

BAB I

PENDAHULUAN

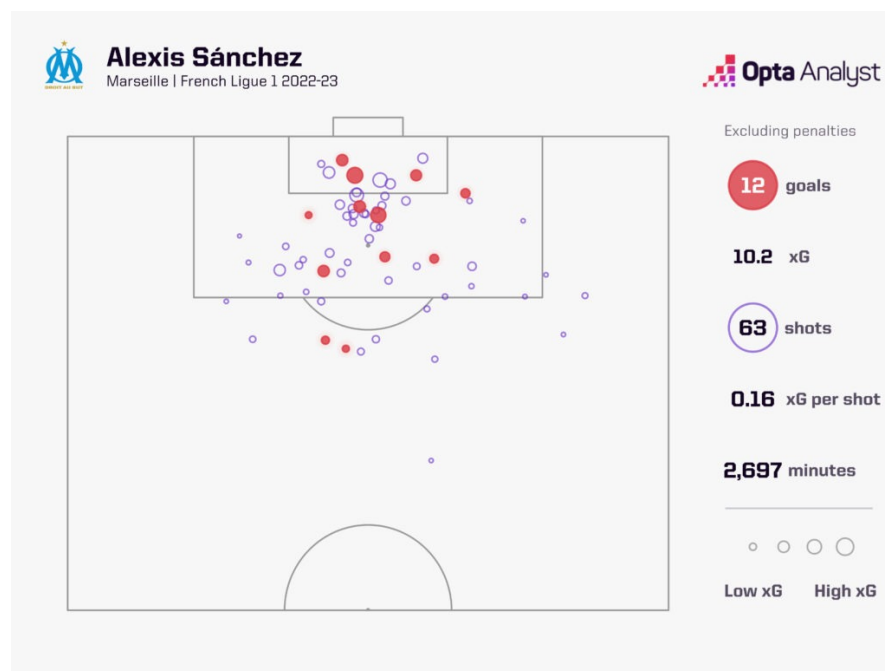
1.1 Latar Belakang

Perkembangan sepak bola menjadi industri global telah mengubah cara pandang klub, di mana pendekatan yang didasari oleh data (*data-driven*) kini menjadi kunci untuk mendapatkan keunggulan kompetitif. Dalam konteks inilah, muncul metrik analitis seperti *Expected Goals* (xG). Secara fundamental, xG adalah sebuah indikator statistik yang mengukur probabilitas sebuah tembakan untuk dikonversi menjadi gol, yang dapat membantu dalam mengoptimalkan rencana permainan dan memprediksi hasil pertandingan (Khrapach & Siryi, 2024). Menurut Cefis & Carpita (2024), xG tidak hanya merepresentasikan kualitas peluang dengan cukup akurat, tetapi juga mampu memberikan wawasan penting terhadap hasil pertandingan secara keseluruhan. Akumulasi nilai xG dari setiap pertandingan dapat digunakan untuk memprediksi hasil yang seharusnya terjadi, sehingga menjadikannya alat yang efektif untuk mengevaluasi performa tim.

Nilai xG berkisar antara 0 (peluang mustahil menjadi gol) hingga 1 (peluang yang hampir pasti menghasilkan gol) untuk mengukur kualitas sebuah peluang (Whitmore, 2023). Sebagai contoh, sebuah tembakan penalti memiliki nilai xG yang sangat tinggi, yaitu sekitar 0,76, yang berarti dari 100 tendangan penalti, diharapkan akan tercipta 76 gol (Kelly, 2019). Sebaliknya, tembakan dari jarak yang sangat jauh, seperti 40 meter, dengan banyak pemain bertahan di depannya akan

memiliki nilai xG yang sangat rendah karena probabilitas untuk menjadi gol secara historis sangat kecil.

Dalam praktiknya, metrik xG dapat divisualisasikan melalui representasi spasial seperti *shot map* yang menggambarkan lokasi dan kualitas tiap tembakan. Sebagai contoh, Gambar 1.1 memperlihatkan visualisasi *shot-map* xG.



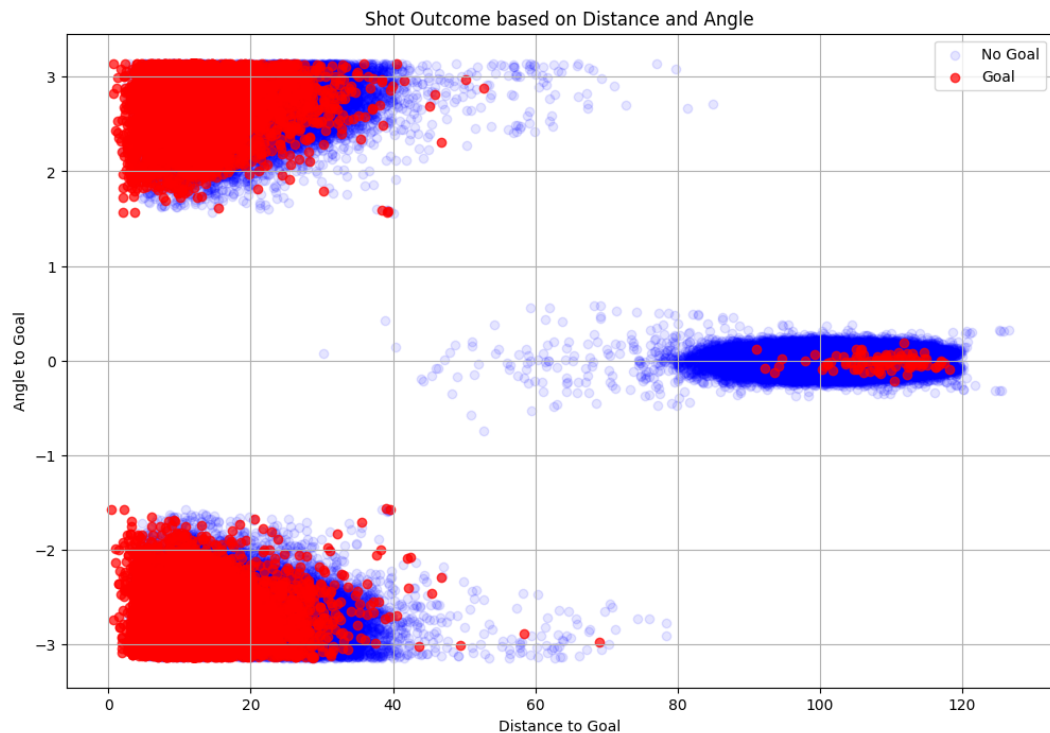
Gambar 1.1 Visualisasi *Shot-map* xG (Whitmore, 2023)

Gambar 1.1 merupakan sebuah *shot map* dari Opta yang menyajikan visualisasi data tembakan yang dilakukan oleh pemain Alexis Sánchez untuk klub Marseille pada kompetisi Liga 1 Prancis musim 2022-2023. Diagram ini memetakan semua lokasi tembakan di sepertiga akhir lapangan, dengan setiap lingkaran merepresentasikan satu tembakan. Ukuran lingkaran menunjukkan kualitas peluang atau nilai xG, di mana lingkaran yang lebih besar menandakan nilai xG yang lebih tinggi, sementara lingkaran yang terisi warna merah

menandakan tembakan yang menghasilkan gol. Berdasarkan data yang disajikan, dapat dilihat bahwa selama 2.697 menit bermain, Alexis Sánchez melepaskan total 63 tembakan (tidak termasuk penalti) dan berhasil mencetak 12 gol. Secara keseluruhan, total nilai xG yang ia kumpulkan adalah 10,2, dengan rata-rata 0,16 xG per tembakan. Tren utama yang terlihat adalah mayoritas tembakan dan semua gol Sánchez berasal dari dalam area penalti, yang merupakan area dengan probabilitas gol tinggi. Hal ini menunjukkan efektivitas pengambilan keputusan dan kemampuan *finishing* sang pemain di area berbahaya.

Akurasi model xG berperan krusial dalam keputusan strategis, baik untuk evaluasi performa pasca-pertandingan, rekrutmen pemain, maupun analisis taktik lawan. Karena itu, model yang mampu menyajikan hasil akurat dengan komputasi yang cepat dapat memberikan keunggulan kompetitif yang nyata.

Pada dasarnya, data sepak bola yang digunakan untuk pemodelan xG bersifat non-linear. Hubungan antara variabel prediktor (seperti jarak dan sudut tembakan) dengan variabel target (apakah tembakan menjadi gol atau tidak) tidak mengikuti pola garis lurus. Sebaliknya, interaksi yang kompleks antara berbagai faktor, posisi pemain bertahan, kecepatan bola, bagian tubuh yang digunakan menciptakan pola yang rumit dan hanya dapat dipetakan secara akurat oleh model yang mampu memahami hubungan non-linear. Sifat non-linear ini dapat divisualisasikan pada Gambar 1.2, yang memetakan hasil tembakan berdasarkan jarak dan sudutnya menggunakan *dataset* milik StatsBomb.



Gambar 1.2 Visualisasi Hubungan Non-Linear Data Sepak Bola (StatsBomb, 2022)

Gambar 1.2 mengilustrasikan sebaran ribuan tembakan, di mana titik merah merepresentasikan gol dan titik biru merepresentasikan tembakan yang tidak menghasilkan gol. Terlihat jelas bahwa zona di mana gol paling sering terjadi (kumpulan titik merah) membentuk sebuah kluster padat pada jarak yang dekat dengan gawang (nilai sumbu x rendah) dan sudut yang tidak terlalu sempit (nilai sumbu y mendekati nol). Pola ini tidak dapat dipisahkan secara efektif menggunakan sebuah garis lurus, yang membuktikan bahwa model linear tidak akan mampu menangkap batas keputusan yang kompleks antara gol dan non-gol.

