BAB II

LANDASAN TEORI

2.1 Berita

Kata “Berita” berasal dari kata sansekerta, vrit (ada atau terjadi) atau vritta (kejadian atau peristiwa). Berita merupakan sesuatu atau seseorang yang dipandang oleh media merupakan subjek yang layak untuk diberitakan. Biasanya subjek pemberitaan merupakan sesuatu atau seseorang yang memang sedang di sorot atau diperhatikan oleh masyarakat umum (Kusumaningrat, 2005).

Menurut Brooks dkk (1980), Kriteria umum berita dalam buku “*New Reporting and Editting*” menunjukan 11 kriteria umum nilai berita yang harus diperhatikan dengan seksama oleh para reporter dan editor media massa. yaitu : Keluar biasaan (*unsualness*), Kebaruan (*newsness*), Akibat (*impact*), Aktual (*timeliness*), Kedekatan (*proximity*), Informasi (*information*), Konflik (*conflict*), Orang penting (*prominence*), Ketertarikan manusiawi (*human interest*), Kejutan (*suprising*), Seks (*sex*).

Adapun jenis–jenis berita yang dikenal dalam dunia jurnalistik antara lain adalah sebagai berikut. (Romli, 2014)

a. *Straight News*: berita langsung, apa adanya, ditulis secara singkat dan lugas. Sebagian besar halaman depan surat kabar atau yang menjadi berita utama (headline) merupakan berita jenis ini.
b. *Depth News*: berita mendalam, dikembangkan dengan pendalaman hal-hal yang ada di bawah suatu permukaan.

c. *Investigation News*: berita yang dikembangkan berdasarkan penelitian atau penyelidikan dari berbagai sumber.

d. *Interpretative News*: berita yang dikembangkan dengan pendapat atau penilaian wartawan berdasarkan fakta yang ditemukan.

e. *Opinion News*: berita mengenai pendapat seseorang, biasanya pendapat para cendekiawan, sarjana, ahli, atau pejabat mengenai suatu hal, peristiwa, kondisi poleksosbudhankam, dan sebagainya.

2.2 Media Massa

Media Massa merupakan alat yang digunakan untuk menyampaikan pesan dari sumber kepada khalayak (penerima) dengan menggunakan alat-alat komunikasi mekanis seperti surat kabar, film, radio, serta televisi. Karakteristik media massa adalah sebagai berikut: (Cangara, 2014)

a. Bersifat melembaga, artinya "pihak yang mengelola media terdiri dari banyak orang, yakni mulai dari pengumpulan pengelolaan sampai pada penyajian informasi".

b. Bersifat satu arah, artinya "komunikasi yang dilakukan kurang memungkinkan terjadinya dialog antara pengirim dan penerima. Kalau toh terjadi reaksi atau umpan balik, biasanya memerlukan waktu dan tertunda".

c. Meluas dan serempak, artinya "dapat mengatasi rintangan waktu dan jarak, karena ia memiliki kecepatan. Bergerak secara luas dan simultan, dimana informasi yang disampaikan diterima oleh banyak orang pada saat yang sama".

d. Memakai peralatan teknis atau mekanis, seperti radio, televisi, surat kabar, dan semacamnya.

e. Bersifat terbuka, artinya “pesannya dapat diterima oleh siapa saja dan dimana saja tanpa mengenal usia, jenis kelamin, dan suku bangsa”.

2.3 Information Retrieval

Information Retrieval (IR) atau temu-kembali informasi adalah ilmu yang mempelajari tentang cara mendapatkan kembali informasi yang pernah tersedia. IR berisi tindakan, metode dan prosedur untuk menemukan kembali data yang tersimpan untuk menyediakan informasi mengenai subyek yang dibutuhkan (Pedrycz dkk).

Menurut husni dkk (2015), terdapat 5 langkah pembangunan *inverted index* dalam *Information Retrieval*, yaitu:

1. Penghapusan format dan markup dari dalam dokumen (halaman web).
2. Pemisahan rangkaian *term* (*tokenization*). *Term* biasanya berupa kata atau frasa di di dalam dokumen. Namun, kata-kata yang tidak memberikan perbedaan seperti ini, itu, saya, kamu, serta tandatanda baca dihilangkan (tidak dianggap sebagai term).
3. Pengembalian term ke bentuk akar kata (*stemming*) atau bentuk umum yang disepakati.
4. Pemberian bobot terhadap *term* (*weighting*) dengan memberlakukan kombinasi perkalian bobot lokal *term frequency* dan bobot *global inverse document frequency*, ditulis TF - IDF.

