BAB II
KAJIAN LITERATUR

2.1 Artificial Intelligence

Artificial Intelligence (AI) merupakan sistem yang dirancang untuk meniru kemampuan kognitif manusia dalam memecahkan masalah. Kesimpulannya, AI dalam penelitian ini berperan sebagai otak sistem yang belajar mengenali pola defisiensi nutrisi melalui data historis (*dataset*) tanpa perlu diprogram secara eksplisit untuk setiap aturan visualnya [8]. Tujuan utama AI adalah untuk mengembangkan sistem yang dapat melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia [14].

Beberapa tujuan utama AI meliputi [15]:

1. Pengolahan Bahasa Alami *(Natural Language Processing)*.

   AI bertujuan untuk mengembangkan sistem yang mampu memahami, menghasilkan, dan berinteraksi dengan bahasa manusia secara alami. Hal ini melibatkan kemampuan sistem untuk memahami konteks, sintaksis, semantik, dan pragmatik dalam bahasa manusia.

2. Pembelajaran Mesin *(Machine Learning)*.

   AI bertujuan untuk mengembangkan algoritma dan teknik yang memungkinkan mesin untuk belajar dari data dan pengalaman, tanpa diprogram secara eksplisit. Pembelajaran mesin melibatkan identifikasi pola, membuat prediksi, dan mengoptimalkan kinerja sistem berdasarkan data yang diberikan.

3. Pengenalan Pola *(Pattern Recognition)*.

   AI bertujuan untuk mengembangkan sistem yang dapat mengenali pola dan fitur dalam data, seperti pengenalan wajah, pengenalan suara, dan pengenalan tulisan tangan. Ini melibatkan penggunaan algoritma dan teknik untuk mengidentifikasi dan mengklasifikasikan pola-pola yang ada dalam data.

4. Penalaran dan Pengambilan Keputusan *(Reasoning and Decision Making)*.

   AI bertujuan untuk mengembangkan sistem yang dapat menganalisis informasi, melakukan penalaran, dan membuat keputusan berdasarkan pemahaman konteks dan aturan yang ditentukan. Ini melibatkan penggunaan logika formal, inferensi, dan pemodelan pengetahuan untuk memungkinkan sistem AI untuk mengambil keputusan yang masuk akal.

Komponen utama dalam sistem AI meliputi [15]:

1. Basis Pengetahuan *(Knowledge Base)*.

   Tempat penyimpanan informasi dan pengetahuan yang digunakan oleh sistem AI. Basis pengetahuan dapat berisi fakta, aturan, konsep, dan model yang relevan dengan domain atau tugas yang sedang dihadapi.

2. Mesin Inferensi *(Inference Engine)*.

   Mesin inferensi bertanggung jawab untuk melakukan proses logika dan inferensi berdasarkan informasi yang ada dalam basis pengetahuan. Ini melibatkan penerapan aturan, penalaran, dan manipulasi pengetahuan untuk mencapai kesimpulan atau membuat keputusan.

3. Antarmuka Pengguna *(User Interface)*.

   Antarmuka pengguna memungkinkan interaksi antara pengguna manusia dan sistem AI. Ini dapat berupa antarmuka grafis, perintah suara, atau interaksi berbasis teks yang memungkinkan pengguna untuk memberikan input, menerima output, dan berkomunikasi dengan sistem AI.

4. Algoritma dan Teknik Kecerdasan Buatan (*AI Algorithms and Techniques*).

   Ada berbagai algoritma dan teknik yang digunakan dalam AI, termasuk pembelajaran mesin, jaringan saraf tiruan, logika proposisional, pemrosesan bahasa alami, dan banyak lagi. Algoritma ini memungkinkan sistem AI untuk memproses data, mengenali pola, belajar dari pengalaman, dan membuat keputusan.

Ada beberapa jenis AI yang digunakan dalam berbagai konteks dan aplikasi. Dalam hal ini, kita akan menjelajahi beberapa jenis utama AI, termasuk sistem berbasis aturan, sistem pakar, pembelajaran mesin, jaringan saraf tiruan, dan AI yang terkait dengan robotika. Berikut adalah beberapa jenis AI yang umum [15]:

1. Sistem Berbasis Aturan.

   Jenis AI ini menggunakan aturan-aturan yang ditentukan sebelumnya untuk menghasilkan output berdasarkan input yang diberikan. Sistem berbasis aturan terdiri dari himpunan aturan dan basis pengetahuan yang menggambarkan hubungan antara *input* dan *output*. Sistem ini bekerja dengan mencocokkan input dengan aturan yang sesuai untuk menghasilkan *output* yang diinginkan.

2. Sistem Pakar.

   Sistem pakar adalah jenis AI yang menggabungkan pengetahuan ahli manusia dengan kemampuan komputasi untuk memecahkan masalah yang kompleks dalam domain tertentu. Sistem ini beroperasi dengan mengumpulkan pengetahuan dari para ahli

manusia dan memanfaatkannya dalam membuat keputusan atau memberikan nasihat yang sesuai. Sistem pakar digunakan dalam bidang-bidang seperti kedokteran, keuangan, dan teknik.

3. Pembelajaran Mesin.

   Pembelajaran mesin adalah cabang AI yang melibatkan pengembangan algoritma dan model statistik yang memungkinkan sistem untuk belajar dari data dan pengalaman. Dalam pembelajaran mesin, sistem menganalisis data untuk mengidentifikasi pola dan membuat prediksi atau keputusan berdasarkan informasi yang diberikan. Algoritma pembelajaran mesin dapat diklasifikasikan menjadi beberapa kategori, termasuk pembelajaran terawasi, pembelajaran tak terawasi, dan pembelajaran penguatan.

4. Jaringan Saraf Tiruan.

   Jaringan saraf tiruan *(artificial neural networks)* adalah jenis AI yang terinspirasi oleh struktur dan fungsi jaringan saraf dalam otak manusia. Jaringan saraf tiruan terdiri dari banyak unit pemrosesan sederhana yang disebut neuron buatan, yang saling terhubung dan berinteraksi untuk memproses informasi. Jaringan saraf tiruan dapat digunakan untuk tugas-tugas seperti pengenalan pola, pengenalan wajah, dan pemrosesan bahasa alami.

5. AI dalam Robotika.

   AI juga digunakan dalam pengembangan robotika, di mana sistem AI dikombinasikan dengan fisik robot untuk menghasilkan perilaku yang cerdas dan adaptif. Robotika AI melibatkan penggunaan algoritma pengolahan sensorik, perencanaan gerakan, dan pembelajaran untuk memungkinkan robot berinteraksi dengan lingkungan sekitarnya dan melakukan tugas-tugas tertentu.

