BAB II

KAJIAN LITERATUR

2.1 Hasil Penelitian Terdahulu

Tabel 2. 1 Tabel Penelitian Terdahulu

No	Peneliti, Tahun, dan Judul	*Dataset*	Metode	Hasil
1	Nufairi dkk. (2024) Analisis sentimen pada ulasan aplikasi Threads di Google Play Store menggunakan algoritma Support Vector Machine	Ulasan aplikasi Threads (Google Play Store)	SVM	Akurasi 88%
2	Rosyida dkk. (2023) Analisis sentimen terhadap Pilpres 2024 berdasarkan opini dari Twitter menggunakan Naïve Bayes dan SVM	Opini publik Twitter (Pilpres 2024)	SVM	Akurasi 98,43%
3	Vina Agustina & Asti Herliana (2025) Analisis Sentimen Publik atas Kebijakan Efisiensi Anggaran 2025 dengan *Text Mining* dan *Natural Language Processing*	Data Twitter (kebijakan anggaran)	Naive Bayes + TF-IDF	Akurasi 93,01%
4	Dianda Rifaldi dkk. (2022) Teknik *Preprocessing* Pada *Text Mining* Menggunakan Data Tweet "Mental Health"	Data Twitter (mental health)	*Preprocessing* Text Mining	Hasil *preprocessing* menghasilkan data yang lebih bersih dan terstruktur sehingga dapat

No	Peneliti, Tahun, dan Judul	*Dataset*	Metode	Hasil
				digunakan dalam tahap analisis lanjutan seperti klasifikasi teks.
5	Nurul Humaera dkk. (2023) Analisis sentimen komentar Instagram terhadap isu PMS menggunakan SVM	Komentar Instagram (PMS)	SVM + *Stemming* Sastrawi	Akurasi 85%

2.2 Analisis Sentimen

Analisis sentimen atau yang juga dikenal sebagai *opinion mining* merupakan bidang dalam *Natural Language Processing* (NLP) yang berfokus pada identifikasi dan ekstraksi opini subjektif dari data teks. Secara lebih spesifik, analisis sentimen bertujuan untuk mendeteksi dan memahami perasaan, sikap, penilaian, serta emosi yang diungkapkan seseorang terhadap suatu entitas, baik berupa topik, produk, layanan, organisasi, maupun individu tertentu [11]. Dengan kemampuannya mengklasifikasikan teks ke dalam kategori sentimen tertentu, teknik ini telah banyak diterapkan di berbagai bidang, mulai dari analisis pasar hingga pemantauan opini publik di media sosial [13].

Perkembangan penelitian dan aplikasi analisis sentimen terus mengalami kemajuan yang signifikan seiring meningkatnya volume data teks yang tersedia di internet [14]. Dalam praktiknya, parameter sentimen dapat dikategorikan menjadi berbagai tingkatan, mulai dari sangat positif, positif, agak positif, netral, agak negatif, negatif, hingga sangat negatif tergantung pada kedalaman analisis yang dibutuhkan [15]. Namun dalam banyak penelitian, termasuk penelitian ini, pengklasifikasian disederhanakan menjadi dua kategori utama yaitu positif dan negatif.

2.2.1 Tingkat Analisis Sentimen

Dalam penerapannya, analisis sentimen dapat diterapkan pada tiga tingkat detail yang berbeda, tergantung pada kebutuhan serta karakteristik data yang dianalisis, yaitu sebagai berikut [16].

1. *Document Level Sentiment Analysis*

   Analisis dilakukan terhadap keseluruhan dokumen teks sebagai satu kesatuan untuk menentukan orientasi sentimen secara umum. Setiap dokumen diperlakukan sebagai unit tunggal yang mencerminkan opini pengguna terhadap satu entitas tertentu. Pendekatan ini paling efektif digunakan pada dokumen yang ditulis oleh satu orang dengan satu topik pembahasan yang konsisten.

2. *Sentence Level Sentiment Analysis*

   Analisis dilakukan pada tingkat kalimat dengan membedakan kalimat yang bersifat objektif (mengandung fakta) dan kalimat yang bersifat subjektif (mengandung opini). Pendekatan ini mampu menangkap variasi sentimen yang lebih detail dalam satu dokumen.

3. *Aspect Level Sentiment Analysis*

   Merupakan pendekatan paling terperinci di mana analisis difokuskan pada aspek atau atribut spesifik dari suatu entitas. Penelitian ini menggunakan pendekatan document level, di mana setiap komentar Instagram diperlakukan sebagai satu unit analisis yang utuh.

2.2.2 Kategori Sentimen

Dalam penelitian ini, klasifikasi sentimen menggunakan dua kategori yang umum digunakan dalam analisis sentimen berbasis teks, yaitu sebagai berikut [5].

1. Sentimen Positif

   Ekspresi teks yang mengandung kata-kata atau frasa yang mencerminkan penilaian baik, kepuasan, dukungan, atau kegembiraan terhadap suatu objek atau konten. Contoh: *"Kontennya sangat bermanfaat, terima kasih banyak!"*

2. Sentimen Negatif

   Teks yang mengandung ekspresi ketidakpuasan, kritik, penolakan, atau kekecewaan terhadap suatu objek atau konten. Contoh: *"Informasi yang disampaikan tidak akurat dan menyesatkan."*

2.2.3 Pendekatan Analisis Sentimen

Terdapat tiga pendekatan utama yang lazim digunakan dalam melaksanakan analisis sentimen, masing-masing dengan karakteristik dan cara kerja yang berbeda, yaitu sebagai berikut [16].

1. *Machine Learning*

   Pendekatan yang memanfaatkan algoritma pembelajaran mesin untuk membangun model yang mampu mengklasifikasikan sentimen secara otomatis berdasarkan pola yang dipelajari dari data latih. Terbagi menjadi *supervised learning* (data berlabel), *unsupervised learning* (data tanpa label), dan *semi-supervised learning* (kombinasi keduanya).

2. *Lexicon-Based*

   Pendekatan yang memanfaatkan kamus sentimen (*sentiment lexicon*) berisi daftar kata beserta nilai orientasi sentimennya. Proses klasifikasi dilakukan dengan menjumlahkan nilai sentimen setiap kata dalam teks. Pendekatan ini tidak memerlukan data latih berlabel namun cenderung kurang akurat dalam menangani konteks yang ambigu.

