PERANCANGAN INFORMATION RETRIEVAL (IR) UNTUK PENCARIAN IDE POKOK TEKS ARTIKEL BERBAHASA INGGRIS DENGAN PEMBOBOTAN VECTOR SPACE MODEL

DwijaWisnu B, Anandini Hetami
STMIK ASIA Malang

ABSTRAK

Artikel berbahasa Inggris merupakan artikel dengan bahasa yang paling cepat menyebar dan terbaru dikarenakan bahasa Inggris merupakan bahasa internasional yang banyak digunakan orang di dunia, namun tidak semua orang fasih berbahasa Inggris dan membutuhkan bantuan untuk mengerti isi sementara artikel yang akan dibaca sangat penting dan panjang sehingga dapat menyita waktu.

Memanfaatkan Information Retrieval pada teks mining untuk menemukan ide pokok dalam teks pada artikel berbahasa Inggris, dapat membantu pembaca untuk lebih mudah memahami isi artikel dan menghemat waktu yang dibutukan untuk membaca secara garis besar dengan memberikan sebuah konten yang lebih ringkas dari artikel awal. Basis pertama yang digunakan adalah Term Frequency Inverse Document Frequency (TF-IDF) untuk memberikan nilai dan menggunakan pembobotan Vector Space Model untuk menarik hasil dari pencarian ide pokok. Kata kunci yang digunakandalam proses peringkasan adalah judul dari artikel.

Hasil kesimpulan yang didapatkan dari sistem pencarian ide pokok otomatis ini memberikan nilai recall 66,68%, precision 72,29%, dan f-measure sebesar 70,38%.

Kata kunci: **Information Retriveal, Pencarian Ide Pokok,***Peringkasan Otomatis*,*Teks Mining*, **Vector Space Model**

ABSTRACT

English article is one of the most fastest for spreading and most update article because English is used as international language and many people learn it but not everyone is fluent in English and need helps to understand the contain of an article that will be read, meanwhile the article is important and long and can takes time to finish.

Using Information Retrieval in text mining to found the main idea in English text article can help the reader to understand the article and reducing the time that needed to read the outline with giving a content that more compact than the original one. The first base is using Term Frequency Inverse Document Frequency (TF-IDF) to give value and using Vector Space Model weighting to get the result for the main idea. The keyword that used in process are the article title.

The Result for the main idea search systemgive 66,68% recall, 72,29% recall, 70,38% f-measure as marginal value.

Kata kunci: *Automatic summarization*, Information Retriveal, Main Idea Searching, *Text Mining*, Vector Space Model

PENDAHULUAN

Artikel adalah salah satu informasi berupa tulisan yang berisikan informasi, gagasan pikiran, atau pendapat penulis yang berguna bagi pembaca. Dalam dunia Internet terdapat banyak sekali informasi-informasi berupa artikel yang tersebar luas dan dalam berbagai bahasa. Salah satu jenis artikel yang banyak dijumpai di dunia maya adalah artikel tentang komputer dan teknologi yang merupakan topik yang banyak dibicarakan di era teknologi modern saat ini.

Salah satu cara untuk peringkasan yang sudah diterapkan adalah Peringkasan Teks Otomatis (*Automated Text Summarization*) atau sering disebut *Text Summarization*, yaitu sebuah proses untuk menghasilkan ringkasan dari teks menggunakan komputer. Tujuannya untuk mengambil sumber infomasi dengan mengutip sebagian besar isi yang penting dan menampilkan kepada pembaca dalam bentuk ringkas sesuai dengan kebutuhan pembaca. Dengan demikian teknologi ini dapat membantu pembaca untuk menyerap informasi yang terdapat dalam teks melalui ringkasan tanpa membaca keseluruhan dokumen.

Vector Space Model (VSM) sebagai metode yang mengukur kemiripan antara suatu dokumen dengan suatu query user dengan menggunakan cosinus dari sudut antar vektor yang dibentuk oleh dokumen dengan vektor dari

kata kunci yang diinputkan oleh user dalam hal ini kata kunci adalah judul dari artikel.

