BAB III
LANDASAN TEORI

III.1 Penambangan Teks (Text Mining)

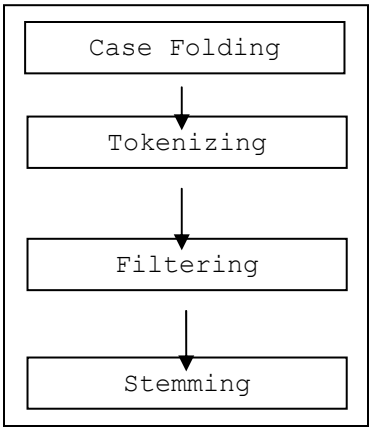
Text Mining memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antara dokumen (Ratna Maria, 2013).

Text mining bisa dianggap subjek riset yang tergolong baru. *Text mining* dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian dan menganalisan *unstructured text* dalam jumlah besar. Dalam member solusi, *text mining* mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti *data mining, information retrieval,* statistic dan matematik, *machine learning, linguistic, natural language processing,* dan *visualization.* Kegiatan riset untuk *text mining* antara lain ekstraksi dan penyimpanan teks, *preprocessing* akan konten teks, pengumpulan data *statistic* dan *indexing* dan analisa konten.

Permasalahan yang dihadapi pada *text mining* sama dengan permasalahan yang terdapat pada *data mining*, yaitu jumlah data yang besar, dimensi yang tinggi, data dan struktur yang terus berubah, dan data *noise.* Perbedaan diantara keduanya adalah pada data yang digunakan, pada *data mining*, data yang digunakan adalah *structured data,* sedangkan pada *text mining,* data yang digunakan pada

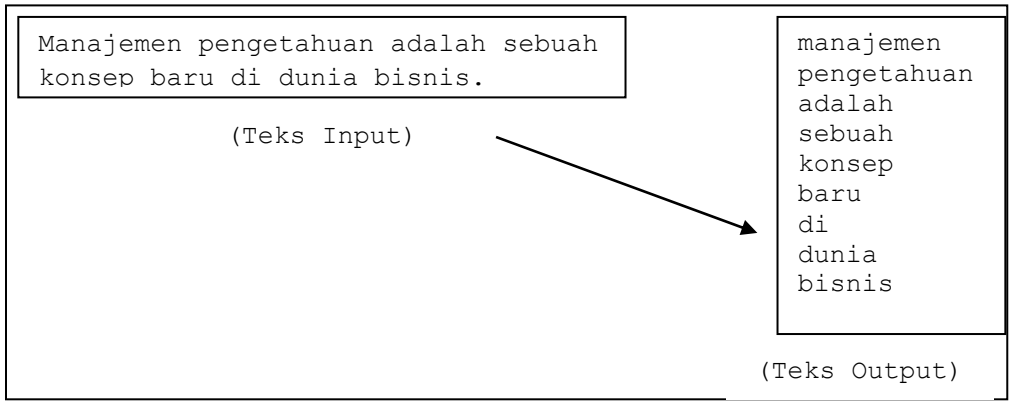
umumnya adalah ***unstructured data,*** atau minimal ***semistructured.*** Hal ini menyebabkan adanya tantangan tambahan pada ***text mining*** yaitu struktur teks yang kompleks dan tidak lengkap, arti yang tidak jelas dan tidak standard, serta bahasa yang berbeda ditambah translasi yang tidak akurat. Tahapan yang dilakukan secra umum yaitu Ekstraksi dokumen.

Teks yang dilakukan proses ***text mining,*** pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi yang tinggi, terhadap noise pada data, dan terdapat struktur teks yang tidak baik. Cara yang digunakan dalam mempelajari struktur data teks adalah dengan terlebih dahulu menentukan fitur-fitur yang mewakili setiap kata untuk setiap fitur yang ada pada dokumen, sebelum menentukan fitur-fitur yang mewakili, diperlukan tahap ***pre-processing*** yang dilakukan secara umum dalam ***text mining*** pada dokumen, yaitu ***case folding, tokenizing, filtering, dan stemming (Raymond J.Mooney, 2006)***, seperti terlihat pada Gambar 3.1.



