini dilakukan reduksi untuk *term* yang memiliki nilai m>1. Aturan-aturan yang digunakan sebagai berikut:

1. (m>1) AL → (hilang)
2. (m>1) ANCE → (hilang)
3. (m>0) ENCE → (hilang)
4. (m>1) ER → (hilang)
5. (m>1) IC → (hilang)
6. (m>1) ABLE → (hilang)
7. (m>1) IBLE → (hilang)
8. (m>1) ANT → (hilang)
9. (m>1) EMENT → (hilang)
10. (m>0) MENT → (hilang)
11. (m>1) ENT → (hilang)
12. (m>1 dan (*S atau *T)) ION → (hilang)
13. (m>0) OU → (hilang)
14. (m>0) ISM → (hilang)
15. (m>0) ATE → (hilang)
16. (m>0) ITI → (hilang)
17. (m>0) OUS → (hilang)
18. (m>0) IVE → (hilang)
19. (m>0) IZE → (hilang)

5) Tahap 5

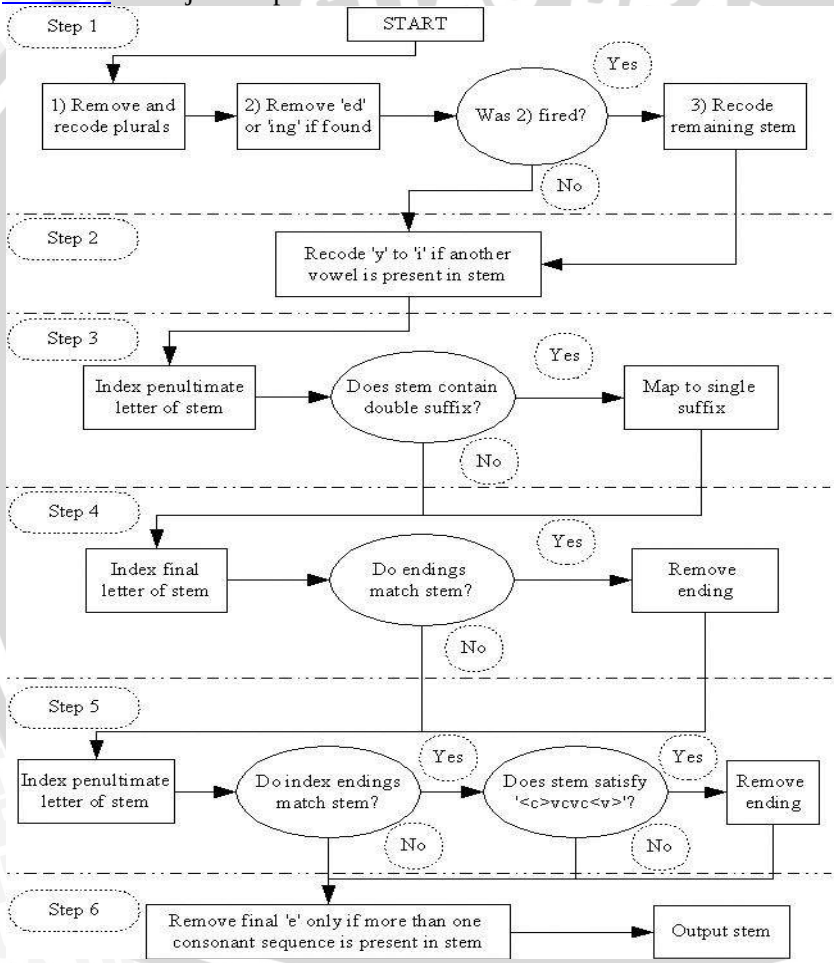
Tahap terakhir ini bertujuan untuk penyesuaian terhadap *stem* yang dihasilkan. Aturan-aturannya yaitu:

a) Membuang huruf terakhir ‘-E’ jika (m>1) atau (m=1) dan bukan *o. Aturannya sebagai berikut:

1. (m>1) E → (hilang)
2. (m=1 dan bukan *o) E → (hilang)

b) Tahap ini untuk menangani duplikasi huruf terakhir. Aturannya yaitu:
1. Untuk (m>1 dan *d dan *L) → hurufnya satu

Flowchart Porter *Stemmer* yang diunduh dari http://www.comp.lancs.ac.uk/computing/research/stemming/Links/porter.htm ditunjukkan pada Gambar 2.1 berikut:



Gambar 2.1 Flowchart Porter Stemmer

2.3.3 Model Ruang Vektor (*Vector Space Model*)

Pada sebuah dokumen terdapat sejumlah n kata yang berbeda sebagai indeks kata (*terms index*). Kata-kata ini akan membentuk ruang vektor yang memiliki dimensi sebesar n. Setiap kata i dalam dokumen diberikan bobot (w), sehingga dokumen direpresentasikan sebagai vektor berdimensi n.

Penentuan relevansi dokumen dengan *query* dipandang sebagai proses pengukuran kesamaan (*similarity document*) antara vektor dokumen dengan vektor *query*. Semakin tinggi nilai kesamaan suatu vektor dokumen dengan vektor *query* maka dokumen dipandang semakin relevan dengan *query*. Pengukuran yang dapat menghitung kesesuaian yang baik adalah dengan memperhatikan perbedaan arah dari kedua vektor tersebut. Perbedaan arah ini akan menghasilkan jarak diantara kedua vektor. Terdapat berbagai macam metode untuk menghitung jarak yang disebut *Similarity Measure* ini. Salah satunya adalah *Cosine Similarity Measure*.

Cosine Similarity Measure

Cosine Similarity Measure merupakan metode perhitungan jarak antara dua vektor yang menghasilkan sudut cosine diantara keduanya. Metode perhitungan cosine similarity secara umum ditunjukkan oleh persamaan 2.1 berikut:

$$Sim\left(V_i, V_j\right) = \frac{\sum_{kata\ \ yang\ \ sama} w_{ik} \times w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2} \times \sqrt{\sum_{k=1}^{n} w_{jk}^2}} \qquad (2.1)$$

Keterangan:

w_{ik} : bobot kata yang sama di vektor i

W_{ik} : bobot kata yang sama di vektor j

2.3.4 Pembobotan Kata (*term weight*)

Koleksi dokumen direpresentasikan dalam ruang vektor sebagai matrik kata-dokumen (*term-document matrix*). Nilai dari elemen matrik w_{ij} adalah bobot kata i dalam dokumen j. Salah satu cara untuk memberi bobot terhadap suatu kata adalah memberikan nilai sesuai jumlah kemunculan kata atau frekuensi kata. Faktor lain yang diperhatikan dalam pemberian bobot yaitu kejarangmunculan kata dalam dokumen koleksi. Pembobotan ini akan memperhitungkan faktor kebalikan frekuensi dokumen yang

mengandung kata tertentu dan disebut dengan *inverse document frequency*. Sebagai tambahan adalah faktor normalisasi terhadap panjang dokumen. Dokumen dalam koleksi memiliki panjang yang beragam. Ketimpangan terjadi karena dokumen yang panjang cenderung mempunyai frekuensi kata yang besar. Sehingga untuk mengurangi ketimpangan tersebut diperlukan faktor normalisasi pada pembobotan.

2.3.4.1 TF.ISF

TS.ISF adalah perkalian antara t*erm frequency dengan inverse sentence frequency*. Ide dasar dari penilaian tf.isf adalah mengevaluasi setiap kata dalam distribusinya pada semuruh kalimat di dokumen. Jadi nilai tf.isf ditentukan untuk mengevaluasi pentingnya sebuah kata dalam dokumen bedasarkan frekuensinya dalam sebuah kalimat dan distribusinya di seluruh kalimat dalam dokumen. Persamaannya ditunjukkan oleh persamaan 2.2 berikut ini:

$$O_k = tf_k \times isf_k = t_{fk} \times \log\frac{N}{n_k} \quad (2.2)$$

O_k : bobot kata ke-k
tf_k : frekuensi dari bobot kata k pada dokumen
N : jumlah seluruh kalimat
n_k : jumlah kalimat yang mengandung kata k

2.3.4.2 TF.IDF

TF.IDF adalah perkalian antara *term frequency dengan inverse document frequency*.Variabel TF merupakan jumlah suatu term/kata dalam suatu dokumen, sedangkan IDF merupakan invers document frequency dari sebuah term/kata. Dengan menggunakan bobot TF-IDF, sebuah dokumen dapat dimodelkan sebagai sebuah vektor. Dokumen D_i dapat dimodelkan atas komponen T_i sehingga jika seluruh dokumen dikumpulkan maka akan terbentuk matriks term-dokumen dengan nilai bobot term/TF-IDF sebagai nilainya. Matriksnya sebagai berikut:

$$\begin{array}{c} \\ D_1 \\ D_2 \\ \vdots \\ D_n \end{array} \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & \dots & T_t \end{array} \\ \begin{bmatrix} W_{11} & T_{12} & \dots & W_{1t} \\ W_{21} & T_{22} & \dots & W_{2t} \\ \vdots & \vdots & & \vdots \\ W_{n1} & W_{n2} & \cdots & W_{nt} \end{bmatrix} \end{array}$$

Persamaan TF.IDF ditunjukkan oleh persamaan 2.3 berikut:

$$W_{dt} = tf_{dt} \times IDF \quad (2.3)$$

Untuk menghitung IDF digunakan persamaan 2.4 berikut:

$$IDF = log \frac{N}{df_t} \quad (2.4)$$

TD.IDF yang dinormalisasi ditunjukkan oleh persamaan 2.5 berikut:

$$W_{dt} = \frac{tf_{dt} \times log \frac{N}{df_t}}{\sqrt{\sum_{k=1}^{n} s_{dt}^2}} \quad (2.5)$$

W_{dt} : Nilai bobot kata ke – t pada dokumen ke – d
tf_{dt} : Jumlah kata/frekuensi kata ke – t pada dokumen ke – d
N : Banyak dokumen
df_t : Jumlah dokumen yang mengandung kata t
s_{dt} : Nilai dari TF.IDF kata ke – t pada dokumen ke – d

2.3.5 Evaluasi *Text Summarization*

Ketika mengevaluasi sistem IR (*information retrieval system*) maka fokus ditujukan pada kecepatan proses pencarian, kenyamanan pengguna, kemungkinan query, hasil dan kemampuan dalam mengambil informasi yang relevan. Proses evaluasi hasil *text summarization* dilakukan menggunakan tiga parameter. Ketiga parameter tersebut adalah *precision*, *recall*, dan *F-measure*. Sebuah sistem informasi dikatakan baik jika tingkat *precision, recall, dan F-measure*nya tinggi. Hovy, 2003 mengevaluasi ringkasan otomatis hasil sistem dengan dibandingkan ringkasan hasil manusia. Parameter yang digunakan Hovy adalah *Precision* dan *Recall*.

1. *Precision*

   Precision adalah jumlah objek yang dapat dikenali dengan benar oleh sistem dibagi jumlah semua objek yang dikenali sistem. Persamaan *precision* ditunjukkan pada persamaan 2.6 berikut :

   $$P = \frac{jumla\ h\ relasi\ \ sistem\ \ yang\ \ benar}{jumla\ h\ relasi\ \ sistem} \quad (2.6)$$

   Hovy mendefinisikan persamaan untuk precision seperti pada persamaan 2.7 berikut:

$$P = \frac{correct}{(correct + wrong)} \quad (2.7)$$

Keterangan:
correct : jumlah kalimat yang diekstrak oleh sistem dan manusia.
wrong : jumlah kalimat yang diekstrak oleh sistem tetapi tidak diekstrak oleh manusia.

2. *Recall*
Recall adalah jumlah objek yang dikenali dengan benar oleh sistem dibagi dengan jumlah objek yang seharusnya dikenali. Persamaan *recall* ditunjukkan pada persamaan 2.8 berikut:

$$R = \frac{jumla\ h\ relasi\ \ sistem\ \ yang\ \ benar}{jumla\ h\ relasi\ \ yang\ \ benar\ \ pada\ \ dokumen\ \ key} \quad (2.8)$$

Hovy mendefinisikan persamaan recall seperti pada persamaan 2.9 berikut:

$$R = \frac{correct}{(correct + missed)} \quad (2.9)$$

Keterangan:
missed : jumlah kalimat yang diekstrak oleh manusia tetapi tidak diekstrak oleh sistem.

3. *F-Measure* atau *Overall Fitness*
F-Measure menggambarkan hubungan antara *recall* dan *precision*. Persamaan *F-Measure* seperti pada persamaan 2.10 berikut:

$$F = \frac{R*P}{0.5\ (R+P)} \quad \text{atau} \quad F = \frac{2*R*P}{(R+P)} \quad (2.10)$$

Nilai F-measure antara 0.0 sampai 1.0, dengan 0.0 mengidentifikasi hasil paling buruk dan 1.0 adalah hasil sempurna. Nilai F-measure mengidentifikasi hasil IR yang terdiri dari informasi tidak penting, disebut *precision* dan hasil IR yang tidak terdiri cukup informasi yang disebut *recall*.

Hubungan antara *correct*, *missed*, dan *wrong* dapat dilihat pada tabel 2.1 berikut:

Tabel 2.1 Hubungan *Correct*, *Missed*, dan *Wrong*

Hasil ringkasan manual	Hasil ringkasan sistem	
	Ada	Tidak
Ada	correct	missed
Tidak	wrong	-

Precision dapat didefinisikan sebagai ukuran ketepatan, sedangkan recall adalah ukuran kelengkapan. Secara sederhana, recall dari suatu dokumen tinggi berarti tidak ada informasi yang terlewat tetapi ada banyak informasi yang tidak berguna yang ikut tersaring (precision rendah). Precision suatu dokumen tinggi berarti bahwa seluruh hasilnya relevan, meski tidak mungkin menemukan semua informasi yang relevan (recall rendah).

2.4 Single Pass Clustering

Algoritma *Single Pass Clustering* dapat dilakukan dengan langkah-langkah sebagai berikut :

1. Masukkan D1 (dokumen pertama) menjadi C1 (Cluster pertama).
2. Untuk D_i (dokumen ke-i) di hitung kesamaan (*similarity*) dengan semua dokumen yang sudah terklaster. Misalnya dengan *cosine similarity*. Hitung *maximum similarity* (Smax).
3. Jika Smax (*Maximum Similarity*) lebih besar dari TV (batas nilai/*threshold value*), maka dokumen ditambahkan pada klaster yang bersesuaian. Jika sebaliknya, maka Di digunakan untuk inisialisasi klaster baru.
4. Jika masih ada Di yang belum terklaster, kembali ke langkah ke-2.

Kelebihan algoritma ini adalah sederhana dan hanya membutuhkan satu kali proses untuk mengklaster semua dokumen. Sedangkan kelemahannya yaitu dapat menghasilkan klaster yang besar pada proses awal dan klaster yang terbentuk bergantung pada urutan input dokumen.

2.5 Algoritma Genetika

Berikut ini akan dijelaskan mengenai definisi, struktur umum serta komponen-komponen dalam algoritma genetika.

2.5.1 Pengertian Algoritma Genetika

Algoritma genetika merupakan keluarga dari model komputasional yang terispirasi oleh evolusi dan mencoba untuk meniru proses evolusi Darwin dalam program komputer. Algortima ini mengkodekan solusi potensial ke suatu masalah yang spesifik pada suatu kromosom dan mengenakan operator rekombinasi. Algoritma genetika adalah algoritma komputasi evolusioner pencarian yang didasarkan pada mekanisme seleksi alami dan evolusi biologis. Algoritma genetik mampu memecahkan masalah menggunakan operator genetik sehingga membentuk suatu mekanisme yang cocok untuk berbagai masalah pencarian. Algoritm ini pertama kali diperkenalkan oleh John Holland sekitar tahun 1970-an. Solusi yang mungkin direpresentasikan dan yang 'terkuat' akan memiliki kesempatan terbesar untuk bereproduksi.

Dalam sistem evolusi, populasi berevolusi dengan tekanan selektif, kawin antara individu, dan perubahan seperti mutasi. Dalam algoritma genetika, solusi yang direpresentasikan dalam populasi dikenai seleksi, operator genetika, dan evaluasi fungsi kemudian didapat individu baru yang bergabung kembali dengan individu awal dan mengubah solusi dalam populasi untuk menciptakan populasi baru. Populasi akan terus dioptimalkan sampai generasi yang diinginkan tercapai.

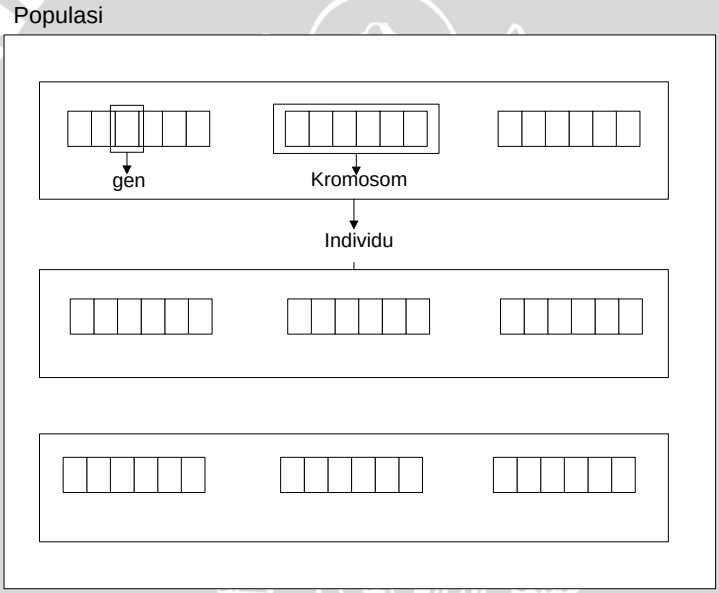
Berbeda dengan teknik pencarian konvensional, algoritma genetika dimulai dari himpunan solusi yang dihasilkan secara acak yang disebut dengan populasi. Setiap individu merepresentasikan sebuah solusi dari permasalahan yang diangkat. Individu-individu berevolusi dalam suatu proses iterasi yang berkelanjutan yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan suatu fungsi evaluasi (Gen dan Cheng,1997). Setelah beberapa generasi maka algoritma genetik akan konvergen pada kromosom terbaik, yang diharapkan merupakan solusi optimal.

2.5.2 Definisi Penting

Pada algoritm genetika terdapat beberapa istilah penting, yaitu sebagai berikut :

1. **Genotype (Gen),** yaitu sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa berupa nilai biner, float, integer maupun karakter.
2. **Allele**, Allele adalah nilai dari gen.

3. **Kromosom**, kromosom merupakan gabungan gen-gen yang membentuk nilai tertentu.
4. **Individu**, individu ini menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat.
5. **Populasi**, populasi merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
6. **Generasi**, generasi menyatakan satuan siklus proses evolusi.
7. **Nilai Fitness**, menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.
8. **Offspring,** adalah kromosom baru yang dihasilkan setelah melewati suatu generasi



Gambar 2.2 Perbedaan Populasi, Individu, Kromosom, Gen

2.5.3 Struktur Umum

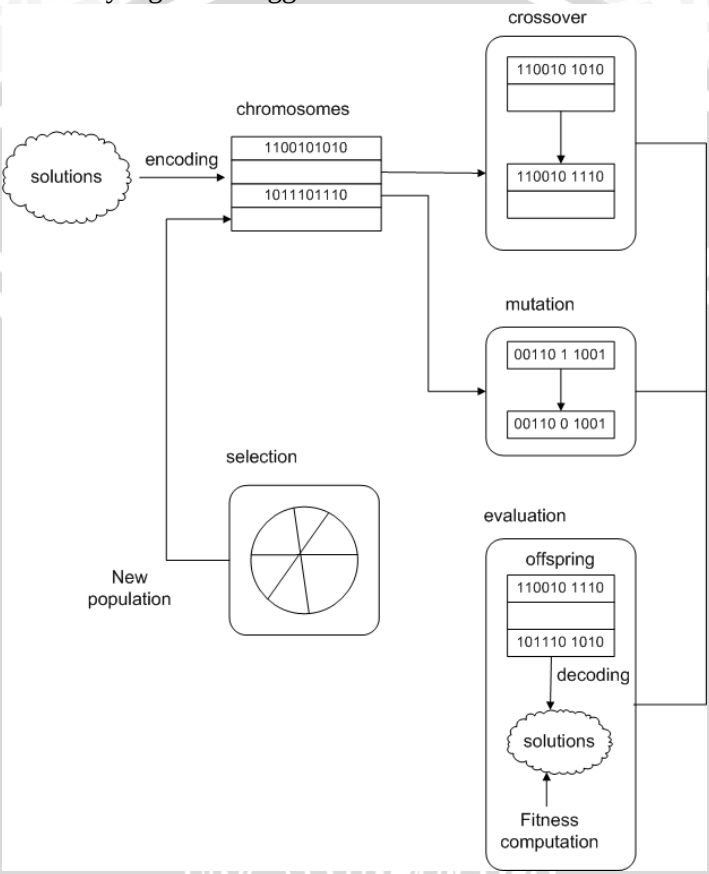
Teknik pencarian dilakukan atas sejumlah solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi biasa disebut dengan kromosom atau bisa juga dalam satu individu terdapat banyak kromosom yang

memiliki banyak gen (pada pengkodean biner). Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi berikutnya merupakan hasil evolusi melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi fitness. Nilai fitness dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) terbentuk dari proses penyilangan (*crossover*) antara dua induk (*parent*) dan proses mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai fitness dari induk pada awal populasi dan nilai fitness dari anak/individu baru yang telah mengalami *crossover* atau mutasi, kemudian mereduksi individu dengan fitness rendah sampai didapatkan sejumlah individu sama dengan jumlah individu awal sehingga ukuran populasi (jumlah individu dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

Secara sistematis struktur algoritma genetik dapat didefenisikan dengan langkah-langkah sebagai berikut:

1. Membangkitkan populasi awal
   Populasi awal ini dibangkitkan secara random sehingga didapatkan solusi awal. Populasi terdiri atas sejumlah individu yang merepresentasikan solusi yang diinginkan.
2. Membentuk generasi baru
   Untuk membentuk generasi baru, dilakukan proses seleksi, crossover dan mutasi. Proses ini dilakukan berulang-ulang sampai batas iterasi atau terpenuhi kriteria berhenti agar didapatkan individu baru untuk membentuk generasi baru dimana generasi baru ini diharapkan dapat memberikan solusi terbaik yang optimal.
3. Evaluasi solusi
   Pada tiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang dinamakan fitness. Nilai fitness suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai fitness setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Bila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah 2. Kriteria berhenti sering digunakan

antara lain: berhenti pada generasi tertentu, berhenti setelah dalam beberapa generasi berturut-turut nilai fitness tertinggi tidak berubah, dan berhenti jika dalam suatu generasi tidak didapatkan nilai fitness yang lebih tinggi.



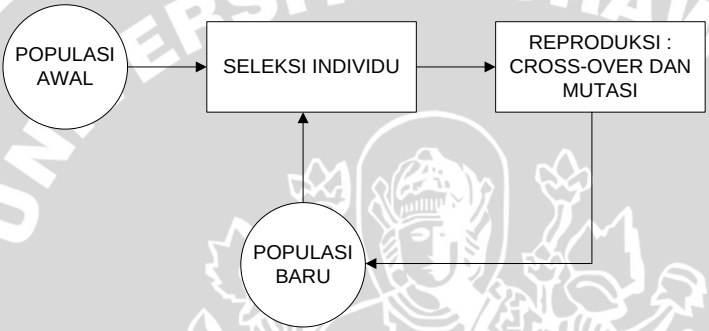
Gambar 2.3 Proses Algoritma Genetika

2.5.3.1 Algoritma Genetika Sederhana

Misalkan P(generasi) adalah populasi dari satu generasi, maka langkah-langkah algoritma genetika secara sederhana sebagai berikut :

1. Generasi = 0 (generasi awal).
2. Inisialisasi populasi awal secara acak.
3. Evaluasi nilai fitness pada setiap individu dalam P(generasi).
4. Generasi = generasi+1 (tambah generasi).

5. Seleksi populasi tersebut untuk mendapatkan kandidat induk, P'(generasi).
6. Lakukan crossover pada P'(generasi).
7. Lakukan mutasi pada P'(generasi).
8. Lakukan evaluasi fitness setiap individu pada P'(generasi).
9. Bentuk populasi baru: P(generasi) = {P(generasi-1) yang survive, P'(generasi)}
10. Jika generasi < maksimum generasi, maka kembali ke langkah 4.



Gambar 2.4 Siklus algoritma Genetika

2.5.3.2 Parameter Kontrol

Pada algoritma genetika diperlukan beberapa parameter, sebagai berikut :

1. Ukuran populasi (popsize)
   Ukuran populasi menunjukkan jumlah kromosom atau individu yang terdapat dalam populasi (dalam satu generasi). Jika hanya sedikit kromosom dalam populasi maka algoritma genetik akan mempunyai sedikit variasi kemungkinan untuk melakukan *crossover* antara orangtua karena hanya sebagian kecil dari *search space* yang dipakai. Jumlah populasi digunakan sebagai batas jumlah seleksi populasi setelah proses persilangan (*crossover*) dan mutasi.
2. Peluang *crossover* (pc)
   Menunjukkan kemungkinan *crossover* terjadi antara 2 kromosom. Jika tidak terjadi *crossove*r maka keturunannya akan sama persis dengan kromosom orangtua, tetapi tidak berarti generasi yang baru akan sama persis dengan generasi yang lama. Jika

probabilitas *crossover* 100% maka semua keturunannya dihasilkan dari *crossover*. *Crossover* dilakukan dengan harapan bahwa kromosom yang baru akan lebih baik.

3. Peluang mutasi (pm)
   Menunjukkan kemungkinan mutasi yang terjadi pada gen-gen penyusun kromosom. Jika probabilitas mutasi 40%, maka diharapkan 40% dari populasi mengalami mutasi.
4. Bisa juga ditambahkan maksimum iterasi
   Yang dimaksud dengan menentukan maksimum iterasi adalah menentukan jumlah populasi atau banyaknya generasi yang dihasilkan yang digunakan sebagai batas akhir proses seleksi, persilangan, dan mutasi.

2.5.3.3 Prosedur Inisialisasi

Parameter kontrol algoritma genetika yang harus ditentukan di awal yaitu ukuran populasi, probabilitas crossover, dan probabilitas mutasi. Ukuran populasi menjelaskan berapa banyak individu yang ada dalam suatu generasi. Jika terlalu sedikit pencarian tidak memiliki ruang solusi yang luas dan banyak kesempatan untuk menemukan solusi terbaik. Semakin besar probabilitas crossover semakin banyak peluang individu baru yang dihasilkan dalam suatu generasi. Sedangkan untuk probabilitas mutasi, semakin besar probabilitas mutasi maka semakin besar peluang keanekaragaman gen baru. Prosedur inisialisasi atau prosedur awal algoritma genetika adalah sebagai berikut :

1. Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan.
2. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut.
3. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

2.5.4 Komponen Utama Algoritma Genetika

Komponen utama algoritma genetika antara lain teknik pengkodean kromosom, seleksi, *crossover*, mutasi, parameter kontrol, fungsi evaluasi dan lainnya seperti yang dijelaskan berikut ini:

2.5.4.1 Teknik Pengkodean (*Encoding*)

Hal paling mendasar dalam algoritma genetika adalah penyandian atau representasi kromosom. Algoritma genetika tidak memproses penyelesaian atau solusi asli dari suatu masalah tetapi memproses penyelesaian yang telah di representasikan. Menurut Mitchell, 1999 penyandian kromosom adalah suatu cara untuk menyatakan kandidat solusi asli suatu masalah ke dalam suatu kromosom dalam sebuah populasi.

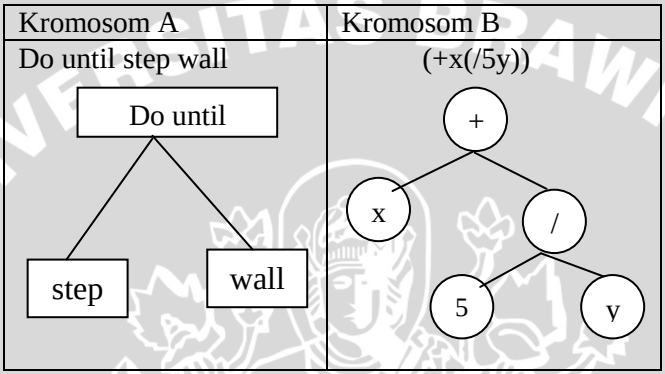
Representasi kromosom merupakan proses awal yang harus dilakukan pada algoritma genetika untuk mendefinisikan individu yang menyatakan salah satu solusi yang mungkin dari suatu permasalahan yang diangkat. Sebuah kromosom harus mengandung atau mewakili informasi mengenai solusi dari masalah yang akan diselesaikan. Merepresentasikan kromosom dilakukan dengan mendefinisikan jumlah dan tipe gen yang digunakan. Teknik pengkodean yang biasa dipakai adalah sebagai berikut:

1. Pengkodean Biner
   Teknik pengkodean biner adalah pengkodean kromosom dimana setiap gen direpresentasikan dalam barisan string bit 1 atau 0. Pengkodean ini yang pertama kali digunakan oleh John Holland.
   Contoh pengkodean biner yaitu :
   Kromosom A : 110010011110001
   Kromosom B : 011100110001001

2. Pengkodean Nilai (*Value Encoding*)
   Pengkodean nilai merepresentasikan kromosom ke dalam kumpulan nilai yang berbentuk huruf atau angka. Contoh pengkodean nilai yaitu:
   Kromosom A : 1.23 4.56 2.34
   Kromosom B : AC BD EG
   Kromosom C : 2 4 7
3. Pengkodean Permutasi
   Teknik pengkodean ini digunakan dalam masalah yang berhubungan dengan pengurutan data atau pengurutan tugas. Pada pengkodean permutasi, setiap kromosom terdiri atas barisan angka yang merepresentasikan angka dalam sebuah urutan. Contoh representasi pengkodean permutasi:

Kromosom A : 0 1 2 3 4 5 6 7
Kromosom B : 4 6 0 2 7 1 3 5

4. Pengkodean Pohon (*Tree Encoding*)
   Pengkodean pohon adalah pengkodean yang biasanya digunakan untuk menyusun program atau fungsi dalam *genetic programming*. Contoh representasi pengkodean pohon sebagai berikut:



Gambar 2.5 Pengkodean Pohon

Pada teknik ini setiap kromosom merupakan pohon setiap fungsi dalam bahasa pemrograman.

2.5.4.2 Fungsi Evaluasi

Ada dua hal yang harus dilakukan dalam melakukan evaluasi kromosom yaitu evaluasi fungsi objektif dan konversi fungsi objektif ke dalam fungsi fitness. Fungsi evaluasi memberikan penilaian pada kromosom yang disebut nilai *fitness*. Fungsi *fitness* merupakan ukuran kinerja suatu individu agar tetap bertahan hidup dalam lingkungannya. Jika solusi yang dicari adalah untuk maksimasi permasalahan, maka *fitness* sama dengan fungsi objektifnya. Sebaliknya, jika solusi yang dicari adalah untuk minimasi, maka nilai *fitness* sama dengan invers dari nilai objektifnya. Fungsi *fitness* tergantung pada permasalahan dari representasi yang digunakan. Fungsi *fitness* digunakan untuk proses evaluasi kromosom agar memperoleh kromosom yang diinginkan. Fungsi ini membedakan kualitas kromosom untuk mengetahui seberapa baik kromosom yang

dihasilkan. Nilai *fitness* adalah nilai yang menyatakan baik tidaknya suatu individu. Semakin besar nilai *fitness* yang dimiliki suatu individu, maka semakin besar pula kesempatan individu tersebut untuk tetap bertahan atau berkembangbiak. Algoritma genetika bertujuan mencari individu dengan nilai *fitness* yang paling tinggi.

Pada penelitian ini, fungsi objektif didapat dengan menggunakan rumus Fan, seperti pada persamaan 2.11 berikut :

Minimumkan $Z = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(max\ b_j^m - b_{ij}\right)^2 w_j^2$ (2.11)

Keterangan :
i : kalimat ke-i
j : fitur ke-j
m : jumlah kalimat
n : jumlah fitur
b_{ij} : matrik keputusan (dalam kasus ini matrik kalimat × fitur)
max b_j : nilai maksimum tiap fitur dari semua kalimat
w_j : bobot fitur ke-j

Dengan batasan :
$\sum w_j = 1$ dan $w_j \geq 0$

Karena fungsi tersebut fungsi minimum sehingga untuk memperoleh nilai Fitness, digunakan persamaan 2.12 berikut :

$$F = \frac{1}{\sum_{i=1}^{m} \sum_{j=1}^{n} \left(max\ b_j^m - b_{ij}\right)^2 w_j^2} \quad (2.12)$$

2.5.4.3 Seleksi

Proses seleksi dalam algoritma genetika digunakan untuk memilih individu-individu terbaik yang berpeluang pada proses kawin silang dan mutasi pada generasi berikutnya. Dari induk yang baik diharapkan menghasilkan keturunan yang baik di generasi berikutnya. Pada proses seleksi nilai fitness diperhitungkan. Proses seleksi bertujuan untuk memberikan kesempatan reproduksi bagi anggota populasi. Ada beberapa metode seleksi antara lain :

1. Rank-based Fitness
   Pada proses seleksi Rank-based Fitness populasi diurutkan menurut nilai fitnessnya.
2. Roulette wheel selection

Nama lain dari Roulette wheel selection adalah stochastic sampling with repalcement. Tiap individu memiliki area yang besarnbya berbanding lurus dengan nilai fitnessnya. Individu-individu digambarkan ke dalam segmen/area secara berurutan dalam suatu lingkaran yang disebut roda roulette. Sebuah bilangan random dibangkitkan dan individu yang memiliki segmen dimana bilangan random tersebut berada akan terseleksi. Proses ini dilakukan berulang hingga didapatkan sejumlah individu yang diharapkan. Individu yang memiliki nilai fitness terbaik mempunyai kesempatan lebih besar untuk terpilih sebagai orang tua sesuai luas areanya.

3. *Stochastic universal sampling*

   Pada *Stochastic universal sampling* individu-individu dipetakan dalam suatu segmen garis secara berurut sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran fitnessnya seperti halnya pada seleksi roda roulette. Kemudian diberikan sejumlah pointer sebanyak individu yang ingin diseleksi pada garis tersebut. Jarak antar pointer sama untuk semua pointer sejumlah individu. Andaikan N adalah jumlah individu yang akan diseleksi, maka jarak antar pointer adalah 1/N, dan posisi pointer pertama diberikan secara acak pada range [1, 1/N].

4. *Truncation selection*

   Seleksi *Truncation* adalah seleksi buatan. Digunakan oleh populasi yang jumlahnya sangat besar. Pemilihan kromosom dilakukan secara acak tetapi tidak semua kromosom mendapatkan kesempatan untuk seleksi, hanya kromosom-kromosom terbaik saja yang berpeluang. Individu-individu diurutkan berdasarkan nilai fitnessnya. Hanya individu yang terbaik saja yang akan diseleksi sebagai induk. Parameter yang digunakan adalah suatu nilai ambang trunc untuk menentukan ukuran populasi yang akan diseleksi sebagai induk yaituberkisar antara 50% - 10%. Individu-individu yang ada dibawah nilai ambang ini tidak akan menghasilkan keturunan.

5. *Tournament Selection*

   Pada *Tournament Selection* ditetapkan suatu nilai tour untuk individu-individu yang dipilih secara random dari suatu populasi. Individu-individu yang terbaik dalam kelomppok ini akan

diseleksi sebagai induk. Parameter yang digunakan adalah ukuran tour yang bernilai antara 2 sampai N (jumlah individu dalam populasi).

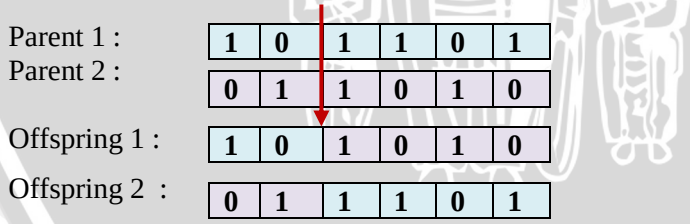
2.5.4.4 Rekombinasi (*Crossover*)

Operasi *crossover* atau yang lebih dikenal dengan kawin silang atau pindah silang merupakan operasi yang menggabungan dua kromosom *parent* agar menghasilkan kromosom anakan sebagai individu baru. Rekombinasi bertujuan menambah keanekaragaman individu dalam populasi. Jumlah kromosom dalam populasi yang mengalami *crossover* dikendalikan oleh suatu parameter yang disebut probabilitas crossover (P_c). Jadi akan dibangkitkan bilangan random antara 0-1 sebanyak jumlah individu. Jika bilangan random kurang dari P_m maka individu tersebut akan diubah nilainya. Beberapa jenis rekombinasi adalah sebagai berikut :

1. Crossover satu titik

   Pada crossover satu titik, penyilangan dimulai dengan memilih posisi titik potong (t) secara acak antara 1 sampai N-1 (panjang kromosom dikurangi satu). Kemudian posisi titik potong yang terpilih sampai akhir dari kromosom ditukar antara orang tua pertama dan kedua. Proses yang demikian dinamakan operator crossover satu titik seperti diperlihatkan pada gambar berikut:

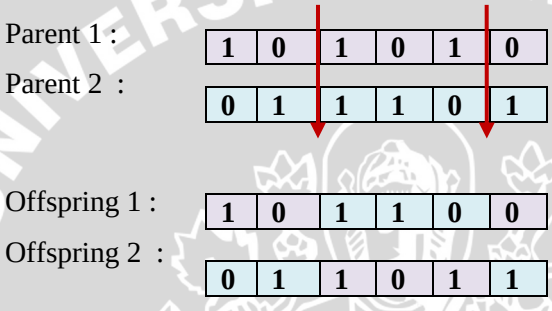
   Titik potong berada pada posisi dua, maka posisi ketiga sampai terakhir akan ditukar antara parent 1 dan parent 2.



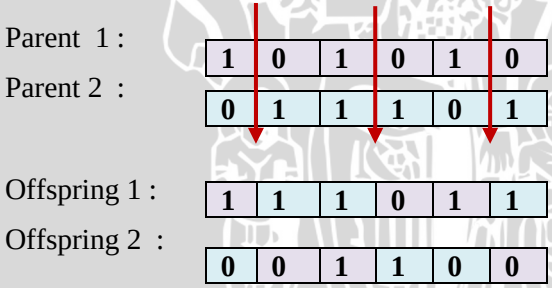
2. Crossover banyak titik

   Pada crossover ini dipilih beberapa titik potong secara acak antara 1 sampai N (jumlah gen dalam kromosom), kemudian pewarisan gen-gen dilakukan secara menyilang (zig-zag). Crossover dua titik juga termasuk crossover banyak titik. Proses

crossover ini dilakukan dengan memilih dua titik crossover. Kromosom keturunan kemudian dibentuk dengan barisan bit dari awal kromosom sampai titik crossover pertama disalin dari orangtua pertama, bagian dari titik crossover pertama dan kedua disalin dari orangtua kedua, kemudian selebihnya disalin dari orangtua pertama lagi. Untuk crossover dua titik ilustrasinya sebagai berikut :



Untuk crossover tiga titik ditunjukkan oleh gambar berikut:



3. *Crossover* seragam

   Crossover seragam manghasilkan kromosom keturunan dengan menyalin bit-bit secara acak dari kedua orangtuanya. Seperti jika dilakukan dengan pelemparan uang logam. Jika sisi 1 yang terlihat maka *offspring* 1 mendapatkan bit dari parent 1. Jika dilempar koin yang terlihat sisi 2 maka bit didapat dari pewarisan parent 2 dan dilakukan sepanjang kromosom.

4. *Crossover* untuk representasi permutasi

Crossover representasi permutasi atau juga dikenal *crossover* untuk menyusun kromosom banyak digunakan dalam masalah penjadwalan seperti *travelling salesman problem* (TSP), job shop, dan lain-lain. Pada TSP representasi kromosom berupa urutan kota.

2.5.4.5 Mutasi

Mutasi merupakan proses mengubah nilai satu atau lebih gen dalam suatu kromosom. Proses ini dilakukan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi berikutnya dengan fitness yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan. Operator mutasi juga dapat digunakan untuk menghindari konvergensi prematur pada operator crossover.

Proses mutasi ini bersifat acak sehingga tidak selalu menjamin bahwa setelah proses mutasi akan diperoleh kromosom dengan fitness yang lebih baik. Operator mutasi dikendalikan oleh sebuah parameter yang disebut Probabilitas Mutasi (P_m). Jadi akan dibangkitkan bilangan random antara 0-1 sebanyak jumlah gen. Jika bilangan random kurang dari P_m maka gen tersebut akan diubah nilainya. Beberapa macam mutasi pada algoritma genetik menurut jenis pengkodeannya antara lain:

1. Mutasi dalam Pengkodean Biner
   Mutasi pada pengkodean biner merupakan operasi yang sangat sederhana. Proses yang dilakukan adalah menginversi nilai bit pada posisi tertentu yang terpilih secara acak, sehingga 0 jadi 1 atau 1 jadi 0. Contohnya sebagai berikut:

Parent :	**1**	**0**	**1**	**0**	**1**	**0**
Offspring :	**1**	**1**	**1**	**0**	**1**	**0**

2. Mutasi dalam Pengkodean Permutasi
   Mutasi ini dilakukan dengan cara memilih dua posisi dari kromosom dan kemudian nilainya saling dipertukarkan. Mutasi permutasi diantaranya adalah :
   1. *Swap Mutation* (mutasi pertukaran)
      Mutasi ini dilakukan dengan memilih dua posisi secara acak, kemudian keduanya ditukar.

2. *Insert Mutation* (mutasi penyisipan)
   Prosesnya yaitu memilih dua posisi secara acak misalkan P1 dan P2, kemudian menyisipkan posisi P2 setelah P1.
3. *Sramble Mutation* (mutasi pengacakan)
   Mutasi ini dilakukan dengan memilih satu segmen kromosom dengan memilih dua titik potong sebagai awal dan akhir posisi. Kemudian mengacak urutan gen yang berada dalam segmen tersebut.
4. *Invertion Mutation* (mutasi pembalikan)
   Mutasi pembalikan hampir sama dengan mutasi pengacakan. Awalnya dipilih dua titik potong sebagai awal dan akhir, kemudian yang membedakannya dengan mutasi pengacakan yaitu urutan gen pada segmen yang terpilih (area di antara dua titik potong) dibalik urutannya.

3. Mutasi dalam Pengkodean Nilai
   Mutasi pada pengkodean nilai hampir sama dengan yang dilakukan pada pengkodean biner, tetapi yang dilakukan bukan menginversi nilai bit. Penerapan mutasi bergantung pada jenis nilai yang digunakan. Sebagai contoh untuk nilai riil, proses mutasi dapat dilakukan seperti yang dilakukan pada pengkodean permutasi, dengan saling mempertukarkan nilai dua gen pada kromosom.

4. Mutasi dalam Pengkodean Pohon
   Mutasi dalam pengkodean pohon dapat dilakukan antara lain dengan cara mengubah operator (+, -, *, /) atau nilai yang terkandung pada suatu verteks pohon yang dipilih. Atau, dapat juga dilakukan dengan memilih dua verteks dari pohon dan saling mempertukarkan operator atau nilainya.

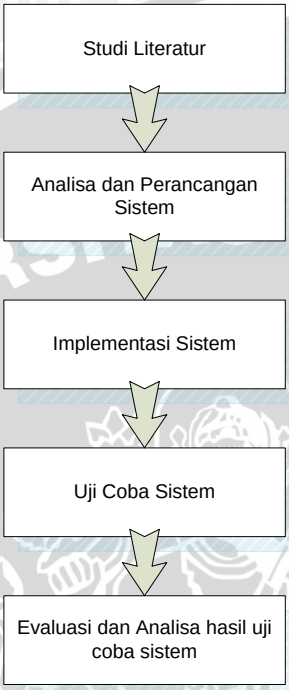

UNIVERSITAS BRAWIJAYA

BAB III
METODE DAN PERANCANGAN

Pada bab ini akan dibahas mengenai perancangan sebuah sistem yang dapat meringkas multi dokumen (banyak dokumen) berbahasa Inggris secara otomatis menggunakan *single pass clustering* berbasis konten dan perangkingan dengan pendekatan algoritma genetika. Sistem ini akan diujikan pada beberapa dokumen berbahasa Inggris. Penelitian ini akan dilakukan dengan tahap-tahap sebagai berikut:

1. Melakukan studi literatur yang berhubungan dengan algortima genetika dan teori-teori lain yang mendukung dalam pemrosesan teks.
2. Menganalisa dan merancang sistem untuk meringkas multi dokumen dengan mengimplementasikan algoritma genetika.
3. Melakukan implementasi sistem berdasarkan analisa dan perancangan yang telah dilakukan sebelumnya.
4. Melakukan uji coba terhadap sistem yang telah dibuat dengan menganalisa hasil. Hasil yang dikeluarkan oleh sistem berupa ringkasan teks yang prosentasenya ditentukan oleh user.
5. Mengevaluasi hasil dari ringkasan teks tersebut dengan menghitung *recall*, *precision*, dan *F-measure*.

