

BAB 2 LANDASAN TEORI

Pada bagian ini dijabarkan teori-teori yang mendasari penelitian yaitu Formula 1, *Machine Learning*, XGBoost, dan *Evaluation Metrics*.

2.1 Formula 1

Formula 1 (F1) merupakan salah satu cabang olahraga balap mobil yang sangat kompetitif dan bergengsi di dunia. Olahraga ini menampilkan mobil-mobil tercepat dan paling maju secara teknologi, yang dikendarai oleh pembalap-pembalap terbaik dari berbagai negara. Formula 1 memiliki sejarah panjang yang kaya, yang dimulai sejak awal abad ke-20, dan terus berkembang hingga menjadi fenomena global yang menarik perhatian jutaan penonton di seluruh dunia [19].

Dalam kejuaraan dunia Formula 1, poin diberikan kepada para pembalap berdasarkan posisi akhir mereka dalam setiap balapan. Sistem poin ini menjadi dasar penentuan peringkat pembalap dan konstruktor sepanjang musim. Terdapat dua jenis balapan yang memberikan poin, yaitu *feature race* (balapan utama) dan *sprint race* (balapan sprint), yang masing-masing memiliki skema distribusi poin yang berbeda [20]. Poin untuk *feature race* dan *sprint race* dapat dilihat pada Tabel 2.1.

Position	Feature Race Points	Sprint Race Points
1st	25	8
2nd	18	7
3rd	15	6
4th	12	5
5th	10	4
6th	8	3
7th	6	2
8th	4	1
9th	2	0
10th	1	0

Tabel 2.1. *Feature race* dan *sprint race* points

Poin dari *feature race* memiliki bobot yang lebih besar karena balapan ini merupakan inti dari setiap akhir pekan *Grand Prix*. Balapan utama ini biasanya berlangsung sejauh sekitar 305 km atau selama kurang lebih 90 menit, dan memberikan kontribusi terbesar terhadap klasemen kejuaraan [20].

Di sisi lain, *sprint race* adalah format balapan singkat yang diperkenalkan pada musim 2021 sebagai upaya untuk menambah daya tarik dan intensitas akhir pekan balapan dengan format yang mendorong pembalap untuk tampil maksimal sejak awal. Setelah menjalani tiga akhir pekan sprint pada musim 2021 dan 2022, jumlahnya kemudian meningkat menjadi enam pada musim 2023 dan tetap dipertahankan hingga musim 2024 dan 2025 [21]. Sprint biasanya berdurasi sekitar 100 km tanpa pit stop wajib, dan hanya memberikan poin kepada delapan pembalap terdepan. Perbedaan dalam distribusi poin ini dimaksudkan untuk mencerminkan durasi serta tingkat kepentingan masing-masing format balapan dalam kalender Formula 1 [22].

Sistem poin Formula 1 telah mengalami beberapa perubahan signifikan dari masa ke masa. Perubahan ini mencerminkan upaya FIA (*Fédération Internationale de l'Automobile*) untuk membuat kompetisi semakin menarik, adil, dan kompetitif. Tabel 2.2 berikut merangkum evolusi sistem poin dari awal hingga usulan terbaru untuk musim 2025.

Season	Feature Race	Sprint Race	Catatan
1950–1953	8–6–4–3–2	—	+1 poin <i>fastest lap</i>
1973–1980	9–6–4–3–2–1	—	Sistem <i>dropped points</i>
1991–2002	10–6–4–3–2–1	—	Top 6 diberi poin
2003–2009	10–8–6–5–4–3–2–1	—	Top 8 diberi poin
2010–2018	25–18–15–12–10–8–6–4–2–1	—	Top 10 diberi poin
2019–2020	25–18–15–12–10–8–6–4–2–1	—	+1 <i>fastest lap (top 10)</i>
2021	25–18–15–12–10–8–6–4–2–1	3–2–1 (top 3)	<i>Sprint debut</i>
2022–2024	25–18–15–12–10–8–6–4–2–1	8–7–6–5–4–3–2–1	<i>Sprint poin penuh</i>
2025	25–18–15–12–10–8–6–4–2–1	8–7–6–5–4–3–2–1	<i>Fastest lap</i> dihapus

