

BAB II

KAJIAN LITERATUR

2.1 Musik dan Genre Musik

Musik dikenal sejak kehadiran manusia modern Homo sapiens yakni sekitar 180.000 hingga 100.000 tahun yang lalu. Tidak ada yang tahu kapan manusia mulai mengenal seni dan musik. Musik merupakan bunyi yang diterima oleh individu dan berbeda-beda berdasarkan sejarah, lokasi, budaya dan selera individu [10]. Definisi sejati tentang musik juga bermacam-macam di antaranya bahwa musik adalah bunyi/kesan terhadap sesuatu yang ditangkap oleh indera pendengar, musik adalah suatu karya seni dengan segenap unsur pokok dan pendukungnya, dan musik adalah segala bunyi yang dihasilkan secara sengaja oleh seseorang atau oleh kelompok individu yang disajikan sebagai musik [11]. Dari beberapa definisi tersebut, maka musik merupakan segala bunyi yang dihasilkan manusia secara sengaja yang disajikan sebagai musik [12]

Musik adalah suatu yang membuahkan hasil karya seni, berupa bunyi berbentuk lagu atau komposisi yang mengungkapkan pikiran serta perasaan penciptanya lewat unsur-unsur pokok musik, yakni irama, melodi, harmoni, serta bentuk atau susunan lagu dan ekspresi sebagai satu kesatuan [13]. Genre musik adalah cara yang paling umum digunakan untuk mengorganisasi database musik digital. Musik yang berada pada genre yang sama biasanya memiliki kemiripan karakteristik tertentu yang terkait dengan instrumentasi, struktur ritme, dan pitch musik [13].

Genre Musik adalah pengklasifikasian beberapa gaya musik terhadap tipe tertentu. Mulai dari genre Pop, Rock, Jazz, Dangdut, dan masih banyak lagi. Oleh karena itu, mengembangkan model *machine learning* yang lebih baik yang secara otomatis mengklasifikasikan musik berdasarkan genre akan sangat bermanfaat bagi penyedia *streaming* musik online [14]. Manusia telah menciptakan dan memanfaatkan genre musik sebagai label untuk mengklasifikasikan dan mengkarakterisasikan musik. Karena genre musik berkembang melalui interaksi yang kompleks dari pengaruh historis, budaya, pemasaran, dan publik, genre musik tidak memiliki definisi dan batasan yang jelas. Anggota genre tertentu memiliki kemiripan satu sama lain dalam hal instrumentasi musik, struktur ritme, dan konten nada [15].

2.2 Audio Digital

Audio digital merupakan representasi sinyal suara dalam bentuk data numerik yang diperoleh melalui proses *sampling* dan nilai suara menjadi digital. Untuk menyampaikan informasi penting dari sinyal suara atau musik, fitur audio adalah ciri atau kualitas yang diambil dari data audio. Pengenalan ucapan, identifikasi genre musik, dan deteksi emosi adalah beberapa aplikasi karakteristik ini dalam analisis data audio [16].

Dalam analisis audio, terdapat dua jenis fitur utama yang dapat diekstraksi, yaitu fitur berbasis waktu dan fitur berbasis frekuensi [17]. Fitur berbasis waktu meliputi *Zero Crossing Rate* (ZCR) dan energi sinyal, sedangkan fitur berbasis frekuensi meliputi spektral *centroid*, spektral *bandwidth*, serta *Mel-Frequency Cepstral Coefficients* (MFCC). Fitur-fitur ini digunakan untuk merepresentasikan karakteristik audio secara komprehensif sehingga dapat digunakan dalam proses klasifikasi [18].

2.3 Ekstraksi Fitur Audio

Tahapan ekstraksi fitur musik dilakukan dengan pemrosesan sinyal digital dalam domain frekuensi audio untuk menganalisis, memanipulasi, dan meningkatkan sinyal audio dalam domain digital. Salah satu teknik yang digunakan dalam ekstraksi fitur adalah *Fast Fourier Transform*, yaitu algoritma untuk mengubah sinyal dari domain waktu ke domain frekuensi [19].

