BAB 2
LANDASAN TEORI

2.1 Data Preprocessing

Data preprocessing adalah sebuah cara untuk mengubah data mentah menjadi bentuk yang diinginkan sehingga informasi yang berguna dapat diperoleh darinya [8]. Proses ini melibatkan serangkaian langkah seperti membersihkan data, menghilangkan *noise*, dan mengorganisir data agar sesuai dengan kebutuhan analisis. Dalam penelitian ini, digunakan beberapa teknik *preprocessing* sebagai berikut:

- *Data Cleaning*: *Data cleaning* adalah proses penting untuk mempersiapkan data teks sebelum diolah oleh model. Langkah-langkah ini melibatkan penghapusan elemen yang tidak relevan atau mengganggu, seperti URL, tautan, atau tanda baca yang tidak dibutuhkan [9].
- *Case Folding*: Proses ini bertujuan untuk mengubah huruf kapital yang terdapat dalam *dataset* menjadi huruf kecil sehingga semua karakter menjadi seragam. Dengan membuat semua kata menjadi huruf kecil, ini akan sangat membantu dalam membuat generalisasi [9].
- *Tokenization*: Bertujuan untuk memecah kalimat menjadi potongan-potongan kata, tanda baca, dan ekspresi bermakna lainnya sesuai dengan ketentuan bahasa yang digunakan [9].
- *Stopwords Removal*: Proses menghilangkan kata-kata yang tidak memiliki makna [9].
- *Lemmatization*: Proses yang dilakukan untuk mengidentifikasi dan mengubah sebuah kata ke bentuk dasar dengan tujuan mempermudah proses pengolahan data [10].
- *Undersampling*: Menghapus atau mengurangi sampel dari kelas mayoritas sehingga seimbang dengan jumlah sampel dari kelas minoritas [11].

2.2 Natural Language Processing

Natural Language Processing (NLP) adalah sebuah bagian dalam ilmu komputer dan *artificial intelligence* yang berfokus pada interaksi mesin dengan manusia [12]. Dalam pengembangannya, NLP bertujuan untuk menangani berbagai tugas dan aplikasi, termasuk tetapi tidak terbatas pada pemahaman makna, sintaksis, dan struktur kalimat, pengenalan entitas, serta interpretasi konteks dalam teks [13].

2.3 Naïve Bayes

Naïve Bayes adalah sebuah teknik klasifikasi statistik berdasarkan prinsip Bayes. Dalam *Naïve Bayes*, istilah *Naïve* mengacu pada gagasan bahwa fitur-fitur independen satu sama lain, yang seringkali tidak realistis, namun dapat menyederhanakan model [14]. *Naïve Bayes* menghitung probabilitas dari sebuah sampel yang termasuk dalam sebuah kelas dengan menggabungkan probabilitas sebelumnya dengan kemungkinan probabilitas. Kelas dengan probabilitas tertinggi akan dipilih menjadi *predicted class* [15]. Walaupun terlihat sederhana, *Naïve Bayes* memiliki performa yang baik, khususnya dalam klasifikasi teks. Persamaan dari *Naïve Bayes* adalah sebagai berikut.

$$P(C|X) = \frac{P(C) \cdot P(X|C)}{P(X)} \tag{2.1}$$

Di mana:

$P(C|X)$ adalah probabilitas posterior kelas C dengan fitur X.

$P(C)$ adalah probabilitas *prior* kelas C.

$P(X|C)$ adalah probabilitas *likelihood* fitur X diberikan kelas C.

$P(X)$ adalah probabilitas fitur X.

2.4 Multinomial Naïve Bayes

Multinomial Naïve Bayes adalah sebuah varian dari algoritma *Naïve Bayes* yang seringkali digunakan dalam klasifikasi dokumen karena efisiensi komputasi dan performa prediksinya [16]. *Multinomial Naïve Bayes* memperhitungkan frekuensi dari setiap kata yang muncul pada suatu dokumen, yang direpresentasikan dengan bilangan bulat seperti 0,1,2, dan seterusnya [17]. *Multinomial Naïve Bayes*

dirancang khusus untuk klasifikasi teks, beroperasi dengan asumsi distribusi fitur multinomial yang mewakili frekuensi kata [18].

2.5 Bag Of Words

Bag Of Words (BOW) merupakan sebuah metode *feature extraction* untuk mengubah data teks menjadi vektor input berukuran tetap [19]. Melalui normalisasi vektor, BOW menghitung frekuensi kemunculan dari setiap kata secara unik, yaitu setiap kata berulang hanya akan ditulis sekali. BOW tidak memperhatikan urutan atau struktur kata dalam sebuah dokumen, melainkan hanya memperhatikan apakah sebuah kata terdapat dalam dokumen [20]. Dalam kasus klasifikasi teks, BOW memiliki akurasi yang lebih tinggi dibandingkan dengan *feature extraction* lain, seperti *Term Frequency-Inverse Document Frequency*, dan *one-hot encoding* [21]. BOW juga memiliki performa yang lebih baik dibandingkan dengan *fastText* saat pengujian klasifikasi teks untuk dataset yang tidak terlalu besar [22]. Meskipun BOW adalah metode yang sangat sederhana dalam representasi teks, BOW memberikan hasil yang memadai untuk banyak tugas dasar NLP seperti klasifikasi dan pengelompokan teks, terutama pada dataset kecil hingga menengah, tanpa memerlukan sumber daya komputasi yang besar [23]. Selain itu, hasil penggunaan BOW mudah diinterpretasikan karena setiap fitur langsung dihubungkan dengan frekuensi kata dalam dokumen, membuat analisis lebih transparan [24].