Data yang digunakan adalah teks artikel berbahasa Inggris yang dapat disimpan dalam format .txt dapat berupa artikel dalam laman di Internet ataupun *ebook* yang memungkinkan untuk disalin isinya. Peringkasan dilakukan per artikel dengan basis pertama adalah TF-IDF sedangkan basis kedua VSM. Peringkasan dilakukan per artikel dengan hasil ektsraksi, kalimat yang dianggap sebagai inti teks mewakili ide pokok akan diambil.

KAJIAN TEORI

INFORMATION RETRIEVAL (SISTEM TEMU KEMBALI)

Information Retrieval merupakan sistem yang menerima *query* dari pengguna, kemudian dilakukan ranking terhadap dokumen berdasar kesesuian terhadap *query*. Hasil ranking yang diberikan pada pengguna merupakan dokumen yang menurut sistem memiliki relevansi terhadap *query*, tetapi tingkat relevansi itu sendiri merupakan hal yang subjektif tergantung dari pengguna yang dipengaruhii oleh berbagai macam faktor seperti topik, pewaktuan, sumber informasi maupun tujuan pengguna. Model sistem temu kembali menentukan detail sistem temu yaitu meliputi representasi dokumen maupun *query*, fungsi pencarian (*retrieval function*), dan notasi kesesuaian (*relevance notation*) dokumen terhadap *query*.

Information Retrieval terbagi dari beberapa bagian yang dijabarkan sebagai berikut:

1. *Text Operations*, meliputi pemilihan kata-kata dalam query maupun dokumen (*term selection*) dalam proses transformasi dokumen atau query menjadi *term index* (indeks kata-kata).
2. *Query formulation*, memberi bobot pada indeks kata-kata query.
3. *Ranking*, mencari dokumen-dokumen yang relevan terhadap query dan mengurungkan dokumen tersebut berdasarkan kesesuaiannya dengan query.
4. *Indexing*, membangung basis data indeks dari koleksi dokumen Dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan.

DATA MINING

Sistem Manajemen Basis Data tingkat lanjut dan teknologi *datawarehousing* mampu untuk menampilkan "banjir" data dan untuk mentransformasikannya ke dalam basis data yang berukuran besar. Diperlukan teknik baru yang secara pintar dan otomatis mentransformasikan data-data yang diproses untuk menghasilkan pengetahuan dan informasi yang berguna. *Data mining* adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Kata *mining* berarti usaha untuk mendapatkan sedikit barang berharga dari sejumlah besar material dasar **(Pramudiono:2003)**.

Secara sederhana *data mining* dapat dikatakan sebagai proses menyaring atau "menambang" pengetahuan suatu pengetahuan baru dari sejumlah data yang besar menggunakan serangkaian proses matematika. Hal penting yang terkait *data mining* adalah:

a. *Data mining* merupakan proses otomatis terhadap data yang ada.
b. Data yang akan diproses berupa data yang sangat besar.

Tujuan *data mining* adalah mendapatkan hubungan atau pola yang mungkin memberikan indikasi bermanfaat.

TEXT MINING

Text mining merupakan salah satu bidang khusus dari *data mining*. *Text mining* dapat didefiniskan sebagai suatu proses menggali informasi dimana seorang user berinteraksi dengan sekumpulan dokumen menggunakan *tool* analisis yang merupakan komponen-komponen dalam *data mining* **(Han dan Kamber:2006)**.

