BAB II

LANDASAN TEORI

2.1 Dokumen Digital

Dokumen adalah suatu aset instansi yang menyimpan berbagai informasi penting dan merupakan pertanggungjawaban kerja yang harus selalu dipelihara, dirawat dan dilindungi keberadaannya.Pada pengertian umum, kata dokumen biasanya diartikan sebagai *informationcarrier* (biasanya pada kertas) yang mengandung tulisan atau gambar informasi untuk tujuan tertentu.Peningkatan dari jumlah dokumen yang diproduksi oleh kegiatan organisasi membuat manajemen dokumen menjadi hal yang sangat penting (Setiawan, 2010).

Dokumen digital merupakan suatu dokumen yang berupa informasi elektronik yang dapat diakses melalui komputer atau media elektronik lainnya.

Contoh format dokumen digital yang ada seperti :

A. Format Dokumen *MicrosoftWord* (*.docx*)

Dokumen digital dalam format *docx* dirancang untuk membuat dokumen yang kontennya dapat diakses dengan mudah.*File* docx dapat dibuka dengan beragam program perangkat lunak keluaran *Microsoft* dalam paket *MicrosoftOffice*.

File *docx* merupakan kumpulan *filexml* yang dikompres. *Filedocx* menyimpan dokumen data dalam *filebinary* tunggal, *filedocx* diciptakan menggunakan format *OpenXML* yang menyimpan dokumen sebagai sebuah kumpulan *file* dan *folder* yang terpisah dalam paket kompresi *zip*. *Filedocx* berisi file *xml* dan tiga *folder*, yakni *docProps*, *Word* dan *_rels* yang menjaga dokumen *properties*, konten dan hubungan di antara *file* (Enterprise, 2010).

B. Format Dokumen *RTF* (*RichTextFormat*)

Format *RTF* merupakan sebuah dokumen digital dalam bentuk teks yang disimpan dalam format *richtext*. Format *richtext* yang dimaksudkan dalam konteks ini, yakni dalam *file* tersebut terdapat beragam tipe format teks. *FileRTF* sebagai standar format *file* teks dapat dibuka dengan program pengolah kata yang banyak beredar seperti *MicrosoftWord, Wordpad, CorelWordPrefect* dan lain-lain. Oleh sebab itu, *file* ini bisa umumnya memiliki opsi ekspor yang dapat digunakan untuk mentransfer format *file* antar-program yang berbeda (Enterprise, 2010).

C. Format Dokumen *TXT*

FileTXT merupakan standar dokumen teks yang berisi rangkaian teks yang tidak terformat. Jenis dokumen ini diakui oleh sembarang program *editing* teks atau pengolah kata (Enterprise, 2010). Dokumen digital dengan format ini biasa dihasilkan oleh program *Windows* yang bernama *Notepad*. Dokumen dengan format *TXT* biasanya juga digunakan sebagai masukan dan keluaran dari beragam program.

2.2 Information Retrieval (IR)

Seni dan ilmu dalam mencari informasi pada dokumen, mencari untuk dokumen mereka sendiri, mencari untuk *metadata* dengan gambaran berbentuk dokumen, atau mencari dalam *database*, apakah itu hubungan *database* yang berdiri sendiri atau hiperteks jaringan *database* seperti internet atau intranet, untuk teks, suara, gambar atau data.(Muchlisin,2012).

2.3 Stemming

Stemming merupakan suatu proses yang terdapat dalam sistem *IR* / *InformationRetrieval* yang mentransformasikan kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya atau kata dasarnya (*rootword*) dengan menggunakan aturan – aturan tertentu (Agusta, 2009).

Contoh :

A. Berlayar => Memiliki awalan / *prefix* ber- yang berfungsi menandakan mempunyai / memiliki dan memiliki kata dasar ‘layar’.

B. Memberikan => Memiliki awalan / *prefix* dan akhiran / *suffix*Me – Kan yang memiliki arti sedang melakukan dengan kata dasarnya ‘beri’.

