BAB 2
LANDASAN TEORI

2.1 Emas sebagai Instrumen Investasi

Emas merupakan salah satu logam mulia yang banyak digunakan sebagai instrumen investasi karena dinilai mampu mempertahankan nilai aset dalam jangka panjang [19]. Berbeda dengan mata uang yang rentan terhadap inflasi, emas cenderung memiliki nilai yang lebih stabil sehingga sering dijadikan sebagai *safe haven* pada kondisi ketidakpastian ekonomi [20]. Selain itu, emas memiliki likuiditas yang tinggi dan mudah diperjualbelikan, baik dalam bentuk fisik maupun digital [21]. Karakteristik tersebut menjadikan emas sebagai salah satu pilihan investasi yang diminati masyarakat.

Seiring dengan meningkatnya minat masyarakat terhadap investasi emas, berbagai opini dan persepsi terkait emas banyak disampaikan melalui media sosial dan platform digital lainnya. Opini tersebut dapat mencerminkan sentimen publik yang berpotensi memengaruhi keputusan investasi. Oleh karena itu, diperlukan suatu metode untuk menganalisis sentimen masyarakat terhadap emas secara otomatis, salah satunya melalui pendekatan analisis sentimen berbasis *Natural Language Processing* (NLP).

2.2 Platform X sebagai Sumber Data

Platform X adalah salah satu media sosial yang memberikan fasilitas bagi penggunanya untuk berbagi informasi, opini, serta pengalaman pribadi dalam bentuk unggahan teks. Karakteristik utama platform ini adalah format pesan yang relatif singkat sehingga pengguna dapat menyampaikan pendapat dengan cepat dan langsung [22]. Selain itu, ketersediaan data dalam skala besar dan sifatnya yang berlangsung secara *real-time* membuat platform X kerap digunakan sebagai sumber data dalam beragam penelitian analisis sentimen [23].

Penelitian ini memanfaatkan latform X sebagai sumber data untuk menggali opini masyarakat terhadap investasi emas. Pengumpulan data dilakukan melalui proses *scraping* menggunakan *Tweet Harvest* dengan memanfaatkan sejumlah kata kunci yang relevan terhadap topik penelitian. Data yang berhasil terkumpul kemudian dilakukan analisis untuk mengetahui kecenderungan sentimen publik terhadap investasi emas.

2.3 Analisis Sentimen

Analisis sentimen adalah salah satu penerapan *Natural Language Processing* yang berfokus pada pengenalan dan pemaknaan pandangan, pendirian, maupun emosi yang terkandung dalam data berbentuk teks [24]. Melalui teknik ini, informasi yang bersifat subjektif dapat diolah untuk memahami pandangan seseorang terhadap suatu objek, layanan, produk, individu, maupun peristiwa tertentu [25].

Kemampuan analisis sentimen dalam memproses data teks berskala besar memungkinkan diperolehnya gambaran umum mengenai kecenderungan opini masyarakat secara lebih sistematis [26]. Pada penelitian yang memanfaatkan data media sosial, Analisis sentimen umumnya diterapkan untuk membagi teks menjadi kategori positif, negatif, dan netral sehingga persepsi publik terhadap suatu topik dapat dianalisis dengan lebih mudah [27].

2.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah salah satu cabang kecerdasan buatan yang berkonsentrasi pada kemampuan komputer untuk memahami mengolah, serta menganalisis bahasa manusia, baik dalam bentuk tulisan maupun lisan. NLP berfungsi dalam mengubah unsur kebahasaan yang semula tidak terstruktur menjadi informasi yang lebih terorganisasi sehingga dapat diproses oleh sistem komputer [28].

Dalam penelitian analisis sentimen, NLP menjadi komponen penting karena mendukung proses pengolahan data sebelum tahap klasifikasi dilakukan. Berbagai teknik seperti *preprocessing*, tokenisasi, dan representasi teks diterapkan untuk meningkatkan kualitas data sehingga algoritma *machine learning* dapat menghasilkan performa yang lebih baik dalam mengenali sentimen [29].

Pemanfaatan NLP pada data media sosial juga cukup relevan karena mampu menangani variasi bahasa yang beragam, termasuk penggunaan singkatan, bahasa tidak baku, maupun istilah populer yang sering ditemukan dalam komunikasi digital. Oleh sebab itu, NLP menjadi fondasi utama dalam proses analisis sentimen berbasis teks.