Dalam perkembangan AI, terdapat berbagai pendekatan dan teknik yang saling mendukung dalam membangun sistem yang cerdas. Setiap pendekatan tersebut memiliki karakteristik dan tujuan penggunaan yang berbeda sesuai dengan kebutuhan permasalahan yang dihadapi. Salah satu pendekatan yang banyak digunakan dalam pengolahan data dan analisis pola adalah pendekatan berbasis pembelajaran dari data, seperti *Machine Learning*.

2.2 Machine Learning

Machine Learning (ML) adalah cabang ilmu komputer yang berkembang dari bidang pengenalan pola dan teori pembelajaran komputer dalam kecerdasan buatan. ML mempelajari konstruksi dan analisis algoritma pembelajaran dan prediksi data. Algoritma ML membangun model berdasarkan *input* model untuk mendapatkan sebuah prediksi atau

keputusan yang didasarkan dari data daripada mengikuti instruksi program statis secara ketat [16].

ML dikategorikan dalam 3 kategori besar yaitu [17]:

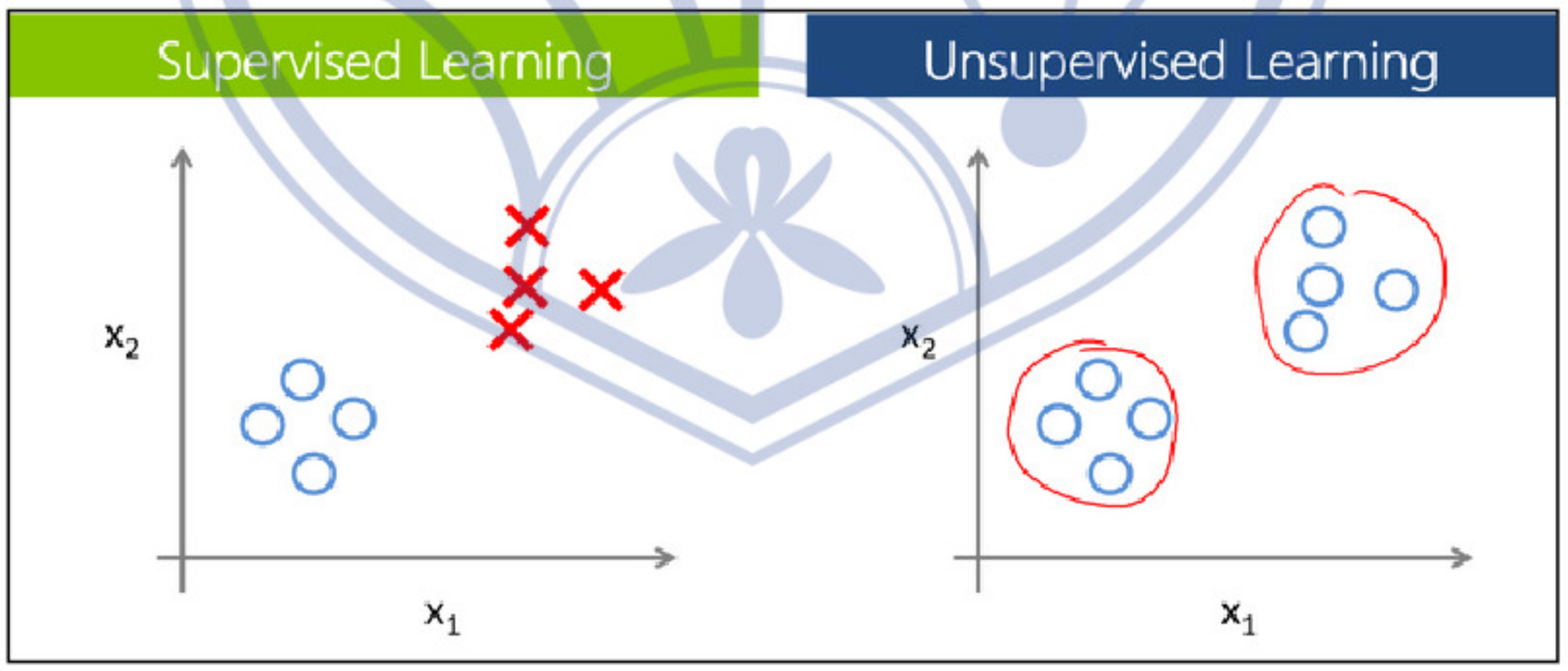
1. *Supervised Learning.*

   Model atau algoritma disajikan dengan contoh masukan dan keluaran yang diinginkan kemudian dicari pola dan hubungan antara masukan dan keluaran tersebut. Tujuannya adalah untuk mempelajari aturan umum yang memetakan *input* ke *output*. Proses pelatihan berlanjut hingga model mencapai tingkat akurasi yang diinginkan pada data pelatihan. Beberapa contoh implementasi dari *Supervised Learning* di kehidupan nyata adalah:

   a. Klasifikasi gambar dimana dilakukan pelatihan dengan menggunakan gambar beserta label yang sesuai. Tujuannya adalah agar komputer dapat mengenali objek pada gambar yang baru di masa yang akan datang.
   b. Prediksi dimana komputer dilatih menggunakan data harga saham sebelumnya untuk menghasilkan prediksi harga di masa depan.

2. *Unsupervised Learning.*

   Pada *Unsupervised Learning* data tidak diberikan label pada algoritma pembelajaran, dimana algoritma tersebut berusaha menemukan pola sebagai masukannya. Implementasi *Unsupervised Learning* dapat dimanfaatkan dalam mengelompokan polulasi yang memiliki perbedaan. *Unsupervised Learning* dapat mengelompokan data tersendiri dan menemukan pola tersembunyi yang berada di dalam data.



Gambar 2.1 Perbedaan *Supervised Learning* dan *Unsupervised Learning* [23]

Seperti yang dapat dilihat pada Gambar 2.1, data dalam *Supervised Learning* diberi label, sedangkan data dalam *Unsupervised Learning* tidak diberi label.

3. *Reinforcement Learning*.

   Suatu bidang interdisipliner dalam pemelajaran mesin dan kontrol optimal yang berkaitan dengan bagaimana suatu agen cerdas dapat mengambil aksi di lingkungan yang dinamis dalam rangka untuk memaksimalkan penghargaan kumulatif. Dalam *Reinforcement Learning*, pengembang merancang metode untuk menghargai perilaku yang diinginkan dan menghukum perilaku negatif. Metode ini memberikan nilai positif pada tindakan yang diinginkan untuk mendorong agen dan nilai negatif pada perilaku yang tidak diinginkan. Metode ini memprogram agen untuk mencari imbalan keseluruhan jangka panjang dan maksimum untuk mencapai solusi yang optimal.

