BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjaun Pustaka

Tinjaun pustaka pertama dengan judul penilitian "Komparasi Metode *Naïve Bayes* dan *K-Nearest Neighbor* Terhadap Analisis Pengguna Aplikasi Pedulilindung". Data yang diambil yaitu tentang sentimen untuk mengetahui opini dari pengguna aplikasi Pedulilindungi. Penelitian menganalisis menggunakan data yang diperoleh dari komentar pengguna di *plystore* dengan data yang di kelompokan dalam kelas positif dan negatif. Dalam hal ini penelitian melakukan kompirasi antara metode *Naïve Bayes* dan *K-Nearest Neighbor* untuk membandingkan akurasi yang dihasilkan agar dapat mengetahui model terbaik dari kedua metode tersebut yang dapat digunakan sebagai analisis sentimen pada aplikasi Peduli Lindugi. Dari metode yang digunakan disimpulkan bahwa metode *K-Nearest Neighbor* memiliki tingkat akurasi lebih baik dibandingkan metode *Naïve Bayes* (Wibowo et al., 2022).

Tinjaun pustaka kedua dengan judul penelitian "Analisis Sentimen Pada Agen Perjalanan Online Menggunakan *Naïve Bayes* dan *K-Nearest Neighbor*". Penelitian menganalisis 3 agen online yang ada di Indonesia dengan pengukuran data dari komentar pengguna Facebook dengan data berlabel positif, negatif dan netral. Dalam penelitian pengujian dilakukan untuk membandingkan metode *K-Nearest Neighbor* dan *Naïve Bayes*, serta menguji data dengan beberapa variasi yaitu huruf kecil, tanpa tada baca dan campuran. Dari hasil yang disimpulkan bahwa metode *K-Nearest Neighbor* memiliki akurasi yang lebih baik dibandingkan *Naïve Bayes*

dan pengujian dua metode menggunakan variasi huruf kecil memiliki akurasi yang sama baik (Sholeha et al., 2022).

Tinjauan pustaka ke tiga dengan judul "Analisis Sentimen Mahasiswa Terhadap Layanan Stmik Primakara Menggunakan Algoritma *Naïve Bayes* dan *K-Nearest Neighbor*". Penelitian dilakukan untuk menganalisis kepuasan mahasiswa terhadap layanan Stimik Primakara dengan meggunakan metode *Naïve Bayes* dan *K-Nearest Neighbor* dengan data berlabel positif, negatif dan netral. Dari hasil pengujian disimpulkan bahwa *K-Nearest Neighbor* memiliki kinerja yang cukup baik dalam melakukan analisa sentimen pada komentar mahasiswa (Sugiarta et al., 2023).

Tinjaun pustaka ke empat dengan judul "Analisis Sentimen Terhadap Twit Maxim Pada *Twitter* Menggunakan R Programming dan *K-Nearest Neighbor*". Dalam penelitian data yang diambil menggunakan Api *Twitter*. Dimana penelitian dilakukan untuk mengklasifikaskan data dengan metode *K-Nearest Neighbor* dengan data yang telah di beri label positif, negatif dan netral (Diwandanu & Wisudawati, 2023).

Berdasarkan tinjaun pustaka diatas terdapat beberapa kemiripan metode yang dipergunakan. Namun memiliki perbedaan pada data yang digunakan dan hasil yang dicari. Pada penelitian pertama kedua dan ketiga penelitian dilakukan untuk membandingkan antara metode Niave Bayes dan *K-Nearest Neighbor*, dan tinjaun pustaka ke 4 mengacu pada analisis sentimen *tweet* maxim dengan data yang diambil menggunakan Api *Twitter*. Sedangkan pada penelitian ini akan melakukan analisis sentimen terhadap program lulus sarjana tidak wajib skripsi menggunakan

metode *K-Nearest Neighbor*. Data diperoleh dari *tweet* tentang opini masyarakat terhadap kebijakan kemendibustrek tentang Penjaminan Mutu Pendidikan Tinggi yang di ambil dari twiiter dengan *tweet harvest*.