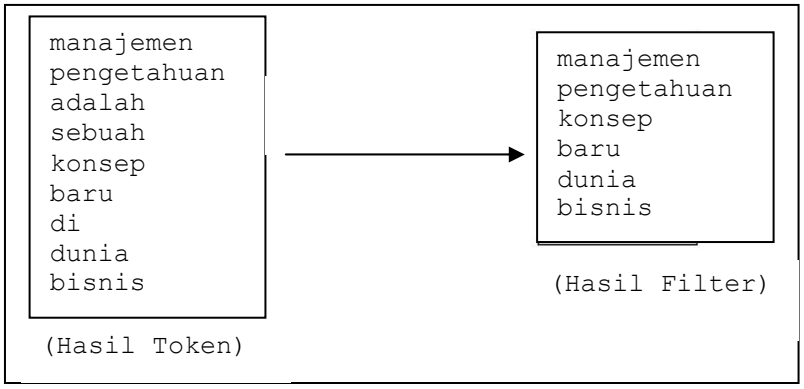
Gambar 3.1: Proses Penambangan Teks

Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil, hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap delimeter. Proses folding seperti pada Gambar 3.2.



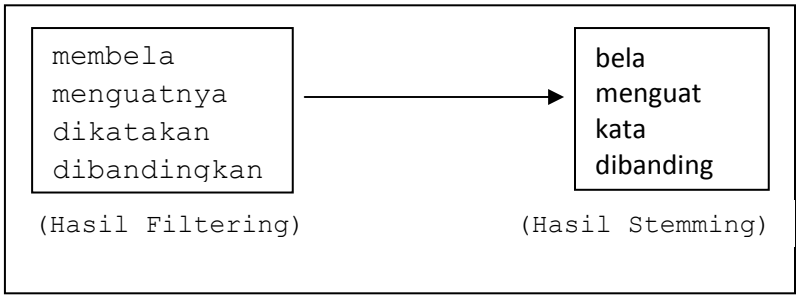
Gambar 3.2: Proses Floding

Tahap *tokenizing* atau *parsing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya, sedangkan tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil term. Bisa menggunakan algoritma *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-word*. Contoh *stopwords* adalah "yang", "dan", "di", "dari", dan seterusnya. Proses tokenizing dan filtering seperti pada Gambar 3.3.



Gambar 3.3: Proses Tokenizing dan Filter

Tahap ***stemming*** adalah tahap mencari root kata dari tiap kata hasil ***filtering***. Pada tahp ini dilakukan proses pengambilan berbagai bentukan kata kedalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen. Proses tahapan stemming pada teks berbahasa Indonesia seperti pada Gambar 3.4.