2.5 Cohen's Kappa

Cohen's Kappa adalah suatu indikator statistik yang menilai tingkat kesepakatan (*agreement*) antara dua anotator dalam memberikan label terhadap data yang sama, dengan memperhatikan kemungkinan adanya kesepakatan akibat faktor kebetulan (*chance agreement*) [30]. Metode ini banyak digunakan dalam proses validasi anotasi data, termasuk pada tugas klasifikasi teks dan analisis sentimen, untuk memastikan bahwa label yang diberikan bersifat konsisten dan tidak terlalu dipengaruhi oleh subjektivitas individu anotator [31].

Berbeda dengan ukuran kesepakatan sederhana seperti persentase kesepakatan (*percentage agreement*), Cohen's Kappa mengoreksi nilai kesepakatan yang diamati dengan nilai kesepakatan yang diharapkan terjadi secara kebetulan berdasarkan distribusi label masing-masing anotator. Hal ini membuat Cohen's Kappa lebih representatif dalam mengukur reliabilitas anotasi, khususnya pada data yang memiliki distribusi kelas tidak seimbang [32]

Rumus 2.1 menunjukkan perhitungan nilai Cohen's kappa.

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{2.1}$$

Keterangan:

- κ = nilai Cohen's Kappa
- P_o = proporsi kesepakatan yang diamati (*observed agreement*) antara dua anotator
- P_e = proporsi kesepakatan yang diharapkan terjadi secara kebetulan (*expected agreement*)

Nilai P_e dihitung berdasarkan distribusi marginal label yang diberikan oleh masing-masing anotator [33], sebagaimana ditunjukkan pada Rumus 2.2.

$$P_e = \sum_{i=1}^{k} \left(\frac{n_i^A}{N} \times \frac{n_i^B}{N} \right) \tag{2.2}$$

Keterangan:

- k = jumlah kategori atau kelas label
- n_i^A = jumlah data yang diberi label kelas i oleh anotator A

- n_i^B = jumlah data yang diberi label kelas i oleh anotator B
- N = jumlah total data yang dianotasi

Rentang nilai κ berada di antara -1 sampai 1, di mana nilai 1 menunjukkan kesepakatan sempurna, nilai 0 berarti kesepakatan setara kebetulan, dan nilai negatif berarti kesepakatan yang lebih rendah dibandingkan kebetulan. Tabel 2.1 menunjukkan interpretasi nilai Cohen's Kappa berdasarkan pedoman konvensional yang umum digunakan dalam berbagai penelitian.

Tabel 2.1. Interpretasi nilai Cohen's kappa

Nilai κ	**Tingkat Kesepakatan**
$< 0,00$	Tidak ada kesepakatan (*poor*)
$0,00 - 0,20$	Kesepakatan rendah (*slight*)
$0,21 - 0,40$	Kesepakatan cukup (*fair*)
$0,41 - 0,60$	Kesepakatan moderat (*moderate*)
$0,61 - 0,80$	Kesepakatan substansial (*substantial*)
$0,81 - 1,00$	Kesepakatan hampir sempurna (*almost perfect*)

Dalam praktiknya, pengukuran *inter-annotator agreement* menggunakan Cohen's Kappa umumnya tidak dilakukan terhadap seluruh data, melainkan terhadap subset data yang dipilih secara acak (*random sampling*) sebagai sampel representatif untuk efisiensi proses validasi [34].

2.6 Text Preprocessing

Pada umumnya, data yang diperoleh dari media sosial tidak dapat langsung diolah dalam proses analisis karena masih terdapat berbagai elemen yang kurang relevan, seperti singkatan, simbol, karakter khusus, serta bentuk penulisan yang tidak konsisten. Oleh sebab itu, diperlukan tahapan *text preprocessing* untuk menyiapkan data sehingga memiliki kualitas yang lebih baik sebelum digunakan dalam proses pembelajaran model *machine learning* [35].

Proses *preprocessing* dalam penelitian ini terdiri dari beberapa tahap, yakni *case folding*, *cleaning*, *tokenizing*, *normalization*, *stopword removal*, serta *stemming*.

2.6.1 Case Folding

Tahap *case folding* dilakukan dengan menyeragamkan seluruh karakter alfabet menjadi huruf kecil. Proses ini bertujuan agar kata yang memiliki perbedaan penggunaan huruf kapital tetap dikenali sebagai kata yang sama selama proses pengolahan data [36].

2.6.2 Cleaning

Pada tahap ini, berbagai komponen yang tidak memberikan informasi penting terhadap analisis dihilangkan dari teks. Komponen tersebut meliputi URL, *mention*, *hashtag*, angka, maupun tanda baca sehingga data yang tersisa lebih relevan untuk diproses pada tahapan berikutnya [37].