Salah satu tantangan metodologis dalam pemodelan xG adalah keterbatasan model linear, seperti regresi logistik, dalam menangkap dinamika pertandingan yang kompleks dan non-linear (Anzer & Bauer, 2021). Superioritas model non-

linear dalam hal akurasi telah divalidasi secara konsisten dalam berbagai penelitian, yang menunjukkan bahwa mereka mampu menghasilkan prediksi yang lebih baik. Tabel 1.1 merangkum beberapa studi kunci yang menyoroti kesenjangan kinerja ini.

Tabel 1.1 Perbandingan Kinerja Model Non-Linear vs. Linear Pada Pemodelan xG

Peneliti (Tahun)	Dataset	Model Non-Linear (ROC AUC)	Model Linear (ROC AUC)
Méndez <i>et al.</i> (2023)	StatsBomb (>12.000 tembakan)	MLP (0,87)	Regresi Logistik (0,82)
Pardo (2020)	OPTA (~20.000 tembakan)	ANN (0,88)	Regresi Logistik (0,78)
Eggels <i>et al.</i> (2016)	ORTEC & Inmotio (~20.000 tembakan)	AdaBoost (0,84)	Regresi Logistik (0,78)

Tabel 1.1 secara konsisten menunjukkan bahwa model non-linear seperti MLP, ANN, dan AdaBoost secara signifikan mengungguli model linear (Regresi Logistik) dalam memprediksi xG. Pada *dataset* yang berbeda-beda, model non-linear secara konsisten mencapai nilai ROC AUC yang lebih tinggi, membuktikan kemampuannya yang lebih baik dalam menangkap hubungan kompleks pada data tembakan sepak bola.

Sebagai perbandingan dengan riset-riset sebelumnya, Tabel 1.2 menyajikan kelebihan dan kekurangan model yang diusulkan terhadap model xG *state-of-the-art*.

Tabel 1.2 Perkembangan Model xG Saat Ini

Peneliti (Tahun)	Algoritma	Dataset (Jumlah Tembakan)	Kontribusi Utama	Hasil Kinerja (ROC AUC)
Cavus & Biecek (2022)	AutoML (XGBoost, LightGBM, dll.)	315.430	Pionir penggunaan AutoML & SHAP untuk	0,873 (LightGBM)

			eksplorasi dan interpretabilitas model kompleks.	
Méndez <i>et al.</i> (2023)	<i>Multilayer Perceptron</i> (MLP)	>12.000	Menunjukkan superioritas model non-linear dalam menangkap pola non-linear dibandingkan regresi logistik.	0,87
Mead <i>et al.</i> (2023)	<i>Random Forest</i>	~250.000	Peningkatan signifikan dengan pengayaan fitur (nilai pemain, <i>rating</i> ELO).	0,91
Bandara <i>et al.</i> (2024)	<i>Random Forest</i>	Data dari 990 pertandingan	Inovasi fitur sekuensial dari 3 <i>event</i> sebelum tembakan untuk konteks yang lebih kaya.	0,833
Xu <i>et al.</i> (2025)	Jaringan Saraf Konvolusional (CNN)	477 tembakan dari <i>dataset</i> publik. + Data SoccerNet-v2: 927 tembakan	Pionir dalam menggunakan data pose/kerangka tubuh pemain (<i>skeleton data</i>) secara langsung untuk estimasi xG.	0,845

Tabel 1.2 memetakan evolusi terkini dalam pemodelan xG, di mana para peneliti secara konsisten mendorong batas akurasi melalui algoritma canggih seperti jaringan saraf konvolusional (CNN) dan inovasi fitur seperti data sekuensial atau *rating* pemain. Namun, kemajuan ini seringkali datang dengan konsekuensi, yaitu meningkatnya kompleksitas dan menurunnya efisiensi komputasi, yang menghasilkan model yang akurat tetapi cenderung lambat dan berat untuk

diimplementasikan. Tabel 1.3 menyajikan perbandingan efisiensi komputasi antara beberapa model *machine learning* pada *dataset* yang berbeda.

Tabel 1.3 Perbandingan Waktu Pelatihan Berdasarkan Kompleksitas Algoritma

Tingkat Kompleksitas	Model/Algoritma	<i>Dataset</i>	Waktu Pelatihan	Peneliti
Rendah	<i>Decision Tree</i>	Gabungan (e.g., MNIST: 70.000 data)	10,2 detik	(Bill <i>et al.</i> , 2024)
Rendah	SVM	Gabungan (e.g., MNIST: 70.000 data)	20,5 detik	(Bill <i>et al.</i> , 2024)
Menengah	<i>Random Forest</i>	Gabungan (e.g., MNIST: 70.000 data)	50,1 detik	(Bill <i>et al.</i> , 2024)
Menengah	<i>Random Forest</i>	Lalu Lintas (~2.8 juta data)	38 menit	(Manatova & Woo, 2023)
Menengah	XGBoost	Lalu Lintas (~2.8 juta data)	60 menit	(Manatova & Woo, 2023)
Tinggi	<i>Feed-Forward NN</i>	Lalu Lintas (~2.8 juta data)	120 menit	(Manatova & Woo, 2023)
Tinggi	Neural Network (Sederhana/MLP)	Gabungan (e.g., MNIST: 70.000 data)	100,8 detik	(Bill <i>et al.</i> , 2024)

Tabel 1.3 secara jelas mengilustrasikan hubungan antara kompleksitas model dan efisiensi komputasi. Terlihat tren di mana model yang lebih kompleks memerlukan waktu komputasi yang lebih lama. Pada *dataset* gabungan, waktu pelatihan meningkat secara signifikan dari model *decision tree* (10,2 detik) yang paling sederhana ke *neural networks* (100,8 detik). Pola serupa juga terlihat pada *dataset* "lalu lintas" yang lebih besar, di mana *random forest* (38 menit) lebih cepat dibandingkan *feed-forward neural network* (120 menit). Tabel ini menyoroti adanya

trade-off fundamental antara potensi akurasi dari model yang kompleks dengan biaya komputasi yang lebih tinggi.

Adapun kontribusi utama yang ditawarkan oleh penelitian ini adalah sebagai berikut:

- a. Merancang model yang dapat mencapai akurasi non-linear tanpa mengorbankan efisiensi komputasi secara signifikan.
- b. Memaksimalkan performa model secara spesifik melalui proses optimalisasi *hyperparameter tuning*.
- c. Memberikan inovasi fitur sekuensial untuk menangkap konteks jenis permainan yang mendahului tembakan.
- d. Melakukan kalibrasi model untuk memastikan nilai probabilitas xG yang dihasilkan lebih andal untuk analisis praktis.

Dalam penelitian ini, *Light Gradient Boosting Machine* (LightGBM) menjadi algoritma yang sangat potensial karena kemampuannya dalam menangani *dataset* besar, memproses fitur dalam jumlah banyak, serta membangun model prediktif non-linear dengan waktu pelatihan yang jauh lebih cepat dibandingkan metode *boosting* konvensional tanpa mengorbankan akurasi (Hartanto *et al.*, 2023).

LightGBM dikenal sebagai *framework gradient boosting* berperforma tinggi yang berbasis algoritma *decision tree*. LightGBM termasuk dalam kategori *machine learning*, karena menggunakan pendekatan *ensemble decision tree* untuk membangun model prediktif (Ke *et al.*, 2017). Sebagai implementasi dari *Gradient Boosting Decision Tree* (GBDT), algoritma ini menawarkan kecepatan pelatihan yang tinggi dan efisiensi dalam menangani *dataset* besar tanpa mengorbankan

akurasi. Keunggulan ini telah dibuktikan dalam berbagai domain, bahkan di sektor kesehatan seperti diagnosis penyakit dan prediksi klinis, di mana kebutuhan akan klasifikasi cepat dan akurat sangat penting (Artzi *et al.*, 2020).

Keunggulan performa dan efisiensi LightGBM tersebut menjadi sangat nyata ketika dihadapkan pada *dataset* dengan karakteristik non-linear yang kompleks, yang umum ditemukan dalam masalah dunia nyata. Untuk memvalidasi klaim ini secara empiris, berbagai studi ilmiah telah melakukan analisis perbandingan dengan menguji LightGBM secara langsung terhadap model-model populer lainnya seperti XGBoost dan *random forest* di berbagai domain. Sejumlah studi secara konsisten menunjukkan bahwa LightGBM lebih unggul dalam menangani data non-linear. Rangkuman terperinci dari studi-studi tersebut dapat dilihat pada Tabel 1.4.