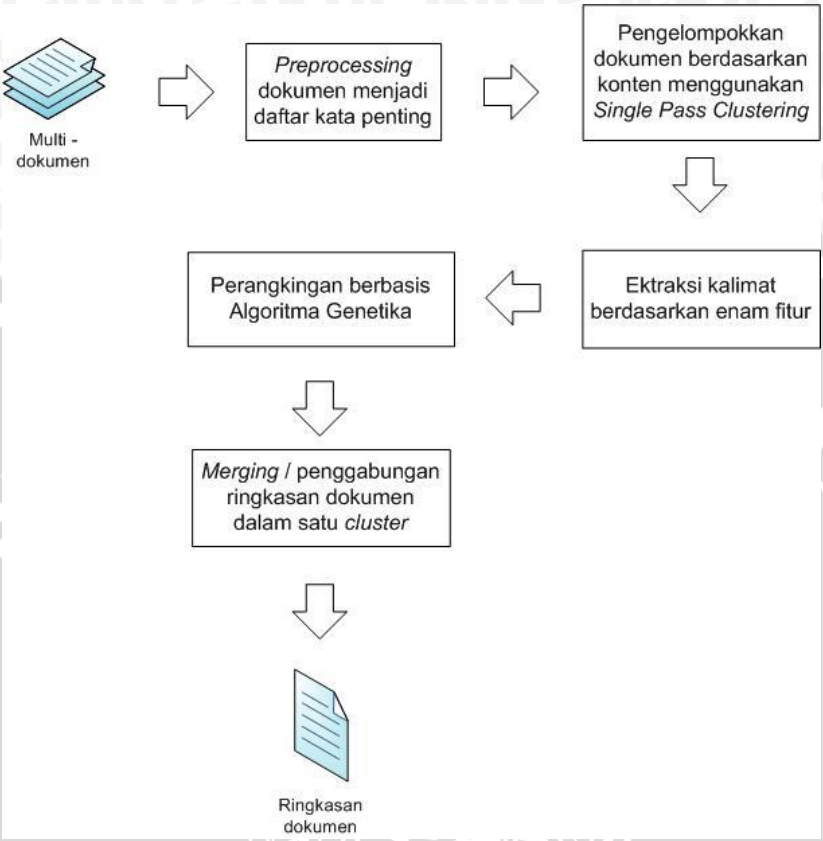
Langkah-langkah penelitian yang dilakukan dapat dilihat pada Gambar 3.1 berikut:



Gambar 3.1 Diagram Alir Pembuatan Perangkat Lunak

3.1 Deskripsi Sistem Keseluruhan

Secara umum, cara kerja sistem yaitu sistem akan menerima input beberapa dokumen berbahasa Inggris dari *user* kemudian dokumen-dokumen tersebut akan diproses melalui beberapa tahapan untuk menghasilkan ringkasan yang merupakan intisari dari dokumen-dokumen masukan. Berikut ini Gambar 3.2 merupakan Gambar aliran data yang terjadi pada sistem secara keseluruhan :



Gambar 3.2 Deskripsi Sistem

Awalnya, sistem akan melakukan *preprocessing* terhadap dokumen-dokumen masukan. Sebelum diringkas dokumen-dokumen masukan tersebut akan dikelompokkan terlebih dahulu. *Clustering* dokumen dilakukan berdasarkan kesamaan konten yang dimiliki karena hanya dokumen yang memiliki banyak kesamaan konten yang dapat diringkas menjadi satu. Selanjutnya, peringkasan untuk masing-masing dokumen dilakukan dalam dua bagian yaitu ekstraksi kalimat dan perangkingan kalimat berbasis algoritma genetika. Hasil dari ringkasan tiap dokumen akan digabung. *Merging* merupakan proses penggabungan ringkasan suatu dokumen dengan ringkasan dokumen lain dalam satu *cluster*, sehingga dihasilkan teks ringkasan untuk multi dokumen. Dokumen-dokumen yang berada dalam satu

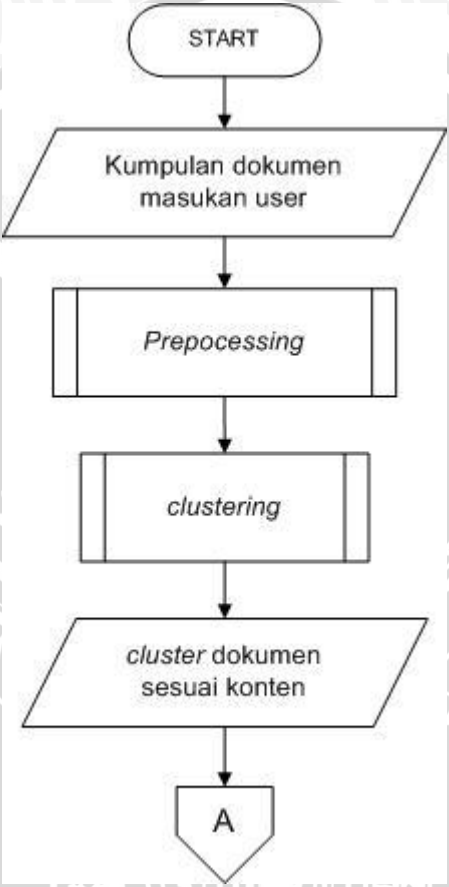
cluster akan menghasilkan sebuah ringkasan, jadi jumlah ringkasan sama dengan jumlah *cluster*. Hasil ringkasan multi dokumen akan dievaluasi menggunakan tiga parameter yaitu *precision*, *recall*, dan *F-measure*.

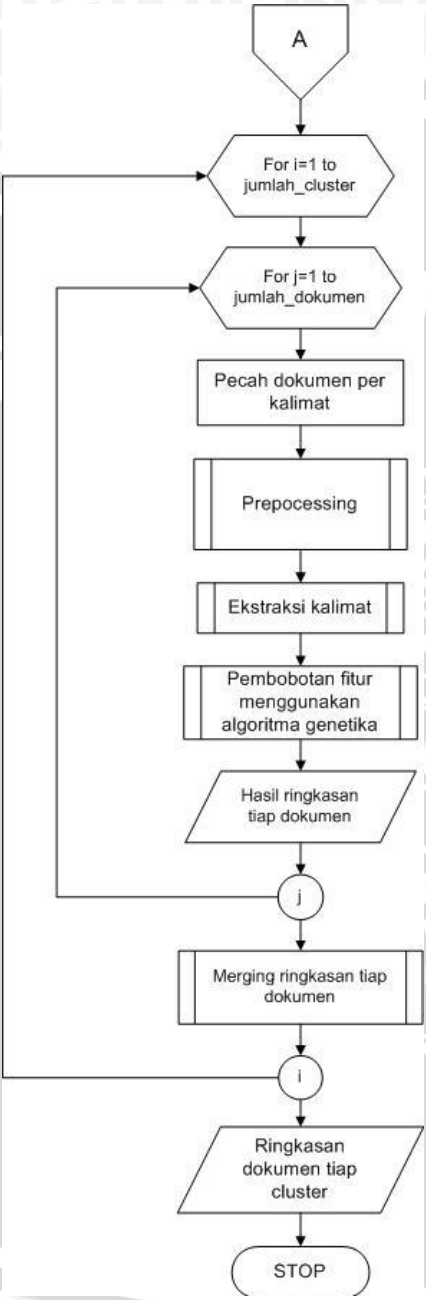
3.2 Rancangan Sistem

Sistem dirancang untuk meringkas multi dokumen berbahasa Inggris berbasis konten menggunakan *single pass clustering* dan perangkingan berbasis algoritma genetika. Pada sistem ini u*ser* akan menginputkan beberapa dokumen berbahasa Inggris berbentuk teks dengan ekstensi .txt. *User* harus memilih presentase ringkasan (*compression ratio*) untuk menetukan berapa banyak ringkasan yang dihasilkan. Setelah itu, sistem akan memulai proses peringkasan dokumen. Tahap-tahap peringkasan multi dokumen berbahasa Inggris sebagai berikut:

1. Setelah dokumen-dokumen dimasukkan, sistem akan melakukan proses *preprocessing*. Tahap-tahap *preprocessing* adalah *tokenizing* dan *case folding*, *filtering* (*stop word removal*), dan *stemming*. Tahap ini akan menghasilkan daftar kata-kata penting.
2. Dokumen-dokumen akan dikelompokkan sesuai kontennya. Perhitungan *single pass clustering* menggunakan *cosine similarity* dan TF.IDF yang dinormalisasi.
3. Proses utama sistem ini adalah meringkas masing-masing dokumen. Prosesnya terdiri dari dua tahap yaitu ekstraksi kalimat dan perangkingan kalimat dengan algoritma genetika.
    a. Ekstraksi kalimat menggunakan enam fitur / kriteria untuk memberi nilai tiap kalimat. Enam kriteria tersebut yaitu fitur panjang kalimat, pembobotan kata (*term weight*), *similarity*, *proper noun*, *thematic word*, dan *numerical data*.
    b. Hasil dari ekstraksi kalimat tiap dokumen akan dilakukan dirangking dengan pendekatan algoritma genetika. Setelah kalimat-kalimat dalam suatu dokumen sudah terurut dari yang paling penting, kalimat diseleksi sesuai ukuran ringkasan yang telah dipilih *user*.
4. Menggabungkan ringkasan dokumen-dokumen tunggal dalam satu *cluster* (*merging*). Proses ini menggunakan perhitungan *cosine similarity* dengan TF.

Proses-proses yang dilakukan oleh sistem secara keseluruhan ditunjukkan oleh Gambar 3.3 berikut ini:





Gambar 3.3 Diagram Alir Proses Sistem

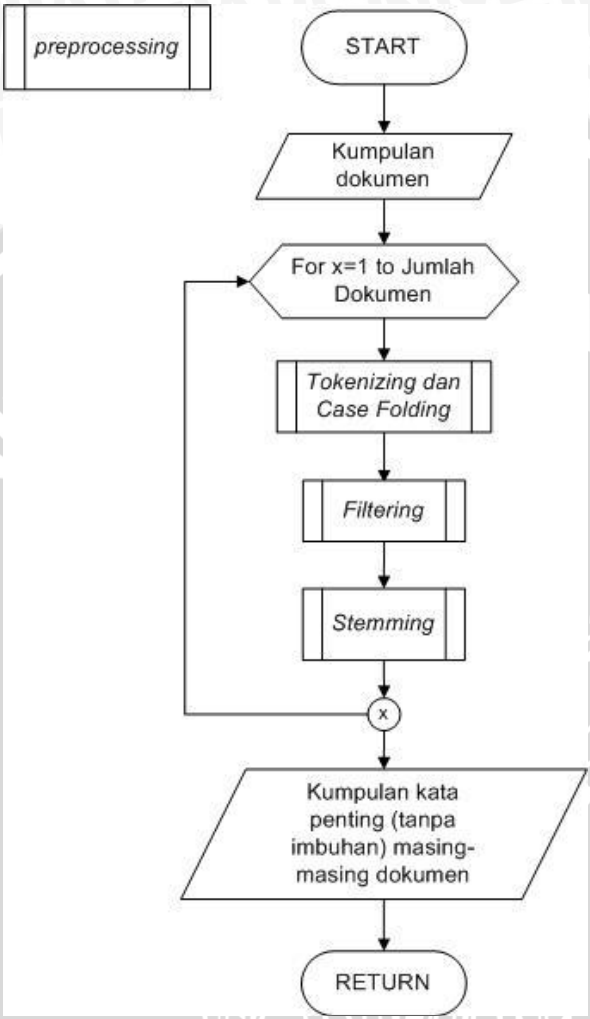
Proses yang pertama kali dilakukan adalah membaca file teks berekstensi .txt yang telah diinputkan *user*. Setelah itu proses selanjutnya sebagai berikut:

3.2.1 Proses *Preprocessing*

Proses ini merupakan sub proses dari sistem secara keseluruhan. Berikut ini prosesnya:

1. Masukan dari proses ini adalah kumpulan dokumen asli.
2. Dokumen-dokumen tersebut akan dilakukan *tokenizing* untuk memecah dokumen menjadi potongan kata penyusunnya. Untuk memecah dokumen menjadi kata-kata penyusunnya digunakan fungsi split dari c#.
3. Setelah itu dilakukan proses menghilangkan semua tanda baca dan mengubah semua huruf menjadi huruf kecil yang disebut *case folding*. Proses case folding menggunakan fungsi dari c# yang disebut Regex (Regular Expression).
4. Pada tahap *filtering* akan dilakukan proses menghilangkan kata-kata yang tidak penting yang ada pada daftar *stop word*.
5. Kemudian dilakukan proses *stemming* untuk membuang imbuhan. *Stemming* yang digunakan adalah *stemming* Porter. Diagram Alir Porter *Stemmer* dapat dilihat pada gambar 2.1.
6. Hasil akhir dari proses ini adalah kumpulan kata-kata penting untuk masing-masing dokumen.

Diagram alir tahap *preprocessing* ditunjukkan pada Gambar 3.4 berikut :



Gambar 3.4 Diagram Alir tahap *preprocessing*

3.2.2 Tahap Clustering Dokumen

Sesuai hasil dari *preprocessing* dilakukan pengelompokkan menggunakan *single pass clustering*. Proses ini merupakan sub proses dari keseluruhan sistem. *Clustering* dilakukan berdasarkan kesamaan konten dokumen. Perhitungan *single pass clustering* menggunakan TF.IDF yang telah dinormalisasi dan *cosine similarity*.

Persamaan TF.IDF yang telah dinormalisasi ditunjukkan pada persamaan 3.1 berikut :

$$C_{uk} = \frac{tf_{uk} \times log \frac{N}{df_k}}{\sqrt{\sum_{k=1}^{n} S_{uk}^2}} \quad (3.1)$$

C_{uk} : Nilai bobot kata ke – k pada dokumen ke – u
tf_{uk} : Jumlah kata ke – k pada dokumen ke – u
N : Banyak dokumen
df_k : Jumlah dokumen yang mengandung kata k
s_{uk} : Nilai dari $tf_{uk} \times log \frac{N}{df_k}$

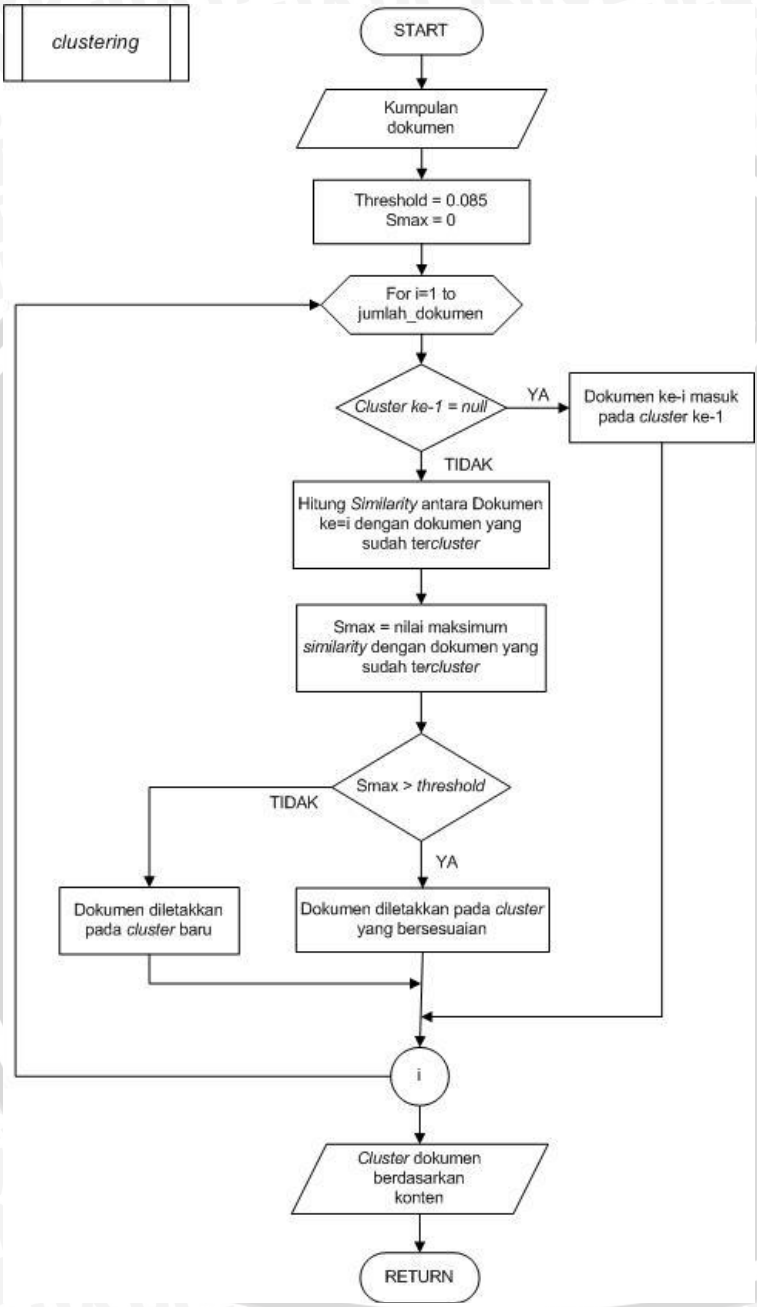
Proses menghitung *similarity* antar dokumen menggunakan *cosine similarity*. Karena perhitungan *cosine similarity* menggunakan TF.IDF yang telah dinormalisasi maka persamaan *cosine similarity* yang digunakan ditunjukkan persamaan 3.2 berikut :

$$\text{Sim}(U_i, U_j) = \sum_{kata\ yang\ sama} c_{ik} \times c_{jk} \quad (3.2)$$

Tahapan proses *clustering* sebagai berikut :

1. Masukan dari proses ini adalah kumpulan dokumen hasil *preprocessing*.
2. Inisialisasi nilai *threshold* = 0.085 dan Smax = 0.
3. Untuk setiap dokumen dilakukan perhitungan *similarity* dengan dokumen yang telah tercluster.
4. Sebelum dilakukan perhitungan *similarity*, dilakukan pengecekan pada *cluster* ke-1. Jika *cluster* ke-1 bernilai *null* (kosong) maka dokumen ke-i langsung dimasukkan pada *cluster* 1 .
5. Nilai Smax dihitung dari maksimum nilai *similarity* dokumen ke-i dengan dokumen lain yang sudah ter*cluster*.
6. Jika nilai Smax lebih besar daripada nilai *threshold* maka dokumen ke-i dimasukkan pada *cluster* yang bersesuain.
7. Jika nilai Smax kurang dari threshold maka dokumen dimasukkan pada *cluster* baru.
8. Hasil akhir proses ini adalah pengelompokkan dokumen berdasarkan konten.

Gambar 3.5 berikut adalah diagram alir proses *clustering* yang dilakukan oleh sistem:



Gambar 3.5 Diagram Alir Proses *Clustering*

3.2.3 Proses Peringkasan Dokumen

Ada dua tahap dalam proses peringkasan dokumen, yaitu ekstraksi kalimat dan perangkingan kalimat dengan algoritma genetika. Setelah kalimat diekstrak menggunakan enam fitur, kemudian perangkingan kalimat berbasis algoritma genetika dilakukan dengan pembobotan pada nilai fitur tersebut. Kedua tahap tersebut akan dijelaskan secara rinci berikut ini:

3.2.3.1 Ekstraksi Kalimat

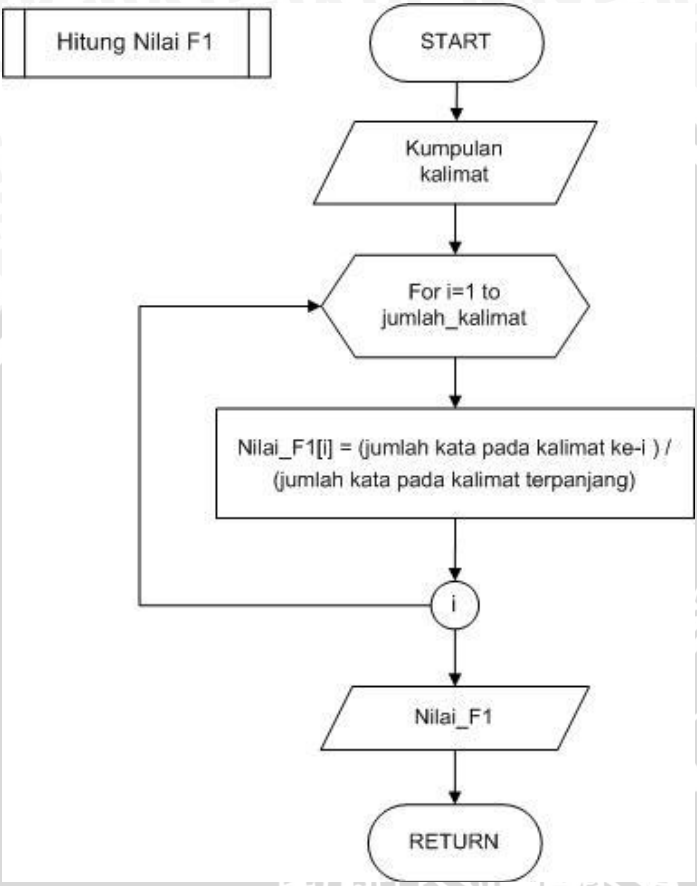
Sebelum melakukan ekstraksi kalimat, dokumen dipecah menjadi tiap kalimat. Setiap kalimat dari dokumen tersebut akan diekstraksi sehingga memiliki nilai yang mewakili kalimat tersebut. Ada enam fitur untuk setiap kalimat. Setiap fitur diberi nilai antara '0 'dan '1'. Enam fitur tersebut sebagai berikut:

1. Fitur Panjang Kalimat (F1)

   Fitur ini berfungsi untuk menyaring kalimat pendek seperti nama pengarang dan datelines seperti pada artikel berita. Kalimat pendek tidak dipakai untuk ringkasan dokumen. Perhitungan fitur ini ditunjukkan dengan persamaan 3.3 berikut:

$$F1 = \frac{jumla\ h\ kata\ \ pada\ \ suatu\ \ kalimat}{jumla\ h\ kata\ \ pada\ \ kalimat\ \ terpanjang} \quad (3.3)$$

   Diagram alir proses perhitungan fitur 1 ditunjukkan pada gambar 3.6 berikut:



Gambar 3.6 Diagram Alir Hitung Fitur 1

2. Fitur Pembobotan Kata/*Term Weight* (F2)

Frekuensi bobot kata yang ada pada dokumen digunakan untuk menghitung pentingnya kalimat. Nilai suatu kalimat dapat dihitung sebagai jumlah nilai bobot kata dalam kalimat tersebut. Di sini akan dilakukan penghitungan menggunakan persamaan TF.IFS seperti persamaan 3.4 berikut :

$$O_k = tf_k \times isf_k = t_{fk} \times \log\frac{N}{n_k} \quad (3.4)$$

O_k adalah bobot kata ke-k
tf_k adalah frekuensi dari bobot kata k pada dokumen
N adalah jumlah seluruh kalimat

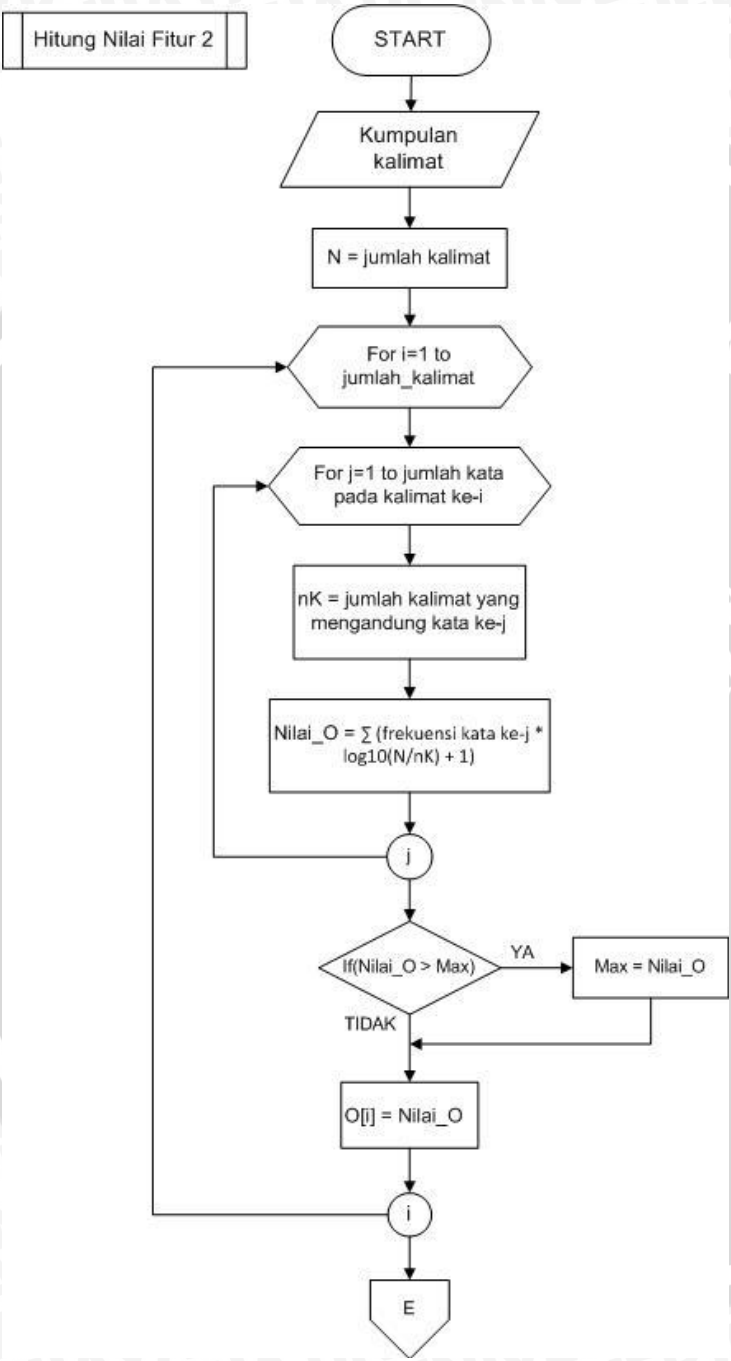
n_k adalah jumlah kalimat yang mengandung kata k

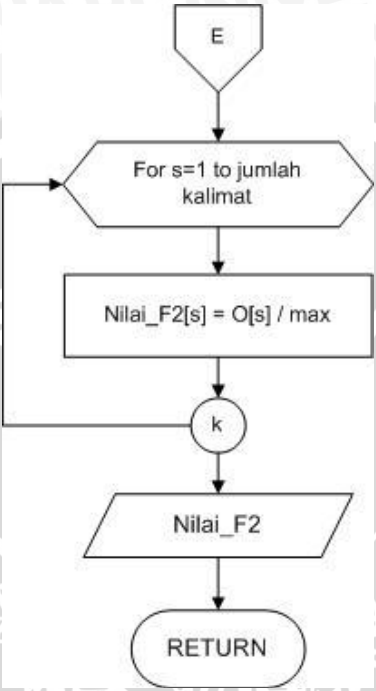
Sehingga persamaan untuk fitur ini ditunjukkan dengan persamaan 3.5 sebagai berikut :

$$F2 = \frac{\sum_{k=1}^{n} O_k\ (S)}{Max\ (\sum_{k=1}^{n} O_k\ (S_k^N))} \quad (3.5)$$

n adalah jumlah kata pada kalimat.

Diagram alir proses perhitungan fitur 2 ditunjukkan pada gambar 3.7 berikut:


Hitung Nilai Fitur 2
START
Kumpulan kalimat
N = jumlah kalimat
For i=1 to jumlah_kalimat
For j=1 to jumlah kata pada kalimat ke-i
nK = jumlah kalimat yang mengandung kata ke-j
Nilai_O = ∑ (frekuensi kata ke-j * log10(N/nK) + 1)
j
If(Nilai_O > Max)
YA
Max = Nilai_O
TIDAK
O[i] = Nilai_O
i
E



Gambar 3.7 Diagram Alir Pehitungan Fitur 2

3. Fitur *Similarity* Kalimat (F3)

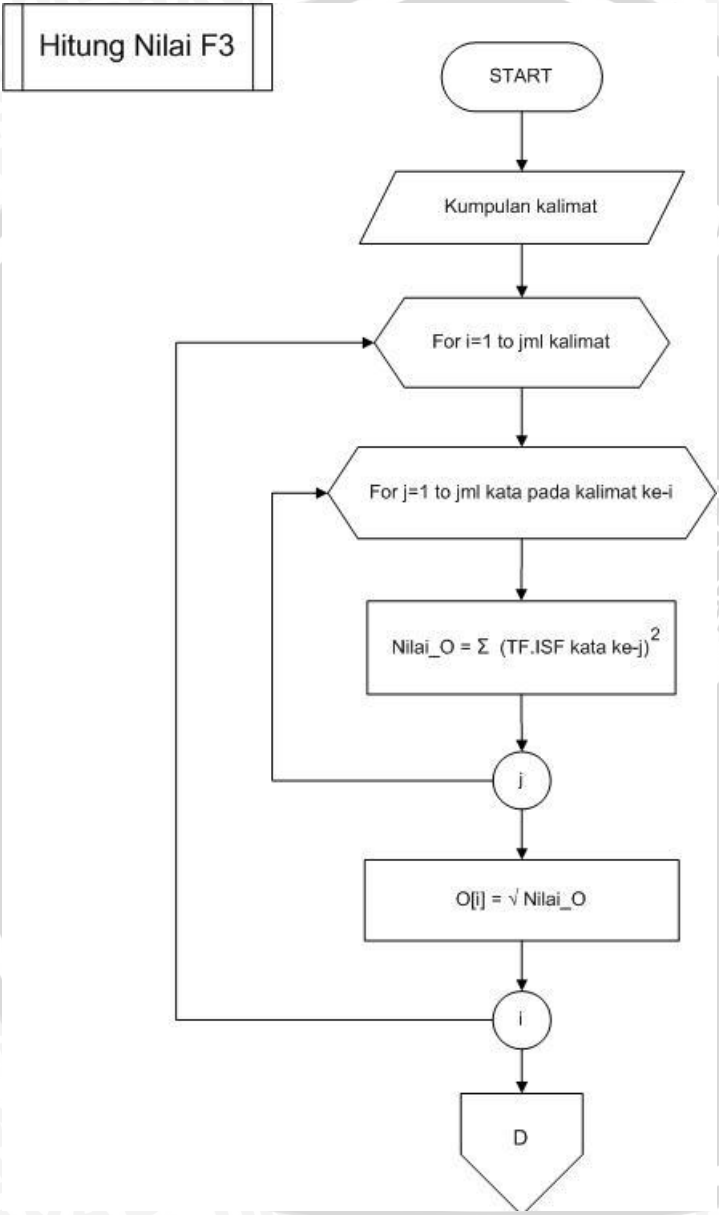
Fitur ini merupakan kesamaan antar kalimat. Untuk tiap kalimat S, similarity antara S dan tiap kalimat lain dihitung menggunakan *cosine similarity* dengan interval nilai antara 0 sampai 1. Bobot kata O_i dan O_j dari kata t sampai n pada kalimat S_i dan S_j direpresentasikan oleh vektor. *Similarity* tiap kalimat ditunjukkan oleh persamaan 3.6 berikut:

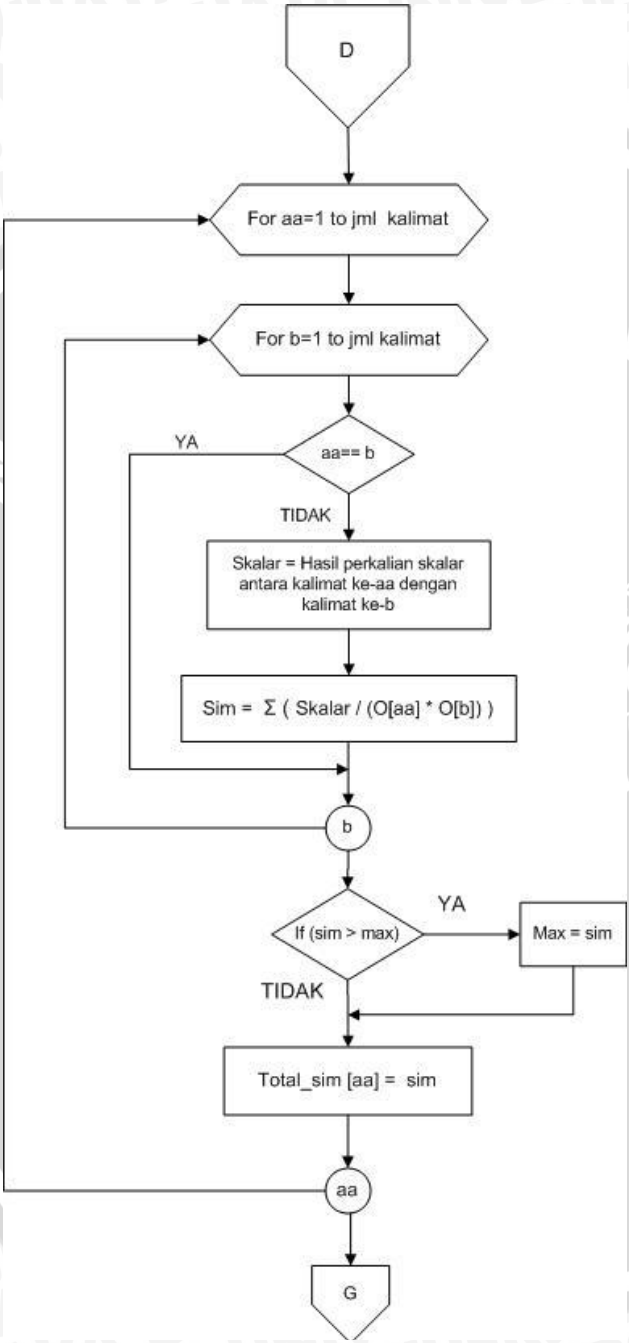
$$Sim\left(S_i, S_j\right) = \frac{\sum_{k=1}^{n} O_{ik} \times O_{jk}}{\sqrt{\sum_{k=1}^{n} O_{ik}^2} \times \sqrt{\sum_{k=1}^{n} O_{jk}^2}} \quad (3.6)$$

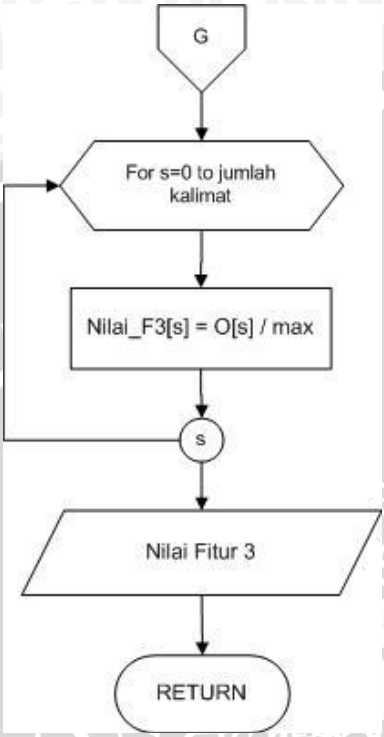
Nilai dari fitur ini untuk kalimat S yaitu dengan persamaan 3.7 berikut :

$$F3 = \frac{\sum Sim\ (S_i, S_j)}{Max\ (\sum\ Sim\,(S_i, S_j))} \quad (3.7)$$

Diagram alir proses perhitungan fitur 3 ditunjukkan pada gambar 3.8 berikut:




D
For aa=1 to jml kalimat
For b=1 to jml kalimat
aa== b
YA
TIDAK
Skalar = Hasil perkalian skalar antara kalimat ke-aa dengan kalimat ke-b
Sim = Σ (Skalar / (O[aa] * O[b]))
b
If (sim > max)
YA
Max = sim
TIDAK
Total_sim [aa] = sim
aa
G



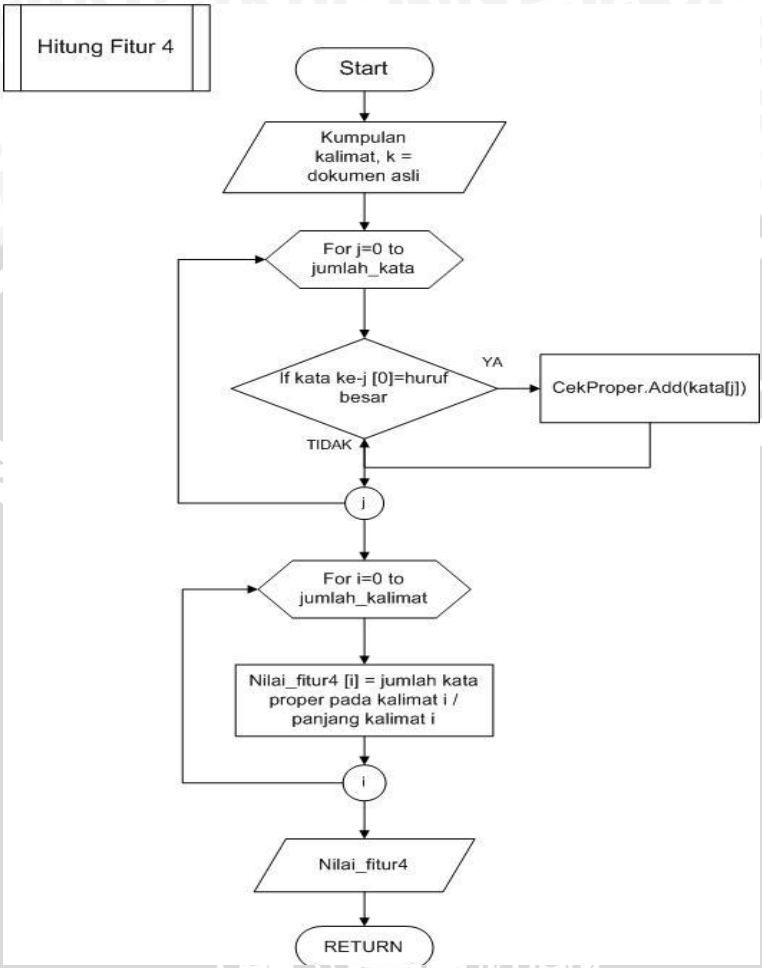
Gambar 3.8 Diagram Alir Perhitungan Fitur 3

4. Fitur *Proper noun* (F4)

Kalimat yang mengandung *proper noun* termasuk kalimat penting yang biasanya masuk dalam ringkasan. *Proper noun* adalah kata yang menunjukkan nama sesuatu, seperti nama orang, nama tempat, nama bulan, dan sebagainya. Berikut ini persamaan 3.8 untuk menghitung nilai *proper noun* :

$$F4 = \frac{jumla\ h\ proper\ \ noun\ \ pada\ \ kalimat\ \ S}{panjang\ \ \ kalimat\ /jumla\ h\ kata\ \ S} \quad (3.8)$$

Diagram alir proses perhitungan fitur 4 ditunjukkan pada gambar 3.9 berikut:



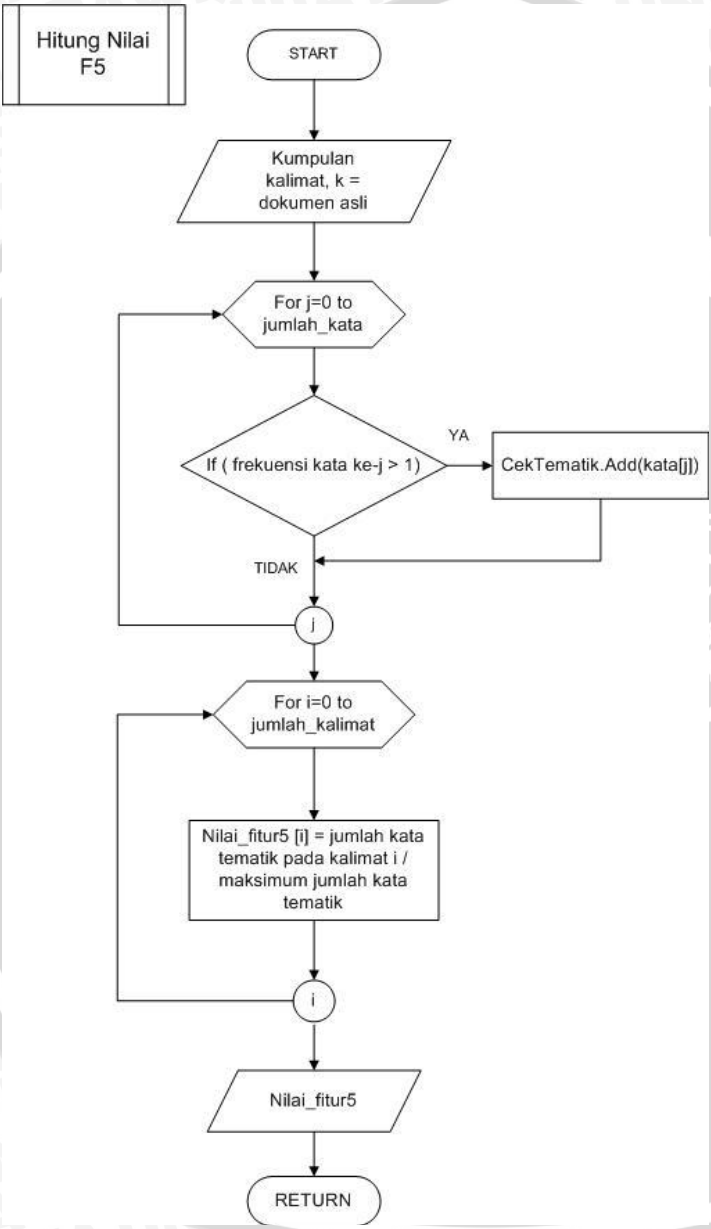
Gambar 3.9 Diagram Alir Perhitungan Fitur 4

5. Fitur *Thematic word* (F5)

Thematic word yang dimaksud adalah kata yang frekuensinya tinggi pada suatu dokumen. Fitur ini penting karena berhubungan dengan topik. Pada penelitian ini diasumsikan *Thematic word* atau kata tematik adalah kata yang frekuensinya lebih dari satu. Persamaannya ditunjukkan oleh persamaan 3.9 berikut :

$$F5 = \frac{jumla\ h\ thematic\ \ word\ \ pada\ \ kalimat\ \ S}{Max\ jumla\ h\ thematic\ \ word\ \ pada\ \ suatu\ \ kalimat} \qquad (3.9)$$

Diagram alir proses perhitungan fitur 5 ditunjukkan pada gambar 3-9 berikut:



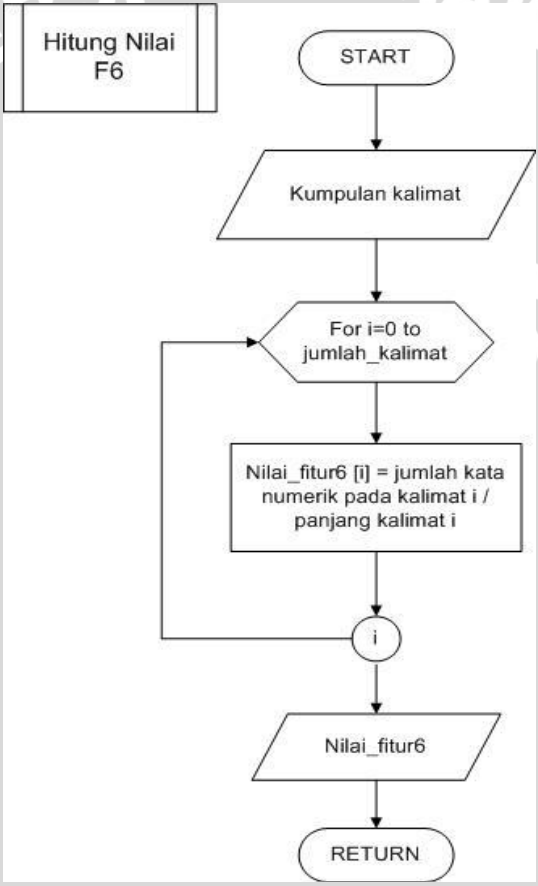
Gambar 3.10 Diagram Alir Perhitungan Fitur 5

6. Fitur *Numerical data* (F6)

Kalimat yang mengandung data numerik dianggap kalimat penting dan biasanya masuk dalam ringkasan. Nilai dari fitur ini dihitung dengan persamaan 3.10 berikut :

$$F6 = \frac{jumla\ h\ data\ \ nume\ rik\ \ pada\ \ kalimat\ \ S}{panjang\ \ \ kalimat\ \ S} \quad (3.10)$$

Diagram alir proses perhitungan fitur 6 ditunjukkan pada gambar 3.11 berikut:



Gambar 3.11 Diagram Alir Perhitungan Fitur 6

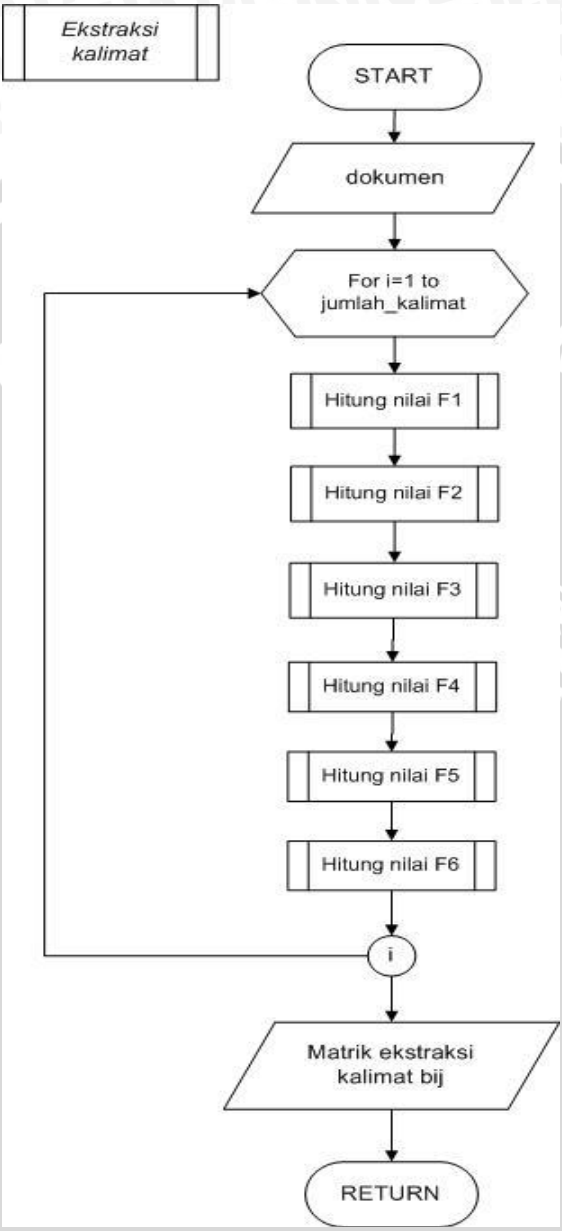
Tahap ekstraksi kalimat sebagai berikut:

1. Masukan dari tahap ini adalah kumpulan kalimat tiap dokumen.

2. Untuk setiap kalimat dalam dokumen akan dihitung menggunakan persamaan-persamaan fitur.
3. Tahap ini akan menghasilkan matriks b_{ij}, dimana i merupakan jumlah kalimat dan j merupakan jumlah fitur. Matriknya sebagai berikut :

$$\begin{array}{c} \\ Kalimat_1 \\ Kalimat_2 \\ \vdots \\ Kalimat_i \end{array} \begin{array}{c} \begin{array}{cccc} F_1 & F_2 & \dots & F_j \end{array} \\ \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1j} \\ b_{21} & b_{22} & \dots & b_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ b_{i1} & b_{i2} & \cdots & b_{ij} \end{bmatrix} \end{array}$$

Diagram alir proses ekstraksi kalimat digambarkan pada Gambar 3.12 dibawah ini:



Gambar 3.12 Diagram Alir Proses Ekstraksi Kalimat

3.2.3.2 Perangkingan Kalimat dengan Algoritma Genetika

Proses perangkingan kalimat dengan algoritma genetika dapat dilakukan seperti proses perangkingan pada Multiple Attribute Decision Making (MADM) pada jurnal Sri Kusumadewi, 2005. Pada jurnal tersebut dilakukan perangkingan alternatif yang memiliki banyak kriteria. MADM adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria/atribut tertentu (Kusumadewi, 2005).