2.6.3 Tokenizing

Setelah teks dibersihkan, kalimat dipecah menjadi sejumlah unit yang lebih kecil berupa kata atau token. Pemecahan ini bertujuan memudahkan komputer dalam mengenali dan mengolah setiap kata secara individual [38].

2.6.4 Normalization

Normalisasi dilakukan untuk menyeragamkan kata-kata yang memiliki variasi penulisan. Kata tidak baku, singkatan, serta istilah informal diubah ke bentuk bakunya agar representasi data menjadi lebih konsisten [39].

2.6.5 Stopword Removal

Tahap ini dilakukan dengan menghapus kata-kata yang sering muncul tetapi memiliki kontribusi informasi yang rendah terhadap proses klasifikasi. Contohnya adalah kata ”dan”, ”yang”, serta ”di” [40].

2.6.6 Stemming

Stemming bertujuan mengembalikan kata yang memiliki imbuhan ke bentuk dasarnya. Dengan cara ini, berbagai variasi suatu kata dapat direpresentasikan dalam bentuk yang sama sehingga proses analisis menjadi lebih konsisten [41].

2.7 GridSearchCV

GridSearchCV merupakan metode *hyperparameter tuning* yang digunakan untuk mencari kombinasi parameter terbaik dengan mengevaluasi seluruh kombinasi parameter dalam suatu *parameter grid* menggunakan teknik *cross-validation*. Kombinasi parameter dengan hasil evaluasi terbaik kemudian ditetapkan sebagai model optimal [42]. Penggunaan *hyperparameter tuning* melalui *grid search* dan *cross-validation* diketahui dapat meningkatkan proses pemilihan model serta membantu memperoleh konfigurasi parameter yang lebih optimal dibandingkan penggunaan parameter bawaan. Nilai k yang umum digunakan dalam praktik adalah $k = 5$ atau $k = 10$, di mana $k = 5$ dinilai lebih efisien secara komputasi dengan tetap memberikan estimasi performa yang stabil [43].

2.8 TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan teknik representasi teks yang mengubah data teks menjadi bentuk numerik untuk kebutuhan analisis. Pada metode ini, setiap kata diberi bobot berdasarkan frekuensi kemunculannya dalam dokumen tertentu serta frekuensi kemunculannya pada seluruh dokumen yang ada dalam *dataset*. Dengan demikian, kata yang sering muncul pada dokumen tertentu tetapi jarang ditemukan pada dokumen lain akan memperoleh bobot lebih tinggi [44].

Dalam analisis sentimen, TF-IDF banyak digunakan sebagai metode ekstraksi fitur karena mampu menonjolkan kata-kata yang lebih relevan dan mengurangi pengaruh kata-kata yang terlalu umum. Hal tersebut membantu model dalam mengenali pola pada data teks dengan lebih baik [45]. Perhitungan TF-IDF melibatkan komponen *Term Frequency* (TF), *Inverse Document Frequency* (IDF), dan kombinasi keduanya [46].

Rumus 2.3 menunjukkan perhitungan *Term Frequency* (TF).

$$TF(t,d) = \frac{\text{Jumlah kemunculan } t \text{ dalam } d}{\text{Total kata dalam } d} \tag{2.3}$$

Rumus 2.4 menunjukkan perhitungan *Inverse Document Frequency* (IDF).

$$IDF(t) = \log\left(\frac{\text{Jumlah } d}{\text{Jumlah } d \text{ yang memuat } t}\right) \tag{2.4}$$

Rumus 2.5 menunjukkan perhitungan TF-IDF sebagai proses pembobotan kata.

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t) \tag{2.5}$$

Keterangan:

- t = kata
- d = dokumen
- $TF(t,d)$ = frekuensi kemunculan kata t dalam dokumen d
- $IDF(t)$ = nilai *Inverse Document Frequency* dari kata t

Contoh Perhitungan:

Misalkan terdapat 3 dokumen sederhana berupa komentar terkait investasi emas sebagai berikut:

- d_1: ”investasi emas sangat menguntungkan”
- d_2: ”harga emas hari ini naik”
- d_3: ”investasi emas aman dan menguntungkan”

Akan dihitung nilai TF-IDF untuk kata ”menguntungkan” pada dokumen d_1. Kata ”menguntungkan” muncul 1 kali dari 4 kata pada d_1, dan muncul pada 2 dari 3 dokumen (yaitu d_1 dan d_3). Berdasarkan Rumus 2.3, 2.4, dan 2.5, diperoleh perhitungan sebagai berikut:

Tabel 2.2. Contoh perhitungan TF-IDF untuk kata ”menguntungkan”

Perhitungan	**Hasil**
$TF(\text{menguntungkan}, d_1) = 1/4$	0.25
$IDF(\text{menguntungkan}) = \log(3/2)$	0.176
$TF\text{-}IDF(\text{menguntungkan}, d_1) = 0.25 \times 0.176$	0.044

Nilai TF-IDF sebesar 0.044 menunjukkan bobot kata ”menguntungkan” pada dokumen d_1. Semakin tinggi nilai TF-IDF suatu kata, semakin penting kata tersebut dalam merepresentasikan isi dokumen dibandingkan dokumen lain dalam *dataset*.

2.9 SMOTE

Synthetic Minority Over-sampling Technique (SMOTE) merupakan teknik penyeimbangan data yang digunakan ketika jumlah data pada masing-masing kelas tidak proporsional. Cara kerja metode ini yaitu membangun sampel sintetis baru pada kelas minoritas berdasarkan kedekatan karakteristik data, sehingga distribusi antar kelas menjadi lebih seimbang [47].

Dengan distribusi data yang lebih merata, model memiliki peluang yang lebih baik untuk mempelajari pola pada kelas minoritas dan tidak terlalu terfokus pada kelas yang jumlah datanya lebih besar. Oleh karena itu, SMOTE banyak diterapkan pada permasalahan klasifikasi yang memiliki ketidakseimbangan kelas, termasuk analisis sentimen. Penggunaan teknik ini dapat membantu meningkatkan kemampuan model dalam mengenali setiap kategori sentimen secara lebih optimal [48].

2.10 Naive Bayes

Naive Bayes merupakan metode klasifikasi berbasis probabilitas yang dibangun berdasarkan Teorema Bayes. Teorema Bayes digunakan untuk menghitung probabilitas suatu kelas berdasarkan informasi atau fitur yang diamati. Dalam proses klasifikasi, probabilitas tersebut digunakan untuk menentukan kemungkinan suatu data termasuk ke dalam kelas tertentu berdasarkan karakteristik yang dimilikinya [49]. Persamaan dasar Teorema Bayes ditunjukkan pada Rumus 2.6.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{2.6}$$

Keterangan:

- A = hipotesis atau kelas yang akan diprediksi
- B = data atau fitur yang diamati
- $P(A|B)$ = probabilitas kelas A berdasarkan data B (*posterior*)
- $P(A)$ = probabilitas awal suatu kelas (*prior*)
- $P(B|A)$ = probabilitas kemunculan fitur B pada kelas A (*likelihood*)

- $P(B)$ = probabilitas data yang diamati (*evidence*)

Pada proses klasifikasi, Naive Bayes menghitung probabilitas setiap kelas berdasarkan fitur yang terdapat pada suatu data. Tahap pertama dilakukan dengan menghitung probabilitas awal (*prior*) dari setiap kelas. Selanjutnya, model menghitung probabilitas kemunculan fitur terhadap masing-masing kelas (*likelihood*). Nilai tersebut kemudian dikombinasikan untuk memperoleh probabilitas akhir (*posterior*). Kelas dengan nilai probabilitas posterior terbesar akan dipilih sebagai hasil prediksi.

Contoh Perhitungan:

Misalkan terdapat data latih sederhana berupa komentar terkait investasi emas dengan dua kelas sentimen, yaitu positif dan negatif sebagai berikut:

- Positif: ”emas aman”
- Positif: ”investasi emas menguntungkan”
- Negatif: ”harga emas turun”

Dokumen baru yang akan diklasifikasikan adalah ”emas aman”. Berdasarkan data latih tersebut, terdapat dua dokumen dengan kelas positif dan satu dokumen dengan kelas negatif. Nilai *prior* masing-masing kelas dihitung sebagai berikut:

$$P(Positif) = \frac{2}{3} = 0.667 \tag{2.7}$$

$$P(Negatif) = \frac{1}{3} = 0.333 \tag{2.8}$$

Selanjutnya, dihitung nilai *likelihood* dari fitur kata yang terdapat pada dokumen baru, yaitu kata ”emas” dan ”aman”. Misalkan berdasarkan data latih diperoleh probabilitas kemunculan kata sebagai berikut:

- $P(\text{emas}|Positif) = \frac{2}{5} = 0.4$
- $P(\text{aman}|Positif) = \frac{1}{5} = 0.2$
- $P(\text{emas}|Negatif) = \frac{1}{3} = 0.333$
- $P(\text{aman}|Negatif) = \frac{0}{3} = 0$

Berdasarkan Teorema Bayes, nilai probabilitas posterior setiap kelas dihitung dengan mengalikan nilai *prior* dan *likelihood* dari setiap fitur. Perhitungan untuk kelas positif adalah sebagai berikut:

$$P(Positif|d) = 0.667 \times 0.4 \times 0.2 = 0.053 \tag{2.9}$$

Sedangkan perhitungan untuk kelas negatif adalah:

$$P(Negatif|d) = 0.333 \times 0.333 \times 0 = 0 \tag{2.10}$$

Berdasarkan hasil perhitungan tersebut, kelas positif memiliki nilai probabilitas posterior yang lebih besar dibandingkan kelas negatif. Oleh karena itu, dokumen baru ”emas aman” diklasifikasikan sebagai sentimen positif.

Pada implementasi nyata, Multinomial Naive Bayes menerapkan Laplace smoothing untuk mengatasi nilai probabilitas nol pada fitur yang tidak ditemukan pada suatu kelas. Selain itu, pada klasifikasi teks, nilai fitur yang digunakan bukan berupa frekuensi kata secara langsung, melainkan representasi numerik hasil ekstraksi fitur seperti TF-IDF.

Nama ”Naive Bayes” berasal dari asumsi sederhana (*naive*) bahwa setiap fitur dianggap tidak memiliki keterkaitan atau bersifat independen terhadap fitur lainnya dalam menentukan suatu kelas. Meskipun pada data teks beberapa kata dapat memiliki hubungan satu sama lain, asumsi independensi tersebut membuat proses klasifikasi menjadi lebih sederhana dan efisien [50].

Secara umum, Naive Bayes memiliki beberapa varian utama, yaitu Gaussian Naive Bayes, Multinomial Naive Bayes, dan Bernoulli Naive Bayes. Gaussian Naive Bayes digunakan untuk data numerik dengan distribusi kontinu. Multinomial Naive Bayes digunakan pada data diskrit seperti jumlah atau bobot kemunculan kata pada dokumen teks. Sementara itu, Bernoulli Naive Bayes digunakan pada fitur biner yang menunjukkan keberadaan atau ketiadaan suatu fitur [51].

Dalam penelitian analisis sentimen, Naive Bayes banyak digunakan karena memiliki proses perhitungan yang sederhana, efisien, serta mampu menghasilkan performa yang baik pada data teks [52]. Penelitian ini menggunakan Multinomial Naive Bayes karena data penelitian berupa teks yang telah direpresentasikan dalam bentuk vektor fitur menggunakan TF-IDF. Multinomial Naive Bayes sesuai untuk klasifikasi dokumen teks dengan fitur berdimensi tinggi karena mampu menghitung probabilitas kelas berdasarkan bobot fitur kata yang terdapat pada dokumen [53].

2.11 Multinomial Naive Bayes

Multinomial Naive Bayes merupakan varian dari algoritma Naive Bayes yang banyak diterapkan pada klasifikasi teks karena memiliki proses perhitungan yang efisien dan mampu memberikan hasil yang cukup akurat [53]. Metode ini menggunakan jumlah kemunculan kata dalam dokumen sebagai dasar pembentukan fitur, di mana setiap kata direpresentasikan dalam bentuk nilai frekuensi seperti 0, 1, 2, dan seterusnya. Pendekatan ini mengasumsikan bahwa data mengikuti distribusi multinomial, sehingga setiap kata dalam dokumen dianggap sebagai bagian dari distribusi tersebut [54]. Dengan memanfaatkan distribusi frekuensi kata, Multinomial Naive Bayes dapat menentukan probabilitas suatu dokumen termasuk ke dalam kelas tertentu, sehingga efektif digunakan dalam tugas analisis sentimen [55].

