BAB II

TINJAUAN PUSTAKA

Pada bab ini menjelaskan teori serta bahan-bahan penelitian lainnya, juga akan diarahkan pada pengembangan konsep-konsep yang berkaitan dengan penelitian dan berisi tentang penjelasan penelitian sebelumnya dan landasan teori yang akan digunakan.

2.1 Penelitian Terdahulu

Penelitian terdahulu atau yang sering disebut sebagai "studi literatur" sangat penting dalam melakukan sebuah penelitian. Studi literatur adalah proses mengumpulkan, mengevaluasi, merangkum dan menganalisis literatur penelitian atau karya-karya terdahulu yang relevan dengan topik penelitian yang peneliti lakukan. Tujuan dari studi literatur adalah untuk memahami dan mensintesis pengetahuan yang sudah ada tentang topik tersebut, mengidentifikasi kesenjangan dalam literatur, dan menyusun dasar teoretis dan metodologis untuk penelitian yang sedang dilakukan. Berikut beberapa alasan mengapa penelitian terdahulu penting dalam penelitian: memahami landasan teori, identifikasi kesenjangan penelitian, mendukung metodologi penelitian, menghindari duplikasi penelitian, memperkuat argumen dan merumuskan hipotesis Penelitian terdahulu merupakan sebuah referensi dalam melakukan sebuah penelitian untuk mendapatkan teori maupun ringkasan yang akan digunakan sebagai acuan dalam penelitian, penelitian terdahulu digunakan penulis untuk memperkuat landasan teori yang digunakan penulis. Jadi, penelitian terdahulu memainkan peran penting dalam memperkuat dan memperdalam pemahaman peneliti tentang topik penelitian, serta memberikan dasar yang kuat untuk penelitian yang dilakukan. Ini juga membantu menjaga kualitas dan relevansi penelitian yang dihasilkan. Studi literatur merupakan langkah kunci dalam proses penelitian yang membantu membangun dasar yang kuat untuk penelitian, mengidentifikasi relevansi penelitian dalam konteks yang lebih luas, dan menghindari duplikasi penelitian yang sudah ada.

Pada penelitian yang dilakukan oleh Dianati Duei Putri, Gigih Forda Nama, dan Wahyu Eko Sulistiono (2022) dengan judul "Analisis Sentiment Kinerja Dewan Perwakilan Rakyat (DPR) pada *Twitter* Menggunakan Metode *Naïve bayes Classifier*". Penelitian tersebut bertujuan untuk membuat sistem analisis sentiment dari *tweet* pada *twitter* mengenai kinerja DPR, hasil dari penelitian ini adalah metode *naïve bayes* dinilai dapat menyelesaikan analisis sentiment terhadap kinerja DPR pada *twitter* dengan akurasi sebesar 80%.

Penelitian yang dilakukan oleh Muhammad Fadli Asshiddiqi dan Kemas Muslim Lhaksmana (2020) dengan judul "Perbandingan Metode Decision Tree dan Support Vector Machine untuk Analisis Sentimen pada Instagram Mengenai Kinerja PSSI". Penelitian tersebut bertujuan untuk membuat sistem analisis sentimen dengan membandingkan metode *decision tree dan support vector machine* serta metode TF-IDF, hasil dari penelitian ini adalah metode *decision tree* tanpa TF-IDF 87,12% *dan support vector machine* 87,45% sedangkan dengan penambahan metode TF-IDF akurasinya meningkan untuk *decision tree* meningkat sebesar 89,69% dan *support vector machine* sebesar 94,36%, TF-IDF dapat mempengaruhi hasil akurasi.

Penelitian yang dilakukan oleh Fajar Sidik, Ibnu Suhada, Azhar Haikal Anwar dan Firman Noor Hasan (2022) dengan judul "Analisis Sentimen Terhadap Pembelajaran Daring dengan Algoritma *Naïve bayes* Classifier". Penelitian tersebut bertujuan untuk melakukan sentiment analisis pada pendapat orang tua terhadap pembelajaran daring yang diterapkan terhadap anaknya, hasilnya dari 200 dataset yang digunakan hanya mendapatkan akurasi sebesar 65%, penentuan kategori kelas positif dan positif memiliki dampak yang signifikan pada akurasi yang didapatkan.

Penelitian yang dilakukan oleh Melati Indah Petiwi, Agung Triayudi dan Ira Diana Sholihati (2020) dengan judul "Analisis Sentimen Gofood Berdasarkan Twitter Menggunakan Metode *Naïve bayes* dan *Support Vector Machine*". Penelitian ini bertujuan untuk membuat sistem guna menganalisis kepuasan masyarakat terhadap kinerja Gofood di Indonesia, hasil dari penelitian ini adalah akurasi yang didapatkan dengan SVM lebih besar dengan presentase sebesar 83% dari *naïve bayes* yang hanya sebesar 74,6%.

Pada penelitian yang dilakukan oleh Abdul Rozaq, Yessi Yunitasari, Kelik Sussolaikah, Eka Resty Novieta Sari dan Restyono Ilham Syahputra (2022) dengan judul "Analisis Sentimen Terhadap Implementasi Program Merdeka Belajar Kampus Merdeka Menggunakan *Naïve bayes, K-Nearest Neighboars* dan *Decision Tree*". Penelitian ini bertujuan untuk melakukan mengetahui analisis sentimen terhadap implementasi program merdeka belajar kampus merdeka dengan melakukan perbandingan 3 metode, hasil dari penelitian ini adalah didapatkan akurasi sebesar 99,22% untuk metode *naïve bayes, K-nearest neighboars* sebesar 96% dan *decision tree* sebesar 37,21%. Metode *naïve bayes* mampu melakukan klasifikasi dengan baik pada penelitian ini.