Untuk mengekstraksi fitur digunakan metode pemrosesan sinyal menggunakan *Mel Frequency Cepstral Coefficients* (MFCC) yang merupakan pengembangan dari analisis frekuensi yang menyesuaikan dengan persepsi pendengaran manusia melalui skala mel. Didalam MFCC terdapat *Fast Fourier Transform* (FFT). FFT merupakan salah satu metode untuk mengubah sinyal audio dari domain waktu ke domain frekuensi sehingga dapat dianalisis komponen frekuensinya. FFT menghasilkan spektrum frekuensi yang masih bersifat mentah [20]. MFCC menggunakan hasil dari FFT sebagai input, kemudian diproses lebih lanjut untuk menghasilkan fitur yang lebih representatif. Dengan demikian, FFT merupakan tahap awal dalam ekstraksi fitur, sedangkan MFCC merupakan fitur akhir yang digunakan dalam proses klasifikasi [21].

Selain menggunakan *Mel-Frequency Cepstral Coefficients* (MFCC), analisis audio juga memanfaatkan berbagai fitur dasar yang mampu merepresentasikan karakteristik sinyal secara lebih menyeluruh. Berikut adalah fitur fiturnya:

1. Tempo

Tempo merupakan fitur yang menggambarkan tingkat kecepatan ritme suatu musik yang dinyatakan dalam *Beats Per Minute* (BPM), dan digunakan untuk membedakan karakter ritmis antar genre musik.

2. Chroma STFT

Chroma STFT adalah fitur yang merepresentasikan distribusi 12 kelas nada dalam sinyal audio yang digunakan untuk menangkap informasi harmoni serta pola melodi musik.

3. RMS (*Root Mean Square*)

RMS merupakan ukuran yang digunakan untuk merepresentasikan energi atau intensitas sinyal audio yang menunjukkan tingkat keras atau lemahnya suara.

4. Spectral Centroid

Spectral centroid adalah fitur yang menunjukkan titik pusat distribusi frekuensi pada spektrum audio yang merepresentasikan tingkat kecerahan suara.

5. Spectral Bandwidth

Spectral bandwidth menggambarkan lebar penyebaran frekuensi dalam sinyal audio yang mencerminkan tingkat kompleksitas spektral suara.

6. Spectral Rolloff

Spectral rolloff merupakan fitur yang menunjukkan batas frekuensi di mana sebagian besar energi sinyal (sekitar 85–95%) terkonsentrasi.

7. Zero Crossing Rate (ZCR)

Zero Crossing Rate (ZCR) mengukur frekuensi perubahan sinyal dari nilai positif ke negatif yang digunakan untuk membedakan karakteristik suara berisik dan suara tonal.

MFCC cenderung berfokus pada aspek spektral atau timbre, sehingga belum sepenuhnya mampu menangkap informasi lain seperti ritme, energi, dan harmoni. Oleh karena itu, digunakan fitur tambahan seperti chroma STFT, *Root Mean Square* (RMS), *spectral centroid*, *spectral bandwidth*, *spectral rolloff*, *Zero Crossing Rate* (ZCR), serta tempo. Fitur-fitur tersebut termasuk dalam kategori *low-level* audio features yang dapat menggambarkan aspek energi, distribusi frekuensi, ritme, serta harmoni dari sinyal audio secara lebih komprehensif.[22]

2.4 Mel-Frequency Cepstrum Coefficients (MFCC)

MFCC adalah teknik ekstraksi fitur yang biasa digunakan dalam pemrosesan sinyal ucapan dan audio untuk mewakili karakteristik spektral sinyal ucapan. MFCC terdiri dari skala mel dan analisis cepstral. Skala mel adalah skala nada yang secara persepsi lebih

konsisten dengan pendengaran manusia dibandingkan skala frekuensi linier, dan analisis cepstral adalah teknik matematis yang digunakan untuk mewakili spektrum sinyal dengan cara yang menekankan informasi yang relevan secara kognitif [23].

Proses penghitungan MFCC melibatkan beberapa langkah yang berawal dari *Framing*, kemudian *Windowing*, *Fast Fourier Transform* (FFT), *Mel-Filtering*, *Logarithm*, dan berakhir dengan *Discrete Cosine Transform* (DCT) [24]. MFCC merupakan salah satu fitur yang paling umum digunakan dalam analisis sinyal audio karena mampu merepresentasikan karakteristik spektral suara yang meniru persepsi pendengaran manusia. Dalam analisis ini setiap file audio diekstraksi sebanyak 13 koefisien MFCC menggunakan pustaka *librosa*, yang mana ekstraksi dilakukan dengan parameter sebagai berikut:

- a. $n_fft = 2048$: jumlah sampel dalam setiap frame analisis (93 ms).
- b. $hop_length = 512$: langkah pergeseran antar frame (23 ms).
- c. $sr = 22050$: sampling rate untuk menjaga konsistensi resolusi waktu.

