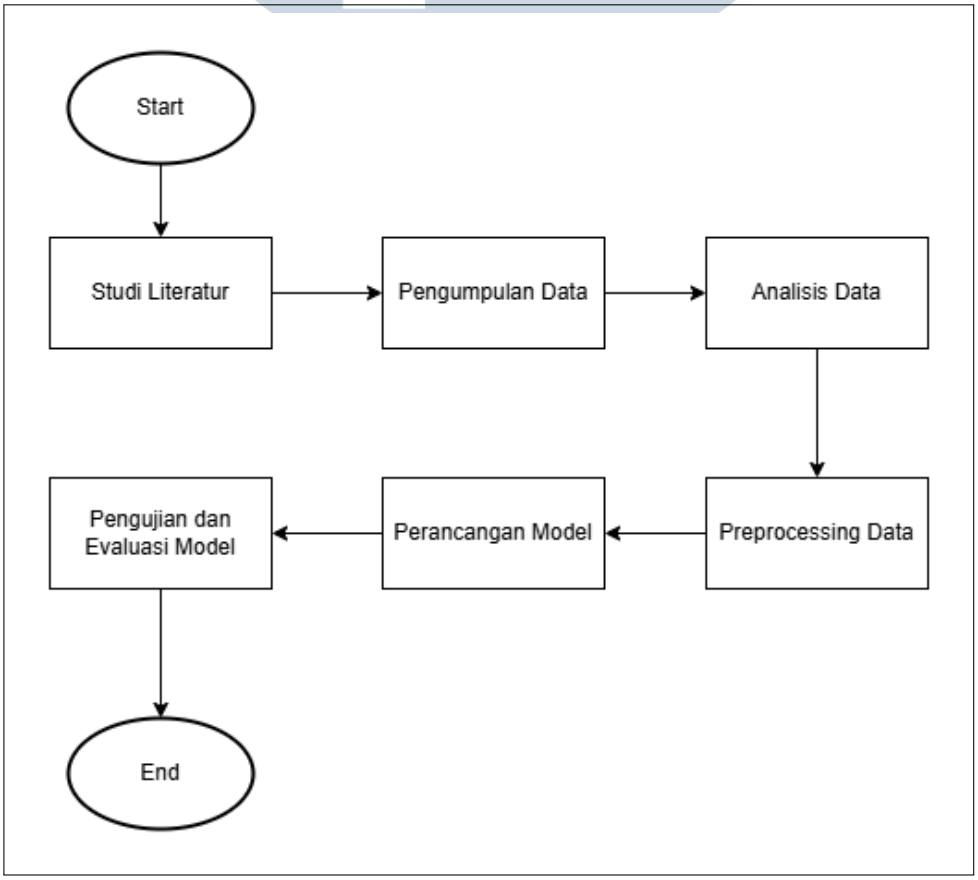
BAB 3
METODOLOGI PENELITIAN

Metodologi penelitian ini dirancang untuk memprediksi pemabalap juara dunia Formula 1 Tahun 2025 menggunakan algoritma *Extreme Gradient Boosting* (XGBoost). Penelitian ini dilakukan secara bertahap dan sistematis untuk memastikan hasil prediksi yang akurat dan dapat diandalkan. Setiap tahapan memiliki peran penting dalam membentuk model prediksi yang optimal dan menghindari kesalahan umum dalam pengolahan data dan pelatihan model.

Gambar 3.1 menggambarkan alur proses penelitian mulai dari studi literatur hingga evaluasi model. Tahapan ini mencakup pengumpulan data historis Formula 1, analisis data, pembersihan dan persiapan data (*preprocessing*), perancangan model prediktif, serta evaluasi performa model berdasarkan metrik yang telah ditentukan. Setiap tahapan dirancang saling terhubung dan mendukung pengambilan keputusan yang tepat dalam pemodelan.



Gambar 3.1. *Flowchart* metodologi penelitian

3.1 Studi Literatur

Studi literatur dilakukan untuk memahami teori-teori yang menjadi dasar dalam pelaksanaan penelitian ini, khususnya yang berkaitan dengan bidang *machine learning*, algoritma XGBoost, serta karakteristik data dalam olahraga balap Formula 1. Referensi yang digunakan dalam studi ini berasal dari berbagai sumber yang kredibel, seperti jurnal ilmiah, publikasi akademik, dokumentasi resmi Formula 1, serta literatur lain yang relevan dengan topik penelitian. Beberapa studi telah dilakukan untuk memprediksi hasil akhir Kejuaraan Dunia Pembalap Formula 1 (*World Driver Champion*/WDC). Tabel 3.1 berikut merangkum studi-studi utama yang relevan beserta metode, keunggulan, dan keterbatasannya masing-masing.