2.2 Jurnal Penelitian

Tabel 2.1 Literatur review

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode	Dianati Duei Putri ,Gigih Forda Nama , Wahyu Eko Sulistio. Jurnal Informatika dan Teknik Elektro Terapan	Untuk mengetahui opini masyarakat perihal kinerja DPR pada twitter	hasil klasifikasi mengenai DPR sebanyak 95 positif, 693 netral dan 758 negatif dari data *crawling* sebanyak 1546. Dengan algoritma	Visualisasi hasil analisis sebaiknya menggunakan fitur filter pada bagian tabel berdasarkan	Perbedaan antara hasil penelitian sebelumnya dengan penelitian yang akan dilakukan adalah tentang

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Naïve bayes Classifier (Duei Putri et al., 2022)	Vol. 10 No. 1, Januari 2022		*Naïve bayes* didapatkan accuracy score sebesar 80% jadi sistem mampu memprediksi 80% secara akurat dari total data *testing* sebesar 20%.	tanggal *crawling* sehingga memper mudah untuk melihat sentimen t analysis.	objek yang di teliti serta kelas yang digunakan
2	Perbandingan Metode *Decision Tree* dan *Support Vector Machine*	Muhammad Fadli Asshiddiqi, Kemas Muslim Lhaksmana. e-Proceeding of	Mengetahui hasil pembandingan Algoritma *Decision Tree* dan *Support Vector*	Berdasarkan hasil pengujian dengan *Decision Tree* mendapatkan hasil akurasi	mencari cara untuk mendeteksi komentar instagram yang	Menggunakan perbandingan 2 metode yaitu *Decision Tree* dan SVM

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	untuk Analisis Sentimen pada Instagram Mengenai Kinerja PSSI (Muhammad Fadli Asshiddiqi, 2020)	Engineering : Vol.7, No.3 Desember 2020	*Machine* pada setiap *Term* weighting.	87,45% sedangkan metode SVM mendapatkan akurasi 94,36%. Jadi SVM lebih baik pada penelitian ini	mengandung isi jual produk dan spam, agar memudahkan pada pemilihan data dan *Preprocessing* serta mengenai penggunaan data yang seimbang untuk setiap kelas.	serta pencarian data berasal dari instagram sedangkan penelitian selanjutnya menggunakan *Naïve bayes* dan data berasal dari komentar responden

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Analisis Sentimen Terhadap Pembelajaran Daring dengan Algoritma *Naïve bayes Classifier* (Sidik et al., 2022)	Fajar Sidik, Ibnu Suhada, Azhar Haikal Anwar, Firman Noor Hasan. Jurnal Linguistik Komputasional (2022), Vol. 5 No. 1 Maret 2022	untuk melakukan prediksi pendapat orangtua terhadap pembelajaran daring serta mengetahui nilai akurasi dari pendapat tersebut dengan algoritma *Naïve bayes Classifier*	Dapat disimpulkan bahwa algoritma *Naïve bayes Classifier* dapat mempredikksi serta melakukan perbandingan terhadap pendapat orangtua terhadap pembelajaran daring dari rumah dengan hasil yang cukup baik	Nilai akurasi yang didapatkan hanya 65% dari 200 dataset positif dan negatif, dikarenaksn berbantung pada kualitas penentuan kategori berita positif dan negatifnya dari dataset yan	Perbedaan antara hasil penelitian sebelumnya dengan penelitian yang akan dilakukan adalah tentang objek yang di teliti

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
					dikumpulkan	
4	Analisis Sentimen Gofood Berdasarkan Twitter Menggunakan Metode *Naïve bayes* dan *Support Vector Machine* (Petiwi et al., 2022)	Melati Indah Petiwi, Agung Triayudi, Ira Diana Sholihati. Jurnal Media Informatika Budidarma Vol. 6, No. 1, Januari 2022, Page 542-550	Menganalisa opini masyarakat terhadap kinerja Gojek (Gofood) pengelompokkannya menjadi kelas positif, negatif dan netral. Mengguna kan metode *Naïve bayes* dan SVM dan membandingkan dua	Hasil klasifikasi positif 5,2 %, negatif 2% dan netral 92,8%. Akurasi metode SVM adalah 83% dari 5000 tweet menggunakan bahasa *Python* sedangkan *Naïve bayes* 74,6%, ini membuktikan SVM lebih akurat sebagai	Terlalu banyak data netral disbanding data positif dan negatifnya sehingga objektivitasnya terasa kurang meyakinkan, disarabkan untuk mencari data yang condong	Penelitian sebelumnya menggunakan 2 metode yaitu SVM dan *Naïve bayes* sedangkan penelitian yang akan dilakukan hanya menggunakan metode *Naïve*

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			metode tersebut	metode pengelompokkan disbanding *Naïve bayes*	kea rah negatif ataupun positif	*bayes* saja
5	Analisis Sentimen Terhadap Implementasi Program Merdeka Belajar Kampus Merdeka Menggunakan *Naïve bayes, K-Nearest Neighbors* Dan *Decision Tree*	Abdul Rozaq, Yessi Yunitasari, Kelik Sussolaikah, Eka Resty Novieta Sari, Restyono Ilham Syahputra. Jurnal Media Informatika Budidarma	Menganalisa opini mahasiswa terhadap program merdeka belajar, pengelompokkannya menjadi kelas positif, negatif dan netral. Mengguna kan metode *Naïve bayes*, K-Nearest	Hasil akurasi sebesar 99,22% untuk *Naïve bayes, K-Nearest Neighbors* sebesar 96,90% dan Decision 37,21%, sehingga metode *Naïve bayes* dan K Nearest Neighbors untuk analisis	Penggunaan data yang terlalu sedikit sehingga mengurangi tingkat akurasi metode	Pada penelitian ini digunakan 3 metode sedangkan untuk penelitian yang akan dilakukan hanya memakai 1 metode