Ekstraksi MFCC menghasilkan matriks berdimensi (13, T) untuk setiap lagu, di mana T adalah jumlah *frame* yang dihasilkan dari pembagian sinyal berdasarkan *hop_length*. Karena jumlah frame yang berbeda-beda dan tergantung pada durasi, maka dilakukan proses agregasi untuk mendapatkan representasi yang bersifat statis dan tetap. Proses agregasi dilakukan dengan cara menghitung rata-rata tiap koefisien MFCC sepanjang waktu ($axis=1$) yang menghasilkan vektor berdimensi 13 untuk setiap lagunya. Langkah ini bertujuan untuk menyederhanakan struktur data dan membuatnya kompatibel dengan model klasifikasi berbasis data tabular. Dengan demikian, proses ekstraksi fitur ini menghasilkan matriks fitur akhir berdimensi (1000,13). Dimana 1000 adalah jumlah lagu dalam dataset dan 13 merupakan jumlah koefisien MFCC yang digunakan sebagai fitur representatif dari masing-masing lagu. Selain itu, penggunaan jumlah koefisien MFCC yang lebih dari 13 juga didukung oleh penelitian lain. Pada studi yang menggunakan dataset UrbanSound8K, setiap sampel audio diekstraksi menjadi 40 fitur MFCC untuk merepresentasikan karakteristik suara lingkungan sebelum dilakukan proses klasifikasi[22]. Ekstraksi dilakukan menggunakan fungsi *librosa.feature.mfcc()* dan *numpy.mean()* untuk proses perataan waktu [25].

Skala mel pada dasarnya adalah melakukan filter pada koefisien frekuensi yang didapatkan dari spektrogram. Filter yang dilakukan adalah linier pada frekuensi 1000Hz dan logaritmik pada frekuensi di atasnya formula (1) digunakan untuk menghitung mel-Frekuensi

$$M(f) = 2595 + \text{LOG} \left(1 + \left(\frac{f}{700} \right) \right) \dots \dots \dots (1)$$

Gelombang suara dari lagu tersebut akan dibuat 13 MFCC agar dapat membentuk matriks. Dimana pada sumbu Y menyatakan MFCC-nya dan sumbu X menyatakan durasi lagu yang ditentukan. Dari setiap blok yang terbentuk memiliki nilai-nilai koefisien. Dari gambar akan terbentuk 13 x 3000 pixel. Nilai 3000 sendiri ada setiap 1 Blok MFCC memiliki waktu yakni 20ms, sehingga 1 detiknya terdapat 50 pixel. Dikarenakan lagu berdurasi 60 detik sehingga terdapat 3000 pixels yang dapat dilihat pada 2.1 gambar berikut :



Gambar 2. 1 Model MFCC

Dalam menyimpan data menggunakan modul Json, data akan disimpan dalam sebuah file. Data dari lagu akan disimpan dan dibagi menjadi 3 bagian data yaitu: *Mapping*, MFCC dan Label [26].

Berikut ini merupakan tahapan dari MFCC:

1. Framing

Framing merupakan proses membagi sinyal audio yang kontinu menjadi segmen-segmen kecil dengan durasi singkat, biasanya sekitar 20–30 milidetik, agar sinyal dapat diasumsikan stabil dalam waktu tersebut. Proses ini digunakan karena karakteristik sinyal suara cenderung berubah seiring waktu, sehingga analisis dalam potongan kecil memungkinkan pengambilan informasi yang lebih akurat. Selain itu, antar frame biasanya dibuat saling tumpang tindih untuk menjaga kesinambungan informasi dan menghindari kehilangan data penting pada batas segmen[27].