Tabel 2.2. Evolusi sistem poin Formula 1 dari masa ke masa

Tabel 2.2 merangkum evolusi sistem poin Formula 1 dari awal penyelenggaraan hingga struktur yang diusulkan pada musim 2025. Sejak awal, sistem poin terus mengalami penyempurnaan untuk menciptakan kompetisi yang lebih adil, seimbang, dan menarik. Misalnya, pada periode 1991–2002, hanya enam pembalap teratas yang menerima poin. Kemudian pada musim 2003 hingga

2009, cakupan pemberian poin diperluas menjadi delapan pembalap teratas agar persaingan klasemen semakin kompetitif.

Perubahan besar dilakukan pada musim 2010 ketika FIA memperkenalkan sistem baru dengan memberikan poin kepada 10 pembalap teratas (*top 10* diberi poin), yaitu 25 poin untuk posisi pertama hingga 1 poin untuk posisi ke-10. Sistem ini menggantikan format sebelumnya yang hanya memberi poin kepada delapan atau enam pembalap teratas. Struktur ini bertahan hingga saat ini dan terus digunakan dengan beberapa penyesuaian, termasuk penambahan poin *fastest lap* dan penerapan format *sprint race*.

Format *sprint race* pertama kali diperkenalkan pada musim 2021 sebagai bentuk balapan pendek untuk menentukan posisi *start* pada balapan utama (*feature race*). Awalnya, hanya tiga pembalap tercepat yang mendapatkan poin dalam sprint. Namun sejak musim 2022, sistem ini diperluas dengan memberikan poin kepada delapan pembalap teratas dalam *sprint race*, yang terus berlaku hingga musim 2025.

Pada musim 2025, FIA secara resmi menghapus pemberian poin tambahan untuk *fastest lap*. Sebelumnya, sejak 2019, satu poin tambahan diberikan kepada pembalap yang mencetak waktu lap tercepat selama balapan dan finis di posisi sepuluh besar. Keputusan penghapusan ini diumumkan sebagai bagian dari penyesuaian regulasi yang bertujuan menjaga kesederhanaan dalam sistem penilaian dan meminimalkan elemen strategis tambahan yang dianggap tidak memberikan dampak signifikan terhadap perebutan gelar juara dunia [23, 24].

Selain itu, pada musim 2014, FIA sempat menerapkan aturan *double points* (poin ganda) yang kontroversial. Aturan ini hanya berlaku pada balapan terakhir musim tersebut, yaitu *Grand Prix Abu Dhabi*, di mana seluruh poin yang diberikan dilipatgandakan. *Double points* dapat dilihat pada Tabel 2.3.

Tabel 2.3. Sistem poin ganda pada *Grand Prix Abu Dhabi* 2014

Position	Double Points (Abu Dhabi 2014)
1st	50
2nd	36
3rd	30
4th	24
5th	20
6th	16
Lanjut pada halaman berikutnya	

Tabel 2.3 Sistem poin ganda pada *Grand Prix* Abu Dhabi 2014 (lanjutan)

Position	Double Points (Abu Dhabi 2014)
7th	12
8th	8
9th	4
10th	2

Tujuan utama dari kebijakan ini adalah untuk menjaga ketegangan dalam perebutan gelar juara hingga akhir musim. Namun, aturan ini mendapatkan banyak kritik dari pembalap dan penggemar karena dianggap tidak adil, sebab satu balapan menjadi jauh lebih menentukan dibandingkan balapan lainnya. Meskipun akhirnya tidak memengaruhi hasil kejuaraan karena Lewis Hamilton tetap menjadi juara dunia setelah memenangi balapan terakhir, aturan ini hanya berlaku sekali dan langsung dihapus pada musim berikutnya [23]. Mulai tahun 2025, FIA berencana melakukan beberapa perubahan, termasuk menghapus pemberian poin untuk *fastest lap*, yang sebelumnya diberikan sejak 2019 bagi pembalap dalam 10 besar [25]. Setelah memahami bagaimana poin diberikan dalam setiap balapan, penting untuk mengetahui bahwa posisi start yang diperoleh melalui sistem kualifikasi sangat memengaruhi peluang pembalap untuk meraih poin maksimal.