5. Menyimpan term yang diperoleh disertai oleh nomor dokumen dimana *term* tersebut muncul dan jumlah kemunculannya. Daftar *term* ini dinamakan *index* atau *inverted index*.

Index tersebut selanjutnya digunakan oleh berbagai metode *information retrieval*, seperti asosiasi, klasifikasi dan *clustering* untuk menemukan suatu kesimpulan mengenai suatu himpunan dokumen.

2.4 Text Preprocessing

Text Preprocessing adalah proses memperkenalkan dokumen baru ke sistem pengambilan informasi di mana setiap dokumen diperkenalkan diwakili oleh seperangkat istilah indeks. Tujuan *preprocessing* dokumen adalah untuk merepresentasikan dokumen sedemikian rupa sehingga penyimpanannya dalam sistem dan pengambilan dari sistem menjadi sangat efisien (Susanto & Shidik, 2017).

Data yang diterima dari *crawling* adalah data mentah memiliki *noise* atau kata-kata yang tidak dibutuhkan, Untuk itu diperlukan proses normalisasi text. Normalisasi teks meliputi *Lower Case*, *Remove Number*, *Remove Punctuation*, *Tokenization*, *Stopword Removal*, *Stemming* dan *Term Weighting*. Berikut tahap-tahap yang akan dilakukan:

2.4.1 Lower Case, Remove Number, and Remove Punctuation (Case Folding)

Berikut adalah langkah-langkah yang akan diproses:

1. *Lower case*, Data akan di konversikan menjadi huruf kecil semua.
2. *Remove Number*, Mengeliminasi nilai angka pada data.
3. *Remove Punctuation*, Mengeliminasi rangkaian symbol dan tanda baca sebagai berikut : [! ”# $% & '() * +, -. / :; <=>? @ [\] ^ _` {|} ~].

2.4.2 Tokenization and Stopword Removal (Filtering)

Berikut adalah langkah-langkah yang akan diproses:

1. *Tokenization*, proses pemisahan teks yang diberikan menjadi potongan-potongan kecil yang disebut token, yaitu kata, angka, tanda baca, dan lainnya.
2. *Stopword Removal*, Mengeliminasi kata-kata yang tidak memiliki arti penting dan umum, seperti : adalah, akan, agak, dll (Lama, 2013).

2.4.3 Stemming

Berikut adalah langkah-langkah yang dilakukan pada tahapan *Stemming*: (Susanto & Shidik, 2017).

1. Menghapus awalan (“ter-“, “te-“, “se-“, “ke-“, “di-“), langkah ini akan terus diulangi sampai tidak ada kata yang memiliki awalan (“ter-“, “te-“, “se-“, “ke-“, “di-“).

2. Menghapus akhiran (“-nya”, “-mu”, “-ku”, “-kah”, “-lah”, "-pun", "-tah"), langkah ini akan terus diulangi sampai tidak ada kata yang memiliki akhiran (“-nya”, “-mu”, “-ku”, “-kah”, “-lah”, "-pun", "-tah").
3. Menyederhanakan kata berulang, seperti “makan-makan” menjadi “makan”
4. Menghapus kombinasi awalan dan akhiran sebagai berikut:

Tabel 2.1 List Kominasi Stemming

Awalan	akhiran
Ke -	-i, -kan
Se-	-i, -kan
Me-	-an
Di-	-an
Be-	-i

2.4.4 Term Weighting (Term Frequency – Inverse Document Frequency)

Dokumen direpresentasikan sebagai vektor. Term weighting adalah konsep penting yang menentukan keberhasilan atau kegagalan sistem klasifikasi. Karena istilah yang berbeda memiliki tingkat kepentingan yang berbeda dalam sebuah teks, bobot istilah dikaitkan dengan setiap istilah sebagai indikator penting. Frekuensi istilah setiap kata dalam dokumen (TF) adalah bobot yang tergantung pada distribusi setiap kata dalam dokumen. Itu mengungkapkan Pentingnya kata dalam dokumen. Frekuensi dokumen terbalik dari setiap kata dalam basis data dokumen (IDF) adalah bobot yang tergantung pada distribusi setiap kata dalam *database*

dokumen. Ini mengungkapkan pentingnya setiap kata dalam *database* dokumen. TF-IDF adalah teknik yang menggunakan TF dan IDF untuk menentukan berat suatu istilah. Skema TF-IDF sangat populer di bidang klasifikasi teks dan hampir semua skema pembobotan lainnya adalah varian dari skema ini (Fatima, 2017).