2. Windowing

Windowing adalah tahap di mana setiap *frame* sinyal dikalikan dengan fungsi jendela tertentu, seperti *Hamming* atau efek perubahan mendadak untuk mengurangi efek

ketidaksinambungan pada ujung-ujung frame. Tanpa proses ini, pemotongan sinyal dapat menimbulkan gangguan pada analisis frekuensi berupa kebocoran spektrum. Dengan penerapan *windowing*, perubahan sinyal menjadi lebih halus sehingga hasil analisis spektrum menjadi lebih akurat dan stabil[27]

3. FFT (*Fast Fourier Transform*)

FFT digunakan untuk mentransformasikan sinyal dari domain waktu ke domain frekuensi sehingga komponen frekuensi dalam sinyal dapat diidentifikasi. Pada tahap ini, setiap frame yang telah melalui proses *windowing* diubah menjadi spektrum frekuensi yang menunjukkan distribusi energi pada berbagai frekuensi. Transformasi ini penting karena sebagian besar informasi penting dalam sinyal audio terletak pada karakteristik frekuensinya[28].

4. Mel Filter Bank

Mel Filter Bank berfungsi mengubah skala frekuensi linier menjadi skala Mel yang lebih sesuai dengan cara kerja pendengaran manusia. Filter tersebut disusun secara merata pada skala Mel dan umumnya memiliki overlap sekitar 50% antar filter yang berdekatan, sehingga mampu menangkap informasi frekuensi secara lebih halus dan sesuai dengan persepsi pendengaran manusia[29].

5. Logarithm

Tahap logaritma dilakukan dengan mengubah nilai energi dari hasil filter Mel ke dalam skala logaritmik. Hal ini bertujuan untuk menyesuaikan dengan sifat pendengaran manusia yang lebih peka terhadap perubahan relatif dibanding absolut. Selain itu, transformasi log juga membantu menyeimbangkan rentang nilai energi sehingga lebih mudah diproses pada tahap selanjutnya. Untuk merefleksikan kompresi logaritmik pada pendengaran manusia, keluaran filter bank biasanya diubah menggunakan fungsi logaritma.[30]

2. Discrete Cosine Transform (DCT)

DCT digunakan untuk mengubah hasil log Mel spectrum menjadi bentuk koefisien yang lebih ringkas dan saling tidak berkorelasi. Proses ini membantu mengurangi redundansi data serta mengekstrak informasi utama yang relevan. Koefisien yang dihasilkan dikenal sebagai MFCC dan umum digunakan sebagai fitur dalam berbagai aplikasi pengolahan sinyal audio seperti pengenalan suara[28]

2.5 Machine learning

Machine Learning (ML) merupakan pendekatan dalam *Artificial Intelligence* (AI) yang digunakan untuk menirukan perilaku manusia dalam menyelesaikan masalah atau melakukan otomatisasi. Secara umum, *machine learning* memiliki dua aplikasi utama, yaitu klasifikasi dan prediksi. Ciri khas *machine learning* adalah adanya proses pelatihan menggunakan data yang disebut sebagai data *training*, sehingga model dapat mempelajari pola untuk menyelesaikan tugas tertentu [31]

Machine Learning merupakan sebuah sub ilmu dari kecerdasan buatan di mana, cabang ilmu ini berfokus pada bagaimana membuat suatu mesin dapat memahami suatu pengetahuan dari hasil observasi yang dilakukan secara berulang-ulang. *Machine learning* dapat memahami suatu pola ataupun ilmu dengan cara melakukan pelatihan menggunakan suatu kumpulan data yang sudah disediakan. Dari pola yang didapatkan dari kumpulan data tersebut, mesin diharapkan dapat membuat suatu analisa dan menghasilkan suatu keluaran [32]. *Machine Learning* (ML) adalah studi ilmiah tentang algoritma dan model statistik yang digunakan sistem komputer untuk melakukan tugas tertentu tanpa diprogram secara eksplisit yang bertujuan untuk mengajarkan mesin mengenai bagaimana cara mengelola data secara lebih efisien[1]. Dalam implementasinya, *machine learning* memiliki banyak metode yang dapat digunakan untuk menangani klasifikasi, clustering, dan pengelolaan data lainnya [33].

$$f(y) = \text{encode}(\text{label}) \dots \dots \dots (2)$$

Dimana:

1. f : fungsi
2. y : input dari fungsi (misalnya "rock").
3. $\text{encode}(\text{label})$: mengubah label teks (misalnya "rock") menjadi angka.