Sistem kualifikasi dalam Formula 1 terdiri dari tiga sesi yang berlangsung secara berurutan pada hari Sabtu, yaitu Q1, Q2, dan Q3. Format ini menggunakan sistem eliminasi untuk menentukan posisi start pembalap dalam balapan utama hari Minggu. Pada sesi pertama (Q1) yang berlangsung selama 18 menit, seluruh 20 pembalap turun ke lintasan untuk mencatatkan waktu tercepat. Lima pembalap dengan catatan waktu terburuk dieliminasi dan akan menempati posisi start ke-16 hingga ke-20. Sesi dilanjutkan dengan Q2 selama 15 menit, di mana 15 pembalap tersisa kembali bersaing untuk lolos ke Q3. Lima pembalap terbawah pada Q2 dieliminasi dan akan mengisi posisi start ke-11 hingga ke-15. Akhirnya, sepuluh pembalap tercepat bersaing di sesi Q3 yang berdurasi 12 menit untuk memperebutkan posisi *pole*, yaitu posisi start terdepan, serta posisi *grid* 1 sampai 10 [26, 27].

Selama sesi kualifikasi, tim harus memperhatikan strategi pemakaian ban, karena terdapat batasan jumlah set ban yang dapat digunakan selama satu akhir pekan *Grand Prix*. Selain itu, setelah sesi kualifikasi dimulai, mobil berada dalam

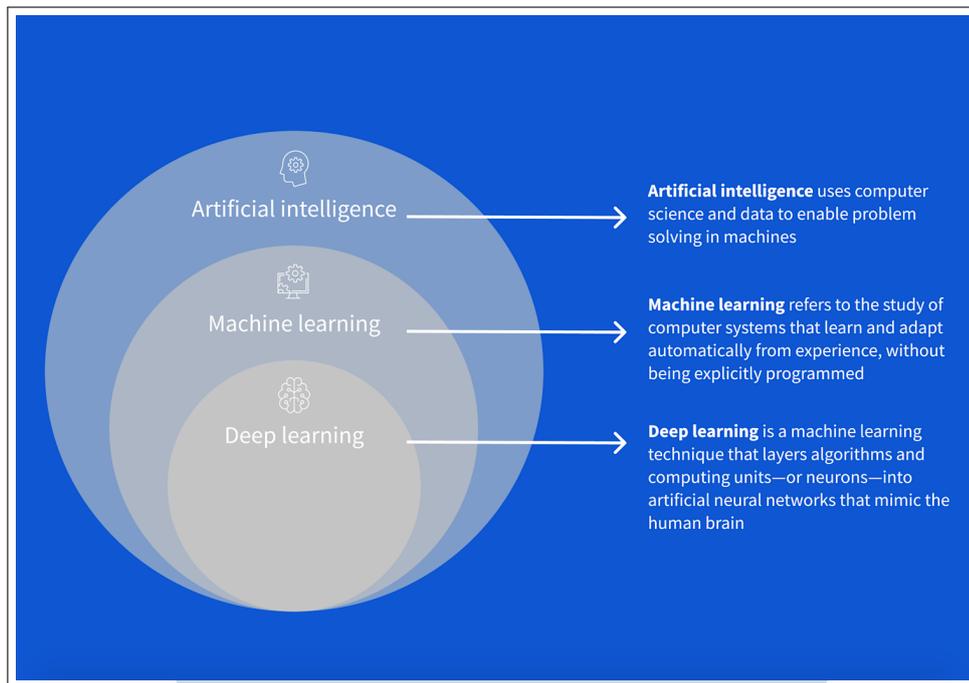
kondisi *parc fermé*, yang berarti tidak diperbolehkan melakukan perubahan besar pada pengaturan mobil hingga balapan berlangsung [26]. Terdapat pula aturan 107% di sesi Q1, di mana setiap pembalap harus mencatatkan waktu putaran tidak lebih lambat dari 107% waktu tercepat; jika tidak, maka ia dapat dilarang mengikuti balapan kecuali diberikan izin khusus oleh *steward* [28, 27].

