BAB II
LANDASAN TEORI

2.1 Tinjauan Studi

Kajian hasil penelitian tentang *Information Retrieval* sampai saat ini sudah terdapat beberapa penelitian yang berkaitan tentang pencarian dokumen ataupun kata yang mengenai temu kembali informasi. Sebelum penelitian dilakukan, penulis telah melakukan *survey* terhadap beberapa penelitian sebelumnya. Seperti penelitian yang dilakukan oleh (Zulianto dan Prasetyo, 2013) Dengan judul Aplikasi Pendeteksi Kata-Kata Penting Dalam Dokumen Menggunakan Metode TF-IDF. Pada penelitian ini dapat digunakan untuk mencari kandidat kata penting untuk dapat dijadikan acuan sebagai glosarium maupun index dalam dokumen dengan nilai rata rata *precision* 13,21% *recall* 62,56% dan nilai *Accuracy* 86,26%.

Penelitian selanjutnya dilakukan oleh (Safitri, 2013), dengan judul Temu Kembali Informasi Pada Pencarian Jurnal Skripsi Menggunakan Metode *Single Pass Clustering*. Pada penelitian tersebut telah dilakukan pengujian sistem secara uji empiris dengan 13 query inputan dan threshold mulai 0-0.25. Data yang digunakan untuk uji coba sistem adalah kumpulan dokumen abstrak sebanyak 73 dokumen abstrak. Berdasarkan hasil pengujian didapat nilai threshold yang cukup ideal adalah 0.15 dengan nilai *precission* 74,4505% , nilai *recall* mencapai 68,0769%, dan *accuracy* mencapai 96,1012%. Dan dapat di simpulkan bahwa penelitian ini untuk mendapatkan tingkat relevanci dokumen ketergantungan terhadap nilai *threshold*. Demikin juga penelitian tesis oleh (Rachman, 2011), dengan judul Sistem Temu Kembali Informasi Dalam Mesin Pencarian Menggunakan Model Ruang Vektor dan *Inverted Index*. Pada penelitian tersebut telah melakukan pengujian sistem dengan tingkat *recall* 84,7% dengan penggunaan stemming.

Mengacu pada ketiga penelitian di atas, maka pada penelitian ini akan dibahas mengenai aplikasi pencarian jurnal skripsi menggunakan metode *vector space model* (model runag vektor).

2.2 Tinjauan Pustaka

2.2.1 Jurnal Skripsi

Skripsi dapat didefinisikan sebagai penulisan karya ilmiah berisi hasil penelitian menyeluruh yang disusun secara sistematis berdasarkan ketentuan metode penelitian ilmiah (Nindayati. Cs, 2010). Mahasiswa Program Strata 1 (S1) Universitas Muhammadiyah Gresik, pada akhir masa studinya diwajibkan untuk menulis karya ilmiah yang disebut dengan skripsi atau tugas akhir sebagai syarat untuk memperoleh gelar sarjana pada bidangnya.

Jurnal skripsi terdiri atas pendahuluan yang mencakup latar belakang, rumusan penelitian, tujuan penelitian, batasan masalah, metodologi penelitian dan sistematika penulisan. Sedangkan pada bagian awal jurnal skripsi terdapat abstrak. Abstrak merupakan bagian dari laporan penelitian. Penulisan abstrak diikuti minimal 4 kata kunci (*keyword*) untuk memudahkan penyusunan bibliografi atau abstract database. Abstrak terdiri dari :

- Paragraf pertama memberikan informasi secara ringkas mengenai tujuan penelitian termasuk alasan pokok dan tujuan khusus dari penelitian,
- Paragraf kedua memberikan informasi mengenai metodologi penelitian (obyek penelitian, pemilihan sample, pengumpulan, analisis data),
- Paragraf ketiga memberikan informasi tentang kesimpulan yang disusun berdasarkan hasil data.

