

BAB II

LANDASAN TEORI

2.1 Aplikasi PLN Mobile

Aplikasi PLN Mobile adalah aplikasi yang dikembangkan oleh PT PLN (Persero) untuk memudahkan pelanggan dalam mengakses berbagai layanan kelistrikan. Aplikasi ini merupakan platform self-service yang terhubung dengan layanan online terpusat untuk pengaduan terkait pemadaman listrik. Selain itu, PLN Mobile juga mengintegrasikan dua aplikasi lainnya, yaitu Aplikasi Pengaduan Terpadu dan Aplikasi Pelayanan Pelanggan Terpusat (Syafrizal et al., 2023). Aplikasi ini menawarkan data pelanggan yang akurat, validasi yang tinggi, serta menjamin keamanan data. Sejak diluncurkan dengan tampilan dan fitur baru pada akhir 2020, PLN terus memperbarui layanan dan fitur guna meningkatkan kenyamanan pelanggan. Saat ini, PLN Mobile memiliki beberapa fitur utama seperti pembelian token listrik untuk pelanggan prabayar, pembayaran tagihan listrik bagi pelanggan pascabayar, serta monitoring penggunaan listrik dan pembelian token. Dengan fitur-fitur tersebut, PLN Mobile berupaya memenuhi kebutuhan pelanggan dalam mengelola konsumsi listrik serta mendukung penggunaan kendaraan listrik yang ramah lingkungan (Rizki Edom Noperion Manalu et al., 2023).

Aplikasi PLN Mobile merupakan bagian dari transformasi yang dilakukan PLN untuk meningkatkan pelayanan kepada pelanggan. Peluncuran aplikasi ini didorong oleh tren teknologi informasi yang semakin mengarah pada penggunaan aplikasi mobile yang berbasis sistem digital online (Natria Aminarti & Rayyan Firdaus, 2024). Sebelum adanya aplikasi ini, layanan serupa dijalankan melalui call center 123, media sosial, dan situs web PLN. Selain itu, PLN Mobile telah terintegrasi dengan Aplikasi Pengaduan dan Keluhan Terpadu (APKT) dan Layanan Pelanggan Terpusat (AP2T), sehingga memudahkan pelanggan dalam mendapatkan informasi yang cepat, tepat, dan akurat (Siti Akromah & Indah Respati Kusumasari, 2023).

2.2 Analisis Sentimen

Analisis sentimen merupakan bagian dari text mining yang mempelajari komputasi dalam Data Mining, *Natural Language Processing* (NLP), dan *Machine Learning* yang diibaratkan pada ekstraksi data sentiment isi kalimat dengan opini orang-orang, emosi, dan sentimen dan bertujuan untuk mengklasifikasikan sentimen pada suatu kalimat yang berupa opini (E. Y. Hidayat et al., 2021). Analisis sentimen dapat menentukan apakah kalimat tersebut merupakan kalimat positif atau negatif berdasarkan polaritasnya dalam suatu teks, yang dapat berupa dokumen, paragraf, kalimat atau klausa. Sistem ini dilatih menggunakan kumpulan data pelatihan, dan setelah proses pelatihan selesai, sistem akan mencoba melakukan klasifikasi otomatis atau membuat prediksi. Efisiensi sistem dievaluasi berdasarkan tingkat akurasi prediksi yang dihasilkan terhadap kumpulan data uji (Wahyuni, 2022).

Tugas utama analisis sentimen adalah mengategorikan polaritas teks yang terdiri dari dokumen, kalimat, dan paragraf. Polaritas pada klasifikasi teks dari dokumen, kalimat dan paragraf tersebut bisa bersifat positif, negatif, atau netral (Fahlevvi, 2022). Oleh karena itu, proses analisis sentimen sering melibatkan teknik pemrosesan bahasa alami (NLP) dan algoritma *machine learning* untuk mengevaluasi kata-kata dan frasa dalam teks. Langkah pertama biasanya adalah prapemrosesan teks, di mana data dibersihkan dari elemen-elemen yang tidak relevan, seperti tanda baca atau stop words. Selanjutnya, fitur teks diidentifikasi dan diekstraksi untuk menganalisis sentimen. Hasil dari analisis ini memberikan wawasan berharga tentang persepsi publik, memungkinkan perusahaan untuk membuat keputusan yang lebih informasional dan strategis. Selain itu, dengan menggunakan data organisasi dapat merespons lebih cepat terhadap umpan balik dan meningkatkan strategi mereka untuk mencapai hasil yang lebih baik.

2.3 Text Mining

Text mining adalah langkah ekstraksi pola informasi dan pengetahuan dari sejumlah data yang tidak terstruktur. Fokus utama dari proses ini adalah untuk mendapatkan pola-pola data, tren, dan ekstraksi pengetahuan yang memiliki potensi

dari data teks tersebut. Text mining melibatkan beberapa metode tahapan, termasuk *Preprocessing* teks untuk membersihkan dan mempersiapkan data, ekstraksi fitur untuk mengidentifikasi elemen penting dalam teks, dan penerapan algoritma *machine learning* untuk mengklasifikasi atau mengelompokkan teks. Salah satu tujuan utama penerapan text mining adalah untuk analisis sentimen, yang memungkinkan organisasi memahami opini dan emosi yang terkandung dalam teks, seperti ulasan pelanggan atau postingan media sosial seperti yang berguna untuk mengekstraksi informasi terhadap sumber data melewati identifikasi serta eksplorasi bentuk yang menarik dalam pengambilan keputusan yang lebih baik dan strategi bisnis yang lebih efektif (Tiara Susilawati et al., 2024).