Pada beberapa akhir pekan balapan tertentu, FIA menerapkan format *Sprint Qualifying*, yang digunakan untuk menentukan posisi start dalam balapan sprint, bukan *Grand Prix* utama. Format ini terdiri dari tiga sesi yang lebih singkat: SQ1 selama 12 menit, SQ2 selama 10 menit, dan SQ3 selama 8 menit. Skema ini bertujuan untuk menambah elemen strategis dan hiburan sepanjang akhir pekan balapan [27]. Sementara itu, hasil dari kualifikasi utama tetap digunakan untuk menentukan posisi start dalam *Grand Prix* hari Minggu. Penalti seperti penggantian komponen mesin atau pelanggaran teknis lainnya dapat mengakibatkan pengurangan posisi start, meskipun pembalap telah mencatat waktu kualifikasi yang kompetitif. Oleh karena itu, kualifikasi tidak hanya menguji kecepatan semata, tetapi juga strategi teknis dan manajemen sumber daya oleh tim dan pembalap.

2.2 Machine Learning

Machine learning (ML) adalah cabang dari *artificial intelligence* (AI) yang memungkinkan sistem untuk secara otomatis belajar dari data, mengidentifikasi pola, dan membuat keputusan atau prediksi dengan intervensi manusia yang minimal. Berbeda dengan pemrograman tradisional yang membutuhkan aturan eksplisit, ML membangun model berdasarkan data historis dan memperbaiki kinerjanya seiring waktu melalui proses pelatihan [29].

Menurut Badillo et al. (2020), *machine learning* dapat diklasifikasikan sebagai metode statistik dan komputasional yang bertujuan untuk mengekstrak pengetahuan dari data, baik dalam bentuk prediksi maupun klasifikasi. ML sangat berguna dalam menangani data besar (*big data*) dan kompleks, serta mampu beradaptasi terhadap perubahan pola dalam data input [30]. *Machine learning* tidak hanya menyelesaikan tugas-tugas secara otomatis, tetapi juga mampu belajar dari pengalaman. Di dalam *machine learning* sendiri terdapat cabang lebih lanjut yang dikenal sebagai *deep learning* (DL), yaitu pendekatan berbasis jaringan saraf tiruan yang meniru cara kerja otak manusia. *Deep learning* (DL) ini memanfaatkan banyak lapisan algoritma untuk mengolah informasi kompleks dan tidak terstruktur, contohnya seperti data gambar atau suara.



Gambar 2.1. Hubungan antara *Artificial Intelligence*, *Machine Learning*, dan *Deep Learning*.

Sumber: [31]

Gambar 2.1 menggambarkan hubungan hierarkis antara AI, ML, dan DL. AI merupakan lingkup paling luas yang mencakup semua sistem komputasi cerdas. *Machine learning* adalah *subset* dari *artificial intelligence* (AI) yang berfokus pada pembelajaran dari data, sedangkan *deep learning* adalah *subset* dari *machine learning* yang menggunakan arsitektur berlapis dalam bentuk jaringan saraf tiruan untuk menangani tugas-tugas yang lebih kompleks [31].