Tabel 2.1 Tinjaun pustaka

Penulis	Objek	Metode	Hasil Penilitian
(Wibowo et al., 2022)	Aplikasi Pedulilindungi	*Naïve Bayes* dan *K-Nearest Neighbor*	Penilitian dilakukan untuk menganalisis pengguna pedulilindungi.Hasil akurasi yang didapatkan ialah sebesar 70,46% untuk *Naïve Bayes* dan akurasi 73,33% untuk *K-Nearest Neighbor*, dengan hasil tersebut dinyatakan bahwa metode *K-Nearest Neighbor* memiliki tingkat akurasi yang lebih baik dibandingkan metode *Naïve Bayes*.
(Sholeha et al., 2022)	Agen Perjalanan Online	*Naïve Bayes* dan *K-Nearest Neighbor*	Hasil peneltian adalah Algoritma *K-Nearest Neighbor* memiliki akurasi yang lebih baik pada rata-rata dari pada *Naïve Bayes*.Hasil akurasi yang lebih tinggi didapatkan pada penggunaan data huruf kecil dengan akurasi 52,35% terhadap dua metode.
(Sugiarta et al., 2023)	Layanan Stmik Primakara	Algoritma *Naïve Bayes* dan *K-Nearest Neighbor*	Dari data uji yang digunakan sebanyak 216 data testing yang memperoleh jumlah prediksi sentimen positif sebanyak 67 data, negatif sebanyak 144 data dan netral sebanyak 55 data. Berdasarkan hasil tersebut dinyatakan bahwa layanan pada

			STMIK Primakara pada tahun 2019 – 2022 cenderung sentimen negatif. Sedangkan Hasil yang didapatkan pada pengujian *K-Nearest Neighbor* dan *Naïve Bayes* tersebut dapat dinyatakan bahwa penggunaan algoritma *K-Nearest Neighbor* lebih unggul dibandingkan Naive Bayes dan penggunaan agoritma *K-Nearest Neighbor* memiliki kinerja yang cukup baik dalam melakukan analisa sentimen pada komentar mahasiswa STMIK Primakara.
(Diwandanu & Wisudawati, 2023)	Maxim	*K-Nearest Neighbor*	Hasil peneliti yaitu hasil akurasi terbaik didapatkan dari skema pertama 80% data latih sebanyak 702 data dan 20% data uji sebanyak 175 data dengan k=1 dengan akurasi sebesar 95,43%.
Usulan Peneliti, 2023	Program sarjana tidak wajib skripsi	*K-Nearest Neighbor*	Dalam penelitian dilakukan untuk menganalisis sentimen masyarakat terkait kebijakan skripsi tidak wajib dengan metode *K-Nearest Neighbor*.

2.2 Landasan Teori

2.2.1 Analisis Sentimen

Sentimen analisis atau disebut juga opini mining adalah bidang studi yang bertujuan untuk menganalisis opini, sentimen, penelitian, evaluasi, sikap dan emosi publik terhadap suatu entitas dari produk, pelayanan, suatu permasalahn, organisasi, peristiwa tertentu, topik yang hangat dibicarakan dan atributnya (Lailany et al., 2023, p. 178)

Tugas dasar dalam analisis sentimen adalah mengelompokkan teks yang ada dalam sebuah kalimat atau dokumen kemudian menentukan kalimat atau dokumen tersebut apakah bersifat positif atau negatif. Sentimen analisis dapat menyatakan perasaan emosional sedih, gembira, atau marah dan mencari pendapat tentang produk-produk, merek atau orang-orang dalam menentukan apakah mereka dilihat positif atau negatif di website (Novianti & Wibowo, 2022, p. 139).