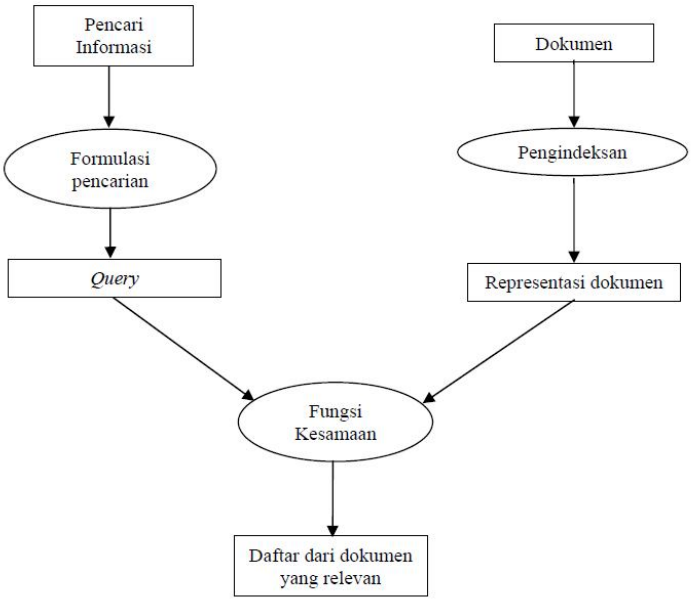
Abstrak sendiri bisa disebut juga sinopsis yang mengandung garis besar mengenai isi dari skripsi, didalam abstrak terdapat keyword-keyword yang mewakili isi dari dari pembahasan dalam jurnal skripsi. Penelitian yang akan diambil oleh peneliti pencarian jurnal skripsi berdasarkan abstraknya.

2.2.2 Sitem Temu Kembali Informasi

Menurut Gerald J.Kowalski di dalam bukunya "*Information Storage and Retrieval Sistems Theory and Implementation*", sistem temu balik informasi adalah suatu sistem yang mampu melakukan penyimpanan, pencarian, dan pemeliharaan informasi. Informasi dalam konteks ini dapat terdiri dari teks (termasuk data numerik dan tanggal), gambar, audio, video, dan objek multimedia lainnya (Kowalski and Maybury, 2002). *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. *Information Retrieval* merupakan suatu pencarian informasi (biasanya berupa dokumen) yang didasarkan pada suatu *query* (inputan *pengguna*) yang diharapkan dapat memenuhi keinginan pengguna dari kumpulan dokumen yang ada.

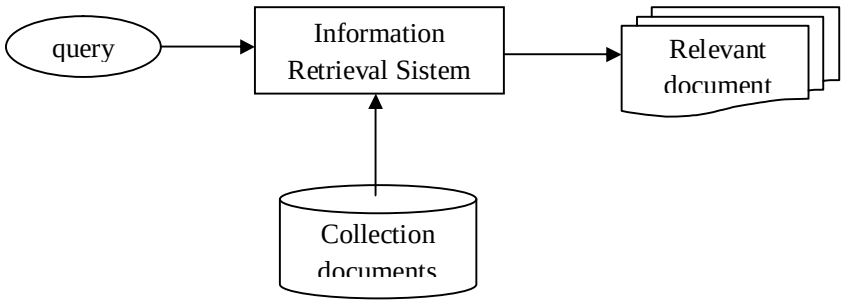
Kerangka sederhana dari sistem temu-kembali informasi terbagi menjadi dua bagian. Bagian yang pertama adalah bagian pengguna dari sistem. Pengguna dari sistem temu-kembali informasi harus menerjemahkan informasi yang dicarinya agar dapat diproses oleh sistem dengan cara memasukan *query*. *Query* tersebut nanti di proses menjadi sebuah query yang dapat dimengerti oleh komputer. Bagian yang kedua adalah bagian dari dokumen. Pada bagian ini dokumen-dokumen direpresentasikan dalam bentuk indeks. *Query* dari pengguna akan diproses melalui fungsi kesamaan untuk membandingkan *query* dengan indeks dari dokumen untuk mendapatkan dokumen yang relevan. Kerangka sistem temu kembali informasi (Ingwersen, 1992) bisa dilihat pada Gambar 2.1.



Gambar 2.1 Kerangka sistem temu kembali informasi

2.2.3 Proses Sistem Temu Kembali Informasi

Proses dalam Sistem Temu Kembali Informasi dapat digambarkan sebagai sebuah proses untuk mendapatkan dokumen yang relevan dari kumpulan dokumen yang melalui pencarian query yang diinputkan pengguna.