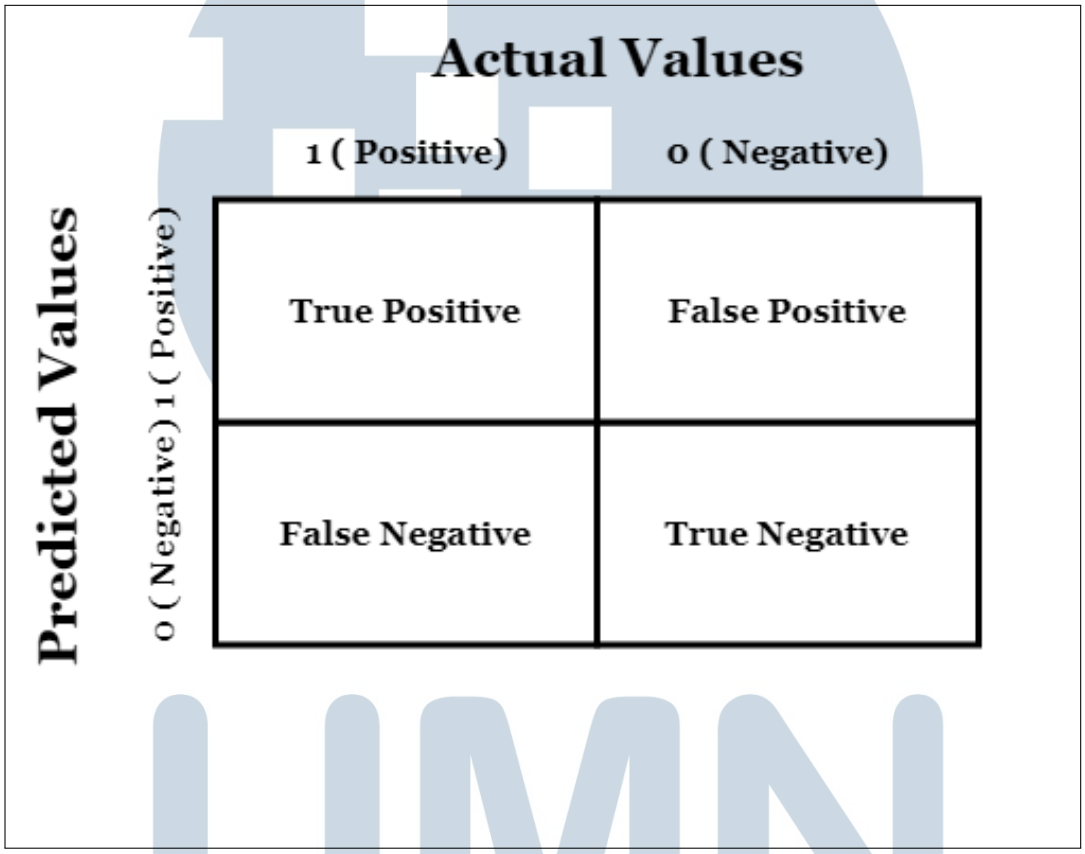
2.6 Grid Search

Grid Search merupakan salah satu teknik *hyperparameter tuning* yang sering digunakan [25]. *Grid search* menentukan *hyperparameter* optimal dengan cara menelusuri setiap konfigurasi optimal dari parameter yang diberikan [26]. *Grid search* dapat membantu untuk meningkatkan akurasi dan efisiensi model, dengan mengidentifikasi dan melakukan percobaan terhadap semua kombinasi *hyperparameter* yang optimal untuk sebuah algoritma [27]. *Grid search* dapat diimplementasikan dengan menggunakan salah satu *library* dari scikit-learn yaitu GridSearchCV.

2.7 Confusion Matrix

Confusion Matrix adalah sebuah alat kunci dalam mengevaluasi performa dari sebuah sistem klasifikasi, yang menyediakan perincian dari klasifikasi

aktual dan prediksi klasifikasi [28]. *Confusion Matrix* berperan penting dalam memvisualisasikan performa dari sebuah algoritme dalam *supervised learning* [29]. Dalam konteks penelitian ini, *Confusion Matrix* dapat memberikan pandangan yang lebih mendalam terkait dengan akurasi dari model klasifikasi. Gambar 2.1 merupakan ilustrasi dari *confusion matrix*.



Gambar 2.1. *Confusion matrix*
Sumber: [30]

Di mana:

True Positive (TP) adalah data kelas positif yang diklasifikasikan positif.
True Negative (TN) adalah data kelas negatif yang diklasifikasikan negatif.
False Positive (FP) adalah data kelas positif yang diklasifikasikan negatif
False Negative (FN) adalah data kelas negatif yang diklasifikasikan positif.

Berdasarkan nilai *Confusion Matrix*, dapat dilakukan pengukuran performa dengan menghitung *precision*, *recall*, dan *F-1 score*.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

$$F1 - Score = 2 * \frac{precision * recall}{precision + recall} \tag{2.5}$$