3. *Hybrid Methods*

   Pendekatan yang mengintegrasikan kelebihan *Machine Learning* dan *lexicon-based* secara bersamaan untuk menghasilkan analisis yang lebih komprehensif. Metode ini mampu menangani ambiguitas makna dengan lebih baik sekaligus mempertahankan akurasi klasifikasi yang tinggi.

2.3 Instagram

Instagram merupakan platform media sosial multimedia yang berada di bawah naungan Meta Platforms. Transformasi Instagram dari aplikasi berbagi foto menjadi platform yang mendukung stories, reels, hingga siaran langsung, menjadikannya instrumen komunikasi visual yang memengaruhi pola interaksi digital penggunanya [17]. Dalam ekosistem ini, pola komunikasi yang terbentuk cenderung bersifat *micro-messaging* yang sangat padat dengan prinsip ekonomi bahasa (*economy of language*) demi efisiensi komunikasi [18]. Hal ini didukung oleh laporan Socialinsider (2026) yang mencatat bahwa rata-rata panjang komentar di Instagram sangat terbatas, yakni berkisar di bawah 45 karakter per interaksi, di mana audiens lebih condong pada interaksi singkat yang praktis dan cepat dipahami. Data ini diakses pada 21 April 2026 melalui laporan tahunan *Social Media Benchmarks* yang menyajikan tren keterlibatan pengguna secara global.

Dalam konteks akademik, pemahaman terhadap karakteristik Instagram menjadi penting karena platform ini menyediakan ruang interaksi virtual sebagai sumber data empiris dalam menganalisis opini publik. Penggunaan elemen paralinguistik seperti emoji pun menjadi sangat dominan, di mana merujuk pada penelitian Lestari dkk. (2024), tercatat bahwa pengguna Instagram menyertakan emoji dalam 48% dari total teks yang mereka unggah untuk

memperjelas intensi emosi [19]. Selain itu, maraknya penggunaan bahasa gaul (*slang*) menjadi tantangan tersendiri dalam analisis data teks di Instagram. Berdasarkan penelitian Sari (2025), sekitar 75% pengguna aktif Instagram menggunakan bahasa tidak formal di kolom komentar. Fenomena ini didominasi oleh singkatan populer seperti "TB", "SM", dan "GG" yang mencapai 40%, serta penggunaan istilah santai seperti "Gue", "Lu", dan "Keren" sebesar 30% dari total interaksi [20]. Tingginya penggunaan bahasa nonbaku ini mengharuskan adanya tahap prapemrosesan yang lebih menyeluruh agar algoritma SVM mampu menghasilkan klasifikasi sentimen yang akurat dan andal.

2.3.1 Karakteristik Komentar Instagram

Komentar pada platform Instagram memiliki karakteristik yang khas dan membedakannya dari data teks pada platform lain. Pemahaman terhadap karakteristik ini sangat penting dalam merancang tahapan *preprocessing* yang tepat untuk menghasilkan model analisis sentimen yang akurat, yaitu sebagai berikut [21] .

1. Bahasa Informal dan Tidak Baku

   Komentar Instagram umumnya menggunakan bahasa sehari-hari yang informal, tidak mengikuti kaidah ejaan baku, dan sering disertai singkatan, *slang*, atau bahasa gaul yang populer di kalangan pengguna.

2. Penggunaan Emoji dan Emotikon

   Pengguna Instagram kerap menyisipkan emoji atau emotikon untuk mengekspresikan perasaan, yang menambah dimensi emosional pada komentar namun juga menambah kompleksitas dalam proses analisis teks.

3. Komentar yang Singkat

   Mayoritas komentar Instagram terdiri dari satu hingga beberapa kalimat pendek sehingga kandungan informasinya relatif terbatas dibandingkan ulasan panjang di platform lain.

4. Konteks yang Bergantung pada Unggahan

   Makna dari sebuah komentar sering kali tidak dapat dipahami sepenuhnya tanpa mempertimbangkan konteks unggahan yang dikomentari.

5. Penggunaan Tagar dan *Mention*

   Komentar Instagram sering menyertakan tagar (#) dan *mention* (@) yang perlu dihapus pada proses pembersihan data agar tidak mengganggu proses analisis.

2.3.2 *Dataset* Komentar Instagram

Dataset yang digunakan dalam penelitian ini merupakan data sekunder komentar Instagram berbahasa Indonesia dengan fokus topik *cyberbullying* terhadap akun artis dan selebgram yang diperoleh secara publik melalui platform Kaggle. Dataset ini berjumlah 650 komentar dengan distribusi seimbang yang terdiri dari 325 data berlabel *Bullying* dan 325 data berlabel *Non-bullying*. Data disimpan dalam format Excel (.xlsx) dengan dua kolom utama, yaitu teks komentar dan label kategori. Karena penelitian ini menerapkan pendekatan analisis sentimen biner, label asli dataset disesuaikan sebelum proses pelatihan model, di mana label *Non-bullying* dipetakan menjadi sentimen positif dan label *Bullying* dipetakan menjadi sentimen negatif guna memastikan konsistensi format dalam *pipeline* klasifikasi.

2.4 Text Mining

Text Mining merupakan salah satu cabang dari data mining yang secara khusus berfokus pada pengolahan dan analisis data berbentuk teks tidak terstruktur guna menghasilkan informasi atau pola yang bermakna [21]. Perbedaan mendasar antara *Text Mining* dan data mining konvensional terletak pada jenis data yang digunakan sebagai input Data mining bekerja dengan data terstruktur yang tersimpan dalam basis data relasional, sementara *Text Mining* mengelola data tidak terstruktur berupa dokumen teks, unggahan media sosial, ulasan pengguna, surat elektronik, dan berbagai bentuk teks lainnya yang tidak memiliki format baku [22].

Dalam praktiknya, *Text Mining* melibatkan serangkaian proses seperti pengkategorian teks, ekstraksi informasi, serta identifikasi kata kunci yang relevan. Untuk tugas klasifikasi, *Text Mining* memanfaatkan berbagai algoritma *Machine Learning* seperti *Naive Bayes*, *Support Vector Machine (*SVM), *K-Nearest Neighbors* (KNN), dan lain sebagainya yang bekerja untuk menemukan pola tersembunyi dalam kumpulan data teks yang tidak terstruktur [23].