Pada Multinomial Naive Bayes, probabilitas suatu dokumen d termasuk ke dalam kelas c dihitung dengan mengalikan probabilitas awal kelas (*prior*) dengan probabilitas kemunculan setiap fitur atau kata pada dokumen (*likelihood*) [56]. Persamaan Multinomial Naive Bayes ditunjukkan pada Persamaan 2.11.

$$P(c|d) \propto P(c) \prod_{i=1}^{n} P(w_i|c)^{f_i} \tag{2.11}$$

Keterangan:

- c = kelas yang diprediksi
- d = dokumen yang akan diklasifikasikan
- $P(c)$ = probabilitas awal suatu kelas (*prior*)
- w_i = kata ke-i pada dokumen
- $P(w_i|c)$ = probabilitas kemunculan kata w_i pada kelas c
- f_i = frekuensi atau bobot fitur kata ke-i pada dokumen
- n = jumlah fitur atau kata pada dokumen

Persamaan tersebut menunjukkan bahwa probabilitas suatu dokumen diperoleh dari hasil perkalian probabilitas awal kelas dengan probabilitas setiap kata yang muncul pada dokumen. Nilai probabilitas setiap kata dipangkatkan dengan

frekuensi atau bobot fitur yang dimiliki kata tersebut. Pada penelitian ini, bobot fitur diperoleh dari hasil pembobotan TF-IDF sehingga nilai f_i merepresentasikan bobot TF-IDF setiap kata [57].

Persamaan 2.6 merupakan bentuk dasar Teorema Bayes yang menghitung probabilitas posterior suatu kelas berdasarkan data yang diamati melalui komponen *prior*, *likelihood*, dan *evidence*, sehingga data masih direpresentasikan sebagai satu kesatuan dalam bentuk $P(B|A)$. Sementara itu, Persamaan 2.11 merupakan pengembangan Teorema Bayes yang digunakan pada algoritma Multinomial Naive Bayes untuk klasifikasi teks. Perbedaannya terletak pada komponen *likelihood*, di mana data diuraikan menjadi sekumpulan fitur berupa kata-kata (w_i) yang diasumsikan saling independen. Oleh karena itu, probabilitas dokumen dihitung melalui hasil perkalian probabilitas setiap kata terhadap kelas tertentu, dengan setiap probabilitas dipangkatkan menggunakan frekuensi atau bobot fitur (f_i). Dengan pendekatan tersebut, kata yang memiliki frekuensi atau bobot lebih besar akan memberikan kontribusi yang lebih besar terhadap hasil klasifikasi [56].

2.12 Evaluasi Model

Confusion Matrix merupakan salah satu metode yang digunakan untuk mengevaluasi kinerja model klasifikasi melalui perbandingan antara hasil prediksi model dan label sebenarnya [58]. Melalui penyajian hasil klasifikasi dalam bentuk matriks, performa model dapat dianalisis secara lebih rinci untuk mengetahui kemampuan model dalam mengidentifikasi setiap kelas yang tersedia. Selain digunakan pada klasifikasi biner, *Confusion Matrix* juga banyak diterapkan pada kasus klasifikasi multikelas sehingga evaluasi dapat dilakukan secara lebih menyeluruh [59].

Tabel 2.3. *Confusion matrix*

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

Sumber: [60]

Confusion Matrix memiliki empat unsur inti, yakni:

- *True Positive* (TP): data berlabel positif yang berhasil dikategorikan sebagai positif oleh model.

- *False Positive* (FP): data berlabel negatif tetapi dikategorikan sebagai positif oleh model.
- *False Negative* (FN): data berlabel positif tetapi dikategorikan sebagai negatif oleh model.
- *True Negative* (TN): data berlabel negatif yang berhasil dikategorikan sebagai negatif oleh model.

Berdasarkan nilai-nilai pada *Confusion Matrix*, kinerja model dapat diukur dengan menggunakan beberapa metrik evaluasi, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*.Setiap metrik tersebut memberikan perspektif yang berbeda dalam mengukur kemampuan model klasifikasi.

Accuracy digunakan untuk menunjukkan proporsi prediksi yang benar terhadap seluruh data yang dievaluasi. Rumus *accuracy* dapat dituliskan sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.12}$$

Precision menggambarkan tingkat ketepatan model ketika memprediksi suatu kelas berdasarkan seluruh prediksi yang dihasilkan pada kelas tersebut. Perhitungan *precision* dapat dilakukan menggunakan rumus berikut:

$$Precision = \frac{TP}{TP + FP} \tag{2.13}$$

Recall menunjukkan kemampuan model dalam menemukan seluruh data yang benar-benar termasuk ke dalam suatu kelas tertentu. Rumus *recall* adalah sebagai berikut:

$$Recall = \frac{TP}{TP + FN} \tag{2.14}$$

Sementara itu, *F1-Score* merupakan ukuran yang mengombinasikan nilai *precision* dan *recall* sehingga dapat memberikan gambaran performa model yang lebih seimbang. Rumus *F1-Score* adalah sebagai berikut:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.15}$$