Gambar 2.2 Proses sistem temu kembali informasi

2.2.3.1 Indexing

Indexing adalah proses subsistem yang merepresentasikan koleksi dokumen kedalam bentuk tertentu untuk memudahkan dan mempercepat proses pencarian dan penemuan kembali dokumen yang relevan (Noor, 2011). Pembangunan index dari koleksi dokumen merupakan tugas pokok pada tahapan preprocessing di dalam proses sistem temu kembali informasi. Index dokumen adalah himpunan term yang menunjukkan isi atau topik yang dikandung oleh dokumen (Chu W, 2002).

Pembuatan *inverted index* harus melibatkan konsep *linguistic processing* yang bertujuan mengekstrak term-term penting dari dokumen (Cios, 2007) Ekstraksi term biasanya melibatkan dua operasi utama berikut (Cios, 2007) :

1. Penghapusan **stopwords**

   Stopword didefinisikan sebagai term yang tidak berhubungan (*irrelevant*) dengan subyek utama dari database meskipun kata tersebut sering kali hadir di dalam dokumen. Berikut ini adalah contoh *stopwords* bahasa Indonesia: yang, juga, dari, dia, kami, kamu, aku, saya, ini, itu, atau, dan, tersebut, pada, dengan, adalah, yaitu, ke, tak,

tidak, di, pada, jika, maka, ada, pun, lain, saja, hanya, namun, seperti, kemudian, dll.

2. Stemming

Stemming proses penghilangan awalan dan akhiran atau bisa disebut juga kata-kata yang muncul di dalam dokumen sering mempunyai banyak varian morfologik. Karena itu, setiap kata yang bukan stopwords direduksi ke bentuk stemmed word (*term*) yang cocok. Kata tersebut distem untuk mendapatkan bentuk akarnya dengan menghilangkan awalan atau akhiran.

Pembangunan inverted index menurut (Manning, 2009) ada 5 tahap yaitu:

a. Penghapusan format dan markup dari dalam dokumen

Menghapus semua tag markup dan format khusus dari dokumen, terutama pada dokumen yang mempunyai banyak tag dan format seperti dokumen (X)HTML.

b. Pemisahan rangkaian kata (*tokenization*)

Tokenization adalah tugas memisahkan deretan kata di dalam kalimat, paragraf atau halaman menjadi token atau potongan kata tunggal atau termed word. Tahapan ini juga menghilangkan karakter-karakter tertentu seperti tanda baca dan mengubah semua token ke bentuk huruf kecil (*lower case*).

c. Penyaringan (*filtration*)

Pada tahapan ini ditentukan term mana yang akan digunakan untuk merepresentasikan dokumen sehingga dapat mendeskripsikan isi dokumen dan membedakan dokumen tersebut dari dokumen lain di dalam koleksi. Term yang sering dipakai tidak dapat digunakan untuk tujuan ini, setidaknya karena dua hal. Pertama, jumlah dokumen yang relevan terhadap suatu query kemungkinan besar merupakan bagian kecil dari koleksi. Term yang efektif dalam pemisahan dokumen yang relevan dari dokumen tidak relevan kemungkinan besar adalah term yang muncul pada sedikit dokumen. Kedua, term yang muncul dalam banyak dokumen tidak mencerminkan definisi dari topik atau sub-

topik dokumen. Karena itu, term yang sering digunakan dianggap sebagai stop-word dan dihapus

d. Konversi term ke bentuk dasar (*stemming*).