Text mining dapat diolah untuk berbagai macam keperluan diantaranya adalah untuk summarization, pencarian dokumen teks dan sentiment analysis. Sentiment analysis merupakan bidang yang melakukan studi mengenai opini orang-orang, sentimen, evaluasi, tingkah laku dan emosi terhadap suatu entitas seperti produk, layanan, organisasi, individu, permasalahan, topik, acara dan atribut-atributnya (B. Lui, 2012). Text mining adalah penambangan yang dilakukan oleh komputer untuk mendapatkan sesuatu yang baru, sesuatu yang tidak diketahui sebelumnya atau menemukan kembali informasi yang tersirat secara implisit, yang berasal dari informasi yang diekstrak secara otomatis dari sumber-sumber data teks yang berbeda-beda. Maka dari itu bahwa text mining adalah informasi terstruktur yang digunakan untuk menganalisis atau mengelompokkan dokumen atau teks dari sejumlah besar dokumen atau teks.

2.4 Media Sosial X

Media sosial adalah platform digital yang memungkinkan individu dan komunitas berinteraksi, berbagi konten, dan berkomunikasi secara online melalui teks, gambar, video, atau suara. Platform seperti Facebook, Instagram, X, dan TikTok memengaruhi berbagai aspek kehidupan, termasuk budaya, politik, bisnis, dan hubungan sosial, dengan memungkinkan informasi menyebar cepat dan luas. Dalam dunia bisnis, media sosial menjadi alat penting untuk pemasaran produk dan interaksi antara perusahaan dengan pelanggan. Selain itu, analisis media sosial

memberikan wawasan berharga dalam riset akademis (Novitasari et al., 2020). Oleh karena itu, dapat menciptakan strategi bisnis yang efektif dan kebijakan publik yang tepat sasaran di tengah perkembangan lingkungan bisnis modern.

X adalah jenis media sosial microblogging yang memfasilitasi pengguna untuk menulis dan memublikasikan aktivitas atau pendapatnya. Secara historis, kehadiran X menawarkan ruang terbatas dengan maksimal konten yang saat ini sudah dapat memuat 280 karakter tulisan. Sama seperti media sosial lainnya, X memungkinkan pengguna menjalin jaringan, menyebarkan informasi, mempromosikan pandangan, serta membahas isu-isu terkini (trending topic). Pengguna dapat menjadi bagian dari isu tersebut dengan turut berkicau (tweet) menggunakan tagar (hashtag) tertentu. Menurut (Mailoor et al., 2019), X juga menjadi platform bagi diskusi publik yang dinamis, di mana berbagai pendapat dan isu dapat berkembang dengan cepat. Sehingga media ini memiliki peran signifikan dalam membentuk percakapan publik di era digital.

X sering digunakan oleh peneliti untuk menganalisis sentimen karena menyediakan data real-time yang kaya terkait opini publik melalui tweet yang diposting setiap hari. Analisis sentimen ini memungkinkan identifikasi emosi, seperti positif, negatif, atau netral, sehingga berguna dalam memahami reaksi masyarakat terhadap peristiwa, kampanye, produk, atau layanan tertentu. Dengan teknik text mining dan natural language processing (NLP), data besar ini dapat diolah untuk menghasilkan wawasan yang mendukung pengambilan keputusan, strategi pemasaran, hingga perencanaan kebijakan (Eka Nur Fitriyani, 2020). Melalui X API, peneliti dapat mengakses data secara terprogram, termasuk tweet, profil pengguna, dan tren, dengan fitur seperti GET search/tweets yang memungkinkan pengambilan data real-time hingga 100 tweet terakhir berdasarkan kata kunci tertentu.

2.5 Text Preprocessing

Text preprocessing adalah langkah awal penting dalam analisis sentimen karena memastikan bahwa teks siap dianalisis secara efektif oleh algoritma *machine learning*. Dalam analisis sentimen, langkah ini membantu menghapus elemen yang

tidak relevan, mengurangi variasi dalam bahasa, dan menyederhanakan teks. Tahapan dalam *Text preprocessing* meliputi Casefolding, cleansing, tokenizing, stopword removal, stemming.

Tahapan *Preprocessing* dalam pengolahan teks terdiri diantaranya sebagai berikut:

1. Case folding

Case folding adalah proses penyamaan huruf besar (kapital) dan huruf kecil dalam teks menjadi satu bentuk standar, biasanya huruf kecil (lowercase). Proses ini penting karena tidak semua dokumen konsisten dalam penggunaan huruf kapital, sehingga pencarian atau analisis teks bisa menjadi tidak akurat. Dengan case folding, kata seperti “Mobil”, “mobil”, dan “MOBIL” akan dianggap sama, yaitu menjadi “mobil”. Hal ini mempermudah proses pencocokan kata dan pengolahan data teks secara otomatis. Contohnya, kalimat “Saya Suka Makan Nasi” setelah case folding menjadi “saya suka makan nasi”.