Alternatif adalah objek-objek yang berbeda dan memiliki kesempatan yang sama untuk dipilih oleh pengambil keputusan sedangkan atribut merupakan karakteristik dari alternatif atau kriteria keputusan (Yanko, 2005). Inti dari MADM adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan.

Pada penelitian ini, alternatif di analogikan dengan kalimat dan atribut dianalogikan dengan fitur. Setiap kalimat memiliki banyak nilai fitur (banyak atribut). Untuk memproses kalimat tersebut agar dapat dirangking maka harus dilakukan pembobotan pada fitur kalimat tersebut sehingga dapat merangking semua alternatif yang mungkin sesuai tujuan dan kriteria.

Kriteria atau atribut memiliki nilai-nilai yang bisa bertentangan, sehingga diperlukan suatu algoritma untuk merangking alternatif dengan situasi semua atribut terpenuhi. Algoritma evolusioner seperti algoritma genetika adalah teknik yang cocok untuk optimasi multiobjektif sehingga sangat membantu untuk mengoptimalkan solusi dari permasalahan MADM. Algoritma genetika dapat mengeksploitasi banyak solusi dengan ruang pencarian yang luas untuk menemukan yang paling optimal. Dengan demikian algoritma ini dapat merangking alternatif secara optimal, dengan situasi semua atribut dapat dipenuhi. Untuk melakukannya harus dilakukan pembobotan pada atribut yang menunjukkan kepentingan relatifnya. Pembobotan bisa dilakukan dengan banyak cara salah satunya dengan persamaan objektif Fan (Kusumadewi, 2005).

Persamaan objektif Fan akan digunakan dalam penghitungan nilai fitness. Karena persamaan objektif Fan merupakan persamaan minimum maka nilai fitness dihitung dari invers persamaan objektif Fan. Nilai fitness pada proses algoritma genetika digunakan sebagai tolak ukur menemukan solusi yang terbaik. Suatu solusi dikatakan

optimal jika tidak ada solusi lain yang ditemukan yang dapat meningkatkan tujuan tertentu tanpa merugikan tujuan lainnya.

Contoh tabel keputusan yang menggambarkan hubungan kalimat sebagai alternatif dan fitur sebagai atribut ditunjukkan dengan tabel 3.1 berikut:

Tabel 3.1 Contoh Tabel keputusan

No.	Alternatif	Atribut (fitur)			
		F1	F2	…	F6
1	Kalimat 1				
2	Kalimat 2				
.	.				
.	.				
.	.				
m	Kalimat m				

Jumlah Kalimat

Proses perangkingan dengan algoritma genetika sebagai berikut :

Parameter yang Digunakan

1. L_i adalah panjang gen setiap kromosom dalam individu.
2. B = { b_{ij} | i=1,2,.....,m; j=1,2,.....,n)} adalah matrik keputusan dengan b_{ij} adalah nilai numeris dari alternafit (kalimat) ke-i pada atribut (fitur) ke-j.
3. w_j (j = 1,2,…,n) adalah bobot yang menunjukkan kepentingan relatif dari tiap atribut.
4. x_j (j = 1,2,…,n) adalah variabel temporer yang digunakan untuk mencari nilai bobot (w_j).
5. v merupakan representasi kromosom untuk setiap individu yang terbagi atas m atribut atau konversi bilangan biner tiap kromosom ke bilangan desimal.
6. Fungsi fitness, menunjukkan kualitas setiap individu. Semakin besar nilai fitness semakin tinggi kualitas individu tersebut.
7. g_i adalah nilai alternatif ke-i yang digunakan untuk merangking kalimat.
8. P_c adalah probabilitas *crossover*.
9. P_m adalah probabilitas mutasi.

Pembangkitan Populasi Awal dan Representasi Kromosom

Teknik pengkodean yang digunakan adalah pengkodean biner. Gen dibangkitkan secara random. Pada penelitian ini kromosom merepresentasikan fitur-fitur yang ada. Satu individu terdiri dari enam kromosom sesuai dengan jumlah fitur. Tiap kromosom terdiri dari sejumlah gen sesuai dengan persamaan 3.11 berikut :

$$L_i = \left\lceil {}^{2}\log[(b-a)10^2 + 1] \right\rceil \qquad (3.11)$$

Dengan b = batas atas interval dan a = batas bawah interval.

Untuk data yang perlu dinormalisasi maka sebelumnya harus dinormalisasi dahulu. Sedangkan pada penelitian ini data (nilai-nilai fitur) tidak perlu dinormalisasi karena nilainya sudah pada interval [0 1]. Maka jumlah gen untuk tiap kromosom sebagai berikut :

$$L_i = \left\lceil {}^{2}\log[(1-0)10^2 + 1] \right\rceil$$
$$= \left\lceil {}^{2}\log[101] \right\rceil = 7$$

Dengan demikian ukuran gen untuk tiap individu dengan enam kromosom (enam fitur) = 7 × 6 = 42. Tabel representasi kromosom ditunjukkan oleh tabel 3.2 berikut ini:

Tabel 3.2 Representasi Kromosom

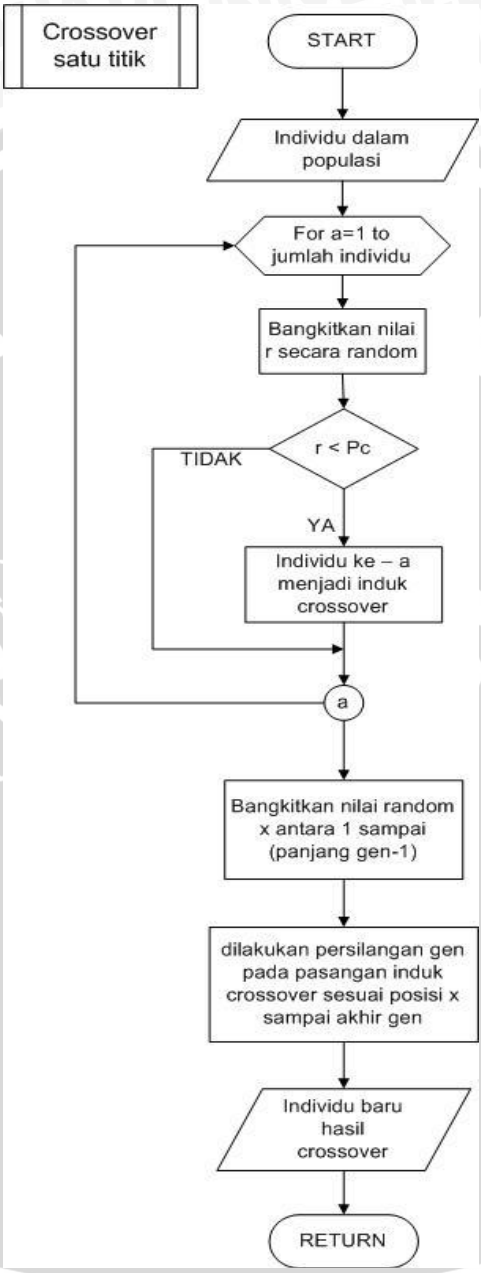
Individu	Representasi Kromosom					
	Fitur 1	Fitur 2	Fitur 3	Fitur 4	Fitur 5	Fitur 6
1	1001011	0011011	1001101	1101110	0100110	0101001
2	1011010	0100101	0010110	0010101	0010010	0011010
. .						
10	110010	101001	0011010	0101001	1100101	0010101

***Crossover* Satu Titik**

Operasi *crossover* yang digunakan pada penelitian ini adalah crossover satu titik. Proses *crossover* tergantung pada probabilitas crossover (P_c). Misalnya probabilitas *crossover* = 0.6 artinya diharapkan 60% individu akan mengalami *crossover*. Tahap-tahap pada proses *crossover* satu titik sebagai berikut:

1. Masukan dari proses ini adalah semua individu dalam populasi dan P_c yang telah ditentukan.
2. Untuk setiap individu, dibangkitkan bilangan random antara 0 sampai 1. Jika nilai random kurang dari probabilitas *crossover* maka individu tersebut dikenai *crossover*. Misalnya untuk 10 individu nilai random yang didapat adalah 0.29, 0.35, 0.89, 0.7, 0.43, 0.96. 0.77, 0.73, 0.89, 0.65, sehingga individu yang menjadi parent *crossover* ada 3 yaitu individu 1,2 dan 5. Kemudian pasangan yang di-*crossover* yaitu 1 dan 2, 1 dan 5, 2 dan 5.
3. Membangkitkan bilangan random antara 1 sampai (n-1) dengan n adalah jumlah gen tiap individu untuk menentukan posisi gen yang di-*crossover*. Misalnya bilangan random yang didapat adalah 40 (dalam kasus ini jumlah gen tiap individu 42) maka untuk *crossover* satu titik, gen pada posisi 40 sampai 42 mengalami *crossover* dengan pasangannya pada posisi yang sama.
4. Hasil proses ini adalah individu baru hasil *crossover*.

Diagram alir proses *crossover* satu titik ditunjukkan pada Gambar 3.13 berikut:



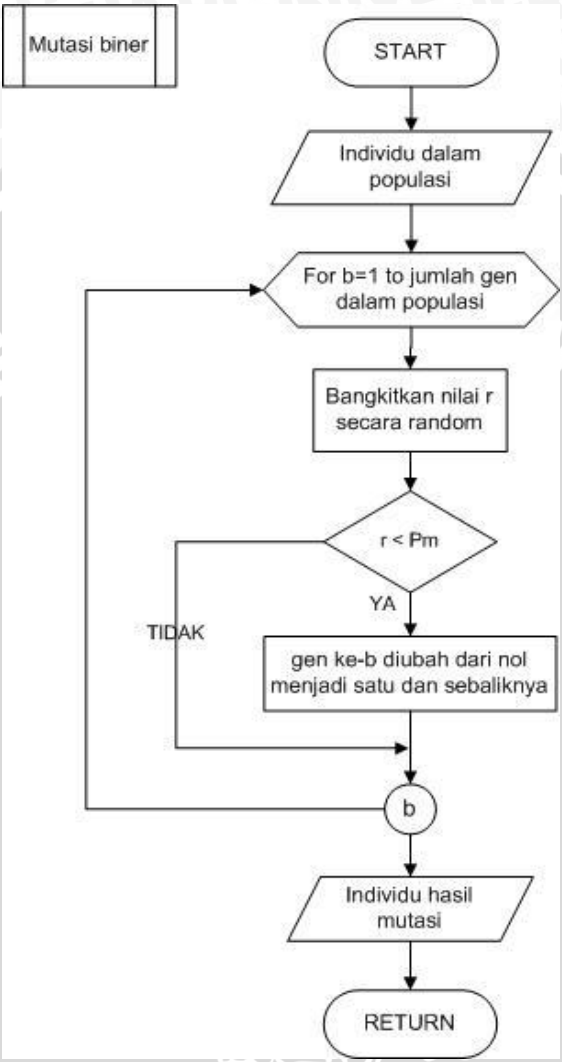
Gambar 3.13 Diagram Alir Proses Crossover Satu Titik

Mutasi Biner

Sama dengan pengkodeannya, mutasi yang digunakan adalah mutasi biner. Mutasi biner yaitu mengubah nilai 1 menjadi 0 dan sebaliknya, 0 jadi 1. Proses mutasi tergantung pada probabilitas mutasi (P_m). Misalkan probabilitas mutasi 0.01, artinya diharapkan 1% dari total gen mengalami mutasi. Proses yang dilakukan pada mutasi biner yaitu:

1. Masukan untuk proses ini adalah individu-individu dalam populasi dan P_m yang telah ditentukan.
2. Untuk setiap individu dibangkitkan bilangan random antara 0 sampai 1 sebanyak jumlah gen tiap individu. Misalkan untuk kasus ini jumlah individu = 10 dengan panjang gen tiap individu = 42. Dengan demikin tiap individu akan dibangkitkan 42 bilangan random dan total bilangan random yang dibangkitkan untuk populasi ini adalah 420.
3. Membandingkan bilangan random tiap gen dengan P_m. Untuk bilangan random yang kurang dari 0.01 maka pada posisi yang sama gen tersebut akan mengalami mutasi. Individu yang dimutasi adalah individu awal yang belum mengalami crossover
4. Hasil proses ini adalah individu baru yang telah termutasi.

Diagram Alir proses mutasi biner ditunjukkan pada Gambar 3.14 berikut:



Gambar 3.14 Diagram Alir Proses Mutasi Biner

E. Perhitungan Nilai Fitness dan Seleksi

Seleksi yang digunakan pada penelitian ini adalah seleksi *Rank-Based fitness*. Seleksi dilakukan dengan mengurutkan individu sesuai nilai fitnessnya kemudian mengambil individu-individu teratas. Seluruh individu, yaitu individu awal, anakan hasil crossover, dan anakan hasil mutasi dihitung nilai fitnessnya kemudian diurutkan

mulai dari yang terbesar sesuai nilai fitness dan diambil yang tertinggi sebanyak ukuran populasi / jumlah individu awal. Misalnya ukuran populasi 10, maka diambil 10 individu dengan fitness terbaik.

Tahap-tahap penghitungan nilai fitness yaitu sebagai berikut:

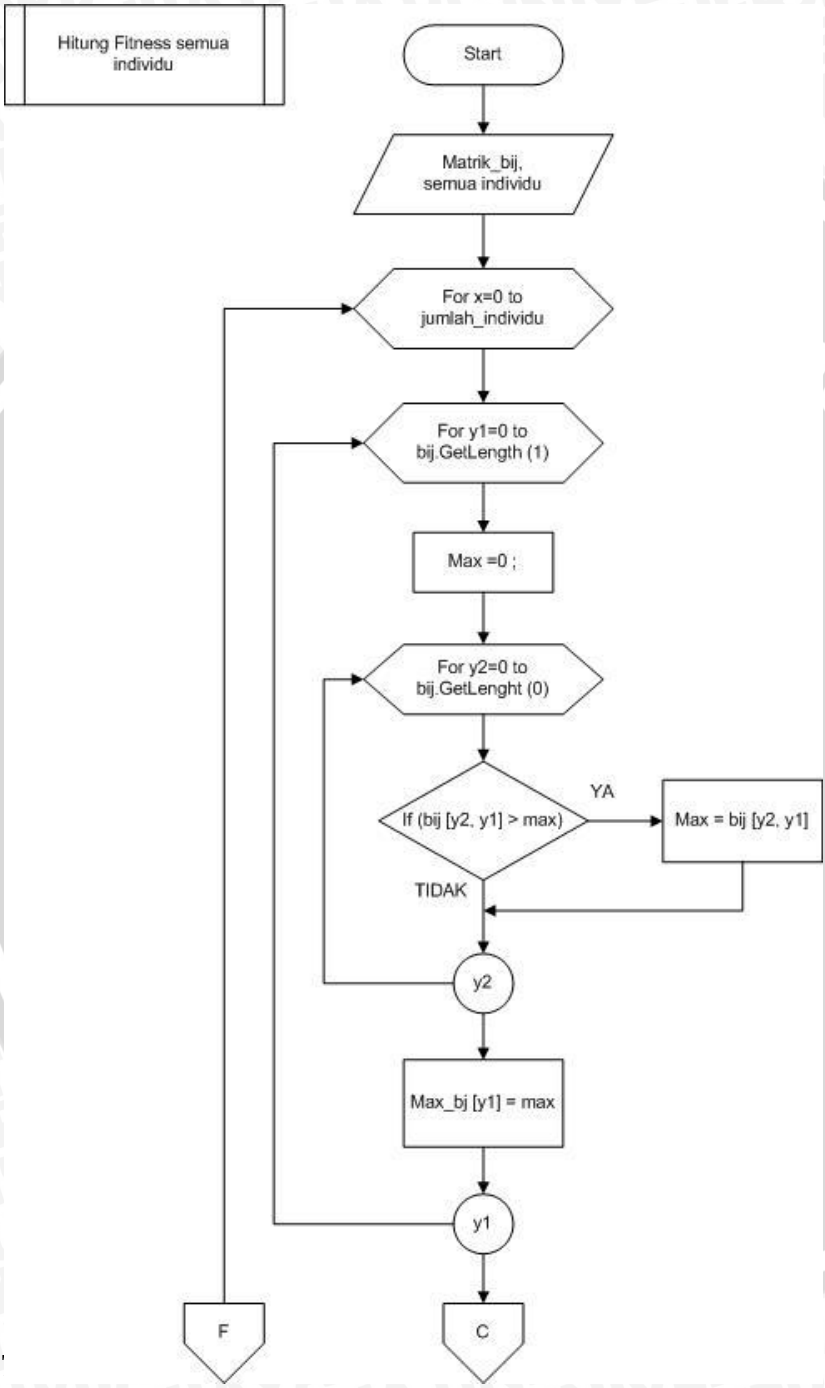
1. Menghitung nilai w (bobot).
   - Mengkonversi bilangan biner kromosom menjadi bilangan desimal sebagai v. v merupakan representasi dari variabel x yang berbentuk string biner. Pada kasus ini tiap individu ada 6 kromosom yang merupakan representasi 6 fitur, maka didapat ada v_1, v_2, …, v_6.
     Contohnya : $v_1 = 0010101 = 21$
   - Menghitung nilai x yang merupakan variabel temporer untuk menghitung w (bobot) dengan persamaan 3.12 berikut :
     $$x_j = a + \left[(b-a)/\left(2^{Lj} - 1\right)\right] \times v_j \qquad (3.12)$$
     Dengan a = batas bawah interval data. Pada kasus ini adalah 0
     b = batas atas interval data. Pada kasus ini adalah 1
     L_j = panjang gen ke-j
     Kemudian dihitung x total untuk tiap individu (pada kasus ini sepanjang 6 kromosom).
   - Sehingga didapatkan nilai w dengan persamaan 3.13 berikut:
     $$w_j = \frac{x_j}{x\ total} \qquad (3.13)$$
     Dimana j = 1,2,…,n (n adalah jumlah kromosom tiap individu)

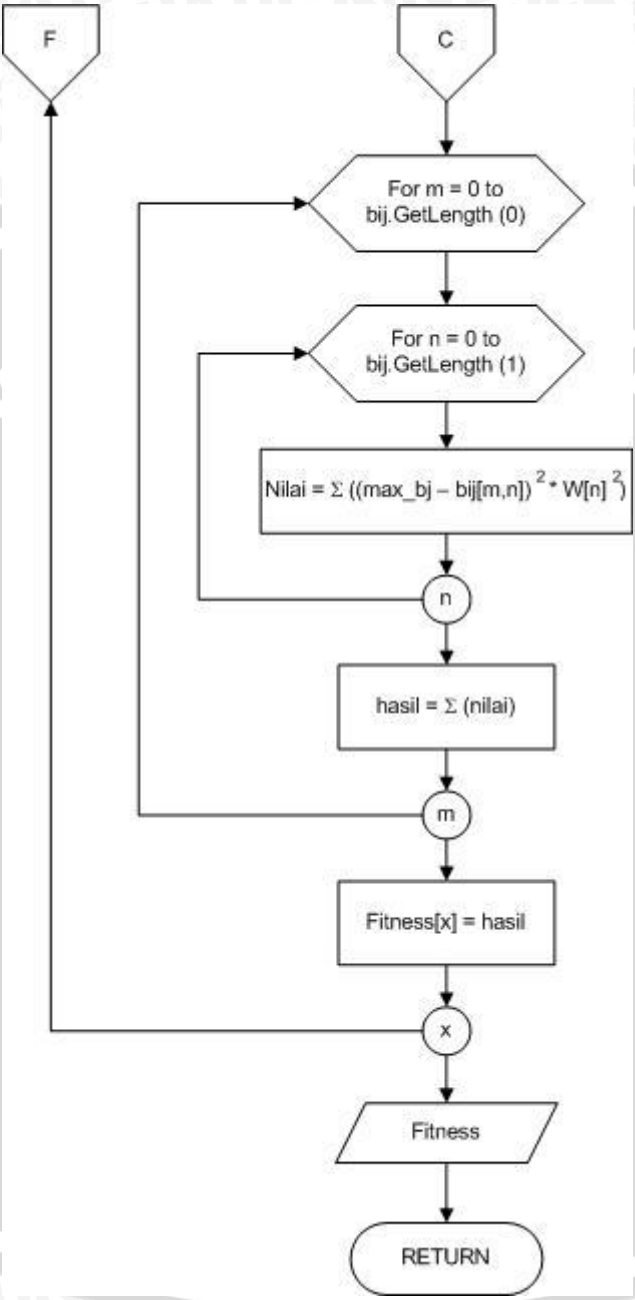
2. Menghitung nilai fitness
   Fungsi fitness didapatkan dari persamaan objektif Fan. Persamaan objektif Fan merupakan persamaan minimum sehingga fungsi fitness dihitung dari invers persamaan objektif Fan. Persamaan fungsi fitness ditunjukkan pada persamaan 3.14.
   $$\text{Fitness} = \frac{1}{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(max\ b_j^m - b_{ij}\right)^2 w_j^2} \qquad (3.14)$$

   Nilai w_j^2 dihitung dari nilai w_j yang telah didapat pada tahap sebelumnya. Nilai max b_j^m merupakan nilai maksimum fitur ke-j pada semua kalimat.

Tahap-tahap yang telah dipaparkan, yaitu crossover, mutasi, perhitungan nilai fitness dan seleksi dilakukan berulang-ulang sampai batas iterasi yang ditentukan.

Diagram alir proses perhitungan nilai fitness ditunjukkan pada gambar 3.15 berikut:


Hitung Fitness semua individu
Start
Matrik_bij, semua individu
For x=0 to jumlah_individu
For y1=0 to bij.GetLength (1)
Max =0 ;
For y2=0 to bij.GetLenght (0)
If (bij [y2, y1] > max)
YA
Max = bij [y2, y1]
TIDAK
y2
Max_bj [y1] = max
y1
F
C



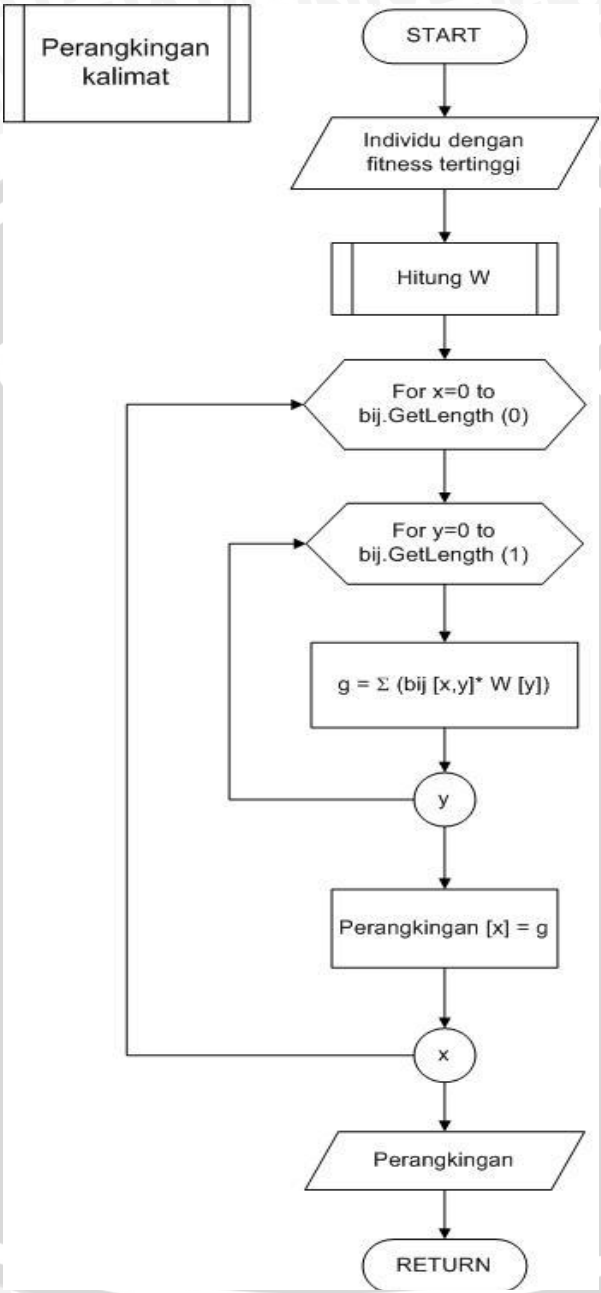
Gambar 3.15 Diagram Alir Perhitungan Fitness

Merangking Kalimat

Proses perangkingan bertujuan untuk memilih alternatif terbaik yang akan terpilih sebagai solusi, dalam kasus ini yaitu kalimat-kalimat terbaik tiap dokumen. Setelah diketahui individu mana yang memiliki nilai fitness tertinggi, individu tersebut dijadikan sebagai solusi terbaik. Untuk mendapatkan urutan rangking, nilai bobot dari individu yang tepilih sebagai solusi terbaik, digunakan untuk menghitung nilai alternatif ke-i sebagai nilai kalimat ke-i dengan persamaan 3.15 berikut :

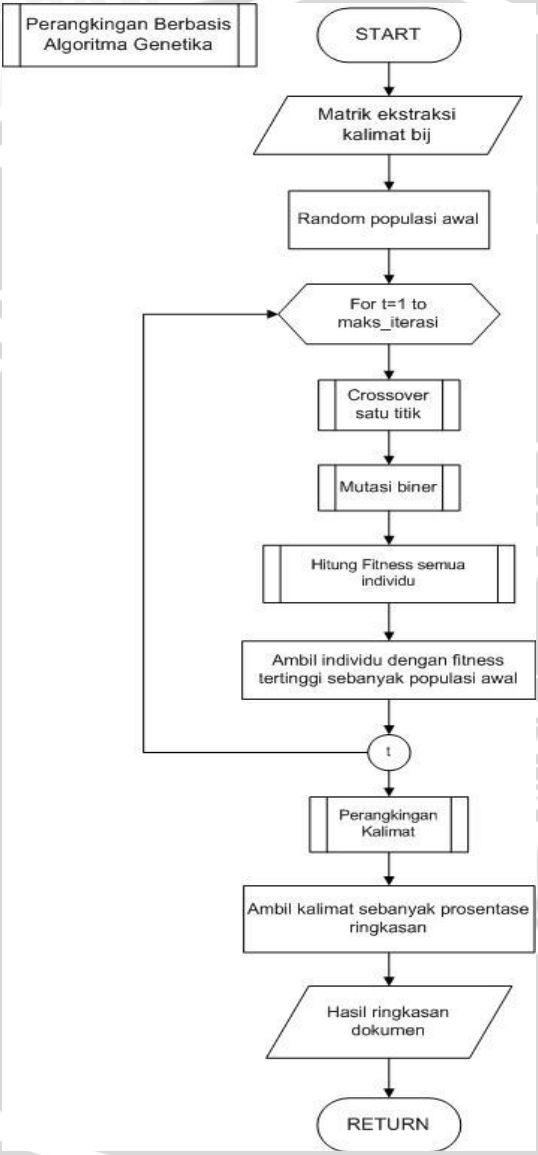
$$g_i = \sum_{j=1}^{n} W_j \ b_{ij} \quad (3.15)$$

Diagram alir proses perangkingan ditunjukkan gambar 3.16 berikut:



Gambar 3.16 Diagram Alir Proses Perangkingan

Berdasarkan proses yang telah dijelaskan, dapat digambarkan diagram alir proses perangkingan berbasis algoritma genetika seperti pada Gambar 3.17 berikut:



Gambar 3.17 Diagram Alir Proses Perangkingan Kalimat Berbasis Algoritma Genetika

3.2.4 *Merging* (Menggabungkan Hasil Ringkasan)

Merging merupakan proses untuk menggabungkan hasil ringkasan masing-masing dokumen yang ada dalam satu *cluster*. Pada tahap ini hasil ringkasan dari semua dokumen digabungkan menggunakan perhitungan *cosine similarity* dengan TF (*term frequency*). TF adalah frekuensi kata pada suatu kalimat atau dokumen. Tahap-tahap proses *merging* sebagai berikut:

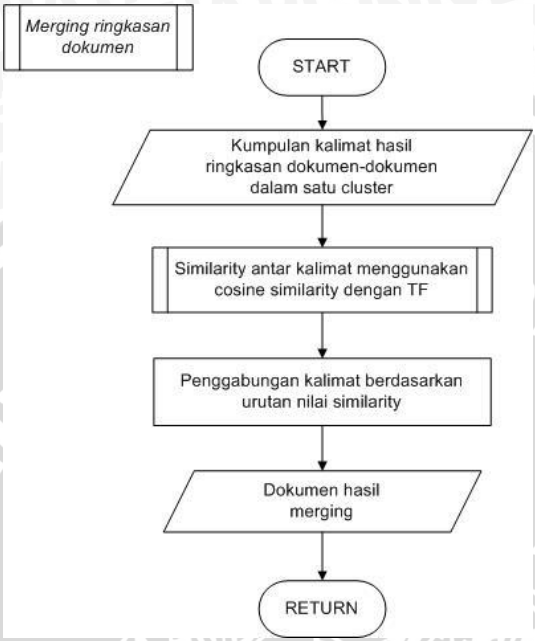
1. Masukan untuk proses ini adalah kumpulan kalimat dalam satu *cluster* hasil proses-proses sebelumnya.
2. Menghitung *similarity* antar kalimat menggunakan perhitungan *cosine similarity* dengan TF seperti ditunjukkan pada persamaan 3.16 berikut:

$$\text{Cosine}\,(S_i, S_j) = \frac{\sum_{k=1}^{n} t_{ik} \times t_{jk}}{\sqrt{\sum_{k=1}^{n} t_{ik}^2} \times \sqrt{\sum_{k=1}^{n} t_{jk}^2}} \qquad (3.16)$$

Dengan t_{ik} : jumlah kata k pada kalimat i
t_{jk} : jumlah kata k pada kalimat j

3. Kalimat digabung dengan mengurutkan nilai *similarity*nya.
4. Hasil proses ini yaitu ringkasan multi dokumen tiap *cluster*.

Diagram alir proses *merging* ditunjukkan oleh Gambar 3.18 berikut:



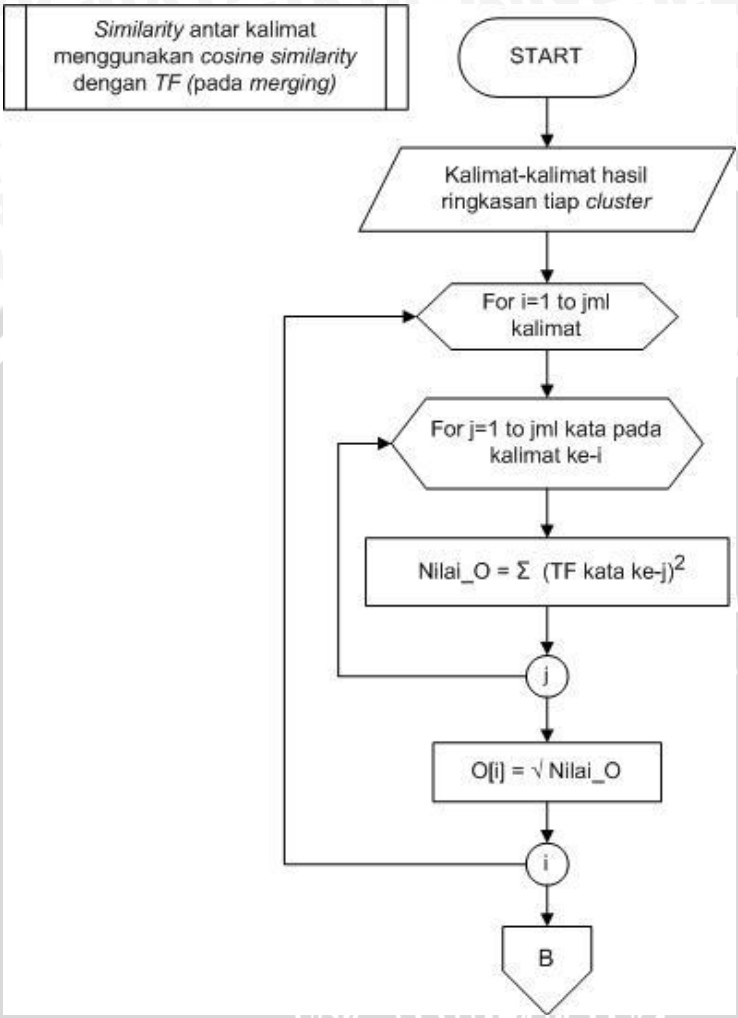
Gambar 3.18 Diagram Alir Proses *Merging*

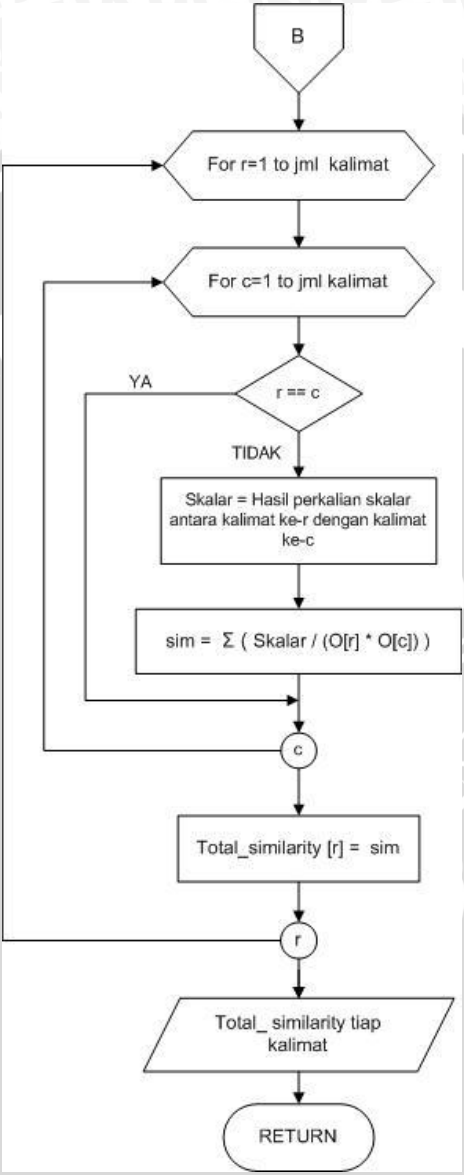
Proses penghitungan *similarity* merupakan sub proses dari proses *merging*. Pada proses *merging* kalimat, perhitungan *cosine similarity* hanya memperhitungkan TF (Frekuensi Kata) , meski pada umumnya perhitungan cosine *similarity* menggunanakan TF dikali IDF (*inverse document frequency*) atau TF dikali ISF (*inverse sentence frequency*). Tahapan perhitungan *cosine similarity* pada proses *merging* sebagai berikut:

1. Masukkan pada proses ini adalah kumpulan kalimat hasil ringkasan dalam satu *cluster*.
2. Untuk setiap kalimat dilakukan perhitungan Nilai_O. Nilai_O dihitung dengan menjumlahkan TF kuadrat semua kata pada suatu kalimat. Kemudian, menghitung O (penyebut pada persamaan 3.16) yang didapat dari akar Nilai_O.
3. Selanjutnya, menghitung skalar (pembilang dari persamaan 3.16) yang didapat dengan menjumlahkan hasil perkalian frekuensi kata yang sama antara dua kalimat (kalimat ke-r dan kalimat ke-c).

4. Menghitung sim (nilai *similarity* antara dua kalimat) dengan persamaan 3.16. Dilakukan untuk semua pasang kalimat. Jika ada 3 kalimat maka dilakukan perhitungan sim antara kalimat 1 dan 2, kalimat 1 dan 3, kalimat 2 dan 1, kalimat 2 dan 3, kalimat 3 dan 1, kalimat 3 dan 2.
5. Untuk setiap kalimat dihitung Total_similarity. Total_similarity didapat dengan menjumlahkan sim antara kalimat tersebut dengan kalimat lainnya. Misalnya ada 3 kalimat, Total_similarity kalimat 1 adalah sim kalimat 1 dengan 2 ditambah dengan sim kalimat 1 dan 3. Untuk kalimat 2 Total_similarity didapat dari sim antara kalimat 2 dengan kalimat 1 ditambah sim kalimat 2 dengan kalimat 3.
6. Hasil akhir proses ini adalah Total_similarity tiap kalimat.

Diagram Alir proses perhitungan similarity antar kalimat menggunakan cosine similarity dengan TF ditunjukkan oleh Gambar 3.19 berikut:


Similarity antar kalimat menggunakan *cosine similarity* dengan *TF* (pada *merging*)
START
Kalimat-kalimat hasil ringkasan tiap *cluster*
For i=1 to jml kalimat
For j=1 to jml kata pada kalimat ke-i
Nilai_O = Σ (TF kata ke-j)^2
j
O[i] = √ Nilai_O
i
B



Gambar 3.19 Diagram Alir Proses *Cosine Similarity* menggunakan TF

3.2.5 Evaluasi Hasil

Evaluasi untuk peringkasan teks dilakukan menggunakan tiga parameter yaitu *precision, recall,* dan *F-measure* untuk ringkasan dokumen dengan berbagai ukuran. Beberapa dokumen dengan berbagai topik akan diujikan agar data pengujiannya terdiri dari beberapa *cluster*.

3.3 Contoh Perhitungan Manual

Berikut merupakan dokumen yang diinputkan oleh *user* dengan ekstensi .txt. Ada 5 dokumen yang diinputkan dengan ekstensi .txt

Dokumen 1

Sony: PlayStation Network Hackers May Have Stolen Credit Card Data

Sony Corp. warns that during the hack that forced the shutdown of its PlayStation Network for the past week, users´ credit card information may have been stolen. The attack affected 77 million user accounts. While some players have brushed off the breach, industry experts say the scale was staggering and may end up costing Sony billions of dollars. Sony noted that it has no direct evidence that credit card information was stolen. However, the company says it can´t rule out the possibility. It says certain users´ names, birth dates, e-mail addresses and log-in information was taken, and purchase history and credit card billing address information may have also been stolen.

Dokumen 2

Sony Suffers Another Hacker Attack

Sony admits a second cyber-hack, this one giving attackers names, addresses, e-mail addresses, birth dates, phone numbers, and other information from 24.6 million PC games customers.

The attack targeted the Sony Online Entertainment (SOE) PC games network, which includes about 12,700 credit or debit card numbers and expiration dates from people outside the US.

Sony has has now disabled the online network and apologized once more to its customers.

Dokumen 3

Hackers Might Be Planning Another Attack on Sony

A news report claims that Sony might soon be the victim of another hackers´ attack. An IRC user said that he witnessed the hackers planning another attack for the weekend. The insider also said that the hackers were intending to publish at least some of the customer names, credit card numbers and addresses they would find. The hackers even claim that they have taken over some of Sony´s servers.

Dokumen 4

WHO: Most Frequent Reason of Death Is Chronic Disease

According to a report by the World Health Organization, the leading reason of death is a chronic disease such as diabetes, heart disease or cancer. The Global Status Report found that over 36 million people died of "non communicable diseases" in 2008. 80 percent of them lived in rather poor countries. The WHO recommended a better promotion of healthier diets and a stronger smoking legislation to prevent many of the deaths.

Dokumen 5

Smoking Could Increase Cancer Pain

A new research found that cancer patients who smoke may suffer stronger cancer pain than non-smokers. The study leaders asked 224 cancer patients to come to that conclusion. The researchers asked about the severity of the pain, distress from it and how the pain affects the patient´s everyday life. The study also found that cancer patients who quit smoking suffer less the more years ago they stopped. That highlights the doctor´s advice for cancer patients to stop smoking. The results of the study will be published next month in the journal "Pain".