Pembobotan TF-IDF adalah suatu pengukuran statistik untuk mengukur seberapa penting sebuah kata dalam kumpulan dokumen. Tingkat kepentingan meningkat ketika sebuah kata muncul beberapa kali dalam sebuah dokumen tetapi diimbangi dengan frekuensi kemunculan kata tersebut dalam kumpulan dokumen. Wi adalah nilai bobot, d adalah dokumen, j adalah angka yang menunjuk kesuatu dokumen, t adalah kata, n adalah angka yang menunjuk pada frekuensi kemunculan kata. Metode TF-IDF akan menghitung nilai bobot dalam suatu dokumen ke-j melalui frekuensi kemunculan suatu kata ke-n atau istilah di setiap dokumen dan frekuensi kemunculan kata pada beberapa dokumen. Oleh karena itu, dihitung terlebih dahulu Term Frequency (TF) nya. Secara matematis, rumus TF adalah sebagai berikut.

$$\text{TF (t, d)} = \sum(t_n, d_j) \qquad \textbf{(2. 1)}$$

Tahap selanjutnya adalah menghitung nilai IDF (Inverse Document Frequency), yaitu nilai bobot suatu term dihitung dari seringnya suatu kata muncul di beberapa dokumen. Semakin sering suatu kata muncul di banyak dokumen, maka nilai IDF nya semakin kecil. |D| adalah jumlah total dokumen dan df(t) merupakan jumlah dokumen yang berisi kata t. Sehingga Inverse Document Frequency (IDF) dapat dihitung dari jumlah total dokumen dibagi dengan banyaknya dokumen yang mengandung kata t atau secara matematis, rumusnya adalah sebagai berikut.

$$\text{IDF(T)} = \text{Log}(\frac{|D|}{df(t)}) \quad \textbf{(2. 2)}$$

Dalam metode TF-IDF, bobot Wi adalah nilai dari TF(t, d) dikalikan dengan nilai dari IDF(t). Bobot suatu kemunculan kata semakin besar jika kata tersebut sering muncul dalam suatu dokumen dan semakin kecil jika kata tersebut muncul dalam banyak dokumen. Skema normalisasi pembobotan TF-IDF dapat dihitung menggunakan rumus matematis sebagai berikut: (Wesley, 2019)

$$W_i = \text{TF (t,d) x IDF (t)} \quad \textbf{(2. 3)}$$

2.5 K- Means

K-Means *Clustering* adalah suatu metode penganalisaan data atau metode *Data Mining* yang melakukan proses pemodelan tanpa supervisi (*unsupervised*) dan merupakan salah satu metode yang melakukan pengelompokan data dengan sistem partisi.

Data clustering menggunakan metode K-Means Clustering ini secara umum dilakukan dengan melakukan tahapan-tahapan sebagai berikut: (Yulian, 2018)

1. Memilih secara acak *k* (jumlah cluster) buah data sebagai pusat cluster
2. Jarak antara data dan pusat cluster dihitung menggunakan Euclidian Distance. Untuk menghitung jarak semua data ke setiap titik pusat cluster dapat menggunakan teori jarak Euclidean yang dirumuskan sebagai berikut:

$$\text{D (i, j)} = \sqrt{\left(X_{1i} - X_{1j}\right)^2} + \left(X_{2i} - X_{2j}\right)^2 + \cdots + \left(X_{ki} - X_{kj}\right)^2 \quad \textbf{(2. 4)}$$

Berdasarkan persamaan diatas, D (i, j) menunjukan jarak antara data ke i ke pusat cluster j, X_{ki} menunjukan data ke i pada atribut data ke k, dan X_{kj} menunjukan titik pusat ke j pada atribut ke k.

3. Data ditempatkan dalam cluster yang terdekat, dihitung dari tengah cluster
4. Pusat cluster baru akan ditentukan bila semua data telah ditetapkan dalam cluster terdekat.
5. Proses penentuan pusat cluster dan penempatan data dalam cluster diulangi sampai nilai centroid tidak berubah lagi.

2.6 Gaussian Mixture Model (GMM)

Gaussian Mixture Model merupakan algoritma pengelompokan lunak. Setiap cluster dianggap sebagai model generatif dengan mean dan varians. Model campuran adalah untuk memperkirakan parameter distribusi probabilitas seperti mean dan varians.

Langkah-langkah untuk algoritma Gaussian Mixture Model sebagai berikut: (Iyer dkk, 2016).