Machine Learning menjadi dasar penting dalam berbagai pendekatan pengolahan data modern yang memungkinkan sistem untuk belajar dari data dan menghasilkan pola yang dapat digunakan dalam berbagai aplikasi, seperti pengolahan teks, analisis bahasa alami, serta teknik klasifikasi data, seperti *Text Mining*, *Natural Language Processing* (NLP), *Naive Bayes Multinomial,* dan *AdaBoost*.

2.2.1 Text Mining

Text Mining dapat didefinisikan secara luas sebagai proses intensif pengetahuan di mana pengguna berinteraksi dengan kumpulan dokumen dari waktu ke waktu dengan menggunakan seperangkat alat analisis. Penambangan teks berusaha untuk mengekstrak informasi yang berguna dari sumber data melalui identifikasi dan eksplorasi pola yang menarik. Dalam kasus penambangan teks, bagaimanapun, sumber data adalah kumpulan dokumen, dan pola yang menarik ditemukan bukan di antara catatan *database* yang diformalkan tetapi dalam data tekstual yang tidak terstruktur dalam koleksi dokumen. Oleh karena itu penambangan teks dan sistem penambangan data menunjukkan banyak kesamaan arsitektur tingkat tinggi. Contohnya, kedua jenis sistem ini bergantung pada rutinitas *preprocessing*, algoritma penemuan pola, dan elemen lapisan presentasi seperti alat visualisasi untuk meningkatkan penelusuran set jawaban [18].

Text Mining memiliki beberapa tipe antara lain [18]:

1. *Search and Information Retrieval*.

   Menyimpan dan menemukan kembali dokumen teks, termasuk mesin pencari dan kata kunci pencarian.

2. *Document Clustering*.

   Pengelompokan dan pengkategorian istilah, potongan, paragraf, atau dokumen menggunakan metode mining.

3. *Document Classification*.

Pengelompokan dan pengkategorian istilah, potongan, paragraf, atau dokumen menggunakan metode document *classification*.

4. *Web Mining Data.*

   Teknik ini diimplementasikan pada internet yang berfokus pada skala dan antar hubungan pada *website*.

5. *Information Extraction.*

   Identifikasi dan ekstraksi fakta yang relevan.

6. *Natural Language Processing*.

   Pemrosesan bahasa tingkat rendah yang biasanya digunakan untuk bahasa komputasi.

7. *Concept Extraction*.

   Pengelompokan kata dan frase dalam grup yang sama tahap *preprocessing text*.

Implementasi *Text Mining* pada kasus nyata adalah sebagai berikut [19]:

1. Klasifikasi *email* untuk menentukan email *spam* atau bukan sehingga digunakan oleh pemilik layanan *email* untuk memasukkan email yang bukan spam ke folder inbox dan email spam dikirim ke folder spam atau junk. Metode yang umum digunakan adalah *Naive Bayes*, *Support Vector Machine* (SVM), dan *Logistic Regression*. Teknik ini membantu penyedia layanan *email* memisahkan *email spam* ke *folder spam* atau *junk*, sementara email non-*spam* masuk ke *folder inbox*.
2. Mengetahui sentimen terhadap tokoh publik dengan mengklasifikasikan komentar masyarakat di media sosial. Dengan pengetahuan ini dapat diimplementasikan dalam sebuah aplikasi yang memungkinkan penggunanya mengetik nama tokoh publik, kemudian aplikasi akan menarik data dari media sosial dengan kata kunci nama tokoh tersebut. Selanjutnya aplikasi akan otomatis untuk menampilkan berapa persen sentimen positif dan negatif. Untuk menganalisis sentimen terhadap tokoh publik, metode seperti *Naive Bayes*, SVM, dan pendekatan berbasis leksikon sering digunakan. Aplikasi dapat menarik data dari media sosial dengan kata kunci nama tokoh tersebut dan menampilkan persentase sentimen positif dan negatif..
3. Aplikasi pemantauan keadaan fasilitas publik dengan mengklasifikasikan komentar penduduk suatu kota dari media sosial. Aplikasi secara otomatis menampilkan fasilitas publik yang rusak atau yang perlu perhatian. Metode seperti *Naive Bayes* dan SVM dapat digunakan untuk mengklasifikasikan komentar terkait kondisi fasilitas publik. Aplikasi secara otomatis menampilkan fasilitas publik yang rusak atau memerlukan perhatian berdasarkan analisis komentar masyarakat.

4. Mengetahui kinerja pelayanan publik suatu lembaga pemerintahan dengan melakukan klasifikasi data komentar masyarakat baik dari survei atau media sosial. Dengan pengetahuan ini dapat dibuat aplikasi infografik yang menampilkan kinerja lembaga-lembaga pemerintahan berdasarkan komentar masyarakat tersebut. Analisis sentimen dengan SVM adalah salah satu teknik untuk memprediksi sentimen pada sebuah teks, baik itu positif, negatif, atau netral. Dengan pengetahuan ini, dapat dibuat aplikasi infografik yang menampilkan kinerja lembaga-lembaga pemerintahan berdasarkan komentar masyarakat tersebut.
5. Mengetahui sentimen terhadap produk atau jasa dengan mengklasifikasikan komentar netizen di media sosial. Metode seperti *Naive Bayes*, SVM, dan pendekatan berbasis leksikon dapat digunakan untuk menganalisis sentimen terhadap produk atau jasa. Implementasinya dapat berupa hasil klasifikasi sentimen yang menampilkan sentimen positif dan negatif terhadap produk atau jasa tertentu.
6. Aplikasi pemantauan keadaan saat terjadi bencana alam dengan mengklasifikasikan komentar masyarakat untuk membedakan pesan bencana yang ditulis oleh masyarakat yang merasakan bencana alam, pesan bencana yang ditulis masyarakat namun bukan saksi mata langsung, dan pesan yang tidak berhubungan dengan bencana. Dengan membedakan ketiga kategori ini maka aplikasi dapat menentukan lokasi bencana dengan menggunakan geolokasi dari komentar yang ditulis oleh masyarakat yang merasakan bencana alam tersebut. Metode seperti *K-Nearest Neighbor* (K-NN) dapat digunakan untuk mengklasifikasikan komentar masyarakat terkait bencana alam.

Untuk mendukung proses analisis dalam *text mining*, diperlukan pendekatan yang mampu memahami dan mengolah bahasa manusia secara alami. *Natural Language Processing* (NLP) memainkan peran penting dan merupakan cabang dari kecerdasan buatan yang secara khusus berfokus pada interaksi antara komputer dan bahasa manusia. Dengan memanfaatkan teknik NLP, sistem dapat menginterpretasikan struktur, makna, dan konteks dari teks yang tidak terstruktur, sehingga meningkatkan efektivitas dalam ekstraksi informasi, klasifikasi dokumen, hingga analisis sentimen [15].