Tabel 3.1. Ringkasan studi terdahulu prediksi juara dunia pembalap Formula 1 dan *research gap*

No	Peneliti	Tahun	Periode Data	Model	Target Prediksi	Metrik Evaluasi	Keterbatasan (Research Gap)
1	Den Hartog	2023	2014–2022	*Logistic Regression*, *Random Forest*, *AdaBoost*, XGBoost	Juara Dunia (*World Driver Champion*)	Akurasi, *F1-score*	Prediksi hanya dilakukan di akhir musim; tidak menggunakan XAI; tidak mempertimbangkan faktor kontekstual seperti cuaca, penalti, strategi pit stop.
2	Van Kesteren & Bergkamp	2022	2014–2021	*Bayesian Multilevel Rank-Ordered Logit Regression*	Kontribusi *Driver* vs *Constructor*	Proporsi variansi kontribusi konstruktor	Hanya analisis kontribusi deskriptif, bukan model prediksi; tidak membangun klasemen atau WDC prediktif.
3	Penelitian ini	2025	1950–2025	XGBoost dengan *hyperparameter tuning* (Optuna)	Posisi klasemen sementara musim 2025 hingga *round* ke-10 (regresi posisi juara dunia)	R^2, MAE, RMSE	Belum menggunakan *time-series* atau XAI; belum mengikutsertakan variabel cuaca, penalti, strategi pit stop; namun telah mengimplementasikan prediksi *mid-season* dan mencakup dataset historis panjang.

Penelitian oleh Den Hartog (2023) berfokus pada prediksi juara dunia pembalap/*World Driver Champion* (WDC) Formula 1 setiap musim menggunakan pendekatan *machine learning*. Penelitian ini menggunakan data dari musim 2014 hingga 2022 dengan algoritma seperti *Logistic Regression*, *Random Forest*, AdaBoost, dan XGBoost. Model terbaik diperoleh dari XGBoost dengan akurasi sebesar 93% dan nilai *F1-score* sebesar 0,85. Fitur utama yang digunakan meliputi rata-rata posisi finis, jumlah kemenangan, dan jumlah podium [40].

Meskipun hasilnya menjanjikan, studi ini memiliki beberapa keterbatasan penting. Prediksi hanya dilakukan di akhir musim, sehingga tidak merepresentasikan kemungkinan penggunaan model selama musim berjalan (*mid-season prediction*). Selain itu, penelitian ini belum menerapkan pendekatan

interpretabilitas seperti *Explainable AI* (XAI) untuk menjelaskan kontribusi setiap fitur terhadap hasil prediksi. Faktor-faktor kontekstual seperti penalti, strategi pit stop, dan kondisi cuaca juga belum dipertimbangkan dalam pemodelan [40].

Di sisi lain, studi oleh Van Kesteren dan Bergkamp (2022) mengkaji kontribusi antara pembalap dan konstruktor terhadap performa tim menggunakan pendekatan *Bayesian multilevel rank-ordered logit regression*. Hasil penelitian menunjukkan bahwa konstruktor berkontribusi sebesar 88% terhadap hasil balapan. Meskipun memberikan wawasan penting mengenai dominasi konstruktor, studi ini hanya bersifat deskriptif dan tidak mengembangkan model prediktif untuk menentukan klasemen akhir atau juara dunia pembalap [41].

Berdasarkan keterbatasan studi sebelumnya, penelitian ini mengambil posisi untuk membangun model prediksi juara dunia pembalap Formula 1 yang lebih menyeluruh dan praktis. Penelitian ini mengusulkan pendekatan *XGBoost* yang dioptimalkan menggunakan *hyperparameter tuning* melalui Optuna. Model dikembangkan berdasarkan data historis lengkap dari tahun 1950 hingga pertengahan musim 2025 (sampai *round* ke-10). Evaluasi dilakukan menggunakan metrik regresi seperti R^2, *Mean Absolute Error* (MAE), dan *Root Mean squared Error* (RMSE). Dengan pendekatan ini, model yang dibangun mampu memberikan prediksi awal terhadap potensi juara dunia berdasarkan klasemen *mid-season*, yang bermanfaat untuk kebutuhan analisis kompetitif di dunia nyata Formula 1.