2.5 Preprocessing

Preprocessing adalah serangkaian tahap pengolahan awal yang wajib dilakukan terhadap data teks mentah sebelum memasuki proses analisis lebih lanjut. Tujuan utamanya adalah mengubah data yang awalnya tidak terstruktur, tidak konsisten, dan penuh dengan derau (*noise*) menjadi data yang bersih, terstandarisasi, dan siap diproses oleh algoritma *Machine Learning*. Kualitas data yang dihasilkan dari tahap *preprocessing* secara langsung berdampak

pada akurasi dan reliabilitas model yang dibangun [24]. Tahap pengolahan awal sangat penting dalam analisis sentimen komentar Instagram karena komentar umumnya berantakan, penuh dengan bahasa gaul (*slang*), singkatan, dan simbol. Proses prapemrosesan meliputi *case folding*, *tokenizing*, *stopword removal*, *stemming*, dan normalisasi, yang bertujuan membersihkan *noise* seperti *mention* (@), *hashtag* (#), dan URL agar data siap dianalisis [25].

Adapun rincian fungsional dari setiap tahapan preprocessing dijelaskan sebagai berikut [26], [27], [28], [29].

1. *Case folding*

   Case folding adalah langkah menyeragamkan seluruh karakter huruf dalam teks menjadi huruf kecil (*lowercase*). Tujuan utama dari proses ini adalah untuk menghilangkan ambiguitas pada sistem, sehingga kata yang memiliki makna sama namun berbeda dalam penggunaan huruf kapital (misalnya "Bully" dan "bully") akan dikenali sebagai satu entitas yang identik.

2. *Data Cleaning*

   Cleaning atau pembersihan data berfungsi untuk mengeliminasi elemen-elemen yang tidak memiliki nilai informatif atau dianggap sebagai gangguan (*noise*) dalam analisis teks. Proses ini mencakup penghapusan *Uniform Resource Locator* (URL), *username* atau *mention* (@), *hashtag* (#), angka, serta simbol-simbol non-alfabet. Langkah ini menjamin bahwa *Dataset* hanya terdiri dari karakter teks bersih yang relevan dengan konten emosional.

3. *Tokenizing*

   Tokenizing merupakan proses pemecahan rangkaian kalimat menjadi potongan-potongan kata tunggal yang disebut sebagai token. Tahapan ini mengubah data dari bentuk *string* panjang menjadi sekumpulan unit kata (*list of words*) agar mempermudah komputer dalam melakukan perhitungan frekuensi kata serta analisis pola pada tahapan selanjutnya.

4. *Standardizing*

   Normalisasi atau standardisasi bertujuan untuk menyelaraskan penggunaan kata-kata tidak baku, singkatan, maupun bahasa *slang* (bahasa gaul) yang sering ditemukan di media sosial ke dalam bentuk baku sesuai kaidah bahasa yang benar (KBBI). Proses ini sangat penting dalam analisis komentar media sosial untuk menyatukan berbagai variasi penulisan kata yang memiliki makna serupa sehingga meningkatkan konsistensi data [10].

5. *Stopword Removal*

Stopword removal adalah proses penyaringan untuk membuang kata-kata umum yang memiliki frekuensi kemunculan tinggi namun tidak membawa pengaruh signifikan terhadap makna sentimen (seperti kata depan dan kata sambung: "dan", "di", "ke", "yang"). Dengan mengeliminasi *stopword*, model dapat bekerja lebih efisien dengan hanya berfokus pada kata-kata kunci yang mengandung muatan informasi inti.

6. *Stemming*

Stemming adalah proses reduksi kata berimbuhan menjadi bentuk kata dasarnya (*root word* atau *stem*). Dengan menghilangkan awalan, sisipan, maupun akhiran, kata-kata yang memiliki akar kata yang sama akan dianggap sebagai satu fitur tunggal. Langkah ini berperan penting dalam mereduksi dimensi data dan menyederhanakan kosakata dalam *Dataset* tanpa menghilangkan esensi maknanya.

2.6 Ekstraksi Fitur

Ekstraksi fitur merupakan salah satu tahap penting dalam proses pengolahan bahasa alami (*Natural Language Processing*). Tahap ini bertujuan untuk mengubah data teks yang awalnya tidak terstruktur menjadi bentuk data numerik yang terstruktur, sehingga dapat diproses oleh algoritma pembelajaran mesin (*Machine Learning*) [30]. Pada dasarnya, komputer tidak dapat memahami bahasa manusia secara langsung, baik dari segi struktur maupun maknanya. Oleh karena itu, ekstraksi fitur berperan dalam menyederhanakan teks menjadi sejumlah informasi penting yang dapat mewakili isi dari suatu dokumen.

Pada proses ini, perubahan tidak hanya sebatas mengubah kata menjadi angka, tetapi juga melibatkan perhitungan bobot tertentu untuk setiap kata. Bobot tersebut ditentukan berdasarkan seberapa sering kata muncul dan bagaimana penyebarannya dalam data. Dengan cara ini, setiap kata memiliki nilai yang menunjukkan tingkat kepentingannya dalam suatu dokumen. Hasil akhirnya berupa representasi dalam bentuk *vektor numerik*, yang memungkinkan sistem untuk mengenali pola, melakukan klasifikasi, dan mengelompokkan data dengan lebih efektif [31].

2.6.1 TF-IDF (Term Frequency – Inverse Document Frequency)

TF-IDF merupakan metode ekstraksi fitur yang berfungsi mentransformasi data teks tidak terstruktur menjadi representasi numerik. Metode ini bekerja dengan memberikan bobot pada setiap kata berdasarkan tingkat kepentingannya dalam suatu dokumen terhadap

keseluruhan korpus. Nilai akhir TF-IDF diperoleh melalui penggabungan dua komponen utama, yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) [32].

1. *Term Frequency* (TF)

   Term Frequency digunakan untuk mengukur frekuensi kemunculan suatu kata dalam sebuah dokumen tertentu. Nilai TF diperoleh dengan membagi jumlah kemunculan kata dengan total seluruh kata dalam dokumen tersebut, sebagaimana dirumuskan sebagai berikut:

   $TF(t,d) = \frac{n_{t,d}}{\sum_k n_{k,d}}$ .....(1)

   Keterangan:

   $TF(t,d)$:Skor frekuensi kata *t* di dalam dokumen *d*

   $n_{t,d}$: Jumlah kemunculan kata *t* dalam dokumen tersebut.