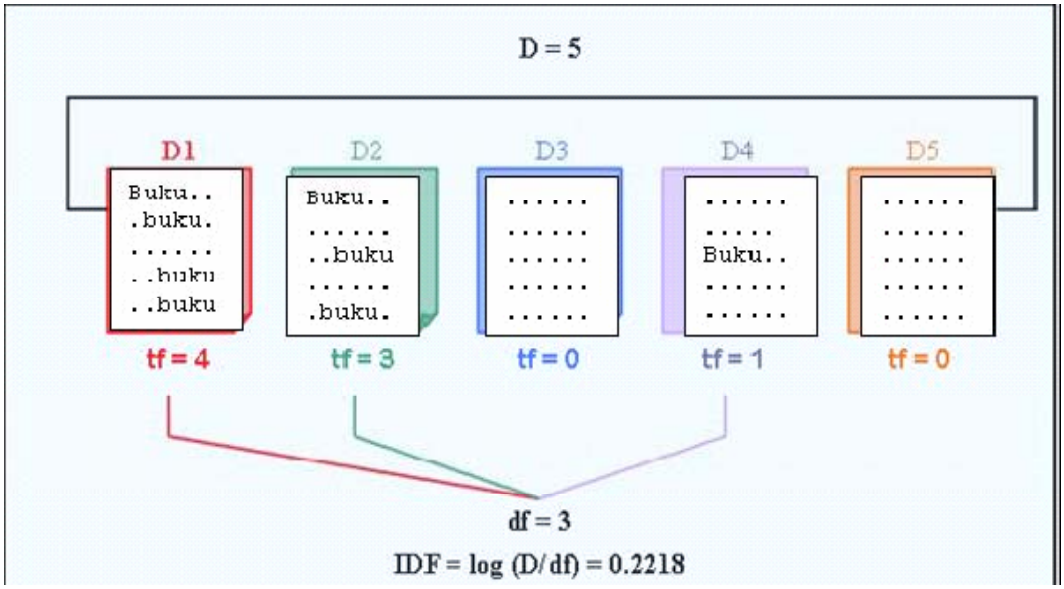


Gambar 3.4: Proses Stemming

III.2 Algoritma TF-IDF

Algoritma ini adalah salah satu jenis pengklasifikasian yang berdasarkan relevansi umpan balik algoritma yang diusulkan oleh Rocchio seperti pada Gambar 3.5. Tiga desain utama dari metode ini adalah:

1. Metode pembobotan kata.
2. Normalisasi panjang dokumen.
3. Ukuran kesamaan.



Gambar 3.5: Ilustrasi Algoritma Penamban

D1, D2, D3, D4, D5= dokumen.

Tf= banyaknya kata yang dicari pada sebuah dokumen.

D= total dokumen.

Df= banyak dokumen yang mengandung kata yang dicari.

Formula yang digunakan untuk menghitung bobot (w) masing-masing dokumen terhadap kata kunci adalah:

Rumus:

$$W_{d,t} = tf_{d,t} * IDF$$

Keterangan :

d= dokumen ke-d

t=kata ke-t dari kata kunci

W= bobot dokumen ke-d terhadap kata ke-t

Rumus mencari nilai IDF :

$$IDF = \log(d/df)$$

setelah bobot (w) masing-masing dokumen diketahui, maka dilakukan proses *sorting*/pengurutan dimana semakin besar nilai W, semakin besar tingkat similaritas dokumen tersebut terhadap kata yang dicari, demikian sebaliknya.

III.3 Cosine Similarity

Cosine similarity adalah metode similaritas yang paling banyak digunakan untuk menghitung similartias dua buah dokumen. Dengan rumus:

$$similarity = \cos \text{è} = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\, x\, \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

Keterangan:

A= bobot TF-IDF dari kata kunci

B= bobot TF-IDF dari dokumen

∑A= penjumlajhan TF-IDF dari kata kunci

∑B= penjumlajhan TF-IDF dari dokumen

III.3.1 Ilustrasi TF/IDF dan *Cosine Similarity*

Dokumen 1 (D1) : manajemen transaksi logistik

Dokumen 2 (D2) : pengetahuan antar individu

Dokumen 3 (D3) : dalam manajemen pengetahuan terdapat transfer pengetahuan logistik

Tabel 3.1 Perhitungan TF/IDF

	Frekuensi			Normal Freq			Df	D/Df	log(D/Df)	TF-IDF		
Terms	D1	D2	D3	D1	D2	D3				D1	D2	D3
Manajemen	1	0	1	0,33	-	0,20	2	1,50	0,176	0,06	-	0,04
Transaksi	1	0	0	0,33	-	-	1	3,00	0,477	0,16	-	-
Logistik	1	0	1	0,33	-	0,20	2	1,50	0,176	0,06	-	0,04
Pengetahuan	0	1	2	-	0,50	0,40	2	1,50	0,176	-	0,09	0,07
Individu	0	1	0	-	0,50	-	1	3,00	0,477	-	0,24	-
Transfer	0	0	1	-	-	0,20	1	3,00	0,477	-	-	0,10
	3	2	5									