3.3.1 *Preprocessing*

Proses yang pertama kali dilakukan adalah proses *preprocessing*. Setelah melalui proses *tokenizing*, *case folding*, *filtering* dan *stemming*, berikut hasil kata dasar penting (tanpa imbuhan) yang didapat :

Dokumen 1
sony corp warn hack force shutdown playstation network past week user credit card information stolen attack affect 77 million user account play brush breach industry expert scale stagger end cost sony billion dollar sony note direct evidence credit card information stolen company rule possibility user birth date email address login information purchase history credit card bill address information stolen

Dokumen 2
sony admit cyber hack attack address mail adress birth date phone number information 246 million pc game customer attack target sony online entertainment soe pc game network include 12700 credit debit card number expiration date people sony disable online network apologize customer

Dokumen 3
report claim sony might victim another hack attack. irc witness hack plan another attack weekend insider hack intend publish customer credit card number address find hack claim sony server

Dokumen 4
report world health organization lead reason death chronic disease diabetes heart disease cancer global status report found 36 million people die communicable disease 2008 80 percent live poor countries recommend promotion health diet strong smoke legislation prevent death

Dokumen 5
research found cancer patient smoke suffer strong cancer pain smoke study lead 224 cancer patient conclusion research severity pain, distress pain affects patient everyday life study cancer patient quit smoke suffer year stop highlight doctor advice cancer patient stop smoke results study publish month journal pain

Tabel 3.3 Daftar dan Frekuensi Kata Semua Dokumen

TOKEN	DOKUMEN				
	1	2	3	4	5
77	✓				
account	✓				
address	✓✓	✓✓	✓		
affect	✓				✓
attack	✓	✓✓	✓✓		
bill	✓				
billion	✓				
birth	✓	✓			
breach	✓				
brush	✓				
card	✓✓✓	✓	✓		
company	✓				
corp	✓				
cost	✓				
credit	✓✓✓	✓	✓		
date	✓	✓✓			
direct	✓				
dollar	✓				
email	✓	✓			
end	✓				
evidence	✓				
expert	✓				
force	✓				
hack	✓		✓✓✓✓		
history	✓				
industry	✓				
information	✓✓✓✓	✓			
login	✓				
million	✓	✓		✓	
network	✓	✓✓			
note	✓				
past	✓				
play	✓				
playstation	✓				
possibility	✓				

purchase	✓				
rule	✓				
scale	✓				
shutdown	✓				
sony	✓✓✓		✓✓		
stagger	✓				
stolen	✓✓✓				
user	✓✓✓				
warn	✓				
week	✓				
12700		✓			
246		✓			
admit		✓			
apologize		✓			
customer		✓✓	✓		
cyberhack		✓			
debit		✓			
disable		✓			
entertainment		✓			
expiration		✓			
game		✓✓			
give		✓			
include		✓			
number		✓✓	✓		
online		✓✓			
pc		✓✓			
people		✓		✓	
phone		✓			
soe		✓			
target		✓			
claim			✓✓		
find			✓		
insider			✓		
intend			✓		
irc			✓		
plan			✓		
publish			✓		
report			✓	✓✓	

server			✓		
victim			✓		
weekend			✓		
witness			✓		
2008				✓	
36				✓	
80				✓	
cancer				✓	✓✓✓✓✓
chronic				✓	
communicable				✓	
country				✓	
death				✓✓	
die				✓	
diabetes				✓	
diet				✓	
disease				✓✓✓	
found				✓	✓
global				✓	
health				✓✓	
heart				✓	
lead				✓	✓
legislation				✓	
live				✓	
organization				✓	
percent				✓	
poor				✓	
prevent				✓	
promot ion				✓	
reason				✓	
recommend				✓	
smoke				✓	✓✓✓✓
status				✓	
strong				✓	✓
world				✓	
advice					✓
ago					✓
conclusion					✓
distress					✓

doctor					✓
everyday					✓
highlight					✓
journal					✓
life					✓
month					✓
pain					✓✓✓✓
patient					✓✓✓✓✓
publish					✓
quit					✓
research					✓✓
result					✓
severity					✓
stop					✓✓
study					✓✓✓
suffer					✓✓
year					✓

3.3.2 *Clustering*

Setelah *preprocessing*, proses selanjutnya adalah *clustering* dokumen sesuai dengan konten atau isi dari dokumen menggunakan *single pass clustering*. *Single pass clustering* dihitung dengan TF.IDF yang telah dinormalisasi dan cosine similarity dengan *threshold* 0.085. Jika nilai similarity maksimum antara suatu dokumen dengan dokumen-dokumen lainnya lebih dari 0.085 maka dokumen tersebut masuk dalam satu cluster, jika sebaliknya maka membentuk *cluster* baru. Berikut akan dilakukan proses perhitungan *clustering* dokumen:

Perhitungan TF-IDF ternormalisasi:

Dokumen 1

Tabel 3.4 Perhitungan TF.IDF dokumen 1

kata (k)	TF_k	N	Df_k	IDF = $\log \frac{N}{Df_k}$	TF.IDF	$(TF.IDF)^2$
77	1	5	1	1.6990	1.6990	2.8865
account	1	5	1	1.6990	1.6990	2.8865
address	2	5	3	1.2218	2.4437	5.9717
affect	1	5	1	1.6990	1.6990	2.8865

attack	1	5	3	1.2218	1.2218	1.4929
bill	1	5	1	1.6990	1.6990	2.8865
billion	1	5	1	1.6990	1.6990	2.8865
birth	1	5	2	1.3979	1.3979	1.9542
breach	1	5	1	1.6990	1.6990	2.8865
brush	1	5	1	1.6990	1.6990	2.8865
card	3	5	3	1.2218	3.6655	13.4362
company	1	5	1	1.6990	1.6990	2.8865
corp	1	5	1	1.6990	1.6990	2.8865
cost	1	5	1	1.6990	1.6990	2.8865
credit	3	5	3	1.2218	3.6655	13.4362
date	1	5	2	1.3979	1.3979	1.9542
direct	1	5	1	1.6990	1.6990	2.8865
dollar	1	5	1	1.6990	1.6990	2.8865
email	1	5	2	1.3979	1.3979	1.9542
end	1	5	1	1.6990	1.6990	2.8865
evidence	1	5	1	1.6990	1.6990	2.8865
expert	1	5	1	1.6990	1.6990	2.8865
force	1	5	1	1.6990	1.6990	2.8865
hack	1	5	2	1.3979	1.3979	1.9542
history	1	5	1	1.6990	1.6990	2.8865
industry	1	5	1	1.6990	1.6990	2.8865
information	4	5	2	1.3979	5.5918	31.2678
login	1	5	1	1.6990	1.6990	2.8865
million	1	5	3	1.2218	1.2218	1.4929
network	1	5	2	1.3979	1.3979	1.9542
note	1	5	1	1.6990	1.6990	2.8865
past	1	5	1	1.6990	1.6990	2.8865
play	1	5	1	1.6990	1.6990	2.8865
playstation	1	5	1	1.6990	1.6990	2.8865
possibility	1	5	1	1.6990	1.6990	2.8865
purchase	1	5	1	1.6990	1.6990	2.8865
rule	1	5	1	1.6990	1.6990	2.8865
scale	1	5	1	1.6990	1.6990	2.8865
shutdown	1	5	1	1.6990	1.6990	2.8865
sony	3	5	3	1.2218	3.6655	13.4362
stagger	1	5	1	1.6990	1.6990	2.8865
stolen	3	5	1	1.6990	5.0969	25.9785

user	3	5	1	1.6990	5.0969	25.9785
warn	1	5	1	1.6990	1.6990	2.8865
week	1	5	1	1.6990	1.6990	2.8865
]JUMLAH						231.7436

$S_1 = \sqrt{\sum_{k=1}^{n}(\text{TF.IDF})_k^2} = \sqrt{231.7436} = 15.2231$

Normalisasi :

Tabel 3.5 TD.IDF yang dinormalisasikan pada dokumen 1

kata (k)	TF.IDF	$C_{1k} = \frac{\text{TF.IDF}}{S_1}$
77	1.6990	0.1116
account	1.6990	0.1116
address	2.4437	0.1605
affect	1.6990	0.1116
attack	1.2218	0.0803
bill	1.6990	0.1116
billion	1.6990	0.1116
birth	1.3979	0.0918
breach	1.6990	0.1116
brush	1.6990	0.1116
card	3.6655	0.2408
company	1.6990	0.1116
corp	1.6990	0.1116
cost	1.6990	0.1116
credit	3.6655	0.2408
date	1.3979	0.0918
direct	1.6990	0.1116
dollar	1.6990	0.1116
email	1.3979	0.0918
end	1.6990	0.1116
evidence	1.6990	0.1116
expert	1.6990	0.1116
force	1.6990	0.1116
hack	1.3979	0.0918
history	1.6990	0.1116
industry	1.6990	0.1116
information	5.5918	0.3673

login	1.6990	0.1116
million	1.2218	0.0803
network	1.3979	0.0918
note	1.6990	0.1116
past	1.6990	0.1116
play	1.6990	0.1116
playstation	1.6990	0.1116
possibility	1.6990	0.1116
purchase	1.6990	0.1116
rule	1.6990	0.1116
scale	1.6990	0.1116
shutdown	1.6990	0.1116
sony	3.6655	0.2408
stagger	1.6990	0.1116
stolen	5.0969	0.3348
user	5.0969	0.3348
warn	1.6990	0.1116
week	1.6990	0.1116

Dokumen 2

Tabel 3.6 Perhitungan TD.IDF dokumen 2

kata (k)	TF_k	N	Df_k	$IDF = \log \frac{N}{Df_k}$	TF.IDF	$(TF.IDF)^2$
address	2	5	3	1.2218	2.4437	5.9717
attack	2	5	3	1.2218	2.4437	5.9717
birth	1	5	2	1.3979	1.3979	1.9542
card	1	5	3	1.2218	1.2218	1.4929
credit	1	5	3	1.2218	1.2218	1.4929
date	2	5	2	1.3979	2.7959	7.8169
email	1	5	2	1.3979	1.3979	1.9542
information	1	5	2	1.3979	1.3979	1.9542
million	1	5	3	1.2218	1.2218	1.4929
network	2	5	2	1.3979	2.7959	7.8169
sony	3	5	3	1.2218	3.6655	13.4362
12700	1	5	1	1.6990	1.6990	2.8865
246	1	5	1	1.6990	1.6990	2.8865
admit	1	5	1	1.6990	1.6990	2.8865

apologize	1	5	1	1.6990	1.6990	2.8865
customer	2	5	2	1.3979	2.7959	7.8169
cyberhack	1	5	1	1.6990	1.6990	2.8865
debit	1	5	1	1.6990	1.6990	2.8865
disable	1	5	1	1.6990	1.6990	2.8865
entertainment	1	5	1	1.6990	1.6990	2.8865
expiration	1	5	1	1.6990	1.6990	2.8865
game	2	5	1	1.6990	3.3979	11.5460
give	1	5	1	1.6990	1.6990	2.8865
include	1	5	1	1.6990	1.6990	2.8865
number	2	5	2	1.3979	2.7959	7.8169
online	2	5	1	1.6990	3.3979	11.5460
pc	2	5	1	1.6990	3.3979	11.5460
people	1	5	2	1.3979	1.3979	1.9542
phone	1	5	2	1.3979	1.3979	1.9542
soe	1	5	1	1.6990	1.6990	2.8865
target	1	5	1	1.6990	1.6990	2.8865
JUMLAH						143.0597

$S_2 = \sqrt{\sum_{k=1}^{n}(\text{TF.IDF})_k^2} = \sqrt{143.0597} = 11.9608$

Normalisasi :

Tabel 3.7 TF.IDF yang dinormalisasikan pada dokumen 2

kata (k)	TF.IDF	$C_{2k} = \frac{\text{TF.IDF}}{S_2}$
address	2.4437	0.2043
attack	2.4437	0.2043
birth	1.3979	0.1169
card	1.2218	0.1022
credit	1.2218	0.1022
date	2.7959	0.2338
email	1.3979	0.1169
information	1.3979	0.1169
million	1.2218	0.1022
network	2.7959	0.2338
sony	3.6655	0.3065
12700	1.6990	0.1420
246	1.6990	0.1420

admit	1.6990	0.1420
apologize	1.6990	0.1420
customer	2.7959	0.2338
cyberhack	1.6990	0.1420
debit	1.6990	0.1420
disable	1.6990	0.1420
entertainment	1.6990	0.1420
expiration	1.6990	0.1420
game	3.3979	0.2841
give	1.6990	0.1420
include	1.6990	0.1420
number	2.7959	0.2338
online	3.3979	0.2841
pc	3.3979	0.2841
people	1.3979	0.1169
phone	1.3979	0.1169
soe	1.6990	0.1420
target	1.6990	0.1420

Perhitungan TF-IDF ternormalisasi untuk dokumen 3, dokumen 4 dan dokumen 5 dilakukan dengan cara yang sama.

Clustering dokumen

- Input dokumen 1:
  Dokumen 1 masuk ke dalam *cluster* 1.

- Input dokumen 2:
  Dokumen 2 dibandingkan dengan *cluster* 1 (dokumen 1).
  Sim (dokumen 2, dokumen 1) = $\sum_{kata\ yang\ sama} C_{1k} \times C_{2k}$
  address → 0.1605 × 0.2043 = 0.0328
  attack → 0.0803 × 0.2043 = 0.0164
  birth → 0.0918 × 0.1169 = 0.0107
  card → 0.2408 × 0.1022 = 0.0246
  credit → 0.2408 × 0.1022 = 0.0246
  date → 0.0918 × 0.2338 = 0.0215
  email → 0.0918 × 0.1169 = 0.0107
  information → 0.3673 × 0.1169 = 0.0429
  million → 0.0803 × 0.1022 = 0.0082

network → 0.0918 × 0.2338 = 0.0215
sony → 0.2408 × 0.3065 = 0.0738
---------- +
Total : 0.1843 (MAX)

Karena nilai *similarity* maksimum lebih besar dari *threshold* (0.1843 > 0.085) maka dokumen 2 masuk pada *cluster* 1.

- Input dokumen 3

  Dokumen 3 dibandingkan dengan *cluster* 1 (dokumen 1 dan dokumen 2).
  Sim (dokumen 3, dokumen 1) = 0.2170 (MAX)
  Sim (dokumen 3, dokumen 2) = 0.1821
  Karena nilai *similarity* maksimum lebih besar dari *threshold* (0.2170 > 0.085) maka dokumen 3 masuk pada *cluster* 1.

- Input dokumen 4

  Dokumen 4 dibandingkan dengan *cluster* 1 (dokumen 1, dokumen 2 dan dokumen 3).
  Sim (dokumen 4, dokumen 1) = 0.0085
  Sim (dokumen 4, dokumen 2) = 0.0248
  Sim (dokumen 4, dokumen 3) = 0.0351 (MAX)
  Karena nilai *similarity* maksimum lebih kecil dari *threshold* (0.0351 < 0.085) maka dokumen 4 masuk pada *cluster* baru, *cluster* 2.

- Input dokumen 5

  Dokumen 5 dibandingkan dengan *cluster* 1(dokumen 1, dokumen 2, dokumen 3) dan *cluster* 2(dokumen 4).
  - Dokumen 5 dibandingkan dengan *cluster* 1.
    Sim (dokumen 5, dokumen 1) = 0.0087
    Sim (dokumen 5, dokumen 2) = 0
    Sim (dokumen 5, dokumen 3) = 0.0139 (MAX)

  - Dokumen 5 dibandingkan dengan *cluster* 2.
    Sim (dokumen 5, dokumen 4) = 0.1328 (MAX)

  Karena nilai *similarity* dokumen 5 dengan *cluster* 2 lebih besar daripada nilai *similarity* dokumen 5 dengan *cluster* 1 maka nilai *similarity* dengan *cluster* 2 dibandingkan dengan *threshold*. Nilai

similarity dokumen 5 dengan *cluster* 2 lebih besar dari *threshold* (0.1328 > 0.05) maka dokumen 5 masuk pada *cluster* 2.

Hasil *clustering*:
Cluster – 1 : Dokumen 1, Dokumen 2, Dokumen 3
Cluster – 2 : Dokumen 4, Dokumen 5

3.3.3 Peringkasan Dokumen

Di bawah ini akan dilakukan peringkasan untuk dokumen-dokumen pada *Cluster* – 1.

3.3.3.1 Ektraksi Kalimat

Di bawah ini akan dilakukan contoh perhitungan untuk dokumen 1.

Sony Corp warn hack force shutdown PlayStation Network week credit card information stole. attack affect 77 million account. play brush breach industry expert scale stagger cost Sony billion dollar. Sony note direct evidence credit card information stole. Company rule possibility. birth date mail address log information purchase history credit card bill address information stole.

Tabel 3.8 Daftar dan Frekuensi Kata Dokumen 1

kata (k)	KALIMAT					
	1	2	3	4	5	6
77		✓				
account		✓				
address						✓✓
affect		✓				
attack		✓				
bill						✓
billion			✓			
birth						✓
breach			✓			
brush			✓			
card	✓			✓		✓

company					✓	
corp	✓					
cost			✓			
credit	✓			✓		✓
date						✓
direct				✓		
dollar			✓			
email						✓
end			✓			
evidence				✓		
expert			✓			
force	✓					
hack	✓					
history						✓
industry			✓			
information	✓			✓		✓✓
login						✓
million		✓				
network	✓					
note				✓		
past	✓					
play	✓					
playstation	✓					
possibility					✓	
purchase						✓
rule					✓	
scale			✓			
shutdown	✓					
sony	✓		✓	✓		
stagger			✓			
stolen	✓			✓		✓
user	✓	✓				✓
warn	✓					
week	✓					

Selanjutnya dilakukan ekstraksi kalimat menggunakan enam fitur untuk memberi nilai agar dapat dilakukan perangkingan menggunakan algoritma genetika sehingga semua kalimat dapat

dirangking dan diringkas sesuai presentase yang diinginkan *user*. Berikut proses ekstraksi kalimat pada *cluster* ke – 1 :

1. Panjang Kalimat

   Kalimat ke-1 : 16
   Kalimat ke-2 : 6
   Kalimat ke-3 : 11
   Kalimat ke-4 : 8
   Kalimat ke-5 : 3
   Kalimat ke-6 : 15
   Kalimat terpanjang : 16

   Skor :
   Kalimat ke-1 : 16 / 16 = 1
   Kalimat ke-2 : 6 / 16 = 0.375
   Kalimat ke-3 : 11 / 16 = 0.6875
   Kalimat ke-4 : 8 / 16 = 0.5
   Kalimat ke-5 : 3 / 16 = 0.1875
   Kalimat ke-6 : 15 / 16 = 0.9375

2. *Term Weight* (PERHITUNGAN TF-ISF)

   Kalimat ke – 1

   Tabel 3.9 Penghitungan Nilai *Term Weight* Kalimat 1 Dokumen 1

kata (k)	**Tfk**	**N**	**nk**	**$O_k=Tf_k*\log(N/n_k)$**
card	1	6	3	1.3010
corp	1	6	1	1.7782
credit	1	6	3	1.3010
force	1	6	1	1.7782
hack	1	6	1	1.7782
information	1	6	2	1.4771
network	1	6	1	1.7782
past	1	6	1	1.7782
play	1	6	1	1.7782
playstation	1	6	1	1.7782
shutdown	1	6	1	1.7782
sony	1	6	3	1.3010

stolen	1	6	3	1.3010
user	1	6	3	1.3010
warn	1	6	1	1.7782
week	1	6	1	1.7782
			JUMLAH	23.9856

Kalimat ke – 2

Tabel 3.10 Penghitungan Nilai *Term Weight* Kalimat 2

Dokumen 1

kata (k)	**Tfk**	**N**	**nk**	**$O_k=Tf_k*\log(N/n_k)$**
attack	1	6	1	1.7782
affect	1	6	1	1.7782
77	1	6	1	1.7782
million	1	6	1	1.7782
user	1	6	3	1.3010
account	1	6	1	1.7782
			JUMLAH	10.1918

Kalimat ke – 3

Tabel 3.11 Penghitungan Nilai *Term Weight* Kalimat 3

Dokumen 1

kata (k)	**Tfk**	**N**	**nk**	**$O_k=Tf_k*\log(N/n_k)$**
billion	1	6	1	1.7782
breach	1	6	1	1.7782
brush	1	6	1	1.7782
cost	1	6	1	1.7782
dollar	1	6	1	1.7782
end	1	6	1	1.7782
expert	1	6	1	1.7782
industry	1	6	1	1.7782
scale	1	6	1	1.7782
sony	1	6	3	1.3010
stagger	1	6	1	1.7782
			JUMLAH	19.0825

Kalimat ke – 4

Tabel 3.12 Penghitungan Nilai *Term Weight* Kalimat 4 Dokumen 1

kata (k)	**Tfk**	**N**	**nk**	**$O_k=Tf_k*\log(N/n_k)$**
sony	1	6	3	1.3010
note	1	6	1	1.7782
direct	1	6	1	1.7782
evidence	1	6	1	1.7782
credit	1	6	3	1.3010
card	1	6	3	1.3010
information	1	6	3	1.3010
stolen	1	6	3	1.3010
JUMLAH				11.8396

Kalimat ke – 5

Tabel 3.13 Penghitungan Nilai *Term Weight* Kalimat 5 Dokumen 1

kata (k)	**Tfk**	**N**	**nk**	**$O_k=Tf_k*\log(N/n_k)$**
company	1	6	1	1.7782
rule	1	6	1	1.7782
possibility	1	6	1	1.7782
JUMLAH				5.3345

Kalimat ke – 6

Tabel 3.14 Penghitungan Nilai *Term Weight* Kalimat 6 Dokumen 1

kata (k)	**Tfk**	**N**	**nk**	**$O_k=Tf_k*\log(N/n_k)$**
birth	1	6	1	1.7782
date	1	6	1	1.7782
email	1	6	1	1.7782
address	2	6	1	2.5563
login	1	6	1	1.7782
information	2	6	3	1.6021
purchase	1	6	1	1.7782
history	1	6	1	1.7782
credit	1	6	3	1.3010
card	1	6	3	1.3010
bill	1	6	1	1.7782

user	1	6	3	1.3010
stolen	1	6	3	1.3010
			JUMLAH	21.8095

Tabel 3.15 Nilai *Term Weight* Semua Kalimat Dokumen 1

Kalimat ke – s	$\sum_{k=1}^{n} O_k$
1	23.9856
2	10.1918
3	19.0825
4	11.8396
5	5.3345
6	21.8095
Max	23.9856

Skor :

Kalimat ke-1 : 23.9856 / 23.9856 = 1
Kalimat ke-2 : 10.1918 / 23.9856 = 0.4249
Kalimat ke-3 : 19.0825 / 23.9856 = 0.7956
Kalimat ke-4 : 11.8396 / 23.9856 = 0.4936
Kalimat ke-5 : 5.3345 / 23.9856 = 0.2224
Kalimat ke-6 : 21.8095 / 23.9856 = 0.9093

3. Kesamaan antar kalimat

3.1 Menghitung perkalian skalar antar kalimat ke- i dengan kalimat lainnya. Hasil perkalian tersebut dijumlahkan.

Kalimat ke – 1

Tabel 3.16 Perkalian Skalar Antara Kalimat 1 dan Kalimat Lainnya pada Dokumen 1

KATA (K1)	K1 * Ki				
	K2	K3	K4	K5	K6
card	0	0	1.6927	0	1.6927
corp	0	0	0	0	0
credit	0	0	1.6927	0	1.6927
force	0	0	0	0	0
hack	0	0	0	0	0
information	0	0	1.9218	0	2.3664
network	0	0	0	0	0
past	0	0	0	0	0

play	0	0	0	0	0
playstation	0	0	0	0	0
shutdown	0	0	0	0	0
sony	0	1.6927	1.6927	0	0
stolen	0	0	1.6927	0	1.6927
user	1.6927	0	0	0	1.6927
warn	0	0	0	0	0
week	0	0	0	0	0
	1.6927	1.6927	8.6925	0.0000	9.1372

Kalimat ke – 2

Tabel 3.17 Perkalian Skalar Antara Kalimat 2 dan Kalimat Lainnya pada Dokumen 1

KATA (K2)	K2 * Ki				
	K1	K3	K4	K5	K6
attack	0	0	0	0	0
affect	0	0	0	0	0
77	0	0	0	0	0
million	0	0	0	0	0
user	1.6927	0	0	0	1.6927
account	0	0	0	0	0
	1.6927	0	0	0	1.6927

Kalimat ke – 3

Tabel 3.18 Perkalian Skalar Antara Kalimat 3 dan Kalimat Lainnya pada Dokumen 1

KATA (K3)	K3 * Ki				
	K1	K2	K4	K5	K6
billion	0	0	0	0	0
breach	0	0	0	0	0
brush	0	0	0	0	0
cost	0	0	0	0	0
dollar	0	0	0	0	0
end	0	0	0	0	0
expert	0	0	0	0	0
industry	0	0	0	0	0
scale	0	0	0	0	0
sony	1.6927	0	1.6927	0	0

stagger	0	0	0	0	0
	1.6927	0	1.6927	0	0

Kalimat ke – 4

Tabel 3.19 Perkalian Skalar Antara Kalimat 4 dan Kalimat Lainnya pada Dokumen 1

KATA (K4)	K4 * Ki				
	K1	K2	K3	K5	K6
sony	1.6927	0	1.6927	0	0
note	0	0	0	0	0
direct	0	0	0	0	0
evidence	0	0	0	0	0
credit	1.6927	0	0	0	1.6927
card	1.6927	0	0	0	1.6927
information	1.9218	0	0	0	2.0843
stolen	1.6927	0	0	0	1.6927
	8.6925	0	0	0	7.1624

Kalimat ke – 5

Tabel 3.20 Perkalian Skalar Antara Kalimat 5 dan Kalimat Lainnya pada Dokumen 1

KATA (K5)	K5 * Ki				
	K1	K2	K3	K4	K6
company	0	0	0	0	0
rule	0	0	0	0	0
possibility	0	0	0	0	0
	0	0	0	0	0

Kalimat ke – 6

Tabel 3.21 Perkalian Skalar Antara Kalimat 6 dan Kalimat Lainnya pada Dokumen 1

KATA (K6)	K6 * Ki				
	K1	K2	K3	K4	K5
birth	0	0	0	0	0
date	0	0	0	0	0
email	0	0	0	0	0
address	0	0	0	0	0
login	0	0	0	0	0

information	2.3664	0	0	2.0843	0
purchase	0	0	0	0	0
history	0	0	0	0	0
credit	1.6927	0	0	1.6927	0
card	1.6927	0	0	1.6927	0
bill	0	0	0	0	0
user	1.6927	1.6927	0	0	0
stolen	1.6927	0	0	1.6927	0
	9.1372	1.6927	0	7.1624	0

3.2 Menghitung panjang setiap kalimat. Mengkuadratkan bobot setiap kata pada kalimat lalu menjumlahkan nilai bobot semua kata pada kalimat kemudian nilai tersebut diakarkan.

Kalimat ke – 1

Tabel 3.22 Penghitungan Kuadrat *Term Weight* Kalimat 1 Dokumen 1

kata (k)	**Tfk**	**N**	**nk**	**$O_k=Tf_k*\log(N/n_k)$**	**$(O_k)^2$**
card	1	6	3	1.3010	1.6927
corp	1	6	1	1.7782	3.1618
credit	1	6	3	1.3010	1.6927
force	1	6	1	1.7782	3.1618
hack	1	6	1	1.7782	3.1618
information	1	6	2	1.4771	2.1819
network	1	6	1	1.7782	3.1618
past	1	6	1	1.7782	3.1618
play	1	6	1	1.7782	3.1618
playstation	1	6	1	1.7782	3.1618
shutdown	1	6	1	1.7782	3.1618
sony	1	6	3	1.3010	1.6927
stolen	1	6	3	1.3010	1.6927
user	1	6	3	1.3010	1.6927
warn	1	6	1	1.7782	3.1618
week	1	6	1	1.7782	3.1618
				JUMLAH	39.1017

Kalimat ke – 2

Tabel 3.23 Penghitungan Kuadrat *Term Weight* Kalimat 2 Dokumen 1

kata (k)	Tfk	N	nk	$O_k=Tf_k*\log(N/n_k)$	$(O_k)^2$
attack	1	6	1	1.7782	3.1618
affect	1	6	1	1.7782	3.1618
77	1	6	1	1.7782	3.1618
million	1	6	1	1.7782	3.1618
user	1	6	3	1.3010	1.6927
account	1	6	1	1.7782	3.1618
				JUMLAH	17.5018

Kalimat ke – 3

Tabel 3.24 Penghitungan Kuadrat *Term Weight* Kalimat 3 Dokumen 1

kata (k)	Tfk	N	nk	$O_k=Tf_k*\log(N/n_k)$	$(O_k)^2$
billion	1	6	1	1.7782	3.1618
breach	1	6	1	1.7782	3.1618
brush	1	6	1	1.7782	3.1618
cost	1	6	1	1.7782	3.1618
dollar	1	6	1	1.7782	3.1618
end	1	6	1	1.7782	3.1618
expert	1	6	1	1.7782	3.1618
industry	1	6	1	1.7782	3.1618
scale	1	6	1	1.7782	3.1618
sony	1	6	3	1.3010	1.6927
stagger	1	6	1	1.7782	3.1618
				JUMLAH	33.3109

Kalimat ke – 4

Tabel 3.25 Penghitungan Kuadrat *Term Weight* Kalimat 4 Dokumen 1

kata (k)	Tfk	N	nk	$O_k=Tf_k*\log(N/n_k)$	$(O_k)^2$
sony	1	6	3	1.3010	1.6927
note	1	6	1	1.7782	3.1618
direct	1	6	1	1.7782	3.1618
evidence	1	6	1	1.7782	3.1618
credit	1	6	3	1.3010	1.6927

card	1	6	3	1.3010	1.6927
information	1	6	3	1.3010	1.6927
stolen	1	6	3	1.3010	1.6927
				JUMLAH	17.9489

Kalimat ke – 5

Tabel 3.26 Penghitungan Kuadrat *Term Weight* Kalimat 5 Dokumen 1

kata (k)	**Tf_k**	**N**	**n_k**	**$O_k=Tf_k*\log(N/n_k)$**	**$(O_k)^2$**
company	1	6	1	1.7782	3.1618
rule	1	6	1	1.7782	3.1618
possibility	1	6	1	1.7782	3.1618
JUMLAH					9.4855

Kalimat ke – 6

Tabel 3.27 Penghitungan Kuadrat *Term Weight* Kalimat 6 Dokumen 1

kata (k)	**Tf_k**	**N**	**n_k**	**$O_k=Tf_k*\log(N/n_k)$**	**$(O_k)^2$**
birth	1	6	1	1.7782	3.1618
date	1	6	1	1.7782	3.1618
email	1	6	1	1.7782	3.1618
address	2	6	1	2.5563	6.5347
login	1	6	1	1.7782	3.1618
information	2	6	3	1.6021	2.5666
purchase	1	6	1	1.7782	3.1618
history	1	6	1	1.7782	3.1618
credit	1	6	3	1.3010	1.6927
card	1	6	3	1.3010	1.6927
bill	1	6	1	1.7782	3.1618
user	1	6	3	1.3010	1.6927
stolen	1	6	3	1.3010	1.6927
				JUMLAH	38.0047

Tabel 3.28 Penghitungan Akar dari Kuadrat *Term Weight* Semua Kalimat Dokumen 1

Kalimat ke – s	$\sum_{k=1}^{N} {O_k}^2$	$\sqrt{\sum_{k=1}^{N} {O_k}^2}$
1	39.1017	6.2531
2	17.5018	4.1835
3	33.3109	5.7716
4	17.9498	4.2366
5	9.4855	3.0798
6	38.0047	6.1648

3.3 Menerapkan rumus *cosine similarity*, kemudian dihitung kesamaan antar kalimat. Untuk kalimat 1:

Kalimat 1 dengan Kalimat 2 = $\frac{1.6927}{6.2531 \text{ x } 4.1835} = 0.0647$

Kalimat 1 dengan Kalimat 3 = $\frac{1.6927}{6.2531 \text{ x } 5.7716} = 0.0469$

Kalimat 1 dengan Kalimat 4 = $\frac{8.6925}{6.2531 \text{ x } 4.2366} = 0.3281$

Kalimat 1 dengan Kalimat 5 = $\frac{0}{6.2531 \text{ x } 3.0798} = 0$

Kalimat 1 dengan Kalimat 6 = $\frac{9.1372}{6.2531 \text{ x } 6.1648} = 0.2370$

Nilai similarity kalimat 1 = 0.0647 + 0.0469 + 0.3281 + 0 + 0.2370

= 0.6767

Dilakukan proses yang sama untuk kalimat lainnya. Nilai selengkapnya ditunjukkan pada proses berikut :

Kalimat 1

	Kalimat 2	Kalimat 3	Kalimat 4	Kalimat 5	Kalimat 6	JUMLAH
Kalimat 1	0.0647	0.0469	0.3281	0	0.2370	0.6767

Kalimat 2

	Kalimat 2	Kalimat 3	Kalimat 4	Kalimat 5	Kalimat 6	JUMLAH
Kalimat 1	0.0647	0	0	0	0.0656	0.1303

Kalimat 3

	Kalimat 2	Kalimat 3	Kalimat 4	Kalimat 5	Kalimat 6	JUMLAH
Kalimat 1	0.0469	0	0.0692	0	0	0.1161

Kalimat 4

	Kalimat 2	Kalimat 3	Kalimat 4	Kalimat 5	Kalimat 6	JUMLAH
Kalimat 1	0.3281	0	0	0	0.2742	0.6023

Kalimat 5

	Kalimat 2	Kalimat 3	Kalimat 4	Kalimat 5	Kalimat 6	JUMLAH
Kalimat 5	0	0	0	0	0	0

Kalimat 6

	Kalimat 2	Kalimat 3	Kalimat 4	Kalimat 5	Kalimat 6	JUMLAH
Kalimat 1	0.2370	0.0656	0	0.2742	0	0.5769

Tabel 3.29 Nilai *Similarity* Semua Kalimat Dokumen 1

Kalimat ke –i	$\sum SIM(Si, Sj)$
1	0.6767
2	0.1303
3	0.1161
4	0.6023
5	0
6	0.5769
MAX	0.6767

Skor :
Kalimat ke-1 : 0.6767 / 0.6767 = 1
Kalimat ke-2 : 0.1303 / 0.6767 = 0.1926
Kalimat ke-3 : 0.1161 / 0.6767 = 0.1716`
Kalimat ke-4 : 0.6023 / 0.6767 = 0.8901
Kalimat ke-5 : 0 / 0.6767 = 0
Kalimat ke-6 : 0.5769 / 0.6767 = 0.8524

4. *Proper Noun*

Tabel 3.30 Daftar dan Frekuensi *Proper Noun* Dokumen 1

Proper Noun	K 1	K 2	K 3	K 4	K 5	K 6
sony	✓		✓	✓		
corp	✓					
playstation	✓					
network	✓					

Jumlah kata pada :
- Kalimat ke-1 : 16
- Kalimat ke-2 : 6
- Kalimat ke-3 : 11
- Kalimat ke-4 : 8
- Kalimat ke-5 : 3
- Kalimat ke-6 : 15

Skor :
Kalimat ke-1 : 4 / 16 = 0.25
Kalimat ke-2 : 0 / 6 = 0
Kalimat ke-3 : 1 / 11 = 0.0909

Kalimat ke-4 : 1 / 8 = 0.125
Kalimat ke-5 : 0 / 3 = 0
Kalimat ke-6 : 0 / 15 = 0

5. Kata Tematik

Tabel 3.31 Daftar dan Frekuensi Kata Tematik Dokumen 1

Thematic Word	frekuensi	K1	K2	K3	K4	K5	K6
address	2						✓✓
card	3	✓			✓		✓
credit	3	✓			✓		✓
information	4	✓			✓		✓✓
stolen	3	✓		✓	✓		
sony	3	✓			✓		✓
user	3	✓	✓				✓
	Jumlah	6	1	1	5	0	8

MAX = 8

Skor :
Kalimat ke-1 : 6 / 8 = 0.75
Kalimat ke-2 : 1 / 8 = 0.125
Kalimat ke-3 : 1 / 8 = 0.125
Kalimat ke-4 : 5 / 8 = 0.625
Kalimat ke-5 : 0 / 8 = 0
Kalimat ke-6 : 8 / 8 = 1

6. Data Numerik

Tabel 3.32 Daftar dan Frekuensi Data Numerik Dokumen 1

Data Numerik	K1	K2	K3	K4	K5	K6
77		✓				
jumlah	0	1	0	0	0	0

Jumlah kata pada :
- kalimat ke-1 : 16
- kalimat ke-2 : 6
- kalimat ke-3 : 11
- kalimat ke-4 : 8

- kalimat ke-5 : 3
- kalimat ke-6 : 15

Skor :

Kalimat ke-1 : 0 / 16 = 0
Kalimat ke-2 : 1 / 6 = 0.1667
Kalimat ke-3 : 0 / 11 = 0
Kalimat ke-4 : 0 / 8 = 0
Kalimat ke-5 : 0 / 3 = 0
Kalimat ke-6 : 0 / 15 = 0

Tabel 3.33 Nilai Hasil Ekstraksi Semua Kalimat Pada Dokumen 1

Kalimat ke-s	Fitur 1	Fitur 2	Fitur 3	Fitur 4	Fitur 5	Fitur 6
Kalimat 1	1	1	1	0.25	0.75	0
Kalimat 2	0.375	0.424	0.193	0	0.125	0.167
Kalimat 3	0.687	0.796	0.172	0.091	0.125	0
Kalimat 4	0.5	0.494	0.89	0.125	0.625	0
Kalimat 5	0.187	0.222	0	0	0	0
Kalimat 6	0.937	0.909	0.852	0	1	0

3.3.3.2 Perangkingan Berbasis Algoritma Genetika

Proses Pengambilan Keputusan untuk Merangking Kalimat Menggunakan Algoritma Genetika Sesuai Bobot Atribut (Bobot Fitur)

Pembangkitan populasi awal

Populasi *encoding* ke bilangan biner dibangkitkan secara random.
Populasi Size (ukuran populasi) = 10 individu
Karena nilai atribut pada interval [0 1] maka jumlah gen untuk setiap kromosom adalah :

$L_i = \lceil 2\log[(b-a)10^2 + 1]] = \lceil 2\log[(1-0)10^2 + 1]] = 7$

Untuk 6 atribut (6 fitur) panjang kromosom adalah 7 × 6 = 42

Matriks B :

$$\begin{array}{c} \\ K_1 \\ K_2 \\ K_3 \\ K_4 \\ K_5 \\ K_6 \end{array} \begin{array}{c} \begin{array}{cccccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0.25 & 0.75 & 0 \\ 0.375 & 0.424 & 0.193 & 0 & 0.125 & 0.167 \\ 0.687 & 0.796 & 0.172 & 0.091 & 0.125 & 0 \\ 0.5 & 0.494 & 0.89 & 0.125 & 0.625 & 0 \\ 0.187 & 0.222 & 0 & 0 & 0 & 0 \\ 0.937 & 0.909 & 0.852 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Tabel 3.34 Populasi Awal (Generasi 1)

No	Populasi					
1	1001010	0011011	1001101	1100101	0101100	1101001
2	0101011	0011101	1110010	0011011	1101100	0100110
3	0010110	1101001	0101001	1010110	1001010	1001101
4	1100101	0011010	0110010	0010110	0011001	0011011
5	1100101	1101001	1100100	0101110	1000110	0010011
6	1011001	0110010	1100110	0011010	0011011	0110010
7	0011001	0011001	0011001	0011001	1101001	1010110
8	1100100	1011001	1010101	0010011	0011101	0011010
9	0011101	0011001	0010101	1100101	0011110	1011100
10	0011110	1100110	1000110	1101110	0001101	0011110

Tabel 3.35 Nilai Desimal Generasi 1

No	Kromosom					
1	74	29	77	94	44	106
2	43	29	114	27	108	38
3	22	105	41	86	74	77
4	101	26	50	22	25	27
5	101	105	100	46	70	19
6	89	50	102	26	27	50
7	25	25	25	25	105	86
8	100	89	85	19	29	26
9	29	25	21	101	30	92
10	30	102	70	110	13	30

Probability *Crossover* = 0.8

Proses *crossover* tergantung pada suatu parameter yaitu probabilitas *crossover* (Pc). Misalkan probabilitas *crossover* adalah 0.8, artinya diharapkan 80% individu yang akan mengalami *crossover*.

Langkah pertama yang dilakukan adalah membangkitkan nilai *random* antara 0 sampai 1. Jika nilai random < 0,8 maka individu tersebut yang akan dikenai proses *crossover*. Misalkan nilai random yang didapat adalah 0,27; 0,86; 0,83; 0,14; 0,85; 0,96; 0.35; 0.81; 0.95; 0.93 Dari nilai random yang didapat, maka diperoleh 3 individu yang akan mengalami proses *crossover*, yaitu individu 1, individu 4, individu 7.

Dibangkitkan bilangan random antara 0 - 42 (42 adalah jumlah gen tiap individu) untuk menentukan posisi *crossover* satu titik, misalkan 32, maka dari gen urutan 32 ke kanan di*crossover*.

Individu 1 dan 4

1001010 0011101 1001101 1011110 0101100 1101010
1100101 0011010 0110010 0010110 0011001 0011011
Menjadi :
1001010 0011101 1001101 1011110 0101001 0011011
1100101 0011010 0110010 0010110 0011100 1101010

Individu 1 dan 7

1001010 0011101 1001101 1011110 0101100 1101010
0011001 0011001 0011001 0011001 1101001 1010110
Menjadi :
1001010 0011101 1001101 1011110 0101001 1010110
0011001 0011001 0011001 0011001 1101100 1101010

Individu 4 dan 7

1100101 0011010 0110010 0010110 0011001 0011011
0011001 0011001 0011001 0011001 1101001 1010110
Menjadi :
1100101 0011010 0110010 0010110 0011001 1010110
0011001 0011001 0011001 0011001 1101001 0011011

Didapatkan 6 individu baru :

Tabel 3.36 Individu Baru hasil *Crosssover* Generasi 1

No	Kromosom					
1	1001010	0011101	1001101	1011110	0101001	0011011
2	1100101	0011010	0110010	0010110	0011100	1101010
3	1001010	0011101	1001101	1011110	0101001	1010110
4	0011001	0011001	0011001	0011001	1101100	1101010
5	1001010	0011101	1001101	1011110	0101001	1010110
6	0011001	0011001	0011001	0011001	1101100	1101010

Mutasi = 0.01

Proses mutasi tergantung pada suatu parameter yaitu probabilitas mutasi (Pc). Misalkan probabilitas mutasi adalah 0.01, artinya diharapkan 1% dari total gen yang akan mengalami mutasi.

Langkah pertama yang dilakukan adalah membangkitkan nilai *random* antara 0 sampai 1. Jika nilai *random* < 0.01 maka gen tersebut yang akan dikenai proses *crossover*. Dari nilai *random* yang didapat, maka diperoleh 3 gen yang akan mengalami proses mutasi, yaitu gen ke 25 individu 1, gen ke 34 individu 3, gen ke 46 individu 4.