   $\sum_k n_{k,d}$: Akumulasi atau total seluruh kata yang terdapat dalam dokumen *d*

2. *Inverse Document Frequency* (IDF)

   Inverse Document Frequency berfungsi untuk mengevaluasi signifikansi sebuah kata di seluruh korpus. Tahap ini bertujuan untuk mereduksi bobot kata yang terlalu umum (seperti kata penghubung) dan meningkatkan bobot kata yang memiliki kekhasan informasi tinggi. Perhitungan IDF dilakukan dengan rumus:

   $IDF_t = \log\left(\frac{D}{df_t}\right)$ .....(2)

   Keterangan:

   IDF_t: nilai *inverse document frequency* untuk kata

   D: total seluruh dokumen dalam *Dataset*

   df_t: jumlah dokumen yang mengandung kata

3. Bobot TF-IDF

   Tahap akhir dalam metode ini adalah melakukan perkalian antara nilai TF dan nilai IDF yang telah diperoleh sebelumnya. Hasil perkalian ini merepresentasikan bobot kepentingan kata dalam dokumen *d.*

   $TF\text{-}IDF_{d,t} = TF_{d,t} \times IDF_t$ .....(3)

   Keterangan:

   $TF_{d,t}$: menyatakan frekuensi kemunculan kata *t* dalam dokumen *d*

   IDF_t: IDF_t: menunjukkan seberapa penting suatu kata dalam seluruh dokumen.

2.6.2 Normalisasi Vektor TF-IDF

Setelah nilai TF-IDF diperoleh, dilakukan proses normalisasi vektor untuk memastikan setiap dokumen memiliki panjang vektor yang seragam (unit length = 1). Normalisasi diperlukan agar perbedaan panjang dokumen tidak memengaruhi perbandingan antar dokumen. Metode normalisasi yang digunakan adalah normalisasi Euclidean (L2-norm), dengan rumus sebagai berikut [32].

$$\text{TF-IDF}_{norm}(d,t) = \frac{\text{TF-IDF}(d,t)}{\sqrt{\sum \text{TF-IDF}(d,k)^2}} \quad \text{.....(4)}$$

Keterangan:

TF-IDF_norm(d,t) : nilai TF-IDF yang telah dinormalisasi untuk kata t pada dokumen d

TF-IDF(d,t) : nilai TF-IDF sebelum normalisasi untuk kata t pada dokumen d

$\sum$ TF-IDF(d,k)² : jumlah kuadrat seluruh nilai TF-IDF dalam dokumen d (digunakan sebagai penyebut normalisasi)

2.6.3 N-gram (Unigram dan Bigram)

N-gram adalah urutan berkesinambungan dari n item (kata) dalam sebuah teks yang digunakan sebagai unit fitur dalam pemrosesan bahasa alami. Dalam penelitian ini, TF-IDF dikonfigurasi dengan *ngram_range* = (1,2) yang berarti menggunakan kombinasi *unigram* dan *bigram* sebagai unit fitur, yaitu sebagai berikut [33].

1. *Unigram* (n=1)

   Setiap kata tunggal diperlakukan sebagai satu fitur secara independen. Contoh: dari kalimat "konten sangat bagus" dihasilkan fitur ["konten", "sangat", "bagus"].

2. *Bigram* (n=2)

   Setiap pasangan dua kata berurutan diperlakukan sebagai satu fitur. Contoh: dari kalimat yang sama dihasilkan fitur ["konten sangat", "sangat bagus"]. Bigram membantu menangkap konteks yang tidak bisa direpresentasikan oleh kata tunggal saja, misalnya "tidak bagus" memiliki makna berbeda dibandingkan "bagus" saja.

Penggunaan kombinasi *unigram* dan *bigram* terbukti meningkatkan representasi fitur teks karena dapat menangkap konteks lokal yang lebih kaya. Dalam penelitian ini, jumlah fitur dibatasi maksimal 5.000 fitur (max_features=5000) untuk menjaga efisiensi komputasi, dan parameter min_df=2 diterapkan untuk mengabaikan kata yang hanya muncul di satu dokumen sebagai penyaring *noise*.

2.7. K-Fold Cross Validation

K-Fold Cross Validation merupakan teknik evaluasi model yang lebih robust dibandingkan single *train-test split*, terutama pada *Dataset* berukuran kecil hingga menengah. Teknik ini bekerja dengan membagi *Dataset* menjadi k bagian (*fold*) yang sama besar, kemudian proses pelatihan dan pengujian dilakukan sebanyak k kali. Pada setiap iterasi, satu fold digunakan sebagai data uji dan k-1 *fold* lainnya digunakan sebagai data latih. Pendekatan ini memungkinkan setiap data memiliki kesempatan yang sama untuk menjadi data latih maupun data uji, sehingga hasil evaluasi model menjadi lebih stabil dan tidak bergantung pada satu pembagian data tertentu [34].

Keunggulan *K-Fold Cross Validation* adalah seluruh data dapat dimanfaatkan secara optimal dalam proses evaluasi, sehingga sangat sesuai untuk dataset berukuran kecil. Metode ini mampu menghasilkan estimasi performa yang lebih representatif dan mengurangi bias karena hasil evaluasi merupakan rata-rata dari beberapa skenario pengujian. Penelitian ini menggunakan nilai (k=5) (5-*Fold Cross Validation*) karena memberikan keseimbangan yang baik antara bias, varians, serta efisiensi komputasi [35]. Selain itu, pemilihan k=5 juga berperan penting dalam meminimalisir risiko *overfitting* pada proses validasi, memastikan model tetap memiliki kemampuan generalisasi yang baik meskipun bekerja dengan jumlah data yang sangat terbatas. Selain itu, pemilihan (k=5) juga berperan penting dalam meminimalisir risiko *overfitting* agar model tetap memiliki kemampuan generalisasi yang baik pada jumlah data terbatas.