C. Persempit => Memiliki awalan / *prefix* per- yang artinya membuat jadi lebih dengan kata dasar ‘sempit’.

2.4 Pencarian / *Searching*

Algoritma pencarian / *searching* merupakan algoritma yang menerima sebuah *input* masukan data yang berisi kata kunci dari informasi yang akan dicari dan dengan langkah – langkah tertentu algoritma ini akan mencari data yang relevan dari kata kunci yang telah dimasukkan dan hasil keluarannya memiliki satu kemungkinan yaitu data yang dicari ditemukan atau tidak ditemukan. Dua algoritma pencarian / *searching* yang sering digunakan adalah *linear search* dan*binary search (*Deitel, 2010*).*

A. *Sequential Search / Linear Search*

Metode *Sequential Search* atau yang disebut pencarian beruntun yang dapat digunakan untuk melakukan pencarian data, baik pada *array* yang sudah terurut maupun yang belum terurut (Utami, 2005). Langkah kerjanya yaitu :

A.1 Membaca *Array* Data

A.2 Menentukan data yang dicari

A.3 Mulai dari data pertama sampai dengan data

terakhir, data yang dicari dibandingkan dengan masing-masing data dalam array

Contoh :

Terdapat 6 buah data yang disimpan dalam *array*, yaitu : 8, 7, 5, 6 ,10, 4 dan yang akan dicari dari array itu adalah 5

A = [0] [1] [2] [3] [4] [5]

8 7 5 6 10 4

Pada Loop 1 (i = 0)

Jika (A[i] = X) Tidak i++

Loop 2 (i=1)

(A[i] = X) Tidak, i++

Loop 3 (i=2)

(A[i] = X) T=Ya, X ditemukan di array ke 2

B. Binary Search

Metode pencarian ini hanya digunakan untuk pencarian data pada *array* / kumpulan data yang telah terurut (Utami, 2005).

Langkah kerjanya yaitu :

B.1 Menentukan data yang akan dicari dari *array* yang telah diurutkan.

B.2 Menetukan elemen tengah dari array. Letak elementengah dapat dicari

dengan rumus (n div 2) + 1. Untuk array yang banyaknya data adalah genap ,posisitengahnya tidak tepat berada di tengah.

B.3 Jika nilai elemen tengah sama dengan data yang dicari , maka pencarian selesai.

B.4 Jika elemen tengah tidak sama dengan data yang dicari, maka:

- Nilai elemen tengah lebih besar dari data yang akan dicari maka proses akan diulang pada setengah *array* pertama
- Nilai elemen tengah lebih kecil dari data yang akan dicarimaka proses akan diulang pada setengah *array* kedua.

2.5 Algoritma Stemming Nazief Adriani

Algorima *stemming* untuk bahasa yang satu berbeda dengan algoritma *stemming* untuk bahasa lainnya. Sebagai contoh bahasa Inggris memiliki morfologi yang berbeda dengan bahasa Indonesia sehingga algoritma *stemming* bahasa tersebut juga berbeda. Proses *stemming* pada teks bahasa Indonesia lebih rumit / kompleks karena terdapat variasi imbuhan yang harus dibuang untuk mendapatkan *root word*(kata dasar) dari sebuah kata (Dyan, 2012).

Pada umunya kata dasar pada bahasa Indonesia terdiri dari :

[AW + [AW + [AW +]]] kata dasar [[+AK] [+KK] [+P]]

Keterangan :

AW : Awalan

AK : Akhiran

KK : Kata ganti kepunyaan

P : Partikel

Berikut aturan *Stemming* kata bahasa Indonesia menggunakan algoritma Nazief Adriani :

A. Kata yang belum di *stem* akan dicari terlebih dahulu dalam kamus kata dasar.Apabila ditemukan maka algoritma berhenti.

B. Hilangkan *Suffix* / Akhiran (“-lah”,”-kah”,”-ku”,”-mu” atau “-nya”).Apabila ditemukan partikel (“-lah”,”-kah”, “-tah”, “-pun”), hapus terlebih dahulu, maka langkah ini diulang kembali untuk menghapus kata ganti kepunyaan / KK (“-ku”,”-mu”,”-nya”) jika ada. Contoh : kata “bajumulah”, proses *stemming* pertama akan menghapus “-lah” sebagai partikel, lalu dilanjutkan menghapus “-mu” sebagai kata ganti kepunyaan / KK nya. Jika kata ditemukan dalam kamus proses berhenti, jika tidak akan lanjut ke tahap berikutnya.

   Hingga bentuk kerangka kata menjadi :

   [AW + [AW + [AW +]]] kata dasar [+AK]

C. Hilangkan juga (AK) *suffix* atau akhiran (“-i”,”-an”,”-kan”) jika ada, sehingga hasil kerangka kata menjadi :

   [AW + [AW + [AW +]]] kata dasar

   Sehingga pada tahap C ini susunan kata sudah tidak memiliki *suffix* atau akhiran..

