PERANCANGAN INFORMATION RETRIEVAL (IR) BERBASIS TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK PERINGKASAN TEKS TUGAS KHUSUS BERBAHASA INDONESIA

Erwien Tjipta Wijaya
Sekolah Tinggi Manajemen Informatika dan Komputer ASIA Malang
e-mail: erwin.cipta@gmail.com

ABSTRAKSI

Tugas khusus merupakan salah satu prasyarat kelulusan di STMIK ASIA. Informasi yang terdapat pada laporan tugas khusus terbilang cukup banyak, pada laporan tugas akhir berkisar antara 50 hingga 100 halaman. Karena begitu banyaknya informasi yang terkandung, tidak jarang seseorang mengalami kesulitan baik dalam mencari maupun memahami isi dari laporan tersebut.

Pemanfaatan Information Retrieval dalam peringkasan teks pada laporan Tugas Khusus, dapat membantu user dalam memahami isi laporan Tugas Khusus secara garis besar. Basis yang digunakan adalah Term Frequency Inverse Document Frequency **(TF-IDF) dan Cosine Similarity.**
Kesimpulan yang dapat diambil dari pembuatan rancangan Information Retrieval ini adalah bahwa sistem yang dirancang untuk membantu user dan memberikan informasi ringkas laporan Tugas Khusus.

Kata kunci: *Information Retrieval, Term Frequency Inverse Document frequency, Cosine Similarity*

ABSTRACT

The specific task is a prerequisite for graduation in STMIK ASIA. The information contained in the report specific task is quite a lot, in the final report ranged from 50 to 100 pages. Because so much information is contained, no less a person having difficulty either in finding and understanding the contents of the report.

Utilization of Information Retrieval in summary text of Task reports, can assist the user in understanding the contents of the report outlines the Special Duties. Base used is the Term Frequency Inverse Document Frequency **(TF-IDF) and the Cosine Similarity**.
The conclusion that can be drawn from the drafting Information Retrieval is that the system is designed to assist users and give concise information reporting Task.

Keywords: *Information Retrieval, the Term Frequency Inverse Document frequency, Cosine Similarity*

PENDAHULUAN

Tugas Khusus (TK) adalah hasil tertulis dari pelaksanaan suatu penelitian, yang dibuat untuk pemecahan masalah tertentu dengan menggunkan kaidah-kaidah yang berlaku dalam bidang ilmu tersebut. STMIK ASIA memberlakukan matakuliah Tugas Khusus (TK) sebagai syarat untuk mengambil matakuliah Tugas Akhir (TA) yang menjadi syarat utama kelulusan. Tugas Khusus (TK) rata-rata memiliki jumlah halaman atara 50 hingga 100 lembar.

Ketersediaan informasi yang banyak dalam Tugas Khusus (TK) menjadikan ringkasan sebagai kebutuhan yang sangat penting untuk proses pencarian. Dengan adanya ringkasan, pembaca dapat dengan cepat dan mudah memahami makna secara garis besar pada Tugas Khusus (TK) tanpa harus membaca keseluruhan isi Tugas Khusus (TK). Hal ini dapat menghemat waktu pembaca karena dapat menghindari pembacaan teks yang tidak relevan dengan informasi yang diharapkan oleh pembaca.

Berbagai cara telah diterapkan dan masih terus dikembangkan oleh para peneliti tentang peringkasan teks. Salah satunya adalah Peringkasan Teks Otomatis (Automated Text Summarization) atau sering disebut Text Summarization yaitu sebuah proses untuk menghasilkan ringkasan (summary) dari suatu Teks dengan menggunakan komputer. Tujuannya adalah mengambil sumber informasi dengan mengutip sebagian besar isi yang penting dan menampilkan kepada pembaca dalam bentuk yang ringkas sesuai dengan kebutuhan pembaca. Dengan demikian teknologi ini dapat membantu pembaca untuk menyerap informasi yang ada dalam Teks melalui ringkasan tanpa harus membaca seluruh isi dokumen

KAJIAN TEORI

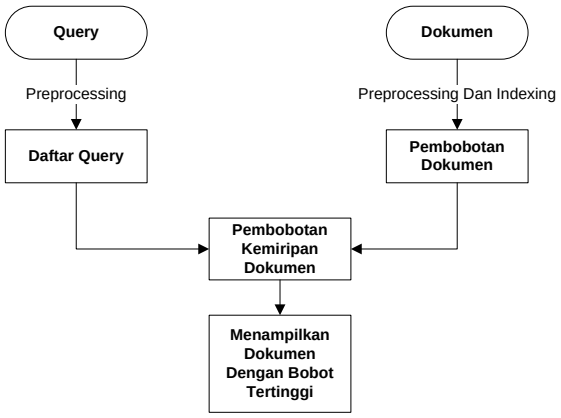
1. information retrieval (ir)

Information Retrieval merupakan bagian dari *computer science* yang berhubungan dengan pengambilan informasi dari dokumen-dokumen yang didasarkan pada isi dan konteks dari dokumen-dokumen itu sendiri. Berdasarkan referensi dijelaskan bahwa Information Retrieval merupakan suatu pencarian informasi yang didasarkan pada suatu *query* yang diharapkan dapat memenuhi keinginan *user* dari kumpulan dokumen yang ada.

a. Definisi Information Retrieval (IR)

"*Information Retrieval* adalah seni dan ilmu mencari informasi dalam dokumen, mencari dokumen itu sendiri, mencari metadata yang menjelaskan dokumen, atau mencari dalam *database*, apakah *relasional database* itu berdiri sendiri atau *database hypertext* jaringan seperti *Internet* atau *intranet*, untuk teks , suara, gambar, atau data "

Information Retrieval adalah "bidang di persimpangan ilmu informasi dan ilmu komputer. Berkutat dengan pengindeksan dan pengambilan informasi dari sumber informasi *heterogen* dan sebagian besar-tekstual. Istilah ini diciptakan oleh Mooers pada tahun 1951, yang menganjurkan bahwa diterapkan ke "aspek intelektual" deskripsi informasi dan sistem untuk pencarian (Mooers, 1951).



Gambar 1 : Diagram IR

Dari gambar diatas dapat dilihat bahwa proses IR memerlukan dua buah inputan yang pertama berupa dokumen yang akan diproses dan yang kedua berupa query atau potongan kalimat sederhana. kedua inputan tersebut kemudian akan dilakukan preprocessing dengan menggunakan langkah text mining dan dilakukan pembobotan, kemudian kedua inputan tersebut akan dibandingkan sehingga menghasilkan bobot yang baru yang akhirnya bobot tersebut akan digunakan sebagai acuan untuk

menampilkan dokumen yang berhubungan dengan query.

b. Pembuatan Index (*Indexing*)

Indexing dilakukan untuk membentuk basisdata terhadap koleksi dokumen yang dimasukkan, atau dengan kata lain, *indexing* merupakan proses persiapan yang dilakukan terhadap dokumen sehingga dokumen siap untuk diproses. Tahapan proses Indexing pada IR menggunakan tahapan preprocessing pada text mining.