3.2 Pengumpulan Data

Data historis Formula 1 diperoleh melalui Jolpi API yang menyediakan akses ke hasil balapan, kualifikasi, klasemen pembalap dan konstruktor, serta informasi terkait tim dan sirkuit [42]. Data yang dikumpulkan mencakup musim 1950 hingga 2025 dan diperoleh dengan menggunakan bahasa pemrograman Python. *Endpoint* yang digunakan dalam proses pengumpulan data yaitu:

- `/races` — informasi jadwal dan lokasi balapan.
- `/results` — hasil akhir setiap balapan (*finish position*).
- `/qualifying` — hasil sesi kualifikasi (*start position*).
- `/drivers` — data demografis dan identitas pembalap.
- `/driverstandings` — total poin dan peringkat pembalap per musim.

- `/constructors` — informasi tim/konstruktor Formula 1.
- `/constructorstandings` — peringkat tim berdasarkan akumulasi poin.

Setelah keseluruhan data diperoleh dari berbagai *endpoint*, data kemudian diolah dan disimpan dalam format `.csv` agar mudah diakses untuk analisis lebih lanjut. Penyimpanan dalam bentuk tabular ini juga memudahkan proses eksplorasi awal dan visualisasi data. Format `.csv` dipilih karena kompatibel dengan berbagai pustaka analisis data seperti `pandas` di Python.

3.3 Analisis Data

Analisis data digunakan untuk memahami struktur data, pola historis, dan hubungan antar fitur. Visualisasi dilakukan dengan pustaka `plotly`. Analisis ini mencakup distribusi poin tiap musim, performa pembalap berdasarkan tim, serta hubungan antara posisi kualifikasi dan hasil akhir balapan. Tahap ini melibatkan eksplorasi awal terhadap data untuk memahami pola, distribusi, serta relasi antar variabel. Analisis dilakukan untuk:

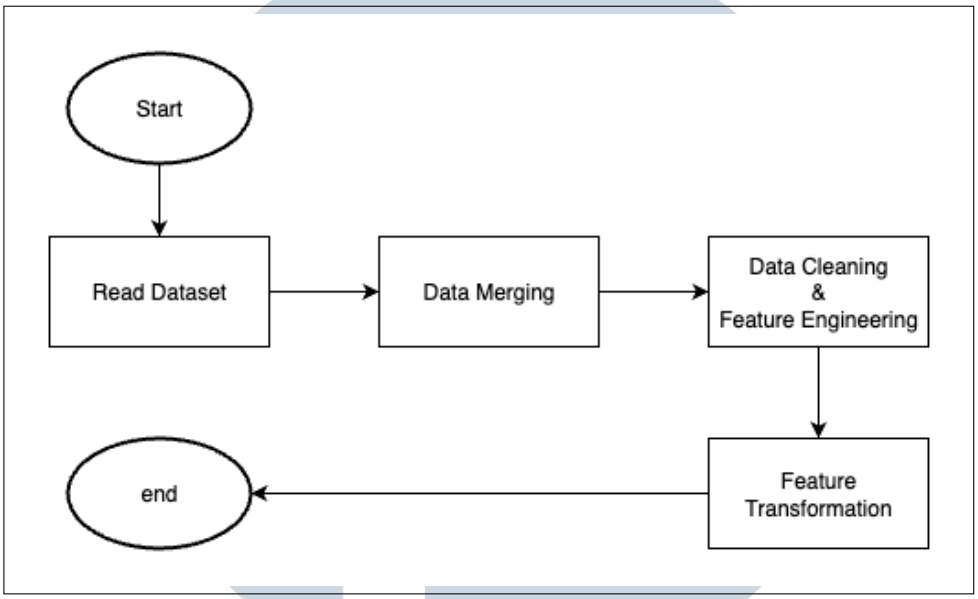
- Mengidentifikasi variabel prediktor potensial terhadap hasil klasemen akhir pembalap.
- Mencari korelasi antar fitur numerik dan kategorikal terhadap target (posisi klasemen akhir).
- Menentukan data relevan yang akan digunakan untuk fitur dalam model prediksi.

Teknik statistik seperti korelasi `Cramér's V` digunakan untuk melihat hubungan antar variabel. Selain itu, dilakukan visualisasi data untuk mendukung pemahaman pola dalam dataset. Hasil analisis dari kedua metode tersebut akan digunakan sebagai dasar utama dalam proses seleksi fitur dan pengembangan model prediksi.