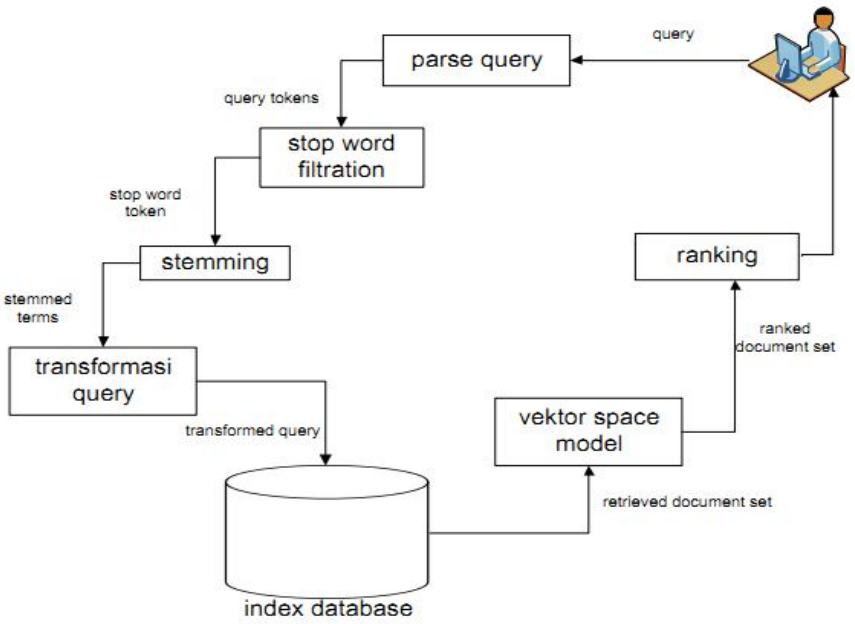
Stemming adalah proses konversi term ke bentuk umumnya. Dokumen dapat pula diekspansi dengan mencarikan sinonim bagi term-term tertentu di dalamnya. Sinonim adalah kata-kata yang mempunyai pengertian serupa tetapi berbeda dari sudut pandang morfologis.

e. Pemberian bobot terhadap term (*weighting*).

Setiap term diberikan bobot sesuai dengan skema pembobotan yang dipilih,apakah pembobotan lokal, global atau kombinasi keduanya.

2.2.3.2 Searching

Pada teknik *information retrieval* ada proses *searching* yang dimana sistem akan melakukan pencarian dokumen yang relevan terhadap query.untuk lebih jelasnya dibawah ini adalah gamabar ilustrasi proses pencarian dalam *Information Retrieval Sistem* (Noor, 2011) .



Gambar 2.3 Proses searching

Proses yang terjadi saat melakukan searching sesuai dengan ilustrasi gambar 2.3, yaitu :

1. *Parse query* yaitu memisah atau memecah query menjadi bentuk token
2. Proses *Stopword filtration* yaitu Token-token dari query yang telah dihasilkan pada proses parse query kemudian di filter melalui proses pembuangan token yang termasuk Stopword.
3. Proses Stemming yaitu hasil dari proses stopword sebelumnya kemudian di filter kembali melalui proses Stemming sehingga menghasilkan stemmed term query.
4. *Transformasi Query* yaitu apabila query yang diinputkan membutuhkan terjemahan ke dalam bentuk query bahasa lain maka sebelum mencari dokumen pada koleksi dokumen, query tersebut diterjemahkan duhulu melalui proses penerjemahan query. Sistem akan membandingkan query tersebut dengan koleksi dokumen sehingga mengembalikan dokumen-dokumen yang relevan dalam suatu bahasa yang berbeda dengan bahasa query.
5. Proses model ruang vector yaitu Tiap term atau kata yang ditemukan pada dokumen dan query diberi bobot dan disimpan sebagai salah satu elemen vektor dan dihitung nilai kemiripan antara query dan dokumen.
6. *Proses perangkingan* yaitu merangking dokumen atau konten berdasarkan nilai kemiripan antara query dan dokumen.

2.2.4 Stemming

Stemming adalah proses penghilangan atau pemotongan imbuhan yang terdapat pada sebuah kata yang mempunyai imbuhan menjadi bentuk kata dasarnya saja, untuk Bahasa Indonesia imbuhan mempunyai peran penting dalam suatu kalimat, karena suatu kata dapat mempunyai arti yang berbeda apabila diberi suatu imbuhan. Stemming juga dapat dikatakan sebagai proses membentuk suatu kata menjadi kata dasarnya. Misalnya:

Bermain > main
Memainkan > main
Permainan > main

Beberapa algoritma dalam stemming antara lain :

a. **Brute force stemming**. Algoritma ini yang paling sederhana. Bermodalkan database kata dengan kata dasarnya. Computer dengan mudah mencari kata dasar. Namun metode ini mempunyai kelemahan yaitu jumlah database kata dan kata dasarnya harus besar. Kesalahan terjadi jika kata tidak di temukan di database dan kemudian dianggap kata dasar, padahal bukan.
b. **Porter Stemmer.** Algoritma ini terkenal digunakan sebagai stemmer untuk bahasa Inggris. Porter Stemmer dalam bahasa Indonesia akan menghasilkan keambiguan karena aturan morfologi bahasa Indonesia (Tala,2003).
c. **Nazief & Adriani**. Algoritma ini paling sering di bicarakan dalam stemming bahasa Indonesia. Algoritma ini hasil penelitian internal UI (Universitas Indonesia) dan tidak di publish secara umum (Nazif, 1996). Algortitma ini merupakan gabungan antara algoritma menghilangkan imbuhan dan brute force stemming. Namun algoritma ini mempunyai dua masalah, yang pertama kemampuannya tergantung dari besarnya database kata dasar, dan yang kedua, hasil stemming tidak selalu optimal untuk aplikasi information retrieval (Tala, 2003).

Pada umumnya kata dasar pada bahasa Indonesia terdiri dari kombinasi:

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

2.2.5 Algoritma Stemming Nazief & Andriani

Adapun langkah-langkah yang digunakan oleh algoritma nazief & andriani (Agusta, 2009) yaitu sebagai berikut :

1. Kata dicari di dalam daftar kamus. Bila kata tersebut ditemukan di dalam kamus maka dapat diasumsikan kata tersebut adalah kata dasar sehingga algoritma dihentikan.

2. Bila kata di dalam langkah pertama tidak ditemukan di dalam kamus, maka diperiksa apakah sufiks tersebut yaitu sebuah partikel (“-lah” atau “-kah”). Bila ditemukan maka partikel tersebut dihilangkan.
3. Pemeriksaan dilanjutkan pada kata ganti milik (“-ku”,”-mu”,”-nya”).bila ditemukan maka kata ganti tersebut dihilangkan.
4. Memriksa akhiran (“-i”,”-an”). Bila ditemukan maka akhiran tersebut dihilangkan.

   Hingga langkah ke-4 dibutuhkan ketelitian untuk memeriksa apakah akhiran “-an” merupakan hanya bagian dari akhiran “-kan” dan memeriksa lagi apakah partikel (“-lah”,”-kah”) dan kata ganti milik (“-ku”,”-mu”,”-nya”) yang telah dihilangkan pada langkah 2 dan 3 bukan merupakan bagian dari kata dasar.
5. Memeriksa awalan (“se-“,”ke-“,”di-“,”te-“,”be-“,”pe-“,”me-“). Bila ditemukan, maka awalan tersebut dihilangkan. Pemeriksaan dilakukan dengan berulang mengingat adanya kemungkinan multi-prefik.Langkah ke-5 ini juga membutuhkan ketelitian untuk memeriksa kemungkinan peluluhan awalan, perubahan prefix yang disesuaikan dengan huruf awal kata dan aturan kombinasi prefix-sufix yang diperbolehkan.
6. Setelah menyelesaikan semua langkah diatas dengan sukses, maka algoritma akan mengembalikan kata dasar yang ditemukan.

2.2.6 Struktur Morfologi Bahasa Indonesia

Morfologi adalah bagian dari ilmu bahasa yang menyelidiki peristiwa-peristiwa umum mengenai seluk beluk kata terhadap fungsi dan arti kata. Morfologi kata bahasa Indonesia bisa berdiri dari struktur infleksional dan derivasional. Infleksional adalah struktur yang paling sederhana yang dinyatakan dengan sufiks dimana tidak mempengaruhi arti sebenarnya dari kata dasar yang dilekati (Tala,2003). Sufiks infleksional dapat dibagi menjadi 2 jenis :

1. Sufiks *lah, kah, pun, tah*. Sufiks ini sebenarnya adalah partikel yang tidak mempunyai arti. Keberadaanya pada suatu kata adalah untuk penekanan.