   Contoh : Kata “membelikan” di *stemming* akan menjadi “membeli”, karena tidak ada di kamus kata dasar maka akan dilakukan penghilangan *prefix* atau awalan pada langkah selanjutnya.

D. Hapus *derivation prefix* / awalan (“di-”,”ke-”,”se-”,”me-”,”be-”,”pe-”,”te-”), jika pada langkah C ada *suffix* / akhiran yang dihapus maka masuk ke langkah D.1, jika tidak masuk ke langkah D.2.

D.1 Periksa table kombinasi awalan – akhiran yang tidak diijinkan .Jika ditemukan maka algoritma berhenti, jika tidak lanjut ke langkah D.2

Tabel 2.1 : Kombinasi Awalan Akhiran yang tidak diijinkan

(Sumber : Dyan dkk, 2012)

Awalan	Akhiran yang tidak diizinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

D.2 Awalan yang dideteksi saat ini sama dengan awalan yang dihilangkan sebelumnya.

D.3 Tiga Awalan telah dihilangkan

Untuk dua tipe awalan yang dideteksi yaitu :

Standar : “di-”,”ke-”,”se-” yang dapat langsung dihilangkan dari kata.

Kompleks : “me-”,”be-”,”pe-”,”te-” adalah tipe-tipe awalan yang dapat bermorfologi sesuai kata dasar yang mengikutinya. Oleh karena itu, gunakan aturan pada tabel 2 untuk mendapatkan pemenggalan kata yang tepat.

Tabel 2.2 : Aturan pemenggalan Awalan *Stemmer* Nazief Adriani

(Sumber :Dyan dkk, 2012)

Aturan	Format Kata	Pemenggalan
1	berV...	ber-V... | be-rV...
2	berCAP...	ber-CAP... dimana C!= 'r' & P!= 'er'
3	berCAerV...	ber-CaerV... dimana C!= 'r'
4	belajar	bel-ajar
5	beC_1erC_2...	be-C_1erC_2... dimana C_1!={'r'|'l'}
6	terV...	ter-V... | te-rV...
7	terCerV...	ter-CerV... dimana C!= 'r'
8	terCP...	ter-CP... dimana C!='r' dan P!='er'
9	teC_1erC_2...	te-C_1erC_2... dimana C_1!='r'
10	me{l|r|w|y}V...	me-{l|r|w|y}V...
11	mem{b|f|v}...	mem-{b|f|v}...
12	mempe{r|l}...	mem-pe...
13	mem{rV|V}...	me-m{rV|V}... | me-p{rV|V}...
14	men{c|d|j|z}...	men-{c|d|j|z}...
15	menV...	me-nV... | me-tV
16	meng{g|h|q}...	meng-{g|h|q}...
17	mengV...	meng-V... | meng-kV...
18	menyV...	meny-sV...
19	mempV...	mem-pV... dimana V!='e'
20	pe{w|y}V...	pe-{w|y}V...
21	perV...	per-V... | pe-rV...
23	perCAP	per-CAP... dimana C!='r' dan P!='er'
24	perCAerV...	per-CAerV... dimana C!='r'
25	pem{b|f|V}...	pem-{b|f|V}...
26	pem{rV|V}...	pe-m{rV|V}... | pe-p{rV|V}...
27	pen{c|d|j|z}...	pen-{c|d|j|z}...
28	penV...	pe-nV... | pe-tV...
29	peng{g|h|q}...	peng-{g|h|q}...
30	pengV...	peng-V... | peng-kV...
31	penyV...	peny-sV...
32	pelV...	pe-lV... kecuali "pelajar" yang menghasilkan "ajar"
33	peCerV...	per-erV... dimana C!={r|w|y|l|m|n}
34	peCP...	pe-CP... dimana C!={r|w|y|l|m|n} dan P!='er'

D4. Cari kata yang telah dihilangkan awalannya ini di dalam kamus kata dasar. Apabila ditemukan , maka keseluruhan proses berhenti.

E. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai kata dasar. Proses selesai.

Untuk mengatasi keterbatasan keterbatasan pada algoritma tersebut, maka ditambahkan aturan – aturan dibawah ini :

A. Aturan untuk reduplikasi.

- Jika kedua kata yang dihubungkan oleh kata penghubung adalah kata sama, maka kata dasar adalah bentuk tunggalnya. Contoh : kata “buku-buku”, kata dasarnya buku
- Kata lain, misalnya “bolak-balik”, “berbalas-balasan” dan “seolah-olah”. Untuk mendapatkan kata dasarnya, kedua kata diartikan terpisah. Jika keduanya memiliki kata dasar yang sama maka diubah menjadi bentuk tunggal contoh kata “berbalas-balasan” memiliki kata dasar yang sama yaitu “balas”. Sebaliknya kata “bolak-balik” memiliki kata dasar “bolak” dan “balik” yang berbeda arti, sehingga kata dasarnya tetap “bolak-balik”.