1. Menginisialisasi nilai μ_k, σ_k, π_k secara random untuk keseluruhan cluster. μ adalah *mean*, σ adalah *variance*, π adalah coefficient campuran, dan k merupakan angka yang menunjuk ke suatu mixture dalam distribusi gaussian, dan k ekuivalen sebagai nilai yang menunjuk ke suatu cluster.
2. Mengevaluasi hasil log-likehood dengan menggunakan parameter μ_k, σ_k, πk. C adalah cluster, ρ adalah probabilitas, i adalah angka yang menunjuk ke suatu distribusi gaussian dan X adalah distribusi gaussian. Misalkan *cluster* C_k diwakili oleh distribusi Gaussian (μ_k, σ_k), maka probabilitas X_i apa pun milik *cluster* C_k dihitung dengan persamaan :

$$\rho(C_k|x_i) = \frac{\rho(x_i|C_k)*\rho(C_k)}{\rho(x_i)} \quad \textbf{(2. 5)}$$

Dilanjutkan dengan menghitung nilai kemungkinan (*likelihood*) :

$$\rho(x_i|C_k) = \frac{1}{\sqrt{2\pi_k}\sigma} * exp\left(\frac{-(x_i-\mu k)^2}{2\sigma^2}\right) \quad \textbf{(2. 6)}$$

Dan nilai evaluasi (*Evidence*) :

$$\rho(x_i) = \sum_k \rho(x_i|C_k) * \rho(C_k) \quad \textbf{(2. 7)}$$

3. Mengubah nilai μ_k, σ_k, $\rho(C_k)$ dengan melakukan proses perhitungan seperti pada persamaan berikut:

$$\mu_k = \frac{\sum_i \rho(C_k|x_i)*x_i}{\sum_i \rho(C_k|x_i)} \quad \textbf{(2. 8)}$$

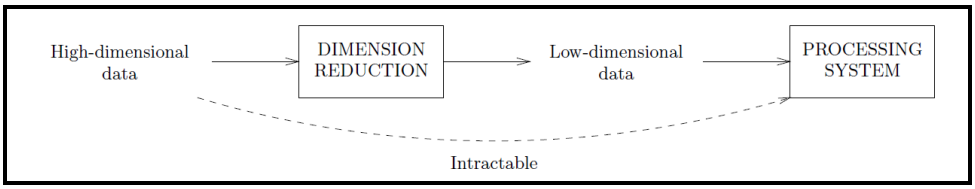
$$\sigma_k = \frac{\sum_i \rho(C_k|x_i)*(x_i-\mu_k)^2}{\sum_i \rho(C_k|x_i)} \quad \textbf{(2. 9)}$$

$$\pi_k = \frac{\sum_i \rho(C_k|x_i)}{n} \quad \textbf{(2. 10)}$$

4. Ulangi langkah 2 dan 3 hingga kriteria konvergensi terpenuhi. Untuk konvergensi, tentukan nilai ambang tertentu untuk perubahan means dan varians dalam iterasi yang berurutan.

2.7 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) adalah teknik transformasi linear *unsupervised* yang digunakan untuk ekstraksi fitur dan pengurangan dimensi. PCA bertujuan untuk menemukan arah varians maksimum dalam data dimensi tinggi dan memproyeksikannya ke subruang baru dengan dimensi yang lebih sedikit daripada yang asli (Kambhatla & Leen, 1997).



Gambar 2.1 Dimensional Reduction (PCA)

Data dimensi tinggi yang tidak direduksi akan menyebabkan *Curse of Dimensionality*, yang mengindikasikan terlalu banyaknya *features*. *Curse of Dimensionality* dapat menyebabkan masalah saat *overfitting model* sehingga hasil performa yang di dapat menjadi buruk, dan juga data menjadi tidak dapat divisualisasikan (Carreira, 1997).

Algoritma Principal Component Analysis (PCA) dengan metode Covariance Matrix ini secara umum dilakukan sebagai berikut: (Kambhatla & Leen, 1997)

1. Menghitung nilai rata-rata dari semua sample seperti pada persamaan 2.11:

$$\mu = \frac{1}{N}\sum_{i=1} x_i \qquad \textbf{(2. 11)}$$

X adalah data matriks (X = [$x_1, x_2, \ldots, x_N$]), N adalah jumlah total sampel, i adalah angka yang menunjuk ke suatu data matriks, μ adalah total rata-rata dari semua sampel.