2.2.2 Natural Language Processing

Pengolahan bahasa alami atau dikenal dengan *Natural Language Processing* (NLP) merupakan bidang AI yang berkaitan dengan pemahaman dan penghasilan bahasa manusia. Dengan teknik NLP, komputer dapat memahami dan menghasilkan teks secara alami, memungkinkan aplikasi seperti penerjemahan otomatis, *chatbot*, dan analisis sentimen [15].

NLP merupakan cabang penting dari kecerdasan artifisial yang biasanya berhubungan dengan interaksi antara mesin/ komputer dengan manusia menggunakan bahasa alami. Salah satu tujuan akhir NLP adalah untuk mendapatkan kecerdasan dari data atau konten tidak terstruktur yang diekspresikan dalam bahasa alami, seperti bahasa Inggris, Bahasa Bengali, Bahasa Indonesia, dan sebagainya. Setiap bahasa memiliki struktur unik pada tata bahasa dan sintaksis, konvensi, serta teknik NLP bisa memungkinkan komputer untuk membaca teks, mendengar ucapan, menafsirkan, mengukur sentimen atau menambang pendapat *(opinion mining)*, dan akhirnya menentukan bagian yang penting dalam sistem intelijen tersebut [20].

NLP digunakan dalam berbagai situasi untuk memecahkan berbagai jenis masalah, di antaranya [20]:

1. Mencari serangkaian karakter.

   Pemrosesan otomatis bahasa alami memungkinkan untuk mengidentifikasi elemen tertentu dalam teks, misalnya untuk mencari kemunculan pada serangkaian karakter dalam dokumen.

2. Pengenalan entitas.

   Misalnya untuk mendapatkan nama tempat, orang, organisasi dan produk dari teks.

3. Analisis sentimen.

   Teknik ini digunakan untuk menentukan perasaan dan sikap orang terhadap suatu produk atau layanan. Hal ini berguna untuk memberikan umpan balik tentang caranya produk atau jasa yang telah dirasakan oleh pengguna/konsumen. Perusahaan di semua sektor industri dapat menganalisis data yang bersumber dari jejaring sosial untuk lebih memahami pendapat pelanggan. Tantangan utamanya adalah mengekstrak ulasan dalam bentuk teks dari konsumen dan menggunakannya secara otomatis dalam mengevaluasi produk atau merek.

4. Mesin pencari.

   Pemanfaatan NLP secara ekstensif untuk berbagai tugas, seperti *query understanding, query expansion, question answering,* pencarian informasi *(information retrieval)*, pemeringkatan, dan pengelompokan hasil *(clustering of results)*.

5. Pesan elektronik.

   Platform *email* menggunakan NLP untuk menyediakan serangkaian fitur, seperti melakukan klasifikasi *spam*, prioritas kotak masuk, dan pelengkap otomatis.

Berdasarkan pemanfaatan NLP dalam pengolahan data teks, hasil ekstraksi dan representasi fitur dari teks tersebut selanjutnya dapat digunakan sebagai input dalam proses

klasifikasi. Pada tahap ini, diperlukan suatu metode yang mampu mengelompokkan atau memprediksi kelas dari data berdasarkan pola yang telah dipelajari sebelumnya. Salah satu metode yang umum digunakan dalam klasifikasi teks adalah *Naive Bayes Classifier*, yang bekerja dengan menghitung probabilitas suatu kelas berdasarkan kemunculan fitur dalam data. Metode ini dipilih karena sederhana, efisien, serta memiliki performa yang baik dalam berbagai kasus klasifikasi, khususnya pada data berbasis teks.

2.2.3 Naive Bayes Classifier

Metode *Naive Bayes Classifier* (NBC) merupakan metode yang digunakan untuk menentukan nilai probabilitas atau kemungkinan dalam memprediksi peluang berdasarkan data pada pengalaman sebelumnya atau memungkinkan untuk membuat pengelompokan pada suatu sistem [21]. Keuntungan dari penggunaan metode NBC adalah bahwa metode ini hanya memerlukan sedikit data pelatihan untuk menentukan estimasi parameter yang diperlukan dalam proses klasifikasi [22]. Secara umum rumus NBC adalah sebagai berikut [21]:

$$P(H \mid X) = \frac{P(H \mid X)\, P(H)}{P(X)} \quad \text{.....(1)}$$

Dimana:

X = Data *class* yang belum diketahui

H = Data X merupakan hipotesa *class* yang spesifik

P(H | X) = Probabilitas hipotesa H berdasarkan kondisi X

P(H) = Probabilitas hipotesa H (prior)

P(X | H) = Probabilitas X berdasarkan kondisi

P (X) = Probabilitas dari X

Selanjutnya agar dapat menentukan kelas yang cocok bagi sampel data yang diuji coba maka rumus umum NBC diubah menjadi [21]:

$$P(H \mid X1 \ldots Xn) = \frac{P(H)\, P\,(X1 \ldots Xn \mid H)}{P(X1 \ldots Xn)} \quad \text{.....(2)}$$

Variabel H merepresentasikan class dan X1...Xn merepresentasikan *class* yang belum diketahui yang dibutuhkan untuk proses klasifikasi [21].

Beberapa varian dari metode *Naive Bayes*, yang penerapannya tergantung pada jenis data yang digunakan. Berikut adalah beberapa jenis-jenis *Naive Bayes*, yaitu [21]:

1. *Gaussian Naive Bayes* yaitu metode yang digunakan ketika data yang tersedia adalah data kontinu (numerik). Asumsi utama adalah bahwa setiap fitur mengikuti distribusi

Gaussian (normal). Fungsi densitas probabilitas untuk setiap fitur dihitung menggunakan nilai *mean* dan varians dari data tersebut.

2. *Multinomial Naive Bayes* digunakan untuk data kategori yang memiliki banyak kategori (misalnya teks, di mana setiap kata dalam dokumen dianggap sebagai kategori). Asumsi dasar dari metode ini adalah bahwa fitur-fitur diwakili oleh distribusi multinomial. Algoritme ini sangat sering digunakan dalam masalah pengklasifikasian teks, seperti analisis sentimen atau deteksi spam.
3. *Bernoulli Naive Bayes* hampir mirip dengan *Multinomial Naive Bayes*, namun digunakan ketika data yang dianalisis berbentuk biner (misalnya, fitur dengan dua nilai: ada atau tidak ada, 0 atau 1). Bernoulli Naive Bayes mengasumsikan bahwa setiap fitur mengikuti distribusi Bernoulli, di mana nilai fitur hanya bisa bernilai 1 atau 0.
4. *Categorical Naive Bayes* digunakan ketika fitur-fitur dalam data bersifat kategorikal, dan model ini mengasumsikan bahwa setiap fitur adalah hasil dari distribusi probabilitas kategori tertentu. Klasifikasi ini berguna pada data dengan banyak fitur kategorikal dan sedikit data numerik.

