BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen merupakan sebuah riset komputasional yang bertujuan untuk menganalisis opini, sentimen, dan emosi. Tujuannya adalah untuk melihat pandangan atau pendapat terhadap suatu masalah atau untuk mengidentifikasi kecenderungan pada pasar. [10]. Analisis sentimen adalah suatu proses yang bertujuan untuk menentukan apakah isi dari *dataset* tersebut memiliki sentimen yang bersifat positif, negatif, ataupun netral. [11].

2.2 Text Preprocessing

Text preprocessing adalah proses penyesuaian data teks menjadi data yang sesuai dengan kebutuhan pemrosesan dalam *data mining* dan diubah menjadi nilai numerik [12]. Adapun tahapan-tahapan pada saat melakukan *text processing* yaitu:

1. *Data Cleaning*
   Membersihkan teks dari informasi yang tidak relevan atau tidak diinginkan seperti tanda baca, karakter khusus, dan angka. Tujuannya untuk mempersiapkan data yang akan digunakan sebelum dianalisis, tanda baca atau karakter yang tidak diinginkan dapat mempengaruhi hasil analisis.

2. *Case Folding*
   Proses pengubahan seluruh huruf teks menjadi huruf kecil ataupun besar, tergantung pada kebutuhan. Tujuannya adalah untuk mempermudah proses analisis, karena huruf kecil maupun besar dianggap sama.

3. *Tokenization*
   Proses pemecahan teks menjadi satuan kata atau frasa. yang bertujuan untuk mempersiapkan teks agar dapat diproses secara efisien dan akurat.

4. *Stopword Removal*
   Merupakan penghapusan kata yang umumnya tidak memiliki makna khusus dan cenderung sering muncul dalam teks, seperti ”dan”, ”atau”, ”yang”, ”di”, dll. Kata-kata ini dianggap tidak relevan dan dapat mempengaruhi hasil analisis.

5. *Stemming*

   Proses dalam pengolahan teks yang berperan untuk menghapus imbuhan pada setiap kata dan mengembalikan kata tersebut ke bentuk dasarnya.

2.3 Term Frequency - Inverse Document Frequency (TF-IDF)

TF-IDF digunakan untuk memberikan bobot antara kata dengan dokumen. Bobot ini dihitung dengan menggabungkan dua buah konsep, yaitu frekuensi kemunculan kata di dalam dokumen dan inverse frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di dalam dokumen menjadikannya kata tersebut penting di dalam dokumen, sedangkan jumlah dokumen yang berisi kata tersebut menunjukkan seberapa sering kata tersebut muncul. Semakin tinggi frekuensi kata di dalam dokumen dan semakin rendah frekuensi keseluruhan dokumen yang mengandung kata tersebut, maka semakin besar bobot hubungan antara kata dan dokumen tersebut [13].

$$tf = 0,5 + 0,5\ x\ \frac{tf}{max(tf)} \tag{2.1}$$

$$idf = log\frac{D}{df_t} \tag{2.2}$$

$$W_{d,t} = tf_{d,t}\ x\ idf_{d,t} \tag{2.3}$$

Keterangan :

1. tf = Jumlah kata yang dicari pada dokumen.
2. $max(tf)$ = Jumlah kemunculan *term* terbanyak pada dokumen yang sama.
3. D = Total keseluruhan dokumen.
4. df_t = Jumlah dokumen yang mengandung *term* t.
5. idf = *Inversed Document Frequency*
6. d = Dokumen ke-d.
7. t = Dokumen ke-t.
8. W = Bobot dokumen.

2.4 Naïve Bayes

Naïve Bayes merupakan metode klasifikasi dalam pembelajaran mesin yang didasarkan pada teorema Bayes. Metode ini berguna dalam memprediksikan kelas dari suatu data berdasarkan informasi data tersebut. Secara khusus, algoritma ini menggunakan asumsi naive atau sederhana bahwa setiap fitur atau atribut pada data adalah independen satu sama lain. Algoritma Naïve Bayes merupakan metode klasifikasi menggunakan probabilitas dan perhitungan statistik. Algoritma ini dianggap sebagai penyederhanaan nilai atribut bersyarat independen ketika nilai output diberikan [14]. Berikut rumus Naïve Bayes pada persamaan 1 [15].