2.6 Klasifikasi

Klasifikasi merupakan metode yang bertujuan untuk memetakan suatu data ke dalam kelas tertentu berdasarkan fitur yang dimiliki. Proses klasifikasi dilakukan dengan membangun model dari data latih. Kemudian model tersebut digunakan untuk memprediksi kelas pada data baru yang belum pernah dipelajari sebelumnya[34][35]. Secara umum, klasifikasi terdiri dari dua tahap utama, yaitu tahap pembelajaran (*training*), di mana algoritma klasifikasi membangun model berdasarkan pola yang terdapat pada data latih, kemudian tahap pengklasifikasian (*testing*), di mana model digunakan untuk memprediksi kelas dari data baru. Klasifikasi dapat dibedakan menjadi dua jenis utama:

1. Klasifikasi biner, yaitu pemetaan data ke dalam dua kelas.

2. Klasifikasi multikelas, yaitu pemetaan data ke dalam lebih dari dua kelas.

Klasifikasi yang dilakukan termasuk klasifikasi multikelas karena dataset yang digunakan terdiri dari sepuluh genre musik. Model klasifikasi dibangun berdasarkan fitur audio yang diekstraksi, kemudian digunakan untuk menentukan genre dari setiap file musik [26]. Klasifikasi juga dapat dianggap sebagai fungsionalitas data mining, karena bertujuan menghasilkan model yang memprediksi kategori dari objek-objek dalam basis data. Dalam konteks ini, model klasifikasi tidak hanya mempelajari properti-properti umum dari objek, tetapi juga memisahkan atribut-atribut ke dalam kategori atau kelas yang sesuai, sehingga dapat digunakan untuk memprediksi kelas objek yang belum diketahui sebelumnya [26].

2.7 Lasso Logistic Regression

Logistic Regression merupakan metode klasifikasi dalam *machine learning* yang digunakan untuk memodelkan probabilitas suatu data termasuk ke dalam kelas tertentu berdasarkan hubungan antara variabel input dan output kategorikal. Metode ini banyak digunakan karena mampu menangani masalah klasifikasi biner maupun multikelas dengan cara mengubah kombinasi linear dari fitur menjadi nilai probabilitas[36].

Logistic Regression merupakan pengembangan dari regresi logistik yang mengintegrasikan regularisasi L1 ke dalam proses estimasi parameter. Regresi logistik digunakan untuk memodelkan probabilitas suatu data termasuk ke dalam kelas tertentu melalui fungsi sigmoid, yang memetakan kombinasi linear fitur ke dalam rentang nilai antara 0 dan 1.

Fungsi sigmoid dinyatakan sebagai berikut [36] [34]:

$$\sigma(z) = \frac{1}{1+e^{-z}} \dots\dots\dots(3)$$

Dimana:

1. $\sigma(\text{sigma})$: fungsi sigmoid
2. z : input (hasil perhitungan)
3. $e(\text{euler})$: konstanta 2.718
4. e^{-z} : bagian eksponensial (nilai besar jadi kecil dan sebaliknya)
5. hasil akhir: angka antara 0 dan 1

Dengan:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p \dots\dots\dots(4)$$

Dimana:

1. β : bobot/pengaruh
2. x : data/variabel input

3. β_0 : bias (nilai datar saat semua $x = 0$)
4. z : hasil gabungan penjumlahan semua variabel
5. digunakan dalam *Logistic Regression* sebelum sigmoid

di mana $x_1, x_2, \dots, x_p$ merupakan fitur dan β adalah parameter model.

LASSO *Logistic Regression* merupakan metode klasifikasi yang menggunakan regularisasi L1 untuk melakukan seleksi fitur secara otomatis, sehingga model menjadi lebih sederhana serta dapat mengurangi dampak memberi informasi yang mirip (multikolinieritas) pada data berdimensi tinggi[34]. Pada metode ini ditambahkan penalti berbasis norma L1 terhadap koefisien regresi sehingga fungsi objektifnya menjadi:

$$L(\beta) = -\sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] + \lambda \sum_{j=1}^p |\beta_j| \dots \dots \dots (5)$$

Dimana:

1. $\log L(\beta)$ = log-likelihood
2. y_i : label asli (0 atau 1)
3. p_i : probabilitas dari sigmoid
4. β_j : koefisien fitur (fitur yang tidak penting otomatis dibuang)
5. λ : parameter kekuatan regularisasi
6. n : jumlah data
7. p : jumlah fitur

Nilai λ mengontrol kekuatan penalti. Semakin besar nilai λ , semakin banyak koefisien yang menjadi nol. Hal ini menyebabkan fitur yang tidak relevan dieliminasi secara otomatis.