Individu 1 gen ke 25

1001010 0011101 1001101 1011110 0101100 1101010

Setelah bermutasi :

1001010 0011101 100110 1010110 0101100 1101010

Individu 3 gen ke 14

0010110 1101001 0101001 1010110 1001010 1001101

Setelah bermutasi :

0010110 1101000 0101001 1010110 1001010 1001101

Individu 4 gen ke 46

1100101 0011010 0110010 0010110 0011001 0011011

Setelah bermutasi :

1100101 0011010 0110010 0010110 0011001 0111011

Didapat 3 individu baru sebagai berikut:

Tabel 3.37 Individu Baru hasil Mutasi Generasi 1

No	Kromosom					
1	1001010	0011101	1001101	1010110	0101100	1101010
2	0010110	1101001	0101001	1010110	1001010	1001101
3	1100101	0011010	0110010	0010110	0011001	0111011

1. **Penghitungan Nilai *Fitness* dan Seleksi menggunakan metode *Rank-based Fitness***

   Mencari nilai bobot

   Untuk mencari nilai bobot (w), sebelumnya digunakan variabel temporer, yaitu variabel x (x1, x2, ..., xn) dengan n adalah jumlah atribut. Kromosom v merupakan representasi dari variabel x yang berbentuk string biner. Kromosom terbagi atas n gen (v1, v2, ..., vn). Sedangkan panjang setiap gen adalah sama. Nilai xi dapat dirumuskan sebagai:

   $xi = a + ((b-a) / (2^{Li}-1))*vi$

   Perhitungan untuk individu 1 :

   INDIVIDU 1

   V1 = 1001010 = 74

   X1 = 0 + ((1 – 0) /(2^7 – 1) * 74) =0,5827

   V2 = 0011101 = 29

   X2 = 0 + ((1 – 0) /(2^7 – 1) * 29) = 0,2283

   V3 = 1001101 = 77

   X3 = 0 + ((1 – 0) /(2^7 – 1) * 77) = 0,6063

   V4 = 1011110 = 94

   X4 = 0 + ((1 – 0) /(2^7 – 1) * 94) = 0,7402

   V5 = 0101100 = 44

   X5 = 0 + ((1 – 0) /(2^7 – 1) * 44) = 0,3465

   V6 = 1101010 = 106

   X6 = 0 + ((1 – 0) /(2^7 – 1) * 106) = 0,8346

   X_TOTAL = 3,3386

W1 = $\frac{X1}{X_{TOTAL}}$ = 0,1745

W2 = $\frac{X2}{X_{TOTAL}}$ = 0,0684

W3 = $\frac{X3}{X_{TOTAL}}$ = 0,1816

W4 = $\frac{X4}{X_{TOTAL}}$ = 0,2217

W5 = $\frac{X5}{X_{TOTAL}}$ = 0,1038

W6 = $\frac{X6}{X_{TOTAL}}$ = 0,25

Max b_j^m = max {b_{1j}, b_{2j}, . . . b_{mj}}

Max b_1 = max {b_{11}, b_{21}, . . . b_{61}} = max {1, 0.375, 0.687, 0.5, 0.187, 0.937} = 1

Max b_2 = max {b_{12}, b_{22}, . . . b_{62}} = max {1, 0.424, 0.796, 0.494, 0.222, 0.909} = 1

Max b_3 = max {b_{13}, b_{23}, . . . b_{63}} = max {1, 0.193, 0.172, 0.89, 0, 0.852} = 1

Max b_4 = max {b_{14}, b_{24}, . . . b_{64}} = max {0.25, 0, 0.091, 0.125, 0, 0} = 0.25

Max b_5 = max {b_{15}, b_{25}, . . . b_{65}} = max {0.75, 0.125, 0.125, 0.625, 0, 1} = 1

Max b_6 = max {b_{16}, b_{26}, . . . b_{66}} = max {0, 0.167, 0, 0, 0, 0 } = 0.167

Perhitungan untuk kalimat 1 dengan 6 fitur :

$$\sum_{j=0}^{n} \left(max\ b_j^m - b_{ij}\right)^2 . w_j^2$$

= (Max b_1 - b_{11})^2. W_1^2 + (Max b_2 – b_{21})^2. W_2^2 + (Max b_3 – b_{31})^2. W_3^2 + (Max b_4 – b_{41})^2. W_4^2 + (Max b_5 – b_{51})^2. W_5^2 + (Max b_6 – b_{61})^2. W_6^2

= (1-1)^2. 0.1745^2 + (1-1)^2. 0.0684^2+ (1-1)^2. 0.1816^2 + (0.25-0.25)^2. 0,2217^2 + (1-0,75)^2. 0,1038^2 + (0-0,167)^2. 0,25^2

= 0 + 0+ 0 + 0 + 0.0007 + 0,0017

= 0,0024

Dilakukan perhitungan yang sama untuk kalimat 2 sampai 6, kemudian hasilnya dijumlahkan sehingga didapatkan nilai $\sum_{i=0}^{m} \sum_{j=0}^{n} (max\ b_j^m - b_{ij})^2 . w_j^2$

Perhitugan nilai *fitness* secara keseluruhan ditunjukkan pada tabel 3.38 berikut :

Tabel 3.38 Penghitungan Nilai Fitness pada Generasi 1 Individu 1

$(max\ b_j^m - b_{ij})^2 . w_j^2$						$\sum_{j=0}^{n} (max\ b_j^m - b_{ij})^2 . w_j^2$
0.0000	0.0000	0.0000	0.0000	0.0007	0.0017	0.0024
0.0119	0.0016	0.0215	0.0031	0.0082	0.0000	0.0462
0.0030	0.0002	0.0226	0.0012	0.0082	0.0017	0.0370
0.0076	0.0012	0.0004	0.0008	0.0015	0.0017	0.0132
0.0201	0.0028	0.0330	0.0031	0.0108	0.0017	0.0715
0.0001	0.0000	0.0007	0.0031	0.0000	0.0017	0.0057
$\sum_{i=0}^{m} \sum_{j=0}^{n} (max\ b_j^m - b_{ij})^2 . w_j^2$						0.1761

$$\text{Fitness} = \frac{1}{\sum_{i=0}^{m}\sum_{j=0}^{n}(max\ b_j^m - b_{ij})^2.w_j^2} = \frac{1}{0.1761} = 5.6772$$

Dari hasil perhitungan, didapat nilai fitness individu 1 yaitu : 5.9169. Dengan cara yang sama hitung nilai fitness seluruh individu dalam populasi. Tabel nilai fitness dari seluruh individu dalam populasi sebagai berikut :

Tabel 3.39 Nilai fitness Seluruh Individu (individu awal, anakan hasil *crossover*, dan anakan hasil mutasi)

Individu	FITNESS
INDIVIDU 1	5.6772
INDIVIDU 2	1.9319
INDIVIDU 3	4.5743
INDIVIDU 4	2.7385
INDIVIDU 5	2.9619
INDIVIDU 6	2.8592
INDIVIDU 7	2.4576
INDIVIDU 8	2.7849
INDIVIDU 9	9.8958
INDIVIDU 10	4.3213
Anakan hasil crossover 1	3.9643
Anakan hasil crossover 2	4.4523
Anakan hasil crossover 3	5.2836
Anakan hasil crossover 4	2.6837
Anakan hasil crossover 5	3.9621
Anakan hasil crossover 6	2.6837
Anakan hasil mutasi 1	5.5223
Anakan hasil mutasi 2	4.5849
Anakan hasil mutasi 3	2.7385

Dipilih 10 individu dengan nilai fitness terbaik untuk dijadikan populasi generasi ke-2 yaitu INDIVIDU 9, INDIVIDU 1, Anakan Hasil Mutasi 1, Anakan hasil crossover 3, Anakan hasil mutasi 2, INDIVIDU 3, Anakan Hasil crossover 2, INDIVIDU 10, Anakan Hasil crossover 1, Anakan Hasil crossover 5. Populasi ini nantinya akan melanjutkan proses genetika dan menghasilkan generasi-generasi berikutnya hingga generasi terakhir (misalkan ditentukan

jumlah generasi =2). Pada generasi terakhir didapat hasil akhir sebagai berikut :

Tabel 3.40 Nilai Fitness Akhir

No	Kromosom	Fitness
1	0011101 0011001 0010101 1100101 0011110 1011100	9.8958
2	1001010 0011101 1001101 1011110 0101001 1101010	5.7231
3	1001010 0011011 1001101 1100101 0101100 1101001	5.6772
4	1001010 0011101 1001101 1010110 0101100 1101010	5.5223
5	1001010 0011101 1001101 1010110 0101100 1101010	5.5223
6	1001010 0011101 1001101 1011110 0101001 1011110	5.4605
7	1101010 0011101 1001101 1010110 0101100 1101010	5.3290
8	1001010 0011101 1001101 1011110 0101001 1010110	5.2836
9	0010110 1101001 0101001 1010110 1001010 1101010	5.1595
10	1001010 0011101 1001101 1010110 0101100 1010110	5.0911

Dari data yang didapat, diketahui bahwa nilai fitness tertinggi terdapat pada individu ke-1 dengan nilai w sebagai berikut:

W1 : 0.0973
W2 : 0.0839
W3 : 0.0705
W4 : 0.3389
W5 : 0.1007
W6 : 0.3087

Proses Perankingan

Untuk mendapatkan urutan ranking, maka dihitung nilai alternatif ke-i, gi, (i=1,2,...,m) dengan rumus sebagai berikut:

$g_i = \sum_{j=1}^{n} w_j . b_{ij}$

Nilai g_i terbesar menunjukkan kalimat terbaik. Jadi, kalimat diurutkan berdasarkan nilai g_i dari yang terbesar. Nilai b_{ij} dapat dilihat pada tabel 3-33.

g1 = (0.0973×1) + (0.0893×1) + (0.0705×1) + (0.3389×0.25) + (0.1007×0.75) + (0.3087×0)
= 0.4119
g2 = 0.1489
g3 = 0.1892
g4 = 0.2581

g5 = 0.0368
g6 = 0.3282
Sehingga urutan kalimat untuk dokumen 1 : Kalimat 1 – Kalimat 6 – Kalimat 4 – Kalimat 3 – Kalimat 2 – Kalimat 5
Kalimat yang menjadi ringkasan dengan rasio 50% pada dokumen 1 adalah : Kalimat 1, Kalimat 4 dan Kalimat 6.

Proses yang sama dilakukan terhadap dokumen 2 dan 3 dengan rasio 50%. Untuk dokumen 2 ada tiga kalimat dengan rasio 50% didapatkan dua kalimat hasil ringkasan. Untuk dokumen 3 ada tempat kalimat dengan rasio 50% didapatkan dua kalimat hasil ringkasan.Hasil ringkasan semua dokumen :

DOKUMEN 1	Kalimat 1 (R1), Kalimat 4 (R2), Kalimat 6 (R3)
DOKUMEN 2	Kalimat 1 (R4), Kalimat 2 (R5)
DOKUMEN 3	Kalimat 1 (R6), Kalimat 2 (R7)

3.3.4 *Merging* Hasil Ringkasan

Proses penggabungan dokumen hasil ringkasan dilakukan menggunakan TF (tanpa perhitungan ISF) dan *cosine similarity*. Kalimat hasil ringkasan dari dokumen yang telah dikelompokkan, dihitung nilai *similarity*nya untuk tiap kalimat. Dilakukan perangkingan kalimat mulai dari nilai *similarity* terbesar sampai terkecil. Proses penghitungan *cosine similarity* menggunakan TF sebagai berikut :

- Menghitung perkalian skalar TF kata yang sama antara kalimat ke- i dengan kalimat lainnya. Kemudian hasil perkalian tersebut dijumlahkan.

  Untuk R1 :

  Tabel 3.41 Perkalian Skalar Antara R1 dengan yang lainnya

(R1)	R1 * Ri					
	R2	R3	R4	R5	R6	R7
card	1	0	1	1	1	0
corp	0	0	0	0	0	0
credit	0	0	0	0	0	0
force	0	0	1	0	1	1
hack	0	0	0	0	0	0
information	0	0	0	0	0	0

network	0	0	0	0	0	0
past	0	0	0	0	0	0
play	0	0	0	0	0	0
playstation	1	1	0	1	0	0
shutdown	1	1	0	1	0	0
sony	1	2	1	0	0	0
stolen	1	1	0	0	0	0
user	0	1	0	0	0	0
warn	0	0	0	0	0	0
week	0	0	0	0	0	0
	5	5	3	3	2	1

Dilakukan penghitungan yang sama untuk R2 sampai R7.

- Menghitung panjang setiap kalimat. Mengkuadratkan nilai TF semua kata pada kalimat lalu menjumlahkan nilai kuadrat tersebut kemudian nilai tersebut diakar.

  Untuk R1 :

  Tabel 3.42 Penghitungan Kuadrat TF untuk R1

kata (k)	**Tf_k**	**$(Tf_k)^2$**
card	1	1
corp	1	1
credit	1	1
force	1	1
hack	1	1
information	1	1
network	1	1
past	1	1
play	1	1
playstation	1	1
shutdown	1	1
sony	1	1
stolen	1	1
user	1	1
warn	1	1
week	1	1
JUMLAH		16

Dilakukan penghitungan yang sama R2 sampai R7.

Tabel 3.43 Penghitungan Akar dari Kuadrat TF Semua Kalimat

R	$\sum_{k=1}^{N} Tf_k{}^2$	$\sqrt{\sum_{k=1}^{N} Tf_k{}^2}$
1	16	4
2	8	2.8284
3	19	4.3589
4	17	4.1231
5	18	4.2426
6	6	2.4495
7	7	2.6458

- Menerapkan rumus *cosine similarity*, kemudian dihitung kesamaan antar kalimat.

R1 dengan R2 = $\frac{5}{4 \text{ x } 2.8284}$ = 0.4419

R1 dengan R3 = $\frac{5}{4 \text{ x } 4.3589}$ = 0.2868

R1 dengan R4 = $\frac{3}{4 \text{ x } 4.1231}$ = 0.1819

R1 dengan R5 = $\frac{3}{4 \text{ x } 4.2426}$ = 0.1768

R1 dengan R6 = $\frac{2}{4 \text{ x } 2.4495}$ = 0.2041

R1 dengan R7 = $\frac{1}{4 \text{ x } 2.6458}$ = 0.0945

Nilai similarity R1 = 0.4419 + 0.2868 + 0.1819 + 0.1768 + 0.2041 + 0.0945
= 1.3860

Tabel 3.44 Nilai *Similarity* Semua Kalimat Ringkasan

Kalimat Ringkasan	SIMILARTY
dokumen 1 – Kalimat 1 (R1)	1.3860
dokumen 1 – Kalimat 4 (R2)	1.4133

dokumen 1 – Kalimat 6 (R3)	1.3014
dokumen 2 – Kalimat 1 (R4)	1.5663
dokumen 2 – Kalimat 2 (R5)	1.2135
dokumen 3 – Kalimat 1 (R6)	1.1466
dokumen 3 – Kalimat 2 (R7)	0.5810

Maka setelah proses perangkingan didapatkan hasil sebagai berikut :

	SIMILARTY
(R4)	1.5663
(R2)	1.4133
(R1)	1.3860
(R3)	1.3013
(R5)	1.2135
(R6)	1.1466
(R7)	0.5810

3.3.5 Hasil Ringkasan Multi Dokumen

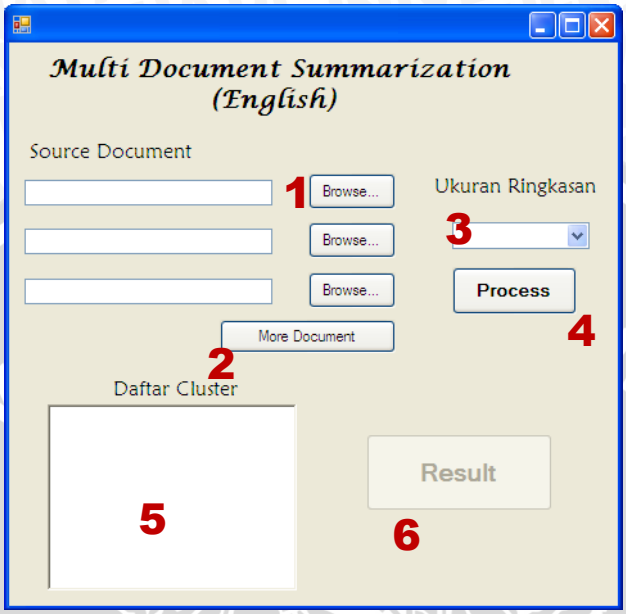
Sony admits a second cyber-hack, this one giving attackers names, addresses, e-mail addresses, birth dates, phone numbers, and other information from 24.6 million PC games customers. Sony noted that it has no direct evidence that credit card information was stolen. Sony Corp. warns that during the hack that forced the shutdown of its PlayStation Network for the past week, users´ credit card information may have been stolen. It says certain users´ names, birth dates, e-mail addresses and log-in information was taken, and purchase history and credit card billing address information may have also been stolen. The attack targeted the Sony Online Entertainment (SOE) PC games network, which includes about 12,700 credit or debit card numbers and expiration dates from people outside the US. A news report claims that Sony might soon be the victim of another hackers´ attack. An IRC user said that he witnessed the hackers planning another attack for the weekend.

3.4 Rancangan Antarmuka

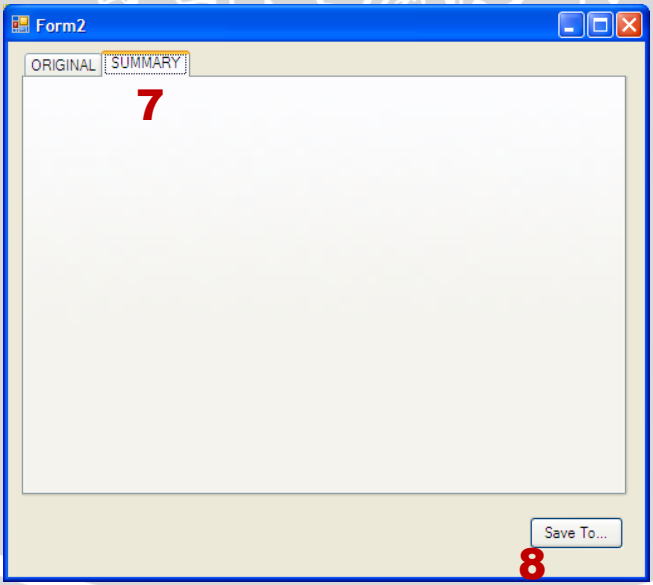
Rancangan *interface* atau antarmuka pada sistem ini terdiri dari dua form. Kedua form saling berurutan dan berhubungan. Form kedua menampilkan hasil dari proses sebelumnya di form pertama.

Pada kedua form ini terdapat tujuh bagian utama sesuai fungsi dengan nomor yang menunjukkan urutannya. Pada form pertama terdapat bagian satu sampai lima dan pada form kedua terdapat bagian enam sampai tujuh. Fungsi ketujuh bagian tersebut sebagai berikut :

- No. 1 adalah tombol "Browse", menyediakan fungsi bagi *user* untuk melakukan *load* dokumen dari direktori komputer.
- No.2 adalah tombol "More Document" digunakan untuk melakukan *load* lebih dari tiga dokumen dan maksimal 10 dokumen.
- No.3 adalah input untuk memilih prosentase ukuran ringkasan pada sistem.
- No. 4 adalah tombol "Process" yang digunakan untuk melakukan proses peringkasan dokumen sesuai dengan dokumen yang telah diinputkan *user* dan prosentase ukuran peringkasan yang telah dipilih oleh *user.*
- No. 5 adalah area untuk menampilkan proses *clustering* dokumen yang diinputkan *user*.
- No. 6 adalah tombol "Result" yang digunakan untuk menampilkan form yang berisi hasil peringkasan dokumen. Tombol ini hanya dapat ditekan saat *user* telah menekan tombol "Process".
- No. 7 adalah area untuk menampilkan hasil ringkasan dan dokumen asli yang telah diinputkan oleh *user*.
- No. 8 adalah tombol "Save To" yang digunakan untuk menyimpan hasil ringkasan dokumen. Tombol ini hanya ada jika memilih menu tab "summary".



Gambar 3.20 Gambar Form 1 pada *Interface*



Gambar 3.21 Gambar Form 2 pada *Interface*

3.5 Rancangan Uji Coba

Pada rancangan uji coba, sistem akan dievaluasi menggunakan tiga parameter yaitu *precision, recall,* dan *F-measure*. Tabel evaluasi ditunjukkan pada tabel 3.45 berikut:

Tabel 3.45 Tabel evaluasi sistem

No. Cluster	Ukuran Ringkasan	P	R	F
CLUSTER 1	25 %			
	50 %			
	75%			
CLUSTER 2	25 %			
	50 %			
	75 %			

BAB IV
IMPLEMENTASI DAN PEMBAHASAN

Untuk melakukan implementasi sistem perlu dipersiapkan lingkungan implementasi untuk memenuhi kebutuhan program dalam mengimplementasikan sistem.

4.1 Perangkat Sistem

Lingkungan implementasi meliputi lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*). Program akan membutuhkan inputan berupa dokumen berbahasa Inggris berbentuk file.txt untuk diringkas.

4.1.1 Perangkat Keras

Perangkat keras yang digunakan untuk mengimplementasikan sistem peringkas otomatis multi dokumen berbahasa Inggris ini adalah :

1. Processor AMD Athlon II Neo K345 Dual-Core Processor 1.40 Ghz.
2. Memory 2 GB.
3. Harddisk 320 GB.
4. VGA 256 MB ATI Mobility Radeon HD 4225.
5. Monitor 13.3".
6. Keyboard

4.1.2 Perangkat Lunak

Perangkat lunak yang diperlukan dalam pengembangan sistem peringkas otomatis untuk multi dokumen berbahasa Inggris adalah :

1. Sistem operasi Windows 7 Ultimate 32-bit
2. Microsoft Visual Studio C# 2008
3. Text Editor Notepad

4.2 Implementasi Program

Berdasarkan perancangan perangkat lunak yang telah diuraikan pada bab 3, maka akan dibahas mengenai implementasi program seperti yang telah dirancang, yaitu sebagai berikut :

4.2.1 Struktur Data

Struktur data untuk menerapkan sistem ini berupa pembagian kelas-kelas utama yang menerapkan tiap proses pada sistem. Kelas-kelas tersebut sebagai berikut:

1. Kelas Preprocessing

Sebelum dokumen diproses dalam *clustering* dan peringkasan, dokumen tersebut dipersiapkan dahulu melalui tahap *preprocessing*. Kelas *preprocessing* digunakan untuk melakukan proses-proses yang ada pada tahap *preprocessing*. Proses-proses tersebut adalah *tokenizing* dan *case folding*, *filtering* juga *stemming*. Namun *stemming* ada pada kelas tersendiri, sehingga pada kelas *preprocessing* tinggal memanggil fungsi pada kelas tersebut.

Dalam kelas ini terdapat beberapa fungsi. Fungsi-fungsi tersebut adalah fungsi `Hitung()` yang merupakan fungsi utama pada kelas ini. Fungsi ini memiliki parameter input `object[]` berupa kumpulan dokumen atau kumpulan kalimat. Selain itu, terdapat fungsi `RemoveSymbols()` dengan parameter input `string` berupa dokumen atau kalimat dan fungsi `IsStopWord()` dengan parameter input `string` berupa kata-kata dalam dokumen atau kalimat.

2. Kelas PorterStemmer

Kelas PorterStemmer digunakan untuk melakukan proses stemming pada dokumen berbahasa Inggris menggunakan aturan Porter. Dengan *stemming* akan didapatkan bentuk dasar dari kata-kata dalam dokumen. Kelas PorterStemmer dibuat oleh Martin Porter yang diunduh pada halaman berikut:
http://www.tartarus.org/~martin/PorterStemmer.
Fungsi-fungsi dalam kelas ini digunakan untuk membuang imbuhan dan menjadikan bentuk asal kata-kata dalam bahasa Inggris. Fungsi `stemTerm()` merupakan fungsi utama pada kelas ini dengan parameter input `string`. Fungsi ini yang akan dipanggil saat proses stemming dengan parameter input berupa kata-kata dalam dokumen atau kalimat.

3. Kelas Clustering

Kelas clustering merepresentasikan single pass clustering untuk mengelompokkan dokumen ke dalam kelompok-kelompok tertentu. Pembagian ini dilakukan sesuai dengan kesamaan (similarity) dari

dokumen-dokumen tersebut berdasarkan kontennya dengan batas kesamaan (threshold similarity) yang telah ditentukan.
Fungsi-fungsi yang ada pada kelas ini yaitu fungsi `Hitung()` dengan parameter input `object[]` berupa kumpulan dokumen hasil preprocessing dan fungsi `HitungSimilarity()` dengan parameter input `SortedDictionary` berupa dokumen-dokumen. Fungsi `Hitung()` merupakan fungsi utama pada kelas ini yang digunakan untuk melakukan pengelompokan dokumen berdasarkan konten.

4. Kelas Summary

Kelas summary digunakan untuk merepresentasikan proses peringkasan dokumen. Kelas ini akan menggunakan fungsi pada kelas lainnya yaitu fungsi pada kelas ekstraksi dan kelas genetika. Proses ektraksi dan genetika dilakukan untuk tiap dokumen. Setelah diketahui bobot dari kalimat-kalimat tiap dokumen maka kalimat akan diurutkan berdasarkan nilai bobot dari yang terbesar untuk kemudian diambil sesuai persentase ringkasan. Hasilnya akan digabungkan dengan ringkasan dokumen-dokumen lain yang ada dalam satu *cluster* melalui proses *merging*.

Fungsi-fungsi pada kelas ini adalah fungsi `Hitung()` dengan parameter input `object[][]` berupa kumpulan dokumen yang telah dikelompokkan, fungsi `PecahTiapKalimat()` dengan parameter input `object` berupa dokumen, fungsi `Merging()` dengan parameter input hasil ringkasan tiap *cluster* bertipe `Arraylist`. Terakhir yaitu fungsi `PerkalianSkalar()` dengan parameter input `SortedDictionary` berupa dua dokumen.

5. Kelas Ekstraksi

Pada kelas ekstraksi dilakukan proses-proses ekstraksi kalimat dengan enam fitur. Dari fitur-fitur tersebut akan dihasilkan nilai-nilai untuk tiap kalimat. Nilai-nilai ini nantinya akan diproses pada kelas Genetika untuk memberi bobot agar tiap kalimat memiliki nilai tunggal dengan memperhatikan nilai-nilai kalimat lain dalam satu dokumen agar dapat diurutkan.
Fungsi-fungsi pada kelas ini yaitu fungsi `ektraksi()` dengan parameter input kumpulan kalimat bertipe `object[]` dan dokumen bertipe `object`, fungsi `fitur1()` dengan parameter input `object[]` berupa kumpulan kalimat, fungsi `fitur2()` dengan

parameter input `object[]` berupa kumpulan kalimat, fungsi `fitur3()` dengan parameter input `object[]` berupa kumpulan kalimat, fungsi `PerkalianSkalar()` dengan parameter input `Sortedictionary`, fungsi `All_Upper()` dengan parameter input kata bertipe `string`, fungsi `fitur4()` dengan parameter input `integer`, `object[]`, dan `object`, fungsi `CekProper()` dengan parameter input kalimat bertipe `SortedDictionary` dan kumpulan lata *proper* bertipe `Arraylist`, fungsi `fitur5()` dengan parameter input `integer`, `object[]`, dan `object`, fungsi `CekTematik()` dengan parameter input kalimat bertipe `SortedDictionary` dan kumpulan kata tematik bertipe `Arraylist`, fungsi `AllNumerik()` dengan parameter input `string` berupa kata dalam kalimat, fungsi `fitur6()` dengan parameter input `integer` dan `object[]`.

6. Kelas Genetika

Kelas genetika merepresentasikan algoritma genetika yang dapat digunakan untuk pembobotan nilai-nilai yang dimiliki kalimat sekaligus mengoptimasinya. Fungsi-fungsi pada kelas Genetika yaitu fungsi `Hitung()` yang merupakan fungsi utama dengan parameter input `double[,]` berupa matrik hasil ekstraksi kalimat, fungsi `HitungFitness()` dengan parameter input `string` berupa individu dan `double[,]` berupa matrik hasil ekstraksi kalimat, fungsi `HitungW()` dengan parameter input `string` berupa individu, fungsi `BinerToDesimal()` dengan parameter input `string` berupa individu.

4.2.2 Proses Preprocessing

Input tahap preprocessing ini adalah sekumpulan dokumen yang terdiri atas paragraf, kalimat dan kata dalam bentuk file teks berekstensi .txt.

4.2.2.1 Proses Tokenizing dan Case Folding

Proses Tokenizing dan case folding ada dalam kelas preprocessing. Proses case folding berfungsi untuk menghapus simbol dan mengubah semua huruf menjadi huruf kecil.

Berikut ini adalah proses case folding yang dilakukan dengan memanggil fungsi `RemoveSymbol()` untuk menghilangkan simbol dan mengubah semua huruf menjadi huruf kecil :

```
1.  String temp =
    RemoveSymbols(dokumen_input1[x].ToString()).ToLower
    ();
```

***Sourcecode* 4.1** Proses *Case Folding*

Berikut ini adalah fungsi untuk menghilangkan simbol :

```
1.  public static string RemoveSymbols(string asli)
2.  {
3.      return Regex.Replace(asli, @"[^A-Za-z0-9\s]",
    "");
4.  }
```

***Sourcecode* 4.2** Fungsi Menghilangkan Simbol

Proses tokenizing merupakan proses untuk memecah kalimat menjadi kata-kata penyusunnya.

```
1.  public static string[] Tokenizing(string temp)
2.  {
3.  char[] tandaSplit = new char[] { ' ', '\r', '\n',
    '\t' };
4.  string[] kumpulanKata = temp.Split
    (tandaSplit,StringSplitOptions.RemoveEmptyEntries
    );
5.     return kumpulanKata;
6.  }
```

***Sourcecode* 4.3** Proses *Tokenizing*

4.2.2.2 Proses Filtering dan Stemming

Proses *Filtering* dan *Stemming* ada di dalam kelas preprocessing. Proses *filtering* atau juga disebut *stopword removal* merupakan proses untuk menghilangkan kata-kata yang ada dalam daftar *stopword*. Fungsi ini akan memanggil fungsi `IsStopWord()`. Jika tidak ada dalam daftar *stopword* maka kata tersebut akan dilakukan *stemming*.

```
1.  public static SortedDictionary<string, int>
    FilterStem(string[] kumpulanKata)
2.  {
3.  SortedDictionary<string, int> tampung = new
    SortedDictionary<string, int>();
4.  // perulangan untuk tiap kata hasil tokenizing
5.   foreach (string kata in kumpulanKata)
6.   {
7.      string hasilStem;
8.      // jika stop word atau lanjut
9.      if (IsStopWord(kata) == true)
```

```
10.         continue;
11.         Else
12.         {
13.     PorterStemmer stemmer = new PorterStemmer();
14.             hasilStem = stemmer.stemTerm(kata);
15.         }
16.         if (tampung.ContainsKey(hasilStem))
17.         {
18.     tampung[hasilStem]=tampung[hasilStem]+1;
19.         }
20.         Else
21.         {
22.             tampung.Add(hasilStem, 1);
23.         }
24.       }
25.       return tampung;
26.     }
```

***Sourcecode* 4.4** Proses *Filtering* dan *Stemming*

Fungsi `IsStopWord()` digunakan untuk memeriksa apakah kata tersebut ada di dalam daftar stopword.

```
1.  public static bool IsStopWord(string kata)
2.  {
3.  // ada di daftar stop word atau tidak
4.  string[] stopword =
    File.ReadAllLines(".\\stopword.txt");
5.    bool stop = false;
6.    foreach (string word in stopword)
7.    {
8.        if (kata == word)
9.        {
10.           stop = true; break;
11.       }
12.   }
13.   return stop;
14. }
```

***Sourcecode* 4.5** Fungsi Pengecekan *Stopword*

4.2.3 Proses *Clustering*

Clustering yang digunakan adalah *single pass clustering* dengan *cosine similarity* untuk menghitung kesamaan dokumen. *Cosine similarity* dihitung menggunakan TF.IDF ternormalisasi untuk memberi nilai pada dokumem. Batas *similarity* yang digunakan adalah 0.085. Misalnya pada *cluster* 1 terdapat 3 dokumen

a, b, dan c. Semua dokumen akan dibandingkan dengan dokumen x yang baru masuk. Dokumen a memiliki nilai *similarity* terbesar dengan x dibanding dokumen b dan c, maka dokumen a yang mewakili *cluster* 1. Jika nilai *similarity* antara dokumen a dan x lebih dari 0.085 maka dokumen x masuk dalam *cluster* dokumen a. Jika kurang dari 0.085 maka dokumen x membentuk *cluster* baru.

Awalnya dilakukan penghitungan TF.IDF ternormalisasi. Parameter inputnya adalah `SortedDictionary<string, int>` dari hasil *preprocessing* yaitu berupa kata-kata unik yang telah melalui tahap *preprocessing* beserta frekuensinya. Parameter outputnya adalah `object[]` yang berfungsi untuk menampung nilai TF.IDF ternormalisasi tiap dokumen.

1. Perhitungan TF.IDF ternormalisasi

Sebelum menghitung *similarity* antar dokumen, masing-masing dokumen harus memiliki nilai untuk perhitungan *similarity*. Nilai ini dihitung menggunakan TF.IDF ternormalisasi.

```
1.  object[] TF_IDF = new object[hasil_pre.Count()];
2.  object[] TF_IDF_Norm = new
    object[hasil_pre.Count()];
3.  N = hasil_pre.Count();
4.  S = new double[hasil_pre.Count()];
5.  int a = 0;
6.  foreach (SortedDictionary<string, int> dok in
    hasil_pre)
7.  {
8.  hasil_pre_TFIDF = new SortedDictionary<string,
    double>();
9.     double nilai_S = 0;
10.    foreach (var kata in dok)
11.    {
12.       double Df = 0;
13. foreach (SortedDictionary<string, int> row1 in
    hasil_pre)
14.       {
15.          if (row1.ContainsKey(kata.Key)) Df++;
16.       }
17.       double x = N / Df;
18.       double IDF = Math.Log10(x)+1; // di tambahkan
    nilai 1 agar nilai tidak 0;
19.       double total = kata.Value * IDF; // hitung
    TFIDF
20. hasil_pre_TFIDF.Add(kata.Key, total);
21.       double TFIDF_kuadrat = total * total;
```

```
22.         nilai_S = nilai_S + TFIDF_kuadrat;
23.     }
24.     TF_IDF[a] = hasil_pre_TFIDF;
25.     nilai_S = Math.Sqrt(nilai_S);
26.     S[a] = nilai_S;
27.     a++;
28. }
29. // TF.IDF ternormalisasi
30. a = 0;
31. foreach (SortedDictionary<string, double> dok in
    TF_IDF)
32. {
33. TFIDF_normalisasi = new SortedDictionary<string,
    double>();
34.     foreach (var kata in dok)
35.     {
36.         double normalisasi = kata.Value / S[a];
37. TFIDF_normalisasi.Add(kata.Key, normalisasi);
38.     }
39.     TF_IDF_Norm[a] = TFIDF_normalisasi;
40.     a++;
41. }
```

***Sourcecode* 4.6** Proses Penghitungan TF-IDF ternormalisasi

2. Perhitungan *Similarity*

Pada proses clustering menggunakan dua `object[,]` yaitu `cluster1` dan `cluster`. `object[,] cluster` digunakan untuk menampung `SortedDictionary` yang berisi kata dan frekuensi pada suatu dokumen. `object[,] cluster1` berisi informasi letak dokumen dalam suatu *cluster*.

```
1.  // similarity untuk clustering
2.  int[,] cluster1 = new int[10,10];
3.  object [,] cluster = new object[10,10];
4.  double threshold = 0.085;
5.  int urutan = 1;
6.  int cluster_ke = 0;
7.  foreach (SortedDictionary<string, double> dok in
    TF_IDF_Norm)
8.  {
9.      double max_1 = 0; double max = 0;
10.     int banyak_dok = 0;
11.     int cluster_baru = 0;
12.
13.     // set dokumen pertama menjadi cluster pertama
14.     if (cluster[0,0] == null)
```

```
15.     {
16.         cluster[0,0] = dok;
17.         cluster1[0,0] = urutan;
18.     }
19.     else
20.     {
21. for (int baris = 0; baris < cluster.GetLength(0);
    baris++ )
22.         {
23.             if (cluster[baris,0] == null)
24.             {
25.                 cluster_baru=baris; break;
26.             }
27.             int j = 0;
28.  for (int kolom = 0; kolom < cluster.GetLength(1);
    kolom++)
29.             {
30. if (cluster[baris, kolom] == null)
31.                 {
32.                     j = kolom; break;
33.                 }
34.  SortedDictionary<string, double> c =
    (SortedDictionary<string, double>)cluster[baris,
    kolom];
35. double nilaiSim = HitungSimilarity(dok, c); //
    hitung nilai similarity antar dokumen
36.
37.                 if (nilaiSim > max)
38.                 {
39.                     max = nilaiSim;
40.                 }
41.             }
42.             if (max > max_1)
43.             {
44.                 cluster_ke = baris;
45.                 banyak_dok = j;
46.                 max_1 = max;
47.                 max = 0;
48.             }
49.         }
50.
51.         if (max_1 > threshold)
52.         {
53.          cluster[cluster_ke,banyak_dok] = dok;
54. cluster1[cluster_ke,banyak_dok] = urutan;
55.         }
56.         else
```

```
57.             {
58.                 cluster[cluster_baru,0] = dok;
59.                 cluster1[cluster_baru,0] = urutan;
60.             }
61.         }
62.         urutan++;
63.     }
64.     return cluster1;
65. }
```

***Sourcecode* 4.7** Proses *Clustering*

3. Proses Menghitung *Similarity*

Untuk menghitung *similarity* digunakan `SortedDictionary(key, value)` x1 dan x2 yang mewakili dua dokumen berbeda yang akan dihitung kesamaannya (*similarity*). Pada fungsi ini `key` merupakan kata dalam suatu dokumen dan `value` merupakan frekuensi kata tersebut. Nilai *similarity* dihitung dengan mengalikan nilai `value` x1 dan x2 pada `key` sama. Hasil perkalian untuk tiap `key` yang sama ditampung oleh variabel `nilaiSim` yang bertipe data `double`.

```
1.  private static double
    HitungSimilarity(SortedDictionary<string, double>
    x1,SortedDictionary<string, double> x2)
2.  {
3.      double nilaiSim = 0;
4.      foreach (var kata1 in x1)
5.      {
6.          foreach (var kata2 in x2)
7.          {
8.              if (kata1.Key == kata2.Key)
9.              {
10. nilaiSim = nilaiSim + (kata1.Value * kata2.Value);
11.             }
12.         }
13.     }
14.     return nilaiSim;
15. }
```

***Sourcecode* 4.8** Fungsi Menghitung *Similarity*

4.2.4 Proses Peringkasan Dokumen

4.2.4.1 Proses Ekstraksi Kalimat

1. Fitur 1 (F1)

Pada fungsi fitur 1 variabel `double[] nilai` digunakan untuk menampung nilai fitur 1 semua kalimat. Untuk mengetahui jumlah kata pada tiap kalimat dilakukan dengan menjumlahkan nilai value pada `SortedDictionary<string, int>` masing-masing kalimat. Untuk nilai tiap kalimat pada fitur 1 dihitung dengan membagi jumlah kata dengan nilai terbesar pada jumlah kata semua kalimat.

```
1.  public static double fitur1(int s, object[]
    kumpulan_kalimat)
2.  {
3.  double[] nilai = new
    double[kumpulan_kalimat.Length];
4.      int x = 0; int max = 0;
5.
6.  foreach (SortedDictionary<string, int> kalimat in
    kumpulan_kalimat)
7.     {
8.         int panjang_kalimat = 0;
9.         foreach (var kata in kalimat)
10.        {
11. panjang_kalimat = panjang_kalimat + kata.Value;
12.        }
13.        nilai[x] = panjang_kalimat;
14.        if (panjang_kalimat > max)
15.        {
16.             max = panjang_kalimat;
17.        }
18.        x++;
19.     }
20.     //skoring
21.     return Math.Round(nilai[s]/ max, 3);
22.  }
```

***Sourcecode* 4.9** Proses Penghitungan Nilai Fitur 1 (Panjang Kalimat)

2. Fitur 2 (F2)

Fungsi Fitur 2 menjelaskan tentang perhitungan TF.ISF untuk masing-masing kalimat. Variabel `double TF_ISF` digunakan untuk menghitung nilai TF.ISF tiap kata pada kalimat. Nilai kalimat pada fitur 2 dihitung dengan membagi nilai TF.ISF tiap kalimat dengan

nilai TF.ISF terbesar pada semua kalimat. Hasil nilai fitur 2 semua kalimat ditampung pada variabel `double[] O`.

```
1.  public static double fitur2(int s, object[]
    kumpulan_kalimat)
2.  {
3.  double[] O = new double[kumpulan_kalimat.Length];
4.  int k = 0; double max = 0; int N =
    kumpulan_kalimat.Length;
5.  foreach (SortedDictionary<string, int> kalimat in
    kumpulan_kalimat)
6.      {
7.          double nilai_O = 0;
8.          foreach (var kata in kalimat)
9.          {
10.             int nK = 0;
11. foreach (SortedDictionary<string, int> kal in
    kumpulan_kalimat)
12.             {
13.                 if (kal.ContainsKey(kata.Key))
14. nK++;
15.             }
16. double TF_ISF = kata.Value * Math.Log10(N / nK) +
    1;
17.             nilai_O = nilai_O + TF_ISF;
18.         }
19.         O[k] = nilai_O;
20.         k++;
21.         if (nilai_O > max)
22.          {
23.              max = nilai_O;
24.          }
25.      }
26.      //skoring
27.      return Math.Round(O[s] / max,3);
28.  }
```

***Sourcecode* 4.10** Proses Penghitungan Nilai Fitur 2 (Pembobotan Kata)

3. Fitur 3 (F3)

Fungsi Fitur 3 menjelaskan tentang perhitungan *cosine similarity* pada masing-masing kalimat. Nilai *cosine similarity* dihitung menggunakan nilai TF.ISF. Perhitungan *cosine similarity* dapat dilihat pada *source code* berikut:

```
1.  public static double fitur3(int s, object[]
    kumpulan_kalimat)
```

```
{
object[] nilai_TFISF = new
object[kumpulan_kalimat.Length];
double[] O = new double[kumpulan_kalimat.Length];
double[,] perkalian_skalar = new
double[kumpulan_kalimat.Length,
kumpulan_kalimat.Length];
double[] nilai_similarity = new
double[kumpulan_kalimat.Length];
int k = 0; int N = kumpulan_kalimat.Length;

    // Menghitung panjang setiap kalimat.
foreach (SortedDictionary<string, int> kalimat in
kumpulan_kalimat)
    {
SortedDictionary<string, double> hasil_TFISF = new
SortedDictionary<string, double>();
       double nilai_O = 0;
       foreach (var kata in kalimat)
       {
          int nK = 0;
foreach (SortedDictionary<string, int> row1 in
kumpulan_kalimat)
          {
              if (row1.ContainsKey(kata.Key))
nK++;
          }
double TF_ISF = kata.Value * (Math.Log10(N / nK)) +
1;
          hasil_TFISF.Add(kata.Key, TF_ISF);

double TF_ISF_kuadrat = TF_ISF * TF_ISF;
          nilai_O = nilai_O + TF_ISF_kuadrat;
       }
      nilai_TFISF[k] = hasil_TFISF;
      nilai_O = Math.Sqrt(nilai_O);
      O[k] = nilai_O;
      k++;
    }

//Menghitung perkalian skalar antar kalimat ke-i
dengan kalimat lainnya.
    //Hasil perkalian tersebut dijumlahkan.
     int row = 0; int col = 0;
foreach (SortedDictionary<string, double> kalimat_i
in nilai_TFISF)
     {
```

```
39.             col = 0;
40.  foreach (SortedDictionary<string, double>
     kalimat_lainnya in nilai_TFISF)
41.             {
42.  double hasil_skalar = PerkalianSkalar(kalimat_i,
     kalimat_lainnya);
43.  perkalian_skalar[row, col] = hasil_skalar;
44.                 col++;
45.             }
46.             row++;
47.         }
48.
49.     // Menerapkan rumus cosine similarity
50.      double max = 0;
51.  for (int aa = 0; aa <
     perkalian_skalar.GetLength(0); aa++)
52.         {
53.             double sim = 0;
54.  for (int b = 0; b < perkalian_skalar.GetLength(1);
     b++)
55.             {
56.                 if (aa == b) continue;
57.  sim = sim + (perkalian_skalar[aa, b] / (O[aa] *
     O[b]));
58.             }
59.             nilai_similarity[aa] = sim;
60.             if (sim > max)
61.             {
62.                 max = sim;
63.             }
64.         }
65.         // skoring
66.  return Math.Round(nilai_similarity[s] / max,3);
67.  }
```

***Sourcecode* 4.11** Proses Penghitungan Nilai Fitur 3 (*Similarity* Kalimat)

Fungsi `PerkalianSkalar()` digunakan untuk menghitung nilai skalar antar 2 kalimat. Input parameter berupa 2 `SortedDictionary<string, double>` yang mewakili kalimat 1 dan 2. Nilai skalar dihitung dengan mengalikan nilai `value` (frekuensi) pada tiap `key` (kata) yang sama antara 2 kalimat.

```
1.  public static double
    PerkalianSkalar(SortedDictionary<string, double>
    kalimat_i, SortedDictionary<string, double>
```

```
    kalimat_lainnya)
2.  {
3.    double hasil_Skalar = 0;
4.    foreach (var kata in kalimat_i)
5.    {
6.      foreach (var kata1 in kalimat_lainnya)
7.      {
8.        if (kata.Key == kata1.Key)
9.        {
10.  hasil_Skalar = hasil_Skalar + (kata.Value *
     kata1.Value);
11.       }
12.     }
13.   }
14.   return hasil_Skalar;
15. }
```

***Sourcecode* 4.12** Fungsi Penghitungan Perkalian Skalar

4. Fitur 4 (F4)

Fungsi fitur 4 digunakan menghitung nilai kalimat berdasarkan *proper noun* tiap kalimat. `ArrayList proper` digunakan untuk menampung kata yang termasuk *proper noun*. Untuk mengetahui kata yang termasuk *proper noun* digunakan fungsi All_Upper(). Hasil fungsi fitur 4 di tampung pada variabel `double[] nilai_proper`.

```
1.  public static double fitur4(int s, object []
    kumpulan_kalimat, object k)
2.  {
3.     ArrayList CekProper = new ArrayList();
4.  double[] nilai_proper = new
    double[kumpulan_kalimat.Length];
5.
6.  // ubah huruf setelah tanda titik menjadi huruf
    kecil agar tidak dikenali sebagai kata proper
7.      k = UbahHurufSetelahTitik(k);
8.  string temp =
    prepocessing.RemoveSymbols(k.ToString());
9.     // tokenizing
10. string[] kumpulanKata =
    prepocessing.Tokenizing(temp);
11.     foreach (string kata in kumpulanKata)
12.     {
```

```
13.             if
    (prepocessing.IsStopWord(kata.ToLower()))
14.             continue;
15.         else
16.         {
17.             if (All_Upper(kata[0].ToString()))
18.             {
19.  PorterStemmer stem = new PorterStemmer();
20.  string hasilStem = stem.stemTerm(kata.ToLower());
21.
22.  if (CekProper.Contains(hasilStem))
23.                 continue;
24.             else
25.             {
26.                 CekProper.Add(hasilStem);
27.             }
28.         }
29.     }
30.   }
31.   int x = 0;
32.
33.  foreach (SortedDictionary<string, int> kalimat in
     kumpulan_kalimat)
34.     {
35.         double jumlah_kata = 0;
     double IsProper = 0;
36.         foreach (var kata in kalimat)
37.         {
38.  jumlah_kata = jumlah_kata + kata.Value;
39.             if (CekProper.Contains(kata.Key))
40.             {
41.             IsProper = IsProper + kata.Value;
42.             }
43.         }
44.  nilai_proper[x] = IsProper / jumlah_kata;
45.         x++;
46.     }
47.     return Math.Round(nilai_proper[s], 3);
48.  }
```

***Sourcecode* 4.13** Proses Penghitungan Nilai Fitur 4 *(Proper Noun)*

Fungsi `UbahHurufSetelahTitik()` digunakan untuk mengubah huruf setelah tanda titik menjadi huruf kecil. Fungsi ini digunakan agar kata setelah tanda titik tidak dikenali sebagai kata *proper*. Input parameter fungsi ini berupa `object k`.

```
1.  public static string
    UbahHurufSetelahTitik(object k)
2.  {
3.  string x = Regex.Replace(k.ToString(),
    "[\n\r\t]", " ");
4.    char[] a = x.ToCharArray();
5.    for (int z = 0; z < a.Length-2; z++)
6.    {
7.       if (a[z] == '.')
8.       {
9.  a[z + 2] =
    char.Parse(a[z+2].ToString().ToLower());
10.      }
11.    }
12.    string hasil = new string(a);
13.    return hasil;
14. }
```

***Sourcecode* 4.14** Fungsi UbahHurufSetelahTitik

Fungsi `All_Upper()` digunakan untuk mengetahui kata input termasuk *proper noun* atau bukan. Input parameter berupa huruf depan pada suatu kata. Jika string input berupa huruf besar maka output bernilai `true` yang berarti kata tersebut termasuk *proper noun*.

```
1.  public static bool All_Upper(string inputString)
2.  {
3.     Regex All_Word_Regex = new Regex("[A-Z]");
4.     if (All_Word_Regex.IsMatch(inputString))
5.     {
6.        return true;
7.     }
8.     return false;
9.  }
```

***Sourcecode* 4.15** Fungsi Pengecekan kata yang termasuk *Proper Noun*

5. Fitur 5 (F5)

Fungsi fitur 5 digunakan menghitung nilai kalimat berdasarkan kata tematik tiap kalimat. `ArrayList Tematik` digunakan untuk menampung kata yang termasuk tematik. Untuk mengetahui kata yang termasuk tematik dengan melihat frekuensi kata tersebut. Jika frekuensi kata lebih dari 1 maka kata tersebut termasuk tematik.