2.8 *Support Vector Machine* (SVM)

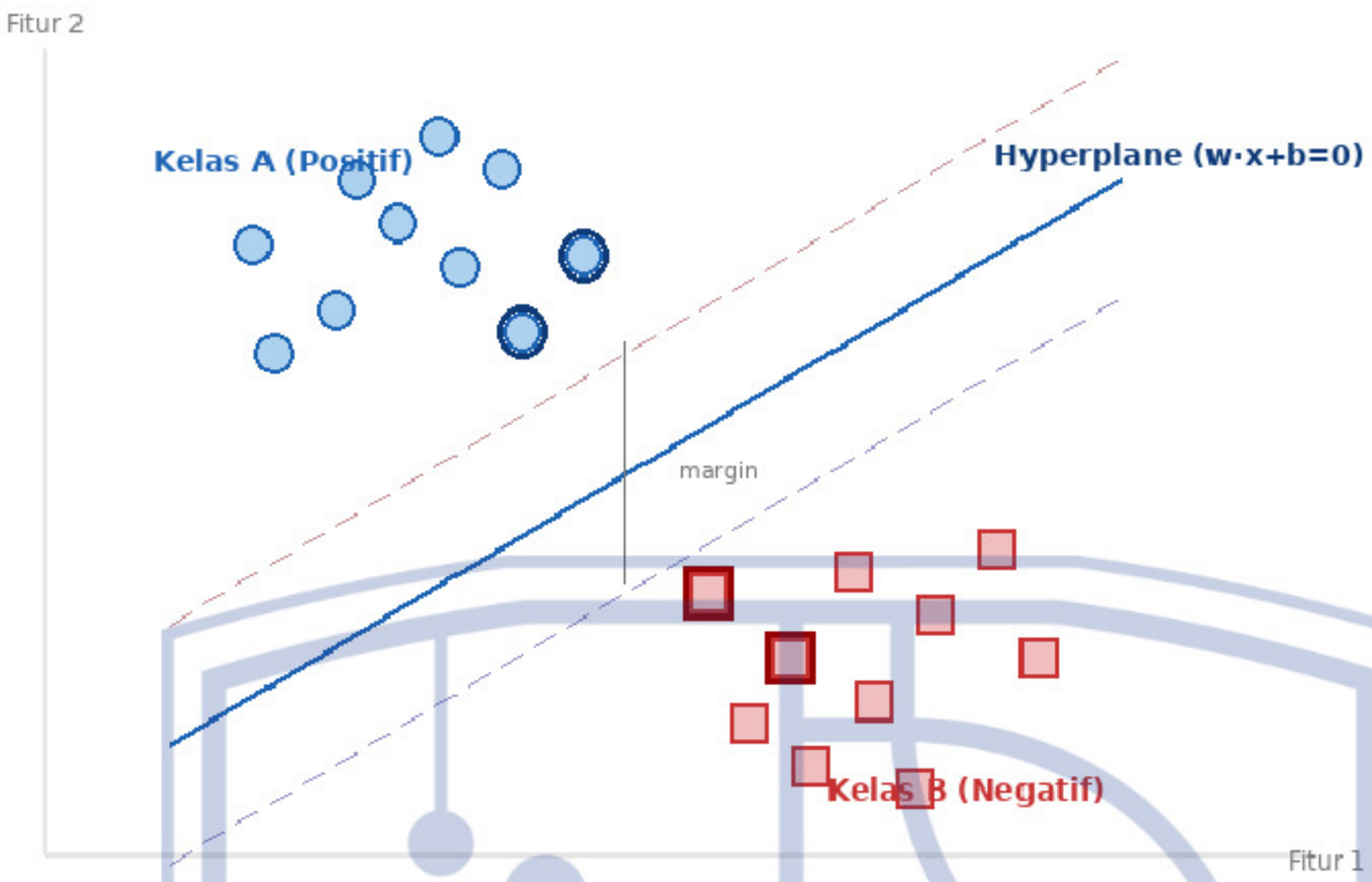
Sejak pertama kali dikembangkan oleh Vladimir N. Vapnik dan Alexey Ya. Chervonenkis pada tahun 1963, *Support Vector Machine* (SVM) telah berkembang menjadi salah satu algoritma dalam *supervised learning [36]*. Dikenal juga dengan istilah *Support Vector Network*, algoritma ini digunakan untuk menangani permasalahan klasifikasi maupun regresi dengan memanfaatkan data yang telah memiliki label. SVM bekerja dengan membentuk suatu batas pemisah yang disebut *hyperplane tunggal* untuk memisahkan data ke dalam beberapa kelas [37]. *Hyperplane* ini ditentukan sedemikian rupa agar dapat memisahkan data dengan jarak tertentu dari masing-masing kelas. Dalam prosesnya, terdapat titik-titik data yang berada paling dekat dengan batas pemisah, yang disebut sebagai *support vector*. Titik-titik tersebut memiliki peran penting dalam menentukan posisi dan orientasi dari *hyperplane* yang terbentuk [38].

Penentuan *hyperplane* dilakukan melalui proses optimasi yang mempertimbangkan jarak antar kelas, sehingga diperoleh batas pemisah yang sesuai dengan distribusi data[39]. Model yang dihasilkan kemudian digunakan untuk memprediksi kelas dari data baru berdasarkan posisi data tersebut terhadap *hyperplane* yang telah terbentuk sebelumnya.Pada kondisi tertentu, data tidak dapat dipisahkan secara linear dalam ruang berdimensi awal. Untuk mengatasi hal tersebut, SVM menggunakan pendekatan fungsi *kernel* yang memungkinkan data dipetakan ke dalam ruang berdimensi lebih tinggi [40]. Melalui proses ini, pola pemisahan antar data menjadi lebih jelas sehingga model dapat mengklasifikasikan data dengan lebih tepat. Dengan demikian, SVM dapat diterapkan pada berbagai jenis data dengan karakteristik yang beragam serta pola yang kompleks.

Dalam penerapannya pada penelitian analisis sentimen, *Support Vector Machine* (SVM) sering digunakan sebagai salah satu metode klasifikasi karena kemampuannya dalam menangani data teks berdimensi tinggi. Namun demikian, dalam beberapa penelitian, performa SVM umumnya dibandingkan dengan metode lain seperti Naïve Bayes atau Logistic Regression untuk melihat konsistensi hasil klasifikasi pada *Dataset* yang sama [20], [42]. Hal ini menunjukkan bahwa evaluasi algoritma tidak hanya dilihat dari satu metode saja, tetapi juga perlu dibandingkan dengan pendekatan lain untuk memperoleh gambaran performa yang lebih objektif.