B. Tambahan bentuk awalan dan akhiran serta aturannya.

- Untuk tipe awalan “mem-”, kata yang diawali dengan awalan “memp-” tetap memiliki awalan “mem-”
- Tipe awalan “meng-”, dengan kata yang diawali dengan awalan “mengk-” tetap memiliki awalan “meng-”.

2.6 Relevansi

Relevansi adalah suatu sifat yang terdapat pada dokumen yang dapat membantu pengarang dalam memecahkan kebutuhan akan informasi. Dokumen dinilai relevan bila dokumen tersebut mempunyai topic yang sama, atau berhubungan dengan subjek yang diteliti (*topical relevance*)(Green, 2001).

2.6 .1 Cosine Similarity

Cosinesimilarity adalah metode *similarity* yang paling banyak digunakan untuk menghitung *similarity* dua buah dokumen (Pang Ning, 2006).

Rumus yang digunakan yaitu : cos (x,y) = $\frac{\mathbf{x}.\mathbf{y}}{||\mathbf{x}||||\mathbf{y}||}$**……(2.1)**

Dimana :

x.y = vektor *dot product* dari x dan y, dihitung dengan $\sqrt{\sum_{k=1}^{n} x_k y_k}$

||x|| = panjang vektor x, dihitung dengan $\sqrt{\sum_{k}^{n} x_k^2}$

||y|| = panjang vektor y, dihitung dengan$\sqrt{\sum_{k}^{n} y_k^2}$

2.7 Metode Evaluasi Paice

Metode ini berfungsi untuk menilai suatu kualitas algoritma *stemming*, karena tiap algoritma *stemming* memiliki kelebihan dan kekurangannya masing-masing yang dikarenakan struktur morfologi dari bahasa penyusunnya.

Menurut Paice, ada dua masalah umum dalam menggunakan algoritma *stemming* sebagai standarisasi kata (Paice, 1994).

1. *Understemming*

   Kesalahan *stemming* ini terjadi dimana kata-kata yang harusnya memiliki kata dasar yang sama, pemenggalannya tidak sampai pada kata dasar yang sama.

2. *Overstemming*

   Kesalahan *stemming* ini terjadi apabila kata-kata yang dikonversi ke kata dasar , pemenggalannya melebihi dari kata dasar tersebut.

Paice mendefinisikan tiga tipe hubungan antara pasangan-pasangan kata.yaitu :

1. Tipe 0

Dua kata yang memiliki bentuk identik dan sudah tergabung, dimana kemungkinan *homograph* diabaikan.

2. Tipe 1

Kedua kata memiliki bentuk berbeda, namun memiliki semantik yang sama

3. Tipe 2

Kedua kata memiliki bentuk berbeda dengan semantik yang berbeda juga.

Dalam aturan Paice, beberapa kata dikelompokkan secara sematik. Misalnya, kata sekolah, bersekolah, disekolahkan, menyekolahkan dan persekolahan. Kata sekolah – sekolah tidak dimasukkan ke dalam kelompok kata semantik tersebut karena mengandung karakter non kata (-). *Homograph* juga dihapus untuk memenuhi prasyarat metode evaluasi ini.

Paice menghitung kesalahan*understemming* dan *overstemming* menggunakan 2 parameter yaitu*understemming index* (UI) dan*overstemming index* (OI). *Understemming Index* merupakan bagian dari pasangan tipe 1 yang tidak berhasil digabungkan oleh algoritma *stemming*. *Overstemming Index* adalah bagian dari pasangan tipe 2 yang digabungkan dengan metode *stemming.*

Untuk mendapatkan nilai *Understemming Index,* ada perhitungan terlebih dahulu pada saat sebelum *stemming* dan setelah *stemming.*

Sebelum *stemming,* didapatkan DMT / *Desired Merge Total* dan GDMT / *Global Desired Merge Total* dengan cara :

DMT / *Desired Merge Total* adalah penggabungan semua kata-kata dalam kelompok, dengan rumus

$$DMT_g = 0.5\, N_g(N_g - 1) \text{.....(2.2)}$$

N_g = Jumlah kata pada kelompok g, jika kelompok hanya terdiri dari 1 bentuk maka nilai DMT = 0