2. Cleansing

Cleansing adalah proses pembersihan data teks dari karakter atau elemen yang tidak relevan atau mengganggu analisis. Dalam pengolahan bahasa alami (Natural Language Processing), cleansing dilakukan untuk meningkatkan kualitas data sebelum dianalisis lebih lanjut. Proses ini meliputi penghapusan tanda baca, angka, simbol, HTML tag, spasi ganda, serta karakter khusus yang tidak memiliki makna penting dalam konteks analisis. Cleansing juga bisa mencakup penghapusan kata-kata yang tidak bermakna (stopwords), seperti "yang", "dan", atau "di". Tujuan utama cleansing adalah menghasilkan teks yang bersih dan konsisten agar proses analisis menjadi lebih akurat dan efisien.

3. Tokenizing

Tokenizing adalah proses memecah teks menjadi bagian-bagian kecil yang disebut token. Token bisa berupa kata, frasa, atau bahkan karakter tergantung pada kebutuhan analisis. Dalam pengolahan teks, tokenizing sangat penting karena merupakan langkah awal untuk memahami struktur dan makna dari teks. Biasanya, token yang dihasilkan berupa kata-kata, sehingga proses ini juga dikenal sebagai

word tokenization. Contohnya, kalimat "Saya belajar mesin pembelajar." setelah tokenizing akan dipecah menjadi ["Saya", "belajar", "mesin", "pembelajar"].

4. Stopword Removal

Stopword removal adalah proses menghapus kata-kata umum yang tidak memiliki makna penting dalam analisis teks. Kata-kata ini disebut stopwords, seperti “yang”, “dan”, “di”, “ke”, “dari”, atau “adalah”. Meskipun sering muncul dalam teks, kata-kata tersebut tidak memberikan kontribusi besar terhadap pemahaman konteks atau isi dokumen. Tujuan dari stopwords removal adalah untuk mengurangi jumlah kata yang diproses dan memfokuskan analisis pada kata-kata yang lebih bermakna. Misalnya, dalam kalimat “Saya pergi ke pasar dan membeli buah”, setelah stopwords removal menjadi “pergi pasar membeli buah”. Proses ini membantu menyederhanakan data dan meningkatkan efisiensi algoritma seperti klasifikasi atau pencarian informasi.

5. Stemming

Stemming adalah proses mengubah kata-kata turunan menjadi bentuk dasarnya agar analisis teks lebih efisien dan konsisten. Dalam bahasa Indonesia, banyak kata memiliki awalan atau akhiran seperti "ber-", "me-", "-kan", atau "-an". Stemming membantu menghapus imbuhan tersebut sehingga kata-kata dengan makna dasar yang sama dapat dikenali sebagai satu entitas. Misalnya, kata "makanan", "memakan", dan "dimakan" semuanya akan dikembalikan menjadi bentuk dasar "makan". Proses ini penting dalam pengolahan teks untuk membantu mengurangi variasi kata, sehingga analisis teks lebih efisien dan akurat.

2.6 Pembobotan Kata TF-IDF

Term Frequency – Inverse Document Frequency (TF-IDF) adalah teknik pembobotan berbasis statistik yang sering diterapkan di berbagai permasalahan penggalian informasi. TF-IDF dibagi menjadi dua komponen utama yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) (W. Hidayat & Amik Riau, 2021).

1. Term Frequency (TF)

Mengukur seberapa sering kata tertentu muncul dalam sebuah dokumen. Jika kata muncul lebih sering, maka nilainya lebih tinggi. Ini adalah salah satu komponen

dalam metode pembobotan TF-IDF. Terdapat beberapa cara menghitung TF, Rumus umum untuk menghitung TF adalah:

$$TF(t, d) = \frac{f_{t,d}}{N_d} \quad (2.1)$$

Keterangan:

- a. f_t , adalah jumlah kemunculan term t dalam dokumen d .
- b. N_d adalah jumlah total term dalam dokumen d .

2. Inverse Document Frequency (IDF)

Mengukur seberapa umum atau jarang sebuah kata di seluruh corpus. Jika kata tersebut muncul di banyak dokumen, nilainya lebih rendah, karena itu dianggap kurang penting. Rumus yang umum digunakan adalah:

$$IDF(t, d) = \log\left(\frac{N}{df(t)}\right) \quad (2.2)$$

Keterangan:

- a. N adalah jumlah total dokumen dalam korpus.
- b. $df(t)$ adalah jumlah dokumen yang mengandung term t .