Analisis sentimen di katagorikan dalam dua pendekatan yaitu pertama berbasis *Lexicon* yang bergantung pada kamus *lexicon* yang menampung daftar kata *lexical* dengan nilai positif atau negatif. Sedangkan yang kedua yaitu pendekatan *Machine Learning* yang dikatagorikan menjadi dua yaitu *supervised learning* dan *unsurpervised learning*, dalam hal ini *machine learning* memiliki akurasi yang lebih baik bahkan lebih cepat dibandingkan berbasis *lexicon* (Sholeha et al., 2022, p. 205)

Menurut A. Tanggu dkk dalam (Utami & Artana, 2022, p. 141) terdapat beberapa langkah analisis sentimen klasifikasi terhadap data *text* mining atau data *text* yaitu:

1. Tahap pertama yaitu mengumpulkan data atau *crawling* data seperti pendapat masyarakat, penilain terhadap produk atau lain-lain
2. *Pre-Processing* merupakan tahapan yang mencakup *Tokenization, Stopwords Removal* dan *Stemming*
3. *Transformation* yaitu melakukan pembobotan dari data teks

4. *Feature Selection* yaitu tahapan untuk mengurangi data yang tidak diperlukan
5. *Classification* yaitu tahapan klasifikasi teks dengan menggunakan metodenya yaitu *K-Nearest Neighbor, SVM, Naive Bayes* dan lain-lain.
6. *Interpretation* atau evaluasi untuk menghitung nilai *Accuracy* dan nilai *Area Unde the Curve*

2.2.2 *Tweet Harvest* (Twitter Crawler)

Tweet Harvest merupakan program perintah yang menggunakan *Playwright* untuk mendapatkan *tweet*s dari hasil pencarian *Twitter* berdasarkan kata kunci dan rentang tanggal yang ditentukan. *Tweet*s yang berhasil didapatkan kemudian akan disimpan dalam file berbentuk CSV. Dalam penggunaan *tweet harvest* dibutuhkan *authorization token* yang didapatkan dengan login ke akun *Twitter* di *browser* anda kemudian mengekstrak *authorization token*.(Vincent, 2023, p. 6)

2.2.3 *Python*

Menurut Sinaga dalam (Almaisah, 2022, p. 26) *python* merupkan bahasa pemprograman interpretatif multiguna dengan filosofi perancanaan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

2.2.4 *Text Preprocessing*

Text preprocessing atau praproses teks adalah suatu proses yang digunakan untuk melakukan transformasi teks dari yang awalnya berbentuk data tidak terstruktur yang memiliki banyak noise menjadi data yang terstruktur sehingga proses analisis sentimen dapat menjadi lebih mudah untuk dilakukan menurut Husada dan Paramita dalam (Aryasaty, 2023, p. 16).

Preprocessing terbagi beberapa tahap dalam pemeriksaan teks untuk melakukan pembersihan, memperbaiki kesalahan pada teks serta menyederhanakan teks sehingga teks dapat diproses lebih lanjut. *Preprocessing* meliputi *cleaning, case folding, normaliasasi, filtering, stemming* dan *tokenizing*.(Mayang, 2021, p. 11)

1. *Case folding*

Case folding digunakan untuk mengkonversi atau mengubah huruf kapital kedalam huruf kecil (*lowercase)* pada semua data yang terdapat didalam dokumen (Furqan et al., 2021, pp. 52–61). Pada gambar 2.1 merupakan contoh hasil *case folding*.

Sebelum	@wahyukris Program MBKM ternyata bener diimplementasikan secara "merdeka"😀
Sesudah	@wahyukris program mbkm ternyata bener diimplementasikan secara "merdeka"😀

Gambar 2.1 Contoh case folding (Pramayasa et al., 2023, p. 93)

2. *Cleaning*

Menurut (Mayang, 2021) *cleaning* merupakan tahapan dalam membersihkan atau menghilangkan karakter yang tidak diperlukan pada data dokumen. Karakter yang dihapus berupa seperti tanda baca, *username*, url, *mention*, *hastag*, re*tweet* serta simbol atau karakter *numeric* seperti : (‘”~&?<>#%{}([0-9]+:;’)[1122]. Pada gambar 2.2 merupakan contoh hasil *cleaning*.