2.3 Extreme Gradient Boosting (XGBoost)

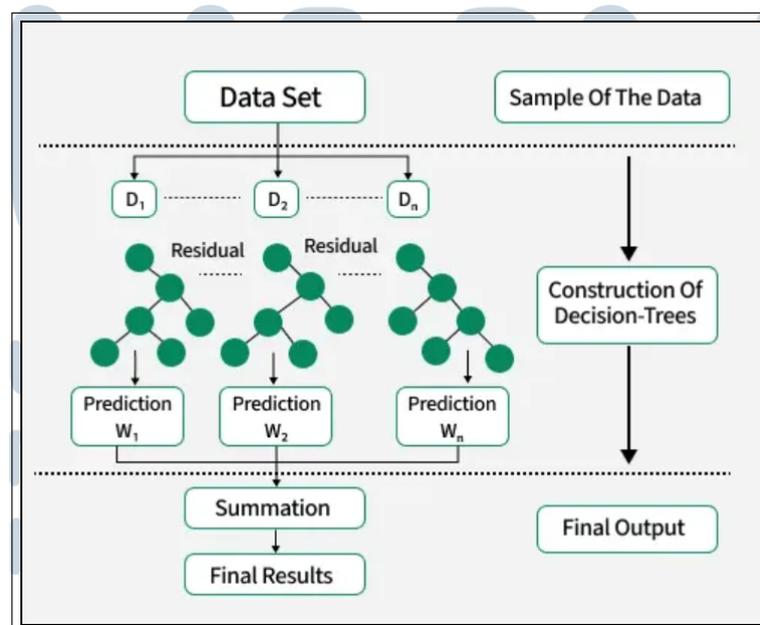
Extreme Gradient Boosting (XGBoost) adalah salah satu algoritma *machine learning* berbasis pohon keputusan (*decision tree*) yang dikembangkan dengan pendekatan *gradient boosting framework*. Algoritma ini dirancang untuk menjadi efisien, fleksibel, dan portabel, serta dikenal karena performanya yang tinggi dalam tugas prediksi baik untuk data tabular maupun kompetisi-kompetisi analitik berskala besar [32, 33].

Dalam pengembangan model prediksi berbasis data tabular, terdapat berbagai algoritma *machine learning* yang umum digunakan, khususnya yang berbasiskan pohon keputusan. Beberapa di antaranya adalah *Decision Tree*

dan *Random Forest*. Algoritma *Decision Tree* bekerja dengan membagi data berdasarkan fitur yang memberikan informasi paling tinggi pada setiap cabang, hingga terbentuk struktur pohon yang dapat digunakan untuk klasifikasi atau regresi [34]. Meskipun sederhana dan mudah diinterpretasikan, model ini cenderung mengalami *overfitting* jika tidak dilakukan proses *pruning*.

Random Forest merupakan metode *ensemble* yang membangun banyak *decision tree* secara paralel dan menggabungkan hasil prediksi dari masing-masing pohon. Pendekatan ini meningkatkan akurasi dan mengurangi variansi dibandingkan *decision tree* tunggal, namun memiliki keterbatasan dalam menangani data dengan noise tinggi atau saat dibutuhkan performa prediktif yang optimal [35].

XGBoost menawarkan keunggulan signifikan dibandingkan kedua metode tersebut. Dengan pendekatan *gradient boosting*, XGBoost membangun model secara sekuensial dan iteratif, di mana setiap pohon baru memperbaiki kesalahan dari model sebelumnya. Model lemah yang digunakan umumnya adalah *decision tree* dengan kedalaman terbatas. Selain itu, XGBoost juga menerapkan teknik regularisasi eksplisit untuk mengontrol kompleksitas model, yang membantu mencegah *overfitting*. Kombinasi efisiensi komputasi, kemampuan generalisasi tinggi, serta fleksibilitas terhadap berbagai tipe data menjadikan XGBoost sebagai pilihan yang tepat untuk prediksi data tabular [32, 33].



Gambar 2.2. Proses pelatihan model pada XGBoost (*Extreme Gradient Boosting*)

Sumber: [36]

Gambar 2.2 menggambarkan proses kerja XGBoost yang merupakan pengembangan dari metode *Gradient Boosting Decision Tree* (GBDT). Proses dimulai dengan penggunaan seluruh dataset yang dibagi menjadi beberapa *batch* data untuk membentuk pohon pertama. Setelah itu, residual atau kesalahan dari prediksi pertama dihitung dan digunakan untuk melatih pohon berikutnya. Proses ini terus berlanjut secara iteratif, di mana setiap pohon berperan untuk memperbaiki kesalahan dari pohon sebelumnya.

Setiap pohon memberikan prediksi parsial yang kemudian dikalikan dengan bobot (w_1, w_2, \dots, w_n) dan dijumlahkan untuk menghasilkan hasil akhir. Teknik ini sangat efektif dalam meningkatkan akurasi karena fokusnya pada penyempurnaan kesalahan model sebelumnya secara bertahap.