TF-IDF merupakan metode yang digunakan untuk tujuan pengurangan efek dari term yang muncul terlalu sering pada suatu dokumen sehingga tidak mengurangi relevansinya. Pembobotan TF-IDF ini akan lebih tinggi untuk kata-kata yang sering muncul dalam dokumen tertentu tetapi jarang muncul di dokumen lain. Persamaan TF-IDF adalah sebagai berikut :

$$W_{i,j} = tf_{ij} \times idf_{ij} = tf_{ij} \times \frac{D}{df_i} \quad (2.3)$$

Keterangan :

- a. $W_{i,j}$ adalah bobot TF-IDF untuk data ke- i dalam dokumen ke- j
- b. tf_{ij} adalah frekuensi kata ke- i dalam dokumen ke- j

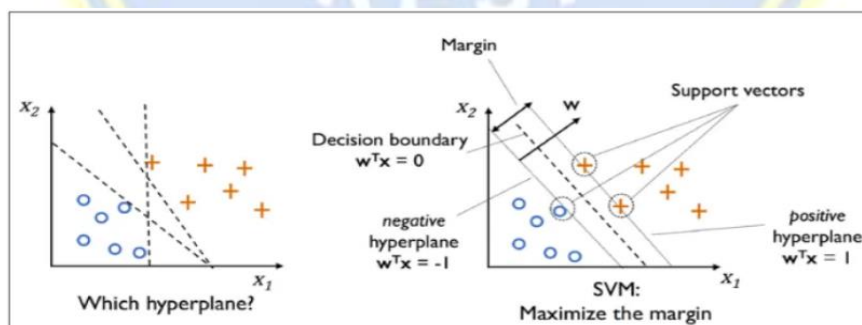
c. D adalah jumlah seluruh dokumen

d. idf_i adalah jumlah dokumen yang mengandung term ke- i

2.7 Support Vector Machine

Support Vector Machine (SVM) adalah teknik pembelajaran mesin klasik yang dapat membantu memecahkan masalah klasifikasi data besar. SVM adalah salah satu mesin terbaik yang diusulkan pada tahun 1990-an dan digunakan sebagian besar untuk pengenalan pola. Algoritma *Support Vector Machine* (SVM) adalah salah satu model machine learning multifungsi yang dapat digunakan untuk menyelesaikan permasalahan *classification, regression, dan outlier detection*. SVM termasuk ke dalam kategori *supervised learning*, dan merupakan salah satu algoritma yang paling sering digunakan dalam *machine learning* (Vamilina & Novita, 2023).

SVM juga merupakan sebuah pendekatan dalam pembelajaran mesin yang memanfaatkan konsep fungsi *hyperplane* untuk mengklasifikasikan data ke dalam berbagai daerah yang mewakili tiap kelas. *Hyperplane*, sebagai fungsi matematis, berperan sebagai pembagi antara kelas-kelas yang ada dalam data. Saat melakukan prediksi terhadap data baru, SVM akan menentukan label kelas berdasarkan letak data tersebut dalam ruang fitur yang dipisahkan oleh tepi garis pemisah dengan memperhitungkan margin atau jarak antara data dan *hyperplane* tersebut. Metode klasifikasi SVM dapat diterapkan pada data yang dapat dipisahkan secara linier. Representasi data yang dapat dipisahkan secara linier pada Gambar 2.1. adalah sebagai berikut :



Gambar 2.1 Hyperplane kelas positif (+1) dan negatif(-1)

Pada dua kelas yang dapat diajarkan secara linear, terdapat perangkat pengambilan keputusan logis yang dapat digunakan untuk mengajarkan dua kelas dengan cara yang tidak rumit membuat *hyperplane*, atau titik perpotongan, untuk kelas pertama atau kedua yang dimaksud. Secara khusus, kriteria untuk menentukan area pushback maksimum SVM ditentukan dengan memanfaatkan area pushback maksimum dari titik tertentu ke tepi pemisah *hyperplane*. Titik data yang digunakan sebagai tolak ukur disebut sebagai vektor pendukung (Fiska, 2017). Secara logis, langkah-langkah untuk membuat vektor yang memiliki dua kelas berbeda pada sejumlah dokumen adalah sebagai berikut:

$$(y_1, x_1), \dots, (y_m, x_m), \mathcal{X} \in R^n, y \in \{-1, 1\} \quad (2.4)$$

Untuk klasifikasi dengan dua kelas, rumus matematis dari *hyperplane* adalah:

$$w \cdot x + b = 0 \quad (2.5)$$

Keterangan w adalah vektor bobot berisi *koefisien* dari setiap fitur. Dimana w adalah vektor bobot dengan ukuran $1 \times n$ dengan $w = \{w_1, w_2, \dots, w_n\}$ adalah jumlah atribut atau term). x adalah vektor fitur input data dengan ukuran $m \times n$, yang dikaitkan dengan y , yaitu jenis sentimen terkait. Di sini, m adalah jumlah dokumen dan b adalah skalar tambahan atau *bias*. Berkaitan dengan persamaan (2.5) aturan keputusan untuk klasifikasi biner dengan SVM didasarkan pada *hyperplane* sebagai berikut :

$$f(x_i) = \begin{cases} +1, & \text{jika } w \cdot x_i + b \geq 0 \\ -1, & \text{jika } w \cdot x_i + b < 0 \end{cases} \quad (2.6)$$

Persamaan (2.6) adalah fungsi keputusan yang menentukan klasifikasi data input. SVM mencari fungsi keputusan terbaik dengan menemukan *hyperplane* yang memaksimalkan margin. Dengan margin ini, fungsi keputusan untuk klasifikasi biner terbentuk menjadi sebagai berikut :

$$f(x_i) = \begin{cases} +1, & \text{jika } w \cdot x_i + b \geq 1 \\ -1, & \text{jika } w \cdot x_i + b < -1 \end{cases} \quad (2.7)$$