Contoh :

Dia	+	kah	> diakah
Duduk	+	lah	> duduklah

2. Sufiks *ku, mu, nya.* Sufiks ini berfungsi sebagai kata ganti kepunyaan.

Contoh :

Computer	+	ku	> komputerku
Buku	+	mu	> bukumu

Sufiks sufiks diatasa dapat melekat pada kata dasar secara bersama sama. Adapun aturan urutannya adalah sufiks pada jenis kedua selalu diletakkan sebelum sufiks jenis pertama. Penambahan sufiks infleksional tidak akan merubah bentuk dasar dari kata berimbuhan (Tala, 2003). Dengan kata lain, tidak ada penghilangan atau peleburan kata dasar pada kata berimbuhan. Kata dasar dapat ditentukan dengan mudah pada struktur infleksional. Struktur derivasional dalam bahasa Indonesia terdiri dari previks, sufiks, dan kombinasi dari keduanya. Previks yang sering dipakai adalah *ber, di, me, ke, meng, peng, per, ter.*

Contoh penggunaan prefiks adalah :

Ber	+	lari	> berlari
Di	+	ketik	> diketik
Ke	+	kasih	> kekasih
Meng	+	antar	> mengantar
Peng	+	atur	> pengatur
Per	+	tebal	> pertebal
Ter	+	baca	> terbaca

Beberapa prefiks seperti ber, meng, peng, per, ter. Mungkin akan berubah menjadi bentuk yang berbeda. Berntuk dari setiap prefiks bergantung pada karakter pertama dari kata dasar yang di lekatinya. Tidak seperti struktur infleksional, pada struktur infleksional pengucapan kata mungkin berubah setelah adanya penambahan prefiks. Seperti contoh menyapu yang terdiri dari prefiks meng dan kata dasar sapu. Prefiks meng

berubah menjadi meny dan karakter pertama dari kata dasar mengalami peleburan. Sufiks derivasional adalah I, kan, an (Tala,2003).

Contoh penggunaan sufiks derivasional adalah :

Gula	+	I	> gulai
Makan	+	an	> makanan
Sampai	+	kan	> sampaikan

Berbeda dengan penggunaan prefiks, penamabahan sufiks tidak akan mengubah bentuk dasar dari suatu kata. Seperti disebutkan sebelumnya, struktur derivasional juga terdiri dari konfliks, yaitu gabungan sebelumnya, struktur juga terdiri dari konfiks, yaitu gabungan dari prefiks dan sufiks yang melekat secara bersamaan pada suatu kata.

Contoh :

Per	+	main	+	an	> permainan
Ke	+	kalah	+	an	> kekalahan
Ber	+	berjatuhan	+	an	> berjatuhan
Meng	+	ambil	+	I	> mengambili

Tabel 2.1 Daftar prefik yang meluluh

Jenis prefiks	Huruf	Hasil peluluhan
Pe-/me-	K	-ng-
Pe-/me-	P	-m-
Pe-/me-	S	-ny-
Pe-/me-	T	-n-

Berdasarkan table 2.1 contoh jenis perfik yang meluluh setelah tahap algoritma nazief & andriani.

- kata depan yang berawalan (Pe-/me) dan hasil peluluhannya (-ng-) menjadi huruf (K) contoh: Pengenalan – Kenal.
- kata depan yang berawalan (Pe-/me) dan hasil peluluhannya (-ny-) menjadi huruf (S) contoh: Penyedia – Sedia.

- kata depan yang berawalan (Pe-/me) dan hasil peluluhannya (-n-) menjadi huruf (T) contoh: Penukar – tukar.

Dan tidak semua prefiks bisa berubah.Ada beberapa prefik yang beruba . perubahan tersebut dapat dilihat di **Table 2.2**

Tabel 2.2 Daftar kemungkinan perubahan prefix

Prefiks	**Perubahan**
Se-	Tidak berubah
Ke-	Tidak berubah
di-	Tidak berubah
Be-	Ber-
Te-	Ter-
Pe-	Pe-,pen-,pem-,peng-
Me-	Men-,mem-,meng-

Tidak semua prefiks dan sufiks bisa bekerjasama membentuk konfiks. Ada beberapa kombinasi prefiks sufiks yang tidak diperbolehkan. Kombinasi tersebut dapat dilihat di **Tabel 2.3**

Tabel 2.3 Daftar kombinasi prefiks dan sufiks yang tidak diperbolehkan prefiks dan sufiks yang tidak diperbolehkan

Prefiks	**Sufiks yang tidak diperbolehkan**
Be-	-i
di-	-an
Ke-	-i,-kan
Me-	-an
Se-	-i,-kan
Te-	-an
Pe-	-kan

2.2.7 Pembobotan

2.2.7.1 *Term frequency (tf)*

Pendekatan dalam pembobotan lokal yang paling banyak diterapkan adalah *term frequency (tf),Term Frequency* adalah banyaknya kemunculan term pada dokumen. Semakin sering suatu kata muncul dalam sebuah dokumen, berarti semakin penting kata tersebut.