2.2.4 Gaussian Naive Bayes

Gaussian Naive Bayes adalah salah satu varian dari metode *Naive Bayes* yang digunakan untuk data kontinu (numerik), dengan asumsi bahwa setiap fitur mengikuti distribusi *Gaussian* (normal). Model ini sangat cocok digunakan pada representasi fitur berbentuk vektor numerik, seperti hasil *embedding* dari model BERT pada teks. Dalam konteks analisis sentimen, setiap dokumen direpresentasikan sebagai vektor fitur numerik, sehingga *Gaussian Naive Bayes* menghitung probabilitas berdasarkan distribusi normal dari setiap fitur pada masing-masing kelas. Model ini digunakan untuk mengklasifikasikan teks atau dokumen dengan memanfaatkan nilai *mean* dan varians dari setiap fitur pada tiap kelas. Rumus *Gaussian Naive Bayes* adalah [23]:

1. Rumus untuk asumsi *Gaussian*.

Untuk *Gaussian Naive Bayes*, *likelihood* $P(x_i|C_k)$dimodelkan menggunakan distribusi normal (*Gaussian*) dengan rumus:

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(x_i-\mu_k)}{2\sigma_k^2}\right) \quad (3)$$

Dimana:

x_i adalah nilai fitur ke-i.

μ_k adalah rata-rata *(mean)* dari fitur pada kelas C_k

σ_k^2 adalah varians dari fitur pada kelas C_k

2. Rumus total *log* probabilitas

Penjumlahan dari probabilitas *posterior* dan semua *log likelihood* (asumsi *Gaussian)*:

$$log\ P(C_k|x) = log\ P(C_k) + \sum_{i=1}^{n} logP(x_i|C_k) \quad (4)$$

Dimana:

$P(C_k)$ adalah *prior probability* dari kelas C_k

$P(x_i|C_k)$ adalah likelihood dari fitur ke-i terhadap kelas C_k

n adalah jumlah fitur

Meskipun *Gaussian Naive Bayes* mampu memberikan klasifikasi teks yang cukup efektif, model ini memiliki keterbatasan dalam menangani fitur yang kompleks dan interaksi antar kata yang bersifat kontekstual. Untuk meningkatkan akurasi klasifikasi sentimen pada ulasan aplikasi Mobile JKN, penerapan *AdaBoost* sebagai metode *ensemble learning* dilakukan. *AdaBoost* bekerja dengan menggabungkan beberapa *weak classifier*, termasuk *Multinomial Naive Bayes*, menjadi satu model yang lebih kuat. Setiap *weak classifier* dilatih secara bertahap, dan bobot diberikan lebih tinggi pada data yang sebelumnya salah diklasifikasikan, sehingga model akhir memiliki kemampuan adaptif dalam menangani kesalahan prediksi dan memperkuat performa klasifikasi secara keseluruhan.

2.2.5 Adaptive Boosting (AdaBoost)

Algoritma *Adaptive Boosting (AdaBoost)* merupakan metode yang digunakan untuk pengambilan keputusan *Adaboost* merupakan *ensemble learning* yang digunakan pada metode *boosting*. Algortima ini ditujukan untuk *supervised learning* yang memberikan nilai atribut pada *dataset* yang digambarkan oleh koleksi atribut dan termasuk salah satu dari serangkaian kelas yang saling berhubungan Langkah-langkah pada metode *Adaboost* adalah sebagai berikut [24]:

1. Inisial setiap bobot awal dengan rumus berikut untuk semua i.

$$W_i^1 = \frac{1}{n} \quad (5)$$

Dimana:

n: jumlah total data sampel dalam *dataset*.

W_i^1: bobot awal untuk data kei pada iterasi pertama. Semua data diberikan bobot yang sama di awal.

2. Untuk setiap m = 1, 2, 3, …, M. Lakukan:
3. Bangun model klasifikasi $G_m(X_i)$ menggunakan bobot $W_i^{(m)}$.

4. Hitung kesalahan klasifikasi dengan rumus:

$$\varepsilon_m = \frac{\sum_{i=1}^{n} w_i^{(m)}, G_m(X_i) \neq Y_i}{\sum_{i=1}^{n} w_i^{(m)}} \quad \text{.....(6)}$$

Dimana:

ε_m: tingkat kesalahan *(error rate)* dari model ke-m

Y_i: label atau nilai target yang sebenarnya

$G_m(X_i)$: kondisi di mana prediksi salah. Rumus ini menjumlahkan bobot dari semua data yang salah diklasifikasikan, lalu membaginya dengan total bobot.

5. Menghitung nilai α menggunakan rumus:

$$\alpha_m = log\frac{1 - \varepsilon_m}{\varepsilon_m} \quad \text{.....(7)}$$

Dimana:

α_m: Bobot pengaruh model G_m dalam keputusan akhir

Jika *error* (ε_m) kecil, maka α_m akan besar (model sangat dipercaya)

6. Perbaharui koefisien pembobotan kesalahan klasifikasi dengan rumus:

$$w_i^{m+1} = w_i^m \exp(-\alpha_m), G_m(X_i) = Y_i \quad \text{.....(8)}$$

$$w_i^{m+1} = w_i^m \exp(\alpha_m), G_m(X_i) \neq Y_i \quad \text{.....(9)}$$

7. Penentuan klasifikasi menggunakan rumus:

$$Y = Sign\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right) \quad \text{.....(10)}$$

Dimana:

Y: prediksi akhir.

$\sum \alpha_m G_m(x)$: penjumlahan keputusan dari semua model yang masing-masing dikalikan dengan bobot kepercayaannya (α_m)

Sign: Fungsi tanda. Jika hasil penjumlahan positif, kelasnya +1; jika negatif, kelasnya -1.

Dengan demikian, proses pembelajaran pada metode *AdaBoost* menghasilkan model klasifikasi yang lebih kuat melalui penggabungan beberapa model lemah *(weak classifier)* menjadi satu model yang lebih akurat. Meskipun efektif dalam meningkatkan performa klasifikasi, metode ini masih bergantung pada representasi fitur yang telah ditentukan sebelumnya. Oleh karena itu, pada perkembangan selanjutnya digunakan pendekatan yang lebih modern seperti *Bidirectional Encoder Representations from Transformers* (BERT) yang mampu memahami konteks kata secara lebih mendalam dalam suatu kalimat, sehingga meningkatkan kualitas pemahaman terhadap data teks sebelum masuk ke proses klasifikasi lanjutan.

2.3 Bidirectional Encoder Representations from Transformers

Transformer merupakan arsitektur *deep learning* yang diperkenalkan oleh Ashish Vaswani dkk. pada tahun 2017 melalui penelitian berjudul *"Attention Is All You Need"*. Arsitektur ini dirancang untuk memproses data sekuens, seperti teks, dengan lebih efisien dibandingkan model sebelumnya seperti *Recurrent Neural Network (RNN)* dan *Long Short-Term Memory (LSTM)*. *Transformer* menjadi dasar bagi berbagai model *Natural Language Processing (NLP)* modern karena kemampuannya dalam menangkap hubungan antar kata dalam sebuah kalimat, meskipun kata-kata tersebut terpisah dalam jarak yang jauh [24].