Dalam *text mining* berbeda dengan dengan *data mining* dimana *data mining* yang digunakan adalah *structured data* sementara dalam *text mining* umumnya data yang ditemui adalah *semi-structured* atau *unstructured*. Sementara keduanya memiliki permasalah yang sama yaitu jumlah data yang besar, dimensi yang tinggi, dan data juga struktur yang terus berubah.Struktur teks yang kompleks dan tidak lengkap, arti yang tidak jelas dan tidak standar, dan bahasa yang berbeda ditambah terjemahan yang tidak akurat memberikan tantangan tambahan pada *text mining.*

Text mining dalam prakeknya mencari pola-pola tertentu, mengasosiasikan satu bagian teks dengan lain berdasar aturan-aturan tertentu, kata-kata yang dapat mewakili sehingga dapat dilakukan analisa keterhubungan antar satu dengan lain,dalam kumpulan dokumen yang sangat banyak. Dokumen yang ada bisa bersifat statis, yaitu dokumen yang tidak akan di perbarui lagi ataupun dinamis yaitu dokumen yang akan selalu diperbarui dalam rentang waktu tertentu.

1. Tahapan *Text Mining*

a. Case Folding

Mengubah semua huruf dalam dokumen menjadi huruf kecil (*lowercase*). Dalam tahap ini juga karakter selain huruf dihilangkan.

b. Tokenizing
Memotong tiap kata dalam kalimat atau *parsing* dengan menggunakan spasi sebagai delimiter yang akan menghasilkan token berupa kata.

c. Filtering
Menyaring kata yang didapat dari proses *tokenizing* yang dianggap tidak penting atau tidak memiliki makna dalam proses *text mining* yang disebut *stoplist*. *Stoplist* atau *stopword* berisi kata-kata umum yang sering muncul dalam sebuah dokumen dalam jumlah banyaknamun tidak memiliki kaitan dengan dengan tema tertentu. Tiap kata yang diperoleh dari *tokenizing* akan dicocokkan dalam kamus *stopword* di dalam *database*, jika kata tersebut cocok dengan salah satu kata dalam *stopword* maka kata tersebut akan dihilangkan, sementara yang tidak cocok akan dianggap cocok dan diproses ke tahap selanjutnya.

d. Stemming
Mengembalkan kata-kata yang diperoleh dari hasil *filtering* ke bentuk dasarnya, menghilangkan imbuhan awal (*prefix*) dan imbuhan akhir (*sufix*) sehingga di dapat kata dasar.Metode stemming memerlukan masukan berupa kata yang terdapat dalam suatu dokumen, dengan menghasilkan keluaran berupa *root word*.

e. Tagging
Merubah kata dalam bentuk lampau (*past tense*) menjadi bentuk sekarang (*future tense*).

f. Analyzing
Keterhubungan antar kata dalam dokumen akan ditentukan dengan menghitung frekuensi term pada dokumen. Tahap *analyzing* lebih sering dikenal dengan tahap pembobotan.

TF-IDF (*TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY*)

Basis pembobotan TF-IDF adalah jenis pembobotan yang sering digunakan dalam IR 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,

$$TF-IDF\left(t_k,d_j\right)=TF\left(t_k,d_j\right)*IDF\left(t_k\right) \quad (2,1)$$

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.

$$TF\left(t_k,d_j\right)=f\left(t_k,d_j\right) \quad (2,2)$$

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

Lalu untuk menghitung nilai IDF bisa menggunakan persamaan sebagai berikut, yaitu :

$$IDF\left(t_k\right)=\frac{1}{df\left(t\right)} \quad (2,3)$$

Atau

$$IDF\left(t_k\right)=\log\frac{N}{df\left(t\right)} \quad 2,4)$$

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

Persamaan 2,3 hanya boleh digunakan apabila hanya terdapat satu buah dokumen saja yang diproses sedangkan persamaan 2,4 digunakan pada proses yang melibatkan banyak dokumen.

VECTOR SPACE MODEL

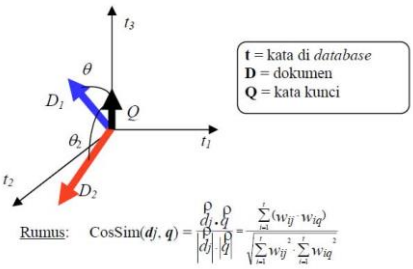
Model ruang vektor dibuat berdasarkan pemikiran bahwa isi dari dokumen ditentukan oleh kata-kata yang digunakan dalam dokumen tersebut. Model ini menentukan kemiripan (*similarity)* antara dokumen dengan *query* dengan cara merepresentasikan dokumen dan *query* masing-masing ke dalam bentuk vektor.