3.4 Pre-processing Data

Data yang telah dikumpulkan dan dianalisis sebelumnya, selanjutnya memasuki tahap *pre-processing data* melalui proses pembersihan dan transformasi agar dapat digunakan sebagai input model. Proses ini melibatkan penggabungan

antar sumber data yang relevan, penghapusan data duplikat, penanganan nilai kosong, serta konversi tipe data. Tahapan *pre-processing* data tersebut dapat dilihat pada Gambar 3.2.



Gambar 3.2. *Flowchart pre-processing data*

Tahapan-tahapan *pre-processing* yang ditunjukkan pada Gambar 3.2 dapat dijelaskan sebagai berikut:

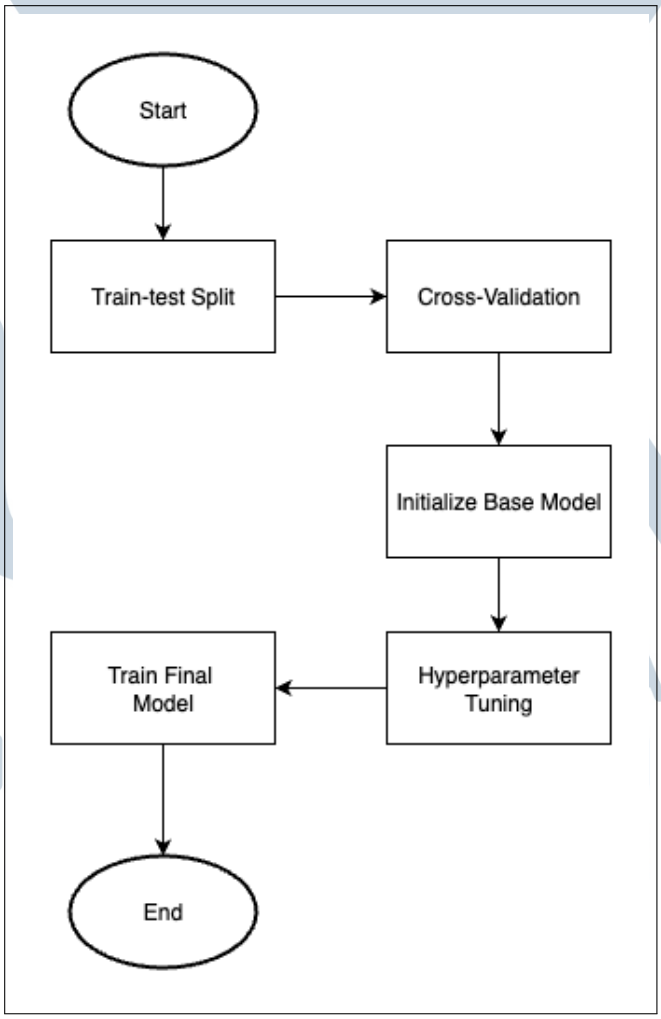
- *Data Merging*: Menggabungkan berbagai dataset menjadi satu dataset utama yang konsisten dan terintegrasi.
- *Data Cleaning* dan *Feature Engineering*: Melakukan pembersihan data dengan menghapus duplikat, menangani *missing values*, dan merapikan format data. Selain itu, dilakukan pembuatan fitur-fitur baru seperti rata-rata posisi finis, rata-rata posisi start, rasio penyelesaian balapan (*DNF rate*), serta tren perolehan poin. Fitur-fitur yang paling berkontribusi terhadap prediksi dipilih berdasarkan hasil eksplorasi dan analisis pentingnya fitur (*feature importance*).
- *Feature Transformation*: Mengubah fitur kategorikal menjadi format numerik menggunakan teknik `LabelEncoder` dan `OrdinalEncoder`, serta melakukan normalisasi pada fitur numerik dengan menggunakan `StandardScaler` untuk memastikan distribusi data seimbang dalam proses pelatihan model.

Target prediksi dalam penelitian ini adalah pembalap yang akan menjadi juara dunia pada musim 2025, yaitu pembalap dengan peringkat akhir pertama.

Oleh karena itu, seluruh fitur yang tersedia hingga sebelum balapan terakhir dimanfaatkan sebagai masukan model, guna memastikan prediksi dilakukan tanpa kebocoran data dari hasil akhir musim.