Pendekatan ini berbeda dengan metode seperti *random forest*, yang membangun semua pohon secara paralel dan independen, sedangkan XGBoost membangun pohon secara sekuensial dan bersifat aditif.

Secara matematis, XGBoost membangun model prediksi secara aditif:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i) \quad (2.1)$$

di mana:

- \hat{y}_i adalah hasil prediksi untuk data ke- i ,
- K adalah jumlah total pohon yang digunakan dalam model,
- $f_k(x_i)$ adalah *output* dari pohon ke- k untuk input x_i , di mana setiap f_k termasuk dalam ruang fungsi pohon regresi: $f_k \in \mathcal{F}$.

Model ini dioptimalkan dengan meminimalkan fungsi objektif:

$$obj(\theta) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (2.2)$$

di mana:

- $obj(\theta)$ adalah fungsi objektif yang akan diminimalkan,
- $l(y_i, \hat{y}_i)$ adalah fungsi kerugian yang mengukur selisih antara nilai aktual y_i dan hasil prediksi \hat{y}_i ,

- $\Omega(f_k)$ adalah fungsi regulasi yang diterapkan pada pohon ke- k untuk mengontrol kompleksitas model,
- θ merupakan parameter model secara keseluruhan, yang mencakup struktur pohon dan bobot keluaran pada setiap daun.

Sementara itu, $\Omega(f_k)$ adalah fungsi regulasi (regularization term) yang bertujuan untuk mengontrol kompleksitas setiap pohon dalam model. Regulasi ini merupakan salah satu keunggulan utama dari XGBoost, karena membantu mencegah *overfitting* dan meningkatkan generalisasi model [32, 33]. Regulasi ini tidak hanya mempertimbangkan jumlah daun pada pohon, tetapi juga penalti terhadap nilai *output* dari setiap daun pohon.

Fungsi regulasi $\Omega(f)$ secara umum didefinisikan sebagai berikut:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (2.3)$$

di mana:

- T adalah jumlah daun (*terminal nodes*) dalam pohon keputusan,
- w_j adalah skor prediksi (nilai *output*) pada daun ke- j ,
- γ adalah parameter untuk penalti terhadap jumlah daun,
- λ adalah parameter regularisasi L2 untuk penalti terhadap besar kecilnya bobot w_j .

Dalam proses pelatihan, XGBoost menentukan titik *split* terbaik dengan memaksimalkan *information gain* atau hanya disebut *gain*. *Gain* mengukur seberapa besar peningkatan akurasi yang diperoleh dengan membagi satu simpul (*node*) menjadi dua simpul anak. Rumus perhitungan *gain* dalam XGBoost diberikan sebagai berikut:

$$Gain = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \quad (2.4)$$

dengan:

- G_L, G_R adalah jumlah gradien dari simpul anak kiri dan kanan,

- H_L , H_R adalah jumlah *Hessian* (turunan kedua) dari simpul anak kiri dan kanan,
- λ adalah parameter regularisasi L2,
- γ adalah penalti terhadap penambahan jumlah simpul (daun).

Rumus ini menunjukkan bahwa pemilihan *split* terbaik dilakukan dengan mencari nilai *gain* tertinggi. Semakin tinggi nilai *gain*, maka semakin besar kontribusi pemisahan simpul tersebut dalam mengurangi kesalahan prediksi (*loss*). Jika nilai *gain* tidak melebihi nilai γ , maka pembelahan simpul tersebut akan dibatalkan untuk mencegah kompleksitas model yang berlebihan.

2.4 Evaluation Metrics

Dalam membangun model prediktif berbasis *machine learning*, diperlukan metrik evaluasi yang tepat untuk mengukur sejauh mana model mampu menghasilkan prediksi yang mendekati nilai sebenarnya. Evaluasi dilakukan untuk memastikan bahwa model tidak hanya memiliki performa baik pada data latih, tetapi juga dapat menggeneralisasi terhadap data baru. Tiga metrik evaluasi utama yang umum digunakan dalam regresi adalah *Mean Absolute Error* (MAE), *Root Mean Square Error* (RMSE), dan *Koefisien Determinasi* (R^2 Score) [37, 38].