Berdasarkan persamaan (2.7), memiliki $w \cdot x + b = 1$ sebagai pembatas pertama yang mereduksi kelas positif dan $w \cdot x + b = -1$ sebagai pembatas awal yang mereduksi kelas negatif. Garis-garis ini membentuk batas antara dua kelas. Tujuannya adalah untuk memaksimalkan margin antara dua hyperplane ini. Titik data yang terletak di dekat atau tepat pada batas *hyperplane* disebut *support vectors*. Dua bidang data berada di sisi yang benar dari hyperplane direpresentasikan dalam persamaan (2.8) dapat diterapkan sebagai berikut:

$$y_i[(w \cdot x_i) + b] - 1 \geq 0 \quad (2.8)$$

Nilai ini adalah bentuk ketidaksamaan untuk memastikan bahwa semua titik data x_i terklasifikasikan dengan benar. Jika $y_i = 1$, maka $w \cdot x_i + b \geq 1$, dan jika $y_i = -1$, maka $w \cdot x_i + b \leq -1$. Berdasarkan rumus jarak antara dua garis, maka nilai daerah perbatasan dirumuskan sebagai berikut :

$$\text{Margin} = \frac{2}{\|w\|} \quad (2.9)$$

Untuk Memaksimalkan masalah margin ini sama dengan meminimalkan nilai $\|w\|$, karena margin $M = \frac{2}{\|w\|}$ akan menjadi lebih besar saat $\|w\|$ lebih kecil. Oleh karena itu, tujuan margin adalah meminimalkan $\frac{1}{2} \|w\|^2$ yang merupakan bentuk yang lebih sederhana untuk optimasi kuadrat. Maka masalah optimasi dirumuskan sebagai berikut :

$$\min \frac{1}{2} \|w\|^2 \quad (2.10)$$

Dengan batasan fungsi $y_i[(w \cdot x_i) + b] - 1 \geq 0$, maka untuk memecahkan masalah optimasi ini, dapat menggunakan variabel *Lagrange Multiplier*. Dengan *Lagrange* $a_i \geq 0$ dapat dirumuskan fungsi ini sebagai berikut:

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i [y_i(w \cdot x_i + b) - 1] \quad (2.11)$$

Dengan nilai $a_i \geq 0$ adalah *koefisien Lagrange*. Fungsi ini membantu dalam memaksimalkan margin antara dua kelas dengan meminimumkan $\|w\|^2$ dan

memastikan bahwa *constraint* $y_i[(w \cdot x_i) + b] - 1 \geq 0$ dengan syarat bahwa semua titik data diklasifikasikan dengan benar. Nilai L optimal harus diminimalkan terhadap w dan b . Hasil ini diperoleh dari turunan parsial L terhadap w dan b . Berikut adalah turunan parsial L terhadap w .

$$\frac{\partial L}{\partial w} = 0 \rightarrow w - \sum_{i=1}^n a_i y_i x_i = 0 \rightarrow w = \sum_{i=1}^n a_i y_i x_i \quad (2.12)$$

Turunan parsial L terhadap b adalah sebagai berikut :

$$(2.13)$$

Karena hasil L minimum untuk b tidak dapat ditentukan, seperti pada persamaan (2.12), maka substitusi w pada persamaan (2.13) menjadi persamaan (2.11), yang diikuti oleh persamaan sebagai berikut:

$$\begin{aligned} L(a, b) &= \frac{1}{2} \left(\sum_{i=1}^n a_i y_i x_i \right) \left(\sum_{j=1}^n a_j y_j x_j \right) - \sum_{i=1}^n a_i \left[y_i \left(\left(\sum_{j=1}^n a_j y_j x_j \right) \cdot x_i + b \right) - 1 \right] \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j x_i x_j - \sum_{i=1}^n a_i y_i \left(\left(\sum_{j=1}^n a_j y_j x_j \right) \cdot x_i + b \right) + \sum_{i=1}^n a_i \\ &= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j x_i x_j - \sum_{i=1}^n a_i \sum_{j=1}^n a_j y_j x_j x_i - b \sum_{i=1}^n a_i y_i + \sum_{i=1}^n a_i \\ &= \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j x_i x_j - b \sum_{i=1}^n a_i y_i \end{aligned} \quad (2.14)$$

Dari persamaan (2.13), suku terakhir pada persamaan (2.14) menjadi 0 sehingga didapatkan persamaan sebagai berikut :

$$(2.15)$$

Pada fungsi dual ini, koefisien a_i menentukan kontribusi masing-masing titik data dalam mempengaruhi *hyperplane*. Hanya titik-titik yang memiliki $a_i \geq 0$ yang disebut *support vectors* yang terletak dekat dengan margin atau di antara kedua kelas. Titik data lainnya, dengan $a_i = 0$ tidak mempengaruhi posisi pemisahan antara dua kelas. Pada akhirnya, tidak mempengaruhi posisi pemisahan antara dua

kelas dengan memaksimalkan fungsi *Multiplier* ini untuk mendapatkan nilai optimal dari a_i yang menghasilkan margin maksimum antara dua kelas. Maka, pencarian pemisahan bidang terbaik dirumuskan sebagai berikut:

$$\max(L(a)) = \sum_{i=1}^n a_i \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j x_i x_j \quad (2.16)$$

persamaan (2.16) merupakan fungsi *Lagrange* untuk *Support Vector Machine* yang memaksimalkan margin antara dua kelas. Persamaan tersebut memaksimalkan fungsi *Lagrange* $L(a)$, yang dihitung dengan mengurangi nilai berdasarkan produk kernel antara data, dengan constraint $\sum_{i=1}^n a_i y_i$ dan $a_i \geq 0$ untuk $i, j = 1, 2, \dots, m, x_i$ dan x_j maka *support vector* yang relevan dapat memenuhi masing-masing kelas. Berdasarkan fungsi *Multiplier Lagrange* dapat dimaksimalkan dan digunakan untuk menentukan hambatan vektor normal w serta bias \hat{b} tersebut dari *hyperplane*. Pada persamaan (2.16) diubah menjadi rumus sebagai berikut:

$$\hat{w} = \sum_{i=1}^n \hat{a}_i x_i y_i \quad (2.17)$$

Dengan diperoleh nilai \hat{b} berdasarkan rumus persamaan (2.17) sebagai berikut:

$$\hat{b} = y_j \sum_{i=1}^n \hat{a}_i y_i [x_i \cdot x_j] \quad (2.18)$$

Dimana x_i dan x_j adalah vektor tumpuan yang memenuhi kelas $\hat{a}_i, \hat{a}_j, > 0, y_i = 1, y_j = -1$. Kedua persamaan ini merupakan hasil dari optimisasi masalah klasifikasi untuk memaksimalkan margin antara dua kelas. Berikut adalah hasil fungsi yang diperoleh berdasarkan persamaan (2.16) dan (2.17) adalah:

$$f(x_k) = \text{sign}(\hat{w} \cdot x + \hat{b}) = \text{sign}\left(\sum_{i=1}^s a_i y_i (x_i \cdot x_k) + b\right) \quad (2.19)$$

Persamaan (2.19) adalah fungsi keputusan untuk menentukan kelas dari data baru x_k dalam metode *Support Vector Machine* (SVM). Fungsi ini menggunakan tanda

sign dari hasil penjumlahan kombinasi linear dari *support vector* x_i , label kelas y_i , dan bobot a_i , ditambah bias \hat{b} . Jika hasil fungsi positif, data x_k diklasifikasikan ke kelas positif, dan jika negatif, diklasifikasikan ke kelas negatif. Fungsi ini diperoleh melalui optimisasi untuk memaksimalkan margin antara dua kelas, menggunakan *support vector* yang berada di tepi margin.

2.8 Python

Python adalah bahasa pemrograman tingkat tinggi yang populer karena sintaksisnya yang sederhana dan kemampuannya untuk mendukung berbagai aplikasi, mulai dari pengembangan perangkat lunak hingga analisis data. Bahasa ini sangat ramah bagi pemula, karena dirancang untuk membuat kode mudah dibaca dan dipahami. Python memiliki ekosistem pustaka yang luas, seperti NumPy, Pandas, dan Matplotlib, yang sangat bermanfaat untuk analisis, manipulasi, dan visualisasi data. Dalam bidang machine learning dan kecerdasan buatan (AI), Python menjadi pilihan utama berkat pustaka seperti TensorFlow, PyTorch, dan Scikit-learn yang mempermudah pengembangan model pembelajaran mesin. Selain itu, Python juga mendukung analisis teks melalui pustaka seperti NLTK dan SpaCy, yang membantu dalam pemrosesan bahasa alami.

NLTK (*Natural Language Toolkit*) adalah pustaka yang menyediakan alat-alat untuk pemrosesan teks, seperti tokenisasi, stemming, lemmatization, dan penghapusan stopwords, yang sangat berguna untuk pemrosesan bahasa alami (Bird, Klein, & Loper, 2009). Scikit-learn, pustaka machine learning populer, menawarkan berbagai algoritma klasifikasi seperti Naive Bayes serta alat untuk evaluasi model (Pedregosa et al., 2011). Selain itu, Python memiliki Pandas dan NumPy, yang menjadi andalan untuk manipulasi data dan komputasi numerik. Pandas memungkinkan manipulasi data yang mudah melalui struktur DataFrame, sedangkan NumPy mendukung operasi matematis dan array multidimensional. Kombinasi pustaka ini menjadikan Python pilihan utama untuk analisis data dan pengembangan aplikasi berbasis AI.

2.9 Evaluasi Kinerja

Evaluasi kinerja adalah model kinerja untuk mengukur akurasi prediksi dari model klasifikasi yang telah dilakukan. Evaluasi model dilakukan dengan cara melihat tingkat akurasi metode melalui *confusion matrix* dan tabel akurasi serta presisi untuk tiap model. Setelah data test diujikan terhadap data training, maka akan menghasilkan daftar kelas-kelas dari data test, sebut sata prediksi kelas (Yuliyanti & Sholihah, 2021). *Confusion Matrix* adalah tabel yang menunjukkan perbandingan antara prediksi model dengan label sebenarnya. Elemen-elemen di dalamnya terdiri dari : *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, *False Negative (FN)* (Powers, 2011).

