BAB 2
LANDASAN TEORI

2.1 Berita Hoaks

Berita hoaks merupakan informasi yang belum dapat dipastikan kebenarannya atau tidak benar-benar terjadi. Adapun tujuan dari penyebaran berita hoaks, diantaranya sebagai bahan candaan, promosi, bahkan sebagai sarana untuk menjelekkan suatu individu atau kelompok. Dengan itu, penerima informasi hoaks biasanya mudah terpancing dan menyebarkannya lagi kepada teman-temannya yang membuat berita hoaks tersebar dengan cepat terutama melalui media sosial. Menurut Dewan Pers, ciri-ciri berita hoaks sebagai berikut [13] :

1. Dapat menyebabkan kecemasan, permusuhan antar individu / kelompok dan kebencian.
2. Sumber informasi yang tidak jelas. Pada media sosial, biasanya berita hoaks disebar oleh media yang tidak jelas dan isi beritanya menyudutkan suatu pihak.
3. Informasi yang ditulis oleh penyebar hoaks biasanya fanatisme terhadap suatu ideologi, selain itu judul yang provokatif dan memberikan informasi tanpa data dan fakta yang ada.

2.2 Count Vectorizer

CountVectorizer merupakan salah satu *library* dalam *machine learning* yang dapat membantu proses pembuatan model *Natural Language Processing* dengan mengambil fitur-fitur teks. CountVectorizer mengambil fitur-fitur teks dengan melakukan perubahan teks menjadi sebuah matriks yang didalamnya memuat jumlah kemunculan setiap kata dalam teks tersebut [14]. CountVectorizer dapat mengubah fitur teks menjadi sebuah representasi vektor.

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

TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah metode yang digunakan untuk menentukan nilai frekuensi sebuah kata dalam sebuah

dokumen/artikel serta frekuensinya diberbagai dokumen. Algoritma TF-IDF biasanya digunakan dalam pengolahan data besar [15]. Algoritma ini memberikan bobot pada setiap kata kunci di tiap kategori untuk menemukan kesamaan antara kata kunci dengan kategori yang ada. Sebelum melakukan pembobotan, terlebih dahulu dilakukan *text processing*, kemudian dilanjutkan dengan perhitungan bobot TF-IDF, bobot relevansi query, dan bobot *similarity*. Perhitungan bobot TF-IDF diawali dengan menghitung nilai *Term Frequency*, singkatnya melakukan perhitungan jumlah munculnya kata. Selanjutnya menghitung nilai *Inverse Document Frequency* yang menghitung jumlah dokumen yang mengandung kata tersebut. Apabila semakin banyak kata tersebut ditemukan pada dokumen lain, maka kata tersebut akan dianggap kurang penting. Perhitungan matematis untuk TF-IDF yakni sebagai berikut:

$$W_{dt} = tf_d \times idf_t \tag{2.1}$$

$$idf_t = \log\left(\frac{N}{df_t}\right) \tag{2.2}$$

Keterangan rumus 2.1 dan 2.2=

W_{dt} = Nilai bobot kata ke-t di dokumen d

tf_d = Jumlah kemunculan kata t di dokumen d

N = Jumlah keseluruhan dokumen

df_t = Jumlah dokumen yang menggunakan kata t

d = Dokumen ke-d

t = Kata ke-t

2.4 Algoritma Multinomial Naive Bayes

Multinomial Naive Bayes merupakan variasi model pengembangan algoritma Naive Bayes yang melakukan klasifikasi teks / dokumen dengan hasil yang akurasi yang baik. Rumus yang digunakan untuk memperhitungkan kelas pada suatu dokumen yaitu [16]:

$$C_{\text{map}} = \arg\max_{c \in C} P(c) \prod_{n=1}^{k} P(t_n \mid c) \tag{2.3}$$

Keterangan rumus 2.3=

C_{map} = Probabilitas suatu dokumen termasuk kelas c

P(c) = Probabilitas *prior* dari kelas c

$P(t_n|c)$ = Probabilitas kata ke-n dengan diketahui kelas c

Rumus MNB yang digunakan dengan pembobotan kata TF-IDF adalah sebagai berikut:

$$P(t_n \mid c) = \frac{W_{ct} + 1}{(\sum W' \in VW') + B'} \tag{2.4}$$

Keterangan rumus 2.4=

W_{ct} = Nilai pembobotan TF-IDF atau W dari kata t di kelas c

$\sum W' \in VW'$ = Jumlah total W dari keseluruhan kata yang berada di kelas c

B' = Jumlah W dari kata unik

2.5 Black Box Testing

Black box testing adalah sebuah metode pengujian *software* yang fokus pada pemeriksaan pemeriksaan fungsi aplikasi berdasarkan masukan dan keluaran yang diharapkan. Penggunaan metode ini dilakukan adalah agar memastikan aplikasi yang dibuat sesuai dengan yang diharapkan tanpa memperhatikan bagaimana aplikasi tersebut bekerja secara internal [17]. Pengujian ini dilakukan berdasarkan persyaratan dan spesifikasi perangkat lunak yang telah ditentukan. Metode ini memungkinkan penguji untuk melakukan pengujian tanpa perlu mengetahui detail kode internal dari aplikasi yang diuji. Pengujian ini mencakup pemeriksaan terhadap masukan yang valid dan tidak valid sesuai dengan kebutuhan.

2.6 Confusion Matrix

Confusion matrix merupakan matriks yang digunakan untuk mengevaluasi performa dari sebuah model klasifikasi yang telah dibuat. Matriks ini akan membandingkan nilai aktual dengan hasil prediksi dari model machine learning. Ukuran dari matriks ini bergantung kepada jumlah target class dari label yang ingin diprediksi. Berikut adalah gambaran dari confusion matrix yang ditunjukkan pada Tabel 4.6.

Tabel 2.1. Tabel *confusion matrix*

	Classifier says YES	Classifier says NO
In reality YES	True positive	False Negative
In reality NO	False Positive	True Negative

Umumnya Confusion matriks akan digunakan untuk menghasilkan 4 buah nilai yaitu accuracy, precision, recall, dan F1 score [18].

Accuracy merupakan seberapa akurat model dalam mengklasifikasikan dalam bentuk persentase [19].

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

Precision menunjukkan nilai kecocokan antara data yang diminta dengan hasil prediksi yang diberikan oleh model [19].

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

Recall menggambarkan keberhasilan model dalam menemukan kembali suatu informasi [19].

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

F1 menunjukkan perbandingan dari nilai rata-rata *precision* dan *recall* yang telah dibobotkan [19].

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{2.8}$$