Hasil fungsi fitur 5 di tampung pada variabel `double[] nilai_tematik`.

```
1.  public static double fitur5(int s, object[] kumpulan_kalimat, object k)
2.  {
3.      ArrayList CekTematik = new ArrayList();
4.  double[] nilai_tematik = new double[kumpulan_kalimat.Length];
5.      // case folding
6.  string temp =prepocessing.RemoveSymbols(k.ToString()).ToLower();
7.
8.      // tokenizing
9.  string[] kumpulanKata = prepocessing.Tokenizing(temp);
10.
11. SortedDictionary<string, int> frekuensi_kata = prepocessing.FilterStem(kumpulanKata);
12.
13.     foreach (var kata in frekuensi_kata)
14.     {
15.         if (kata.Value > 1)
16.         {
17.             CekTematik.Add(kata.Key);
18.         }
19.     }
20.
21.     // cek kata yang termasuk tematik word
22.     double max = 0; int x = 0;
23. foreach (SortedDictionary<string, int> kalimat in kumpulan_kalimat)
24.      {
25.          double IsTematik =0;
26.          foreach (var kata in kalimat)
27.          {
28.              if(CekTematik.Contains(kata.Key))
29.              {
30. IsTematik = IsTematik + kata.Value;
31.              }
32.          }
33.          nilai_tematik[x] = IsTematik;
34.          if (IsTematik > max)
35.          {
36.              max = IsTematik;
37.          }
38.              x++;
```

```
39.         }
40.         //skoring
41.  return Math.Round(nilai_tematik[s] / max,3);
42.   }
```

***Sourcecode* 4.16** Proses Penghitungan Nilai Fitur 5 *(Thematic Word)*

6. Fitur 6 (F6)

Fungsi `fitur6()` digunakan untuk menghitung nilai kalimat berdasarkan data numerik pada kalimat tersebut. Untuk mengetahui suatu kata termasuk kata numerik digunakan fungsi `AllNumeric()`. Nilai tiap kalimat dihitung dengan membagi jumlah data numerik dengan jumlah kata pada suatu kalimat. Hasil nilai semua kalimat ditampung pada variabel `double[] nilai_numerik`.

```
1.  public static double fitur6(int s, object[]
    kumpulan_kalimat)
2.  {
3.  double[] nilai_numerik = new
    double[kumpulan_kalimat.Length];
4.
5.      int x = 0;
6.  foreach (SortedDictionary<string, int> kalimat in
    kumpulan_kalimat)
7.      {
8.          double jumlah_kata = 0;
    double IsNumerik = 0;
9.          foreach (var kata in kalimat)
10.         {
11. jumlah_kata = jumlah_kata + kata.Value;
12.
13.             if (AllNumeric(kata.Key))
14.             {
15.           IsNumerik = IsNumerik + kata.Value;
16.             }
17.         }
18.
19. nilai_numerik[x] = IsNumerik / jumlah_kata;
20.         x++;
21.     }
22.     return Math.Round(nilai_numerik[s], 3);
23. }
```

***Sourcecode* 4.17** Proses Penghitungan Nilai Fitur 6 *(Numeric Word)*

Fungsi `AllNumeric()` digunakan untuk mengetahui kata input termasuk data numerik atau bukan. Jika string input berupa angka antara 0 - 9 maka output bernilai `true` yang berarti kata tersebut termasuk data numerik.

```
1.  public static bool AllNumeric(string kata_key)
2.  {
3.  Regex All_Numeric_Regex = new Regex("[0-9]");
4.      if (All_Numeric_Regex.IsMatch(kata_key))
5.      {
6.              return true;
7.      }
8.      return false;
9.  }
```

***Sourcecode* 4.18** Fungsi Pengecekan kata numerik

4.2.4.2 Proses Perangkingan dengan Algoritma Genetika

1. Pembangkitan Populasi Awal

Populasi awal ditampung pada `string[] populasi` sejumlah populasi yang dibentuk. Pembentukan populasi dilakukan secara random antara 0-1. Untuk membangkitkan bilangan random digunakan kelas `Random()`.

```
1.  Random data = new Random(5);
2.
3.  // bangkitkan populasi awal
4.  string[] populasi = new string[jumlah_individu];
5.  for (int i = 0; i < jumlah_individu; i++)
6.  {
7.      string temp = "";
8.      for (int j = 0; j < panjang_gen; j++)
9.      {
10.         int bil = data.Next(0, 2);
11.         temp = temp + bil.ToString();
12.     }
13.     populasi[i] = temp;
14. }
```

***Sourcecode* 4.19** Proses Pembangkitan Populasi Awal

2. Proses Crossover

Individu yang akan dicrossover ditampung pada `ArrayList individu_crossover`. Penggunaan `ArrayList` dikarenakan jumlah objek yang akan ditampung belum diketahui banyaknya. Variabel `double banding` digunakan untuk membangkitkan

bilangan random antara 0 – 1. Bilangan ini digunakan untuk menentukan apakah suatu individu akan dicrossover atau tidak. Bilangan antara 0 – 1 tersebut yang kurang dari probabilitas crossover akan membuat individu tersebut dicrossover. Individu hasil crossover ditampung pada ArrayList anakan_crossover.

```
1.  // Melakukan perulangan sampai sejumlah generasi
2.  for (int awal = 0; awal < Max_Generasi; awal++)
3.  {
4.      // proses crossover
5.  ArrayList individu_crossover = new ArrayList();
6.  // menentukan individu yang akan mengalami
    crossover
7.  for (int ii = 0; ii < jumlah_individu; ii++)
8.      {
9.          double banding = data.NextDouble();
10.         if (banding < Pc)
11.         {
12.       individu_crossover.Add(populasi[ii]);
13.         }
14.     }
15. object[] individu_cross_fix =
    individu_crossover.ToArray();
16. ArrayList anakan_crossover = new ArrayList();
17.
18. int posisi_crossover = data.Next(1, panjang_gen-
    1);
19. for (int k = 0; k < individu_cross_fix.Length -
    1; k++)
20.     {
21. for (int l = k + 1; l <
    individu_cross_fix.Length; l++)
22.         {
23.             string parent1a = "";
    string parent1b = "";
24.             string parent2a = "";
    string parent2b = "";
25.
26. parent1a =
    individu_cross_fix[k].ToString().Substring(0,
    posisi_crossover);
27. parent1b =
    individu_cross_fix[l].ToString().Substring(0,
    posisi_crossover);
28.
29. parent2a =
    individu_cross_fix[k].ToString().Substring(posisi
```

```
    _crossover);
30. parent2b =
    individu_cross_fix[l].ToString().Substring(posisi
    _crossover);
31.
32.             string hasil1 = parent1a + parent2b;
33.             string hasil2 = parent1b + parent2a;
34.
35.             anakan_crossover.Add(hasil1);
36.             anakan_crossover.Add(hasil2);
37.         }
38. }
```

***Sourcecode* 4.20** Proses *Crossover*

3. Proses Mutasi

`ArrayList anakan_mutasi` digunakan untuk menampung individu hasil mutasi. Awalnya individu yang bertipe data string dikonversi menjadi tipe data `char[]` agar dapat dioperasikan per karakter penyusunnya (yang biasa dilihat sebagai gen). Variabel `double banding1` digunakan untuk menentukan apakah gen tersebut mengalami mutasi atau tidak.

```
1.  // proses mutasi
2.  ArrayList anakan_mutasi=new ArrayList();
3.  for (int p = 0; p < populasi.Length; p++)
4.  {
5.  char[] child = populasi[p].ToCharArray();
6.     bool cekMutasi = false;
7.     for (int r = 0; r <child.Length; r++)
8.     {
9.        double banding1=data.NextDouble();
10.       if(banding1 < Pm)
11.       {
12.          cekMutasi = true;
13.          if (child[r]== '1')
14.          {
15.             child[r] = '0';
16.          }
17.          else
18.          {
19.              child[r] = '1';
20.          }
21.       }
22.    }
23.    if (cekMutasi)
24.    {
```

```
25.         string child1 = new string(child);
26.         anakan_mutasi.Add(child1);
27.     }
28. }
```

***Sourcecode* 4.21** Proses Mutasi

4. Penghitungan Bobot (w)

Penghitungan bobot digunakan untuk menghitung fitness tiap individu. Meski akhirnya nilai bobot yang dipakai, tetapi dengan menghitung fitness bisa menghasilkan bobot dengan nilai tunggal yang optimal. Nilai w dihitung untuk tiap kromosom pada suatu individu. Perlu dilakukan pemecahan individu untuk membaginya dalam kromosom-kromosom sesuai jumlah gen yang diketahui pada variabel Li. Variabel `double[] W` merupakan hasil perhitungan nilai W pada individu.

```
1.  private static double[] HitungW(string individu)
2.  {
3.      string temp = "";
4.      double[] X = new double[6];
5.      double[] W = new double[6];
6.      double X_total = 0; int urutan = 0;
7.      // menghitung nilai X
8.  for (int indeks = 0; indeks < individu.Length;
    indeks++)
9.      {
10.        temp = temp + individu[indeks];
11.        if ((indeks + 1) % Li == 0)
12.        {
13.           double v = BinerToDesimal(temp);
14. double x_i = (1 / (Math.Pow(2, Li) - 1)) * v;
15.            X_total = X_total + x_i;
16.           X[urutan] = x_i;
17.           urutan++;
18.           temp = "";
19.         }
20.     }
21.      // menghitung nilai W
22.      for (int p = 0; p < X.Length; p++)
23.      {
24.        W[p] = X[p] / X_total;
25.     }
26.     return W;
27. }
```

***Sourcecode* 4.22** Proses Penghitungan Bobot (w)

5. Penghitungan Nilai Fitness

Fungsi `HitungFitness()` digunakan untuk mengetahui nilai fitness pada suatu individu. Input parameter berupa `string individu` dan `double[,] b_ij`. Variabel `double[,] b_ij` merupakan nilai ekstraksi kalimat suatu dokumen. Untuk menghitung nilai fitness suatu individu diperlukan nilai W yang didapat dengan memanggil fungsi `HitungW()`.

```
1.  private static double HitungFitness(string
    individu, double[,] b_ij)
2.  {
3.      double hasil = 0;
4.      double[] W = HitungW(individu);
5.
6.      double[] max_bj = new double[6];
7.      // menghitung nilai max bj
8.    for(int y1=0; y1< b_ij.GetLength(1); y1++)
9.     {
10.        double max = 0;
11.     for(int y2=0; y2<b_ij.GetLength(0); y2++)
12.        {
13.            double bj = b_ij[y2,y1];
14.            if(bj > max)
15.            {
16.                max = bj;
17.            }
18.         }
19.        max_bj[y1] = max;
20.     }
21.
22. for(int row=0; row<b_ij.GetLength(0); row++)
23.     {
24.        double fitness = 0; double nilai = 0;
25. for(int col=0; col<b_ij.GetLength(1); col++)
26.        {
27. fitness = ((Math.Pow( max_bj[col] - b_ij[row,
    col] , 2)) * (Math.Pow(W[col] , 2)));
28.          nilai = nilai + fitness ;
29.        }
30.       hasil = hasil + nilai;
31.     }
32.     return hasil;
33. }
```

***Sourcecode* 4.23** Proses Penghitungan Nilai *Fitness*

6. Perangkingan Kalimat

```
1.  string best_fitness = populasi[0];
2.  double[] W_best = HitungW(best_fitness);
3.
4.  double[] perangkingan = new
    double[b_ij.GetLength(0)];
5.  for(int xx=0; xx < b_ij.GetLength(0); xx++ )
6.  {
7.      double g=0;
8.  for (int yy=0; yy < b_ij.GetLength(1); yy++)
9.      {
10.         g = g + (b_ij[xx,yy]*W_best[yy]);
11.     }
12.     perangkingan[xx] = g;
13. }
14. // ambil kalimat dengan nilai terbaik sejumlah
    compression rate
15.
16. int jumlah_comp
    =(int)(Math.Round(b_ij.GetLength(0) *
    compression_rate, 0));
17. int[] urutan_kalimat = new int[jumlah_comp];
18.
19. for (int v = 0; v < urutan_kalimat.Length; v++)
20. {
21.     double max = -1; int indek_max = 0;
22. for (int w = 0; w < perangkingan.Length; w++)
23.     {
24.         if (perangkingan[w] > max)
25.         {
26.         max = perangkingan[w]; indek_max = w;
27.         }
28.     }
29.     urutan_kalimat[v] = indek_max;
30.     perangkingan[indek_max] = -1;
31.   }
32.   return urutan_kalimat;
33. }
```

***Sourcecode* 4.24** Proses Perangkingan

4.2.4.3 Tahap Merging

Fungsi `Merging()` menjelaskan tentang perhitungan *cosine similarity* pada kalimat hasil proses genetika. Nilai *cosine similarity* dihitung menggunakan nilai TF (frekuensi kata). Variabel `double[] nilai_similarity` digunakan untuk menampung nilai similarity

semua kalimat. Perhitungan *cosine similarity* dapat dilihat pada source code berikut:

```
1.  public static double[] Merging(ArrayList
    hasil_tiap_cluster)
2.  {
3.    object[] input_ = prepocessing.Hitung(
    hasil_tiap_cluster.ToArray());
4.    double[] O=new double[input_.Length];
5.    double[,] perkalian_skalar = new
    double[input_.Length, input_.Length];
6.    double[] nilai_similarity = new
    double[input_.Length];
7.    int k = 0;
8.   // Menghitung panjang setiap kalimat.
9.   foreach (SortedDictionary<string, int> kalimat
    in input_)
10.  {
11.       double nilai_O = 0;
12.       foreach (var kata in kalimat)
13.       {
14.            nilai_O = nilai_O + kata.Value *
    kata.Value;
15.        }
16.       nilai_O = Math.Sqrt(nilai_O);
17.       O[k] = nilai_O;
18.       k++;
19.    }
20.   //MEnghitung perkalian skalar antar kalimat ke-i
    dengan kalimat lainnya.
21.   // HAsil perkalian tersebut dijumlahkan.
22.    int row = 0; int col = 0;
23.    foreach (SortedDictionary<string, int>
    kalimat_i in input_)
24.    {
25.       col = 0;
26.       foreach (SortedDictionary<string, int>
    kalimat_lainnya in input_)
27.       {
28.            double hasil_skalar =
    PerkalianSkalar(kalimat_i, kalimat_lainnya);
29.            perkalian_skalar[row, col] =
    hasil_skalar;
30.            col++;
31.        }
32.       row++;
33.    }
34. // Menerapkan rumus cosine similarity
```

```
35.   for (int aa = 0; aa < perkalian_skalar.GetLength(0); aa++)
36.   {
37.       double sim = 0;
38.       for (int b = 0; b < perkalian_skalar.GetLength(1); b++)
39.       {
40.           if (aa == b) continue;
41.           sim = sim + (perkalian_skalar[aa, b] / (O[aa] * O[b]));
42.       }
43.       nilai_similarity[aa] = sim;
44.   }
45.   return nilai_similarity;
46. }
```

***Sourcecode* 4.25** Proses Merging Kalimat

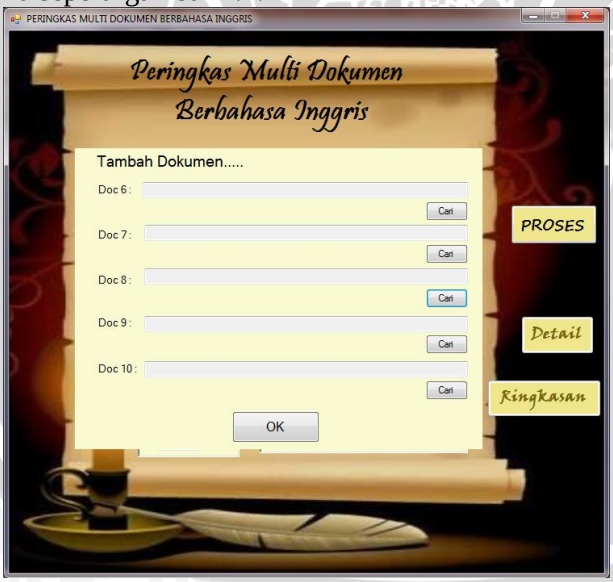
4.3 Penerapan Aplikasi

Pada form utama terdapat 5 tombol "Cari" yang digunakan user untuk memasukkan dokumen-dokumen yang akan diringkas. Tombol "Tambah Dokumen" digunakan jika user ingin menambahkan dokumen yang akan diringkas sampai 10 dokumen. TextBox pada setiap tombol "Cari" berisi *path* file yang diinputkan user. Setelah user memasukkan dokumen input, tombol "PROSES" baru dapat diakses. Tombol "PROSES" digunakan untuk mengklaster dokumen input. Hasil dari klaster akan muncul pada richTextBox "Clustering". Setelah itu, tombol "Detail" dan tombol "Ringkasan" akan dapat diakses. Tombol "Detail" digunakan untuk melihat rincian hasil proses *preprocessing*, ekstraksi kalimat dan algoritma genetika yang ditampung pada form "Detail". Tombol "Ringkasan" digunakan untuk melihat hasil ringkasan dokumen dan dokumen asli yang ditampilkan pada form "Ringkasan". TextBox "Jumlah Dokumen" berisi jumlah dokumen yang telah diinputkan oleh user. User dapat mengatur ukuran ringkasan dengan memilih prosentase ukuran ringkasan pada comboBox "Ukuran ringkasan".

Dokumen diinputkan dengan menekan tombol "Cari" seperti pada form di gambar 4.1. Pada form tersebut diketahui jumlah dokumen yang diinputkan sebanyak 4. Prosentase ukuran ringkasan yang dapat dipilih adalah 25%, 50%, dan 75%.

Gambar 4.1 Form Utama

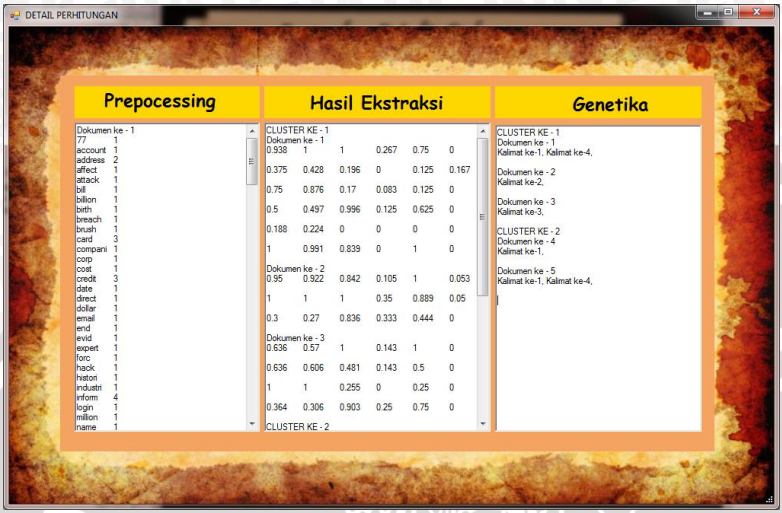
Jika dokumen yang ingin diinputkan lebih dari 5, maka setelah ditekan tombol "Tambah Dokumen" akan muncul panel baru pada form utama seperti gambar 4.2.



Gambar 4.2 Panel Tambah Dokumen

Gambar 4.3 Clustering Dokumen

Dari 4 dokumen yang telah diinputkan, ukuran ringkasan yang dipilih adalah 25%, setelah itu ditekan tombol "PROSES" dan proses *clustering* menghasilkan pengelompokkan dokumen seperti pada panel *clustering* yaitu terdiri dari 2 *cluster*. *Cluster* 1 berisi dokumen 1, 2, dan 3, sedangkan *cluster* 2 berisi dokumen 4 dan 5. Jika ingin melihat detail perhitungan dapat dilakukan dengan menekan tombol "Detail". Kemudian akan muncul form "Detail Perhitungan" yang terdiri atas 3 panel *preprocessing* yang menampilkan daftar kata yang telah melalui proses *preprocessing* beserta frekuensinya, panel ekstraksi yang akan menampilakan matriks kalimat × fitur, dan panel genetika untuk menampilkan hasil perangkingan dengan algoritma genetika yang berisi kalimat dari suatu dokumen.



Gambar 4.4 Form Detail Perhitungan

Gambar 4.5 Form Ringkasan

Form “Ringkasan” berfungsi untuk menampilkan menu tab yang berisi dokumen asli (pada menu tab “original”) dan ringkasan dokumen (pada menu tab “ringkasan”) sesuai *cluster*. pada form ini terdapat tombol “Save To” yang berfungsi untuk menyimpan ringkasan dalam bentuk file .txt.

4.4 Skenario Pengujian

Dokumen yang diujikan adalah dokumen teks dengan format file.txt. Dokumen teks merupakan suatu rangkaian kata-kata yang memiliki isi dan bentuk untuk menyampaikan pesan tertentu dan menurut isi, sintaksis, dan pragmatik membentuk satu kesatuan yang berinteraksi menghasilkan makna yang utuh.

Dokumen yang diujikan berisi teks berbahasa Inggris. Tiap file dokumen yang diinputkan berisi satu judul wacana. Untuk mengetahui sejauh mana sistem mencakup informasi yang relevan dilakukan dengan pengujian. Suatu dokumen dikatakan relevan jika ada kecocokan dengan kebutuhan informasi yang diinginkan *user*. Pengujian akan dilakukan pada berbagai macam dokumen dengan konten yang berbeda sehingga dihasilkan beberapa *cluster*.

Pengujian dilakukan dengan membandingkan hasil ringkasan manusia dan hasil ringkasan sistem. Pengujian dilakukan untuk mengetahui seberapa dekat hasil ringkasan sistem dengan hasil ringkasan manusia. Untuk itu dilakukan penghitungan terhadap tiga parameter yang berbeda yaitu precision, recall, dan F-measure. Ketiga parameter ini dihitung untuk ukuran ringkasan dokumen yang berbeda, yaitu 25%, 50%, dan 75%.

Presicion adalah jumlah kalimat yang benar dibagi dengan jumlah semua kalimat dalam ringkasan. Recall adalah jumlah kalimat yang benar dibagi dengan jumlah kalimat yang seharusnya dikenali. Pada evaluasi sistem ini, kalimat yang benar adalah kalimat dalam ringkasan manusia. Perhitungan *precision* sesuai dengan persamaan 2.8 dan *recall* sesuai dengan persamaan 2.10. Untuk perhitungan *F-measure* menggunakan nilai *precision* dan *recall* seperti pada persamaan 2.11. Setelah dilakukan pengujian akan diketahui kemampuan sistem dalam menghasilkan informasi yang relevan yang diinginkan user.

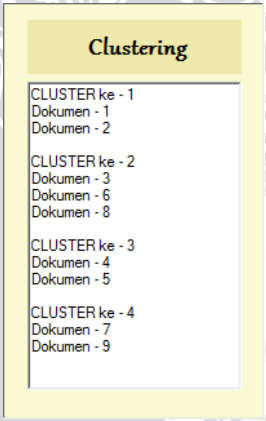
4.5 Hasil Pengujian

Pengujian dilakukan pada berbagai macam dokumen dengan jumlah kalimat yang berbeda dan konten yang berbeda. Berikut ini dokumen-dokumen uji (isi dokumen dapat dilihat pada lampiran) :

Pengujian dilakukan pada seberapa dekat hasil ringkasan yang dilakukan sistem dengan hasil ringkasan manusia.

Doc 1 : Alternative fuel vehicle (19 kalimat). (5, 10, 14)
Doc 2 : Automakers Produce Cars of the Future (16 kalimat). (4, 8, 12)
Doc 3 : Cutting Salt Might Increase Heart Risks (16 kalimat). (4, 8, 12)
Doc 4 : Global Effort Heal Antarctic Ozone Hole (26 kalimat). (7, 13, 20)
Doc 5 : Ozone and the Ozone Layer (24 kalimat). (6, 12, 18)
Doc 6 : Pressure Mounts in US to Restrict Salt Levels in Processed Foods (17 kalimat). (4, 9, 13)
Doc 7 : Remedies to Erase Wrinkles and Aging (18 kalimat). (5, 9, 14)
Doc 8 : Study Shows Coffee Does Not Increase Blood Pressure (15 kalimat). (4, 8, 12)
Doc 9 : What Ingredients Must Seem for Wrinkle Lotions (23 kalimat). (6, 12, 17)

Setelah dikelompokkan hasilnya :



Gambar 4.6 *Clustering* Dokumen Uji

Ringkasan hasil sistem dan hasil manusia dengan ukuran ringkasan 25%, 50%, dan 75% untuk *cluster* 1 sampai *cluster* 4 dapat dilihat pada lampiran 3. Berikut ini contoh perbandingan hasil ringkasan sistem dan ringkasan manusia untuk *cluster* 1 dengan ukuran ringkasan 25% :

Hasil sistem

Due to a combination of heavy taxes on fuel, particularly in Europe, tightening environmental laws, particularly in California, and the possibility of further restrictions on greenhouse gas emissions, work on alternative power systems for vehicles has become a high priority for governments and vehicle manufacturers around the world. Solar cars are not a practical form of transportation; insufficient power falls on the roof of a practically sized and shaped vehicle to provide adequate performance. Other RD efforts in alternative forms of power focus on developing fuel cells, alternative forms of combustion such as GDI and HCCI and even the stored energy of compressed air. Mercedes' Smart car has been popular in Europe for some time. While both can be obtained from petroleum or natural gas, ethanol may be the most interesting because many believe it to be a renewable resource, easily obtained from sugar or starch in crops and other agricultural produce such as grain, sugarcane or even lactose. It is called PUMA which means Personal Urban Mobility and Accessibility Vehicle. Since ethanol occurs in nature whenever yeast happens to find a sugar solution such as overripe fruit, most organisms have evolved some tolerance to ethanol, whereas methanol is toxic. Chrysler's GEM Peapod and Toyota's IQ are two small cars that have been specially designed for city traffic. General Motors has already showed one of its prototypes.

Hasil manusia

Due to a combination of heavy taxes on fuel, particularly in Europe, tightening environmental laws, particularly in California, and the possibility of further restrictions on greenhouse gas emissions, work on alternative power systems for vehicles has become a high priority for governments and vehicle manufacturers around the world.

Automakers around the world are planning to make cars that are smaller, use up less fuel and do not damage our environment.

It is called P.U.M.A. , which means Personal Urban Mobility and Accessibility Vehicle.

While GM's car may never be produced other car makers have

already made mini-cars that you can buy.
Chrysler's GEM Peapod and Toyota's IQ are two small cars that have been specially designed for city traffic.
Man people want to buy small cars because they save fuel.
Other R&D efforts in alternative forms of power focus on developing fuel cells, alternative forms of combustion such as GDI and HCCI, and even the stored energy of compressed air.
Alternative Fuel Vehicle refers to a vehicle that runs on a fuel other than traditional gasoline or diesel; any method of powering an engine that does not involve petroleum.
Current research and development is largely centered on "hybrid" vehicles that use both electric power and internal combustion.

Jumlah *correct* dari ringkasan tersebut adalah 4, dengan *missed* (yang berwarna biru) 5 dan *wrong* (yang berwarna ungu) 5. Nilai precision dan nilai recall dari ringkasan tersebut sama-sama 0.4444 , sehingga nilai F-measurenya pun juga sama yaitu 0.4444.

4.6 Analisa Hasil

Penghitungan ketiga parameter evaluasi dilakukan untuk semua cluster dengan ukuran ringkasan 25%, 50%, dan 75%. Dari hasil pada subbab sebelumnya perhitungan evaluasi sistem ditunjukkan pada tabel 4.1 berikut :

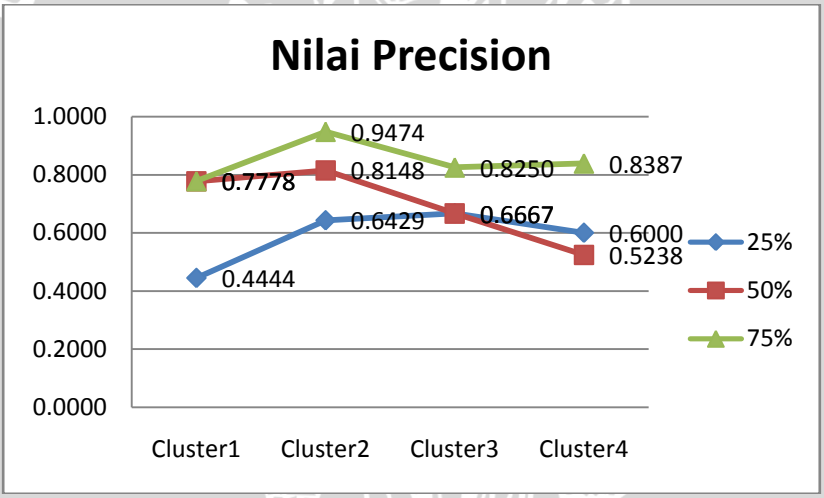
Tabel 4.1 Perhitungan Evaluasi Sistem

		Dokumen Cluster			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
25%	Precision	0.4444	0.6429	0.6667	0.6000
	Recall	0.4444	0.6923	0.6154	0.6000
	F-Measure	0.4444	0.6667	0.6400	0.6000
50%	Precision	0.7778	0.8148	0.6667	0.5238
	Recall	0.7778	0.8148	0.6154	0.5238
	F-Measure	0.7778	0.8148	0.6400	0.5238
75%	Precision	0.7778	0.9474	0.8250	0.8387
	Recall	0.7778	0.9000	0.8462	0.8387
	F-Measure	0.7778	0.9231	0.8354	0.8387

Efektifitas ringkasan dokumen dapat dilihat dengan seberapa banyak menemukan semua informasi yang relevan (*precision*) dan kemampuannya untuk tidak menemukan informasi yang tidak relevan (*recall*).

Precision merupakan perbandingan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi yang terambil oleh sistem baik yang relevan maupun tidak. *Precision* digunakan untuk mengukur kemampuan sistem dalam menampilkan hanya kalimat yang tepat. *Precision* akan mengevaluasi seberapa baik ketepatan sistem dalam memprediksi kalimat penting untuk menjadi ringkasan. Semakin tinggi nilai *precision* semakin baik sistem dalam melewatkan kalimat yang tidak relevan.

Nilai *precision* terendah dihasilkan oleh *cluster* 1 dengan ukuran ringkasan 25%. Nilai *precision* tertinggi dihasilkan *cluster* 2 dengan ukuran ringkasan 75%. Grafik nilai *precision* ditunjukkan pada Gambar 4.6 berikut:

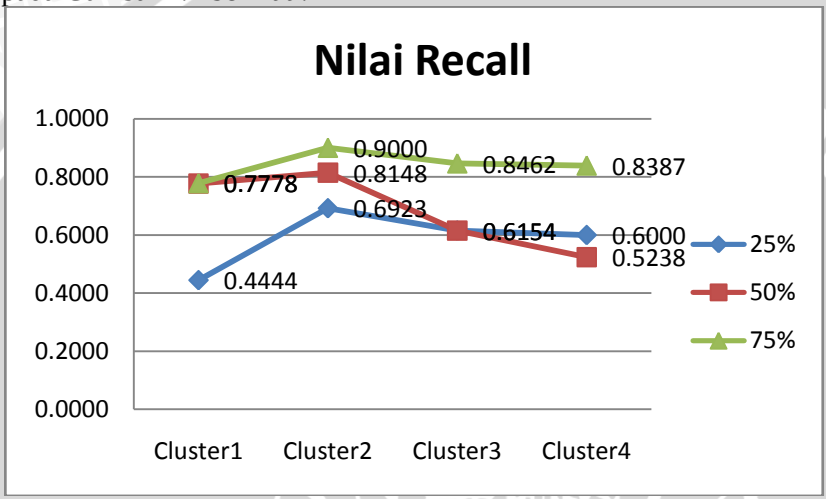


Gambar 4.6 Grafik Nilai Precision

Recall merupakan perbandingan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi relevan yang ada dalam koleksi informasi (baik yang terambil atau tidak terambil oleh sistem). *Recall* digunakan untuk mengukur kemampuan sistem dalam menampilkan seluruh kalimat yang tepat dalam ringkasan. Semakin tinggi nilai *recall* semakin efektif sistem dalam mengambil kalimat-

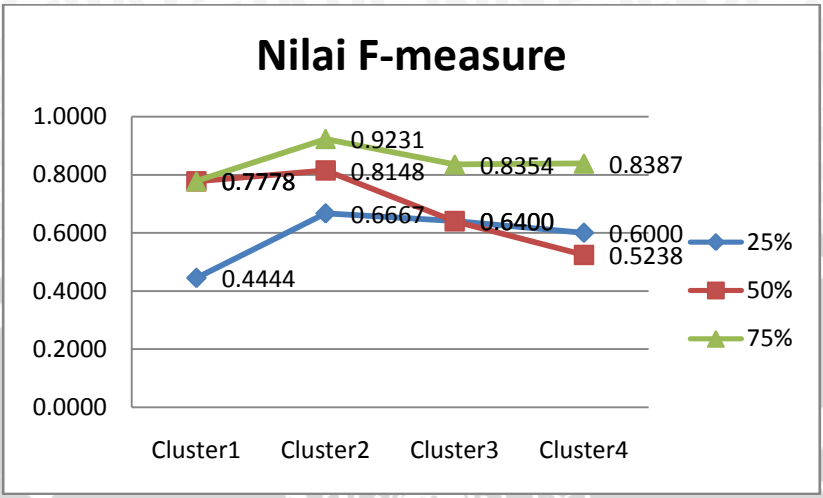
kalimat yang relevan. *Recall* dihitung untuk mengevaluasi seberapa besar cakupan suatu sistem dalam memprediksi kalimat penting untuk menjadi ringkasan.

Nilai *recall* terendah dihasilkan oleh *cluster* 1 dengan ukuran ringkasan 25% sedangkan nilai *recall* tertinggi dihasilkan oleh *cluster* 2 dengan ukuran ringkasan 75%. Grafik recall ditunjukkan pada Gambar 4.7 berikut :



Gambar 4.7 Grafik Nilai Recall

Untuk menetukan evaluasi yang paling baik, digunakan nilai *F-measure* yang merupakan kombinasi dari nilai *recall* dan *precision*. *F-measure* merepresentasikan akurasi sistem, sehingga jika nilai F-measure tinggi, berarti akurasi sistem tinggi. Representasi nilai *F-measure* ditunjukkan pada Gambar 4.8 berikut:



Gambar 4.8 Grafik Nilai F-measure

Sama dengan nilai *precision* dan *recall*, nilai *F-measure* terendah dihasilkan oleh *cluster* 1 dengan ukuran ringkasan 25%, sedangkan nilai tertinggi dihasilkan oleh *cluster* 2 dengan ukuran ringkasan 75%. Untuk ukuran ringkasan 25% didapatkan rata-rata nilai *F-measure* 0.5878, ukuran ringkasan 50% didapatkan rata-rata nilai *F-measure* 0.6891, dan untuk ukuran ringkasan 75% rata-rata nilai *F-measure* sebesar 0.8438. Di sini terlihat nilai *F-measure* semakin naik untuk ukuran ringkasan yang semakin besar pula. Rata-rata nilai *F-measure* merepresentasikan akurasi sistem ini, yaitu 0.7069.

Ada dua aspek penting dalam mengukur sistem *information retrieval*, yaitu pemanfaatan sumber daya sehingga pemahaman terhadap dokumen dilakukan dengan mudah dan keakuratan untuk mendapatkan informasi yang relevan. Denagn demikian, peringkasan dokumen ini telah memenuhi syarat *information retrieval* yang baik, yaitu peringkasan dilakukan secara otomatis menggunakan komputer, begitu juga pengelompokkan dokumen otomatis sebelum diringkas dan cukup akuratnya sistem ini dalam menyajikan ringkasan yang ditunjukkan dengan rata-rata nilai F-measure.

Akurasi yang dihasilkan dipengaruhi pula oleh kelemahan sistem ini. Kelemahan sistem ini dapat dikarenakan dalam proses *preprocessing*, metode *stemming* yang digunakan kurang optimal.

Seperti pada *cluster* 1 kata *made*, *make* dan *maker* menjadi kata yang berbeda atau kata *react* dan *reaction* pada *cluster* 3. Seharusnya ada kamus kata dasar atau metode untuk mengembalikan kata jadian ke kata asal (past verb dan past participle verb ke infinitife/present verb). Dengan perbedaan kata-kata yang seharusnya sama maka akan berpengaruh pada nilai bobot kata (fitur 2) dan nilai *similarity* (fitur 3) menjadi semakin kecil.

BAB V
KESIMPULAN DAN SARAN

Pada bab ini akan dipaparkan mengenai kesimpulan dari penelitian yang telah dilakukan serta saran untuk penelitian lebih lanjut atau bagi pihak-pihak yang membutuhkan.

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini sebagai berikut :

1. Pada skripsi ini telah dibuat sistem peringkasan multi dokumen berbahasa Inggris berbasis konten menggunakan *single pass clustering* dan perangkingan berbasis algoritma genetika.
2. Evaluasi sistem dilakukan dengan membandingkan hasil ringkasan sistem dan hasil manusia. Dari hasil tersebut diperoleh nilai F-*measure* tertinggi untuk ukuran ringkasan 75% yaitu sebesar 0.8438, sedangkan nilai F-*measure* untuk ukuran ringkasan 25% sebesar 0.5878 dan ukuran ringkasan 50% sebesar 0.6891. Jadi semakin besar ukuran ringkasan semakin tinggi nilai F-*measure*. Pengujian ini dilakukan dengan inisialisasi parameter algoritma genetika 50 iterasi, 20 individu, probabilitas crossover 0.8, dan probabilitas mutasi 0.01. Rata-rata nilai F-*measure* untuk semua ukuran ringkasan merepresentasikan akurasi sistem yaitu sebesar 0.7069.

5.2 Saran

Saran yang dapat diberikan setelah dilakukan penelitian ini sebagai berikut:

1. Dapat menggunakan metode khusus untuk melakukan *merging* antar ringkasan dokumen dalam satu *cluster* misalnya dengan metode *chronological sentence ordering* dan dapat dilakukan evaluasi terhadap susunan kalimat ringkasan dalam satu *cluster*.
2. Untuk meningkatkan akurasi sistem dapat ditambahkan fitur pada ekstraksi kalimat seperti fitur *Sentence-to-Sentence Cohesion* dan *Sentence-to-Centroid Cohesion*.

DAFTAR PUSTAKA

Ahmed. 2006. *Using Genetic Algorithm to Improve Information Retrieval System*. World Academy of Science, Engineering and Technology.

Basagi R., Krupic D., dan Suzic, Bojan. 2009 *Automatic Text Summarization*. Institute for Information Systems and Computer Media Graz University of Technology. Graz.

Basuki, Ahmad. 2003. *Algoritma Genetika*. Politeknik Elektronika Negeri Surabaya. Surabaya.

Bubenhofer, Noah. 2002. *Text Summarization*. Applications of Computational Linguistics Pius ten Hacken, English Seminar University of Basel. Basel.

Chong, L. dan Chen, Y.. 2009. *Text Summarization for Oil and Gas News Article*. World Academy of Science, Engineering and Technology.

Dias, G., Alves, E., dan Nunes, C.. 2005. *Topic Segmentation: How Much Can We Do By Counting Words And Sequences of Words*. Beira Interior University. Covilha.

Fattah, M. A. dan Ren, Fuji. 2008. *Automatic Text Summarization*. World Academy of Science, Engineering and Technology. Singapore.

Fleming dan Purshouse. 2001. *Genetic Algorithm in Control System Engineering*. Department of Automatic Control and Systems Engineering University of Sheffield. Sheffield.

Gen, M. Cheng, R. 2000. *Genetic Algorithms and Engineering Optimization*. John Wiley & Son, Inc. USA.

Glockner. *Fuzzy Information Retrieval*. 2005. Int. Information and Communication Systems Group of Prof. Helbig, Seminar Softcomputing University Hagen. Hagen.

Hariharan, S. 2010. *Extraction Based Multi Document Summarization using Single Document Summary Cluster*. B.S. Abdur Rahman University. Vandalur.

Haupt, Randy dan Haupt, Sue E.. 2004. *Practical Genetic Algorithms Second Edition*. John Wiley & Sons, Inc. New Jersey.

Hovy, Eduard. 2003. *Handbook Text Summarization*. Oxford University Press. . Oxford.

Kogilavani, A., dan Balasubramani, P.. 2010. *Clustering Feature Specific Sentence Extraction Based Summarization of Multiple Documents*. Kongu Engineering College. Erode.

Kuo, J. dan Chen, H.. 2006. *Cross-document Event Clustering Using Knowledge Mining From Co-reference Chains*. National Taiwan University. Taipei.

Kusumadewi, Sri. 2005. *Pencarian Bobot Atribut Pada Multiple Attribute Decision Making (MADM) dengan Pendekatan Obyektif Menggunakan Algoritma Genetika*. Gematika Jurnal Manajemen Informatika. Jakarta.

Llorent, Elena. *Text Summarization : An Overview*. Universidad de Alicante. San Vicente del Raspeig.

Melanie, Mitchell. 1999. An *Introduce to Genetic Algorithms*. Cambridge MIT Press. London.

Nedunchelian, R., Muthucumarasamy, R., dan Saranathan, E.. 2011. *Comparison Of Multi Document Summarization Techniques*. IJCSNS (International Journal of Computer Science and Network Security).