Tabel 2.1 Confusion Matrix

| True class | Predicted Class | |
|------------|---------------------|---------------------|
| | Positive | Negative |
| Positive | True Positive (TP) | False Negative (FN) |
| Negative | False Positive (FP) | True Negative (TN) |

Berikut penjelasan singkat tentang elemen-elemen dalam *Confusion matrix* :

- True Positive (TP): Jumlah data kelas positif yang diprediksi dengan benar sebagai positif.
- True Negative (TN): Jumlah data kelas negatif yang diprediksi dengan benar sebagai negatif.
- False Positive (FP): Jumlah data kelas positif yang salah diprediksi sebagai negatif.
- False Negative (FN): Jumlah data kelas negatif yang salah diprediksi sebagai positif.

Akurasi adalah metrik yang menghitung persentase prediksi benar dari keseluruhan prediksi. Berikut ini adalah rumusnya:

$$Accuracy(\%) = \frac{TP + TNL + TNF}{TP + TNL + TNF + FP + FNL + FNF} \quad (2.20)$$

Recall adalah untuk mengukur seberapa baik model mendeteksi kelas positif dari semua data kelas positif yang ada. Rumusnya pada persamaan (2.21).

$$Recall(\%) = \frac{T(Class)}{T(Class) + F(Class)} \quad (2.21)$$

Precision adalah untuk mengukur seberapa akurat model dalam memprediksi kelas positif dari total prediksi positif. Rumusnya pada persamaan (2.22).

$$Precision(\%) = \frac{T(Class)}{T(Class) + F(Class)} \quad (2.22)$$

F1 – score adalah metrix yang menggabungkan *precision* dan *recall* menjadi satu nilai. Rumusnya pada persamaan (2.23).

$$F1 - score(\%) = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (2.23)$$

Metrik-metrik ini memberikan gambaran yang lebih komprehensif tentang kinerja model, khususnya dalam konteks klasifikasi yang tidak seimbang.

2.10 Google Collaboratory

Google Colaboratory atau Google Colab adalah platform berbasis cloud yang memungkinkan pengguna menjalankan kode Python langsung di browser tanpa memerlukan instalasi perangkat lunak tambahan. Google Colab digunakan untuk pengolahan data skala besar dan pelatihan model machine learning karena platform ini menyediakan akses gratis ke GPU dan TPU. Google Colab mendukung berbagai pustaka *Python* seperti TensorFlow, Keras, dan PyTorch, sehingga memudahkan para data scientist, peneliti, dan pengembang dalam bekerja. Selain itu, dokumen Google Colab dapat disimpan di Google Drive, memfasilitasi kolaborasi antar pengguna dan berbagi hasil pekerjaan secara real-time.

2.11 Penelitian Sejenis

Terdapat penelitian-penelitian sebelumnya yang telah dilakukan oleh orang lain yang serupa dan menjadi referensi untuk penelitian ini diantaranya adalah sebagai berikut:

1. Ni Made Ayu Juli Astari dkk. (2020) dalam penelitiannya menggunakan metode *Naive Bayes* untuk menganalisis sentimen pengguna X terhadap dampak virus corona. Tujuan dari penelitian ini adalah untuk memahami opini publik mengenai masyarakat mengekspresikan pandangan mereka mengenai situasi yang dihadapi dan membantu pemerintah dalam menyusun strategi komunikasi yang efektif untuk meningkatkan kesadaran di masyarakat dengan hasil akurasi mencapai 67%. Penelitian ini membantu pemerintah dalam strategi komunikasi pada dampak virus corona (Astari et al., 2020).
2. Wanda Athira Luqyana dkk. (2018) dalam penelitiannya menggunakan metode *Support Vector Machine* untuk menganalisis pada fenomena cyberbullying yang semakin marak terjadi di media sosial khususnya di platform Instagram. Penelitian ini mengevaluasi bagaimana algoritma *Support Vector Machine* menganalisis sentimen dalam komentar Instagram dengan mengidentifikasi dan mengklasifikasikan komentar yang bersifat positif dan negatif terkait cyberbullying. Akurasi yang dicapai menggunakan metode *Support Vector Machine* adalah 86,81% untuk mengklasifikasikan sentimen komentar (Athira Luqyana et al., 2018).
3. Farras Naufal Majid dkk. (2023) dalam penelitiannya membandingkan efektivitas metode *Support Vector Machine* (SVM) dan *Naive Bayes* (NB) dalam mengklasifikasikan pengguna aplikasi PeduliLindungi. Hasil menunjukkan SVM lebih unggul dengan akurasi 80%, presisi 83%, recall 80%, dan skor F1 81%, dibandingkan NB yang memiliki akurasi 76%, presisi 76%, recall 82%, dan skor F1 79%. Temuan ini menegaskan bahwa SVM lebih efektif untuk analisis sentimen. (Farras Naufal Majid, 2023).
4. Hibatullah Faisal dkk. (2024) dalam penelitiannya menggunakan metode *Support Vector Machine* untuk menganalisis sentimen pengguna aplikasi PLN Mobile di google playstore. Tujuan penelitian ini adalah untuk memahami bagaimana pengguna merasakan aplikasi tersebut dan apa yang mereka sukai atau tidak sukai. Hasil akurasi yang dicapai dengan metode SVM adalah sebesar 78,12%. Hasil analisis ini diharapkan dapat digunakan

oleh PT PLN (Persero) untuk merumuskan strategi yang lebih baik dalam meningkatkan layanan pelanggan dan memperkuat interaksi dengan pengguna melalui aplikasi (Faisal et al., 2024)