3.5 Perancangan Model

Model prediksi yang digunakan dalam penelitian ini adalah XGBoost (*Extreme Gradient Boosting*), yaitu algoritma *ensemble learning* berbasis pohon keputusan yang sangat efisien untuk regresi dan klasifikasi. Target yang diprediksi adalah posisi pembalap di akhir musim 2025, dan fitur masukan meliputi hasil kualifikasi, hasil balapan, performa historis pembalap, serta tim yang dibela. Untuk memodelkan prediksi posisi akhir pembalap pada musim 2025, dilakukan serangkaian langkah perancangan model berbasis algoritma XGBoost. Alur lengkap proses perancangan model tersebut divisualisasikan dalam Gambar 3.3.



Gambar 3.3. *Flowchart* perancangan model

Langkah-langkah perancangan model yang ditunjukkan pada Gambar 3.3 dapat dijelaskan sebagai berikut:

- *Train-test Split*: Dataset dibagi berdasarkan waktu (temporal split), di mana data historis dari musim 2014 hingga musim 2024 digunakan sebagai data latih, dan data musim 2025 digunakan sebagai data uji.
- Inisialisasi model dasar: Model XGBoost diinisialisasi menggunakan parameter *default* sebagai *baseline* untuk evaluasi awal.
- *Cross-validation*: Dilakukan validasi menggunakan teknik *5-Fold Cross-Validation* pada data latih untuk menilai stabilitas dan kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya.
- *Hyperparameter Tuning*: Optimasi parameter dilakukan menggunakan *library* `Optuna` yang bekerja berdasarkan prinsip *Bayesian optimization* untuk mencari kombinasi parameter yang meminimalkan kesalahan.
- Pelatihan model akhir: Model akhir dilatih menggunakan parameter terbaik hasil tuning dan seluruh data latih, sebelum digunakan untuk prediksi pada musim 2025.

3.5.1 Train-test Split

Pembagian dataset dalam penelitian ini menggunakan pendekatan temporal (berbasis waktu), yaitu dengan memisahkan data historis sebagai data latih (*train*) dan data musim 2025 sebagai data uji (*test*). Pendekatan ini dipilih karena dalam konteks prediksi dunia nyata, data masa depan tidak tersedia saat pelatihan, sehingga hanya data masa lalu yang dapat digunakan. Namun, agar model tidak terlalu bergantung pada data latih dan untuk menghindari optimisme berlebih terhadap performa prediksi 2025, digunakan data validasi melalui proses *cross-validation* pada data latih. Baru setelah itu model diterapkan ke data uji 2025.

Eksperimen dilakukan dengan tiga cakupan data historis: 10 musim (2014–2024), 20 musim (2004–2024), dan 30 musim (1994–2024). Tujuan dari variasi ini adalah untuk menguji seberapa besar pengaruh cakupan data historis terhadap kualitas prediksi. Semakin panjang data historis yang digunakan, diharapkan model dapat belajar pola performa jangka panjang dari pembalap dan tim.

3.5.2 Cross-Validation

Setelah model dasar diinisialisasi, dilakukan proses *cross-validation* untuk mengukur performa awal model terhadap data latih. Metode yang digunakan adalah *5-Fold Cross-Validation*, di mana data latih dibagi menjadi lima subset yang seimbang. Pada setiap iterasi, empat *subset* digunakan sebagai data latih dan satu *subset* sebagai data validasi, hingga seluruh *subset* bergiliran menjadi data validasi.

Hasil dari lima iterasi ini dirata-ratakan untuk menghasilkan metrik evaluasi awal model berupa *Mean Absolute Error*, *Root Mean squared Error* (RMSE), dan R^2 *Score* (Koefisien Determinasi). Evaluasi ini penting sebagai tolok ukur performa sebelum dilakukan tuning dan sebagai strategi untuk mencegah *overfitting*. Melalui validasi ini, model dapat belajar dari variasi internal dalam data latih tanpa mengintip data musim 2025, sehingga prediksi akhir yang dilakukan lebih meyakinkan dan dapat diandalkan.

3.5.3 Hyperparameter Tuning

Hyperparameter merupakan parameter yang tidak dipelajari secara langsung oleh model dari data, melainkan harus ditentukan sebelum proses pelatihan. Beberapa contoh parameter penting pada XGBoost antara lain adalah kedalaman maksimum pohon (`max_depth`), laju pembelajaran (`learning_rate`), jumlah pohon estimasi (`n_estimators`), dan tingkat regularisasi (`reg_alpha`, `reg_lambda`) [43]. Penentuan nilai parameter-parameter ini sangat memengaruhi kinerja akhir dari model yang dibangun, terutama dalam konteks prediksi berbasis data historis yang kompleks.

