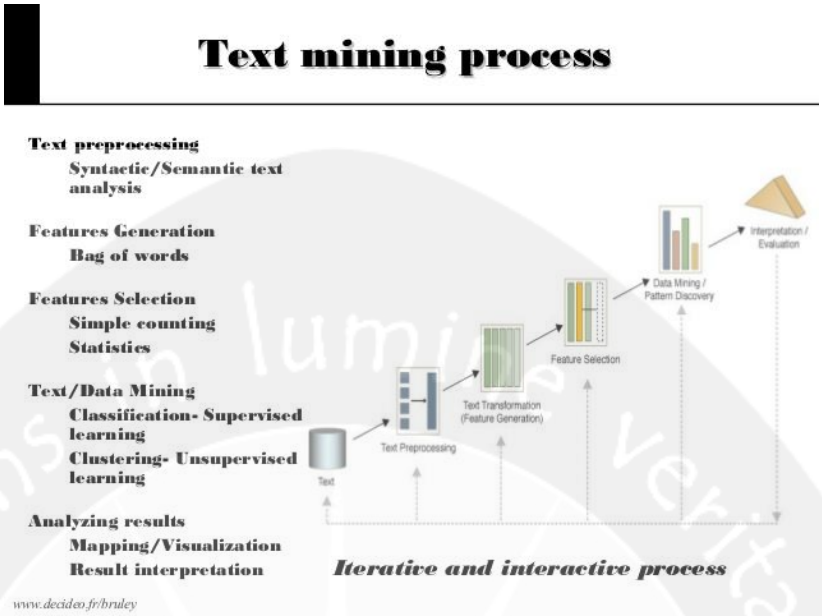
# BAB 3

# LANDASAN TEORI

## 3.1 *Text Mining*

*Text mining* merupakan suatu teknologi untuk menemukan suatu pengetahuan yang berguna dalam suatu koleksi dokumen teks sehingga diperoleh tren, pola, atau kemiripan teks bahasa alamiah. *Text mining* adalah proses penggalian, valid, dan dapat ditindaklanjuti pengetahuan yang tersebar di seluruh dokumen dan memanfaatkan pengetahuan ini untuk lebih mengorganisir informasi untuk referensi di masa mendatang. Penambangan berasal dari penggalian barang berharga dari biji dari batuan yang tidak bernilai, ini merupakan emas yang tersembunyi di pegunungan data *text*.

*Text mining*, biasa dikenal dengan *Text Data Mining*(TDM), adalah penemuan oleh komputer era baru, informasi yang sebelumnya tidak diketahui, secara otomatis dengan mengekstrasi informasi dari sumber daya yang datanya tidak terstruktur (Ojo & Adeyemo, 2013).




**Gambar 3.1 Text Mining Process**

(*Sumber: www.decideo.fribruley*)

**Gambar *3.1*** merupakan gambaran umum proses dari *text mining*. Secara garis besar *text mining* dimulai dari *text preprocessing* yaitu melakukan analisis *text*, setelah itu *features generation* adalah mengumpulkan kata-kata yang sudah dibersihkan saat *text preprocessing*. Setelah itu ada *feature selection* yaitu melakukan penghitungan sesuai yang dibutuhkan, setelah itu dilakukan *text/data mining* yang disini melakukan *clustering*, dan yang terakhir menganalisa hasil.

## 3.2 *Automatic Text Summarization*

*Automatic Text Summarization* dapat dikatakan sebagai pemecahan untuk merubah 1 atau lebih dokumen ke versi yang lebih sederhana tetapi tetap menjaga content yang ada di dalamya. Metode ini dibagi menjadi dua bagian utama yaitu ekstrasi dan abstraksi. Ringkasan dengan metode ekstrasi

terdiri dari kumpulan kalimat yang diambil dari dokumen-dokumen dengan menggunakan statistical atau heuristic berdasar dari informasi yang paling sering muncul. Ringkasan abstraktif mengandung analisis *semantic* untuk menginterpretasi sumber informasi dan menemukan konsep yang baru untuk mengubah teks yang akan menjadi ringkasan (Motta & Tourigny, 2011).

## 3.3 *K-Means*

K-means merupakan salah satu algoritma klaster yang paling terkenal dan sering digunakan untuk menyelesaikan permasalahan *clustering* yaitu dengan mengelompokan sejumalah k *cluster* (dimana jumlah k telah di definisikan sebelumnya).