Salah satu komponen utama dalam *Transformer* adalah mekanisme *self-attention*, yaitu teknik yang memungkinkan model untuk memberikan bobot atau tingkat kepentingan pada setiap kata dalam suatu kalimat terhadap kata lainnya. Dengan adanya self-attention, model dapat memahami konteks secara menyeluruh tanpa harus memproses data secara berurutan. Arsitektur *Transformer* terdiri dari dua bagian utama, yaitu [25]:

1. *Encoder*, yang bertugas untuk memahami dan merepresentasikan *input* teks ke dalam bentuk vektor.
2. *Decoder*, yang bertugas untuk menghasilkan *output* berdasarkan representasi dari *encoder*.

Keunggulan utama *Transformer* dibandingkan model sebelumnya adalah kemampuannya dalam melakukan proses secara paralel, sehingga lebih cepat dalam pelatihan dan mampu menangani data dalam jumlah besar. Selain itu, *Transformer* juga lebih efektif dalam menangkap hubungan jangka panjang dalam teks. Dengan keunggulan tersebut, Transformer menjadi fondasi bagi berbagai model lanjutan, salah satunya adalah *Bidirectional Encoder Representations from Transformers* (BERT), yang memanfaatkan bagian *encoder* dari *Transformer* untuk memahami konteks bahasa secara lebih mendalam [25].

Metode *Bidirectional Encoder Representations from Transformers* (BERT) merupakan model salah satu model kontekstual yang dirancang diatas *transformer* model, Oleh karena itu BERT mampu menangkap dan mengklasifikasikan suatu kata dalam konteks dan level tertentu sesuai dengan struktur model yang dirancang. BERT merupakan terobosan penting dalam pemahaman konteks dalam bahasa dengan memproses teks secara dua arah (*bidirectional*). Berbeda dengan model sebelumnya yang membaca teks secara satu arah (dari kiri ke kanan atau sebaliknya), BERT menggunakan arsitektur *Transformer* untuk menangkap konteks dari kedua arah secara bersamaan, sehingga model ini dapat memahami

makna kata berdasarkan kata-kata di sekitarnya. Terdapat beberapa konsep utama BERT [25]:

1. Konteks *Bidirectional*.

   Salah satu inovasi utama BERT adalah kemampuannya untuk membaca teks dalam dua arah. Berbeda dengan model bahasa tradisional yang memprediksi kata berikutnya berdasarkan kata sebelumnya (dari kiri ke kanan) atau memprediksi kata sebelumnya berdasarkan kata berikutnya (dari kanan ke kiri), BERT mempertimbangkan konteks kiri dan kanan dari sebuah kata secara bersamaan. Hal ini membantu menciptakan pemahaman yang lebih akurat dan lengkap tentang makna suatu kata.

2. Arsitektur *Transformer*.

   BERT didasarkan pada model *Transformer*, yang merupakan jenis arsitektur pembelajaran mendalam yang diperkenalkan oleh Vaswani et al. pada tahun 2017. *Transformer* menggunakan mekanisme *self-attention* untuk menilai pentingnya setiap kata dalam teks input, memungkinkan model untuk menangkap hubungan antara kata-kata, meskipun kata-kata tersebut terpisah jauh dalam kalimat.

3. *Pre-training dan Fine-tuning*.

   BERT dilatih sebelumnya *(pre-trained)* dengan menggunakan sejumlah besar data teks, di mana model ini belajar untuk memprediksi kata yang hilang (*Masked Language Model*/MLM) dan untuk menentukan apakah satu kalimat secara logis mengikuti kalimat lainnya (*Next Sentence Prediction*/NSP). Setelah tahap *pre-training*, BERT dapat disesuaikan lebih lanjut *(fine-tuned)* untuk tugas-tugas NLP tertentu, seperti analisis sentimen, pengenalan entitas bernama (NER), dan klasifikasi teks, dengan menggunakan dataset yang sesuai.

4. *Named Entity Recognition* (NER).

   Dalam konteks kode yang diberikan, BERT digunakan untuk tugas *Named Entity Recognition* (NER). NER adalah sub-tugas dari ekstraksi informasi yang bertujuan untuk mengidentifikasi entitas penting dalam teks, seperti nama produk, nama perusahaan, lokasi, dan istilah-istilah relevan lainnya. Dalam model NER menggunakan BERT, setiap token dalam teks diprediksi untuk mengidentifikasi apakah itu adalah entitas yang relevan, sehingga membantu dalam ekstraksi informasi yang lebih mendalam dari teks yang dianalisis.

Berikut ini dijelaskan secara detail berupa rumus dasar dari metode BERT adalah [26]:

$$Attention(Q,\ K,\ V)\ =\ Soft\max\left(\frac{Q\,K^T}{\sqrt{d_k}}\right)V \quad \text{.....(11)}$$

Di mana:

Q adalah matriks *query*,

K^t adalah transpose dari matriks *key*,

D_k adalah dimensi dari *vector key*,

Softmax adalah fungsi untuk menormalkan skor perhatian.

Sebelum vektor hasil BERT digunakan pada tahap klasifikasi atau pemodelan lebih lanjut, dilakukan proses normalisasi fitur untuk menyamakan skala nilai antar dimensi. Hal ini penting karena output embedding BERT memiliki rentang nilai yang bervariasi pada setiap dimensi, sehingga dapat mempengaruhi kinerja model *Machine Learning* seperti SVM, *Logistic Regression*, atau *Neural Network*. Teknik normalisasi yang umum digunakan adalah *Standard Scaler (Z-score normalization).*

2.4 Standar Scaler (Z-score Normalization)

Standard Scaler atau *Z-score Normalization* merupakan salah satu metode normalisasi data yang digunakan untuk mengubah nilai fitur agar memiliki skala yang seragam. Teknik ini dilakukan dengan cara mentransformasikan data sehingga memiliki rata-rata *(mean)* = 0 dan standar deviasi = 1. Dengan demikian, setiap fitur akan berada pada distribusi yang sama sehingga tidak ada fitur yang mendominasi karena perbedaan skala nilai. Secara matematis, *Standard Scaler* dinyatakan dengan rumus berikut [27]:

$$z\ =\ \frac{x-\mu}{\sigma} \quad \text{.....(12)}$$

Di mana:

z = nilai hasil normalisasi (*z-score*)

x = nilai data asli

μ = rata-rata (*mean*) dari data

σ = standar deviasi dari data

Proses standardisasi ini bekerja dengan mengukur seberapa jauh suatu nilai x berada dari rata-rata dalam satuan standar deviasi. Jika nilai z bernilai positif, maka data berada di atas rata-rata, sedangkan jika bernilai negatif berarti berada di bawah rata-rata. Penggunaan *Standard Scaler* memberikan beberapa keuntungan, antara lain meningkatkan kestabilan proses pelatihan model, mempercepat konvergensi algoritma, serta menghindari dominasi fitur dengan nilai besar. Dalam konteks pemrosesan teks menggunakan representasi vektor

seperti hasil *embedding* BERT, normalisasi ini dapat membantu menyamakan distribusi nilai antar dimensi sehingga lebih optimal untuk proses klasifikasi atau analisis lebih lanjut.