Tabel 1.4 Perbandingan Performa LightGBM pada Data Non-Linear

Studi Kasus / <i>Dataset</i>	Model Pemingbanding	Metrik Evaluasi	Performa LightGBM	Performa Pemingbanding	Sumber (Penulis, Tahun)
Klasifikasi Indeks Pembangunan Manusia (IPM) Indonesia	XGBoost, <i>Random Forest</i>	Akurasi	0,944	0,931, 0,911	Indah <i>et al.</i> (2025)
<i>Human Activity Recognition (Sensor Smartphone)</i>	XGBoost	Akurasi	97,23%	96,67%	Türkmen & Sezen (2024)
Prediksi Waktu Pengiriman Makanan (<i>Zomato</i>)	XGBoost, <i>Random Forest</i>	R-squared (R^2)	0,76	0,71, 0,66	Garg <i>et al.</i> (2025)

<i>Benchmark</i> 12 <i>Dataset</i> Publik	CatBoost, XGBoost	<i>F1-Score</i>	0,865	0,854, 0,862	Florek & Zagdański (2023)
Prediksi Pasar Saham (Bursa Hong Kong 2021)	XGBoost	<i>Annualized Return</i>	3,29%	2,79%	Zhao <i>et al.</i> (2023)

Tabel 1.4 merangkum analisis komparatif dari lima studi kasus yang berbeda untuk memvalidasi performa superior LightGBM pada data non-linear. Hasilnya menunjukkan bahwa LightGBM secara konsisten mengungguli model-model canggih lainnya seperti XGBoost dan CatBoost di berbagai domain. Keunggulan ini terukur secara kuantitatif melalui metrik spesifik seperti Akurasi (0,944 vs 0,931), *F1-Score* (0,865 vs 0,862), *R-squared* (0,76 vs 0,71), dan bahkan metrik finansial seperti *annualized return* (3,29% vs 2,79%), yang menegaskan efektivitas dan keandalannya dalam memodelkan pola data yang kompleks.

Pada pembangunan model xG, LightGBM menawarkan kemampuan untuk belajar dari data historis dengan efisiensi tinggi. Menurut Ke *et al.* (2017), LightGBM dikembangkan untuk mengatasi keterbatasan GBDT dalam menangani *big data*, dengan waktu pelatihan yang hingga 20 kali lebih cepat namun tetap mempertahankan tingkat akurasi yang sebanding (Hartanto *et al.*, 2023). Selain itu, LightGBM menunjukkan efisiensi komputasi yang tinggi dan sensitivitas rendah terhadap *hyperparameter*, menjadikannya pilihan yang andal untuk berbagai aplikasi (Sheridan *et al.*, 2021).

Untuk mencapai efisiensi tersebut, LightGBM memperkenalkan dua inovasi utama yaitu, *Gradient-based One-Side Sampling* (GOSS) dan *Exclusive Feature Bundling* (EFB). GOSS berfungsi dengan memprioritaskan data yang memiliki

gradien besar yang menunjukkan kesalahan prediksi tinggi dan mengabaikan sebagian data dengan gradien kecil untuk mengurangi beban komputasi tanpa kehilangan informasi penting. Di sisi lain, EFB bertujuan mengurangi dimensi fitur dengan cara menggabungkan fitur-fitur yang saling eksklusif (tidak aktif bersamaan) ke dalam satu kelompok fitur baru (Bentéjac *et al.*, 2020).

Salah satu keunggulan utama yang ditawarkan oleh LightGBM adalah efisiensi dan kecepatan komputasi yang tinggi, terutama saat menangani *dataset* dalam skala besar. Untuk memvalidasi klaim ini, sebuah penelitian *benchmark* terkini melakukan serangkaian eksperimen untuk membandingkan waktu komputasi (*runtime*) antara LightGBM dengan implementasi *gradient boosting* populer lainnya, yaitu GBM, XGBoost, dan CatBoost. Sebagaimana disajikan pada Tabel 1.5, hasil perbandingan tersebut menunjukkan keunggulan kecepatan LightGBM di berbagai skenario.

Tabel 1.5 Perbandingan Waktu Pelatihan pada Model *Boosting* (Florek & Zagdański, 2023)

<i>Dataset</i>	LightGBM (detik)	GBM (detik)	XGBoost (detik)	CatBoost (detik)
<i>Heart Disease</i>	0,583	1,417	60,449	4,143
<i>Mushrooms</i>	6,432	11,659	45,696	13,889
<i>Breast Cancer</i>	0,722	5,296	85,364	53,840
Leukemia	3,010	32,633	51,568	375,332
<i>Credit Card Fraud</i>	11,541	367,291	119,502	123,376
<i>Gina Agnostic</i>	27	90,250	129,121	98,137

Data pada Tabel 1.5 secara konsisten menunjukkan bahwa LightGBM memiliki keunggulan kecepatan komputasi yang signifikan. Sebagai contoh, pada *dataset* leukemia, LightGBM hanya membutuhkan 3,01 detik, jauh mengungguli

GBM (32,6 detik) dan XGBoost (51,6 detik). Keunggulan ini juga terlihat jelas pada *dataset* lain seperti *credit card fraud*, di mana LightGBM (sekitar 11,5 detik) terbukti lebih dari 10 kali lebih cepat daripada XGBoost (119,5 detik) dan CatBoost (123,4 detik). Hasil ini menegaskan reputasi LightGBM sebagai salah satu implementasi *gradient boosting* tercepat dan paling efisien untuk berbagai kasus penggunaan (Florek & Zagdański, 2023).

Perbedaan utama antara LightGBM dan *gradient boosting* lain seperti XGBoost terletak pada cara masing-masing algoritma meningkatkan performa *gradient boosting*. XGBoost melakukan pemrosesan secara paralel dengan memanfaatkan banyak inti pada *Central Processing Unit* (CPU) melalui distribusi perhitungan, optimalisasi *cache*, dan kemampuan *out-of-core processing*. Sementara itu, LightGBM menerapkan strategi pertumbuhan pohon *leaf-wise*, bukan *level-wise* seperti XGBoost, yang membuatnya lebih efisien dalam menemukan *split* dengan *loss* terkecil dan lebih cepat dalam proses pelatihan (Chen *et al.*, 2019).

Keunggulan efisiensi komputasi ini tidak hanya berhenti pada metrik teknis, tetapi juga dapat diterjemahkan menjadi keunggulan kompetitif dan finansial yang nyata. Dalam industri sepak bola yang dinamis, kecepatan dalam mengolah data menjadi faktor krusial yang membuka berbagai peluang strategis mulai dari analisis siaran langsung hingga efisiensi biaya operasional yang sebelumnya tidak dapat dijangkau oleh model tradisional yang lebih lambat. Untuk mengilustrasikan dampak praktis dari efisiensi ini, Tabel 1.6 menunjukkan bagaimana kecepatan LightGBM memberikan solusi konkret pada berbagai skenario bisnis.

Tabel 1.6 Keunggulan Bisnis dari Efisiensi Komputasi LightGBM dalam Skenario Praktis

Skenario Aplikasi & Bisnis	Tantangan dengan Model Lain	Solusi yang Ditawarkan LightGBM	Dampak Bisnis & Finansial
Analisis Siaran Langsung	Performa lambat, tidak bisa tayangan <i>live</i> .	Analisis instan selaras dengan tayangan <i>live</i> ..	Konten siaran lebih premium & menarik sponsor.
Pembaruan & Pelatihan Model	Pelatihan ulang pada model adaptasi data baru sangat lambat.	Pelatihan ulang pada model adaptasi data baru sangat cepat.	Menguntungkan tim analis dengan model yang selalu <i>up-to-date</i> secara cepat.
Efisiensi Biaya Infrastruktur	Butuh CPU/GPU mahal & konsumsi energi tinggi.	Efisien pada CPU standar & konsumsi energi rendah.	Penghematan biaya operasional & investasi infrastruktur.

Dari Tabel 1.6 didapatkan bahwa efisiensi komputasi LightGBM secara fundamental mengubah peran analisis data dari yang semula bersifat reaktif menjadi proaktif. Kemampuannya untuk memberikan analisis instan memungkinkan media menyajikan konten siaran yang lebih premium. Bagi tim analis, kecepatan pelatihan ulang memberdayakan mereka dengan data yang selalu *up-to-date* untuk keputusan strategis yang lebih baik. Terakhir, dari sisi operasional, efisiensi LightGBM pada CPU standar menawarkan penghematan biaya infrastruktur yang signifikan, membuat analisis canggih lebih mudah diakses.

Penelitian ini menggunakan data yang bersumber dari StatsBomb *Open Data*, sebuah *dataset* publik yang secara resmi dirilis oleh perusahaan StatsBomb untuk mendorong kegiatan penelitian akademik dan pengembangan analisis dalam dunia sepak bola. *Dataset* ini tersedia untuk publik dan mencakup berbagai liga serta kompetisi ternama. Penggunaan *dataset* ini selaras dengan misi StatsBomb dalam "*encouraging academic research and analysis through open access to high-quality football data*" (StatsBomb, 2022).

StatsBomb merupakan perusahaan penyedia data olahraga berbasis riset yang didirikan oleh para analis sepak bola profesional. Dengan visi untuk menyajikan data yang paling komprehensif dan presisi, StatsBomb telah menjadi rujukan utama dalam banyak riset akademik dan industri berkat kedalaman data serta platform fleksibel yang mereka kembangkan (StatsBomb, 2024). Gambar 1.3 berikut menampilkan logo resmi dari perusahaan StatsBomb yang menjadi sumber data utama dalam penelitian ini.