Penyesuaian *hyperparameter* secara otomatis melalui proses *hyperparameter tuning* menawarkan keunggulan dibandingkan pendekatan manual. Penyetelan parameter secara manual cenderung bersifat subjektif dan kurang sistematis, sehingga rentan menghasilkan performa yang tidak optimal, terutama pada model yang memiliki kompleksitas tinggi seperti XGBoost. Sebaliknya, metode tuning otomatis memungkinkan eksplorasi ruang parameter secara menyeluruh dan terarah, serta meningkatkan kemungkinan diperolehnya kombinasi parameter yang optimal. Tidak seperti parameter model yang dipelajari selama pelatihan, *hyperparameter* harus ditentukan terlebih dahulu dan memiliki pengaruh signifikan terhadap kinerja akhir model [44].

Terdapat beberapa metode umum yang digunakan untuk melakukan *hyperparameter tuning*, seperti *Grid Search* dan *Random Search*. *Grid Search* bekerja dengan mengevaluasi seluruh kombinasi parameter dalam ruang pencarian yang telah ditentukan secara eksplisit, namun sangat memakan waktu terutama saat jumlah parameter banyak [45]. Sementara itu, *Random Search* memilih kombinasi parameter secara acak dari distribusi tertentu dan biasanya lebih efisien dari segi waktu, namun kurang sistematis dan dapat melewatkan kombinasi terbaik [45].

Library yang digunakan untuk melakukan *hyperparameter tuning* adalah Optuna. Optuna merupakan *library open-source* untuk optimasi otomatis *hyperparameter* berbasis pendekatan *Bayesian optimization*. Tidak seperti *Grid Search* atau *Random Search*, Optuna menggunakan strategi adaptif untuk mengeksplorasi ruang parameter berdasarkan hasil evaluasi sebelumnya. Optuna menerapkan algoritma *Tree-structured Parzen Estimator* (TPE) yang mampu mempelajari distribusi parameter dan secara iteratif memilih kombinasi parameter yang paling menjanjikan untuk diuji berikutnya [46].

Selain itu, Optuna mendukung fitur *early stopping*, yaitu menghentikan pencarian parameter saat hasil yang diperoleh tidak lagi menunjukkan peningkatan yang signifikan. Hal ini membuat proses tuning menjadi jauh lebih efisien secara komputasi dan waktu. Oleh karena itu, pemilihan Optuna untuk *hyperparameter* pada model prediksi ini dianggap paling tepat karena ruang parameter XGBoost relatif kompleks dan besar, sehingga diperlukan strategi pencarian parameter yang cerdas dan hemat sumber daya.

3.6 Pengujian dan Evaluasi Model

Evaluasi dilakukan untuk mengukur performa model dalam memprediksi juara dunia pembalap Formula 1 di akhir musim 2025. Evaluasi ini penting untuk memastikan bahwa model tidak hanya cocok pada data latih, tetapi juga mampu melakukan generalisasi dengan baik terhadap data uji. Beberapa metrik yang digunakan dalam evaluasi ini adalah sebagai berikut:

- *Mean Absolute Error* (MAE): Mengukur rata-rata selisih absolut antara nilai prediksi dan nilai aktual. MAE memberikan gambaran seberapa besar kesalahan prediksi model secara umum, tanpa memperhatikan arah kesalahan.

- *Root Mean Squared Error* (RMSE): Merupakan akar dari rata-rata kuadrat selisih antara nilai prediksi dan nilai aktual. RMSE memberikan penalti yang lebih besar terhadap kesalahan yang ekstrem dibandingkan MAE, sehingga lebih sensitif terhadap *outlier*.
- R^2 *Score* (Koefisien Determinasi): Mengukur proporsi variasi dari variabel target yang dapat dijelaskan oleh fitur input model. Nilai R^2 yang mendekati 1 menunjukkan bahwa model memiliki kemampuan prediksi yang baik.

Evaluasi ini menjadi dasar dalam menilai seberapa andal dan efektif model XGBoost dalam memprediksi juara dunia pembalap Formula 1 musim 2025, serta memberikan wawasan terhadap tingkat akurasi dan keandalan model terhadap data yang belum pernah dilihat sebelumnya. Selain itu , hasil evaluasi dapat memberikan gambaran terhadap keterbatasan model saat ini.