2. Text Mining

Text mining adalah *data mining* dengan *input* data berupa *text. Text mining* muncul karena sekitar 90% data di dunia dalam bentuk format tidak terstruktur, adanya kebutuhan bisnis, yang asalnya document retrieval menjadi knowledge discovery.

Text mining adalah proses untuk menemukan pengetahuan baru, yang belum pernah diketahui, secara otomatis oleh komputer dari sumber-sumber tertulis yang berbeda **(Fan Weiguo, Wallace Linda, Rich Stephanie, and Zhang Zhongju, 2005)**.

Tahapan awal yang dilakukan dalam *text mining* disebut *preprocessing*. Tahapan yang dilakukan secara umum dalam *preprocessing* yaitu Casefolding, *Tokenizing*, *Filtering*, *Stemming*, *Tagging*, *dan Analyzing.*

Pada penelitian ini tahapan tagging tidak digunakan karena objek penelitian yang diambil berupa teks bahasa indosesia. Sehingga pada preprocessing tahapan yang akan dilakukan adalah Casefolding, Tokenizing, Filtering, Stemming, dan Analyzing, Yaitu :

Case Folding

Case folding adalah tahapan proses mengubah semua huruf dalam teks dokumen menjadi huruf kecil atau huruf besar, serta menghilangkan karakter selain a-z.

Tokenizing

Tokenizing adalah proses pemotongan string input berdasarkan tiap kata yang menyusunnya. Pemecahan kalimat menjadi kata-kata tunggal dilakukan dengan mencari kalimat dengan pemisah (delimiter) white space (spasi, tab, dan newline).

Filtering

Filtering merupakan proses penghilangan stopword. Stopword adalah katakata yang sering kali muncul dalam dokumen namun artinya tidak deskriptif dan tidak memiliki keterkaitan dengan tema tertentu. Didalam bahasa Indonesia stopword dapat disebut sebagai kata tidak penting, misalnya “di”, ”oleh”, “pada”, ”sebuah”, ”karena” ,”yang”,”dengan”.”yaitu” dan lain sebagianya.

Stemming

Tahap Stemming adalah proses mengubah sebuah kata turunan menjadi kata dasarnya dengan menggunakan aturan-aturan tertentu dari tiap kata hasil filtering.

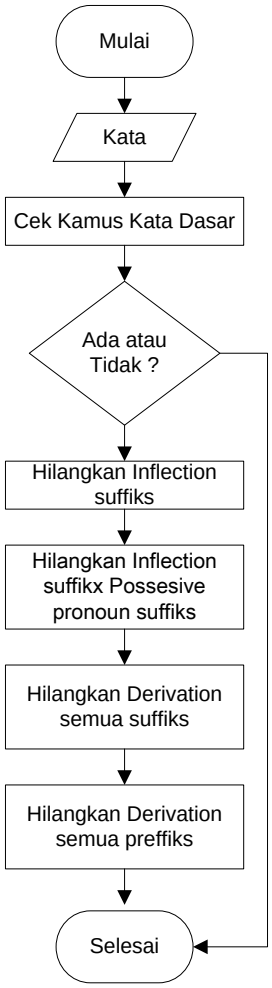
Pada dasarnya, bentuk umum kata berimbuhan dalam bahasa Indonesia adalah seperti berikut :

Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1

Dalam penelitian ini, Algoritma stemming yang akan digunakan adalah Algoritma Nazief dan Adriani (1996).

Algoritma ini mengacu pada aturan morfologi bahasa Indonesia yang mengelompokkan imbuhan, yaitu imbuhan yang diperbolehkan atau imbuhan yang tidak diperbolehkkan. Pengelompokan ini termasuk imbuhan di depan (awalan), imbuhan kata di belakang (akhiran), imbuhan kata di tengah (sisipan) dan kombinasi imbuhan pada awal dan akhir kata (konfliks). Algoritma ini menggunakan kamus kata keterangan yang digunakan untuk mengetahui bahwa *stemming* telah mendapatkan kata dasar.

Langkah-langkah algoritma Nazief dan Adriani berdasarkan flowchart berikut :



Gambar 2 : Flowchart Algoritma Nazief Adriani

Kelebihan algoritma stemming nazief dan adriani adalah algoritma ini memperhatikan kemungkinan adanya partikel-partikel yang mungkin mengikuti suatu kata berimbuhan. Sehingga dapat melihat rumus untuk algoritma ini yaitu adanya penempatan *possessive pronoun* dan juga *particle* yang mungkin ada pada sebuah imbuhan yang mungkin ada pada suatu kata berimbuhan. Akhir dari algoritma ini yaiu apabila pemotongan semua imbuhan tersebut pada rumus maka algoritma ini yaitu apbila pemotongan semua imbuhan telah berhasil dan hasil pemotongan imbuhan tersebut terdapat pada kamus maka algoritma ini dapat dikatakan berhasil dalam penentuan kata dasarnya. Dan apabila sebaliknya bahwa algoritma ini setelah dilakukan pemotongan kata dan tidak terdapat pada kamus maka kata berimbuhan yang telah mengalami pemotongan dikembalikan ke keadaan semula.

Aturan Stemming Nazief-Adriani

Untuk menyempurnakan algoritma di atas, maka ditambahkan aturan-aturan dibawah ini:

1. Aturan untuk reduplikasi.

a. Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata yang sama maka *root word* adalah bentuk tunggalnya, contoh : "buku-buku" *root word*-nya adalah "buku".

b. Kata lain, misalnya "bolak-balik", "berbalas-balasan, dan "seolah-olah". Untuk mendapatkan *root word*-nya, kedua kata diartikan secara terpisah. Jika keduanya memiliki *root word* yang sama maka diubah menjadi bentuk tunggal, contoh: kata "berbalas-balasan", "berbalas" dan "balasan" memiliki *root word* yang sama yaitu "balas", maka *root word* "berbalas-balasan" adalah "balas". Sebaliknya, pada kata "bolak-balik", "bolak" dan "balik" memiliki *root word* yang berbeda, maka *root word*-nya adalah "bolak-balik".