Kata kunci (Q) : pengetahuan logistik

Terms	**Q**	TF-IDF			Sim(Q,Di)						
		D1	D2	D3	Q,D1	Q,D2	Q,D3	Qi2	Dki2	Dki2	Dki2
Manajemen	0	0,06	-	0,04	-	-	-	0	0,003445348	0	0,00124
Transaksi	0	0,16	-	-	-	-	-	0	0,025293855	0	0
Logistik	1	0,06	-	0,04	0,06	-	0,04	1	0,003445348	0	0,00124
Pengetahuan	1	-	0,09	0,07	-	0,09	0,07	1	0	0,007752	0,004961
Individu	0	-	0,24	-	-	-	-	0	0	0,056911	0
Transfer	0	-	-	0,10	-	-	-	0	0	0	0,009106
					0,06	0,09	0,11	2,00	0,03	0,06	0,02
					Sim(Q,D1) 0,23	Sim(Q,D2) 0,24	Sim(Q,D3) 0,58				

Perhitungan:

Sqrt(Q) = Sqrt($\sum_{j=1}^{n} Qj2$)Dimana j adalah kata di basis data. Misalnya untuk Sqrt(Q) = Sqrt($\sum_{j=1}^{n} Qj2$)

Sqrt(Q) = $\sqrt{0+0+1+1+0+0} = \sqrt{2}$ = 1,41

Sqrt(Di) = Sqrt($\sum_{j=1}^{n} D^2{}_{i.j}$)Dimana j adalah kata di basis data. Misalnya untuk Sqrt(Di) = Sqrt($\sum_{j=1}^{n} D^2{}_{2.j}$)

Sqrt(D1) = $\sqrt{0{,}003445 + 0{,}025294 + 0{,}003445 + 0 + 0 + 0} = \sqrt{0\,,0322}$ = 0,1794

Sqrt(D2) = $\sqrt{0+0+0+0{,}007752+0{,}056911+0} = \sqrt{0{,}06}$ = 0,25

Sqrt(D3) = $\sqrt{0{,}00124+0+0{,}00124+0{,}004961+0+0{,}009106} = \sqrt{0{,}02}$ = 0,1286

Sum(Q.Di) = $\sum_{j=1}^{n} Q_j D_{i.j}$ Dimana j adalah kata di basis data. Misalnya untuk Sum(Q.Di) = $\sum_{j=1}^{n} Q_j D_{3.j}$

Sum(Q.D1)=0+0+0,06+0+0+0 = 0,06

Sum(Q.D2)=0+0+0+0,09+0+0 = 0,09

Sum(Q.D1)=0+0+0,04+0,07+0+0 = 0,11

Selanjutnya menghitung nilai cosinus sudut antara vektor kata kunci dengan tiap dokumen dengan rumus : Cosine(Di) = sum(Q.Di)/[sqrt(Q)*sqrt(Di)]

Misalnya untuk Di maka :

Cosine(D1) = sum(Q.D1) / [sqrt(Q)*sqrt(D1)]
= 0,06/[0,141*0,1794]
= 0,23

Cosine(D2) = sum(Q.D2) / [sqrt(Q)*sqrt(D2)]
= 0,09/[0,141*0,25]
= 0,24

Cosine(D3) = sum(Q.D3) / [sqrt(Q)*sqrt(D3)]
= 0,11/[0,141*0,1286]
= 0,58

sehingga hasil yang diperoleh untuk ketiga dokumen diatas adalah seperti berikut ini.

Tabel 3.3 Hasil *Vector Space Model*

	D1	**D2**	**D3**
Cosine	0,23	0,24	0,58
	Rank 3	**Rank 2**	**Rank 1**

Dari hasil akhir maka dapat diketahui bahwa dokumen ke-3 (D3) memiliki tingkat kesamaan tertinggi kemudian diikuti dengan D2 lalu D1.