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	(Rozaq et al., 2022)	Vol. 6, No. 2, April 2022, Page 746-750	Neighboars Dan *Decision Tree*	sentimen mampu mengklasifikasikan komentar dengan baik dari pada metode *Decision Tree*		

2.3 Analisis Sentimen

Analisis sentimen atau ekstraksi opini merujuk pada bidang yang meliputi berbagai aspek pemrosesan bahasa alami, komputasi linguistik, dan eksplorasi teks. Tujuan utamanya adalah untuk mengurai opini, sentimen, penilaian, sikap, evaluasi, dan emosi seseorang. Tugas pokok dari analisis sentimen adalah mengategorikan polaritas teks dalam dokumen, kalimat, atau mengidentifikasi apakah aspek-aspek pendapat yang terdapat dalam dokumen, kalimat, atau entitas tersebut bersifat positif atau negatif (Syah & Witanti, 2022). Berdasarkan tujuan dari analisis sentimen tersebut sangat memungkinkan dalam mengatasi permasalahan yang muncul dari opini publik terhadap hal yang sedang diteliti, baik opini positif maupun opini yang berlawanan.

Analisis sentimen akan diklasifikasikan berdasar sentimen positif dan negatif.

1. Sentimen Positif

Sentimen positif adalah respons atau sikap dari individu yang mengindikasikan persetujuan, kesepakatan, dan memiliki potensi untuk meningkatkan nilai seseorang atau sesuatu.

2. Sentimen Negatif

   Sentimen negatif menggambarkan respons atau sikap penolakan dari seseorang yang berpotensi merendahkan nilai seseorang atau sesuatu, serta dapat mengakibatkan penurunan dan menciptakan tren negatif. Biasanya, sentimen negatif ditandai dengan penggunaan kata-kata yang bersifat negatif.

2.4 *Python* 3

Python adalah sebuah bahasa pemrograman yang dirancang oleh Guido Van Rossum dan pertama kali diperkenalkan pada tahun 1991. Bahasa pemrograman ini umumnya digunakan dalam berbagai konteks seperti pembuatan aplikasi web, pengembangan perangkat lunak, eksplorasi data, dan penerapan pembelajaran mesin. Keunggulan *Python* terletak pada kemampuannya yang mudah dipelajari, efisien, serta kompatibilitasnya dengan berbagai platform. Keistimewaan lainnya adalah perangkat lunak *Python* dapat diunduh secara gratis dan memiliki kemampuan integrasi yang baik dengan berbagai jenis sistem (Azhar et al., 2022).

Pada tahun 2008, *Python* 3.0 dirilis sebagai revisi besar dalam bahasa pemrograman ini. Versi ini tidak sepenuhnya kompatibel dengan versi sebelumnya, sehingga banyak kode yang ditulis dalam *Python* 2 tidak dapat berjalan tanpa modifikasi di lingkungan *Python* 3.

Gambar 2.1 Logo *python*

2.5 *Jupyter Notebook*

Jupyter Notebook adalah sebuah aplikasi web gratis yang sering digunakan oleh para ilmuwan data. Aplikasi ini memungkinkan pembuatan dan berbagi dokumen yang menggabungkan perhitungan, visualisasi, dan teks. Di dalam

Jupyter Notebook, terdapat tiga bahasa pemrograman utama yaitu *Julia*, *Python*, dan R, yang memiliki peranan penting dalam kegiatan ilmuwan data (Sholeh et al., 2022). Secara sederhana, tujuan utama dari *Jupyter Notebook* adalah mendukung ilmuwan data dalam merangkai narasi komputasi. Narasi komputasi tersebut berfungsi untuk mengurai makna dari data dan memberikan wawasan tentang informasi yang terkandung di dalamnya. Dalam hal ini, *Jupyter Notebook* berperan sebagai alat bantu yang sangat berguna (Asyrofi & Asyrofi, 2023).

Dalam konteks aplikasi *Jupyter Notebook* dengan menggunakan *Python*, alat ini juga sangat cocok untuk analisis sentimen. Penggunaan *Jupyter Notebook* dengan bahasa pemrograman *Python* terbukti sederhana dan mudah dipahami. Salah satu kelebihan kunci dari *Jupyter Notebook Python* adalah kemampuannya untuk menampilkan langkah-langkah program yang berhasil dieksekusi atau menunjukkan kesalahan (error) pada bagian tertentu dari program (Pebralia, 2022).

2.6 *Library Numpy*

Numpy merupakan sebuah pustaka yang populer digunakan oleh para pengembang untuk mempermudah pembuatan dan pengelolaan rangkaian data, melakukan manipulasi bentuk logis, serta melaksanakan operasi aljabar linear. Pustaka *Numpy* memiliki peran yang sangat krusial dan menjadi dasar bagi banyak pustaka dalam pengolahan data dan pembelajaran mesin seperti *Pandas*, *scipy*, *scikit-learn*, dan lain sebagainya (Hasri & Alita, 2022).

Ada beberapa alasan kuat mengapa penggunaan *Numpy* menjadi pilihan yang lebih unggul dibandingkan dengan menggunakan struktur data list standar yang ada dalam *Python*. Beberapa kelebihan *Numpy* antara lain :

- Penggunaan momori yang lebih efisien
- Fleksibilitas yang tinggi untuk menangani Multidimensional Objek
- Kemampuan komputasi numerik yang lebih cepat
- Broadcasting fungsi dan operasi
- Banyak *library machine learning* yang dibangun berdasarkan *Numpy*

Install *library Numpy* menggunakan *Python package* management seperti conda dan pip.

