BAB 2

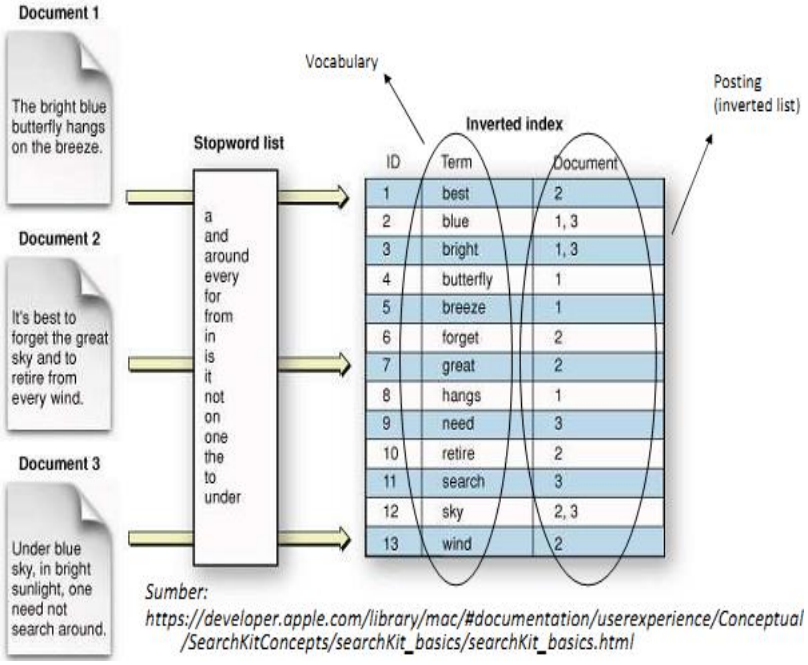
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tujuan dari sistem peringkas teks otomatis adalah untuk memilih informasi paling penting dari meratanya sumber-sumber teks. Sebuah rutinitas pertumbuhan cepat dari data di internet membuat acara pencapaian dari tujuan seperti itu menjadi tantangan besar (Binwahlan, M. S., Naomie, S., dan Ladda, S., 2011).

Secara umum terdapat dua tipe peringkasan yaitu ekstraktif dan abstraktif. Ekstraktif meringkas suatu dokumen dengan memilih sebagian dari kalimat yang ada dalam dokumen asli. Sedangkan metode abstraktif melakukan peringkasan dengan cara menginterpretasi teks asal melalui proses transformasi suatu kalimat asli (G. Erkan, dan Dragomir R. R., 2004).

Dalam penelitian ini menggunakan tipe peringkasan ekstraktif dengan menggunakan metode TF-IDF untuk memberikan bobot pada setiap dokumen serta memisahkan kata yang berbeda dengan menggunakan *inverted index*. *Inverted index* adalah sebuah struktur data *index* yang dibangun untuk memudahkan *query* pencarian. Pada dasarnya, *inverted index* adalah struktur data yang memotong tiap kata (*term*) yang berbeda dari suatu daftar *term* dokumen. Berikut pada gambar 2.1 merupakan contoh proses *inverted index*. Sedangkan penulisan sejenis sebelumnya telah dirangkumkan seperti terlihat pada tabel 2.1.



Gambar 2. 1 contoh *inverted index*

Dalam gambar 2.1, ditampilkan sebuah contoh dari perigkasan teks dimana ada tiga dokumen yang akan diringkas. Tiga dokumen tersebut kemudian dimasukan kedalam proses *filtering* dimana semua kata sambung yang terdapat dalam ketiga dokumen tersebut dihilangkan. Setelah proses penghilangan kata sambung, kemudian semua yang tersisa dihitung berapa jumlah kata yang sama dan hasil dari proses tersebut adalah sebuah *inverted index* yang berasal dari ketiga dokumen tersebut.

Tabel 2. 1 daftar tinjauan pustaka

Penulis / tahun	Judul / topik	Metode	Hasil
Binwahlan, M. S., Naomie, S., dan Ladda, S. (2011).	*Fuzzy Swarm Diversity Based Text Summarization*.	- *Fuzzy Swarm,* - *Particle Swarm Optimization (PSO),* - *benchmark*	- Metode *Particle Swarm Optimization* mempunyai kinerja yang lebih baik - mengungguli model *Swarm* dan metode *benchmark*
Priyo Adyaksa R. (2010).	*Aplikasi Peringkasan Teks Berbasiskan Web Menggunakan Metode Ekstrasi dan Skema Pembobotan LOG TF-IDF*,	- Ekstrasi - Skema pembobotan LOG TF-IDF - bahasa pemrogram-an php - algoritma peringkasan teks Porter Steaming	- Web dapat meringkas teks tunggal Bahasa Indonesia. - Aplikasi hanya dapat mengolah data teks pada textarea, file(txt, html/htm), dan URL. - algoritma porter stemmer menghasilkan hasil ringkasan yang lebih baik dengan menggunakan *stem*.