Gambar 1.3 Logo Statsbomb

Berdasarkan latar belakang serta hasil dari penelitian-penelitian sebelumnya, penulis menyimpulkan bahwa terdapat kebutuhan untuk mengembangkan model xG dengan algoritma yang lebih efisien dan juga akurat. LightGBM, dengan kemampuan dan keunggulannya dalam menangani *big data* khusus nya data non-linear pada sepak bola, menawarkan peluang untuk menghasilkan model yang lebih baik dibandingkan model atau algoritma lain yang telah diterapkan. Oleh karena itu, tugas akhir ini dilakukan sebagai upaya inovatif dalam analisis sepak bola dengan mengimplementasikan LightGBM untuk xG. Dengan demikian, tugas akhir ini disusun dengan judul: **"PENERAPAN *LIGHT GRADIENT BOOSTING MACHINE (LIGHTGBM)* UNTUK PREDIKSI NILAI *EXPECTED GOALS (XG)* DALAM ANALISIS SEPAK BOLA"**

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan, berikut merupakan identifikasi masalah pada penelitian ini:

- a. Ketidakmampuan algoritma linear tradisional dalam merepresentasikan hubungan non-linear pada data sepak bola menyebabkan rendahnya performa dan akurasi model xG, sehingga memerlukan pendekatan alternatif yang lebih unggul.
- b. Meskipun model non-linear yang kompleks mampu menangkap pola rumit dalam data sepak bola, seringkali hal ini berimplikasi pada waktu komputasi yang tinggi sehingga menjadi tantangan dalam efisiensi analisis performa secara cepat.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah yang telah dipaparkan, berikut merupakan rumusan masalah pada penelitian ini:

- a. Bagaimana penerapan algoritma LightGBM untuk meningkatkan performa akurasi dalam perhitungan xG dalam analisis sepak bola yang memiliki hubungan non-linear, di mana performa akurasi diukur menggunakan metrik ROC AUC, *Brier Score*, akurasi, presisi, *recall*, *F1-Score*, dan *Log-Loss*?
- b. Bagaimana efisiensi komputasi dari algoritma LightGBM dalam perhitungan xG yang diukur melalui waktu komputasi?

1.4 Batasan Masalah

Batasan masalah yang terdapat pada penelitian ini yaitu:

- a. Penelitian ini hanya berfokus pada implementasi LightGBM untuk perhitungan xG dalam analisis sepak bola.
- b. Analisis sepak bola dibatasi hanya pada data tembakan (*shot event*) dalam penerapan xG, dan tidak mencakup analisis terhadap teknis permainan lainnya.
- c. Data yang digunakan diambil dari StatsBomb *open-data* yang berlisensi resmi oleh StatsBomb Services Ltd yang berkantor pusat di University of Bath Innovation Centre, Carpenter House, Broad Quay, Bath, BA1 1UD.
- d. Data terbatas pada *event* data statistik pertandingan, termasuk posisi, jarak, teknik, sudut tembakan dan lainnya.
- e. Penelitian ini fokus pada perhitungan xG menggunakan LightGBM tanpa membandingkan dengan model lain.
- f. *Preprocessing* dilakukan menggunakan *Python*, fokus pada pembersihan dan transformasi data.
- g. Data dibagi untuk *training* dan *testing* dengan validasi silang hanya pada kalibrasi model.
- h. Evaluasi performa model terbatas pada pengukuran akurasi menggunakan metrik ROC AUC, Brier Score, akurasi, presisi, *recall*, *F1-Score*, dan *Log-Loss*, serta pengukuran efisiensi komputasi berdasarkan waktu pemrosesan.

1.5 Tujuan Penelitian

Tujuan dilakukannya penelitian ini adalah sebagai berikut:

- a. Penerapan LightGBM dalam upaya meningkatkan akurasi dan efisiensi perhitungan metrik xG pada analisis data sepak bola yang memiliki hubungan

non-linear dengan mengukur akurasi prediksi melalui metrik ROC AUC, *Brier Score*, akurasi, presisi, *recall*, *F1-Score*, dan *Log-Loss*.

- b. Evaluasi efisiensi komputasi performa model LightGBM dengan mengukur berdasarkan waktu komputasi.

1.6 Manfaat Penelitian

Manfaat dari penelitian ini yaitu sebagai berikut:

- a. Bagi peneliti, penelitian ini merupakan implementasi dari teori yang telah dipelajari dalam bidang analisis data dan *machine learning*, sehingga dapat lebih memahami penerapan algoritma LightGBM dalam perhitungan nilai xG. Selain itu, penelitian ini juga merupakan salah satu syarat kelulusan Strata Satu (S1) Sistem Informasi UIN Syarif Hidayatullah Jakarta.
- b. Bagi Universitas, penelitian ini dapat dijadikan sebagai tolak ukur pengetahuan mahasiswa terkait penerapan algoritma *machine learning* dalam analisis sepak bola, serta sebagai kontribusi dalam pengembangan penelitian di bidang ilmu komputer dan sistem informasi.
- c. Bagi pembaca, penelitian ini dapat memberikan informasi yang komprehensif mengenai algoritma LightGBM dan aplikasinya dalam perhitungan xG, serta dapat dijadikan sebagai referensi tambahan terkait penelitian dalam Program Studi Sistem Informasi UIN Syarif Hidayatullah Jakarta, khususnya dalam konteks analisis data olahraga. Penelitian ini juga dapat memberikan pemahaman tentang pentingnya analisis data dalam pengambilan keputusan dalam sepak bola.

1.7 Metode Penelitian

Metode penelitian ini dibagi menjadi dua bagian, yaitu:

a. Pengumpulan Data

1) Studi Literatur

Metode studi literatur dilakukan dengan mengumpulkan dan menganalisis berbagai sumber tertulis, seperti buku, artikel ilmiah, dan laporan penelitian yang relevan dengan topik penelitian.

2) *Data Extraction*

Data extraction adalah proses pengambilan data dari berbagai sumber untuk dianalisis lebih lanjut. Dalam penelitian ini, data yang digunakan diambil dari StatsBomb *open-data* yang tersedia di GitHub dengan lisensi resmi.

b. Analisis Data

Penelitian ini menggunakan metode *data mining* yang dikenal sebagai *Knowledge Discovery in Databases* (KDD). Metode KDD terdiri atas beberapa tahap yang saling berhubungan, sebagai berikut:

1) *Data Selection*

Data selection adalah proses pemilihan sub set data yang relevan dari kumpulan data yang lebih besar untuk analisis lebih lanjut. Dalam penelitian ini, pemilihan data difokuskan pada informasi yang terkait dengan tembakan dan peluang gol, sehingga dapat digunakan dalam perhitungan metrik xG.

2) *Data Preprocessing*

Data preprocessing adalah langkah yang dilakukan untuk menyiapkan dan membersihkan data sebelum analisis. Ini melibatkan penghapusan data yang tidak relevan, pengisian nilai yang hilang, dan pengubahan format data agar sesuai dengan kebutuhan analisis. Tahap ini penting untuk memastikan bahwa data yang digunakan dalam penelitian akurat dan dapat diandalkan.

3) *Data Transformation*

Data transformation adalah proses mengubah data ke dalam format yang lebih sesuai untuk analisis. Ini termasuk teknik seperti normalisasi, pengkodean variabel kategorial, dan agregasi data. Proses ini memungkinkan model *machine learning* untuk memproses data dengan lebih efisien dan efektif.

4) *Data Mining*

Pada tahap *data mining*, penelitian ini menggunakan algoritma LightGBM untuk membangun model prediktif berdasarkan data yang telah diproses. LightGBM dipilih karena kemampuannya dalam menangani data besar dengan efisiensi tinggi, serta akurasi yang dihasilkannya dalam perhitungan xG.

5) *Evaluation*

Setelah model dibangun, evaluasi performa dilakukan secara komprehensif menggunakan serangkaian metrik. Metrik utama yang digunakan adalah ROC AUC untuk mengukur kemampuan diskriminasi dan *Brier Score* untuk mengukur akurasi probabilitas. Selain itu, evaluasi juga dilengkapi dengan metrik klasifikasi standar seperti akurasi, presisi, *recall*, F1-Score, serta

metrik kesalahan tambahan yaitu *Log-Loss*. Untuk melengkapi analisis, efisiensi komputasi juga dievaluasi dengan mengukur waktu yang dibutuhkan untuk pemrosesan, sehingga diperoleh gambaran performa yang *robust* dan menyeluruh.

1.8 Sistematika Penulisan

Laporan pada penelitian ini terdiri atas lima bab, yaitu:

BAB 1 PENDAHULUAN

Bab ini membahas tentang latar belakang, identifikasi masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan dari penelitian ini.

BAB 2 TINJAUAN PUSTAKA

Bab ini membahas tentang teori-teori yang berkaitan dengan metrik xG dalam sepak bola, serta penerapan algoritma LightGBM dalam model prediksi, termasuk tinjauan mengenai penelitian-penelitian terdahulu yang relevan.

BAB 3 METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tahapan metode yang digunakan dalam penelitian, meliputi metode pengumpulan data, proses *preprocessing*, analisis data, dan implementasi menggunakan algoritma LightGBM, serta tahapan evaluasi.

BAB 4 HASIL DAN PEMBAHASAN

Bab ini berisi hasil dari penerapan algoritma LightGBM dalam perhitungan metrik xG, serta analisis mendalam mengenai kinerja model berdasarkan evaluasi yang dilakukan. Hasil juga dibandingkan dengan model lain untuk menunjukkan efektivitas LightGBM.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian mengenai penerapan algoritma LightGBM dalam perhitungan metrik xG, serta saran-saran yang dapat digunakan untuk penelitian selanjutnya dalam bidang analisis sepak bola dan penerapan *machine learning*.

BAB II

TINJAUAN PUSTAKA

2.1 *Data Mining*

Data dalam jumlah besar yang terus terakumulasi sering kali menyimpan informasi dan pola tersembunyi yang sangat berharga. Namun, besarnya volume data membuat analisis manual menjadi tidak mungkin dilakukan. Di sinilah peran *data mining* menjadi sangat penting. *Data mining* adalah proses penemuan pola, anomali, dan korelasi yang menarik dan bermanfaat dari kumpulan data berskala besar untuk memprediksi hasil di masa depan (Han *et al.*, 2022).