Segmen 2.1 Instalasi dan *Import numpy*

```
1   Import Numpy as np
```

2.7 *Library Natural Language Tool Kit*

Natural Language Toolkit (NLTK) adalah sebuah pustaka atau *library* yang digunakan untuk mendukung berbagai tugas yang terkait dengan pemrosesan bahasa alami (NLP) dalam pemrograman (Rifano et al., 2020). NLTK membantu dalam melakukan berbagai tugas pemrosesan teks, seperti klasifikasi, tokenisasi (memecah teks menjadi bagian-bagian yang lebih kecil), *stemming* (mengubah kata-kata menjadi bentuk dasarnya), penandaan kata (*part-of-speech tagging*), analisis sintaksis, dan penalaran semantik (Jimly Hanif et al., 2023).

Sebelum dapat menggunakan pustaka NLTK dalam proyek atau skrip *Python*, langkah pertama yang harus dilakukan adalah menginstalnya di lingkungan kerja. Setelah NLTK terinstal, dapat mengimpor dan menggunakan berbagai fitur dan fungsi yang disediakan oleh pustaka ini dalam proyek-proyek NLP (Purnama et al., n.d.). NLTK menyediakan berbagai alat dan sumber daya yang dapat sangat berguna dalam melakukan analisis teks dan pemrosesan bahasa alami. Dengan menggunakan NLTK, kita dapat mengatasi banyak tugas pemrosesan bahasa alami dengan lebih efisien dan efektif, karena pustaka ini menawarkan berbagai fungsi dan algoritma yang telah dikembangkan untuk mendukung pengolahan teks dalam berbagai konteks dan bahasa (Yolanda Talahaturuson et al., 2022). Cara installnya adalah sebagai berikut :

Segmen 2.2 Instalasi *natural language tool kit*

```
1   pip install nltk
```

2.8 *Library Sastrawi*

Salah satu kelemahan yang dapat diidentifikasi pada NLTK adalah kurangnya dukungan yang memadai untuk bahasa Indonesia. Karena alasan ini, kami akan mengadopsi penggunaan pustaka tambahan bernama *Sastrawi* (Jimly Hanif et al., 2023). *Sastrawi* adalah pustaka pengolahan bahasa alami (NLP) yang dirancang khusus untuk bahasa Indonesia. Awalnya, pustaka ini dikembangkan dan disusun untuk digunakan dalam bahasa pemrograman PHP. Tetapi, karena tingginya

popularitas, *Sastrawi* kemudian diadaptasi untuk mendukung penggunaan dalam bahasa pemrograman *Python* juga. Untuk melakukan instalasi, hanya perlu menjalankan skrip berikut :

Segmen 2.3 Install *library sastrawi*

```
1   Pip install Sastrawi
```

2.9 *Library Pandas*

Pandas adalah sebuah pustaka sumber terbuka (*open source*) dalam bahasa pemrograman *Python* yang sering digunakan untuk melakukan pengolahan data, termasuk membersihkan data, memanipulasi data, dan melakukan analisis data. Saat melakukan analisis, langkah pertama adalah mempersiapkan data mentah sebelum bisa digunakan. Data mentah harus melalui proses pra-analisis yang dikenal sebagai data *Wrangling*. *Wrangling* adalah proses di mana data dikelola dan diatur sedemikian rupa sehingga menjadi lebih terstruktur dan siap untuk dianalisis (Albanna et al., 2022). Tahap ini sangat penting karena memerlukan kecermatan dan harus mampu menjawab permasalahan yang ingin dipecahkan.

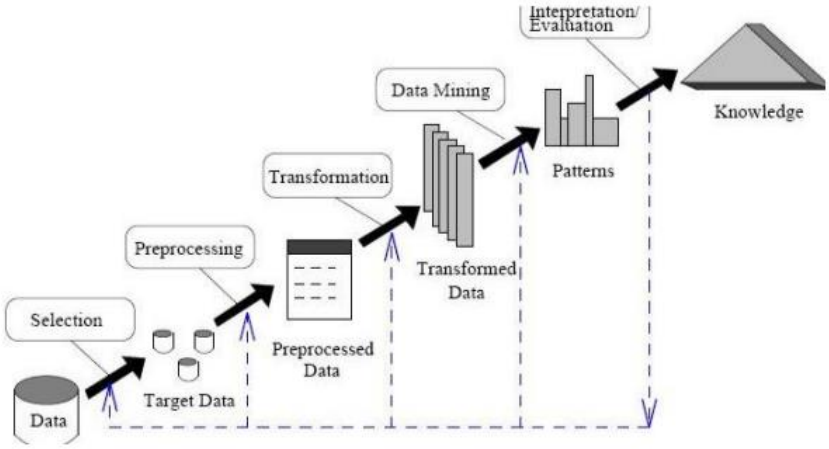
Pustaka *Pandas* mampu mengolah berbagai jenis format data, termasuk csv, txt, *excel, html*, dan lain sebagainya. Dalam penggunaannya, *Pandas* biasanya diberi alias sebagai "pd". Berikut adalah skrip instalasi untuk *Pandas*:

Segmen 2.4 Instalasi dan *Import pandas*

```
1   Import Pandas as pd
```

2.10 Data *Mining*

Data *mining* adalah sebuah kegiatan yang bertujuan untuk menemukan pola-pola unik dari kumpulan data yang berskala besar. Data yang digunakan bisa disimpan dalam berbagai tempat penyimpanan informasi seperti *Database*, gudang data, dan lain sebagainya (Setiyani et al., 2020). Dalam praktiknya, data mining adalah komponen yang penting dalam proses *Knowledge Discovery in Database* (KDD), dimana tujuannya adalah mengekstraksi pola atau model dari data menggunakan algoritma-algoritma tertentu (Prasetyo et al., 2020).



