BAB 2
LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian ini mengadopsi beberapa hal dari beberapa penelitian terdahulu setelah dilakukan studi literatur. Riset gap beberapa penelitian terdahulu dapat dilihat pada Tabel 2.1.

Tabel 2.1. Riset Gap

Penulis	Judul	Jumlah Data	Metode	Hasil Penelitian
Ratih Puspitasari, Yulian Findawati, dan Mochamad Alfan Rosid	Analisis Sentimen Terhadap Inflasi Pasca COVID-19 Berdasarkan Twitter dengan Metode Klasifikasi K-Nearest Neighbor dan Support Vector Machine	5989 Tweet	Algoritma SVM dengan KNN	Algoritma SVM mendapat akurasi tertinggi sebesar 79% dibanding dengan KNN sebesar 54%
Lanjut pada halaman berikutnya				

Tabel 2.1 Riset Gap (lanjutan)

Penulis	Judul	Jumlah Data	Metode	Hasil Penelitian
Ilham Firman Ashari, Fadhillah A., M. Daffa, dan Sekar Ali	Sentiment Analysis of Tweets About Allowing Outdoor Mask Wear Using Naïve Bayes and TextBlob	1000 Tweet	Naive Bayes 20% data *test* dengan 30% data *test*	Algoritma Naive Bayes dengan data *test* 20% menghasilkan akurasi lebih tinggi sebesar 85% dibanding dengan data *test* 30% sebesar 83%
Sri Diantika, Windu Gata, Hiya Nalatissifa, dan Mareanus Lase	Komparasi Algoritma SVM Dan Naive Bayes Untuk Klasifikasi Kestabilan Jaringan Listrik	10000 *instances*, 14 *attributes* dan 2 *attribute class*	SVM dan Naive Bayes	Algoritma SVM mendapat akurasi lebih baik sebesar 98,9% dibanding dengan Naive Bayes sebesar 97,64%
Widia Ningsih, Baginda Alfianda, Rahmaddeni, dan Denok Wulandari	Perbandingan Algoritma SVM dan Naïve Bayes dalam Analisis Sentimen Twitter pada Penggunaan Mobil Listrik di Indonesia	1517 Tweet	SVM dan Naive Bayes	Algoritma SVM mempunyai nilai akurasi tertinggi sebesar 70,82% dibanding dengan Naive Bayes sebesar 63,02%

Terdapat empat penelitian terdahulu dari Tabel 2.1 yang diadopsi. Penelitian pertama berjudul "*Analisis Sentimen Terhadap Inflasi Pasca COVID-19 Berdasarkan Twitter dengan Metode Klasifikasi K-NEAREST NEIGHBOR dan SUPPORT VECTOR MACHINE*" yang membandingkan algoritma *KNN* dengan *SVM*. Dengan total 5989 data *tweet*, algoritma *SVM* mendapat akurasi tertinggi sebesar 79% dibanding *KNN* sebesar 54%. Hal yang didapatkan dari penelitian ini adalah metodologi perbandingan kedua algoritma, hasil performa model, serta *data labelling* menggunakan *VADER* [8].

Penelitian kedua berjudul "*Sentiment Analysis of Tweets About Allowing Outdoor Mask Wear Using Naive Bayes and TextBlob*" membandingkan *testing data* algoritma *Naive Bayes* antara 20% data *test* dan 30% data *test* dengan total 1000 data *tweet*. Hasil akurasi lebih tinggi pada 20% data *test* sebesar 85%. Hal yang didapatkan dari penelitian ini adalah metode *training* dan *testing* data dengan membandingkan data 20% data *test* dan 30% data *test* menggunakan algoritma *Naive Bayes* [7].

Penelitian ketiga berjudul "*Komparasi Algoritma SVM Dan Naive Bayes Untuk Klasifikasi Kestabilan Jaringan Listrik*" yang membandingkan *SVM* dan *Naive Bayes*. Dengan total 1000 *instances*, 14 *attributes*, dan 2 *attribute class*, algoritma SVM mendapatkan akurasi tertinggi sebesar 98,9% dibanding Naive Bayes sebesar 97,64%. Hal yang didapatkan dari penelitian ini adalah metodologi komparasi analisis antara kedua algoritma *SVM* dan *Naive Bayes* serta metode evaluasi model selain *confusion matrix* yaitu *roc curve* dan validasi model menggunakan *K-Fold Cross Validation* [9].