2. Tambahan bentuk awalan dan akhiran serta aturannya. Untuk tipe awalan "mem-", kata yang diawali dengan awalan "memp-" memiliki tipe awalan "mem-". Tipe awalan "meng-", kata yang diawali dengan awalan "mengk-" memiliki tipe awalan "meng-".

a. **Analyzing**

Tahap terakhir adalah tahap *analyzing*, yaitu tahap penentuan seberapa jauh keterhubungan antar kata-kata pada dokumen yang ada dengan menghitung frekuensi term pada dokumen. Tahap ini disebut juga tahap pembobotan, yaitu dijelaskan sebagai berikut :

1. Pembobotan Term

Term adalah suatu kata atau suatu kumpulan kata yang merupakan ekspresi verbal dari suatu pengertian. Dalam i*nformation retrieval* sebuah

term perlu diberi bobot,karena semakin sering suatu *term* muncul pada suatu dokumen, maka kemungkinan *term* tersebut semakin penting dalam dokumen. Pembobotan *term* (*term weighting*) merupakan salah satu operasi yang dibutuhkan untuk membantu suatu proses *information retrieval* yaitu dengan menghitung kemunculan frekuensi suatu *term* pada sebuah dokumen. Pembobotan term dibutuhkan dalam menentukan peringkat dokumen (*document ranking*). Salah satu basis pembobotan *term* adalah TF (*Term Frequency*). TF merupakan frekuensi kemunculan *term* pada dokumen.

Dari proses Pembobotan Term maka akan didapatkan hasil akhir berupa *Term Frequency* (TF) yaitu merupakan frekuansi atau jumlah masing-masing kata. Hasil pembobotan Term kemudian akan digunakan sebagai dasar perhitungan pada basis Term Frequency-Inverse Document Frequency (TF-IDF).

2. *Term Frequency-Inverse Document Frequency* (TF-IDF)

Term Frequency-Inverse Document Frequency (TF-IDF) adalah cara pemberian bobot hubungan suatu kata (*term*) terhadap dokumen. Untuk dokumen tunggal tiap kalimat dianggap sebagai dokumen. Basis ini menggabungkan dua konsep untuk perhitungan bobot, yaitu *Term frequency* (TF) merupakan frekuensi kemunculan kata (*t*) pada kalimat (*d*). *Document frequency* (DF) adalah banyaknya kalimat dimana suatu kata (*t*) muncul.

Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen (Robertson, 2005).

Basis pembobotan TF-IDF adalah jenis pembobotan yang sering digunakan dalam IR (information retrieval) dan text mining. Pembobotan ini adalah suatu pengukuran statistik untuk mengukur seberapa penting sebuah kata dalam kumpulan dokumen. Tingkat kepentingan meningkat ketika sebuah kata muncul beberapa kali dalam sebuah dokumen tetapi diimbangi dengan frekuensi kemunculan kata tersebut dalam kumpulan dokumen. TF-IDF dapat dirumuskan sebagai berikut :

$$TF-IDF\left(t_k,d_j\right)=TF\left(t_k,d_j\right)*IDF\left(t_k\right)$$

Keterangan:
d_j = Dokumen ke-j
t_k = Term ke-k

Dimana sebelumnya dihitung terlebih dahulu *Term Frequency* (TF) yaitu frekuensi kemunculan suatu *term* di tiap dokumen. Kemudian dihitung *Inverse Document Frequency* (IDF) yaitu nilai bobot suatu term dihitung dari seringnya suatu term muncul di beberapa dokumen. Semakin sering suatu term muncul di banyak dokumen, maka nilai IDF nya akan kecil. Berikut rumus-rumus TF dan IDF.

$$TF\left(t_k,d_j\right)=f\left(t_k,d_j\right)$$

Keterangan :
TF = Jumlah frekuensi term
f = Jumlah frekuensi kemunculan
d_j = Dokumen ke-j
t_k = Term ke-k

Untuk menghitung nilai IDF bisamenggunakan persamaan sebagai berikut, yaitu :

$$IDF\left(t_k\right)=\frac{1}{df\left(t\right)}$$

Atau

$$IDF\left(t_k\right)=\log\frac{N}{df\left(t\right)}$$

Keterangan :
IDF = bobot term
N = Jumlah total dokumen
df = Jumlah kemunculan dokumen
d_j = Dokumen ke-j
t_k = Term ke-k

Persamaan diatas hanya boleh digunakan apabila hanya terdapat satu buah dokumen saja yang diproses sedangkan persamaan 2,4 digunakan pada proses yang melibatkan banyak dokumen.

3.PERINGKASAN TEKS (TEXT SUMMARYZATION)

Peringkasan teks otomatis (*automatic text summarization*) adalah pembuatan versi yang lebih singkat dari sebuah teks dengan memanfaatkan aplikasi yang dijalankan pada komputer . Hasil peringkasan ini mengandung poin-poin penting dari teks asli. Dalam Hovy (**2001**), *summary* atau ringkasan didefinisikan sebagai sebuah teks yang dihasilkan dari satu at au lebih teks, mengandung informasi dari teks asli dan panjangnya tidak lebih dari setengah teks asli.

Terdapat dua pendekatan pada peringkasan teks, yaitu ekstraksi (*shallower approaches*) dan abstraksi (*deeper approaches*). Pada teknik ekstraksi, sistem menyalin informasi yang dianggap paling penting dari teks asli menjadi ringkasan (sebagai contoh, klausa utama, kalimat utama, atau paragraf utama). Sedangkan teknik abstraksi melibatkan parafrase dari teks asli. Pada umumnya, abstraksi dapat meringkas teks lebih kuat daripada ekstraksi, tetapi sistemnya lebih sulit dikembangkan karena mengaplikasikan teknologi *natural language generation* yang merupakan bahasan yang dikembangkan tersendiri.

Berdasarkan jumlah sumbernya, ringkasan teks dapat dihasilkan dari satu sumber (*single-document*) atau dari banyak sumber (*multi-document*). Suatu ringkasan dapat bersifat *general*, yaitu ringkasan yang berupaya mengambil sebanyak mungkin informasi umum yang mampu menggambarkan keseluruhan isi teks. Selain itu dapat juga informasi yang diambil untuk ringkasan berdasar pada *query* yang didefinisikan *user*.

Terdapat 2 (dua) jenis tipe peringkasan otomatis yang dikembangkan dengan teks *mining*, yaitu :

1. Peringkasan otomatis tipe abstraksi

Yaitu tipe peringkasan otomatis yang dilakukan oleh mesin dengan meniru cara peringkasan yang dilakukan menggunakan bahasa manusia (*human language*), yaitu dengan melakukan perubahan pada susunan kata dengan cara melakukan penambahan atau pengurangan pada setiap kalimat yang dianggap inti kalimat.

Contoh :

Kalimat = "*text mining* adalah algoritma pengembangan dari *data mining* yang digunakan untuk pengolahan data berupa *text document*. Tahapan pada text mining disebut *preprocessing*. *Information retrieval* menggunakan preprocessing dalam melakukan *indexing*"

Query = "*text mining*"

Ringkasan = "*text mining* merupakan bagian dari *information retrieval* pada *indexing*"

2. Peringkasan otomatis tipe ekstraksi

Peringkasan otomatis tipe ekstraksi adalah peringkasan yang dilakukan mesin dengan melakukan

pembobotan pada setiap kalimat dan hanya menampilkan informasi dengan nilai bobot tertinggi tanpa melakukan perubahan (pengurangan atau penambahan) kata pada setiap kalimat yang ditampilkan.