Tiap kata yang ditemukan pada dokumen dan *query* diberi bobot dan disimpan sebagai salah satu elemen vektor.

Kemiripan antar dokumen didefinisikan berdasarkan representasi *bag-of-words* dan dikonversi ke suatu model ruang vektor (*vector space model*, VSM). Pada VSM, setiap dokumen di dalam database dan query pengguna direpresentasikan oleh suatu vektor multi-dimensi. Dimensi sesuai dengan jumlah term dalam dokumen yang terlibat pada model ini:

1. *Vocabulary* merupakan kumpulan semua *term* berbeda yang tersisa dari dokumen setelah *preprocessing* dan mengandung *t term index*. *Term-term* ini membentuk suatu ruang vektor.
2. Setiap *term i* di dalam dokumen atau *query j*, diberikan suatu bobot (*weight*) bernilai *real* W*ij.*
3. Dokumen dan *query* diekspresikan sebagai vektor *t* dimensi *dj* = (W1, W2, ..., W*tj*) dan terdapat *n* dokumen di dalam koleksi, yaitu *j* = 1, 2, ..., n.

Contoh dari model ruang vektor tiga dimensi untuk dua dokumen D1 dan D2, satu query pengguna Q1, dan tiga term T1, T2 dan T3 diperlihatkan pada gambar 2.14.



Gambar 1VSM

Dalam model ruang vektor, koleksi dokumen direpresentasikan oleh matriks *term-document* (atau matriks *term-frequency*). Setiap sel dalam matriks bersesuaian dengan bobot yang diberikan dari suatu term dalam dokumen yang ditentukan. Nilai nol berarti bahwa *term* tersebut tidak hadir di dalam dokumen.Contoh matrik *term-document* untuk database dengan n dokumen dan t term berikut adalah gambar matriks term document:

$$\begin{bmatrix} & T_1 & T_2 & \cdots & T_t \\ D_1 & w_{11} & w_{21} & \cdots & w_{t1} \\ D_2 & w_{12} & w_{22} & \cdots & w_{t2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ D_n & w_{1n} & w_{2n} & \cdots & w_{tn} \end{bmatrix}$$

Gambar 2 Contoh Matriks VSM

Dokumen-dokumen yang panjang sering dianggap lebih relevan dibandingkan dokumen yang pendek, padahal belum tentu demikian. Untuk mengurangi pengaruh perbedaan panjang dokumen ini, pada pembobotan kata digunakan satu faktor lagi yang disebut sebagai normalisasi panjang dokumen. Normalisasi yang digunakan adalah normalisasi kosinus. Berdasarkan rumus normalisasi kosinus yaitu :

$$\frac{\Sigma(w_{ik} w_{jk})}{\sqrt{\Sigma_k w_{ik}^2} * \sqrt{\Sigma_k w_{jk}^2}}$$

Dengan W adalah bobot dari query dan dokumen.

Setelah mendapatkan nilai cosine tiap-tiap dokumen, maka hasil bobot dari kata kunci diurutkan. Bobot yang besar menjadi prioritas sebagai dokumen yang memiliki hubungan dengan kata kunci.