Gambar 2.2 Data *mining*
Sumber : Setiyani et al., 2020

Berikut adalah proses *Knowledge Discovery Database* :

1. Data selection : pemilihan data dari data mentah
2. *Preprocessing* : proses pembersihan sampai pengembalian teks data pada kata awal.
3. *Transformation* : proses perubahan pada data yang terseleksi.
4. Data *mining* : proses mencari pola pada data dengan teknik atau metode tertentu.
5. *Interpretation / Evaluation* : validasi pemeriksaan pola atau informasi yang dihasilkan bertentangan dengan fakta atau hipotesa yang sebelumnya atau tidak.
6. *Knowledge* : proses visualisasi hasil dari pengolahan data.

2.11 *Text Mining*

Menurut *Text Mining* adalah proses intensif dalam penggalian informasi yang menggunakan alat dan metode khusus untuk menganalisis kumpulan dokumen. Ini merupakan bagian dari penambangan data yang fokus pada data teks. *Text Mining* digunakan untuk menggambarkan teknologi yang mampu menganalisis data teks yang bersifat semi terstruktur atau tidak terstruktur (Tambunan & Hapsari, 2021). Meskipun *Text Mining* memiliki tujuan dan proses yang serupa dengan data mining, namun input yang digunakan berbeda. Data yang digunakan dalam *Text Mining* adalah data yang cenderung tidak terstruktur atau kurang terstruktur, seperti

dokumen Word, PDF, kutipan teks, dan sejenisnya, sedangkan data mining biasanya beroperasi pada data yang terstruktur.

Pendekatan yang digunakan dalam memahami struktur data teks adalah dengan menentukan fitur-fitur yang mewakili setiap kata atau elemen dalam dokumen tersebut. Sebelum menentukan fitur-fitur ini, langkah *Pre-processing* biasanya diterapkan pada dokumen (Zhafira et al., 2021). Tahap *Pre-processing* ini melibatkan beberapa langkah seperti pembersihan (cleansing) data, tokenisasi (memecah teks menjadi kata-kata atau bagian), penyaringan (*filtering/remove stopword*), dan *stemming* (penghapusan imbuhan dalam kata). Semua langkah ini bertujuan untuk menghasilkan data yang lebih terstruktur dan cocok untuk analisis lebih lanjut.

Proses *Pre-processing* adalah tahapan penting yang harus dilalui sebelum melangkah ke tahap analisis lebih lanjut dalam memeriksa topik yang sedang dijalankan. *Pre-processing* data dilakukan untuk mempersiapkan teks atau data mentah menjadi format yang seragam dan terstruktur, sehingga siap untuk diolah pada tahap analisis selanjutnya (Fauziyyah, 2020). Tahap *Pre-processing* melibatkan serangkaian langkah yang bertujuan untuk membersihkan, merapihkan, dan mengorganisir data teks agar lebih sesuai untuk analisis.

Tujuan dari tahap *Pre-processing* dalam konteks ini adalah untuk menghilangkan kata-kata yang tidak diperlukan atau yang tidak memiliki arti dari data teks, terutama saat mengolah komentar dari masyarakat (Hardi et al., 2021). Tahap *Pre-processing* dilakukan untuk membersihkan dan merapikan teks, sehingga kata-kata yang tidak memiliki kontribusi signifikan terhadap analisis dapat dihilangkan. Hal ini membantu meningkatkan akurasi dan relevansi analisis yang akan dilakukan pada data tersebut. Dengan menghapus kata-kata yang tidak memiliki arti atau yang umumnya tidak memberikan informasi penting (seperti *stopwords*), analisis dapat lebih fokus pada kata-kata yang memang memiliki makna dan kontribusi penting dalam mengungkapkan opini atau informasi dalam komentar. Ini dapat menghasilkan hasil yang lebih bersih, efisien, dan akurat dalam analisis yang dilakukan. Gambaran tahapan *preprocessing* sebagaimana terlihat pada gambar 2.3 berikut :

Gambar 2.3 Alur *text mining* pada *preprocessing*

Adapun urutan beserta pejelasan dari tahapan *Preprocessing* yang akan dilakukan adalah sebagai berikut:

a. *Cleansing*

Pada tahap ini memang kritis dalam mempersiapkan data teks untuk analisis lebih lanjut. Berikut adalah penjelasan berdasarkan langkah-langkah disebutkan:

1. Pengambilan Data: Tahap ini melibatkan pengumpulan data dari sumber-sumber yang relevan, seperti komentar masyarakat.
2. *Cleansing* (Pembersihan): Dalam tahap ini, data teks diperiksa untuk menghilangkan "*noise*" atau gangguan, seperti karakter khusus, tanda baca, atau simbol yang tidak relevan. Ini membantu mencegah duplikasi data dan menghasilkan data yang lebih bersih.
3. Menghilangkan *Noise*: *Noise* atau gangguan dalam teks seperti karakter tak penting atau simbol dapat mengganggu analisis. Menghilangkan *noise* membantu menjaga kesatuan dan akurasi data.
4. Penghilangan Angka dan Simbol: Angka romawi, desimal, atau simbol yang tidak penting juga dihilangkan karena tidak memberikan kontribusi signifikan dalam analisis sentimen.
5. Mengubah ke Huruf Kecil: Merubah seluruh teks menjadi huruf kecil adalah langkah penting untuk menghindari perbedaan dalam analisis karena perbedaan kapitalisasi. Misalnya, "babi" dan "Babi" akan diperlakukan sama setelah diubah ke huruf kecil.