Contoh :

Kalimat = "*text mining* adalah algoritma pengembangan dari *data mining* yang digunakan untuk pengolahan data berupa *text document*. Tahapan pada text mining disebut *preprocessing*. *Information retrieval* menggunakan preprocessing dalam melakukan *indexing*"

Query = "*text mining*"

Ringkasan = "*Text Mining* Adalah Algoritma Pengembangan Dari *Data Mining* Yang Digunakan Untuk Pengolahan Data Berupa *Text Document*. Tahapan Pada Text Mining Disebut *Preprocessing*."

Cosine Similarity

Cosine similarity digunakan untuk menghitung pendekatan relevansi *query* terhadap dokumen. Penentuan relevansi sebuah *query* terhadap suatu dokumen dipandang sebagai pengukuran kesamaan antara *vektor query* dengan *vektor* dokumen. Semakin besar nilai kesamaan *vector query* dengan *vektor* dokumen maka *query* tersebut dipandang semakin relevan dengan dokumen. Saat mesin menerima *query*, mesin akan membangun sebuah *vektor* Q (w q1 ,w q2 ,…w qt) berdasarkan istilah-istilah pada query dan sebuah *vector* D (d i1 ,d i2 ,…d it) berukuran t untuk setiap dokumen. Adapun persamaannya yaitu sebagai berikut, yaitu :

$$CS\left(\bar{q},\bar{d}\right)=\frac{\bar{q}.\bar{d}}{|\bar{q}|.|\bar{d}|}$$

Keterangan:

$CS\left(\bar{q},\bar{d}\right)$ = Cosine Similarity vektor query dan dokumen

$\bar{q}$ = Vektor query

$\bar{d}$ = Vector dokumen

$|\bar{q}|$ = panjang vector query

$|\bar{d}|$ = panjang vector dokumen

Pada umumnya *cosine similarity* (CS) dihitung dengan rumus *cosine measure* (**Grossman**, **1998**).

$$CS\left(b1,b2\right)=\frac{\sum_{t}^{n}=1W_{t,b1}W_{t,b2}}{\sqrt{\sum_{t}^{n}=1W^{2}_{t,b1}\sum_{t}^{n}=1W^{2}_{t,b2}}}$$

Keterangan:

CS(b1,b2) = Cosine Similarity dalam blok b1 dan b2

t = *term* dalam kalimat

W t,b1 = bobot term t dalam blok b1

W t,b2 = bobot term t dalam blok b2

PEMBAHASAN

1. Analisa Data

Sebelum proses text mining dilakukan, diperlukan proses data converse dari data berbentuk document (.doc) menjadi plain text (.txt). Tujuan dari *converse* ini adalah menghilangkan isi dari document yang tidak diperlukan dalam text mining seperti gambar, tabel, dan format text sehingga prosesnya menjadi lebih mudah dan efektif untuk kebutuhan pengguna, dengan indicator sebagai berikut :

1. Mendapatkan hasil yang lebih akurat.
2. Pengurangan waktu komputasi untuk masalah dengan skala besar.

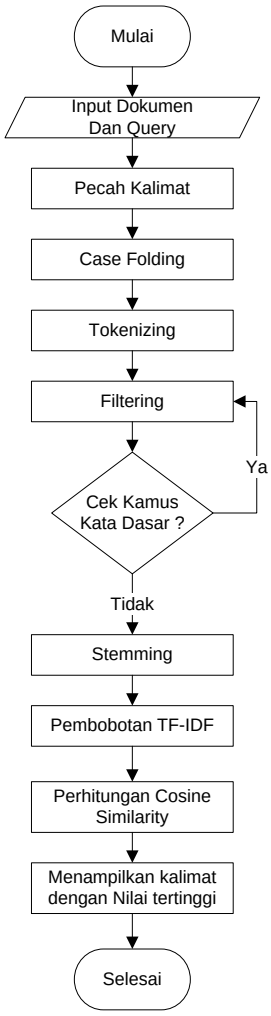
Membuat nilai data menjadi lebih kecil tanpa merubah informasi yang dikandungnya.

2. Flowchart Sistem

Alur peringkasan secara keseluruhan memiliki beberapa tahapan proses sebagai berikut :

Berdasarkan gambar flowchart dibawah maka dapat dijelaskan alur

peringkasan adalah sebagai berikut, yaitu :



Gambar 3 : Flowchart Sistem

Contoh Kasus

Pada pembahasan Contoh kasus digunakan suatu contoh latar belakang Tugas Khusus untuk peringkasan tersebut adalah sebagai berikut, yaitu:

Latar Belakang Tugas Khusus

Koperasi serba usaha merupakan salah satu badan usaha yang bergerak dalam perdagangan *retail* berskala menengah, dikelola seorang manager dan berusaha berdampingan dengan warung atau toko perorangan, pasar tradisional dan pasar modern lainnya. Sebagai suatu bentuk usaha maka koperasi serba usaha termasuk dalam kategori usaha penyedia kebutuhan sehari-hari yang berorientasi pada pemenuhan kebutuhan anggota dan masyarakat pada umumnya. Karena itu koperasi serba usaha harus dikelola dengan *management* dan kebijakan yang tepat seperti layaknya sebuah perusahaan perdagangan.

Karena banyaknya kebutuhan sehari-hari maka koperasi memantau kebiasaan belanja anggota dan masyarakat umum lainnya. Manager koperasi menganalisa dan memahami infomasi yang berhubungan dengan perilaku pelanggan adalah alat untuk memenangkan persaingan. Oleh karenanya diperlukan sebuah sistem yang dapat memberikan informasi yang dibutuhkan. Sistem tersebut bertugas memberikan analisa asosiasi barang sebagai dasar pertimbangan dalam merencanakan atau meramalkan barang-barang yang harus di stok dalam gudang dan menetapkan persediaan minimum.

Penggunaan sistem penggalian data (*data mining*) dengan algoritma Apriori. Untuk mencari seberapa besar tingkat asosiasi bahwa barang A dibeli bersamaan dengan barang B, atau sebaliknya. Dengan mengetahui pola assosiasi tersebut maka diharapkan manager dapat mengambil keputusan yang tepat terhadap perkembangan usaha tersebut. Apabila telah ditemukan beberapa pola assosiasi, maka dapat menentukan strategi pemasaran misalnya memberikan potongan harga pada setiap pola assosiasi atau menjadikan pola assosiasi tersebut menjadi sebuah paket penjualan dengan harga yang lebih murah di bandingkan harga ecerannya. Membandingkan hasil akhir dari kedua kondisi yaitu sebelum diambil kebijakan dan sudah diambil kebijakan dapat dianalisa tingkat keberhasilan suatu sistem.