Nopriadi, I. P. N.. 2009. Algoritma Genetika Dasar Komputasi Cerdas. Teknik Elektro Universitas Udayana. Bali.

Okumura, Manabu. 2007. *Text Summarization*. Tokyo Institute of Technology. Tokyo.

Permadi, Tedi. 2011. Teks, *Tekstologi, dan Kritik Teks*. Universitas Pendidikan Indonesia. Bandung.

Sanjoyo. 2006. *Aplikasi Algoritma Genetika*.

Sivanandam, S. dan Deepa, S.. 2008. *Introduction to Genetic Algorithms*. Springer. Berlin.

Sofyan, Agus Nero. 2007. *Bahasa Indonesia Dalam Penulisan Karya Imliah.* Universitas Widyatama. Bandung.

Yogatama, Dani. 2008. *Studi Penggunaan Stemming untuk Meningkatkan Performasi Sistem Temu Balik Informasi*. Institut Teknologi Bandung. Bandung.

LAMPIRAN 1

Daftar *Stop Word*

Sumber : http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words/

1.	a	33.	anyhow	65.	behind
2.	a's	34.	anyone	66.	being
3.	able	35.	anything	67.	believe
4.	about	36.	anyway	68.	below
5.	above	37.	anyways	69.	beside
6.	according	38.	anywhere	70.	besides
7.	accordingly	39.	apart	71.	best
8.	across	40.	appear	72.	better
9.	actually	41.	appreciate	73.	between
10.	after	42.	appropriate	74.	beyond
11.	afterwards	43.	are	75.	both
12.	again	44.	aren't	76.	brief
13.	against	45.	around	77.	but
14.	ain't	46.	as	78.	by
15.	all	47.	aside	79.	c
16.	allow	48.	ask	80.	c'mon
17.	allows	49.	asking	81.	c's
18.	almost	50.	associated	82.	came
19.	alone	51.	at	83.	can
20.	along	52.	available	84.	can't
21.	already	53.	Away	85.	cannot
22.	also	54.	awfully	86.	cant
23.	although	55.	b	87.	cause
24.	always	56.	be	88.	causes
25.	am	57.	became	89.	certain
26.	among	58.	because	90.	certainly
27.	amongst	59.	become	91.	changes
28.	an	60.	becomes	92.	clearly
29.	and	61.	becoming	93.	co
30.	another	62.	been	94.	com
31.	any	63.	before	95.	come
32.	anybody	64.	beforehand	96.	comes

97.	concerning	129.	eight	161.	forth
98.	consequently	130.	either	162.	four
99.	consider	131.	else	163.	from
100.	considering	132.	elsewhere	164.	further
101.	contain	133.	enough	165.	furthermore
102.	containing	134.	entirely	166.	g
103.	contains	135.	especially	167.	get
104.	corresponding	136.	et	168.	gets
105.	could	137.	etc	169.	getting
106.	couldn't	138.	even	170.	given
107.	course	139.	ever	171.	gives
108.	currently	140.	every	172.	go
109.	d	141.	everybody	173.	goes
110.	definitely	142.	everyone	174.	going
111.	described	143.	everything	175.	gone
112.	despite	144.	everywhere	176.	got
113.	did	145.	ex	177.	gotten
114.	didn't	146.	exactly	178.	greetings
115.	different	147.	example	179.	h
116.	do	148.	except	180.	had
117.	does	149.	f	181.	hadn't
118.	doesn't	150.	far	182.	happens
119.	doing	151.	few	183.	hardly
120.	don't	152.	fifth	184.	has
121.	done	153.	first	185.	hasn't
122.	down	154.	five	186.	have
123.	downwards	155.	followed	187.	haven't
124.	during	156.	following	188.	having
125.	e	157.	follows	189.	he
126.	each	158.	for	190.	he's
127.	edu	159.	former	191.	hello
128.	eg	160.	formerly	192.	help

193.	hence	225.	indicate	257.	less
194.	her	226.	indicated	258.	lest
195.	here	227.	indicates	259.	let
196.	here's	228.	inner	260.	let's
197.	hereafter	229.	insofar	261.	like
198.	hereby	230.	instead	262.	liked
199.	herein	231.	into	263.	likely
200.	hereupon	232.	inward	264.	little
201.	hers	233.	is	265.	look
202.	herself	234.	isn't	266.	looking
203.	hi	235.	it	267.	looks
204.	him	236.	it'd	268.	ltd
205.	himself	237.	it'll	269.	m
206.	his	238.	it's	270.	mainly
207.	hither	239.	its	271.	many
208.	hopefully	240.	itself	272.	may
209.	how	241.	j	273.	maybe
210.	howbeit	242.	just	274.	me
211.	however	243.	k	275.	mean
212.	i	244.	keep	276.	meanwhile
213.	i'd	245.	keeps	277.	merely
214.	i'll	246.	kept	278.	might
215.	i'm	247.	know	279.	more
216.	i've	248.	knows	280.	moreover
217.	ie	249.	known	281.	most
218.	if	250.	l	282.	mostly
219.	ignored	251.	last	283.	much
220.	immediate	252.	lately	284.	must
221.	in	253.	later	285.	my
222.	inasmuch	254.	latter	286.	myself
223.	inc	255.	latterly	287.	n
224.	indeed	256.	least	288.	name

289.	namely	321.	okay	353.	provides
290.	nd	322.	old	354.	q
291.	near	323.	on	355.	que
292.	nearly	324.	once	356.	quite
293.	necessary	325.	one	357.	qv
294.	need	326.	ones	358.	r
295.	needs	327.	only	359.	rather
296.	neither	328.	onto	360.	rd
297.	never	329.	or	361.	re
298.	nevertheless	330.	other	362.	really
299.	new	331.	others	363.	reasonably
300.	next	332.	otherwise	364.	regarding
301.	nine	333.	ought	365.	regardless
302.	no	334.	our	366.	regards
303.	nobody	335.	ours	367.	relatively
304.	non	336.	ourselves	368.	respectively
305.	none	337.	out	369.	right
306.	noone	338.	outside	370.	s
307.	nor	339.	over	371.	said
308.	normally	340.	overall	372.	same
309.	not	341.	own	373.	saw
310.	nothing	342.	p	374.	say
311.	novel	343.	particular	375.	saying
312.	now	344.	particularly	376.	says
313.	nowhere	345.	per	377.	second
314.	o	346.	perhaps	378.	secondly
315.	obviously	347.	placed	379.	see
316.	of	348.	please	380.	seeing
317.	off	349.	plus	381.	seem
318.	often	350.	possible	382.	seemed
319.	oh	351.	presumably	383.	seeming
320.	ok	352.	probably	384.	seems

385.	seen	417.	such	449.	these
386.	self	418.	sup	450.	they
387.	selves	419.	sure	451.	they'd
388.	sensible	420.	t	452.	they'll
389.	sent	421.	t's	453.	they're
390.	serious	422.	take	454.	they've
391.	seriously	423.	taken	455.	think
392.	seven	424.	tell	456.	third
393.	several	425.	tends	457.	this
394.	shall	426.	th	458.	thorough
395.	she	427.	than	459.	thoroughly
396.	should	428.	thank	460.	those
397.	shouldn't	429.	thanks	461.	though
398.	since	430.	thanx	462.	three
399.	six	431.	that	463.	through
400.	so	432.	that's	464.	throughout
401.	some	433.	thats	465.	thru
402.	somebody	434.	the	466.	thus
403.	somehow	435.	their	467.	to
404.	someone	436.	theirs	468.	together
405.	something	437.	them	469.	too
406.	sometime	438.	themselves	470.	took
407.	sometimes	439.	then	471.	toward
408.	somewhat	440.	thence	472.	towards
409.	somewhere	441.	there	473.	tried
410.	soon	442.	there's	474.	tries
411.	sorry	443.	thereafter	475.	truly
412.	specified	444.	thereby	476.	try
413.	specify	445.	therefore	477.	trying
414.	specifying	446.	therein	478.	twice
415.	still	447.	theres	479.	two
416.	sub	448.	thereupon	480.	u

481.	un	513.	we'll	545.	why
482.	under	514.	we're	546.	will
483.	unfortunately	515.	we've	547.	willing
484.	unless	516.	welcome	548.	wish
485.	unlikely	517.	well	549.	with
486.	until	518.	went	550.	within
487.	unto	519.	were	551.	without
488.	up	520.	weren't	552.	won't
489.	upon	521.	what	553.	wonder
490.	us	522.	what's	554.	would
491.	use	523.	whatever	555.	would
492.	used	524.	when	556.	wouldn't
493.	useful	525.	whence	557.	x
494.	uses	526.	whenever	558.	y
495.	using	527.	where	559.	yes
496.	usually	528.	where's	560.	yet
497.	uucp	529.	whereafter	561.	you
498.	v	530.	whereas	562.	you'd
499.	value	531.	whereby	563.	you'll
500.	various	532.	wherein	564.	you're
501.	very	533.	whereupon	565.	you've
502.	via	534.	wherever	566.	your
503.	viz	535.	whether	567.	yours
504.	vs	536.	which	568.	yourself
505.	w	537.	while	569.	yourselves
506.	want	538.	whither	570.	z
507.	wants	539.	who	571.	zero
508.	was	540.	who's		
509.	wasn't	541.	whoever		
510.	way	542.	whole		
511.	we	543.	whom		
512.	we'd	544.	whose		

LAMPIRAN 2

Dokumen Uji

Dokumen 1

Alternative Fuel Vehicle refers to a vehicle that runs on a fuel other than traditional gasoline or diesel; any method of powering an engine that does not involve petroleum. Due to a combination of heavy taxes on fuel, particularly in Europe, tightening environmental laws, particularly in California, and the possibility of further restrictions on greenhouse gas emissions, work on alternative power systems for vehicles has become a high priority for governments and vehicle manufacturers around the world.

Current research and development is largely centered on "hybrid" vehicles that use both electric power and internal combustion. Other R&D efforts in alternative forms of power focus on developing fuel cells, alternative forms of combustion such as GDI and HCCI, and even the stored energy of compressed air. The use of alcohol as a fuel for internal combustion engines, either alone or in combination with other fuels, has been given much attention mostly because of its possible environmental and long-term economical advantages over fossil fuel. Both ethanol and methanol have been considered for this purpose. While both can be obtained from petroleum or natural gas, ethanol may be the most interesting because many believe it to be a renewable resource, easily obtained from sugar or starch in crops and other agricultural produce such as grain, sugarcane or even lactose. Since ethanol occurs in nature whenever yeast happens to find a sugar solution such as overripe fruit, most organisms have evolved some tolerance to ethanol, whereas methanol is toxic. Other experiments involve butanol, which can also be produced by fermentation of plants.

A hybrid vehicle uses multiple propulsion systems to provide motive power. This most commonly refers to gasoline-electric hybrid vehicles, which use gasoline (petrol) and electric batteries for the energy used to power internal-combustion engines and electric motors. These powerplants are usually relatively small and would be considered "underpowered" by themselves, but they can provide a

normal driving experience when used in combination during acceleration and other maneuvers that require greater power.
A hydrogen car is an automobile which uses hydrogen as its primary source of power for locomotion. These cars generally use the hydrogen in one of two methods: combustion or fuel-cell conversion. In combustion, the hydrogen is "burned" in engines in fundamentally the same method as traditional gasoline cars. In fuel-cell conversion, the hydrogen is turned into electricity through fuel cells which then powers electric motors. With either method, the only byproduct from the spent hydrogen is water. A small number of prototype hydrogen cars currently exist, and a significant amount of research is underway to make the technology more viable. A solar car is an electric vehicle powered by solar energy obtained from solar panels on the car. Solar cars are not a practical form of transportation; insufficient power falls on the roof of a practically sized and shaped vehicle to provide adequate performance.

Dokumen 2
Automakers around the world are planning to make cars that are smaller, use up less **fuel** and do not **damage** our **environment**. At auto shows around the **globe** car producers are presenting what they **have in mind**. General Motors has already showed one of its **prototypes.** It is called P.U.M.A. , which means Personal **Urban** Mobility and **Accessibility Vehicle**. It looks like a **cart**, has two batteries and seats for two people. While GM's car may never be produced other car makers have already made mini-cars that you can buy.
Mercedes' Smart car has been **popular** in Europe for some time. Chrysler's GEM Peapod and Toyota's IQ are two small cars that have been specially **designed** for city traffic. Man people want to buy small cars because they save fuel. That means saving money on petrol plus being able to find parking spaces more easily in crowded cities.
Carmakers are also spending money on **research** to make alternative-fuel cars. Maybe one day, most of us will drive biodiesel, **hydrogen**, or **solar-powered** cars. Hybrids are cars that are already on the market today. Toyota and GM already produce hybrid cars, trucks and **SUV's** that run on petrol and electricity. Other companies,

like BMW, have made cars that use hydrogen and electricity. **Although** such cars are still too expensive to produce in great numbers carmakers are continuing to improve them and make them cheaper.

Dokumen 3

A new study casts doubt on the merits of reducing salt in our diet. The researchers found that a modest lowering of blood pressure may be offset by other less desirable effects of a low-sodium diet. For years it's been part of accepted medical wisdom: reducing salt in your diet will lower your blood pressure, which will lower your risk of heart attack and stroke. But according to Danish researcher Niels Graudal of Copenhagen University Hospital, the effect of a reduced-salt diet is less dramatic than you might think. "We found that in normal persons with normal blood pressure, the effects on the blood pressure were surprisingly small. In patients with hypertension, the effect was somewhat bigger: the decrease was about 3.5 percent," he says. Graudal mathematically combined the results of 167 previous studies to come up with his results.

The people in the studies were largely European and North American, although Asians and blacks on low-sodium diets showed a somewhat larger reduction in their blood pressure. Even a modest reduction for people with high blood pressure is probably a good thing, but the study also found that people on a low-sodium diet had higher levels of cholesterol, which is associated with an increased risk of heart disease. Graudal says the body's natural salt-regulation system was also affected by a low-sodium diet. "And we saw that when you reduce the sodium intake, the hormone system was activated, which means that the body obviously felt that there was a danger that the amount of salt in the body could become too low." It might be more helpful to look directly for a link between salt consumption and heart disease or death, but Graudal says very few studies have tried to do that. "It would be very difficult to make a randomized, a big randomized study and keep people on special diets for many years," he explains. "Practically that would be very, very difficult." Graudal cautions that his study should not be interpreted as a license to eat as much salt as you want. He says it just confirms that a moderate amount of salt in a normal diet is probably not harmful,

and reducing salt intake has both positive and negative effects that might not help reduce the risk of heart attack and stroke.

Dokumen 4

Scientists say there is good news for the ozone layer in the atmosphere that protects the earth from harmful ultraviolet solar rays. Global efforts to halt the effects of ozone-depleting chemicals are working. The ozone hole isn't really a hole at all. It is a thinning of the protective ozone layer in the earth's atmosphere.

In 1986 Susan Solomon, senior scientist with the National Oceanic Atmospheric Administration, led an expedition to Antarctica to check out what British scientists had detected the year before. "There were a lot of doubts in the scientific community at that time that the ozone hole was even real, but certainly there is not doubt anymore that the ozone hole is a real phenomenon and that it covers essentially the entire Antarctic continent," she says, adding that is about "twice the size of the continental United States." Solomon's team put the blame on chlorofluorocarbons (or CFCs), industrially produced chemicals, which had been used since the 1930s as refrigerants and propellants in products like aerosol sprays. Solomon says these ozone-depleting chemicals do not cause global climate change, but do have adverse effects on human health and ecosystems. "If we have a thinner ozone layer, we have a increased risk of skin cancer, if we have a thinner ozone layer, we have an increased risk of cataracts," she says. "There are [also] questions about the kinds of biological effects that can result.

People in the Antarctic program in particular study things like whether changes in the ultraviolet [rays] due to the ozone hole can effect krill and other things in the southern oceans." The international community acted quickly to address these problems. By 1989 a United Nations treaty was in place to phase out CFCs. Ten years later, CFC production had dropped by 90 percent. David Hofman, who heads Global Monitoring for the National Oceanic and Atmospheric Administration, says the news is good. "The data indicates that the reduction in ozone has stopped. It has come down

and flattened out. It is not getting any worse. This is what has been called the first stage of ozone recovery."
Susan Solomon says one obstacle is to contain CFC's from sources not anticipated by the U.N. treaty such as used refrigerators, air conditioning units and insulated foams from landfills and demolition sites. "They are continuing to leak to the atmosphere, actually in levels somewhat higher than we would have thought," she says. Solomon expects a full recovery of the ozone hole by 2060. But, she cautions, a lot of work must be done to reach that goal. "I think that it is very important to make sure that we actually measure ozone not only not getting any worse, but actually starting to improve to make sure that the actions that we have taken internationally have been effective." CFCs are long-lived and remain in the atmosphere for 50 to 100 years. But with global phase-out efforts, Solomon expects to see signs of a reduction in the ozone hole within a decade. Her job, she says, is to measure that process.

Dokumen 5

Ozone is a kind of **oxygen** in which each molecule has three atoms instead of two. The formula is O3. Ozone is often produced when electricity passes through the air. That is why there is often an **unpleasant** smell after a **thunderstorm** or around electrical **equipment**.
Ozone is a blue gas that is **explosive** and **poisonous**. It is **denser** than **oxygen** and **condenses** into a dark blue **liquid** at - 112° C. This liquid freezes at – 251° C. Ozone is used in many industries. **Factories** use it for chemical reactions because it reacts more easily than **oxygen** does. Ozone also kills **germs** , which makes it useful for **removing** bad smells and **sterilizing** drinking water. Ozone is also used to **bleach** color out of other **substances**.
Ozone **occurs** in our atmosphere in two forms. Near the ground even small **amounts** of ozone can **cause** health problems. It **irritates** your eyes and can lead to coughing and **asthma**. Ozone is **especially** dangerous on clear days when **exhaust fumes** or cars pollute the air. Older people and babies are often told to stay **indoors** because ozone may **weaken** your **immune system.** About 30 -50 km above the

Earth's **surface** there is a layer of ozone in the atmosphere that **protects** us. It **absorbs** harmful **ultraviolet rays** from the sun. If too much of this **radiation** reaches the Earth it may **injure** your eyes and lead to **skin** cancer and other **diseases**.
In the 1970s **scientists** found out that chemicals **released** into the atmosphere have been **destroying** this **ozone layer**. The first hole in the ozone layer was found over Antarctica. In the last 20 years this hole has been getting bigger and now lies over some parts of Australia, New Zealand and the northern **hemisphere** as well. The hole in the ozone layer is **caused** by **CFCs**, chemicals often used in spray cans and **refrigerators**. They **escape** into the atmosphere and break up the ozone molecules.

Dokumen 6

For years doctors have been telling patients with high blood pressure to cut down on the amount of salt in their diet. Now a prestigious panel of experts from the Institute of Medicine in Washington wants the U.S. government to put limits on how much salt, or sodium, food manufacturers can use. It is hard to believe that many of us eat this much salt during one meal or even in one day. Doctors and nutritionists advise us not to eat more than a teaspoon of salt daily. That is about 2300 milligrams of sodium.
Salt makes food tastier. But too much of it can have risky consequences, says Michael Jacobson of the Center for Science in the Public Interest. "Salt is probably the single most harmful thing in our food, contributing to high blood pressure," Jacobson explained. "Which causes heart attacks and strokes." The World Health Organization reports an estimated 17 million people die of heart disease and stroke every year. One major risk factor is a diet filled with fat and salt.
Morton Satin of the Salt Institute says salt, or sodium, is not really the culprit. "The problem is not salt," he said. "The problem that we're dealing with is that we don't have a balanced diet." When the Food and Drug Administration does issue new limits on salt, the changes on the menu and on grocery food labels will take time to implement. "Reducing salt in the food supply needs to be a gradual process for a couple of reasons," Dr. Margaret Hamburg said. Hamburg is the commissioner of the FDA. One, we need consumer

tastes to adapt to a reduced level. Also, for industry, salt in the composition of food products, is something that cannot be changed overnight. They need to rework their recipes." Some food manufacturers have already begun the process of cutting back on sodium content. Meanwhile, the best advice may come from doctors who are advising patients to cut back on salt voluntarily.

Dokumen 7

Aging is one in every of those stunning phases of life when your persona shimmers with knowledge and experience. Deep wrinkles, fine lines and dark circles drop down the charm of your facial skin and outrage the sweetness of your charisma. Though, aging is irreversible but the seven signs of aging can positively be reversed. You can brighten out those unwanted wrinkles and get a firm, smooth and flawless skin with this advanced Wrinkle Cream. This can be an advanced wrinkle lifting answer to create your skin flourish with a swish gleam.

The moving on years generally sneak away your own skin's natural fats as well as wetness leaving the skin dull dried up and dull. Just like the simply results in of the plant whither in the absence associated with wet, your own skin color misplaces its activity and fresheners when it's lacking regarding moisture. Effortlessly its natural ingredients, anti wrinkle cream helps to reconstruct the actual extra gases of the human external body part skin color uncovering a lighten up vibrant glow on your own skin.

Anti aging Creams move to the layers of one's human external body part skin and provides strong bite coming via within which makes it plump and glow skin just similar to never before.The actual move regarding youth slowdown the particular albumin creation in your body which is required to stay skin organization and tight. This ends up inside a sagging skin having dreadful facial lines as well as collections. The actual unimaginable formula regarding this crease reluctant enhances the collagen production thereby increasing the firmness and also litheness of your skin.Owing to the fervid lifestyle as well as the actually growing pollution, premature skin aging is a frequent downside as well as provides additional be concerned outraces to beauty nut ladies. Skin spots, super skin discoloration,

dark musca volitans are the skin color difficulties that ladies have to expression of their early mid-thirty-something.

Anti aging Cream is really a superb response to all of your skin problems. It's a great awfully secure as well as economical natural healthy skin treatment answer that can make your own skin celestial stunning. This wonderful anti-aging cream can reverse your biological clock and cause you to look years younger. Get an ideal swish blemish free skin with this fascinating wrinkle reducing formula and let your friends flip green with envy while your husband remains guessing the secret to your youthful skin.

Dokumen 8

Women who drink a lot of caffeinated cola might be increasing their risk of heart attack and stroke while those who drink caffeinated coffee apparently do not. That's according to a new medical study published this week. There's good news for women who drink coffee: it does not increase the risk of high blood pressure, according to a new study in JAMA, the Journal of the American Medical Association. The study ran for 12 years and included more than 150,000 women.

Dr. Wolfgang Winkelmayer and his colleagues conducted the study expected to find drinking caffeinated coffee would produce a greater risk of high blood pressure. "If anything, coffee drinking was associated with a preventive effect, in that women who drank more coffee were less likely to have high blood pressure," said the doctor. Dr. Winkelmayer and his staff found that even women who drank six or more cups of coffee per day had no greater risk of high blood pressure. But, he says the same did not hold true for women who drank caffeinated colas. "We found that drinking soda beverages that contained caffeine, regular cola or diet cola, was associated with a greater risk of high blood pressure." Regular cola increased the risk by 44 percent for older women and by 28 percent for younger women. However, Dr. Winkelmayer says more research is needed to find out why caffeinated colas seem to increase blood pressure. "I would not jump to any conclusions at this point. I really believe that more research is necessary to solidify these findings," he said. But Dr. Winkelmayer says his findings about caffeinated coffee are clear.

The American Beverage Association released a statement saying other factors can cause high blood pressure: lifestyle, stress and other health conditions and that further study needs to be done to determine how important these other factors are with respect to consumption of soft drinks and high blood pressure.

Dokumen 9

The wisdom of centuries of regular medication, put together with the latest cutting-edge scientific and dermatologic research, is mixed in present-day major anti wrinkle pores and skin care items. Some formulas are powerful ample to generate effects which were formerly only achievable with plastic medical procedures. Other formulas merely you should not measure up. The usefulness of anti-wrinkle creams is dependent in component about the energetic ingredient or components.

Here are several popular elements which will end result in slight to modest advancement inside the visual appeal of wrinkles. Anti-oxidants are in excess of just the most up-to-date pattern on the subject of the ideal anti wrinkle cream. These vitamins are substances derived from all-natural foodstuff sources that defend our body against the ravages of totally free radicals. Free of charge radicals mutate skin cells and so are the final result of normal aging, smoking cigarettes, also significantly daylight or simply a bad diet plan. Vitamin A is in the category all of it's possess. When coupled with vitamins C and E in an anti-wrinkle cream it packs a triple punch of anti-oxidant motion.

In anti-aging creams, many other forms of vitamin A also are employed – retinol (pure vitamin A), retinyl palmitate (also identified as pro-retinol A or pro-vitamin A), retinyl acetate and retinyl linoleate. Views fluctuate concerning the effectiveness of every type. Some say retinol has much more powerful. These forms of Vitamin A make the skin stronger likewise as inspire the expansion of new collagen. Hydroxy acids, including alpha and beta hydroxy acids, and poly hydroxy acids, are artificial acids that imitate all those obtained from milk and specified fruits. Hydroxy acids are highly effective as exfoliants, accelerating cell alternative, which consequently final results in younger-looking pores and skin. Hydroxy acids are an extremely common ingredient in anti-wrinkle

creams, as they accelerate the production of collagen and boost skin hydration and smoothness. Vitamin Like Vitamin A, Vitamin C is an antioxidant that fights the totally free radicals that direct to pores and skin getting older and wrinkles. In addition, it increases the synthesis of collagen which allows to raise the skin's supporting framework. And lastly, it's got been proven to even pores and skin tone and support inside the fix of sunlight harmed pores and skin. Other elements normally found in anti wrinkle lotions include things like: kinetin, a plant hormone that can help the pores and skin retain moisture and smooth out wrinkles and unevenness; extract, a robust anti-inflammatory and antioxidant; and copper peptides that have the ability to regenerate tissue. Skinlastin is reviewed by famous skin experts and recognized as one of the best wrinkle cream. Skinlastin help you to avoid the sign of aging and Make Your Look more attractive by providing anti aging wrinkles solution.

LAMPIRAN 3

1. Ringkasan dengan ukuran ringkasan 25%
   a. Cluster 1

      Hasil sistem

      Due to a combination of heavy taxes on fuel, particularly in Europe, tightening environmental laws, particularly in California, and the possibility of further restrictions on greenhouse gas emissions, work on alternative power systems for vehicles has become a high priority for governments and vehicle manufacturers around the world. Solar cars are not a practical form of transportation; insufficient power falls on the roof of a practically sized and shaped vehicle to provide adequate performance. Other RD efforts in alternative forms of power focus on developing fuel cells, alternative forms of combustion such as GDI and HCCI and even the stored energy of compressed air. Mercedes' Smart car has been popular in Europe for some time. While both can be obtained from petroleum or natural gas, ethanol may be the most interesting because many believe it to be a renewable resource, easily obtained from sugar or starch in crops and other agricultural produce such as grain, sugarcane or even lactose. It is called PUMA which means Personal Urban Mobility and Accessibility Vehicle. Since ethanol occurs in nature whenever yeast happens to find a sugar solution such as overripe fruit, most organisms have evolved some tolerance to ethanol, whereas methanol is toxic. Chrysler's GEM Peapod and Toyota's IQ are two small cars that have been specially designed for city traffic. General Motors has already showed one of its prototypes.

      Hasil manusia

      Due to a combination of heavy taxes on fuel, particularly in Europe, tightening environmental laws, particularly in California, and the possibility of further restrictions on greenhouse gas emissions, work on alternative power systems for vehicles has become a high priority for

governments and vehicle manufacturers around the world.
Automakers around the world are planning to make cars that are smaller, use up less **fuel** and do not **damage** our **environment**.
It is called P.U.M.A. , which means Personal **Urban** Mobility and **Accessibility Vehicle.**
While GM's car may never be produced other car makers have already made mini-cars that you can buy.
Chrysler's GEM Peapod and Toyota's IQ are two small cars that have been specially designed for city traffic.
Man people want to buy small cars because they save fuel.
Other R&D efforts in alternative forms of power focus on developing fuel cells, alternative forms of combustion such as GDI and HCCI, and even the stored energy of compressed air.
Alternative Fuel Vehicle refers to a vehicle that runs on a fuel other than traditional gasoline or diesel; any method of powering an engine that does not involve petroleum.
Current research and development is largely centered on "hybrid" vehicles that use both electric power and internal combustion.

Jumlah *correct* dari ringkasan tersebut adalah 4, dengan *missed* 5 dan *wrong* 5. Nilai precision dan nilai recall dari ringkasan tersebut sama-sama 0.4444, sehingga nilai F-measurenya pun juga sama yaitu 0.4444.

b. Cluster 2
Hasil sistem

For years doctors have been telling patients with high blood pressure to cut down on the amount of salt in their diet.
Even a modest reduction for people with high blood pressure is probably a good thing, but the study also found that people on a low-sodium diet had higher levels of cholesterol, which is associated with an increased risk of heart disease.
"Salt is probably the single most harmful thing in our food, contributing to high blood pressure," Jacobson explained.

For years it's been part of accepted medical wisdom: reducing salt in your diet will lower your blood pressure, which will lower your risk of heart attack and stroke.

There's good news for women who drink coffee: it does not increase the risk of high blood pressure, according to a new study in JAMA the Journal of the American Medical Association.

The American Beverage Association released a statement saying other factors can cause high blood pressure: lifestyle, stress and other health conditions and that further study needs to be done to determine how important these other factors are with respect to consumption of soft drinks and high blood pressure.

"We found that drinking soda beverages that contained caffeine, regular cola or diet cola, was associated with a greater risk of high blood pressure. Dr Wolfgang Winkelmayer and his colleagues conducted the study expected to find drinking caffeinated coffee would produce a greater risk of high blood pressure. The people in the studies were largely European and North American, although Asians and blacks on low-sodium diets showed a somewhat larger reduction in their blood pressure. He says it just confirms that a moderate amount of salt in a normal diet is probably not harmful, and reducing salt intake has both positive and negative effects that might not help reduce the risk of heart attack and stroke. "Reducing salt in the food supply needs to be a gradual process for a couple of reasons," Dr Margaret Hamburg said. " When the Food and Drug Administration does issue new limits on salt, the changes on the menu and on grocery food labels will take time to implement. Now a prestigious panel of experts from the Institute of Medicine in Washington wants the US government to put limits on how much salt, or sodium, food manufacturers can use. " The World Health Organization reports an estimated 17 million people die of heart disease and stroke every year.

Hasil manusia

For years it's been part of accepted medical wisdom: reducing salt in your diet will lower your blood pressure, which will lower your risk of heart attack and stroke. "Salt is probably the single most harmful thing in our food, contributing to high blood pressure," Jacobson explained.

For years doctors have been telling patients with high blood pressure to cut down on the amount of salt in their diet.

There's good news for women who drink coffee: it does not increase the risk of high blood pressure, according to a new study in JAMA, the Journal of the American Medical Association. "We found that drinking soda beverages that contained caffeine, regular cola or diet cola, was associated with a greater risk of high blood pressure."

Dr. Winkelmayer and his staff found that even women who drank six or more cups of coffee per day had no greater risk of high blood pressure.

Women who drink a lot of caffeinated cola might be increasing their risk of heart attack and stroke while those who drink caffeinated coffee apparently do not.

The researchers found that a modest lowering of blood pressure may be offset by other less desirable effects of a low-sodium diet. Even a modest reduction for people with high blood pressure is probably a good thing, but the study also found that people on a low-sodium diet had higher levels of cholesterol, which is associated with an increased risk of heart disease.

He says it just confirms that a moderate amount of salt in a normal diet is probably not harmful, and reducing salt intake has both positive and negative effects that might not help reduce the risk of heart attack and stroke. "And we saw that when you reduce the sodium intake, the hormone system was activated, which means that the body obviously felt that there was a danger that the amount of salt in the body could become too low."

The American Beverage Association released a statement saying other factors can cause high blood pressure: lifestyle, stress and other health conditions and that further

study needs to be done to determine how important these other factors are with respect to consumption of soft drinks and high blood pressure.
Now a prestigious panel of experts from the Institute of Medicine in Washington wants the U.S. government to put limits on how much salt, or sodium, food manufacturers can use.

Jumlah *correct* dari ringkasan tersebut adalah 9, dengan *missed* 4 dan *wrong* 5. Sehingga di dapatkan nilai precision 0.6429, nilai recall 0.6923, dan nilai F-measure 0.6667.

c. Cluster 3
Hasil sistem

The hole in the ozone layer is caused by CFCs, chemicals often used in spray cans and refrigerators. The first hole in the ozone layer was found over Antarctica. In the 1970s scientists found out that chemicals released into the atmosphere have been destroying this ozone layer. About 30 -50 km above the Earth's surface there is a layer of ozone in the atmosphere that protects us. They escape into the atmosphere and break up the ozone molecules.
"There were a lot of doubts in the scientific community at that time that the ozone hole was even real, but certainly there is not doubt anymore that the ozone hole is a real phenomenon and that it covers essentially the entire Antarctic continent," she says, adding that is about "twice the size of the continental United States.
People in the Antarctic program in particular study things like whether changes in the ultraviolet [rays] due to the ozone hole can effect krill and other things in the southern oceans. "Solomon's team put the blame on chlorofluorocarbons (or CFCs), industrially produced chemicals, which had been used since the 1930s as refrigerants and propellants in products like aerosol sprays. In 1986 Susan Solomon, senior scientist with the National Oceanic Atmospheric Administration, led an expedition to Antarctica to check out what British scientists had detected

the year before. " Susan Solomon says one obstacle is to contain CFCs from sources not anticipated by the UN treaty such as used refrigerators, air conditioning units and insulated foams from landfills and demolition sites. By 1989 a United Nations treaty was in place to phase out CFCs. Ten years later, CFC production had dropped by 90 percent.

Hasil manusia

About 30 -50 km above the Earth's **surface** there is a layer of ozone in the atmosphere that **protects** us. It **absorbs** harmful **ultraviolet rays** from the sun. Ozone is a blue gas that is **explosive** and **poisonous**.

In the 1970s **scientists** found out that chemicals **released** into the atmosphere have been **destroying** this **ozone layer** . The first hole in the ozone layer was found over Antarctica.

The hole in the ozone layer is **caused** by **CFCs**, chemicals often used in spray cans and **refrigerators** .

CFCs are long-lived and remain in the atmosphere for 50 to 100 years.

"If we have a thinner ozone layer, we have a increased risk of skin cancer, if we have a thinner ozone layer, we have an increased risk of cataracts," she says.

Susan Solomon says one obstacle is to contain CFC's from sources not anticipated by the U.N. treaty such as used refrigerators, air conditioning units and insulated foams from landfills and demolition sites.

Solomon's team put the blame on chlorofluorocarbons (or CFCs), industrially produced chemicals, which had been used since the 1930s as refrigerants and propellants in products like aerosol sprays.

By 1989 a United Nations treaty was in place to phase out CFCs.

"There were a lot of doubts in the scientific community at that time that the ozone hole was even real, but certainly there is not doubt anymore that the ozone hole is a real phenomenon and that it covers essentially the entire

Antarctic continent," she says, adding that is about "twice the size of the continental United States."
Ozone also kills **germs** , which makes it useful for **removing** bad smells and **sterilizing** drinking water.

Jumlah *correct* dari ringkasan tersebut adalah 8, dengan *missed* 5 dan *wrong* 4. Dengan demikian didapatkan nilai precision 0.6667, nilai recall 0.6154, dan nilai F-measure 0.6400.

d. Cluster 4
Hasil sistem

Effortlessly its natural ingredients, anti wrinkle cream helps to reconstruct the actual extra gases of the human external body part skin color uncovering a lighten up vibrant glow on your own skin. Anti aging Creams move to the layers of one's human external body part skin and provides strong bite coming via within which makes it plump and glow skin just similar to never before. Hydroxy acids are an extremely common ingredient in anti-wrinkle creams, as they accelerate the production of collagen and boost skin hydration and smoothness. Get an ideal swish blemish free skin with this fascinating wrinkle reducing formula and let your friends flip green with envy while your husband remains guessing the secret to your youthful skin. Other elements normally found in anti wrinkle lotions include things like: kinetin, a plant hormone that can help the pores and skin retain moisture and smooth out wrinkles and unevenness; extract, a robust anti-inflammatory and antioxidant; and copper peptides that have the ability to regenerate tissue. The wisdom of centuries of regular medication, put together with the latest cutting-edge scientific and dermatologic research, is mixed in present-day major anti wrinkle pores and skin care items. Free of charge radicals mutate skin cells and so are the final result of normal aging, smoking cigarettes, also significantly daylight or simply a bad diet plan. Owing to the fervid lifestyle as well as the actually growing pollution, premature skin aging is a frequent downside as well as

provides additional be concerned outraces to beauty nut ladies. Hydroxy acids, including alpha and beta hydroxy acids, and poly hydroxy acids, are artificial acids that imitate all those obtained from milk and specified fruits. In anti-aging creams, many other forms of vitamin A also are employed : retinol (pure vitamin A retinyl palmitate (also identified as pro-retinol A or pro-vitamin A retinyl acetate and retinyl linoleate.

Hasil manusia

The wisdom of centuries of regular medication, put together with the latest cutting-edge scientific and dermatologic research, is mixed in present-day major anti wrinkle pores and skin care items. Anti aging Cream is really a superb response to all of your skin problems. Anti aging Creams move to the layers of one's human external body part skin and provides strong bite coming via within which makes it plump and glow skin just similar to never before. Anti-oxidants are in excess of just the most up-to-date pattern on the subject of the ideal anti wrinkle cream. You can brighten out those unwanted wrinkles and get a firm, smooth and flawless skin with this advanced Wrinkle Cream. Effortlessly its natural ingredients, anti wrinkle cream helps to reconstruct the actual extra gases of the human external body part skin color uncovering a lighten up vibrant glow on your own skin. Hydroxy acids are an extremely common ingredient in anti-wrinkle creams, as they accelerate the production of collagen and boost skin hydration and smoothness. Other elements normally found in anti wrinkle lotions include things like: kinetin, a plant hormone that can help the pores and skin retain moisture and smooth out wrinkles and unevenness; extract, a robust anti-inflammatory and antioxidant; and copper peptides that have the ability to regenerate tissue. In anti-aging creams, many other forms of vitamin A also are employed – retinol (pure vitamin A), retinyl palmitate (also identified as pro-retinol A or pro-vitamin A), retinyl acetate and retinyl linoleate. Vitamin Like Vitamin A, Vitamin C is an

antioxidant that fights the totally free radicals that direct to pores and skin getting older and wrinkles.

Jumlah *correct* dari ringkasan tersebut adalah 6, dengan *missed* dan *wrong* 4. Ringkasan dengan ukuran 25% untuk cluster 4 memiliki nilai precision, recall dan F-measure yang sama yakni 0.6000.

2. Ringkasan dengan ukuran ringkasan 50%
   a. Cluster 1
      Hasil sistem

Alternative Fuel Vehicle refers to a vehicle that runs on a fuel other than traditional gasoline or diesel; any method of powering an engine that does not involve petroleum. Man people want to buy small cars because they save fuel. This most commonly refers to gasoline-electric hybrid vehicles, which use gasoline (petrol) and electric batteries for the energy used to power internal-combustion engines and electric motors. Toyota and GM already produce hybrid cars, trucks and SUVs that run on petrol and electricity. Hybrids are cars that are already on the market today. Other companies, like BMW have made cars that use hydrogen and electricity. Due to a combination of heavy taxes on fuel, particularly in Europe, tightening environmental laws, particularly in California, and the possibility of further restrictions on greenhouse gas emissions, work on alternative power systems for vehicles has become a high priority for governments and vehicle manufacturers around the world. Solar cars are not a practical form of transportation; insufficient power falls on the roof of a practically sized and shaped vehicle to provide adequate performance. A small number of prototype hydrogen cars currently exist, and a significant amount of research is underway to make the technology more viable. Chrysler's GEM Peapod and Toyota's IQ are two small cars that have been specially designed for city traffic. The use of alcohol as a fuel for internal combustion engines, either alone or in combination with other fuels, has been given much attention mostly because of its possible environmental and

long-term economical advantages over fossil fuel. Mercedes' Smart car has been popular in Europe for some time. Other RD efforts in alternative forms of power focus on developing fuel cells, alternative forms of combustion such as GDI and HCCI and even the stored energy of compressed air. These powerplants are usually relatively small and would be considered "underpowered" by themselves, but they can provide a normal driving experience when used in combination during acceleration and other maneuvers that require greater power. It is called PUMA which means Personal Urban Mobility and Accessibility Vehicle. While both can be obtained from petroleum or natural gas, ethanol may be the most interesting because many believe it to be a renewable resource, easily obtained from sugar or starch in crops and other agricultural produce such as grain, sugarcane or even lactose. General Motors has already showed one of its prototypes. Since ethanol occurs in nature whenever yeast happens to find a sugar solution such as overripe fruit, most organisms have evolved some tolerance to ethanol, whereas methanol is toxic.

Hasil manusia

Due to a combination of heavy taxes on fuel, particularly in Europe, tightening environmental laws, particularly in California, and the possibility of further restrictions on greenhouse gas emissions, work on alternative power systems for vehicles has become a high priority for governments and vehicle manufacturers around the world.
Automakers around the world are planning to make cars that are smaller, use up less **fuel** and do not **damage** our **environment**. It is called P.U.M.A. , which means Personal **Urban** Mobility and **Accessibility Vehicle**.
While GM's car may never be produced other car makers have already made mini-cars that you can buy.
Chrysler's GEM Peapod and Toyota's IQ are two small cars that have been specially **designed** for city traffic.
Mercedes' Smart car has been **popular** in Europe for some

time.

Man people want to buy small cars because they save fuel.

Solar cars are not a practical form of transportation; insufficient power falls on the roof of a practically sized and shaped vehicle to provide adequate performance.

Alternative Fuel Vehicle refers to a vehicle that runs on a fuel other than traditional gasoline or diesel; any method of powering an engine that does not involve petroleum.

Current research and development is largely centered on "hybrid" vehicles that use both electric power and internal combustion. Other R&D efforts in alternative forms of power focus on developing fuel cells, alternative forms of combustion such as GDI and HCCI, and even the stored energy of compressed air.

Toyota and GM already produce hybrid cars, trucks and **SUV's** that run on petrol and electricity.

The use of alcohol as a fuel for internal combustion engines, either alone or in combination with other fuels, has been given much attention mostly because of its possible environmental and long-term economical advantages over fossil fuel. Since ethanol occurs in nature whenever yeast happens to find a sugar solution such as overripe fruit, most organisms have evolved some tolerance to ethanol, whereas methanol is toxic. Other experiments involve butanol, which can also be produced by fermentation of plants.

While both can be obtained from petroleum or natural gas, ethanol may be the most interesting because many believe it to be a renewable resource, easily obtained from sugar or starch in crops and other agricultural produce such as grain, sugarcane or even lactose.

These powerplants are usually relatively small and would be considered "underpowered" by themselves, but they can provide a normal driving experience when used in combination during acceleration and other maneuvers that require greater power.

Other companies, like BMW, have made cars that use hydrogen and electricity.

Jumlah *correct* dari ringkasan tersebut adalah 14, dengan *missed* 4 dan *wrong* 4. Dengan demikian didapatkan nilai precision, recall, dan F-masure yang sama yaitu 0.7778.

b. Cluster 2

Hasil sistem

For years doctors have been telling patients with high blood pressure to cut down on the amount of salt in their diet. For years it's been part of accepted medical wisdom: reducing salt in your diet will lower your blood pressure, which will lower your risk of heart attack and stroke. Even a modest reduction for people with high blood pressure is probably a good thing, but the study also found that people on a low-sodium diet had higher levels of cholesterol, which is associated with an increased risk of heart disease. There's good news for women who drink coffee: it does not increase the risk of high blood pressure, according to a new study in JAMA the Journal of the American Medical Association. "We found that drinking soda beverages that contained caffeine, regular cola or diet cola, was associated with a greater risk of high blood pressure. "Salt is probably the single most harmful thing in our food, contributing to high blood pressure," Jacobson explained. Dr Winkelmayer and his staff found that even women who drank six or more cups of coffee per day had no greater risk of high blood pressure. "If anything, coffee drinking was associated with a preventive effect, in that women who drank more coffee were less likely to have high blood pressure," said the doctor. Dr Wolfgang Winkelmayer and his colleagues conducted the study expected to find drinking caffeinated coffee would produce a greater risk of high blood pressure. The American Beverage Association released a statement saying other factors can cause high blood pressure: lifestyle, stress and other health conditions and that further study needs to be done to determine how important these other factors are with respect to consumption of soft drinks and high blood pressure. He says it just confirms that a moderate amount of salt in a normal diet is probably not

harmful, and reducing salt intake has both positive and negative effects that might not help reduce the risk of heart attack and stroke. The researchers found that a modest lowering of blood pressure may be offset by other less desirable effects of a low-sodium diet. The people in the studies were largely European and North American, although Asians and blacks on low-sodium diets showed a somewhat larger reduction in their blood pressure. One major risk factor is a diet filled with fat and salt. Women who drink a lot of caffeinated cola might be increasing their risk of heart attack and stroke while those who drink caffeinated coffee apparently do not. "Reducing salt in the food supply needs to be a gradual process for a couple of reasons," Dr Margaret Hamburg said. Also for industry, salt in the composition of food products, is something that cannot be changed overnight. " When the Food and Drug Administration does issue new limits on salt, the changes on the menu and on grocery food labels will take time to implement. Meanwhile, the best advice may come from doctors who are advising patients to cut back on salt voluntarily. Now a prestigious panel of experts from the Institute of Medicine in Washington wants the US government to put limits on how much salt, or sodium, food manufacturers can use. "It would be very difficult to make a randomized, a big randomized study and keep people on special diets for many years," he explains. "Regular cola increased the risk by 44 percent for older women and by 28 percent for younger women. "And we saw that when you reduce the sodium intake, the hormone system was activated, which means that the body obviously felt that there was a danger that the amount of salt in the body could become too low. " Some food manufacturers have already begun the process of cutting back on sodium content. But according to Danish researcher Niels Graudal of Copenhagen University Hospital, the effect of a reduced-salt diet is less dramatic than you might think. " The World Health Organization reports an estimated 17 million people die of heart disease and stroke every year. But too much of

it can have risky consequences, says Michael Jacobson of the Center for Science in the Public Interest.