Semua langkah ini bertujuan untuk membersihkan data, menjaga kualitas data, dan menghilangkan variabel yang tidak relevan sehingga analisis dapat dilakukan dengan lebih fokus dan akurat pada aspek sentimen yang sebenarnya.

b. *Remove Stopword*

Pada Penghapusan *stopwords* adalah langkah yang umum dalam *preprocessing* teks yang bertujuan untuk menghilangkan kata-kata yang umum dan tidak memiliki signifikansi dalam analisis atau klasifikasi teks lebih lanjut. Berikut adalah contoh penghapusan *stopwords* dalam kalimat:

Kalimat awal: "Pada tahap ini, kita melakukan proses *filtering* untuk menghilangkan kata-kata yang tidak memiliki hubungan dengan pengolah kata yang akan dipelajari dan tidak berpengaruh pada proses klasifikasi." Kalimat setelah penghapusan *stopwords*: "tahap melakukan proses *filtering* menghilangkan kata-kata hubungan pengolah dipelajari berpengaruh klasifikasi."

Dalam contoh di atas, kata-kata seperti "pada", "ini", "kita", "untuk", "yang", "dan", "tidak", "dengan", "akan", "dan", "pada" adalah *stopwords* yang dihapus karena tidak memiliki pengaruh signifikan terhadap analisis dan klasifikasi. Hal ini membantu memfokuskan analisis pada kata-kata yang lebih penting dalam konteks yang sedang dipelajari.

c. *Tokenizing*

Tokenisasi adalah langkah penting dalam *preprocessing* teks. Dalam tahap tokenisasi, kalimat atau teks akan dipecah menjadi bagian-bagian yang lebih kecil, yang disebut sebagai token. Setiap token dapat berupa kata, frasa, atau elemen lain yang memiliki makna dalam konteks bahasa.

Contoh: Kalimat: "saya sedang mandi susu."

Setelah tokenisasi:

- Token 1: ["saya"]
- Token 2: ["sedang"]
- Token 3: ["mandi"]
- Token 4: ["susu"]

Dalam contoh di atas, setiap kata dalam kalimat telah dipecah menjadi token-token yang terpisah. Tokenisasi membantu mengubah teks menjadi bentuk yang lebih mudah untuk dianalisis, diolah, atau dimengerti oleh komputer dalam konteks pengolahan bahasa alami.

d. *Stemming*

Stemming adalah proses di mana kata-kata dalam teks diubah atau dikembalikan ke bentuk kata dasarnya dengan menghapus imbuhan atau akhiran, sehingga memungkinkan kata-kata yang memiliki akar yang sama diperlakukan sebagai satu kata(Muhammad Fadli Asshiddiqi, 2020). Ini membantu

mengurangi variasi kata dan menghasilkan data yang lebih terstruktur dalam analisis.

Stemming adalah salah satu tahap penting dalam *preprocessing* teks, terutama ketika menggunakan bahasa Indonesia. Saat menganalisis kata-kata dalam bahasa Indonesia, kata-kata tersebut perlu melalui tahap *stemming* agar diubah menjadi bentuk dasarnya yang dapat ditemukan dalam kamus. *Python* memiliki pustaka yang disediakan khusus untuk *stemming* dalam bahasa Indonesia, yang dapat digunakan untuk melakukan tahap *stemming* ini. Dengan menggunakan algoritma *stemming* yang sesuai, dapat memastikan konsistensi dan keterbacaan dalam data yang telah diolah.

2.12 *Machine Learning*

Seiring Seiring berlalunya waktu, mesin pintar semakin menggantikan dan meningkatkan kemampuan manusia di berbagai bidang. Kemampuan yang ditunjukkan oleh mesin ini dikenal sebagai kecerdasan buatan (*Artificial Intelligence* atau AI), yang merupakan subbidang dalam ilmu komputer. Kecerdasan buatan bertujuan untuk menciptakan perangkat lunak dan perangkat keras yang mampu berpikir seperti manusia (Retnoningsih & Pramudita, 2020).

Pembelajaran mesin (*Machine Learning*) adalah cabang dari kecerdasan buatan. Ini adalah pendekatan di mana komputer menggunakan algoritma matematika dan data untuk belajar dari pengalaman dan membuat prediksi di masa depan. Proses pembelajaran melibatkan dua tahap utama, yaitu pelatihan dan pengujian. Pembelajaran mesin melibatkan pertanyaan tentang cara membangun program komputer yang dapat meningkatkan dirinya sendiri berdasarkan pengalaman.

Pembelajaran mesin dapat dibagi menjadi tiga kategori utama:

1. Pembelajaran Terawasi (*Supervised Learning*): Di mana algoritma diberi data berlabel dan dipelajari untuk memahami hubungan antara input dan output.

2. Pembelajaran Tanpa Pengawasan (*Unsupervised Learning*): Di sini, algoritma mencari pola atau struktur dalam data yang tidak berlabel tanpa panduan eksternal.
3. Pembelajaran Penguatan (*Reinforcement Learning*): Algoritma belajar melalui interaksi dengan lingkungan dan diberi umpan balik dalam bentuk hadiah atau hukuman.

Semua ini merupakan bagian penting dari perkembangan teknologi yang terus bergerak maju dalam menghadapi tantangan dan peluang yang diciptakan oleh kecerdasan buatan dan pembelajaran mesin.