GDMT / *Global Desired Merge Total* adalah ketidakmampuan algoritma *stemmer* untuk menggabungkan semua kata-kata ke dalam kelompok tertentu pada kata dasar yang sama., dihitung dengan rumus

$$GDMT_g = \sum_{i \in n_g} DMT_g \text{ ... ...(2.3)}$$

n_g= jumlah total kelompok semantik

Setelah *stemming,* didapatkan UMT / *Unachieved Merge Total* dan GUMT / *Global Unachieved Merge Total* dengan cara :

UMT / *Unachieved Merge Total* adalah ketidakmampuan algoritma *stemming* untuk menggabungkan semua kata-kata ke dalam kelompok semantik tertentu pada kata dasar yang sama.

UMT / Unachieved Merge Total dapat didapatkan dengan cara :

$$UMT_g = 0.5 \sum_{i \in [1 \ldots f_g]} n_{gi(N_g - n_{gi})} \text{.....(2.4)}$$

f_g = jumlah kata dasar yang berbeda di kelompok semantik g

n_{gi}= jumlah kata yang sudah menjadi kata dasar i

Setelah didapatkan *Unachieved Merge Total* kita dapat GUMT / *Global Unachieved Merge Total.*

$$GUMT = \sum_{i \in n_g} UMT_{gi} \text{.....(2.5)}$$

Nilai *understemming index* didapatkan dari perhitungan GUMT / GDMT

UI = GUMT / GDMT**.....(2.6)**

Perhitungan *overstemming index* (OI) membutuhkan nilai *Wrongly Merged Total* / WMT dan *Desired Non-Merge Total* / DNT.

Wrongly-Merged Total adalah jumlah salah penggabungan kata ke dalam kelompok kata dasar tertentu yang berisi kata dasar dari kelompok semantik berbeda.

$$WMT_s = 0.5 \sum_{i \in [1 \ldots f_{s_]}} n_{si}(N_s - n_{si}) \text{.....(2.7)}$$

f_s = jumlah kata dari semantik lain

N_s = jumlah *items* dari kelompok kata dasar s.

n_{si} = jumlah kata dasar yang berasal dari kelompok semantik ke I yang asli

Global Wrongly-Merged Total

$$GWMT = \sum_{i \in n_s} WMT_{si} \text{.....(2.8)}$$

n_s = jumlah kelompok kata dasar

Desired Non-Merge Total adalah kemungkinan untuk kata dalam suatu kelompok tertentu digabungkan dengan kata-kata lain dari kelompok semantik yang berbeda.

Desired Non-Merge Total

$$DNT_g = 0.5\ N_g(W - N_g) \text{.....(2.9)}$$

W = Semua jumlah kata

Global Desired Non-Merge Total

$$GDNT = \sum_{i \in [1 \ldots n_g]} DNT_{gi} \text{.....(2.10)}$$

Overstemming Index (OI) didapatkan dari *Global Wrongly-Merge Total / GWMT* dibagi dengan *Global Desired Non-Merge Total / GDNT.*

Overstemming Index

$$OI = GWMT / GDNT \text{.....(2.11)}$$

Berikut ini adalah contoh penggunaan evaluasi Paice

Tabel 2.3 Contoh kelompok kata semantik

Group	Full Words
G1	Sekolah Bersekolah Disekolahkan Menyekolahkan Persekolahan
G2	seko

Tabel 2.4 Hasil setelah proses *stemming*, UI = 0,6

Group	FullWords	Stemmed Words
G1	Sekolah Bersekolah Disekolahkan Menyekolahkan Persekolahan	Seko Seko Sekolah Sekolah Sekolah
G2	seko	Seko

$$DMT_g = 0.5\ N_g(N_g - 1)$$

$$DMT_{g1} = 0.5\ .5(5 - 1) = 10$$

$$DMT_{g2} = 0$$

$$GDMT_g = \sum_{i \in n_g} DMT_g$$

$$GDMT_g = 10 + 0 = 10$$

$$UMT_g = 0.5 \sum_{i \in [1 \ldots f_g]} n_{gi(N_g - n_{gi})}$$

$$UMT_1 = 0.5\ \big(3(5 - 3) + 3(5 - 3)\big) = 6$$

$$UMT_2 = 0$$

$$GUMT = \sum_{i \in n_g} UMT_{gi}$$

$$GUMT = 6 + 0 = 6$$

UI = GUMT / GDMT = 6/10 = 0.6