2.4.1 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) adalah ukuran rata-rata dari selisih absolut antara nilai aktual dan nilai prediksi. Metrik ini mengukur seberapa besar kesalahan prediksi model secara umum tanpa memperhatikan arah kesalahan. MAE bersifat lebih stabil terhadap nilai ekstrem karena tidak mengkuadratkan selisihnya [38].

MAE dihitung dengan rumus sebagai berikut:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.5)$$

di mana y_i adalah nilai aktual ke- i , \hat{y}_i adalah hasil prediksi ke- i , dan n adalah jumlah total data. Nilai MAE yang kecil menunjukkan bahwa model memiliki kesalahan rata-rata yang rendah dalam memprediksi target.

2.4.2 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) adalah metrik yang menghitung akar kuadrat dari rata-rata selisih kuadrat antara nilai aktual dan prediksi. Berbeda dengan MAE, RMSE memberikan penalti lebih besar terhadap kesalahan prediksi yang besar karena melibatkan kuadrat dari selisih nilai [38].

RMSE dihitung dengan rumus:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.6)$$

Pada rumus di atas, y_i adalah nilai aktual ke- i , \hat{y}_i adalah hasil prediksi ke- i , dan n adalah jumlah total data. RMSE yang lebih kecil menunjukkan bahwa kesalahan prediksi model cenderung kecil dan performa model cukup baik.

2.4.3 R^2 Score (Koefisien Determinasi)

R^2 Score atau *Koefisien Determinasi* adalah metrik yang mengukur proporsi variasi dari nilai target yang dapat dijelaskan oleh model. Nilai R^2 berkisar antara 0 hingga 1, di mana nilai yang lebih tinggi menunjukkan model mampu menjelaskan variasi data dengan lebih baik [37].

Rumus R^2 Score sebagai berikut:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.7)$$

Pada rumus tersebut, y_i adalah nilai aktual ke- i , \hat{y}_i adalah nilai prediksi ke- i , dan \bar{y} merupakan rata-rata dari seluruh nilai aktual. Semakin kecil total kesalahan prediksi dibandingkan variasi data aslinya, semakin besar nilai R^2 yang menunjukkan kualitas model.

2.4.4 Cross-Validation

Cross-validation adalah teknik validasi model yang digunakan untuk mengevaluasi performa generalisasi model dengan membagi dataset ke dalam beberapa *subset*. Tujuannya adalah untuk memastikan bahwa model tidak hanya berperforma baik pada data latih, tetapi juga pada data yang belum pernah dilihat sebelumnya. Salah satu teknik yang paling umum digunakan adalah *k-fold cross-validation* [39].

Pada metode *k-fold cross-validation*, data dibagi ke dalam k bagian atau lipatan (*folds*) yang sama besar. Proses pelatihan dan pengujian dilakukan sebanyak k kali, dengan setiap lipatan secara bergiliran digunakan sebagai data uji dan sisanya sebagai data latih. Nilai metrik evaluasi (seperti MAE, RMSE, dan R^2) dihitung pada setiap iterasi, lalu dirata-ratakan untuk memberikan gambaran umum performa model.

Sebagai contoh, dalam *5-fold cross-validation*:

- Data dibagi menjadi 5 *subset*.
- Pada iterasi ke-1, *subset* ke-1 digunakan sebagai data uji, dan *subset* ke-2 s.d. ke-5 sebagai data latih.
- Pada iterasi ke-2, *subset* ke-2 sebagai data uji, sisanya sebagai data latih, dan seterusnya.
- Hasil evaluasi dari kelima iterasi dirata-ratakan.

Teknik ini membantu dalam mengurangi bias dalam evaluasi model dan memberikan estimasi yang lebih stabil terhadap kinerja model dibandingkan pembagian data tunggal (*train-test split*). *Cross-validation* sangat penting terutama ketika jumlah data terbatas.