Langkah-langkah algoritma K-Means adalah sebagai berikut:

1. Tentukan nilai k sebagai jumlah klaster yang ingin dibentuk.
2. Bangkitkan k centroid (titik pusat klaster) awal secara random.
3. Hitung jarak setiap data ke masing-masing *centroid* menggunakan rumus korelasi antar dua objek yaitu Euclidean Distance dan kesamaan Cosine.
4. Kelompokan setiap data berdasarkan jarak terdekat antara data dengan centroidnya.
5. Tentukan posisi centroid baru (k C) dengan cara menghitung nilai ratarata dari data-data yang ada pada centroid yang sama.

$$C_k = \left(\frac{1}{n_k}\right)\sum d_i \qquad (1)$$

Dimana k n adalah jumlah dokumen dalam cluster k dan i d adalah dokumen dalam cluster

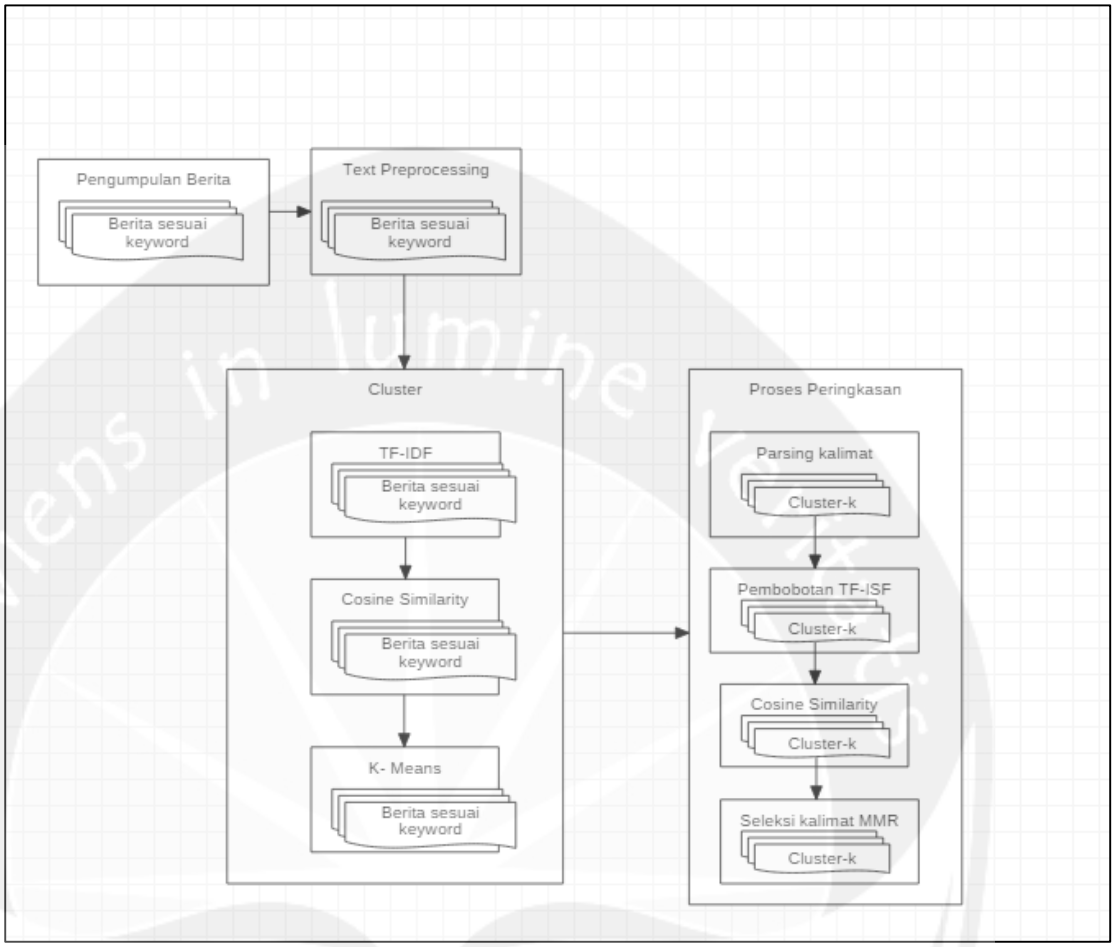
6. Kembali ke langkah 3 jika posisi centroid baru dengan centroid lama tidak sama (Luthfiarta, et al., 2014).

$$Sim(d_x d_y) = \frac{\sum_{k=1}^{n} x_k \times y_k}{\sqrt{\sum_{k=1}^{n} x_k{}^2} \times \sqrt{\sum_{k=1}^{n} y_k{}^2}} \quad (2)$$

Ambang batas *cosine similarity* adalah 0.7 didapat dari penelitian pergerakan data dalam jumlah besar, apabila di atas 0.7 keterkaitan dokumen menurun drastis (Jatowt, et al., 2013).