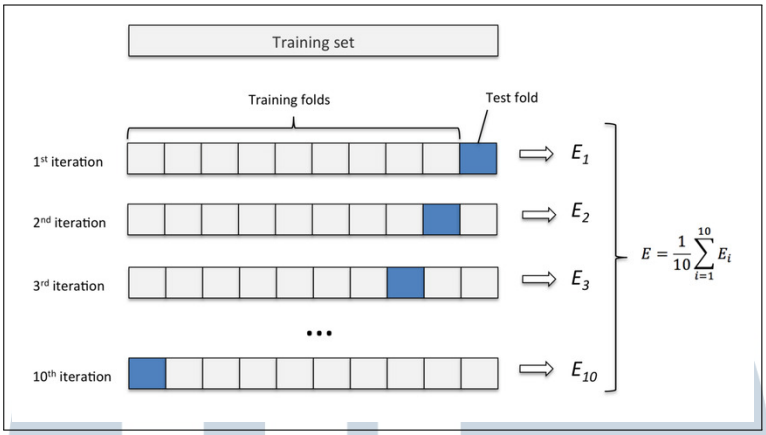
$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \tag{2.4}$$

Keterangan :

1. X = Data dengan kelas yang tidak diketahui.
2. H = Hipotesis data adalah kelas yang spesifik.
3. P(H—X) = Probabilitas H berdasarkan kondisi X.
4. P(H) = Probabilitas H.
5. P(X—H) = Probabilitas X berdasarkan kondisi dari H.
6. P(X) = Probabilitas X.

2.5 K-Fold Cross Validation

K-Fold Cross Validation merupakan metode untuk menilai keakuratan hasil analisis. metode validasi model yang membagi data menjadi k himpunan bagian (*fold*) dengan ukuran yang sama. Setiap subset secara bergantian digunakan sebagai data uji sedangkan subset lainnya digunakan sebagai data latih. Proses ini diulangi beberapa kali, dengan menggunakan setiap subset sebagai data uji, sehingga setiap subset berfungsi sebagai data uji dan latih. Metode ini dilakukan supaya setiap data berkesempatan untuk menjadi data latih dan data uji [16]. Contoh ilustrasi dapat dilihat pada Gambar 2.1



Gambar 2.1. Ilustrasi *K-Fold Cross Validation*
sumber: [2]

2.6 Confusion Matrix

Confusion Matrix merupakan metode yang umum digunakan untuk menghitung tingkat akurasi pada algoritma *data mining* [17]. *Confusion Matrix* berisi informasi tentang jumlah prediksi yang benar (*true positive* dan *true negative*) dan jumlah prediksi yang salah (*false positive* dan *false negative*) dari model. Confusion matrix sangat berguna untuk menghitung *accuracy*, *precision*, *recall*, *F1-score*. Berikut Tabel 2.1 dari *Confusion Matrix*.

Tabel 2.1. *Confusion Matrix*

Class	*Positive*	*Negative*
Positive	*True Positive* (TP)	*False Positive* (FP)
Negative	*False Negative* (FN)	*True Negative* (TN)

Keterangan :
True Positive = Jumlah data positif yang diprediksi positif.
False Positive = Jumlah data negatif yang diprediksi positif.
False Negative = Jumlah data positif yang diprediksi negatif.
True Negative = Jumlah data negatif yang diprediksi negatif.
Selanjutnya menghitung metrik evaluasi berdasarkan Tabel 2.1.

1. *Accuracy*

   Accuracy adalah ukuran evaluasi yang menunjukkan proporsi keseluruhan prediksi yang benar [18].

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.5}$$

2. *Precision*

   Precision adalah rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. [18].

$$Precision = \frac{TP}{TP + FP} \tag{2.6}$$

3. *Recall*

   Recall adalah rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif yang ada dalam kumpulan data [18].

$$Recall = \frac{TP}{TP + FN} \tag{2.7}$$

4. *F1-Score*

   F1-Score atau *F-Measure* merupakan parameter tunggal ukuran keberhasilan retrieval yang menggabungkan *Recall* dan *Precision*. *F1-Score* dihitung dengan mengalikan *Precision* dan *Recall* kemudian dibagi dengan jumlah keduanya, kemudian hasilnya dikalikan dengan dua [19].

$$F1\text{-}Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision\ +\ Recall} \tag{2.8}$$