METODE PENELITIAN

1. **Studi Pustaka (*Library Research*)**
   Studi Pustaka dilakukan dengan cara mempelajari teori-teori literatur dan buku-buku yang berhubungan dengan objek kajian sebagai dasar dalam penelitian ini, dengan tujuan memperoleh dasar teoritis gambaran dari apa yang dilakukan. Teori yang dipelajari yaitu: *text mining*, *Term Frequency-Inverse Document Frequency* (TF-IDF), vector space model, Porter Stemming, metode tagging, dansebagainya.
2. **Melakukan kajian secara *online* di Internet**
   Browsing pada halaman-halaman situs di Internet yang membahas tentang algoritma-algoritma yang akan digunakan dalam pembuatan program, seperti contoh algoritma untuk melakukan stemming dan tagging. Browsing juga dilakukan untuk mengumpulkan *ebook* ataupun artikel yang akan dibutuhkan dalam proses peringkasan.
3. **Analisa data**
   Penelitian dilakukan menggunakan berkas data yang didapat untuk selanjutnya dilakukan analisa dan pembandingan dengan ringkasan yang sudah dibuat secara manual. Setelah dilakukan pengumpulan data, tahap selanjutnya dilakukan studi pustaka dan analisa atas data yang sudah diperoleh untuk membuat perancangan dan implementasi aplikasi *Information Retrieval* (IR) untuk peringkasan teks otomatis pada

teks artikel berbahasa Inggris untuk kemudian dilakukan evaluasi.

IMPLEMENTASI DAN PENGUJIAN

Pengujian dilakukan dengan menggunakan 10 artikel berbahasa Inggris yang diambil dari e-book "Bypassing Censorship" pada website http://en.flossmanuals.net/bypassing-censorship/. Data yang akan diuji adalah sebagai berikut, yaitu :

No.	Judul	Halaman
1	Who Exactly Is Blocking My Access To The Internet? Query: bypassing, censorship, who, exactly, is, blocking, my, access, to, the, internet?	4
2	How the Net Works Query: bypassing, censorship, how, the, net, works, network, connection	13
3	Am I Being Blocked or Filtered? Query: bypassing, censorship, and, the, net, am, i, being, blocked, or, filtered	20
4	Port Blocking Query: bypassing, censorship, and, the, net, port, blocking	25
5	Risks when not using HTTPS Query: bypassing, censorship, circumvention, and, safety, risks, when, not, using, https, confidentiality	33
6	Certificate Warning Query: bypassing, censorship, circumvention, and, safety, certificate, warning, confidentiality, https	33
7	Use Alternative ISPs Query: bypassing, censorship, get, creative, use, alternative, isps	41
8	Any Communication Channel Could Be A Circumvention Channel Query: bypassing, censorship, get, creative, any, communication, channel, could, be, a, circumvention, channel	42
9	Obfuscation is not Encryption Query: bypassing, censorhip, web, proxies, security risks, with, web, proxies, obfuscation, is, not, encryption	47
10	The History of Psiphon Query: bypassing, censorship, psiphon, the, history, of	49

Setelah dilakukan uji coba pada 10 artikel dengan judul yang berbeda diperoleh hasil uji coba seperti terlihat pada tabel berikut:

No.	Jumlah Kalimat	Jumlah Kalimat Ringkasan	No.Kalimat hasil Ringkasan
1	12	10	1,2, 3, 4, 5, 7, 9, 10, 11, 12
2	21	15	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 17, 18, 19, 20
3	15	7	2, 4, 7, 8, 9, 11, 12
4	11	10	1, 2, 3, 5, 6, 7, 8, 9, 10
5	12	5	1, 2, 6, 7, 12
6	9	7	1, 2, 3, 4, 6, 7, 8, 9
7	8	7	1, 3, 4, 5, 6, 7, 8
8	17	7	1, 2, 3, 8, 9, 15, 16
9	21	12	1, 5, 6, 10, 13, 14, 15, 16, 17, 18, 19, 20
10	18	12	1, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 18

Sementara secara manual diperoleh:

No.	Jumlah Kalimat	Jumlah Kalimat Ringkasan	No.Kalimat hasil Ringkasan
1	12	8	1, 2, 3, 4, 9, 10, 11, 12
2	21	12	2, 3, 5, 6, 1, 10, 11, 12, 17, 18, 20, 21
3	15	9	1, 2, 3, 8, 9, 10, 12, 13, 14