Penulis / tahun	Judul / topik	Metode	Hasil
			- Skema pembobotan LOG TF-IDF dan algoritma porter stemmer untuk Bahasa Indonesia meningkatkan ketepatan hasil ringkasan sekitar 6,2%.
Dragomir, R. R., dan G. Erkan, (2011).	*LexRank: Graph-based Lexical Centrality as Salience in Text Summarization*.	- TF-IDF - bahasa pemrogram-an Markov - algoritma peringkasan teks MEAD[4] Summarizati on system	- eigenvector dapat meningkatkan probabilitas dengan masing-masing objek - Probabilitas dapat dijadikan dasar pada klasifikasi objek - Seseorang dapat memilih label apa yang akan dipesan, - sentralitas eigenvector menilai semua objek.

Penulis / tahun	Judul / topik	Metode	Hasil
David Kristiawan (usulan)	*Implementasi Metode TF-IDF Untuk Aplikasi Peringkas Document Base*	- TF-IDF - Bahasa pemrogram-an python - algoritma peringkasan teks Porter Steaming	menghasilkan aplikasi peringkasan teks Bahasa Indonesia berbasis dokumen

2.2 Landasan Teori

2.2.1 Metode TF-IDF

Metode TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah statistic numerik yang digunakan untuk memperlihatkan betapa pentingnya sebuah kata dalam dokumen diantara koleksi atau *corpus* (Onno W. Purbo, 2019).

Berikut ini adalah ilustrasi dari penghitungan vektorisasi TF-IDF:

$$TF(d,t)=\begin{cases} 0 \text{ jika } freq(d,t) = 0, \\ 1+\log(1+\log(freq(d,t))), \end{cases}$$

$$IDF(t)= \log \frac{1+|d|}{|d_t|},$$

TF-IDF(d,t)= TF(d,t) x IDF(t).

Keterangan:

- *freq*(d,t) adalah angka dari kejadian dari kata t dalam dokumen d
- t adalah kata(*term*)

- d adalah dokumen
- d_t adalah jumlah dokumen yang berisi kata t.

Contoh:

Terdapat 3 kalimat:

Kal1 = [“saya suka makan sate”]

Kal2 = [“terutama sate daging kambing”]

Kal3 = [“saya suka membeli sate kambing di warung makan ’sate kambing samirono’”]

Kemudian tiga kalimat tersebut dibentuk kedalam tabel seperti yang diperlihatkan pada tabel 2.2:

Tabel 2. 2 tabel frekuensi kata(*term*) dalam dokumen/kalimat(*document*)

Kata(t)\kalimat(d)	Kal1	Kal2	Kal3
saya	1	0	1
suka	1	0	1
Makan	1	0	1
Sate	1	1	2
Terutama	0	1	0
Daging	0	1	0
kambing	0	1	2
Membeli	0	0	1
Warung	0	0	1
Samirono	0	0	1

Proses penghitungan nilai TF:

TF(Kal3,sate) = $1+\log(1+\log(freq(d,t)))$ = 1 + log(1 + log 2) = 1 + log(1 + 0,30102) = 1 + log(1,30102) = 1 + 0,11428 = 1,11428

TF(Kal3,sate) = 1,11429

Proses penghitungan nilai IDF:

IDF(sate) = log ((1 + |d|) / $|d_t|$) = log ((1 + 3) / 3) = log(4/3)

IDF(sate) = log(1,33333) = 0,12493

IDF(sate) = 0,12494

TF-IDF(Kal3,Sate) = TF(Kal3,sate) * IDF(sate) = 1,11429 * 0,12494 = 0,13921

TF-IDF(Kal3,Sate) = 0,13921

2.2.2 Porter Stemming

Algoritma Porter Stemming (atau ' Porter Stemmer ') adalah sebuah proses untuk menghapus morfologi umum dan akhiran inflexional dari kata dalam bahasa Inggris. Penggunaan utamanya adalah sebagai bagian dari proses normalisasi kata yang biasanya dilakukan saat membuat sistem pencarian informasi.(https://tartarus.org)

Stemming adalah proses pemetaan dan penguraian berbagai bentuk (variants) dari suatu kata menjadi bentuk kata dasarnya. Proses ini juga disebut sebagai conflation. Proses stemming secara luas sudah digunakan di dalam kegiatan Information retrieval (pencarian informasi) untuk meningkatkan kualitas informasi yang didapatkan. Cara kerja stemming dapat dilakukan dengan menggunakan kamus kata dasar maupun menggunakan aturan-aturan imbuhan. Porter stemmer untuk Bahasa Indonesia atau yang biasa disebut dengan stemmer Tala menggunakan rule base analisis untuk mencari root sebuah kata. Stemmer Tala tidak menggunakan kamus dalam proses, melainkan menggunakan algoritma berbasis aturan.(https://ojs.uajy.ac.id)