Dengan maksud tersebut diatas maka dibuatlah tugas akhir dengan judul "PERANCANGAN DAN IMPLEMENTASI PENERAPAN DATA MINING UNTUK ANALISIS POLA

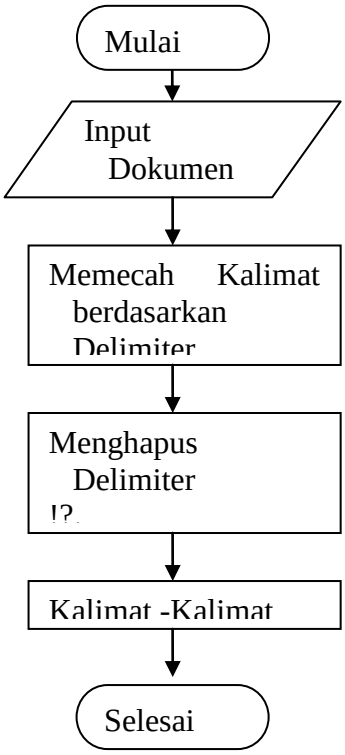
ELASTISITAS MINAT KONSUMEN DAN PENETAPAN HARGA PADA KOPERASI DENGAN METODE APRIORI".

Untuk melakukan peringkasan diperlukan sebuah query yang bertujuan sebagai pembanding dala menentukan bobot. Dalam kasus ini query yang digunakan adalah judul dari Tugas Khusus itu sendiri, yaitu:

Query
Perancangan dan implementasi data mining untuk analisa pola elastisitas minat konsumen dan penetapan harga pada koperasi dengan metode apriori

3. Pemecahan Kalimat

Pemecahan kalimat adalah tahap memecah string dokumen menjadi kumpulan kalimat dengan melakukan pemotongan dokumen menjadi kalimat-kalimat berdasarkan tanda baca akhir kalimat *(delimiter).* Flowchartnya adalah sebagai berikut :



Gambar 4 : Flowchart Pemecahan Kalimat

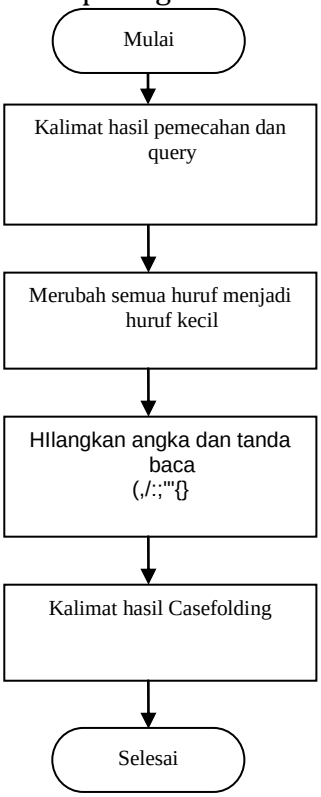
Pada kasus ini tanda baca akhir kalimat yang dijadikan sebagai patokan pemotongan kalimat adalah tanda baca titik ".", tanda baca tanya "?" dan tanda baca seru "!". Sehingga menghasilkan potongan kalimat-kalimat seperti terlihat sebagai berikut :

No	Kalimat
1	Koperasi serba usaha merupakan salah satu badan usaha yang bergerak dalam perdagangan *retail* berskala menengah dikelola seorang manager dan berusaha berdampingan dengan warung atau toko perorangan, pasar tradisional dan pasar modern lainnya
2	Sebagai suatu bentuk usaha maka koperasi serba usaha termasuk dalam kategori usaha penyedia kebutuhan sehari-hari yang berorientasi pada pemenuhan kebutuhan anggota dan masyarakat pada umumnya
3	Karena itu koperasi serba usaha harus dikelola dengan management dan kebijakan yang tepat seperti layaknya sebuah perusahaan perdagangan
4	Karena banyaknya kebutuhan sehari-hari maka koperasi memantau kebiasaan belanja anggota dan masyarakat umum lainnya
5	Manager koperasi menganalisa dan memahami infomasi yang berhubungan dengan perilaku pelanggan adalah alat untuk memenangkan persaingan
6	Oleh karenanya diperlukan sebuah sistem yang dapat memberikan informasi yang dibutuhkan
7	Sistem tersebut bertugas memberikan analisa asosiasi barang sebagai dasar pertimbangan dalam merencanakan atau meramalkan barang-barang yang harus di stok dalam gudang dan menetapkan persediaan minimum
8	Penggunaan sistem penggalian data (data mining) dengan algoritma Apriori
9	Untuk mencari seberapa besar tingkat asosiasi bahwa barang A dibeli bersamaan dengan barang B, atau sebaliknya
10	)engan mengetahui pola assosiasi tersebut maka diharapkan manager dapat mengambil keputusan yang tepat terhadap perkembangan usaha tersebut
11	ıpabila telah ditemukan beberapa pola assosiasi, maka dapat menentukan strategi pemasaran misalnya memberikan potongan harga pada setiap pola assosiasi atau menjadikan pola assosiasi tersebut menjadi sebuah paket penjualan dengan harga yang lebih murah di bandingkan harga ecerannya
12	Membandingkan hasil akhir dari kedua kondisi yaitu sebelum diambil kebijakan dan sudah diambil kebijakan dapat dianalisa tingkat keberhasilan suatu sistem
13	Dengan maksud tersebut diatas maka dibuatlah tugas akhir dengan judul "PERANCANGAN DAN IMPLEMENTASI PENERAPAN DATA MINING UNTUK ANALISIS POLA ELASTISITAS MINAT KONSUMEN DAN PENETAPAN HARGA PADA KOPERASI DENGAN METODE APRIORI"

4. Case Folding

Case Folding adalah tahapan mengubah kalimat menjadi huruf kecil

(*lower case*) dan menghilangkan semua jenis tanda baca dan symbol sehingga menghasilkan kalimat yang murni dan sederhana. Tahapan Case Folding seperti terlihat pada gambar berikut :



Gambar 5 : Flowchart Case Folding

5. Tokenizing

Tokenizing merupakan tahapan memecah kalimat menjadi kata-kata yang berdiri sendiri-sendiri, yaitu dengan cara melakukan pemisahan masing-masing kata dengan cara memotong semua kata berdasarkan spasi " ". Tahapan tokenizing seperti telihat pada gambar berikut :

Pada kasus ini, kumpulan kalimat hasil dari *case folding* kemudian dilakukan proses *tokenizing* kata yaitu memotong kalimat menjadi kata-kata berdasarkan karakter pemisah (*delimiter*) yang menyusunnya berupa karakter spasi (UTF8 kode 0020). Berdasarkan tabel 3.4 dan tabel 3.5. Proses *tokenizing* menghasilkan token kata sejumlah 184 kata yang berbeda pada dokumen dan 18 kata yang berbeda pada *query*. Yaitu terlihat pada tabel berikut :