2.13 Term Frecuency – Inverse Document Frequency

Metode TF-IDF adalah salah satu metode yang paling umum digunakan dalam pengolahan teks untuk menghitung bobot setiap kata dalam suatu dokumen. Metode ini dianggap efektif, mudah diterapkan, dan menghasilkan hasil yang akurat dalam analisis teks (Mayasari & Indarti, 2022). Metode TF-IDF menggabungkan dua konsep perhitungan bobot:

1. *Term Frequency* (TF): Ini mengukur seberapa sering kata tertentu muncul dalam sebuah dokumen. Jika kata tersebut muncul lebih sering, maka bobotnya akan lebih tinggi dalam dokumen tersebut.
2. *Inverse Document Frequency* (IDF): Ini mengukur seberapa umum kata tersebut di seluruh dokumen koleksi. Jika kata tersebut jarang muncul di semua dokumen, bobotnya akan lebih besar.

Dengan menggabungkan TF dan IDF, kita bisa mendapatkan bobot kata yang menggambarkan seberapa penting kata tersebut dalam konteks dokumen dan seluruh koleksi (Vitandy et al., 2021). Kata yang muncul sering dalam dokumen tetapi jarang di koleksi keseluruhan akan memiliki bobot yang tinggi karena dianggap signifikan dalam dokumen tersebut.

Metode TF-IDF membantu dalam mengekstrak kata-kata yang memiliki informasi penting dan relevan dalam sebuah dokumen atau korpus teks(Rahman et al., 2022). Ini adalah alat penting dalam analisis teks untuk mengidentifikasi kata-kata kunci atau fitur-fitur yang mempengaruhi konten dokumen.

Rumusnya adalah sebagai berikut :

$$tf = 0{,}5 + 0{,}5 * \frac{ft, d}{\max(ft, d)} \quad (2.1)$$

$$idf = log\frac{N}{df} \quad (2.2)$$

$$W = tf * idf \quad (2.3)$$

Keterangan :

d : Dokumen

t : kata pada dokumen

ft,d : frekuensikata pada d

tf : banyaknya kata i pada sebuah dokumen

N : total jumlah dokumen

dtf : banyak dokumen yang mengandung kata i

idf : inversed *Document Frequency*

W : bobot dokumen ke-d terhadap kata ke t

Setelah nilai bobot (W) untuk setiap dokumen dihitung menggunakan metode TF-IDF atau metode lainnya, langkah berikutnya adalah melakukan proses pengurutan dokumen. Pengurutan dilakukan berdasarkan nilai bobot yang dihasilkan untuk setiap dokumen.

Dalam analisis pengurutan, dokumen yang memiliki nilai bobot (W) yang lebih tinggi akan dianggap lebih relevan dengan kata kunci atau topik yang dicari. Dengan demikian, semakin besar nilai bobot (W) suatu dokumen, semakin besar kemungkinan dokumen tersebut memiliki konten atau informasi yang lebih relevan dengan kata kunci atau topik tertentu.

Pengurutan dokumen berdasarkan nilai bobot (W) ini memungkinkan untuk menghasilkan urutan dokumen yang paling relevan terhadap kata kunci atau topik yang dianalisis. Ini membantu pengguna atau peneliti untuk mengidentifikasi dan mengeksplorasi dokumen-dokumen yang paling relevan terlebih dahulu dalam rangka mendapatkan informasi yang paling sesuai dengan tujuan analisis mereka (Atika et al., 2022).

2.14 Naïve bayes Clasiifier

Algoritma *Naïve bayes* adalah metode klasifikasi statistik yang berdasarkan pada Teorema Bayes. *Naïve bayes Classifier* adalah suatu algoritma pengklasifikasi yang menggunakan probabilitas dan statistik yang awalnya diusulkan oleh ilmuwan Inggris bernama Thomas Bayes (Atthahara & Priyanti, 2021). Teorema Bayes digunakan untuk memprediksi peluang masa depan berdasarkan pengalaman sebelumnya. "*Naïve*" dalam *Naïve bayes* mengacu pada asumsi bahwa atribut-atribut dalam dataset dianggap saling bebas, meskipun dalam kenyataannya ini mungkin tidak selalu terjadi.

Dalam klasifikasi *Naïve bayes*, diasumsikan bahwa keberadaan atau ketiadaan suatu karakteristik tertentu dari suatu kelas tidak berkaitan dengan karakteristik lain dari kelas tersebut. Ini membuat algoritma ini relatif sederhana dan cocok untuk digunakan dalam klasifikasi data. Keuntungan utama dari pendekatan *Naïve bayes* adalah kinerja yang baik ketika digunakan dengan dataset yang besar. Dalam klasifikasi *Naïve bayes*, proses pembelajaran lebih berfokus pada perhitungan dan estimasi probabilitas. Algoritma ini cocok untuk digunakan dalam berbagai jenis klasifikasi, seperti klasifikasi teks, klasifikasi spam, dan banyak lagi. Meskipun memiliki asumsi "*Naïve*" yang mungkin tidak selalu sesuai dengan kondisi nyata, *Naïve bayes* sering memberikan hasil yang memuaskan dalam banyak kasus, terutama ketika memiliki jumlah data yang besar (Wijaya, 2023).

Namun, perlu diingat bahwa algoritma *Naïve bayes* tidak selalu cocok untuk semua jenis data dan kasus. Terkadang, asumsi kemandirian atribut dapat mengarah pada hasil yang kurang akurat dalam situasi di mana ketergantungan antara atribut penting. Tetapi secara umum, *Naïve bayes* merupakan algoritma yang cukup kuat dan efisien dalam banyak konteks klasifikasi.