Ada empat cara yang bisa digunakan untuk mendapatkan nilai TF (Ramadhany, 2008; Karhendana, 2008):

1. Raw Tf

   Raw Tf ialah Nilai Tf sebuah term dihitung berdasarkan kemunculan term tersebut dalam dokumen.

2. Logarithmic Tf

   Untuk memperoleh nilai Tf, cara ini menggunakan fungsi logaritmik dalam matematika.

   $$tf = 1 + \log(tf) \qquad \text{persamaan (2.1)}$$

3. Binary TF

   Cara ini, akan menghasilkan nilai boolean berdasarkan kemunculan term pada dokumen tersebut. Akan bernilai 0 apabila term tidak ada pada sebuah dokumen, dan bernilai 1 apabila term tersebut ada dalam dokumen. Sehingga banyaknya kemunulan term pada sebuah dokumen tidak berpengaruh.

4. Augmented TF

   $$tf = 0,5 + 0,5 \times \frac{tf}{\max(tf)} \qquad \text{persamaan (2.2)}$$

   Keterangan:

   - Nilai tf adalah jumlah kemunculan term pada sebuah dokumen
   - Nilai max(tf) adalah jumlah kemunculan terbanyak term pada dokumen yang sama.

Perhitungan Tf yang akan digunakan dalam implementasi Information Retrieval Sistem ini adalah Raw Tf

2.2.7.2 *Inverse Document Frequency (idf)*

Pendekatan dalam pembobotan global digunakan untuk memberikan tekanan terhadap term yang mengakibatkan perbedaan dan berdasarkan pada penyebaran dari *term* tertentu di seluruh dokumen.

Pembobotan global digunakan untuk memberikan tekanan terhadap term yang mengakibatkan perbedaan dan berdasarkan pada penyebaran dari term tertentu di seluruh dokumen. Banyak skema didasarkan pada pertimbangan bahwa semakin jarang suatu term muncul di dalam total koleksi maka term tersebut menjadi semakin berbeda.

Pendekatan terhadap pembobotan global mencakup inverse document frequency (idf), squared idf, probabilistic idf, GF-idf, entopy. Pendekatan idf merupakan pembobotan yang paling banyak digunakan saat ini. Beberapa aplikasi tidak melibatkan bobot global, hanya memperhatikan tf, yaitu ketika tf sangat kecil atau saat diperlukan penekanan terhadap frekuensi term di dalam suatu dokumen.(Polettini, 2004).

Bobot global dari suatu term i pada pendekatan inverse document frequency (idf_i) dapat didefinisikan sebagai

$$Idfi = \frac{Log2(N)}{\text{dfi}} \qquad \text{persamaan (2.3)}$$

Normalisasi untuk nilai Idf adalah sebgai berikut:

$$Idfi = log\left(\frac{N}{dfi}\right) + 1 \qquad \text{persamaan (2.4)}$$

Keterangan:

- N adalah jumlah artikel dalam koleksi dokumen
- Dfi adalah frekuensi dokumen dari *term I* dan sama dengan jumlah dokumen yang mengandung *term i.*
- Log_2 diguakan untuk memperkecil pengaruhnya relatif terhadap tf_{ij}

Bobot dari term I di dalam Information Retrieval Sistem (w_{ij}) dihitung menggunakan ukuran tf.idf yang didefinisikan sebagai berikut :

$$W_{ij} = tf_{ij} \times idf_i \qquad \text{persamaan (2.5)}$$

Keterangan :

- i = dokumen ke-i
- j = kata ke-j dari kata kunci
- w= bobot dokumen ke-i terhadap kata ke-j

TF-IDF akan menghasilkan nilai bobot dari term term dalam dokumen yang telah di masukkan yang akan dirangking.