2.8.1 Cara Kerja SVM

Secara teknis, kekuatan utama SVM terletak pada kemampuannya mencari hyperplane atau batas keputusan (*decision boundary*) yang paling optimal untuk memisahkan kelas-kelas data dalam sebuah ruang input. Strategi utama algoritma ini adalah memaksimalkan *margin* atau jarak antara titik-titik data terdekat dari setiap kelas yang disebut support vectors, untuk meminimalisir kesalahan klasifikasi [38]. Bentuk *hyperplane* ini sangat bergantung pada dimensi data yang diolah; ia dapat berupa garis lurus dalam ruang dua dimensi, namun menjadi bidang datar yang kompleks saat bekerja pada ruang multidimensi [43].



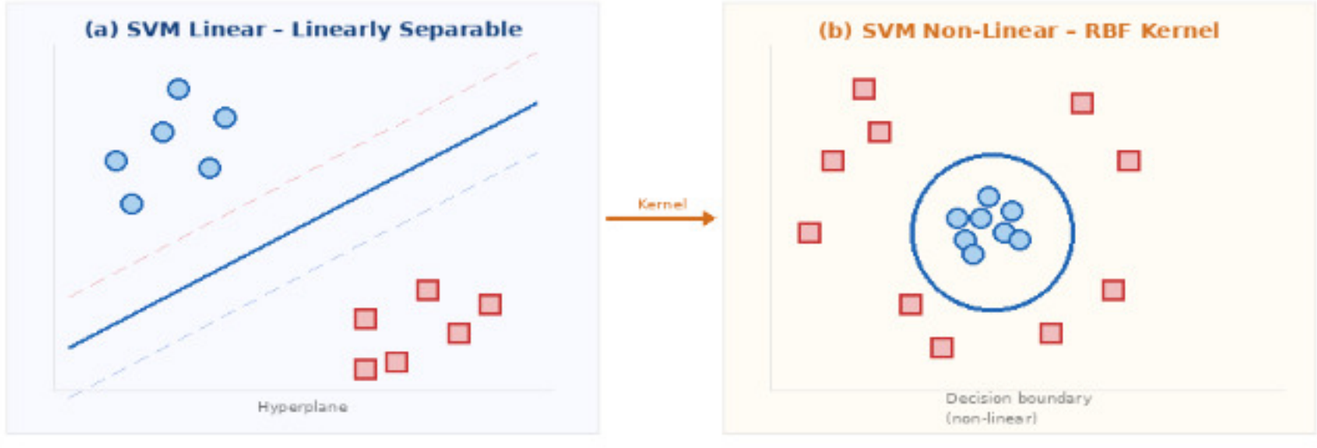
Gambar 2. 1 Ilustrasi Hyperplane dan Margin pada SVM [42]

Gambar 2.1 menunjukkan konsep dasar SVM dalam memisahkan dua kelas data menggunakan *hyperplane* yang optimal. Titik-titik berbentuk lingkaran mewakili Kelas A (data positif) dan titik-titik berbentuk kotak mewakili Kelas B (data negatif). Garis tebal berwarna biru merupakan hyperplane pemisah ($w \cdot x + b = 0$), sedangkan garis putus-putus di kedua sisinya merupakan batas *margin*. *Support vectors* adalah titik-titik data yang berada paling dekat dengan *hyperplane* dan berperan penting dalam menentukan posisi serta orientasi *hyperplane* tersebut [42].

Tujuan utama SVM adalah memaksimalkan lebar *margin* antara kedua kelas, karena *margin* yang lebih lebar mengindikasikan model yang lebih *robust* dan memiliki kemampuan generalisasi yang lebih baik terhadap data baru [44]. Pada kasus klasifikasi biner seperti penelitian ini (sentimen positif dan negatif), SVM cukup membentuk *satu hyperplane* yang secara langsung memisahkan kedua kelas tersebut.

2.8.2 Fungsi Kernel SVM

Fungsi kernel merupakan teknik matematis yang memungkinkan SVM bekerja di ruang dimensi tinggi tanpa harus secara eksplisit menghitung koordinat di ruang tersebut. Pemilihan kernel yang tepat sangat memengaruhi performa model. Gambar 2.2 berikut menunjukkan perbandingan antara SVM linear dan SVM non-linear dengan kernel *trick*.



Gambar 2. 2 Perbandingan SVM *Linear* dan *Non-Linear* (*Kernel Trick*) [45]

Jenis-jenis kernel yang umum digunakan adalah sebagai berikut [46].

1. *Linear Kernel*

   Linear Kernel digunakan untuk data yang dapat dipisahkan secara linear. Kernel ini paling efisien untuk data berdimensi tinggi seperti representasi TF-IDF pada klasifikasi teks, karena pemisahan linear sudah cukup optimal dengan beban komputasi yang ringan. Kernel ini digunakan dalam penelitian ini.

2. *Polynomial Kernel*

   Digunakan untuk pola data dengan keterkaitan fitur yang lebih kompleks, bekerja dengan memetakan data ke ruang polynomial berdimensi tinggi.

3. *Radial Basis Function* (RBF)

   Kernel paling populer yang mampu menangani hubungan antar-label yang sangat rumit dan non-linear. Cocok digunakan ketika distribusi data tidak dapat dipisahkan secara linear.

4. *Sigmoid Kernel*

   Bekerja dengan karakteristik mirip fungsi aktivasi pada jaringan saraf tiruan.

Dalam penelitian ini digunakan kernel linear karena data teks yang direpresentasikan oleh TF-IDF memiliki sifat yang cenderung *linearly separable* di ruang dimensi tinggi, sehingga kernel linear sudah mencukupi untuk menghasilkan akurasi yang optimal dengan beban komputasi yang lebih ringan dibandingkan kernel *non-linear*.

2.8.3 Parameter SVM

Untuk menghasilkan model klasifikasi yang handal, SVM mengintegrasikan beberapa parameter teknis yang harus dikonfigurasi secara tepat, yaitu sebagai berikut [37].

1. Parameter C (*Regularization*)

   Berfungsi untuk mengatur keseimbangan antara pencarian margin yang lebar dengan upaya meminimalisir kesalahan klasifikasi. Nilai C yang besar membuat model lebih ketat (sedikit *error* pada data latih namun berisiko *overfitting*), sedangkan nilai C yang kecil membuat

model lebih toleran terhadap kesalahan (margin lebih lebar, generalisasi lebih baik). Dalam penelitian ini digunakan nilai C=0,5 sebagai nilai standar yang seimbang.