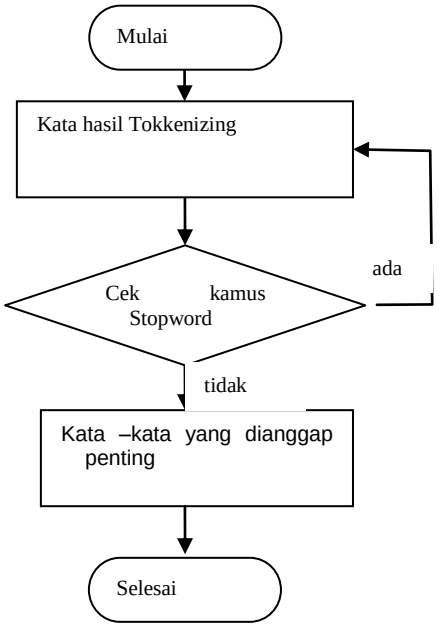
Kata			
koperasi	Analisa	pemenuhan	menentukan
serba	Asosiasi	anggota	Strategi
Usaha	Barang	masyarakat	Pemasaran
Merupakan	Dasar	umumnya	Misalnya
Salah	pertimbangan	karena	Potongan
Satu	merencanakan	itu	Harga
Badan	meramalkan	harus	Setiap
Yang	Di	management	Menjadikan
Bergerak	Stok	kebijakan	Menjadi
Dalam	Gudang	tepat	Paket

Tabel ditas merupakan sebagian dari semua kata yang telah mengalami proses tokeniaing.

Query			
perancangan	minat	Untuk	Koperasi
Dan	konsumen	Analisa	Dengan
implementasi	penetapan	Pola	Metode
Data	harga	elastisitas	Apriori
Mining	pada		

6. Filtering

Filtering merupakan tahapan dimana kata-kata yang dianggap tidak memiliki makna dihilangkan. Pada tahapan filtering diperlukan sebuah kamus stopword yang berisikan list dari kata-kata yang dianggap tidak memiliki makna.



Gambar 7 : Flowchart Filtering

Pada tahap ini, kata-kata hasil yokenizing dilakukan pembuangan kata-kata yang dianggap tidak memiliki makna sesuai dengan kata-kata yang terdapat pada kamus stopword. *Stopword* adalah kata-kata yang kurang deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Pembuangan *stopword* dilakukan dengan mengecek kamus *stopword*. Jika terdapat kata yang sama dengan kamus maka akan dihapus. Kata hasil token dicek terlebih dahulu untuk dicocokkan dengan kamus *stopword*. Jika dalam pencocokan terdapat kata yang sama dalam kamus maka kata tersebut dihilangkan. kata-kata yang termasuk dalam *stopword* dalam contoh kasus ini adalah : "yang", "dan", "akan", "dengan", "di" , "pada" , "yaitu" , "atau" dan "itu" , "adalah". Hasil *Filtering* berdasarkan tabel 3.6 dan tabel 3.7, maka sebagian hasilnya seperti pada tabel berikut :

Kata			
koperasi	menganalisa	perdagangan	tugas
analisa	kedua	menetapkan	bentuk
pemenuhan	seberapa	seperti	assosiasi
menentukan	memahami	penjualan	judul
serba	kondisi	Retail	Maka
asosiasi	warung	Persediaan	diharapkan
anggota	besar	Layaknya	karenanya
strategi	infomasi	Lebih	perancangan
usaha	tingkat	Berskala	termasuk
barang	sebelum	Minimum	mengambil

Query			
perancangan	Minat	Analisa	koperasi
implementasi	konsumen	Pola	metode
data	penetapan	Elastisitas	apriori
mining	Harga		

7. Stemming

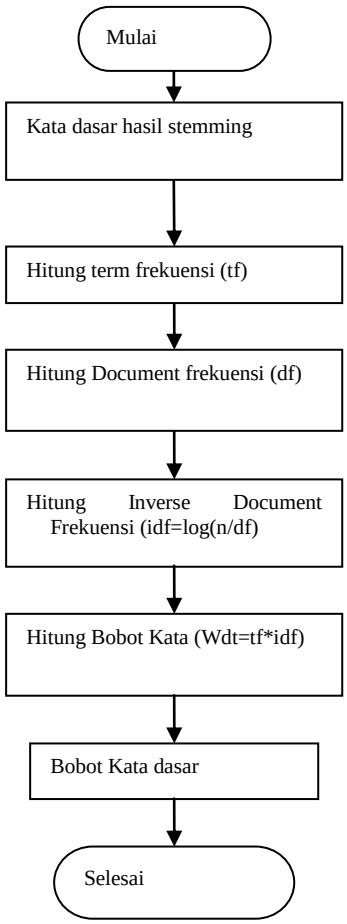
Stemming adalah tahapan mengubah sebuah kata turunan menjadi kata dasarnya dengan menggunakan aturan-aturan. Aturan stemming yang digunakan adalah aturan dari algoritma Nazief-Adriani, dimana pada aturan algoritma Nazief-Adriani yaitu melakukan pengecekan pada kamus kata dasar sebelum melakukan pemotongan awalan dan akhiran untuk mencari kata dasar. Hasil *filtering* kemudian di-*stemming* untuk mendapatkan kata dasar (*root*). Proses *stemming* menggunakan bantuan kamus-kamus kecil yang digunakan untuk membedakan suatu kata yang mengandung imbuhan baik prefiks maupun sufiks yang salah satu suku katanya merupakan bagian dari imbuhan, terutama dengan kata dasar yang mempunyai suku kata lebih besar dari dua.

Dokumen	
Kata	
koperasi ==> koperasi	potongan ==> potong
menganalisa ==> analisa	dianalisa ==> analisa
perdagangan ==> dagang	banyaknya ==> banyak
tugas ==> tugas	dapat ==> dapat
analisa ==> analisa	satu ==> satu
kedua ==> dua	tradisional ==> tradisional
menetapkan ==> tetap	ecerannya ==> ecer
bentuk ==> bentuk	analisis ==> analisis
pemenuhan ==> penuh	merencanakan ==> rencana
seberapa ==> berapa	bersamaan ==> sama
seperti ==> seperti	seorang ==> orang
assosiasi ==> assosiasi	kebutuhan ==> butuh
menentukan ==> tentu	harga ==> harga
memahami ==> paham	keberhasilan ==> hasil
penjualan ==> jual	data ==> data
judul ==> judul	perkembangan ==> kembang

8. Analyzing

Analyzing merupakan tahapan penghitungan bobot setiap kata untuk menentukan seberapa jauh keterhubungan antar kata-kata pada dokumen yang ada dengan menghitung frekuensi term pada dokumen. Pada tahapan ini penghitungan dilakukan dengan menggunakan algoritma *Term Frequency-Inverse Document Frequency* (TF-IDF). Tahapan algoritma *Term Frequency-Inverse Document Frequency*