Sebelum	@wahyukris program mbkm ternyata bener diimplementasikan secara "merdeka"😀
Sesudah	program mbkm ternyata bener diimplementasikan secara merdeka

Gambar 2.2 Contoh cleaning (Pramayasa et al., 2023, p. 93)

3. *Tokenizing*

Menurut Rasenda dalam (Syarifuddin, 2020, p. 60) *tokenizing* merupakan proses memilih dan memisahkan suatu kalimat menjadi beberapa kata, yang disebut dengan token. Dalam kalimat tertentu, proses dapat menghilangkan suatu tanda baca yang tidak diperlukan sehinggah memudahkan dalam proses olah data pada rapidminer. Pada gambar 2.3 merupakan contoh hasil *tokenizing*.

Sebelum	program mbkm ternyata bener diimplementasikan secara merdeka
Sesudah	['program', 'mbkm', 'ternyata', 'bener', 'diimplementasikan', 'secara', 'merdeka']

Gambar 2.3 Contoh tokenizing (Pramayasa et al., 2023, p. 93)

4. *Stopword Removal*

Filtering atau *stopword* merupakan tahapan dalam menghilangkan kata yang muncul dalam jumlah besar tetapi dianggap tidak memiliki makna. Pada dasarnya daftar *stopword* adalah sekumpulan kata yang banyak digunakan dalam berbagai bahasa (Syarifuddin, 2020). Pada gambar 2.4 merupakan contoh hasil *stopword*.

Sebelum	['program', 'mbkm', 'ternyata', 'benar', 'diimplementasikan', 'secara', 'merdeka']
Sesudah	['program', 'mbkm', 'diimplementasikan', 'merdeka']

Gambar 2.4 Contoh stopword (Pramayasa et al., 2023, p. 93)

5. *Stemming*

Stemming merupakan proses pemetaan dan penguraian yang berbentuk dari suatu kata menjadi bentuk kata dasarnya. Algortima *stemming* dikembangkan berdasarkan aturan mafologi bahasa Indonesia, yang mengelompokkan imbuhan menjadi awalan (*prefix*), sisipan (*infix*), akhiran (*sufflix*) dan gabungan awalan akhiran (*confixes*) (Syarifuddin, 2020, p. 90). Pada gambar 2.5 merupakan contoh hasil *stemming*.

Sebelum	['program', 'mbkm', 'diimplementasikan', 'merdeka']
Sesudah	['program', 'mbkm', 'implementasi', 'merdeka']

Gambar 2.5 Contoh stemming (Pramayasa et al., 2023, p. 93)

2.2.5 Pembobotan Kata

TF – IDF merupakan tahap yang dilakukan setelah tahap pembersihan data, tahap ini dilakukan untuk mengubah kata ke dalam bentuk angka dengan dilakukan proses pembobotan kata yang bertujuan untuk menghilangkan bobot pada masing – masing kata yang akan digunakan sebagai fitur, semakin banyak dokumen yang akan di proses maka semakin banyak fitur. Pada tahap ini terbagi atas dua bagian yaitu TF (*Trem Frequency*) dan IDF (*Inverse Document Frequency*). TF merupakan jumlah kemunculan tiap kata pada sebuah dokumen semakin kata muncul pada tiap dokumen maka semakin besar nilai TF. IDF merupakan jumlah nilai dokumen pada setiap kata yang berbanding terbalik yaitu jika suatu kata jarang muncul pada sebuah dokumen maka nilai IDF lebih besar dari pada kata yang sering muncul (Septian et al., 2019, pp. 45–56)

Menurut Salton dalam (Tupari et al., 2023) nilai IDF didapatkan dengan persamaan berikut pada 2.1:

$$IDF = \log \frac{D}{dfi} \quad (2.1)$$

D = Jumlah dokumen

dfi = jumlah kemunculan *term* terdapat D.

Perhitungan bobot TF-IDF (W) untuk setiap dokumen terdapat kata kunci dengan rumus pada 2.2:

$$W_{ij} = tf_{ij} \log(\frac{D}{dfi}) \quad (2.2)$$

2.2.6 *K-Nearest Neighbor*

K-Nearest Neighbor atau disingkat *KNN* merupakan salah satu algoritma *artifical learning* yang berbasis pada kesamaan yang bekerja berdasarkan tetangga terdekat, dalam algortima ini membutuhkan pelatihan data dan nilai K yang telah ditentukan sebelum mencari K terdekat berdasarkan jarak komputasi menurut J. Diz, G. Marreiros, dan A. Freitas, dalam (Sholeha et al., 2022, p. 205).