Penelitian keempat berjudul "Perbandingan Algoritma *SVM* dan *Naive Bayes* dalam Analisis Sentimen Twitter pada Penggunaan Mobil Listrik di Indonesia". Dengan 1517 jumlah data, perbandingan antara *Naive Bayes* dan *SVM* menyimpulkan bahwa algoritma *SVM* mempunyai nilai akurasi tertinggi sebesar 70,82% dibanding *Naive Bayes* sebesar 63,02%. Hal yang didapatkan dari penelitian ini adalah Metodologi penelitian yang dilakukan sebelum melakukan *modelling SVM* dan *Naive Bayes* seperti *preprocessing* [10].

2.2 Media Sosial Twitter

Media sosial merupakan platform online yang memfasilitasi pengguna untuk berpartisipasi, berbagi, dan menciptakan konten seperti blog, jejaring sosial, wiki,

forum, serta dunia virtual dengan mudah. Jenis media sosial yang paling umum digunakan oleh masyarakat di seluruh dunia meliputi blog, jejaring sosial, dan wiki. Selain itu, ada pandangan yang menyatakan bahwa media sosial adalah media online yang mendukung interaksi sosial dan menggunakan teknologi berbasis web untuk mengubah komunikasi menjadi dialog yang interaktif. [11].

Salah satu media sosial tersebut adalah *Twitter*. Twitter adalah layanan *microblogging* di media sosial yang secara teori dapat mempublikasikan peristiwa dan kejadian ke dunia opini yang lebih luas. Pengguna dapat mengunggah *tweet* dalam bentuk *URL*, gambar, video, atau teks dengan batas maksimal 280 karakter (huruf, simbol, atau angka) [12]. Namun, menurut laman Tekno Kompas, Twitter secara resmi telah meningkatkan batas maksimum karakter untuk setiap twit. Sekarang para pengguna *twitter* bisa mengunggah twit hingga 4.000 karakter [13].

2.3 *Text Mining*

Text mining adalah algoritma yang digunakan untuk menggali data dengan tujuan memenuhi kebutuhan informasi, melalui metode seperti pembelajaran mesin, pemrosesan bahasa alami, manajemen pengetahuan, dan pencarian informasi. Teknik *Text mining* ini melibatkan pra-pemrosesan dokumen seperti pengkategorian teks, ekstraksi informasi, dan ekstraksi kata, yang berguna untuk mendapatkan informasi dari sumber data dengan mengidentifikasi dan mengeksplorasi pola yang menarik. Meskipun banyak aspek dari *Text mining* mirip dengan penambangan data, perbedaannya terletak pada penggunaan pola yang diambil dari bahasa alami yang tidak terstruktur, berbeda dengan penambangan data yang menggunakan pola dari basis data terstruktur. Tahapan umum dari algoritma *Text mining* dimulai dengan pra-pemrosesan teks [14].

2.4 *Text Preprocessing*

Text preprocessing adalah salah satu bagian pada algoritma Text Mining yang melibatkan pengolahan teks untuk mengubah dokumen menjadi data terstruktur sesuai dengan kebutuhannya agar dapat diolah lebih lanjut dalam proses *Text Mining* [14].

Langkah ini mengonsolidasikan data yang sudah melalui pembersihan, sehingga menjadi satu barisan data yang terintegrasi. Oleh karena itu, proses ini bertujuan untuk membersihkan data dari gangguan dan ketidakkonsistenan,

mentransformasikan data agar konsisten, dan secara otomatis mengurangi duplikasi. Metode ini dapat diterapkan dalam tahap klasifikasi dokumen, klasterisasi, ekstraksi, analisis sentimen, dan pencarian informasi [12]. Tahap *preprocessing* yang akan dilakukan dibagi menjadi enam tahap, yaitu:

1. *Data Cleaning*
   Proses membersihkan data yang berbentuk teks dengan menghapus beberapa karakter seperti angka, tanda baca, *URL* atau *link*, emoji, serta mengganti angka dengan string kosong [15].

2. *Translate*
   Pada tahap ini, data akan diterjemahkan dari bahasa Indonesia ke dalam bahasa Inggris. Penerjemahan data menggunakan *library google translate* dari *python* [8].

3. *Case Folding*
   Proses untuk mengubah seluruh teks dalam dokumen menjadi bentuk standar melibatkan pengubahan semua huruf dalam dokumen tersebut menjadi huruf kecil atau *lowercase*. [15].

4. *Tokenizing*
   Proses memotong *string input* berdasarkan setiap kata penyusunnya, kemudian mengubahnya menjadi array yang terdiri dari kata-kata hasil pemotongan dari kalimat tersebut [16].

5. *Filtering*
   Proses pengambilan kata-kata kunci dari sebuah kalimat dan membuang kata-kata penghubung atau keterangan sesuai dengan daftar *stopwords* yang sudah ditetapkan [17].