(TF-IDF) seperti terlihat pada flowchart gambar 8, adalah sebagai berikut



Gambar 7 : Flowchart Filtering

Hasil proses *text preprosessing* dilakukan pembobotan tf-idf. Pembobotan secara otomatis biasanya berdasarkan jumlah kemunculan suatu kata dalam sebuah dokumen (*term frequency*) dan jumlah kemunculannya dalam koleksi dokumen (*inverse document frequency*). Bobot kata semakin besar jika sering muncul dalam suatu dokumen dan semakin kecil jika muncul dalam banyak dokumen. Berikut adalah hasil penghitungan *Document Term Frequency* :

Dokumen			
Kata	**frekuensi**	**Kata**	**frekuensi**
koperasi	6	barang	5
serba	3	dasar	1
usaha	9	timbang	1
rupa	1	rencana	1
salah	1	ramal	1
Satu	1	stok	1
badan	1	gudang	1
gerak	1	tetap	2
dagang	2	dia	1
retail	1	minimum	1
skala	1	guna	1
tengah	1	gali	1
kelola	2	data	3
orang	2	mining	2
manager	3	algoritma	1

Term Frequency Inverse Documen frequency

Setelah diketahui *Document Term Frequency* maka selanjutnya adalah penghitungan idf.

Dokumen					
Kata	**df**	**Idf**	**Kata**	**df**	**Idf**
koperasi	6	0.3357	barang	5	0.4149
serba	3	0.6368	dasar	1	1.1139
usaha	9	0.1597	timbang	1	1.1139
rupa	1	1.1139	rencana	1	1.1139
salah	1	1.1139	ramal	1	1.11392
Satu	1	1.1139	stok	1	1.1139
badan	1	1.1139	gudang	1	1.1139
gerak	1	1.1139	tetap	2	0.8129
dagang	2	0.8129	dia	1	1.1139
retail	1	1.1139	minimum	1	1.1139

Hasil penghitungan Term Frquency tersebut kemudian dihitung dengan menggunakan metode tf-idf sehingga dihasilkan nilai idf untuk masing-masing kata dasar yang telah di preprocessing. Hasil penghitungan tersebut kemudian akan digunakan sebagai dasar penghitungan Wdt untuk tiap kata yang dimiliki masing-masing kalimat.

Nilai TF/IDF kalimat ke -1. Perhitungan ini dilakukan untuk semua kalimat.

Kalimat 1							
kata	**IDF**	**Tf**	**Wdt**	**kata**	**IDF**	**Tf**	**Wdt**
koperasi	0.3357	1	0.3357	tengah	1.1139	1	1.1139
serba	0.6368	1	0.6368	kelola	0.8129	1	0.8129
usaha	0.1597	3	0.4791	orang	0.8129	2	1.6258
rupa	1.1139	1	1.1139	manager	0.6368	1	0.6368

salah	1. 1139	1	1. 1139	damping	1. 1139	1	1. 1139
satu	1. 1139	1	1. 1139	warung	1. 1139	1	1. 1139
badan	1. 1139	1	1. 1139	toko	1. 1139	1	1. 1139
gerak	1. 1139	1	1. 1139	pasar	0.6368	2	1.2736
dagang	0.8129	1	0.8129	tradisional	1. 1139	1	1. 1139
retail	1. 1139	1	1. 1139	modern	1. 1139	1	1. 1139
skala	1. 1139	1	1. 1139	lain	0.8129	1	0.8129

9. Cosine Similarity

Setelah penghitungan pada masing-masing kalimat, selanjutnya melakukan penghitungan kemiripan vector query dengan setiap kalimat yang ada. Menghitung kemiripan vector dilakukan dengan menggunakan *cosine similarity*. Yang dilakukan pertama kali dalam menghitung kemiripan vector yaitu menghitung skalar masing-masing kalimat yaitu dengan mengkuadratkan bobot idf pada setiap term yang dimiliki query lalu mengalikannya dengan tf masing-masing kalimat. Perhitungan seperti terlihat pada table.

Term skalar query Frequency

Query Term	**K1**	**K2**	**K3**	**K4**	**K5**	**K6**	**K7**	**K8**	**K9**	**K10**	**K11**	**K12**	**K13**
rancang	0	0	0	0	0	0	0	0	0	0	0	0	**1**
implementasi	0	0	0	0	0	0	0	0	0	0	0	0	**1**
data	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**
mining	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**
minat	0	0	0	0	0	0	0	0	0	0	0	0	**1**
konsumen	0	0	0	0	0	0	0	0	0	0	0	0	**1**
tetap	0	0	0	0	0	0	0	0	0	0	0	0	**1**
harga	0	0	0	0	0	0	0	0	0	0	**3**	0	**1**
analisa	0	0	0	0	**1**	0	**1**	0	0	0	0	**1**	**1**
pola	0	0	0	0	0	0	0	0	0	**1**	**3**	0	**1**
elastisitas	0	0	0	0	0	0	0	0	0	0	0	0	**1**
koperasi	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	**1**
metode	0	0	0	0	0	0	0	0	0	0	0	0	**1**
apriori	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**

Query Term	**Q = (tf*idf) *(tf*idf)**	**K1**	**K2**	**K3**	**K5**	**K6**	**K7**
rancang	1.2408	0	0	0	0	0	0
Implementasi	1.2408	0	0	0	0	0	0
Data	0.4055	0	0	0	0	0	0
Mining	0.6608	0	0	0	0	0	0
Minat	1.2408	0	0	0	0	0	0
konsumen	1.2408	0	0	0	0	0	0
Tetap	0.6608	0	0	0	0	0	0
Harga	0.2620	0	0	0	0	0	0
Analisa	1.2408	0	0	0	0	**1.2408**	0
Pola	0.1722	0	0	0	0	0	0
elastisitas	1.2408	0	0	0	0	0	0
koperasi	0.1127	**0.1127**	**0.1127**	**0.1127**		**0.1127**	0
Metode	1.2408	0	0	0	0	0	0
Apriori	0.6608	0	0	0	0	0	0
Total Skalar	**11.6210**	**0.1127**	**0.1127**	**0.1127**	**1.3536**	**0**	**1.2408**

Query Term	**Q = (tf*idf) *(tf*idf)**	**K8**	**K9**	**K10**	**K11**	**K12**	**K13**
rancang	1.2408	0	0	0	0	0	**1.2408**
implementasi	1.2408	0	0	0	0	0	**1.2408**
data	0.4055	**0.4055**	0	0	0	0	**0.4055**
mining	0.6608	**0.6608**	0	0	0	0	**0.6608**
minat	1.2408	0	0	0	0	0	**1.2408**
konsumen	1.2408	0	0	0	0	0	**1.2408**
tetap	0.6608	0	0	0	0	0	**0.6608**
harga	0.2620	0	0	0	**0.7860**	0	**0.2620**
analisa	1.2408	0	0	0	0	**1.2408**	**1.2408**
pola	0.1722	0	0	**0.1722**	**0.5166**	0	**0.1722**
elastisitas	1.2408	0	0	0	0	0	**1.2408**
koperasi	0.1127	0	0	0	0	0	**0.1127**
metode	1.2408	0	0	0	0	0	**1.2408**
apriori	0.6608	**0.6608**	0	0	0	0	**0.6608**
Total Skalar	**11.6210**	**1.7272**	**0**	**0.1722**	**1.3026**	**1.2408**	**11.621**