Hasil manusia

For years it's been part of accepted medical wisdom: reducing salt in your diet will lower your blood pressure, which will lower your risk of heart attack and stroke. The World Health Organization reports an estimated 17 million people die of heart disease and stroke every year.

"Salt is probably the single most harmful thing in our food, contributing to high blood pressure," Jacobson explained. "We found that in normal persons with normal blood pressure, the effects on the blood pressure were surprisingly small.

For years doctors have been telling patients with high blood pressure to cut down on the amount of salt in their diet.

Doctors and nutritionists advise us not to eat more than a teaspoon of salt daily.

There's good news for women who drink coffee: it does not increase the risk of high blood pressure, according to a new study in JAMA, the Journal of the American Medical Association.

Dr. Wolfgang Winkelmayer and his colleagues conducted the study expected to find drinking caffeinated coffee would produce a greater risk of high blood pressure.

Dr. Winkelmayer and his staff found that even women who drank six or more cups of coffee per day had no greater risk of high blood pressure. But, he says the same did not hold true for women who drank caffeinated colas.

Regular cola increased the risk by 44 percent for older women and by 28 percent for younger women.

Women who drink a lot of caffeinated cola might be increasing their risk of heart attack and stroke while those who drink caffeinated coffee apparently do not.

"We found that drinking soda beverages that contained caffeine, regular cola or diet cola, was associated with a greater risk of high blood pressure."

The American Beverage Association released a statement

saying other factors can cause high blood pressure: lifestyle, stress and other health conditions and that further study needs to be done to determine how important these other factors are with respect to consumption of soft drinks and high blood pressure.

Now a prestigious panel of experts from the Institute of Medicine in Washington wants the U.S. government to put limits on how much salt, or sodium, food manufacturers can use. When the Food and Drug Administration does issue new limits on salt, the changes on the menu and on grocery food labels will take time to implement.

Some food manufacturers have already begun the process of cutting back on sodium content.

Also, for industry, salt in the composition of food products, is something that cannot be changed overnight. "Reducing salt in the food supply needs to be a gradual process for a couple of reasons," Dr. Margaret Hamburg said.

Meanwhile, the best advice may come from doctors who are advising patients to cut back on salt voluntarily.

The researchers found that a modest lowering of blood pressure may be offset by other less desirable effects of a low-sodium diet.

Morton Satin of the Salt Institute says salt, or sodium, is not really the culprit. The people in the studies were largely European and North American, although Asians and blacks on low-sodium diets showed a somewhat larger reduction in their blood pressure.

Even a modest reduction for people with high blood pressure is probably a good thing, but the study also found that people on a low-sodium diet had higher levels of cholesterol, which is associated with an increased risk of heart disease.Graudal cautions that his study should not be interpreted as a license to eat as much salt as you want. "And we saw that when you reduce the sodium intake, the hormone system was activated, which means that the body obviously felt that there was a danger that the amount of salt in the body could become too low."

He says it just confirms that a moderate amount of salt in a

normal diet is probably not harmful, and reducing salt intake has both positive and negative effects that might not help reduce the risk of heart attack and stroke.

Jumlah *correct* dari ringkasan tersebut adalah 22, dengan *missed* 5 dan *wrong* 5. Sehingga ringkasan tersebut memiliki nilai precision, recall, dan F-measure yang sama yakni 0.8148.

c. Cluster 3

Hasil sistem

The first hole in the ozone layer was found over Antarctica. In the 1970s scientists found out that chemicals released into the atmosphere have been destroying this ozone layer. The hole in the ozone layer is caused by CFCs , chemicals often used in spray cans and refrigerators. Ozone occurs in our atmosphere in two forms. They escape into the atmosphere and break up the ozone molecules. Scientists say there is good news for the ozone layer in the atmosphere that protects the earth from harmful ultraviolet solar rays. About 30 -50 km above the Earth's surface there is a layer of ozone in the atmosphere that protects us. "If we have a thinner ozone layer, we have a increased risk of skin cancer, if we have a thinner ozone layer, we have an increased risk of cataracts," she says. Ozone is a kind of oxygen in which each molecule has three atoms instead of two. But with global phase-out efforts, Solomon expects to see signs of a reduction in the ozone hole within a decade. "There were a lot of doubts in the scientific community at that time that the ozone hole was even real, but certainly there is not doubt anymore that the ozone hole is a real phenomenon and that it covers essentially the entire Antarctic continent," she says, adding that is about "twice the size of the continental United States. Ozone is a blue gas that is explosive and poisonous. The formula is O3 Ozone is often produced when electricity passes through the air. People in the Antarctic program in particular study things like whether changes in the ultraviolet [rays] due to the ozone hole can effect krill and other things in the

southern oceans. It is denser than oxygen and condenses into a dark blue liquid at - 112° C This liquid freezes at - 251° C Ozone is used in many industries. In 1986 Susan Solomon, senior scientist with the National Oceanic Atmospheric Administration, led an expedition to Antarctica to check out what British scientists had detected the year before. "CFCs are long-lived and remain in the atmosphere for 50 to 100 years. "I think that it is very important to make sure that we actually measure ozone not only not getting any worse, but actually starting to improve to make sure that the actions that we have taken internationally have been effective. David Hofman, who heads Global Monitoring for the National Oceanic and Atmospheric Administration, says the news is good. " Solomon's team put the blame on chlorofluorocarbons (or CFCs), industrially produced chemicals, which had been used since the 1930s as refrigerants and propellants in products like aerosol sprays. By 1989 a United Nations treaty was in place to phase out CFCs Ten years later, CFC production had dropped by 90 percent. "Susan Solomon says one obstacle is to contain CFCs from sources not anticipated by the UN treaty such as used refrigerators, air conditioning units and insulated foams from landfills and demolition sites. Solomon says these ozone-depleting chemicals do not cause global climate change, but do have adverse effects on human health and ecosystems.

Hasil manusia

About 30 -50 km above the Earth's **surface** there is a layer of ozone in the atmosphere that **protects** us. It **absorbs** harmful **ultraviolet rays** from the sun.
Ozone is a blue gas that is **explosive** and **poisonous**.
Ozone also kills **germs**, which makes it useful for **removing** bad smells and **sterilizing** drinking water.
Near the ground even small **amounts** of ozone can **cause** health problems.
Older people and babies are often told to stay **indoors** because ozone may **weaken** your **immune system**.

Factories use it for chemical reactions because it reacts more easily than **oxygen** does. Ozone is also used to **bleach** color out of other **substances**.
Ozone is **especially** dangerous on clear days when **exhaust fumes** or cars pollute the air.

In the 1970s **scientists** found out that chemicals **released** into the atmosphere have been **destroying** this **ozone layer**. The first hole in the ozone layer was found over Antarctica.

"If we have a thinner ozone layer, we have an increased risk of skin cancer, if we have a thinner ozone layer, we have an increased risk of cataracts," she says.

The hole in the ozone layer is **caused** by **CFCs**, chemicals often used in spray cans and **refrigerators**.
Susan Solomon says one obstacle is to contain CFC's from sources not anticipated by the U.N. treaty such as used refrigerators, air conditioning units and insulated foams from landfills and demolition sites.
Solomon's team put the blame on chlorofluorocarbons (or CFCs), industrially produced chemicals, which had been used since the 1930s as refrigerants and propellants in products like aerosol sprays.
CFCs are long-lived and remain in the atmosphere for 50 to 100 years. If too much of this **radiation** reaches the Earth it may **injure** your eyes and lead to **skin** cancer and other **diseases**.
By 1989 a United Nations treaty was in place to phase out CFCs.
Ten years later, CFC production had dropped by 90 percent.
People in the Antarctic program in particular study things like whether changes in the ultraviolet [rays] due to the ozone hole can effect krill and other things in the southern oceans."
"There were a lot of doubts in the scientific community at that time that the ozone hole was even real, but certainly

there is not doubt anymore that the ozone hole is a real phenomenon and that it covers essentially the entire Antarctic continent," she says, adding that is about "twice the size of the continental United States."
Solomon says these ozone-depleting chemicals do not cause global climate change, but do have adverse effects on human health and ecosystems.
But with global phase-out efforts, Solomon expects to see signs of a reduction in the ozone hole within a decade.
"I think that it is very important to make sure that we actually measure ozone not only not getting any worse, but actually starting to improve to make sure that the actions that we have taken internationally have been effective."
"The data indicates that the reduction in ozone has stopped.
Solomon expects a full recovery of the ozone hole by 2060.

Jumlah *correct* dari ringkasan tersebut adalah 16, dengan *missed* 10 dan *wrong* 8. Dari ringkasan tersebut diperoleh nilai F-measure 0.6400 dengan precision 0.6667 dan recall 0.6154.

d. Cluster 4
Hasil sistem

Effortlessly its natural ingredients, anti wrinkle cream helps to reconstruct the actual extra gases of the human external body part skin color uncovering a lighten up vibrant glow on your own skin. Skin spots, super skin discoloration, dark musca volitans are the skin color difficulties that ladies have to expression of their early mid-thirty-something. Anti aging Creams move to the layers of one's human external body part skin and provides strong bite coming via within which makes it plump and glow skin just similar to never before. And lastly, it's got been proven to even pores and skin tone and support inside the fix of sunlight harmed pores and skin. Get an ideal swish blemish free skin with this fascinating wrinkle reducing formula and let your friends flip green with envy while your husband remains guessing the secret to your youthful skin. Vitamin Like Vitamin A Vitamin C is an antioxidant that fights the totally free radicals that direct to

pores and skin getting older and wrinkles. Hydroxy acids are highly effective as exfoliants, accelerating cell alternative, which consequently final results in younger-looking pores and skin. The moving on years generally sneak away your own skin's natural fats as well as wetness leaving the skin dull dried up and dull. Other elements normally found in anti wrinkle lotions include things like: kinetin, a plant hormone that can help the pores and skin retain moisture and smooth out wrinkles and unevenness; extract, a robust anti-inflammatory and antioxidant; and copper peptides that have the ability to regenerate tissue. Hydroxy acids are an extremely common ingredient in anti-wrinkle creams, as they accelerate the production of collagen and boost skin hydration and smoothness. Free of charge radicals mutate skin cells and so are the final result of normal aging, smoking cigarettes, also significantly daylight or simply a bad diet plan. The wisdom of centuries of regular medication, put together with the latest cutting-edge scientific and dermatologic research, is mixed in present-day major anti wrinkle pores and skin care items. Just like the simply results in of the plant whither in the absence associated with wet, your own skin color misplaces its activity and fresheners when it's lacking regarding moisture. Deep wrinkles, fine lines and dark circles drop down the charm of your facial skin and outrage the sweetness of your charisma. The actual move regarding youth slowdown the particular albumin creation in your body which is required to stay skin organization and tight. Owing to the fervid lifestyle as well as the actually growing pollution, premature skin aging is a frequent downside as well as provides additional be concerned outraces to beauty nut ladies. Skinlastin help you to avoid the sign of aging and Make Your Look more attractive by providing anti aging wrinkles solution. Here are several popular elements which will end result in slight to modest advancement inside the visual appeal of wrinkles. These vitamins are substances derived from all-natural foodstuff sources that defend our body against the ravages of totally free radicals. Hydroxy acids, including alpha and beta

hydroxy acids, and poly hydroxy acids, are artificial acids that imitate all those obtained from milk and specified fruits. In anti-aging creams, many other forms of vitamin A also are employed : retinol (pure vitamin A retinyl palmitate (also identified as pro-retinol A or pro-vitamin A retinyl acetate and retinyl linoleate.

Hasil manusia

Free of charge radicals mutate skin cells and so are the final result of normal aging, smoking cigarettes, also significantly daylight or simply a bad diet plan.

The wisdom of centuries of regular medication, put together with the latest cutting-edge scientific and dermatologic research, is mixed in present-day major anti wrinkle pores and skin care items. Anti aging Cream is really a superb response to all of your skin problems.

Anti aging Creams move to the layers of one's human external body part skin and provides strong bite coming via within which makes it plump and glow skin just similar to never before.

The actual move regarding youth slowdown the particular albumin creation in your body which is required to stay skin organization and tight.

It's a great awfully secure as well as economical natural healthy skin treatment answer that can make your own skin celestial stunning.

Anti-oxidants are in excess of just the most up-to-date pattern on the subject of the ideal anti wrinkle cream.

Effortlessly its natural ingredients, anti wrinkle cream helps to reconstruct the actual extra gases of the human external body part skin color uncovering a lighten up vibrant glow on your own skin. Hydroxy acids are an extremely common ingredient in anti-wrinkle creams, as they accelerate the production of collagen and boost skin hydration and smoothness.

Other elements normally found in anti wrinkle lotions include things like: kinetin, a plant hormone that can help the pores and skin retain moisture and smooth out wrinkles and

unevenness; extract, a robust anti-inflammatory and antioxidant; and copper peptides that have the ability to regenerate tissue.

In anti-aging creams, many other forms of vitamin A also are employed – retinol (pure vitamin A), retinyl palmitate (also identified as pro-retinol A or pro-vitamin A), retinyl acetate and retinyl linoleate.

Vitamin Like Vitamin A, Vitamin C is an antioxidant that fights the totally free radicals that direct to pores and skin getting older and wrinkles.

When coupled with vitamins C and E in an anti-wrinkle cream it packs a triple punch of anti-oxidant motion.

These forms of Vitamin A make the skin stronger likewise as inspire the expansion of new collagen.

Hydroxy acids, including alpha and beta hydroxy acids, and poly hydroxy acids, are artificial acids that imitate all those obtained from milk and specified fruits.

Hydroxy acids are highly effective as exfoliants, accelerating cell alternative, which consequently final results in younger-looking pores and skin.

In addition, it increases the synthesis of collagen which allows to raise the skin's supporting framework.

The actual unimaginable formula regarding this crease reluctant enhances the collagen production thereby increasing the firmness and also litheness of your skin.

This wonderful anti-aging cream can reverse your biological clock and cause you to look years younger.

This can be an advanced wrinkle lifting answer to create your skin flourish with a swish gleam.

You can brighten out those unwanted wrinkles and get a firm, smooth and flawless skin with this advanced Wrinkle Cream.

Jumlah *correct* dari ringkasan tersebut adalah 11, dengan *missed* 10 dan *wrong* 10. Didapatkan nilai precision 0.5238, recall 0.5238 dan F-measure yang sama yaitu 0.5238.

3. Ringkasan dengan ukuran ringkasan 75%
   a. Cluster 1
      Hasil sistem

A solar car is an electric vehicle powered by solar energy obtained from solar panels on the car. While GMs car may never be produced other car makers have already made mini-cars that you can buy. Toyota and GM already produce hybrid cars, trucks and SUVs that run on petrol and electricity. Other companies, like BMW have made cars that use hydrogen and electricity. Alternative Fuel Vehicle refers to a vehicle that runs on a fuel other than traditional gasoline or diesel; any method of powering an engine that does not involve petroleum. Man people want to buy small cars because they save fuel. This most commonly refers to gasoline-electric hybrid vehicles, which use gasoline (petrol) and electric batteries for the energy used to power internal-combustion engines and electric motors. Hybrids are cars that are already on the market today. Current research and development is largely centered on "hybrid" vehicles that use both electric power and internal combustion. In fuel-cell conversion, the hydrogen is turned into electricity through fuel cells which then powers electric motors. In combustion, the hydrogen is "burned" in engines in fundamentally the same method as traditional gasoline cars. Solar cars are not a practical form of transportation; insufficient power falls on the roof of a practically sized and shaped vehicle to provide adequate performance. Automakers around the world are planning to make cars that are smaller, use up less fuel and do not damage our environment. A small number of prototype hydrogen cars currently exist, and a significant amount of research is underway to make the technology more viable. At auto shows around the globe car producers are presenting what they have in mind. Carmakers are also spending money on research to make alternative-fuel cars. A hybrid vehicle uses multiple propulsion systems to provide motive power. Due to a combination of heavy taxes on fuel, particularly in Europe, tightening environmental laws, particularly in California, and the possibility of further

restrictions on greenhouse gas emissions, work on alternative power systems for vehicles has become a high priority for governments and vehicle manufacturers around the world. Mercedes' Smart car has been popular in Europe for some time. Chrysler's GEM Peapod and Toyota's IQ are two small cars that have been specially designed for city traffic. The use of alcohol as a fuel for internal combustion engines, either alone or in combination with other fuels, has been given much attention mostly because of its possible environmental and long-term economical advantages over fossil fuel. Other RD efforts in alternative forms of power focus on developing fuel cells, alternative forms of combustion such as GDI and HCCI and even the stored energy of compressed air. These powerplants are usually relatively small and would be considered "underpowered" by themselves, but they can provide a normal driving experience when used in combination during acceleration and other maneuvers that require greater power. It is called PUMA which means Personal Urban Mobility and Accessibility Vehicle. While both can be obtained from petroleum or natural gas, ethanol may be the most interesting because many believe it to be a renewable resource, easily obtained from sugar or starch in crops and other agricultural produce such as grain, sugarcane or even lactose. General Motors has already showed one of its prototypes. Since ethanol occurs in nature whenever yeast happens to find a sugar solution such as overripe fruit, most organisms have evolved some tolerance to ethanol, whereas methanol is toxic.

Hasil manusia

Due to a combination of heavy taxes on fuel, particularly in Europe, tightening environmental laws, particularly in California, and the possibility of further restrictions on greenhouse gas emissions, work on alternative power systems for vehicles has become a high priority for governments and vehicle manufacturers around the world.
Solar cars are not a practical form of transportation;

insufficient power falls on the roof of a practically sized and shaped vehicle to provide adequate performance.
A solar car is an electric vehicle powered by solar energy obtained from solar panels on the car.
Automakers around the world are planning to make cars that are smaller, use up less **fuel** and do not **damage** our **environment**.
It is called P.U.M.A. , which means Personal **Urban** Mobility and **Accessibility Vehicle**.
While GM's car may never be produced other car makers have already made mini-cars that you can buy.
Chrysler's GEM Peapod and Toyota's IQ are two small cars that have been specially **designed** for city traffic.
Mercedes' Smart car has been **popular** in Europe for some time.
Man people want to buy small cars because they save fuel.
That means saving money on petrol plus being able to find parking spaces more easily in crowded cities.
Carmakers are also spending money on **research** to make alternative-fuel cars.
Alternative Fuel Vehicle refers to a vehicle that runs on a fuel other than traditional gasoline or diesel; any method of powering an engine that does not involve petroleum.
Other R&D efforts in alternative forms of power focus on developing fuel cells, alternative forms of combustion such as GDI and HCCI, and even the stored energy of compressed air.
In fuel-cell conversion, the hydrogen is turned into electricity through fuel cells which then powers electric motors.
A hydrogen car is an automobile which uses hydrogen as its primary source of power for locomotion.
Current research and development is largely centered on "hybrid" vehicles that use both electric power and internal combustion.
This most commonly refers to gasoline-electric hybrid vehicles, which use gasoline (petrol) and electric batteries for the energy used to power internal-combustion engines and electric motors.

The use of alcohol as a fuel for internal combustion engines, either alone or in combination with other fuels, has been given much attention mostly because of its possible environmental and long-term economical advantages over fossil fuel.
Since ethanol occurs in nature whenever yeast happens to find a sugar solution such as overripe fruit, most organisms have evolved some tolerance to ethanol, whereas methanol is toxic.
Other experiments involve butanol, which can also be produced by fermentation of plants.
These powerplants are usually relatively small and would be considered "underpowered" by themselves, but they can provide a normal driving experience when used in combination during acceleration and other maneuvers that require greater power.
Both ethanol and methanol have been considered for this purpose.
While both can be obtained from petroleum or natural gas, ethanol may be the most interesting because many believe it to be a renewable resource, easily obtained from sugar or starch in crops and other agricultural produce such as grain, sugarcane or even lactose.
Toyota and GM already produce hybrid cars, trucks and **SUV's** that run on petrol and electricity.
Other companies, like BMW, have made cars that use hydrogen and electricity.
Although such cars are still too expensive to produce in great numbers carmakers are continuing to improve them and make them cheaper.
Maybe one day, most of us will drive biodiesel, **hydrogen**, or **solar-powered** cars.

Jumlah *correct* dari ringkasan tersebut adalah 21, dengan *missed* 6 dan *wrong* 6. Untuk cluster 1 dengan ukuran ringkasan 75% didapatkan nilai precision, recall, dan F-measure yang sama yakni.

b. Cluster 2

Hasil sistem

For years doctors have been telling patients with high blood pressure to cut down on the amount of salt in their diet. For years it's been part of accepted medical wisdom: reducing salt in your diet will lower your blood pressure, which will lower your risk of heart attack and stroke. "We found that drinking soda beverages that contained caffeine, regular cola or diet cola, was associated with a greater risk of high blood pressure. Even a modest reduction for people with high blood pressure is probably a good thing, but the study also found that people on a low-sodium diet had higher levels of cholesterol, which is associated with an increased risk of heart disease. There's good news for women who drink coffee: it does not increase the risk of high blood pressure, according to a new study in JAMA the Journal of the American Medical Association. "Salt is probably the single most harmful thing in our food, contributing to high blood pressure," Jacobson explained. "If anything, coffee drinking was associated with a preventive effect, in that women who drank more coffee were less likely to have high blood pressure," said the doctor. Dr Winkelmayer and his staff found that even women who drank six or more cups of coffee per day had no greater risk of high blood pressure. Dr Wolfgang Winkelmayer and his colleagues conducted the study expected to find drinking caffeinated coffee would produce a greater risk of high blood pressure. He says it just confirms that a moderate amount of salt in a normal diet is probably not harmful, and reducing salt intake has both positive and negative effects that might not help reduce the risk of heart attack and stroke. The American Beverage Association released a statement saying other factors can cause high blood pressure: lifestyle, stress and other health conditions and that further study needs to be done to determine how important these other factors are with respect to consumption of soft drinks and high blood pressure. The researchers found that a modest lowering of blood pressure may be offset by other less desirable effects of a low-sodium

diet. However, Dr Winkelmayer says more research is needed to find out why caffeinated colas seem to increase blood pressure. "We found that in normal persons with normal blood pressure, the effects on the blood pressure were surprisingly small. The people in the studies were largely European and North American, although Asians and blacks on low-sodium diets showed a somewhat larger reduction in their blood pressure. One major risk factor is a diet filled with fat and salt. Women who drink a lot of caffeinated cola might be increasing their risk of heart attack and stroke while those who drink caffeinated coffee apparently do not. Morton Satin of the Salt Institute says salt, or sodium, is not really the culprit. "Reducing salt in the food supply needs to be a gradual process for a couple of reasons," Dr Margaret Hamburg said. " It might be more helpful to look directly for a link between salt consumption and heart disease or death, but Graudal says very few studies have tried to do that. Meanwhile, the best advice may come from doctors who are advising patients to cut back on salt voluntarily. " When the Food and Drug Administration does issue new limits on salt, the changes on the menu and on grocery food labels will take time to implement. Also for industry, salt in the composition of food products, is something that cannot be changed overnight. Now a prestigious panel of experts from the Institute of Medicine in Washington wants the US government to put limits on how much salt, or sodium, food manufacturers can use. " Regular cola increased the risk by 44 percent for older women and by 28 percent for younger women. Doctors and nutritionists advise us not to eat more than a teaspoon of salt daily. "It would be very difficult to make a randomized, a big randomized study and keep people on special diets for many years," he explains. But he says the same did not hold true for women who drank caffeinated colas. "And we saw that when you reduce the sodium intake, the hormone system was activated, which means that the body obviously felt that there was a danger that the amount of salt in the body could become too low. The study ran for 12 years and included

more than 150,000 women. But according to Danish researcher Niels Graudal of Copenhagen University Hospital, the effect of a reduced-salt diet is less dramatic than you might think. " Some food manufacturers have already begun the process of cutting back on sodium content. " The World Health Organization reports an estimated 17 million people die of heart disease and stroke every year. "The problem that we're dealing with is that we don't have a balanced diet. Hamburg is the commissioner of the FDA (Food and Drug Administration). In patients with hypertension, the effect was somewhat bigger: the decrease was about 3,5 percent," he says. Graudal mathematically combined the results of 167 previous studies to come up with his results. One we need consumer tastes to adapt to a reduced level. But too much of it can have risky consequences, says Michael Jacobson of the Center for Science in the Public Interest.

Hasil manusia

For years it's been part of accepted medical wisdom: reducing salt in your diet will lower your blood pressure, which will lower your risk of heart attack and stroke.
"Salt is probably the single most harmful thing in our food, contributing to high blood pressure," Jacobson explained.
For years doctors have been telling patients with high blood pressure to cut down on the amount of salt in their diet.
Doctors and nutritionists advise us not to eat more than a teaspoon of salt daily.
The World Health Organization reports an estimated 17 million people die of heart disease and stroke every year.
The people in the studies were largely European and North American, although Asians and blacks on low-sodium diets showed a somewhat larger reduction in their blood pressure.
Now a prestigious panel of experts from the Institute of Medicine in Washington wants the U.S. government to put limits on how much salt, or sodium, food manufacturers can use.
When the Food and Drug Administration does issue new

limits on salt, the changes on the menu and on grocery food labels will take time to implement.
Some food manufacturers have already begun the process of cutting back on sodium content.
"Reducing salt in the food supply needs to be a gradual process for a couple of reasons," Dr. Margaret Hamburg said.
One, we need consumer tastes to adapt to a reduced level.
Also, for industry, salt in the composition of food products, is something that cannot be changed overnight.
They need to rework their recipes."
Meanwhile, the best advice may come from doctors who are advising patients to cut back on salt voluntarily.
There's good news for women who drink coffee: it does not increase the risk of high blood pressure, according to a new study in JAMA, the Journal of the American Medical Association.
The study ran for 12 years and included more than 150,000 women.
Women who drink a lot of caffeinated cola might be increasing their risk of heart attack and stroke while those who drink caffeinated coffee apparently do not.
Dr. Wolfgang Winkelmayer and his colleagues conducted the study expected to find drinking caffeinated coffee would produce a greater risk of high blood pressure.
"If anything, coffee drinking was associated with a preventive effect, in that women who drank more coffee were less likely to have high blood pressure," said the doctor.
Dr. Winkelmayer and his staff found that even women who drank six or more cups of coffee per day had no greater risk of high blood pressure.
But Dr. Winkelmayer says his findings about caffeinated coffee are clear.
"We found that drinking soda beverages that contained caffeine, regular cola or diet cola, was associated with a greater risk of high blood pressure."
But, he says the same did not hold true for women who drank caffeinated colas.

Regular cola increased the risk by 44 percent for older women and by 28 percent for younger women.
However, Dr. Winkelmayer says more research is needed to find out why caffeinated colas seem to increase blood pressure.
But according to Danish researcher Niels Graudal of Copenhagen University Hospital, the effect of a reduced-salt diet is less dramatic than you might think.
"We found that in normal persons with normal blood pressure, the effects on the blood pressure were surprisingly small.
In patients with hypertension, the effect was somewhat bigger: the decrease was about 3.5 percent," he says.
Graudal mathematically combined the results of 167 previous studies to come up with his results.
Even a modest reduction for people with high blood pressure is probably a good thing, but the study also found that people on a low-sodium diet had higher levels of cholesterol, which is associated with an increased risk of heart disease.
Graudal says the body's natural salt-regulation system was also affected by a low-sodium diet.
"And we saw that when you reduce the sodium intake, the hormone system was activated, which means that the body obviously felt that there was a danger that the amount of salt in the body could become too low."
Graudal cautions that his study should not be interpreted as a license to eat as much salt as you want.
He says it just confirms that a moderate amount of salt in a normal diet is probably not harmful, and reducing salt intake has both positive and negative effects that might not help reduce the risk of heart attack and stroke.
The American Beverage Association released a statement saying other factors can cause high blood pressure: lifestyle, stress and other health conditions and that further study needs to be done to determine how important these other factors are with respect to consumption of soft drinks and high blood pressure.
The researchers found that a modest lowering of blood

pressure may be offset by other less desirable effects of a low-sodium diet.
Morton Satin of the Salt Institute says salt, or sodium, is not really the culprit.
But too much of it can have risky consequences, says Michael Jacobson of the Center for Science in the Public Interest.
"The problem that we're dealing with is that we don't have a balanced diet."
It might be more helpful to look directly for a link between salt consumption and heart disease or death, but Graudal says very few studies have tried to do that.

Jumlah *correct* dari ringkasan tersebut adalah 36, dengan *missed* 4 dan *wrong* 2. Nilai precision ringkasan tersebut yaitu, dengan nilai recall dan nilai F-measure.

c. Cluster 3
Hasil sistem

It is a thinning of the protective ozone layer in the earth's atmosphere. Ozone occurs in our atmosphere in two forms. The first hole in the ozone layer was found over Antarctica. In the 1970s scientists found out that chemicals released into the atmosphere have been destroying this ozone layer. The hole in the ozone layer is caused by CFCs , chemicals often used in spray cans and refrigerators. They escape into the atmosphere and break up the ozone molecules. About 30 -50 km above the Earth’s surface there is a layer of ozone in the atmosphere that protects us. Scientists say there is good news for the ozone layer in the atmosphere that protects the earth from harmful ultraviolet solar rays. Solomon expects a full recovery of the ozone hole by 2060. Ozone is a kind of oxygen in which each molecule has three atoms instead of two. "If we have a thinner ozone layer, we have a increased risk of skin cancer, if we have a thinner ozone layer, we have an increased risk of cataracts," she says. But with global phase-out efforts, Solomon expects to see signs of a reduction in the ozone hole within a decade. Ozone is also

used to bleach color out of other substances. "There were a lot of doubts in the scientific community at that time that the ozone hole was even real, but certainly there is not doubt anymore that the ozone hole is a real phenomenon and that it covers essentially the entire Antarctic continent," she says, adding that is about "twice the size of the continental United States. Ozone is a blue gas that is explosive and poisonous. Near the ground even small amounts of ozone can cause health problems. The formula is O3 Ozone is often produced when electricity passes through the air. People in the Antarctic program in particular study things like whether changes in the ultraviolet [rays] due to the ozone hole can effect krill and other things in the southern oceans. Ozone is especially dangerous on clear days when exhaust fumes or cars pollute the air. "I think that it is very important to make sure that we actually measure ozone not only not getting any worse, but actually starting to improve to make sure that the actions that we have taken internationally have been effective. Ozone also kills germs , which makes it useful for removing bad smells and sterilizing drinking water. Older people and babies are often told to stay indoors because ozone may weaken your immune system. It is denser than oxygen and condenses into a dark blue liquid at - 112° C This liquid freezes at - 251° C Ozone is used in many industries. " CFCs are long-lived and remain in the atmosphere for 50 to 100 years. In 1986 Susan Solomon, senior scientist with the National Oceanic Atmospheric Administration, led an expedition to Antarctica to check out what British scientists had detected the year before. Solomon says these ozone-depleting chemicals do not cause global climate change, but do have adverse effects on human health and ecosystems. Global efforts to halt the effects of ozone-depleting chemicals are working. David Hofman, who heads Global Monitoring for the National Oceanic and Atmospheric Administration, says the news is good. " Solomon's team put the blame on chlorofluorocarbons (or CFCs), industrially produced chemicals, which had been used since the 1930s as refrigerants and propellants in

products like aerosol sprays. "They are continuing to leak to the atmosphere, actually in levels somewhat higher than we would have thought," she says. " Susan Solomon says one obstacle is to contain CFCs from sources not anticipated by the UN treaty such as used refrigerators, air conditioning units and insulated foams from landfills and demolition sites. By 1989 a United Nations treaty was in place to phase out CFCs Ten years later, CFC production had dropped by 90 percent. Factories use it for chemical reactions because it reacts more easily than oxygen does. "There are [also] questions about the kinds of biological effects that can result. But she cautions, a lot of work must be done to reach that goal. " The international community acted quickly to address these problems.

Hasil manusia

Scientists say there is good news for the ozone layer in the atmosphere that protects the earth from harmful ultraviolet solar rays.
About 30 -50 km above the Earth's **surface** there is a layer of ozone in the atmosphere that **protects** us.
It **absorbs** harmful **ultraviolet rays** from the sun.
Ozone is a blue gas that is **explosive** and **poisonous**.
Ozone is a kind of **oxygen** in which each molecule has three atoms instead of two.
Ozone is often produced when electricity passes through the air.
It is **denser** than **oxygen** and **condenses** into a dark blue **liquid** at - 112° C.
Ozone is **especially** dangerous on clear days when **exhaust fumes** or cars pollute the air.
Ozone is used in many industries.
Factories use it for chemical reactions because it reacts more easily than **oxygen** does.
Ozone is also used to **bleach** color out of other **substances**.
In the 1970s **scientists** found out that chemicals **released** into the atmosphere have been **destroying** this **ozone layer**.
The first hole in the ozone layer was found over Antarctica.

The hole in the ozone layer is **caused** by **CFCs** , chemicals often used in spray cans and **refrigerators**.
CFCs are long-lived and remain in the atmosphere for 50 to 100 years.
In the last 20 years this hole has been getting bigger and now lies over some parts of Australia, New Zealand and the northern **hemisphere** as well.
Solomon's team put the blame on chlorofluorocarbons (or CFCs), industrially produced chemicals, which had been used since the 1930s as refrigerants and propellants in products like aerosol sprays.
Susan Solomon says one obstacle is to contain CFC's from sources not anticipated by the U.N. treaty such as used refrigerators, air conditioning units and insulated foams from landfills and demolition sites.
"They are continuing to leak to the atmosphere, actually in levels somewhat higher than we would have thought," she says.
In 1986 Susan Solomon, senior scientist with the National Oceanic Atmospheric Administration, led an expedition to Antarctica to check out what British scientists had detected the year before.
People in the Antarctic program in particular study things like whether changes in the ultraviolet [rays] due to the ozone hole can effect krill and other things in the southern oceans."
"There were a lot of doubts in the scientific community at that time that the ozone hole was even real, but certainly there is not doubt anymore that the ozone hole is a real phenomenon and that it covers essentially the entire Antarctic continent," she says, adding that is about "twice the size of the continental United States."
The ozone hole isn't really a hole at all.
It is a thinning of the protective ozone layer in the earth's atmosphere.
"If we have a thinner ozone layer, we have a increased risk of skin cancer, if we have a thinner ozone layer, we have an increased risk of cataracts," she says.

Ozone also kills **germs**, which makes it useful for **removing** bad smells and **sterilizing** drinking water.
If too much of this **radiation** reaches the Earth it may **injure** your eyes and lead to **skin** cancer and other **diseases** .
Near the ground even small **amounts** of ozone can **cause** health problems.
It **irritates** your eyes and can lead to coughing and **asthma**.
Older people and babies are often told to stay **indoors** because ozone may **weaken** your **immune system** .
By 1989 a United Nations treaty was in place to phase out CFCs.
Solomon says these ozone-depleting chemicals do not cause global climate change, but do have adverse effects on human health and ecosystems.
But with global phase-out efforts, Solomon expects to see signs of a reduction in the ozone hole within a decade.
"I think that it is very important to make sure that we actually measure ozone not only not getting any worse, but actually starting to improve to make sure that the actions that we have taken internationally have been effective."
Solomon expects a full recovery of the ozone hole by 2060.
David Hofman, who heads Global Monitoring for the National Oceanic and Atmospheric Administration, says the news is good.
Global efforts to halt the effects of ozone-depleting chemicals are working.
"The data indicates that the reduction in ozone has stopped.
Ten years later, CFC production had dropped by 90 percent.

Jumlah *correct* dari ringkasan tersebut adalah 33, dengan *missed* 6 dan *wrong* 7. Dengan demikian untuk cluster 3 ukuran ringkasan 75% diperoleh nilai precision, nilai recall sehingga didapat nilai F-measure yaitu

d. Cluster 4
Hasil sistem

Effortlessly its natural ingredients, anti wrinkle cream helps to reconstruct the actual extra gases of the human external

body part skin color uncovering a lighten up vibrant glow on your own skin. Anti aging Creams move to the layers of one's human external body part skin and provides strong bite coming via within which makes it plump and glow skin just similar to never before. Skin spots, super skin discoloration, dark musca volitans are the skin color difficulties that ladies have to expression of their early mid-thirty-something. You can brighten out those unwanted wrinkles and get a firm, smooth and flawless skin with this advanced Wrinkle Cream. Get an ideal swish blemish free skin with this fascinating wrinkle reducing formula and let your friends flip green with envy while your husband remains guessing the secret to your youthful skin. Its a great awfully secure as well as economical natural healthy skin treatment answer that can make your own skin celestial stunning. And lastly, it's got been proven to even pores and skin tone and support inside the fix of sunlight harmed pores and skin. Vitamin Like Vitamin A Vitamin C is an antioxidant that fights the totally free radicals that direct to pores and skin getting older and wrinkles. The moving on years generally sneak away your own skin's natural fats as well as wetness leaving the skin dull dried up and dull. This can be an advanced wrinkle lifting answer to create your skin flourish with a swish gleam. These forms of Vitamin A make the skin stronger likewise as inspire the expansion of new collagen. Hydroxy acids are an extremely common ingredient in anti-wrinkle creams, as they accelerate the production of collagen and boost skin hydration and smoothness. Other elements normally found in anti wrinkle lotions include things like: kinetin, a plant hormone that can help the pores and skin retain moisture and smooth out wrinkles and unevenness; extract, a robust anti-inflammatory and antioxidant; and copper peptides that have the ability to regenerate tissue. Hydroxy acids are highly effective as exfoliants, accelerating cell alternative, which consequently final results in younger-looking pores and skin. In addition, it increases the synthesis of collagen which allows to raise the skin's supporting framework. The wisdom of centuries of regular medication,

put together with the latest cutting-edge scientific and dermatologic research, is mixed in present-day major anti wrinkle pores and skin care items. The actual unimaginable formula regarding this crease reluctant enhances the collagen production thereby increasing the firmness and also litheness of your skin. Deep wrinkles, fine lines and dark circles drop down the charm of your facial skin and outrage the sweetness of your charisma. Free of charge radicals mutate skin cells and so are the final result of normal aging, smoking cigarettes, also significantly daylight or simply a bad diet plan. Just like the simply results in of the plant whither in the absence associated with wet, your own skin color misplaces its activity and fresheners when it's lacking regarding moisture. The actual move regarding youth slowdown the particular albumin creation in your body which is required to stay skin organization and tight. Owing to the fervid lifestyle as well as the actually growing pollution, premature skin aging is a frequent downside as well as provides additional be concerned outraces to beauty nut ladies. Anti-oxidants are in excess of just the most up-to-date pattern on the subject of the ideal anti wrinkle cream. Skinlastin help you to avoid the sign of aging and Make Your Look more attractive by providing anti aging wrinkles solution. Here are several popular elements which will end result in slight to modest advancement inside the visual appeal of wrinkles. When coupled with vitamins C and E in an anti-wrinkle cream it packs a triple punch of anti-oxidant motion. These vitamins are substances derived from all-natural foodstuff sources that defend our body against the ravages of totally free radicals. In anti-aging creams, many other forms of vitamin A also are employed : retinol (pure vitamin A retinyl palmitate (also identified as pro-retinol A or pro-vitamin A retinyl acetate and retinyl linoleate. Hydroxy acids, including alpha and beta hydroxy acids, and poly hydroxy acids, are artificial acids that imitate all those obtained from milk and specified fruits. Aging is one in every of those stunning phases of life when your persona shimmers with knowledge and experience. Some formulas

are powerful ample to generate effects which were formerly only achievable with plastic medical procedures.

Hasil manusia

The moving on years generally sneak away your own skin's natural fats as well as wetness leaving the skin dull dried up and dull.

Skin spots, super skin discoloration, dark musca volitans are the skin color difficulties that ladies have to expression of their early mid-thirty-something.

Free of charge radicals mutate skin cells and so are the final result of normal aging, smoking cigarettes, also significantly daylight or simply a bad diet plan.

Owing to the fervid lifestyle as well as the actually growing pollution, premature skin aging is a frequent downside as well as provides additional be concerned outraces to beauty nut ladies.

And lastly, it's got been proven to even pores and skin tone and support inside the fix of sunlight harmed pores and skin.

Just like the simply results in of the plant whither in the absence associated with wet, your own skin color misplaces its activity and fresheners when it's lacking regarding moisture.

Though, aging is irreversible but the seven signs of aging can positively be reversed.

The actual move regarding youth slowdown the particular albumin creation in your body which is required to stay skin organization and tight.

The wisdom of centuries of regular medication, put together with the latest cutting-edge scientific and dermatologic research, is mixed in present-day major anti wrinkle pores and skin care items.

Anti aging Cream is really a superb response to all of your skin problems.

This wonderful anti-aging cream can reverse your biological clock and cause you to look years younger.

Anti aging Creams move to the layers of one's human external body part skin and provides strong bite coming via

within which makes it plump and glow skin just similar to never before.

The actual unimaginable formula regarding this crease reluctant enhances the collagen production thereby increasing the firmness and also litheness of your skin.

You can brighten out those unwanted wrinkles and get a firm, smooth and flawless skin with this advanced Wrinkle Cream.

It's a great awfully secure as well as economical natural healthy skin treatment answer that can make your own skin celestial stunning.

The usefulness of anti-wrinkle creams is dependent in component about the energetic ingredient or components.

Effortlessly its natural ingredients, anti wrinkle cream helps to reconstruct the actual extra gases of the human external body part skin color uncovering a lighten up vibrant glow on your own skin.

Some formulas are powerful ample to generate effects which were formerly only achievable with plastic medical procedures.

Anti-oxidants are in excess of just the most up-to-date pattern on the subject of the ideal anti wrinkle cream.

Hydroxy acids are an extremely common ingredient in anti-wrinkle creams, as they accelerate the production of collagen and boost skin hydration and smoothness.

Hydroxy acids, including alpha and beta hydroxy acids, and poly hydroxy acids, are artificial acids that imitate all those obtained from milk and specified fruits.

Hydroxy acids are highly effective as exfoliants, accelerating cell alternative, which consequently final results in younger-looking pores and skin.

Other elements normally found in anti wrinkle lotions include things like: kinetin, a plant hormone that can help the pores and skin retain moisture and smooth out wrinkles and unevenness; extract, a robust anti-inflammatory and antioxidant; and copper peptides that have the ability to regenerate tissue.

Vitamin Like Vitamin A, Vitamin C is an antioxidant that

fights the totally free radicals that direct to pores and skin getting older and wrinkles.
When coupled with vitamins C and E in an anti-wrinkle cream it packs a triple punch of anti-oxidant motion.
These vitamins are substances derived from all-natural foodstuff sources that defend our body against the ravages of totally free radicals.
Some say retinol has much more powerful.
In anti-aging creams, many other forms of vitamin A also are employed – retinol (pure vitamin A), retinyl palmitate (also identified as pro-retinol A or pro-vitamin A), retinyl acetate and retinyl linoleate.
These forms of Vitamin A make the skin stronger likewise as inspire the expansion of new collagen.
In addition, it increases the synthesis of collagen which allows to raise the skin's supporting framework.
This can be an advanced wrinkle lifting answer to create your skin flourish with a swish gleam.

Jumlah *correct* dari ringkasan tersebut adalah 26, dengan *missed* 5 dan *wrong* 5. Sehingga didapatkan nilai precision, recall dan F-measure yang sama yaitu 0.8387.