6. *Lemmatization*
   Proses untuk melakukan pengelompokan kata-kata berbeda menjadi kata dasar yang sama dengan makna serupa, meskipun memiliki bentuk berbeda akibat imbuhan, termasuk penghilangan awalan, sisipan, dan akhiran. Lemmatisasi digunakan untuk mengekstrak makna kata dari sebuah teks, meningkatkan penalaran dan terjemahan mesin [16].

2.5 Ekstraksi Fitur (*TF-IDF*)

Merupakan proses menghitung atau mengekstraksi kata-kata menjadi angka berbentuk vektor, yang digunakan untuk menentukan bobot suatu kata dalam sebuah dokumen atau kumpulan dokumen. Bobot ini berguna untuk menentukan seberapa penting kata tersebut dalam dokumen. Perhitungan atau rumus *TF-IDF* terbagi menjadi dua yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) memiliki rumus dan cara kerja yang berbeda, tetapi keduanya akan digabungkan pada akhir perhitungan antara *TF* dan *IDF* [18]. Rumus *TF-IDF* dijabarkan sebagai berikut:

A *Term Frequency* (TF)

Menghitung frekuensi kemunculan kata dalam sebuah dokumen. Karena setiap dokumen memiliki panjang yang berbeda-beda, nilai TF kemudian dibagi dengan panjang dokumen:

$$\mathrm{tf}_{t,d} = \frac{\mathrm{n}_{t,d}}{(\textit{Total number of term in document})} \tag{2.1}$$

Keterangan:

Tf = frekuensi kemunculan kata pada sebuah dokumen.

B *Inverse Document Frequency* (IDF)

Setelah menghitung nilai TF, langkah selanjutnya adalah menghitung nilai IDF. Nilai IDF digunakan untuk mengukur tingkat kepentingan sebuah kata. Semakin kecil nilai IDF, semakin kurang penting kata tersebut dianggap dan sebaliknya.

$$\mathrm{idf}_d = \log \frac{\textit{Number of document}}{\textit{Total number of term in document}} \tag{2.2}$$

Keterangan:

Idf = mengukur penting atau tidak sebuah kata dalam dokumen.

C *Term Frequency - Inverse Document Frequency* (TF-IDF)

Setelah mendapatkan nilai TF dan IDF akan dihitung nilai TF-IDF dengan rumus sebagai berikut:

$$\mathrm{tfidf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_d \tag{2.3}$$

Keterangan:

TF-IDF = hasil penggabungan antara TF dan IDF.

2.6 SMOTE

Synthetic Minority Over-sampling Technique (*SMOTE*) adalah teknik *oversampling* yang digunakan untuk menyeimbangkan distribusi jumlah dataset pada kelas minoritas dengan cara mensintesis dataset minoritas hingga jumlahnya setara dengan jumlah dataset pada kelas mayoritas [19].

$$\text{Xsyn} = \text{Xi} + (\text{Xknn} - \text{Xi}) * \sigma \tag{2.4}$$

Keterangan:
X syn = data sintesis yang akan diciptakan.
Xi = data yang akan direplikasi.
Xknn = data yang memiliki jarak terdekat dari data yang akan direplikasi.
σ = nilai random antara 0 dan 1.

2.7 K-Fold Cross-Validation

Teknik pembagian data juga dapat menggunakan metode *K-fold cross validation*. *K-fold cross-validation* adalah metode untuk mengestimasi kinerja model yang telah dibuat [19]. *K-fold cross-validation* secara kontinu membagi data menjadi data latih dan data uji, memastikan bahwa setiap bagian data akan memiliki kesempatan untuk menjadi data uji.

Tabel 2.2. Ilustrasi Pembagian Data dalam *K-Fold Cross Validation*

Fold 1	*Fold 2*	*Fold 3*	...	*Fold K*
Test	Train	Train	...	Train
Train	Test	Train	...	Train
Train	Train	Test	...	Train
Train	Train	Train	...	Train
Train	Train	Train	...	Test

Ilustrasi yang ditampilkan pada Tabel 2.2 menjelaskan bahwa validasi silang *K* kali lipat digunakan dalam percobaan pelatihan model. K merujuk pada jumlah pembagian yang digunakan dalam proses ini untuk memisahkan data latih dan data uji [14].