2. *Hard Margin vs. Soft Margin*

   Hard Margin digunakan pada data yang bersih di mana tidak diperbolehkan adanya kesalahan klasifikasi. Soft Margin memberikan toleransi terhadap beberapa kesalahan dengan menggunakan parameter C, sehingga model lebih fleksibel dan mampu menghindari *overfitting* pada data dunia nyata seperti komentar Instagram.

Kombinasi kernel linear dengan parameter C=0,5 dipilih dalam penelitian ini karena terbukti efektif dan efisien untuk klasifikasi teks berdimensi tinggi. Data teks yang direpresentasikan oleh TF-IDF memiliki sifat yang cenderung *linearly separable* di ruang dimensi tinggi, sehingga kernel linear sudah mencukupi untuk menghasilkan akurasi yang optimal.

2.9 Pustaka Sastrawi

Sastrawi adalah pustaka (*libraryopen-source*) berbasis *Python* yang dirancang khusus untuk pemrosesan bahasa alami (*Natural Language Processing*/NLP) dalam Bahasa Indonesia. Sastrawi menyediakan dua komponen utama yang digunakan dalam penelitian ini, yaitu *Stemmer* dan *StopWordRemover*. Pustaka ini dikembangkan oleh Teguh Wahyudi dan diadaptasi dari algoritma Enhanced Confix Stripping (ECS) Stemmer yang merupakan penyempurnaan dari algoritma Nazief-Adriani untuk *stemming* Bahasa Indonesia [47].

Komponen pertama adalah *Stemmer*, yang diinisialisasi melalui kelas *StemmerFactory()*. Stemmer Sastrawi bekerja dengan mengenali imbuhan-imbuhan Bahasa Indonesia (awalan seperti me-, ber-, ter-; akhiran seperti -kan, -an, -i) dan menghilangkannya untuk mendapatkan kata dasar. Sebagai contoh, kata "membully" akan di*stemming* menjadi "bully", kata "perundungan" menjadi "rundung", dan kata "menganalisis" menjadi "analisis". Komponen kedua adalah *StopWordRemover*, yang diinisialisasi melalui *StopWordRemoverFactory()*. Komponen ini menyediakan daftar kata henti (stopwords) Bahasa Indonesia yang mencakup kata-kata seperti "yang", "di", "ke", "dan", "atau", "dengan", dan lain-lain. Dengan menghapus kata-kata tersebut, model dapat berfokus pada kata-kata yang benar-benar membawa muatan makna sentimen dalam teks [48].

2.10 Kamus *Slang* (Normalisasi Kata Tidak Baku)

Kamus *slang* adalah struktur data berbentuk kamus (*dictionary*) yang memetakan kata-kata tidak baku, singkatan, dan bahasa gaul (*slang*) yang umum digunakan di media sosial ke dalam bentuk kata baku Bahasa Indonesia yang sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). Penggunaan kamus *slang* merupakan bagian dari tahap normalisasi atau standardisasi pada proses prapemrosesan teks. Pada platform Instagram, pengguna sering menuliskan kata-kata dalam bentuk yang disingkat atau diubah ejaannya, sehingga tanpa normalisasi, model akan menganggap kata-kata yang sesungguhnya bermakna sama sebagai fitur yang berbeda dan tidak berhubungan [49].

Penerapan kamus *slang* pada tahap normalisasi berperan dalam meningkatkan kualitas fitur pada proses ekstraksi teks. Hal ini disebabkan oleh tingginya variasi penulisan kata di media sosial, sehingga tanpa penyamaan bentuk kata, proses klasifikasi dapat menghasilkan fitur yang tidak konsisten. Melalui kamus *slang*, kata-kata informal diubah menjadi bentuk standar sehingga representasi data menjadi lebih seragam**.** Normalisasi kata dapat meningkatkan performa model dalam analisis sentimen karena mampu mengurangi *noise* dan memperjelas pola linguistik dalam data teks [50].

2.11 Char TF-IDF (*Character-level* TF-IDF)

Char TF-IDF atau *Character-level TF-IDF* adalah varian dari TF-IDF yang menggunakan urutan karakter (huruf) sebagai unit fitur, bukan urutan kata. Berbeda dengan Word TF-IDF yang memecah teks menjadi kata-kata, Char TF-IDF memecah teks menjadi urutan n karakter yang disebut karakter n-gram. Pendekatan berbasis karakter ini memiliki keunggulan dalam menangani variasi ejaan, kata tidak baku, dan singkatan yang umum ditemukan pada data media sosial, karena kata-kata yang mirip secara karakter akan menghasilkan fitur yang saling tumpang tindih meskipun ejaannya berbeda [51].

Dalam implementasinya menggunakan scikit-learn, Char TF-IDF dijalankan dengan mengatur parameter *analyzer='char_wb'* pada *TfidfVectorizer*. Mode *char_wb* mengekstraksi n-gram karakter di dalam batas kata (*word boundary*), sehingga n-gram tidak melewati batas antar kata. Dalam penelitian ini, Char TF-IDF dikonfigurasi dengan *ngram_range=(2,5)*, yang berarti setiap kata diekstraksi menjadi urutan karakter sepanjang 2 hingga 5 huruf. Hasil Char TF-IDF kemudian digabungkan dengan hasil Word TF-IDF menggunakan operasi penggabungan matriks jarang (*sparse matrix hstack*) sehingga menghasilkan satu matriks fitur gabungan yang lebih kaya dan komprehensif sebagai masukan bagi model SVM [52].

2.12 Confusion Matrix

Evaluasi model merupakan tahap penting dalam proses *Machine Learning* yang bertujuan untuk mengukur kinerja model dalam melakukan klasifikasi secara objektif. Pada penelitian analisis sentimen, evaluasi model digunakan untuk mengetahui seberapa baik algoritma SVM dalam mengklasifikasikan data ke dalam kategori yang telah ditentukan. Metode evaluasi utama yang digunakan dalam penelitian ini adalah *confusion matrix* beserta metrik-metrik turunannya [53].