K-Nearest Neighbor merupakan algoritma *mechine learning* yang memiliki dua sifat yaitu metode yang bersifat *non-parametic* yang memiliki makna bahwa metode tersebut tidak membuat asumsi apapun tentang distribusi data yang mendasarinya. Dengan kata lain, tidak ada jumlah parameter yang tetap dalam model, terlepas data tersebut berukuran kecil ataupun besar. *K-Nearest Neighbor* juga bersifat *lazy learning* yang artinya tidak menggunakan titik data *training* untuk membuat model. Singkatnya *K-Nearest Neighbor* tidak ada fase *training*, kalaupun ada juga sangat minim. Semua data *training* digunakan pada tahap testing lebih lambat dan cenderung mahal atau membutuhkan banyak cost dari sisi waktu dan memori. *K-Nearest Neighbor* mengasumsikan bahwa suatu yang mirip akan ada dalam jarak yang berdekatan atau bertetangga. Artinya data-data cenderung serupa akan dekat satu sama lain (Lailany et al., 2023, p. 178).

Algoritma *K-Nearest Neighbor* dengan fungsi jarak *cosine similirity* berfungsi untuk membandingkan kemiripan antar dokumen, dalam hal ini yang dibandingkan adalah *query* dengan dokumen data latih (Riska, Indriati dan Rizal, 2019). Perhitungan *Cosine Similirty* ditunjukkan pada persamaan 2.3.

$$CosSim(q, d_j) = \frac{d_j \cdot q}{|d_j| \cdot |q|} = \frac{\sum_{i=1}^{t}(w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \quad (2.3)$$

Keterangan :

$CosSim(q, d)$: Nilai kemiripan antara dokumen uji (q) denga dokumen latih ke j (d_i)

t : Jumlah *term* (kata)

d : dokumen

q : kata kunci (*query*)

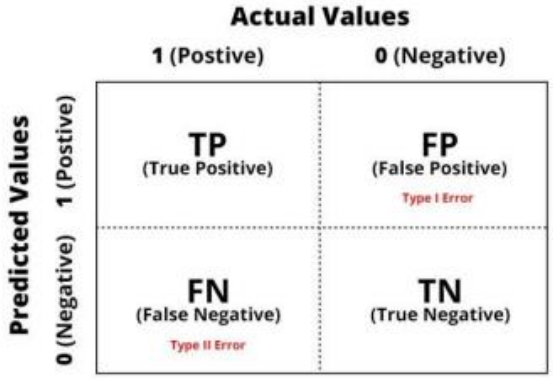
w_{ij} : Bobot *term* (kata) ke i pada dokumen latih j

w_{iq} : Bobot *term* (kata) ke i pada dokumen uji q

2.2.7 *Confusion Matrix*

Confusion matrix adalah metode yang dinilai dengan fungsi untuk apakah klasifikasi pada metode yang dinilai memiliki label baik, buruk atau netral menurut Han dalam (Larasati et al., 2023). Evaluasi menggunakan Confusion matrix menghasilkan nilai akurasi (*accurasi*), presisi (*precision*), dan recall.

Parameter TP, FP, FN, TN berdasarakan Confusion Matrix seperti pada gambar 2.6.



Gambar 2.6 Tabel confusion matrix oleh Nugroho (Hikmawan et al., 2020)

1. *Accuracy*

Accuracy adalah total nilai True Positif dan True Negatif dibagi dengan jumlah keseluruhan data. Rumus accurasi dapat dilihat pada rumus 2.4

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.4)$$

2. *Precision*

Precision adalah presentasi nilai True Positif dari seluruh nilai positif yang diprediksi. Rumus *precision* dapar dilihat pada rumus 2.5

$$Precision = \frac{TP}{TP+FP} \quad (2.5)$$

3. *Recall*

Recall adalah presentasi prediksi Positif dibandingkan dengan True Positif. Rumus *recall* dapat dilihat pada rumus 2.6

$$Recall = \frac{TP}{TP+FN} \quad (2.6)$$