### 3.4 *Metode Maximum Marginal Relevance (MMR)*

Maximum Marginal Relevance (MMR) merupakan salah satu metode peringkasan dokumen yang menggunakan teknik ekstraksi. Metode ini mengkombinasikan cosine similarity antara kalimat dengan *query(query-relevance)*dan kalimat dengan kalimat lain yang telah terpilih sebagai ringkasan dengan tujuan memaksimalkan kesamaan kalimat dengan *query* dan meminimalkan redundansi kalimat atau dengan kata lain meminimalkan adanya kalimat yang mempunyai kesamaan makna pada hasil ringkasan.




**Gambar 3.2 Skema Peringkasan**

**Gambar** ***3.2*** merupakan gambar urutan peringkasan dengan menggunakan *MMR*.

1. *Text preprocessing*

   *Text preprocessing* merupakan tahapan awal yang dilakukan sebelum input dokumen menjadi kelompok-kelompok kalimat. Dalam *text preprocessing* juga terdapat tahap-tahap tertentu:

   a. Tokenization/Segmentation

      Tahap ini merupakan tahapan pemotongan string input berdasarkan tiap kata yang menyusunnya. Contoh:

Text input: "Riyo bernyanyi riang."

Melalui tokenization:

Riyo

Bernyanyi

riang

b. *Stopword Removal*

*Stopword removal* merupakan metode untuk menghilangkan kata-kata yang tidak relevan dalam dokumen. Misal kata: dari, ke, merupakan.

c. *Stemming*

*Stemming* dilakukan untuk mencari akar dari suatu kata. Dalam pembangunan aplikasi ini digunakan algoritma Porter untuk melakukan *stemming*. Contoh:belajar->ajar,menulis->tulis.

2. TF-IDF

Salah satu tahapan MMR adalah dengan menggunakan metode TFIDF. TF untuk perhitungan frekuensi suatu kata dalam dokumen, sedangkan IDF merupakan nilai dari masing-masing kata.

a. Kalimat-kalimat yang ada dalam dokumen kemudian dipecah menjadi kata-kata. Hitung nilai TF dari kata tersebut menggunakan rumus TF dengan $f_{(t,d)}$ merupakan frekuensi sebuah kata $(t)$ muncul dalam dokumen $d$, sedangkan $\sum t,d$ merupakan total keseluruhan kata dalam dokumen $d$.

$$TF = \frac{f_{(t,d)}}{\sum t,d} \quad (3)$$

b. Kata-kata yang sudah dihitung nilai TF nya maka dicari nilai IDF dari masing-masing

kata. ***D*** merupakan jumlah dokumen, sedangkan ***DF*** merupakan jumlah dokumen dimana f muncul dalam dokumen ***D***.

$$IDF = log \frac{D}{DF} \quad (4)$$

c. Masing-masing kata dihitung menggunakan rumus berikut. Untuk dilihat seberapa sering kata tersebut muncul dalam suatu dokumen.

$$TF - IDF_{(t)} = TF_{(t)} * IDF_{(t)} \quad (5)$$

3. Cosine Similarity

Cosine similarity adalah ukuran kesamaan yang lebih umum digunakan dalam information retrieval dan merupakan ukuran sudut antara vektor dokumen (titik (*ax*,*bx*)) dan (titik (*ay*,*by*)) (Imbar, et al., 2014).

$$sim(s_1, s_2) = \frac{\vec{s_1} \cdot \vec{s_2}}{|\vec{s_1}||\vec{s_2}|} = \frac{\sum_i w_{1,i} \cdot w_{2,i}}{\sqrt{\sum_i w_{1,i}^2}\sqrt{\sum_i w_{2,i}^2}} \quad (6)$$

4. Rumus Maximum Marginal Relevance

Maximum Marginal Relevance (MMR) adalah salah satu dari sekian metode ekstraksi teks yang dapat diterapkan untuk meringkas dokumen tunggal maupun multidokumen dengan cara melakukan rangking ulang dan membandingkan similarity antar dokumen. Jika kesamaan (similarity) antara satu kalimat dengan kalimat yang lain tinggi, maka kemungkinan terjadi redundansi. Rumus untuk menghitung nilai MMR yang dapat mengurangi redundansi adalah :

$$MMR(S_i) = \lambda . Sim1\,(Si, Q) - (1 - \lambda)\,. max\; Sim2(Si, Sj) \quad (7)$$

Keterangan :

$\lambda$ = parameter bobot untuk mengatur tingkat relevansi

$Si$ = vektor bobot kata yang menjadi kandidat

$Sj$ = vektor bobot kata selain yang menjadi kadidat

$Q$ = vektor bobot kata dari query user (judul berita)

$Sim1$ = nilai similarity antara query dengan tiap kalimat

$Sim2$ = nilai similarity antara kalimat (Yulita, 2015)