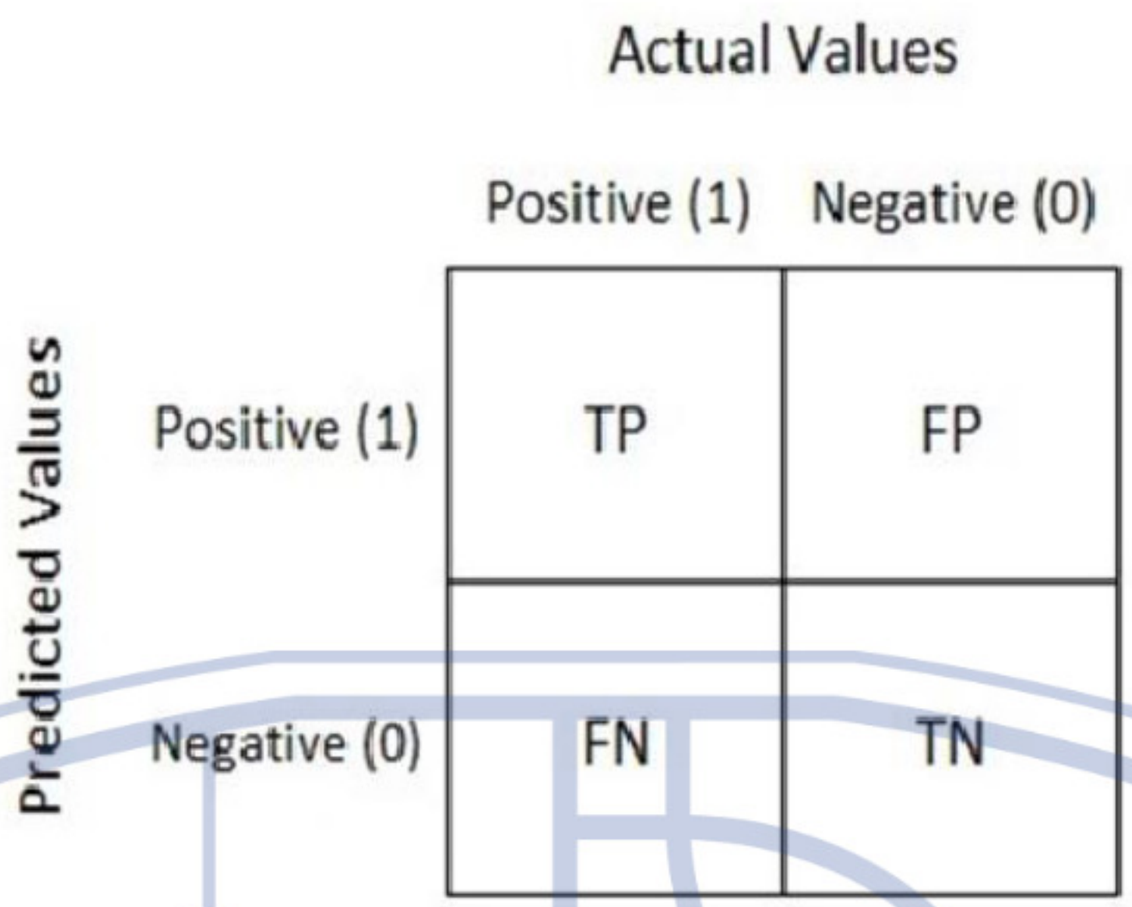
Confusion matrix adalah tabel yang menggambarkan performa model klasifikasi dengan membandingkan hasil prediksi model dengan label data aktual. Dalam kasus klasifikasi *biner*, *confusion matrix* terdiri dari empat komponen utama, yaitu sebagai berikut [54].

1. *True Positive* (TP): jumlah data positif yang diprediksi dengan benar sebagai positif oleh model.
2. *True Negative* (TN): jumlah data negatif yang diprediksi dengan benar sebagai negatif oleh model.
3. *False Positive* (FP): jumlah data negatif yang salah diprediksi sebagai positif oleh model (kesalahan Tipe I).
4. *False Negative* (FN): jumlah data positif yang salah diprediksi sebagai negatif oleh model (kesalahan Tipe II).

Tabel 2. 2 *Confusion Matrix*

Klasifikasi Prediksi	**Aktual: Positif**	**Aktual: Negatif**
Prediksi: Positif	TP (True Positive)	FP (False Positive)
Prediksi: Negatif	FN (False Negative)	TN (True Negative)

Penelitian ini menggunakan dua kelas sentimen yaitu positif dan negatif, sehingga confusion matrix yang digunakan bersifat 2x2. Visualisasi confusion matrix dua kelas ditunjukkan pada Gambar 2.3 berikut.



Gambar 2. 3 *Confusion Matrix* Klasifikasi 2 Kelas Sentimen [55]

Gambar 2.3 menyajikan *confusion matrix* untuk mengevaluasi kinerja klasifikasi sentimen biner (Positif dan Negatif). Matriks ini membandingkan nilai aktual dengan prediksi model melalui empat indikator: TP dan TN untuk prediksi benar, serta FP dan FN untuk prediksi salah. Melalui indikator ini, performa model dapat diukur menggunakan metrik akurasi, presisi, dan sensitivitas. Identifikasi pola kesalahan dalam matriks tersebut membantu peneliti melihat kecenderungan model dalam mengklasifikasikan komentar. Hal ini memudahkan analisis data yang sulit dikenali, seperti penggunaan bahasa informal di Instagram, sehingga langkah perbaikan yang tepat dapat diambil untuk meningkatkan performa klasifikasi secara keseluruhan. [55].

Berdasarkan *confusion matrix* tersebut, peneliti dapat mengidentifikasi pola kesalahan model, seperti kecenderungan *False Positive* atau *False Negative*. Informasi ini sangat berguna untuk memahami data yang sulit dikenali dan menentukan langkah perbaikan, seperti evaluasi pembobotan kata atau penambahan variasi data. Berdasarkan nilai-nilai dalam matriks tersebut, metrik evaluasi dapat dihitung sebagai berikut [54].

1. *Accuracy*

   Accuracy merupakan ukuran tingkat ketepatan model dalam melakukan klasifikasi secara keseluruhan.

   $$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \ldots\ldots\ldots\ldots (9)$$

2. *Precision*

   Precision menunjukkan tingkat ketepatan prediksi positif yang dihasilkan oleh model.

   $$Precision = \frac{TP}{TP+FP} \quad \ldots\ldots\ldots\ldots\ldots (10)$$

3. *Recall*

   Recall atau sensitivitas menunjukkan kemampuan model dalam mendeteksi data positif secara benar.

   $Recall = \frac{TP}{TP+FN}$ .....(11)

4. *F1-score*

   F1-score merupakan rata-rata harmonis antara *precision* dan *recall* yang digunakan untuk menyeimbangkan kedua metrik tersebut.

$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision+Recall}$ .....(12)