Selanjutnya melakukan perhitungan panjang vector untuk setiap kalimat yaitu dengan mengkuadratkan nilai Wdt masing-masing pada tabel 3.14 hingga tabel 3.26, sehingga menghasilkan nilai seperti terlihat pada tabel :

10. Hasil Peringkasan

Tahap terakhir yaitu menampilkan hasil kalimat-kalimat yang dianggap sebagai inti teks. Untuk menentukan kalimat-kalimat mana saja yang akan ditampilkan berdasarkan bobotnya

yaitu ditentukan berdasarkan nilai rata-rata query relevancy, yaitu:

$$Bk = \frac{\sum Cos(Q)}{n}$$

Keterangan :
Bk = Batas nilai kalimat

$\sum Cos(Q)$ = Total query relevancy
n = jumlah kalimat

Sehingga menghasilkan perhitungan sebagai berikut, yaitu :

$$Bk = \frac{\sum Cos(Q)}{n} = \frac{\mathbf{0.9477}}{13} = 0.0729$$

Setelah didapatkan batas nilai kalimat maka selanjutnya adalah menampilkan semua kalimat dengan nilai yang lebih besar atau sama dengan batas nilai kalimat yang diperoleh. Hasil kalimat yang memenuhi syarat akan diurutkan sesuai urutan awal kalimat, seperti terlihat pada tabel berikut :

K-n	Cos(Q)	Kalimat
5	0.1001	Manager koperasi menganalisa dan memahami infomasi yang berhubungan dengan perilaku pelanggan adalah alat untuk memenangkan persaingan.
7	0.0958	Sistem tersebut bertugas memberikan analisa asosiasi barang sebagai dasar pertimbangan dalam merencanakan atau meramalkan barang-barang yang harus di stok dalam gudang dan menetapkan persediaan minimum
8	0.1130	Penggunaan sistem penggalian data (data mining) dengan algoritma Apriori.
11	0.0982	Apabila telah ditemukan beberapa pola assosiasi, maka dapat menentukan strategi pemasaran misalnya memberikan potongan harga pada setiap pola assosiasi atau menjadikan pola assosiasi tersebut menjadi sebuah paket penjualan dengan harga yang lebih murah di bandingkan harga ecerannya.
12	0.0958	Membandingkan hasil akhir dari kedua kondisi yaitu sebelum diambil kebijakan dan sudah diambil kebijakan dapat dianalisa tingkat keberhasilan suatu sistem
13	0.2933	Dengan maksud tersebut diatas maka dibuatlah tugas akhir dengan judul "PERANCANGAN DAN IMPLEMENTASI PENERAPAN DATA MINING UNTUK ANALISIS POLA ELASTISITAS MINAT KONSUMEN DAN PENETAPAN HARGA PADA KOPERASI DENGAN METODE APRIORI".

11. Uji Coba Sistem

Uji coba sistem dilakukan dengan menghitung nilai *recall* dan *precision*. Dalam uji coba ini digunakan 10 dokumen tugas khusus. Tabel dibawah ini menunjukan nilai perhitungan recall dan precisionnya.

DocId	Jml Kal	Jml Kal Ringkasan	Recall %)	Precision %)
1	13	6	100	80
2	13	5	100	70
3	15	7	90	75
4	18	7	85	80
5	16	7	80	80
6	15	6	80	75
7	14	7	85	70
8	16	6	85	80
9	16	7	80	80
10	17	5	100	80

PENUTUP

Kesimpulan

Dari serangkaian uraian yang telah dijabarkan pada bab sebelumnya maka dapat disimpulkan sebagai berikut :

1. Penggunaan algoritma pengolahan teks Information Retrieval dapat digunakan sebagai sistem peringkasan yang dapat mengekstraksi sebuah teks menjadi lebih ringkas seperti yang telah dijabarkan pada bab sebelumnya.
2. Berdasarkan hasil yang ditunjukan, keakuratan dalam menghasilkan ringkasan sangat dipengaruhi oleh pemilihan kata pada query.

SARAN

Dari serangkaian penghitungan pada bab sebelumnya, diharapkan sistem peringkasan ini dapat dikembangkan hingga menghasilkan ringkasan yang lebih baik, seperti :

1. Menambahkan algoritma peringkasan yang dapat digunakan dalam pengembangan sistem peringkasan seperti algoritma Maximum Marginal relevancy yang dapat digunakan sebagai algoritma yang dapat memaksimalkan algoritma Cosine Similarity seperti pada pembahasan sebelumnya.
2. Melakukan analisa lebih lanjut untuk menentukan stopword yang akan digunakan, karena pada pengembangan sistem peringkasan penentuan kata yang akan di filtering juga dapat sangat menentukan dalam pembobotan masing-masing kalimat.

DAFTAR PUSTAKA

1. Arifin, A Z, (2002). Penggunaan Digital Tree Hibrida pada Aplikasi Information Retrieval untuk Dokumen Berita, Jurusan Teknik Informatika, FTIF, Institut Teknologi Sepuluh Nopember. Surabaya
2. Budhi, G S,. Intan R,. Silivia R., Stevanus R. (2007) Indonesian Automated Text Summarization , Petra Christian University, Informatics Engineering Dept. Siwalankerto , Surabaya.
3. Hovy, E. (2001). Automated Text Summarization. In R. Mitkov (Ed.), Ebook of computation linguistics. Oxford: Oxford University Press. , Diambil 10 oktober 2012
4. Mustaqhfiri, M., Abidin Z., Kusumawati,R.(2011). Peringkasan Teks Otomatis Berbahasa Indonesia Menggunakan Metode Maximum Marginal Relevance. Ejournal Matics,4,4,135-147. Diambali 12 Oktober 2012
5. Purwasih, N. (2008).Sistem Peringkas Teks Otomatis untuk Dokumen Tunggal Berita Berbahasa Indonesia dengan Menggunakan Graph-based Summarization Algorithm dan Similarity . Departemen Teknik Informatika, Institut Teknologi Telkom Bandung
6. Tala, fadilah Z. (2003), A Study of Stemming Efects on Information Retrieval ini Bahasa Indonesia. Institute for logic, Language and Computation University itvan Amsterdam the Netherlands.
7. Iyan. M, Sena .R, Herfina (2012). Penerapan Term Frequency – Inverse Document Frequency Pada Sistem Peringkasan Teks Otomatis Dokumen Tunggal Berbahasa Indonesia, Ejournal, Diambali 14 Oktober 2012