5. Frencis Matheos Sarimole dkk. (2024) dalam penelitiannya mengklasifikasi opini masyarakat terhadap aplikasi satu sehat dengan menggunakan metode *Support Vector Machine* dan *Naive Bayes* untuk sentimen dari data X. Hasil penelitian menunjukkan bahwa SVM memberikan akurasi yang lebih tinggi 87.95% dibandingkan dengan Naive Bayes 81.65% . Dari total 1080 data uji, SVM berhasil mengklasifikasikan 132 data sebagai sentimen positif dan 947 data sebagai sentimen negatif. Sementara itu, Naive Bayes mengklasifikasikan 132 data sebagai sentimen positif dan 947 data sebagai sentimen negatif juga. (Frencis Matheos Sarimole, 2024).
6. Faizal Fakhri Irfani dkk. (2020) dalam penelitiannya menggunakan metode *Support Vector Machine* untuk menganalisis sentimen pengguna aplikasi Ruangguru. Tujuan penelitian ini adalah untuk mengevaluasi penilaian pengguna terhadap aplikasi terkait peningkatan layanan dan fitur aplikasi. Hasil akurasi dalam penelitian sekitar 90%, menunjukkan bahwa analisis sentimen pengguna terhadap aplikasi Ruangguru cenderung positif dengan penggunaan data latih 60% dan data uji 40%. Hasil ini diharapkan dapat membantu pengembang aplikasi untuk memahami penerimaan pengguna dan melakukan perbaikan serta pengembangan lebih lanjut pada aplikasi. (Irfani, 2020) .
7. Erna Dwi Nurindah Sari dkk. (2019) dalam penelitiannya menganalisis sentimen nasabah layanan perbankan menggunakan Twitter dengan metode Regresi Logistik Biner, NBC, dan SVM. Tujuannya untuk mengidentifikasi umpan balik dan meningkatkan kualitas layanan. Hasil terbaik diperoleh dari SMOTE-SVM (kernel RBF) pada BRI dengan AUC 99,2% dan SMOTE-NBC pada Bank Mandiri dengan AUC 99,6%, membuktikan efektivitas SVM dalam memahami sentimen nasabah secara mendalam. (Erna Dwi Nurindah Sari dan Irhamah, 2019).

8. Dimas Diandra Audiansyah dkk. (2022) dalam penelitiannya menggunakan metode *Support Vector Machine* untuk menganalisis sentimen pengguna aplikasi MyXL. Tujuan penelitian ini adalah untuk mengevaluasi dalam peningkatan layanan pengguna. Hal itu dikarenakan terdapat beberapa pengguna yang memberikan rating yang tinggi namun ulasan yang diberikan merupakan review negatif. Hasil terbaik yang didapatkan dengan algoritma SVM dengan 2 kelas menggunakan nilai data latih dan data uji 80%:20%, total data seimbang dengan data 160 positif dan 160 negatif, percobaan dengan cross validation K=5 dan penggunaan kernel linear. Hasil yang didapatkan untuk nilai accuracy 88%, precision 88%, recall 88% dan f-measure 88%. (Diandra Audiansyah et al., 2022).
9. Hendry Cipta Husada dkk. (2021) dalam penelitiannya mengklasifikasi opini masyarakat terhadap maskapai penerbangan menggunakan metode *Support Vector Machine (SVM)* dengan data berbahasa Inggris pada platform X. Hasil penelitian menunjukkan bahwa kernel RBF dengan parameter complexity = 10 dan gamma = 1 memberikan akurasi terbaik sebesar 84,37%. Pengujian dengan 10-fold cross-validation menghasilkan rata-rata akurasi 80,41%. Selain itu, kernel RBF dengan parameter yang sama juga memberikan nilai precision, recall, dan F-score tertinggi, yaitu 84,33%, 84,67%, dan 84,50%. Hasil ini menunjukkan bahwa SVM dengan kernel RBF memiliki performa optimal dalam analisis sentimen maskapai penerbangan. (Husada & Paramita, 2021).
10. Khoirul Abbi Rokhman dkk. (2021) dalam penelitiannya membandingkan review komentar pada aplikasi transportasi online Gojek menggunakan metode *Support Vector Machine* dan *Decision Tree*. Penelitian ini menganalisis sentimen ulasan di Google Play untuk membandingkan keakuratan metode *Support Vector Machine* dalam mengklasifikasikan ulasan menjadi dua kategori, yaitu positif dan negatif. Hasil akurasi kemudian dibandingkan dengan metode *Decision Tree*. Melalui klasifikasi diperoleh hasil akurasi sebesar 90.20% untuk metode *Support Vector Machine* sedangkan 89.80% untuk metode *Decision Tree*. Maka, bisa

disimpulkan untuk metode Support Vector Machine nilai akurasinya lebih tinggi dibandingkan metode Decision Tree. (Khoirul Abbi Rokhman 1), 2021).