4	11	8	1, 2, 3, 5, 6, 8, 9, 10
5	12	5	1, 4, 5, 6, 7
6	9	6	1, 2, 4, 7, 8, 9
7	8	5	2, 3, 4, 5, 7
8	17	10	1, 2, 4, 5, 8, 10, 14, 15, 16, 17
9	21	13	1, 2, 4, 5, 6, 7, 8, 11, 12, 16, 18, 19, 20
10	18	12	1, 2, 3, 4, 5, 7, 9, 10, 12, 13, 17, 18

Setelah dilakukan pengujian pada perangkat lunak, pada tahap berikut akan dijelaskan pengukuran terhadap kualitas hasil ringkasan yang dihasilkan. Untuk mengukur kualitas hasil ringkasan maka dilakukan perbandingan hasil antara ringkasan yang dihasilkan oleh sistem dengan ringkasan yang dihasilkan secara manual.

No.	**Jumlah Kalimat**	**Ringkasan Sistem (A)**	**Ringkasan Manual (B)**	**Kemiripan (A∩B)**
1	12	1,2, 3, 4, 5, 7, 9, 10, 11, 12	1, 2, 3, 4, 9, 10, 11, 12	1, 2, 3, 4, 9, 10, 11, 12
2	21	1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 17, 18, 19, 20	2, 3, 5, 6, 1, 10, 11, 12, 17, 18, 20, 21	2, 3, 5, 10, 11, 12, 17, 18, 20
3	15	2, 4, 7, 8, 9, 11, 12	1, 2, 3, 8, 9, 10, 12, 13, 14	2, 8, 9, 12
4	11	1, 2, 3, 5, 6, 7, 8, 9, 10	1, 2, 3, 5, 6, 8, 9, 10	1, 2, 3, 5, 6, 8, 9, 10
5	12	1, 2, 6, 7, 12	1, 4, 5, 6, 7	1, 6, 7
6	9	1, 2, 3, 4, 6, 7, 8, 9	1, 2, 4, 7, 8, 9	1, 2, 4, 7, 8, 9
7	8	1, 3, 4, 5, 6, 7, 8	2, 3, 4, 5, 7	3, 4, 5, 7
8	17	1, 2, 3, 8, 9, 15, 16	1, 2, 4, 5, 8, 10, 14, 15, 16, 17	1, 2, 8, 15, 16
9	21	1, 5, 6, 10, 13, 14, 15, 16, 17, 18, 19, 20	1, 2, 4, 5, 6, 7, 8, 11, 12, 16, 18, 19, 20	1, 5, 6, 16, 18, 20
10	18	1, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 18	1, 2, 3, 4, 5, 7, 9, 10, 12, 13, 17, 18	1, 4, 5, 7, 9, 10, 12, 13, 18

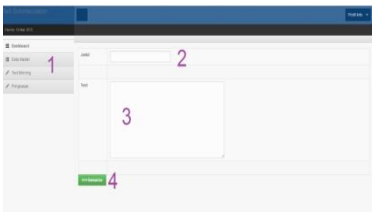
Setelah semua hasil ringkasan dijadikan satu tabel untuk dilakukan perbandingan maka selanjutnya hasil perbandingan tersebut akan dilakukan perhitungan untuk mencari nilai *recall* dan *precision* nya.

Setelah didapatkan nilai *recall* dan *precision* untuk masing-masing hasil ringkasan maka selanjutnya dilakukan penghitungan *f-measure* untuk mendapatkan nilai kemiripan yang akan dihadikan sebagai tolak ukur untuk menentukan kelayakan hasil peringkasan.