Tujuan utama dari *data mining* adalah untuk mengubah data mentah (*raw data*) menjadi pengetahuan yang dapat ditindaklanjuti (*actionable knowledge*). Proses ini tidak hanya sekadar mengekstraksi data, tetapi juga melibatkan penggunaan teknik dari disiplin ilmu lain seperti statistika, kecerdasan buatan (*artificial intelligence*), dan *machine learning* untuk mengidentifikasi tren yang sebelumnya tidak diketahui (Tan *et al.*, 2019). Dengan menemukan pola-pola tersebut, sebuah organisasi dapat memperoleh wawasan strategis, meningkatkan efisiensi operasional, dan membuat keputusan yang lebih baik berdasarkan bukti data.

Secara fungsional, tugas-tugas dalam *data mining* dapat dikategorikan menjadi dua jenis utama: prediktif dan deskriptif. Tugas prediktif bertujuan untuk memprediksi nilai dari suatu atribut tertentu berdasarkan nilai dari atribut lainnya, sedangkan tugas deskriptif bertujuan untuk menemukan pola yang menggambarkan

data dan dapat ditafsirkan oleh manusia (Tan *et al.*, 2019). Beberapa tugas utama dalam *data mining* yaitu:

a. Klasifikasi (*Classification*)

Klasifikasi adalah salah satu tugas prediktif yang paling umum. Klasifikasi bertujuan untuk membangun sebuah model yang dapat memetakan suatu objek data ke dalam salah satu dari beberapa kelas yang telah ditentukan sebelumnya. Model ini dibangun berdasarkan analisis dari sekumpulan data latih yang label kelasnya sudah diketahui. Contoh penerapannya adalah mengklasifikasikan email sebagai "*spam*" atau "bukan *spam*", atau menentukan apakah seorang nasabah bank berisiko tinggi atau rendah untuk kredit macet (Han *et al.*, 2022).

b. Regresi/Prediksi (*Regression/Prediction*)

Serupa dengan klasifikasi, regresi juga merupakan tugas prediktif. Perbedaannya terletak pada *output* yang dihasilkan. Jika klasifikasi memprediksi label kelas yang bersifat kategoris (diskrit), maka regresi memprediksi nilai yang bersifat kontinu (numerik). Contohnya adalah memprediksi harga sebuah rumah berdasarkan luas bangunan, jumlah kamar, dan lokasi, atau meramalkan angka penjualan produk pada kuartal berikutnya (Zaqy *et al.*, 2023).

c. *Clustering*

Berbeda dengan klasifikasi dan regresi, *clustering* atau klasterisasi adalah tugas deskriptif. Tujuannya adalah untuk mengelompokkan sekumpulan objek data ke dalam beberapa grup atau *cluster* sedemikian rupa sehingga objek-objek dalam satu *cluster* memiliki tingkat kemiripan yang tinggi, sementara objek-

objek di *cluster* yang berbeda memiliki tingkat kemiripan yang rendah. Dalam *clustering*, label kelas dari data tidak diketahui sebelumnya. Contoh aplikasinya adalah segmentasi pelanggan berdasarkan pola pembelian untuk strategi pemasaran yang lebih tertarget (Tan *et al.*, 2019).

2.2 Klasifikasi Probabilistik (*Probabilistic Classification*)

Klasifikasi probabilistik adalah salah satu pendekatan inti dalam *machine learning* yang bertujuan untuk memetakan data masukan ke label kelas berdasarkan probabilitas (Murphy, 2022). Berbeda dari klasifikasi yang hanya menghasilkan satu label kelas tunggal sebagai *output*, klasifikasi probabilistik bekerja dengan mengestimasi distribusi probabilitas untuk seluruh kelas yang memungkinkan bagi setiap data input (Vaddella & Hosseinzadeh, 2021). Pada dasarnya, pendekatan ini tidak hanya menjawab pertanyaan "data ini termasuk kelas apa?", melainkan juga "seberapa besar kemungkinan data ini termasuk dalam setiap kelas?". Kemampuan untuk mengukur ketidakpastian (*quantifying uncertainty*) ini merupakan keunggulan utamanya, yang menjadikannya sangat krusial dalam domain yang memerlukan pengambilan keputusan berbasis risiko, seperti pada diagnosis medis, penyaringan spam, dan penilaian kredit (Murphy, 2022).

Landasan matematis dari sebagian besar model klasifikasi probabilistik adalah Teorema Bayes. Teorema ini menyediakan cara untuk memperbarui keyakinan (*belief*) kita terhadap sebuah hipotesis berdasarkan bukti (*evidence*) baru yang diperoleh. Dalam konteks klasifikasi, tujuan utamanya adalah untuk menghitung probabilitas posterior, yaitu probabilitas suatu data termasuk dalam

kelas y setelah fitur x diketahui $P(y | x)$. Rumus teorema Bayes dinyatakan pada persamaan (2.1) (Russell & Norvig, 2020).

$$P(y | x) = \frac{P(x | y)P(y)}{P(x)} \quad (2.1)$$

Landasan dari klasifikasi probabilistik terletak pada perhitungan probabilitas posterior $P(y | x)$, yaitu probabilitas sebuah data x benar-benar termasuk dalam kelas y setelah data tersebut diamati. Untuk menghitungnya, teorema Bayes memanfaatkan tiga komponen utama. Pertama adalah *likelihood* $P(x | y)$, yang merepresentasikan probabilitas mengamati data x dengan asumsi berasal dari kelas y , di mana komponen ini dimodelkan dari data latih. Kedua adalah probabilitas *prior* $P(y)$, yang merupakan probabilitas awal dari setiap kelas sebelum data diobservasi, sering kali didasarkan pada frekuensi kelas dalam data latih. Komponen terakhir adalah *evidence* $P(x)$, yang berfungsi sebagai faktor normalisasi untuk memastikan total probabilitas posterior di semua kelas berjumlah satu.

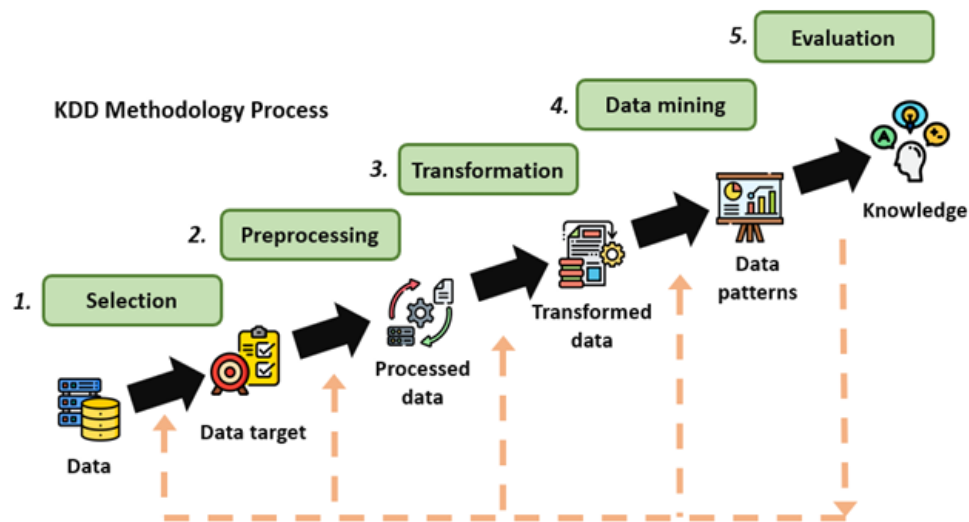
Berdasarkan cara pemodelan komponen-komponen tersebut, pendekatan klasifikasi probabilistik dapat dibagi menjadi dua paradigma utama: generatif dan diskriminatif. Model generatif secara eksplisit memodelkan *likelihood* $P(x | y)$ dan *prior* $P(y)$, sehingga secara konseptual mampu "membangkitkan" data baru untuk setiap kelas. Karena faktor *evidence* $P(x)$ bernilai sama untuk semua kelas, prediksi seringkali disederhanakan menjadi $\operatorname{argmax}_y P(x | y)P(y)$. Sebaliknya, model diskriminatif tidak memodelkan distribusi data asli, melainkan langsung memodelkan probabilitas posterior $P(y | x)$ sebagai fungsi dari data input x .

Pendekatan ini berfokus langsung pada penentuan batas keputusan antar kelas (Géron, 2020).

2.3 *Knowledge Discovery in Databases (KDD)*

Knowledge Discovery in Databases atau KDD adalah proses yang bertujuan untuk mengekstraksi informasi yang dapat dipahami, menarik, dan bernilai dari data yang tidak terstruktur (Solanki & Sharma, 2021). Proses ini digunakan di berbagai bidang, seperti ilmu kehidupan, perdagangan, keuangan, dan kedokteran, untuk mengidentifikasi pola-pola yang tersembunyi dalam data yang besar dan kompleks (Solanki & Sharma, 2021). Proses ini mencakup berbagai teknik dan metode yang dapat digunakan untuk menggali wawasan dari data yang belum terorganisir.

KDD merupakan suatu bidang yang mengandalkan metode cerdas dalam *data mining* untuk menemukan pola-pola yang menjadi inti pengetahuan (Balkir, & El-Mouadib, 2021). Pola-pola ini memungkinkan pengguna untuk memahami informasi yang terkandung dalam *dataset* besar, memberikan wawasan yang dapat diterapkan untuk pengambilan keputusan yang lebih baik dalam berbagai disiplin ilmu, dan alur prosesnya diilustrasikan pada Gambar 2.1.