2.8 Analisis Sentimen

Analisis sentimen adalah sebuah proses yang luas dalam pemrosesan bahasa alami, linguistik komputasional, dan *Text Mining* yang bertujuan untuk mengevaluasi opini, perasaan, penilaian, sikap, dan emosi seseorang. Kategori dalam analisis sentimen ini mencakup pengelompokan polaritas teks dalam sebuah kalimat atau dokumen, di mana teks tersebut diklasifikasikan ke dalam salah satu polaritas sentimen, seperti positif, negatif, atau netral, berdasarkan ruang lingkup teks yang dianalisis [20].

2.9 *VADER (Valence-Aware Dictionary and sEntiment Reasoner)*

Vader menghitung sentimen teks dengan mengkategorikan fitur leksikal, seperti kata -kata, sebagai positif atau negatif berdasarkan orientasi semantik kata tersebut. *Vader* dapat dioptimalisasikan untuk data media sosial dan berkinerja baik ketika dikombinasikan dengan data dari *Twitter*, *Facebook*, dan jaringan media sosial lainnya. Hasilnya mengungkapkan kata polaritas dan probabilitas menjadi positif, negatif, netral, atau kompleks [21].

Menurut Hutto dan Gilbert, setiap fitur leksikal memiliki skor rata-rata nol dan standar deviasi kurang dari 2.5. Ada lebih dari 7500 fitur leksikal dengan skor valensi tervalidasi yang menunjukkan polaritas sensorik (positif/negatif) dan intensitas perasaan pada skala -4 (negatif), 0 (netral), dan 4 (positif). Misalnya, kata 'okay' memiliki skor 0.9, '*good*' 1.9, '*bad*' -2.5, dan '*severed*' -1.5. Setiap teks akan dianalisis oleh Vader dan menghasilkan skor positif, negatif, atau netral. Semua skor tersebut kemudian dijumlahkan untuk membentuk nilai *compound*, yaitu matriks yang menghitung semua skor yang dinormalisasi dalam rentang -1 hingga +1 [22]. Nilai *treshold* yang umum digunakan sudah berdasarkan skor akhir yang dinormalisasi dalam rentang -1 hingga +1, sehingga para ahli dan pembuat *VADER* itu sendiri menetapkan *treshold* adalah sebagai berikut:

- Sentimen positif: *Compound score* >= 0,05
- Sentimen netral: (*Compound score* > -0,05) dan (*Compound score* < 0,05)
- Sentimen negatif: *Compound score* <= -0,05

Berikut adalah rumus menghitung *compound score* yang menghitung jumlah skor valensi (valency score):

$$x = \frac{x}{\sqrt{x^2 + \alpha}} \tag{2.5}$$

Keterangan:

x = total penjumlahan *valence scores* kata - kata penyusun teks

α = nilai *default* konstanta normalisasi = 15

2.10 *Naive Bayes Classifier*

Salah satu metode untuk klasifikasi melibatkan pengelompokan teks atau kalimat ke dalam kategori yang sesuai. Klasifikasi naive bayes adalah pendekatan probabilistik yang mengevaluasi probabilitas setiap kata dalam teks atau kalimat, dengan kata-kata berprobabilitas tertinggi dianggap sebagai hasil klasifikasi [23]. Pada tahap pelatihan, proses dilakukan terhadap sampel data yang diusahakan untuk mewakili data tersebut. Probabilitas prior untuk setiap kategori ditentukan berdasarkan sampel data tersebut. Pada tahap klasifikasi, kategori suatu data ditentukan berdasarkan istilah yang muncul dalam data yang diklasifikasikan [20]. Persamaan umum dari klasifikasi naive bayes adalah sebagai berikut:

$$P(C|X) = \frac{(P(X|C) \times P(C))}{P(X)} \tag{2.6}$$

Keterangan:

- X = data kelas yang belum diketahui
- C = hipotesis data X
- $P(C|X)$ = probabilitas hipotesis X berdasarkan kondisi C
- P(C) = probabilitas hipotesis C
- P(X) = probabilitas dari X

Secara keseluruhan, teorema Bayes menyatakan bahwa titik data x merupakan representasi dari kelas yang belum diketahui, sementara kelas c merujuk pada kelas spesifik yang terkait dengan data tersebut. Probabilitas hipotesis berdasarkan kondisi dilambangkan sebagai $P(C|X)$, sedangkan probabilitas kondisi berdasarkan hipotesis dinyatakan sebagai $P(X|C)$. P(C) menunjukkan probabilitas prior, sementara P(X) menggambarkan probabilitas dari C [23]. Kernel yang

digunakan pada penelitian ini adalah *Bernoulli* dikarenakan kernel tersebut berfungsi dengan baik dalam metode mengklasifikasikan suatu artikel dalam kumpulan dataset hingga mencapai performa yang tinggi [24].