2.5 Analisis Sentimen

Analisis sentimen di media sosial adalah proses menilai dan memahami opini atau perasaan pengguna terhadap suatu topik melalui data teks yang dihasilkan di platform media sosial. Proses ini bertujuan untuk mengidentifikasi apakah sentimen yang diekspresikan bersifat positif, negatif, atau netral. Dengan meningkatnya penggunaan media sosial, analisis sentimen menjadi alat penting bagi bisnis dan organisasi untuk memahami persepsi publik terhadap produk, layanan, atau isu tertentu. Informasi ini dapat digunakan untuk menginformasikan strategi pemasaran, meningkatkan layanan pelanggan, dan mengelola reputasi merek [28].

Terdapat beberapa pendekatan utama dalam analisis sentimen [28]:

1. Pendekatan berbasis pengetahuan dimana metode ini mengklasifikasikan teks berdasarkan kategori afektif dengan mendeteksi kata-kata yang secara eksplisit mengekspresikan emosi, seperti “senang”, “sedih”, atau “takut”.
2. Metode statistik dimana pendekatan ini memanfaatkan teknik pembelajaran mesin, seperti analisis semantik laten dan model ruang semantik, untuk mengidentifikasi pola dalam data teks yang menunjukkan sentimen tertentu.
3. Pendekatan hibrida dimana metode ini menggabungkan teknik pembelajaran mesin dengan elemen representasi pengetahuan, seperti ontologi dan jaringan semantik, untuk mendeteksi makna yang diekspresikan secara implisit dalam teks.

Analisis sentimen pada media sosial dapat dilakukan secara manual, namun cara tersebut kurang efisien ketika jumlah data sangat besar, sehingga diperlukan metode otomatis untuk mempercepat proses analisis. Dengan bantuan sistem, berbagai unggahan di media sosial dapat dikumpulkan dan dianalisis, termasuk teks yang menyebutkan suatu bisnis tanpa harus diberi tag secara langsung, kemudian diklasifikasikan ke dalam sentimen positif, negatif, atau netral. Hasil analisis ini dapat dimanfaatkan sebagai dasar dalam pengambilan keputusan, khususnya dalam evaluasi dan strategi pemasaran. Salah satu metode yang umum digunakan adalah pendekatan berbasis leksikon (*lexicon-based*), yaitu metode yang menggunakan daftar kata yang telah memiliki nilai sentimen tertentu untuk menentukan kecenderungan opini dalam suatu teks, sehingga analisis dapat dilakukan secara lebih sederhana tanpa memerlukan data pelatihan [28].

2.5.1 Analisis Sentimen Berbasis *Lexicon*

Analisis sentimen berbasis *lexicon* merupakan salah satu pendekatan dalam pengolahan teks yang menggunakan daftar kata (kamus lexicon) yang telah diberi nilai polaritas sentimen secara manual, seperti positif, negatif, atau netral. Pendekatan ini bekerja dengan cara mencocokkan setiap kata dalam teks dengan kamus yang tersedia, kemudian menghitung skor keseluruhan berdasarkan akumulasi nilai sentimen dari kata-kata yang ditemukan. Hasil akhir dari proses ini digunakan untuk menentukan kecenderungan sentimen suatu teks secara keseluruhan [29].

Keunggulan utama dari pendekatan berbasis *lexicon* adalah tidak memerlukan proses pelatihan data seperti pada metode *Machine Learning*, sehingga lebih sederhana dan cepat dalam implementasinya. Selain itu, metode ini juga dapat digunakan pada kondisi data yang terbatas karena hanya bergantung pada kamus sentimen yang telah tersedia. Namun, pendekatan ini memiliki keterbatasan dalam memahami konteks kalimat, seperti kata yang memiliki makna berbeda tergantung penggunaannya, serta sulit menangani bahasa informal atau slang yang sering muncul pada data media sosia [29].

Dalam implementasinya, analisis sentimen berbasis lexicon dapat menggunakan berbagai jenis kamus sentimen yang telah dikembangkan untuk mendukung proses perhitungan polaritas kata secara lebih akurat. Setiap kamus *lexicon* memiliki karakteristik dan cakupan kata yang berbeda, sehingga pemilihan lexicon yang tepat sangat berpengaruh terhadap hasil analisis sentimen yang diperoleh. Salah satu bentuk lexicon yang banyak digunakan dalam analisis data teks, khususnya pada data media sosial, adalah *VADER Lexicon* yang dirancang untuk menangani bahasa informal serta memberikan hasil analisis yang lebih adaptif terhadap konteks teks yang dianalisis.

2.5.2 VADER Lexicon

Salah satu alat analisis dari *lexicon based* adalah*(Valance Dictionary and Sentiment Reasoner (VADER)*. *VADER Lexicon* digunakan untuk menganalisis data berdasarkan *Lexicon* (kamus). Hasil analisisnya berupa kelas polaritas positif, netral, dan negatif dengan tambahan *compound score* atau skor total. *VADER Lexicon* memiliki 7.500 kata yang didalamnya terdapat sentimen yang terkait dengan sinonim dan akronim serta kata berbahasa Inggris [30].

Leksikal merupakan kamus yang digunakan sebagai bahasa pokok dalam metode *lexicon based*. Untuk mendeteksi klasifikasi atau sentimen, dilakukan dengan menghitung skor *compound* melalui rumus [31]:

$$Compound\ Score = \frac{\sum_{i=1}^{n} Si}{\sqrt{(\sum_{i=1}^{n} S_i\ )^2 \alpha}} \quad \text{.....(13)}$$

Dimana:

Si adalah skor sentimen untuk kata ke-i dalam teks.

n adalah jumlah kata yang dianalisis.

Skor sentimen untuk setiap kata Si diperoleh dari Vader Lexicon, yang memberikan nilai sentimen berdasarkan konteks kata tersebut.

α adalah konstanta *smoothing* empiris pada algoritma VADER yang digunakan untuk menstabilkan hasil normalisasi *compound score* biasanya bernilai 15.