Gambar 2.1 Proses KDD (Paucar & Andrade-Arenas, 2025)

Proses KDD, seperti yang diilustrasikan pada Gambar 2.1, merupakan alur kerja sistematis yang terdiri dari beberapa tahapan penting untuk mengubah data mentah menjadi pengetahuan yang berguna (Paucar & Andrade-Arenas, 2025). Tahapan-tahapan tersebut adalah:

- a. Seleksi (*Selection*): Tahap awal di mana *subset* data yang relevan dengan tujuan analisis dipilih dari kumpulan data yang besar. Ini melibatkan pemilihan variabel dan sampel data yang akan menjadi fokus utama dalam proses penemuan pengetahuan.
- b. Pra-pemrosesan (*Preprocessing*): Pada tahap ini, data yang telah dipilih dibersihkan untuk memastikan kualitasnya. Aktivitas yang dilakukan meliputi penanganan data yang hilang (*missing values*), penghapusan *noise* atau data yang tidak konsisten, dan perbaikan kesalahan data.

- c. Transformasi (*Transformation*): Data yang sudah bersih diubah atau dikonsolidasikan ke dalam format yang sesuai untuk proses penambangan data. Ini bisa mencakup normalisasi data, agregasi, atau pembuatan atribut baru (dikenal sebagai *feature engineering*) untuk meningkatkan efektivitas analisis.
- d. Penambangan Data (*Data Mining*): Ini adalah tahap inti di mana berbagai metode dan algoritma cerdas (seperti klasifikasi, *clustering*, atau analisis asosiasi) diterapkan pada data yang telah ditransformasi untuk mengidentifikasi pola-pola yang berpotensi menarik dan bermanfaat.
- e. Evaluasi dan Interpretasi (*Evaluation*): Pola-pola yang ditemukan pada tahap sebelumnya dievaluasi untuk memverifikasi validitasnya dan diinterpretasikan untuk menjadi pengetahuan. Hanya pola yang dianggap signifikan, baru, dan dapat ditindaklanjuti yang akan disajikan kepada pengguna sebagai hasil akhir.

Dalam KDD, *machine learning* berperan penting untuk menganalisis data, mengenali korelasi, dan memprediksi hasil yang akan terjadi (Kodati & Selvaraj, 2021). Teknik-teknik *machine learning* digunakan untuk melatih model dalam mengidentifikasi pola-pola yang ada dalam data, yang kemudian dapat digunakan untuk membuat prediksi yang lebih akurat dalam berbagai aplikasi, seperti analisis kesehatan atau analisis perilaku konsumen.

Aplikasi KDD sangat luas, salah satunya adalah dalam bidang kesehatan, di mana KDD digunakan untuk mengembangkan sistem medis yang dapat mendeteksi dan memberikan saran pengobatan untuk penyakit dengan upaya minimal

(Nwankwo, Ngene, & Onuora, 2023). Selain itu, KDD berbasis metode *gradient boosting machine* juga diterapkan dalam prediksi energi listrik, memberikan referensi praktis bagi aplikasi KDD pada sektor energi lainnya (Xie *et al.*, 2022).

KDD juga memiliki keterkaitan yang erat dengan analisis olahraga, khususnya sepak bola, di mana pendekatan KDD yang komprehensif memungkinkan persiapan data yang tepat untuk prediksi hasil pertandingan olahraga, termasuk hasil pertandingan sepak bola (Głowania, Kozak, & Juszcuk, 2023). Dengan menggunakan teknik KDD, analisis yang lebih mendalam dapat dilakukan terhadap data pertandingan untuk mengidentifikasi faktor-faktor yang mempengaruhi hasil akhir pertandingan.

2.4 *Exploratory Data Analysis (EDA)*

Analisis data eksploratif atau *Exploratory Data Analysis* (EDA) merupakan sebuah pendekatan fundamental dan tahap krusial dalam siklus hidup ilmu data (*data science life cycle*). EDA bukan sekadar serangkaian pemeriksaan data awal, melainkan sebuah filosofi dan strategi investigasi yang sistematis untuk menganalisis himpunan data guna merangkum karakteristik utamanya, sering kali dengan menggunakan metode grafis statistik dan teknik visualisasi lainnya (Fikri *et al.*, 2023). Pendekatan ini, yang dipelopori oleh matematikawan Amerika John W. Tukey, menekankan pentingnya mengeksplorasi data untuk melihat apa yang dapat diungkapkannya di luar pemodelan formal atau pengujian hipotesis tradisional (Regin & Rajesh, 2024). Dalam praktiknya, EDA adalah proses interaktif di mana analis mengajukan pertanyaan tentang data, memvisualisasikan, mentransformasi,

dan memodelkannya untuk mengungkap wawasan, mengidentifikasi anomali, dan menginformasikan langkah-langkah analisis selanjutnya (Regin & Rajesh, 2024).

Sebagai langkah kritis pertama, EDA memiliki serangkaian tujuan yang saling terkait dan dirancang untuk memastikan validitas dan kekokohan seluruh proses penelitian (Patel & Patel, 2024). Kegagalan dalam melakukan EDA secara menyeluruh dapat menyebabkan pemilihan model yang tidak tepat, kesimpulan yang keliru, dan pada akhirnya, penelitian yang tidak valid. Tujuan-tujuan utama dari EDA meliputi (Gelman *et al.*, 2020):

- a. Menilai kualitas dan integritas data
- b. Mendeteksi *outlier* dan anomali
- c. Memeriksa asumsi statistik
- d. Menemukan pola dan struktur
- e. Membangkitkan hipotesis
- f. Mendukung pemilihan model dan rekayasa fitur (*feature engineering*)

Dengan menguraikan rencana EDA yang jelas dan terstruktur, penelitian ini menunjukkan komitmen terhadap praktik analisis data yang teliti dan bertanggung jawab, yang merupakan fondasi untuk menghasilkan temuan yang valid dan dapat diandalkan.

2.5 *Machine Learning*

Machine Learning (ML) merupakan kemampuan suatu sistem untuk belajar dari data pelatihan yang spesifik terhadap masalah tertentu, dengan tujuan untuk mengotomatisasi proses pembangunan model analitik serta memecahkan tugas-

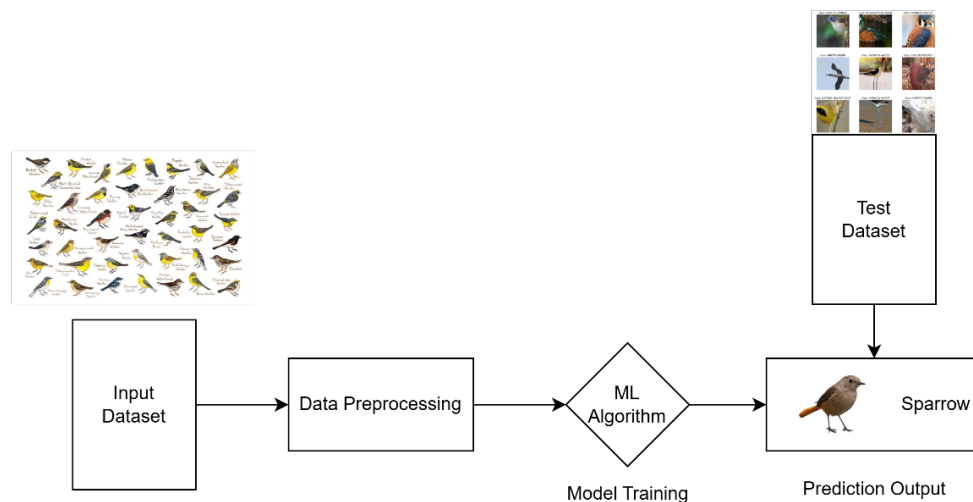
tugas terkait. Dalam konteks ini, ML memungkinkan sistem komputer untuk mengidentifikasi pola dalam data tanpa campur tangan manual yang intensif, sehingga memungkinkan solusi otomatis terhadap berbagai masalah kompleks berbasis data (Janiesch *et al.*, 2021).

Secara lebih mendalam, ML dapat dilihat sebagai bentuk *Artificial Intelligence* (AI) yang memanfaatkan data untuk melatih komputer dalam melakukan berbagai tugas tertentu, menggunakan algoritma untuk membangun serangkaian aturan secara otomatis. Proses ini memungkinkan sistem untuk secara mandiri mengenali pola serta membuat keputusan berdasarkan data tanpa perlu diinstruksikan secara eksplisit, yang pada akhirnya meningkatkan ketepatan dan efisiensi sistem dalam memecahkan masalah kompleks (Janiesch *et al.*, 2021).

ML berbeda dari *data mining* dan statistik tradisional, baik dalam aspek filosofis maupun metodologis. Terdapat tiga pendekatan utama dalam ML yang membedakannya, yaitu statistika klasik, teori pembelajaran statistik Vapnik, serta teori pembelajaran komputasional (Kodama *et al.*, 2023). Ketiga pendekatan ini menyediakan dasar yang berbeda untuk pengembangan algoritma, dimana ML fokus pada kemampuan untuk terus memperbaiki kinerja model berdasarkan data pelatihan, dibandingkan hanya melakukan analisis data historis sebagaimana dalam statistik tradisional.