2.11 *Support Vector Machine*

Support Vector Machine (SVM) merupakan metode klasifikasi yang memanfaatkan konsep penentuan hyperplane optimal untuk memisahkan kelompok data. SVM adalah sistem pembelajaran yang menerapkan fungsi linear dalam ruang fitur berdimensi tinggi guna melakukan prediksi [20]. Untuk mendapatkan *hyperplane* yang optimal dalam membedakan dua kelas data, dilakukan perhitungan *margin hyperplane* dan penentuan titik maksimal *margin* tersebut. Dalam memperoleh *hyperplane* pada *support vector machine*, dapat menggunakan persamaan sebagai berikut:

$$(w \cdot x_i) + b = 0 \tag{2.7}$$

Pada data x_i, yang termasuk pada kelas -1 dapat dirumuskan sebagai persamaan berikut:

$$(w \cdot x_i) + b \leq 1, y_i = -1 \tag{2.8}$$

Sedangkan data - data x_i, yang termasuk pada kelas +1 dapat dirumuskan sebagai persamaan berikut:

$$(w \cdot x_i) + b \geq 1, y_i = -1 \tag{2.9}$$

Dalam proses klasifikasi menggunakan *support vector machine*, sering kali menghasilkan data input yang dapat dipisahkan secara linear dan membentuk *hyperplane* yang optimal [25]. Terdapat beberapa fungsi kernel, yaitu kernel *linear*, *RBF*, *polinomal*, dan *sigmoid*. Pada penelitian ini menggunakan *kernel rbf* dari *library SVC* yang berasal dari *sklearn*

2.12 *Confusion Matrix*

Salah satu cara untuk evaluasi kinerja algoritma adalah *Confusion Matrix*. Confusion matrix adalah tabel yang berisi kombinasi nilai hasil klasifikasi. Dalam confusion matrix, ada empat istilah: *True Positive* (TP), *True Negative* (TN), *False*

Positive (FP), dan *False Negative* (FN) seperti yang ditampilkan pada Tabel 2.3. *False Positive* (FP) terjadi ketika data yang sebenarnya negatif diklasifikasikan sebagai positif, sedangkan *True Negative* (TN) adalah data yang benar-benar negatif dan diidentifikasi dengan benar sebagai negatif [26].

Tabel 2.3. *Confusion Matrix* yang terdiri dari prediksi dan nilai aktual positif dan negatif.

Prediction	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Berdasarkan hasil dari *confusion matrix* pada Tabel 2.3, dapat diperoleh nilai akurasi, *recall*, *precision*, dan *f-1 score* [12].

Akurasi adalah ukuran seberapa dekat nilai prediksi dengan nilai yang sebenarnya. Rumus akurasi sama seperti pada Rumus (2.10)

$$Accuracy = \frac{TP + TN}{Total} \tag{2.10}$$

Nilai *precision* adalah ukuran presisi antara data yang tersedia dengan hasil prediksi [27]. Rumus *precision* seperti pada Rumus (2.11).

$$Precision = \frac{TP}{TP + FP} \tag{2.11}$$

Recall menunjukkan seberapa baik model berhasil menemukan informasi yang relevan. Rumus *recall* seperi pada Rumus (2.12).

$$Recall = \frac{TP}{TP + FN} \tag{2.12}$$

Nilai *F-1 Score* adalah perbandingan rata-rata antara precision dan recall. Nilai ini dihitung menggunakan rumus tertentu [12]. Nilai ini dihitung menggunakan rumus yang ada pada Rumus (2.13).

$$F - 1\,Score = \frac{2 * precision * recall}{precision + recall} \tag{2.13}$$

2.13 *ROC - AUC Score*

Untuk menampilkan informasi kinerja algoritma klasifikasi dalam bentuk grafik, dapat digunakan kurva *Receiver Operating Characteristic* (ROC). Kurva *ROC* dibuat berdasarkan nilai yang diperoleh dari perhitungan dengan confusion matrix, yaitu antara *False Positive Rate* dan *True Positive Rate*. Untuk membandingkan kinerja setiap algoritma, dapat dilakukan dengan membandingkan luas di bawah kurva atau *Area Under Curve* (AUC). Keuntungan dari penggunaan kurva *ROC* dalam evaluasi klasifikasi adalah *ROC* tidak hanya mencari rata-rata akurasi, tetapi juga memvisualisasikan semua *threshold* klasifikasi yang mungkin. Sementara itu, *error rate classifier* hanya mewakili tingkat kesalahan dan akurasi untuk satu *threshold* saja [28].