Persamaan dari Teorema Bayes adalah sebagai berikut :

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

Keterangan :

X : Data dengan class yang belum diketahui

H : Hipotesis data X merupakan suatu class spesifik

$P(H|X)$:Probabilitas hipotesis H berdasar kondisi (*posteriori probability*)

$P(H)$:Probabilitas hipotesis H (*prior probability*)

$P(X|H)$:Probabilitas X berdasarkan kondisi pada hipotesis

$P(X)$:Probabilitas X

Kaitan Keterkaitan antara *Naïve bayes* dengan klasifikasi, korelasi hipotesis, dan bukti dengan klasifikasi adalah bahwa hipotesis dalam teorema Bayes adalah label kelas yang menjadi target pemetaan dalam klasifikasi, sedangkan bukti adalah fitur-fitur yang menjadi input dalam model klasifikasi. . Jika X adalah vektor masukan yang berisi fitur dan Y adalah label kelas. *Naïve bayes* ditulis sebagai P(Y|X). Notasi ini berarti bahwa probabilitas label kelas Y diperoleh setelah fitur X diamati. Notasi ini juga disebut probabilitas posterior untuk Y, sedangkan P(Y) disebut probabilitas sebelumnya untuk Y. Selama proses pelatihan, perlu mempelajari probabilitas akhir (P(Y|X) dalam model untuk setiap kombinasi dari X dan Y). berdasarkan informasi yang diperoleh dari data pelatihan. Dengan membangun model, data uji X' dapat diklasifikasikan dengan mencari nilai Y' dengan memaksimalkan nilai P(Y'|X') yang diperoleh.

Untuk menjelaskan teorema *Naïve bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, persamaan di atas dapat disesuaikan seperti berikut :

$$P(C|F1 \dots Fn)\frac{P(C)P(F1 \dots Fn|C)}{P(F1 \dots Fn)} \qquad (2.5)$$

Di mana Variabel C merepresentasikan kelas, sementara variabel F1 . Fn merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (*Posterior*) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik sampel pada kelas C (disebut juga *Likelihood*), dibagi dengan peluang kemunculan karakteristik karakteristik sampel secara global

(disebut juga *evidence*). Karena itu, rumus diatas dapat pula ditulis secara sederhana sebagai berikut :

$$Posterior = \frac{Prior * Likelihood}{evidence} \quad (2.6)$$

Metode *Naïve bayes* menempuh dua tahap dalam proses klasifikasi teks, yaitu tahap pelatihan dan tahap klasifikasi. Pada tahap pelatihan dilakukan proses terhadap sampel data yang sedapat mungkin dapat menjadi representasi data. Selanjutnya adalah penetuan probabilitas *prior* berdasarkan sampel data. Pada tahap klasifikasi ditentukan kategori dari suatu data berdasarkan *Term* yang muncul dalam data yang diklasifikasi. Teorema *Naïve bayes* dapat dinyatakan dalam persamaan

$$P(X_k|Y) = \frac{P(Y|X_k)}{\Sigma i P(Y|Xi)} \quad (2.7)$$

Dimana keadaan Posterior (Probabilitas Xk didalam Y) dapat dihitung dari keadaan *Prior* (Probabilitas Y dalam Xk dibagi dengan jumlah dari semua probabilitas Y dalam semua Xi). Untuk menghindari adanya nilai nol pada probabilitas, maka diberlakukan *Laplace Smooting*. Tujuan diberlakukan *Laplace Smooting* adalah untuk mengurangi probabilitas dari hasil atau keluaran yang terobservasi dan juga sekaligus meningkatkan dan menambah probabilitas hasil yang belum terobservasi sehingga persamaan menjadi sebagai berikut :

$$P(V1|C = c) = \frac{CountTerms\big(v1, docs(c)\big) + 1}{AllTerms\big(docs(c)\big) + v} \quad (2.8)$$

Dimana |V| menunjuk pada jumlah semua kata dalam data komentar yang ada di dataset. Untuk dapat mengklasifikasikan suatu komentar, dapat dilakukan dapat dilakukan dengan persamaan :

$$P(V1|C = c) = \frac{CountTerms\big(v1, docs(c)\big)}{AllTerms\big(docs(c)\big)} \quad (2.9)$$

Dimana v1 dalam persamaan diatas adalah satu kata tertentu dalam data, sedangkan *CountTermsv1*. docsv(c) menunjukan pada jumlah kemunculan suatu kata berlabel c (“positif”, “netral” atau “negatif”). *AllTerms1docs*(c) menunjuk pada jumlah semua kata berlabel c yang ada pada dataset.

Naïve bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara conditional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas individu. Keuntungan penggunaan *Naïve bayes* adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*training* data) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Tahapan dari perhitungan *Naïve bayes* adalah :

1. Tahap awal dalam proses *Naïve bayes Classifier* adalah menghitung probabilitas masing-masing kelas dari keseluruhan data *training*.
2. Proses *testing*. Proses ini untuk mengetahui keakuratan model yang dibangun pada proses *training*, umumnya digunakan data yang disebut test set untuk memprediksi label.

Metode *Naïve bayes Classifier* terdiri dari dua tahapan proses klasifikasi teks, yaitu tahap pelatihan dan tahap klasifikasi. Pada tahap pelatihan dilakukan proses analisis terhadap dokumen sampel berupa pemilihan kosakata yang merupakan kata-kata yang mungkin muncul dalam kumpulan dokumen sampel yang merupakan dokumen representatif. Langkah selanjutnya adalah menentukan probabilitas untuk setiap kategori berdasarkan sampel dokumen (Wijaya, 2023).