Terdapat berbagai kategori dalam ML, meliputi *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Masing-masing pendekatan ini memiliki teknik-teknik unik, seperti *zero-shot learning*, *active learning*, *contrastive learning*, *self-supervised learning*, dan *semi-supervised learning* (Mahadevkar *et*

al., 2022). Pada Gambar 2.2, ditunjukkan contoh implementasi *supervised learning*, di mana model dilatih menggunakan data berlabel untuk dapat mengklasifikasikan atau memprediksi berdasarkan pola yang telah dikenali. Teknik-teknik ini memperkaya cara sistem mempelajari data visual, baik dengan data yang memiliki label atau tanpa label.



Gambar 2.2 Contoh Implementasi *Supervised Learning* (Mahadevkar *et al.*, 2022)

Algoritma dasar dalam ML sangat beragam, mencakup *decision tree*, *Random Forest*, *artificial neural network*, *Support Vector Machine* (SVM), serta algoritma *boosting* dan *bagging*, yang membantu dalam meningkatkan kinerja model dengan menggabungkan prediksi dari beberapa model. Selain itu, algoritma *backpropagation* (BP) berperan penting dalam *neural networks* untuk mengoptimalkan bobot model berdasarkan kesalahan yang dihasilkan pada prediksi awal, sehingga meningkatkan kemampuan sistem dalam memprediksi hasil dengan lebih akurat (Jin, 2020).

Dalam ML, metrik evaluasi adalah instrumen logis dan matematis yang digunakan untuk mengukur seberapa dekat hasil prediksi model terhadap nilai aktualnya. Metrik evaluasi memungkinkan analisis kinerja model secara mendalam, sehingga aspek seperti akurasi, kesalahan, dan ketepatan dalam memprediksi dapat diukur secara kuantitatif. Hal ini penting untuk memahami performa model dan menentukan langkah-langkah penyempurnaan lebih lanjut dalam pengembangan model (Plevris *et al.*, 2022).

Beberapa metrik evaluasi yang paling sering digunakan dalam ML mencakup *Root Mean Squared Error* (RMSE), *Mean Absolute Error* (MAE), *Pearson correlation coefficient*, dan *coefficient of determination* (R^2) (Plevris *et al.*, 2022). Metrik-metrik ini membantu dalam mengukur seberapa akurat dan presisi prediksi model terhadap data yang diujikan, sehingga para praktisi dapat memilih metrik evaluasi yang paling relevan dengan konteks data dan tujuan analisis mereka.

2.6 *Gradient Boosting*

Gradient boosting merupakan teknik *machine learning* yang sangat efektif dan sering digunakan untuk menangani tugas dengan fitur heterogen serta data yang cenderung *noise*. Teknik ini bekerja dengan menggabungkan prediksi dari sejumlah model sederhana atau *weak learners* untuk menghasilkan prediksi yang kuat. Dalam klasifikasi, *Gradient boosting* menghasilkan distribusi pada label kelas, sementara dalam regresi, model ini memberikan prediksi nilai tunggal atau *point prediction* untuk mendekati hasil yang diinginkan. Kemampuan *gradient boosting* dalam

menghadapi variasi pada fitur dan ketidakpastian dalam data menjadikannya alat yang sangat kuat dalam berbagai aplikasi *machine learning* (Ustimenko & Prokhorenkova, 2021).

Proses *gradient boosting* dimulai dengan mengombinasikan *weak learners*, yaitu model yang performanya sedikit lebih baik dari prediksi acak, untuk membentuk *strong learner* secara iteratif. *gradient boosting* merupakan algoritma *boosting* yang dirancang khusus untuk masalah regresi.

Dalam algoritma ini, diberikan kumpulan data pelatihan $D = \{x_i, y_i\}_1^N$, dengan tujuan utama mencari estimasi $\hat{F}(x)$ dari fungsi $F^*(x)$, yang memetakan instance x ke nilai output y , melalui minimisasi nilai ekspektasi dari fungsi loss tertentu $L(y, F(x))$. *Gradient boosting* membangun estimasi tambahan dari $F^*(x)$ sebagai jumlah berbobot dari sejumlah fungsi, sehingga memungkinkan model meningkatkan akurasi prediksi melalui iterasi yang berfokus pada mengurangi kesalahan residu (Bentéjac, Csörgő, & Martínez-Muñoz, 2020).

Pada persamaan (2.2) menunjukkan bagaimana setiap model baru (x) ditambahkan secara bertahap dengan bobot pada iterasi ke- m , yang bertujuan untuk mengurangi kesalahan prediksi dari model sebelumnya.

$$F_m(x) = F_{m-1}(x) + \rho_m h_m(x) \quad (2.2)$$

Dalam proses iteratif *gradient boosting*, ρ_m adalah bobot yang diberikan pada fungsi ke- m , yaitu $h_m(x)$. Fungsi-fungsi ini merupakan model-model dalam *ensemble*, seperti *decision tree*. Estimasi dari $F^*(x)$ dibangun secara bertahap, dimulai dengan mendapatkan aproksimasi konstan untuk $F^*(x)$ pada iterasi pertama. Hal ini dicapai dengan meminimalkan nilai *loss function* $L(y_i, \alpha)$ untuk

setiap data pelatihan, dengan α adalah parameter konstanta yang mengoptimalkan fungsi tersebut. Pada iterasi pertama, estimasi ini diberikan oleh persamaan (2.3).

$$F_0(x) = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^N L(y_i, \alpha) \quad (2.3)$$

Persamaan (2.3) menunjukkan bahwa pada awalnya, model menghasilkan prediksi yang didasarkan pada nilai konstanta α yang meminimalkan kesalahan prediksi keseluruhan, $L(y_i, \alpha)$, di seluruh *dataset*. Pendekatan ini digunakan untuk membangun dasar dari model *gradient boosting* sebelum melanjutkan ke iterasi selanjutnya, di mana model-model tambahan (seperti *decision tree*) akan berfungsi untuk memperbaiki prediksi dari model sebelumnya (Bentéjac, Csörgő, & Martínez-Muñoz, 2020).

Pada iterasi selanjutnya, model yang dibangun diharapkan dapat meminimalkan fungsi pada persamaan (2.4).

$$(\rho_m, h_m(x)) = \underset{\rho, h}{\operatorname{argmin}} \sum_{i=1}^N L(y_i, F_m - 1(x_i) + \rho h(x_i)) \quad (2.4)$$

Namun, pendekatan ini tidak menyelesaikan masalah optimisasi secara langsung, melainkan secara iteratif dengan menambahkan model baru secara berurutan. Setiap model h_m dapat dipandang sebagai langkah *greedy* dalam optimisasi menggunakan metode *gradient descent* untuk F^* . Untuk itu, setiap model h_m dilatih menggunakan *dataset* baru $D = \{x_i, r_{mi}\}_{i=1}^N$, di mana *residual* palsu r_{mi} dihitung berdasarkan turunan dari fungsi *loss* $L(y, F(x))$ terhadap $F(x)$, yang dievaluasi pada $F(x) = F_{m-1}(x)$, dengan rumus yang ditunjukkan pada persamaan (2.5).

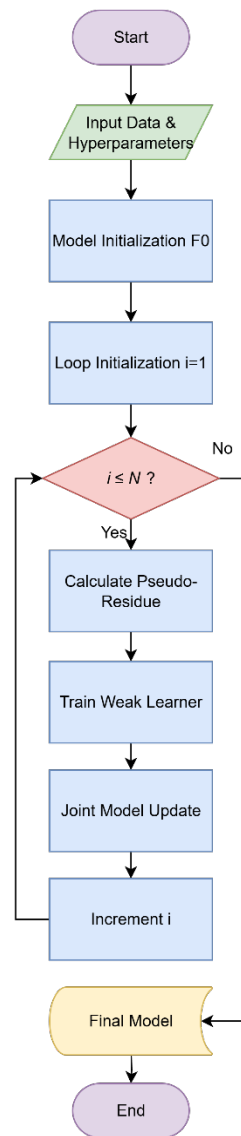
$$r_{mi} = \left[\frac{\partial L(y_i, F(x))}{\partial L(x)} \right]_{F(x)=F_{m-1}(x)} \quad (2.5)$$

Nilai dari ρ_m kemudian dihitung dengan menyelesaikan masalah optimisasi. Proses ini, meskipun sangat efektif, dapat mengalami *overfitting* jika langkah-langkah iteratif tidak diatur dengan benar. Beberapa fungsi *loss* (misalnya *loss* kuadratik) dapat menyebabkan *residual* palsu menjadi nol pada iterasi berikutnya jika model h_m sangat cocok dengan *residual* palsu, yang akan menyebabkan proses tersebut berhenti terlalu cepat. Untuk mengatasi masalah ini dan mengontrol proses penambahan dalam *gradient boosting*, beberapa parameter regularisasi dipertimbangkan. Salah satu cara alami untuk meredakan *overfitting* adalah dengan menerapkan *shrinkage*, yang berfungsi untuk mengurangi setiap langkah *gradient descent* (Bentéjac, Csörgö, & Martínez-Muñoz, 2020).

Gradient boosting membedakan dirinya dari metode *boosting* lainnya dengan menggabungkan konsep-konsep dari teori klasifikasi untuk estimasi dan seleksi efek prediktor dalam model regresi. Dalam hal ini, *gradient boosting* mempertimbangkan efek acak dan menawarkan pendekatan pemodelan yang lebih organik dan tidak bias. Berbeda dengan algoritma *boosting* lainnya yang mungkin mengasumsikan hubungan linier atau terlalu bergantung pada keputusan acak dalam tahap pemilihan model, *gradient boosting* memastikan bahwa estimasi prediktor disesuaikan secara cermat dengan data, meningkatkan akurasi model secara keseluruhan (Griesbach, Säfken, & Waldmann, 2020).