No.	**Recall (%)**	**Precision (%)**	$\mathbf{f-measure = \frac{2 * recall * prec}{recall + preci} * 100\%}$
1	80%	100%	88,89%
2	60%	75%	66,67%
3	57,14%	44,44%	49,99%
4	80%	100%	88,89%
5	60%	60%	60%
6	85,71%	100%	92,30%
7	57,14%	80%	85,33%
8	71,42%	50%	58,82%
9	41,66%	38,46%	39,99%
10	75%	75%	75%

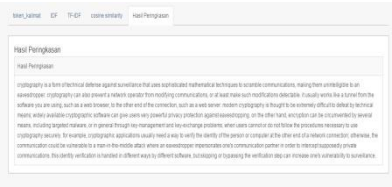
Dari 10 dokumen yang dilakukan proses peringkasan teks secara manual dan peringkasan menggunakan sistem menghasilkan nilai rata-rata *recall* 66,86% nilai rata-rata *precision* 72,29% dan nilai rata-rata *f-measure* 70,38%. Pada evaluasi didapatkan nilai tertinggi recall 85,71% dengan *precision* 100% dan *f-measure* 92,30% lalu didapatkan juga nilai terendah *recall* 41,66% dengan *precision* 38,46% dan *f-measure* 39,99%.

Tampilan aplikasi pencarian ide pokok adalah sebagai berikut:



Gambar 3 Tampilan *preprocessing data input text*

1. menu halaman.
2. *text field* untuk mengisi judul (*query*).
3. *text box* untuk mengisi teks awal yang akan diringkas.
4. tombol "summarize" berfungsi untuk memproses teks yang telah diinputkan.



Gambar 4 Tampilan Hasil Ringkasan

KESIMPULAN

Dari apa yang sudah dijabarkan pada sebelumnya, maka dapat diambil kesimpulan sebagai berikut:

1. Dari hasil perhitungan TF-IDF yang telah dijalankan pada sistem dapat disimpulkan bahwa semakin banyak kata yang sama ditemukan pada sebuah kata maka nilai TF-IDF akan semakin tinggi, tapi bila kata yang sama ditemukan pada kalimat lain maka nilai TF-IDF nya akan semakin rendah.
2. Sistem masih belum bisa mengenali mana alamat sebuah web atau sebuah kata yang memerlukan titik di tengah kalimat dengan limiter titik yang digunakan untuk memecah kalimat sehingga terkadang sistem memecah sebuah alamat web menjadi kalimat tersendiri yang dapat memengaruhi hasil peringkasan karena dimungkinkan memunculkan kalimat yang tidak utuh.
3. Dengan nilai rata-rata *recall* sebesar 66,86%, *precision* 72,29%, *f-measure* 70,38% hasil peringkasan belum bisa dianggap optimal, salah satu faktor yang mempengaruhi adalah query yang digunakan, diperlukan tambahan query yang relevan dengan artikel untuk hasil peringkasan yang lebih optimal.

DAFTAR PUSTAKA

[1] Arifin, A Z. Penggunaan Digital Tree Hibrida pada Aplikasi Information Retrieval untuk Dokumen Berita. Jurusan Teknik Informatika, FTIF, Institut Teknologi Sepuluh Nopember. Surabaya. 2002

[2] Grossma D, Ophir F. Information Retrieval : Algorithm and Heuristics. Kuwer Academic Publisher. 1998

[3] Han, Jiawei dan Kamber, Micheline. Data Mining : Concept and Techniques Second Edition. Morgan Kaufmann Publishers. 2006

[4] Herwansyah, Adit. Aplikasi Pengkategorian Dokumen Dan PengukuranTingkat Similaritas Dokumen Menggunakan Kata Kunci Pada Dokumen Penulisan Ilmiah Universitas Gunadarma.http://www.gunadarma.ac.id/library/articles/graduate/computer-science/2009/Artikel_10105046.pdf

[5] Karmayasa, Oka. Mahendra, Ida Bagus. Implementasi Vector Space Model Dan Beberapa Notasi Metode Term Frequency Inverse Document Frequency (TF-IDF) Pada Sistem Temu Kembali Informasi. http://ojs.unud.ac.id/index.php/JLK/article/download/2787/1981

[6] Manning Christopher D, Prabhakar Raghavan dan Hinrich Schutze. An Introduction to Information Retrieval. England. Cambgridge University Press. 2009