2. Mengurangi semua sampel dengan total rata-rata dari semua sampel, seperti pada persamaan 2.12:

$$D = \{ d_1, d_2, \ldots, d_N \} = \sum_{i=1}^{N} x_{i-\mu} \qquad \textbf{(2. 12)}$$

D adalah matriks yang berisikan data baru dengan Mean-Centring Data (setiap sampel di kurangi dengan total rata-rata dari semua sampel)

3. Menghitung nilai kovarian matriks, seperti pada persamaan 2.13:

$$\sum = \frac{1}{N-1} D \; x \; D^t \qquad \textbf{(2. 13)}$$

$\sum$ adalah kovarian matriks, D^t adalah transpose matriks D

4. Menghitung *eigenvectors* dan *eigenvalues* dari kovarian matriks, dengan persamaan 2.14:

$$V \sum = \prec V \qquad \textbf{(2. 14)}$$

V adalah *eigenvector* dari kovarian matriks ($\sum$), $\prec$ adalah *eigenvalues* dari kovarian matriks ($\sum$).

5. Mengurutkan *eigenvectors* sesuai dengan *eigenvalues* dalam urutan menurun.

6. Memilih *eigenvectors* yang memiliki *eigenvalues* terbesar, *eigenvectors* yang terpilih akan mewakili ruang proyeksi PCA, yang selanjutnya direpresentasikan dengan $W = \{V_1, \dots, V_k\}$. W adalah dimensi yang lebih rendah (PCA Space), k adalah nilai yang menunjuk dimensi dari *W*.

7. Semua data sampel selanjutnya diproyeksikan pada ruang dimensi yang lebih rendah (*W*) seperti pada persamaan 2.15:

$$Y = W^T D = \sum_{i=1}^{N} W^t (x_i - \mu) \qquad \textbf{(2. 15)}$$

Y adalah hasil data yang diproyeksikan pada ruang dimensi yang lebih rendah, W^T adalah transpose dari matriks W.

2.8 Sillhouete Coeficient

Menurut Hudin dkk (2018), *Silhoutte Coefficient* merupakan salah satu metode yang digunakan untuk menguji kualitas dan kekuatan dari sebuah *cluster*. Metode *silhouette coefficient* merupakan gabungan dari metode *cohesion* dan metode *separation*. Metode *cohesion* sendiri merupakan suatu metode yang digunakan untuk mengukur seberapa dekat relasi antar objek dalam satu *cluster* yang sama. Sedangkan metode *separation* digunakan untuk mengukur seberapa jauh sebuah *cluster* terpisah dengan *cluster* yang lain.

Silhouette memiliki tiga tahap dalam perhitungannya, Berikut tahap perhitungan silhouette coefficient: (Yuan & Yang, 2019)

1. Menghitung perbedaan jarak rata-rata antara titik ί dan semua titik data lainnya di *cluster* yang sama dengan menggunakan persamaan:

$$\alpha(\text{ί}) = \frac{1}{|C_i|-1} \sum^{j \in C_i, j \neq i} d(i,j) \qquad \textbf{(2. 16)}$$

$d(i,j)$ adalah jarak antara titik i dan j pada *Cluster* C_i. α(ί) dapat digunakan sebagai ukuran seberapa baik suatu titik ί ditetapkan di *cluster*-nya, dengan indikasi semakin kecil nilainya maka semakin baik.

2. Menghitung rata-rata jarak minimum dari titik ί ke semua cluster lain (C_k). Proses perhitungan tersebut dilakukan dengan menggunakan persamaan:

$$b(i) = \min \frac{1}{|C_k|} \sum^{j \in C_k} d(i,j) \qquad \textbf{(2. 17)}$$

Proses perhitungan dimulai dengan menghitung perbedaan rata-rata dari titik ί dengan titik j ke beberapa *cluster* lain (C_k) sebagai rata-rata jarak dari titik ί dengan titik j pada C_k, dimana C_k tidak sama dengan C_i. Selanjutnya, menentukan nilai jarak yang terkecil antara titik ί dengan titik j pada C_k

dimana titik í bukan merupakan anggota dari C_k. Cluster dengan perbedaan rata-rata terkecil ini disebut sebagai cluster tetangga dari titik í, karena merupakan *cluster* paling cocok berikutnya untuk titik í.

3. Kemudian menghitung nilai silhouette dengan menggunaan persamaan:

$$S(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \quad \textbf{(2. 18)}$$

Nilai Silhouette berkisar antara + 1 dan – 1. Nilai Silhouette tinggi (mendekati +1) menunjukan pemisahan yang tinggi antar *cluster*, Apabila Nilai Silhouette rendah (mendekati -1) mengindikasikan tumpang tindih antar *cluster*.