2.2.8 *Vektor Space Model* (Model Ruang Vektor)

2.2.8.1 Definisi

Vektor Space Model (Model ruang Vektor) adalah model sistem temu balik informasi yang mengukur kemiripan antara suatu dokumen dengan query. Pada model ini, query dan dokumen dianggap sebagai vektor-vektor pada ruang n-dimensi, dimana n adalah jumlah dari seluruh term yang ada dalam leksikon. Leksikon adalah daftar semua term yang ada dalam indeks. Pada *Vektor Space Model* (Model Ruang Vektor), setiap dokumen di dalam database dan *query* direpresentasikan oleh suatu vektor multi-dimensi(Polettini.2004 dan Cios. 2007).

2.2.8.2 *Vektor Space Model* (Model ruang Vektor) dalam Information retrieval

Pada proses Information Retrieval Sistem terdapat beberapa metode yang digunakan salah satunya adalah dengan menggunakan *Vector Space Model* (Model Ruang Vektor). 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.

Salah satu pengukuran derajat kemiripan antar dokumen dengan query ataupun antar dua dokumen yang paling populer adalah dengan cosine similarity(). Pada metode ini bobot term yang terkandung pada setiap dokumen direpresentasikan dalam sebuah vektor. Misalnya $\vec{x} = \{x_1, x_2, ..., x_t\}$ dan dokumen y direpresentasikan oleh vektor $\vec{y} = \{y_1, y_2, ...y_t\}$

Korelasi ini dapat dikuantifikasi dengan sudut cosine antar dua vektor pada persamaan (Cios, 2007):

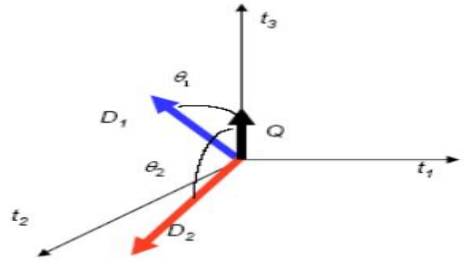
$$sim(x,y) = \frac{\vec{x} \bullet \vec{y}}{|\vec{x}| \bullet |\vec{y}|} = \frac{\sum_{i=1}^{t} W_{i,x} \bullet W_{i,y}}{\sqrt{\sum_{i=1}^{t} W^2{}_{i,x}} \bullet \sqrt{\sum_{i=1}^{t} W^2{}_{i,y}}}$$ persamaan (2.6)

Dengan $|\vec{x}|$ dan $|\vec{y}|$ merupakan norms x dan y. Nilai sim(x,y) bervariasi dari 0 sampai 1. Nilai ini menunjukkan bahwa semakin tinggi nilai sim(x,y), maka makin besar kemiripan dua vektor tersebut.

Pada Vector Space Model (Model Ruang Vektor), 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:

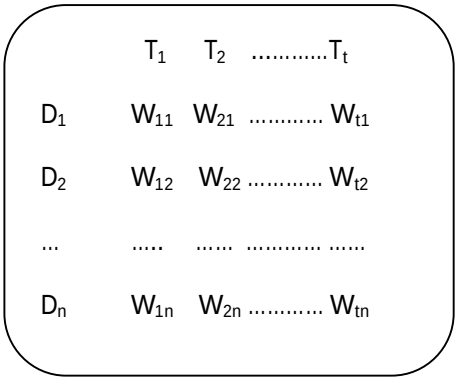
a. Vocabulary merupakan kumpulan semua term berbeda yang tersisa dari dokumen setelah preprocessing dan mengandung t term index. Term-term ini membentuk suatu ruang vector.
b. Setiap term i di dalam dokumen atau query j, diberikan suatu bobot (weight) bernilai real Wij.
c. Dokumen dan query diekspresikan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

Contoh dari model ruang untuk dua dokumen D1 dan D2, satu *query* pengguna Q1, dan tiga term T1, T2 dan T3 diperlihatkan padab gambar 2.4.



Gambar 2.4 Ukuran kesamaan antara *vector* dokumen dan *query*, Dimana: *t*= kata di database. *D*=dokumen, *Q*= kata kunci atau *query*

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.



Gambar 2.5 Contoh matriks term document untuk database dengan *n* document *t* term.