Untuk proses klasifikasi sentimen dapat dilakukan dengan persamaan 14 berikut [32].

$$Sentiment = \begin{cases} positif\ if\ compound > 0{,}05 \\ netral\ if\ compound - 0{,}05 \leq compound \leq 0{,}05 \\ negatif\ if\ compound < -0{,}05 \end{cases} \quad \text{.....(14)}$$

2.6 Multilabel Classification

Multilabel classification adalah sebuah masalah klasifikasi dimana setiap sampel data dapat diklasifikasikan ke dalam lebih dari satu kelas pada saat yang sama. Terdapat beberapa cara untuk menyelesaikan masalah *multilabel classification*, yang secara umum dapat dibagi menjadi dua kategori utama, yaitu metode *Algorithm Adaptation* (mengadaptasi algoritma agar bisa menangani multilabel secara alami) dan metode *Problem Transformation* (mengubah masalah multilabel menjadi beberapa masalah klasifikasi biner standar). Salah satu pendekatan yang paling populer dan banyak digunakan dalam kategori *Problem Transformation* adalah pendekatan *One-vs-Rest* (OvR) atau yang sering disebut juga sebagai *Binary Relevance* [33].

One-versus-Rest (OvR) digunakan untuk menyelesaikan masalah *multilabel classification*. Pendekatan ini juga dikenal sebagai *One-versus-All* (OvA). Pada pendekatan OvR, setiap kelas atau label dianggap sebagai label positif, dan label-label lainnya dianggap sebagai label negatif. Sebuah model klasifikasi biner dipelajari untuk setiap label positif dengan membedakan sampel yang termasuk dalam kelas positif tersebut dari sampel yang termasuk dalam kelas negatif [33].

Setelah model klasifikasi multilabel dibangun menggunakan pendekatan *One-vs-Rest*, langkah krusial berikutnya adalah mengukur sejauh mana model tersebut mampu memprediksi label dengan akurat. Untuk mengevaluasi performa dari setiap model biner

yang terbentuk tersebut, digunakan sebuah matriks evaluasi yang dikenal sebagai *Confusion Matrix.*

2.7 Confusion Matrix

Pada penelitian ini metode evaluasi sistem yang digunakan yaitu *Confusion Matrix. Confusion Matrix* adalah salah satu cara yang sering digunakan pada proses evaluasi model data mining klasifikasi dengan memprediksi kebenaran objek. *Confusion Matrix* digambarkan dengan tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan. Berikut ini akan ditunjukkan tabel *Confusion Matrix* pada Tabel 2.1 [34].

Tabel 2.1 *Confusion Matrix* Untuk Klasifikasi 2 Kelas [33]

Correct Classification	Classified as	
	Predicted "+"	*Predicted* "-"
Actual "+"	*True Positives*	*False Negatives*
Actual "-"	*False Positives*	*True Negatives*

Berdasarkan tabel *Confusion Matrix* di atas [34]:

1. *True Positives* (TP) adalah jumlah *record* data positif yang diklasifikasikan sebagai nilai positif.
2. *False Positives* (FP) adalah jumlah *record* data negatif yang diklasifikasikan sebagai nilai positif.
3. *False Negatives* (FN) adalah jumlah *record* data positif yang diklasifikasikan sebagai nilai negatif.
4. *True Negatives* (TN) adalah jumlah *record* data negatif yang diklasifikasikan sebagai nilai *negative.*

Nilai yang dihasilkan melalui metode *Confusion Matrix* adalah berupa evaluasi sebagai berikut [34]:

1. *Accuracy* merupakan ukuran yang menunjukkan seberapa banyak data yang berhasil diklasifikasikan dengan benar oleh model dibandingkan dengan seluruh data. Semakin tinggi nilai *accuracy*, maka semakin baik kinerja model secara keseluruhan karena sebagian besar data berhasil diprediksi dengan tepat. Sebaliknya, semakin rendah nilai *accuracy* menunjukkan bahwa model masih banyak melakukan kesalahan dalam klasifikasi.

$$Accuracy = \frac{(\mathrm{TP}+\mathrm{TN})}{\mathrm{Total\ Data}} \quad \text{.....(15)}$$

2. *Precision* menunjukkan tingkat ketepatan model dalam memprediksi kelas positif, yaitu seberapa banyak prediksi positif yang benar dibandingkan dengan seluruh prediksi positif. Semakin tinggi nilai precision, maka semakin sedikit kesalahan dalam bentuk *false positive*, sehingga model lebih akurat dalam menentukan data yang benar-benar positif. Sebaliknya, nilai *precision* yang rendah menunjukkan bahwa banyak data yang diprediksi positif ternyata salah.

$$Precision = \frac{(\mathrm{TP})}{(\mathrm{TP+FP})} \quad \text{.....(16)}$$

3. *Recall* mengukur kemampuan model dalam menemukan seluruh data yang benar-benar positif. Semakin tinggi nilai *recall*, maka semakin baik model dalam mendeteksi data positif dan semakin sedikit *false negative* yang terjadi. Sebaliknya, nilai *recall* yang rendah menunjukkan bahwa masih banyak data positif yang tidak berhasil terdeteksi oleh model.

$$Recall = \frac{(\mathrm{TP})}{(\mathrm{TP+FN})} \quad \text{.....(17)}$$

4. *F1-score* mengukur keseimbangan antara *precision* dan *recall*, sehingga memberikan gambaran menyeluruh mengenai kinerja model klasifikasi, terutama pada dataset yang tidak seimbang.

$$F1 - score = 2 \text{ x } \frac{\text{Precision x Recall}}{\text{Precision+Recall}} \quad \text{.....(18)}$$

5. *Misclassification (Error) Rate* menunjukkan persentase data yang salah diklasifikasikan oleh model. Semakin rendah nilai *error rate*, maka semakin baik performa model karena kesalahan prediksi semakin sedikit. Sebaliknya, jika nilai *error rate* tinggi, maka menunjukkan bahwa model masih kurang akurat dalam melakukan klasifikasi.

$$Misclassification\ (Error) Rate = 1 - Accuracy \quad \text{.....(19)}$$

6. *Support* menunjukkan jumlah data aktual pada setiap kelas yang digunakan dalam proses evaluasi model. Nilai support digunakan sebagai dasar dalam menghitung rata-rata berbobot *(*weighted average*)* dan memberikan informasi mengenai distribusi data pada masing-masing kelas.
7. *Macro average* merupakan nilai rata-rata *precision*, *recall*, dan *F1-score* dari seluruh kelas dengan memberikan bobot yang sama pada setiap kelas, tanpa mempertimbangkan jumlah data pada masing-masing kelas.
8. *Weighted average* merupakan nilai rata-rata *precision*, *recall*, dan *F1-score* yang dihitung dengan mempertimbangkan jumlah data (*support*) pada setiap kelas, sehingga

memberikan gambaran performa model yang lebih sesuai dengan distribusi data yang sebenarnya.