Parameter λ mengontrol tingkat penalti terhadap koefisien model. Semakin besar nilai λ , semakin banyak koefisien yang ditekan menuju nol. Koefisien yang bernilai nol menunjukkan bahwa fitur tersebut tidak berkontribusi signifikan terhadap model, sehingga secara otomatis dieliminasi. Dengan demikian, metode ini tidak hanya berfungsi sebagai alat klasifikasi, tetapi juga sebagai mekanisme seleksi fitur yang efektif [37].

Untuk memperoleh nilai koefisien regresi yang optimal dengan mempertimbangkan penalti LASSO, digunakan fungsi objektif yang menggabungkan fungsi *log-likelihood* pada *Logistic Regression* dengan penalti L1. Fungsi ini bertujuan untuk meminimalkan kesalahan prediksi sekaligus membatasi kompleksitas model melalui penyusutan nilai koefisien. Secara matematis, berikut adalah rumus utama pada metode LASSO[38]:

$$\beta_{LASSO} = \arg \min_{\beta} \left\{ \left(\frac{1}{2n} \right) * \sum (y_i - \beta_0 - \sum \beta_j x_{ji})^2 + \lambda \sum |\beta_j| \right\} \dots \dots \dots (6)$$

Dimana:

1. β_{LASSO} : Hasil akhir atau nilai perkiraan koefisien dari model LASSO.
2. $\arg \min_{\beta}$: Artinya kita mencari nilai β yang membuat hasil perhitungan menjadi paling kecil (minimum)
3. y_i : Nilai asli/target pada data ke-i
4. β_0 : Nilai dasar (konstanta) dalam model
5. x_{ji} : Pengaruh masing-masing variabel ke-j terhadap hasil
6. Nilai variabel ke-j pada data ke-i
7. $\left(\frac{1}{2n}\right) * \sum (y_i - \beta_0 - \sum \beta_j x_{ji})^2$: Bagian yang mengukur seberapa jauh hasil prediksi dari nilai asli, semakin kecil semakin bagus
8. λ : Angka pengatur, variabel yang besar dibuat menjadi nol kemudian, bila variabel kecil bebas dari regresi biasa
9. β_j : membuat sebagian koefisien bisa menjadi nol.

2.8 Evaluasi Model

Tahap evaluasi dilakukan untuk mengukur kinerja dari model klasifikasi yang telah dibangun. Evaluasi dilakukan pada data uji yang dipisahkan dari data latih, menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score* yang dihitung dari *Confusion Matrix*. *Confusion Matrix* sendiri mengevaluasi kinerja model dengan menyajikan empat komponen utama sebagai dasar perhitungan [39].

- a. True Positive (TP): Data yang benar-benar positif dan diprediksi sebagai positif
- b. True Negative (TN): Data yang benar-benar negatif dan diprediksi sebagai negatif
- c. False Positive (FP): Data yang sebenarnya negatif namun diprediksi sebagai positif
- d. False Negative (FN): Data yang sebenarnya positif namun diprediksi sebagai negative

Dari keempat komponen tersebut, metrik evaluasi dihitung menggunakan rumus berikut:

1. Akurasi (*Accuracy*)

Accuracy mengukur seberapa banyak prediksi yang benar dibandingkan seluruh data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots \dots \dots (7)$$

2. Rumus *Precision*

Precision mengukur seberapa tepat prediksi positif yang dilakukan model

$$Precision = \frac{TP}{TP + FP} \dots \dots \dots (8)$$

3. Rumus *Recall* (*Sensitivity*)

Recall mengukur seberapa banyak data positif yang berhasil dideteksi model

$$Recall = \frac{TP}{TP + FN} \dots \dots \dots (9)$$

4. Rumus **F1-Score**

F1-Score adalah rata-rata harmonik antara Precision dan Recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \dots \dots \dots (10)$$

atau bisa ditulis langsung:

$$F1 = \frac{2TP}{2TP + FP + FN} \dots \dots \dots (11)$$