Selain itu, *gradient boosting* juga menawarkan kemampuan untuk menghasilkan perbaikan pada model non-konstan, dengan menggabungkan

pengetahuan sebelumnya atau wawasan fisik terkait proses yang menghasilkan data (Wozniakowski *et al.*, 2021). Ini menjadi keunggulan lain dari *gradient boosting*, karena ia tidak hanya mengandalkan data murni, tetapi juga dapat memanfaatkan pengetahuan domain atau pemahaman fisik tentang bagaimana data tersebut terbentuk. Dengan pendekatan ini, *gradient boosting* dapat meningkatkan prediksi dalam konteks yang lebih luas, termasuk dalam situasi di mana model yang lebih sederhana mungkin gagal. Alur kerja dari algoritma *gradient boosting* tersebut diilustrasikan secara *flowchart* pada Gambar 2.3.



Gambar 2.3 *Flowchart Gradient Boosting* (Zhang *et al.*, 2023)

Gambar 2.3 menyajikan *flowchart* dari algoritma *gradient boosting*. Proses diawali dengan tahap persiapan, di mana data latih (*training data*) beserta konfigurasi *hyperparameter* (seperti jumlah pohon N) dimasukkan ke dalam sistem. Selanjutnya, model diinisialisasi dengan membuat sebuah prediksi awal yang menjadi dasar bagi iterasi berikutnya.

Alur kerja kemudian memasuki sebuah perulangan utama yang dikontrol oleh kondisi $i \leq N$, yang memastikan proses akan berjalan sebanyak N kali. Pada setiap iterasi, langkah pertama adalah menghitung pseudo-residu, yaitu selisih atau kesalahan (*error*) dari prediksi model gabungan saat ini terhadap nilai target sebenarnya. Residu ini kemudian menjadi target pembelajaran bagi sebuah *weak learner* baru yang akan dilatih. Setelah model lemah tersebut terbentuk, ia ditambahkan untuk memperbarui (*update*) model gabungan, sehingga secara bertahap memperbaiki akurasi. Terakhir, penghitung iterasi (i) dinaikkan satu tingkat.

Proses iteratif ini terus berlanjut hingga target jumlah *weak learner* (sebanyak N) terpenuhi. Setelah keluar dari perulangan, alur kerja akan menghasilkan sebuah Model Final, yang merupakan *ensemble* (gabungan) kuat dari seluruh model lemah yang telah dilatih secara sekuensial. Dengan demikian, proses ini pun berakhir.

Sebagai algoritma *ensemble learning* yang semakin berkembang, telah terbukti unggul dalam meningkatkan prediksi dibandingkan dengan model lain, seperti *artificial neural network*, terutama dalam konteks pemodelan dinamis *bioprocess* (Mowbray *et al.*, 2020). Dalam penerapan ini, *gradient boosting* menggabungkan beberapa model pembelajaran yang lemah untuk menghasilkan prediksi yang lebih akurat, menunjukkan keunggulannya dalam memodelkan dan memprediksi proses yang dinamis dan kompleks, serta mampu mengatasi variasi yang ada dalam data yang digunakan.

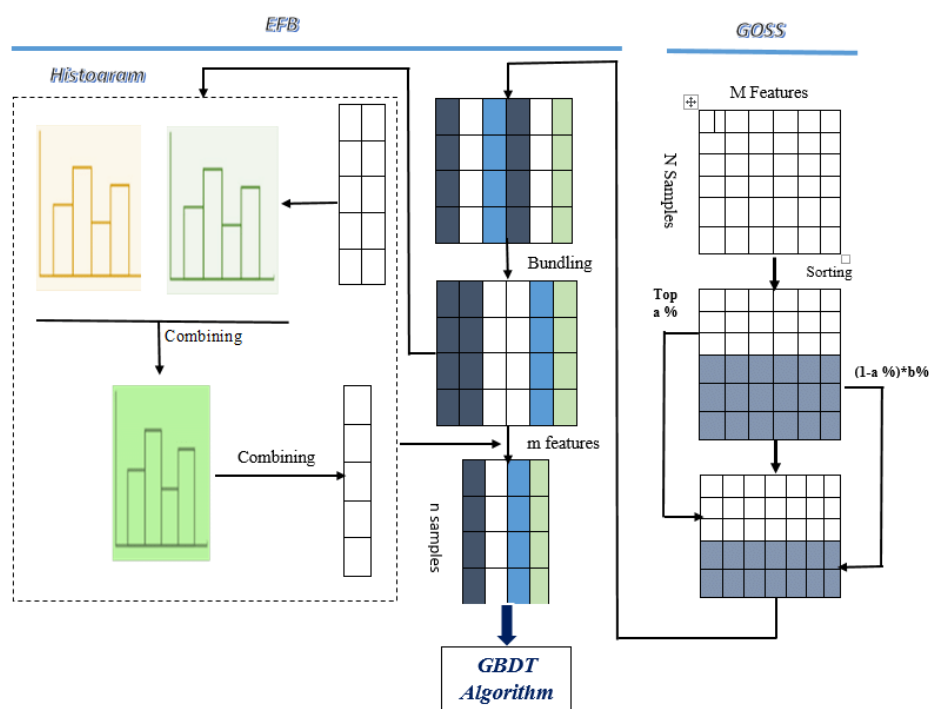
Beberapa parameter dalam *gradient boosting*, seperti jumlah *node*, kedalaman maksimum, dan tingkat pembelajaran, dapat disesuaikan berdasarkan kinerja model pada *testing* set (Hu *et al.*, 2023). Pengaturan parameter ini penting untuk memastikan model tidak hanya memberikan prediksi yang akurat, tetapi juga menghindari *overfitting*. Menyesuaikan parameter-parameter tersebut memungkinkan pemodel untuk mengoptimalkan performa model sesuai dengan karakteristik data yang digunakan, menjadikannya lebih fleksibel dan dapat diandalkan dalam berbagai jenis aplikasi.

2.7 *Light Gradient Boosting Machine*

Light Gradient Boosting Machine (LightGBM) adalah sebuah implementasi *gradien boosting* berkinerja tinggi yang didasarkan pada algoritma *Gradient Boosting Decision Tree* (GBDT), namun disempurnakan dengan teknik-teknik inovatif untuk meningkatkan kecepatan dan efisiensi. LightGBM memiliki beberapa keunggulan, termasuk kecepatan pelatihan yang lebih tinggi, penggunaan memori yang lebih rendah, akurasi yang lebih baik, serta dukungan untuk distribusi data dalam jumlah besar. LightGBM dikembangkan untuk mengatasi keterbatasan dalam GBDT tradisional, khususnya dalam hal kinerja dan efisiensi komputasi, sehingga memungkinkan pelatihan model pada *dataset* yang lebih besar dengan waktu yang lebih singkat (Huang & Chen, 2023).

LightGBM pertama kali dikembangkan pada tahun 2016 oleh tim peneliti di Microsoft sebagai peningkatan atas model GBDT yang populer, yaitu XGBoost. LightGBM diperkenalkan untuk meningkatkan efisiensi dan kecepatan yang lebih

tinggi dari XGBoost, yang sering mengalami kendala kecepatan pada data berukuran besar. Dalam pengembangan LightGBM, tim peneliti memperkenalkan dua teknik baru: *Gradient-based One-Side Sampling* (GOSS) dan *Exclusive Feature Bundling* (EFB). Teknik ini dirancang untuk mengurangi jumlah sampel data dan fitur yang perlu diproses dalam pelatihan GBDT, sehingga mengatasi tantangan komputasi yang terkait dengan pemrosesan *dataset* besar (Kriuchkova, Toloknova, & Drin, 2024). Gambar 2.4 adalah arsitektur peningkatan algoritma GBDT dengan EFB dan GOSS.



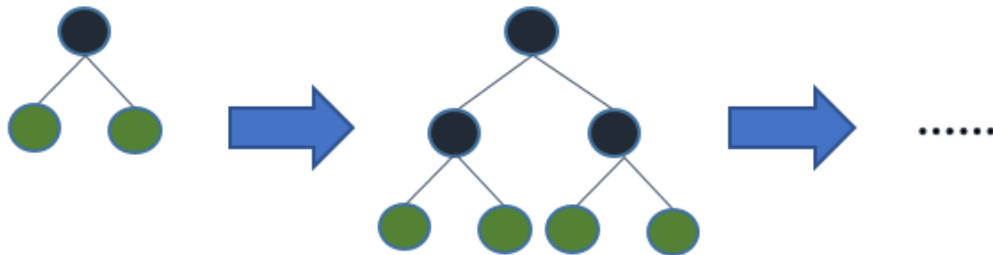
Gambar 2.4 Arsitektur GOSS dan EFB

Pada Gambar 2.4 disajikan arsitektur dan aliran data dalam kerangka kerja GBDT pada LightGBM yang ditingkatkan, mengintegrasikan teknik EFB dan GOSS. EFB bertujuan untuk mengurangi dimensi fitur dengan menggabungkan

fitur-fitur yang jarang aktif bersamaan ke dalam bundel tunggal, sehingga menghasilkan matriks fitur yang lebih ringkas (m fitur dari M fitur awal). Proses ini melibatkan pembentukan histogram untuk setiap fitur dan kemudian menggabungkannya, yang secara efektif mengurangi kompleksitas komputasi tanpa mengorbankan informasi signifikan. Matriks fitur yang telah dibundel kemudian disatukan dengan sampel-sampel yang telah diseleksi oleh GOSS.

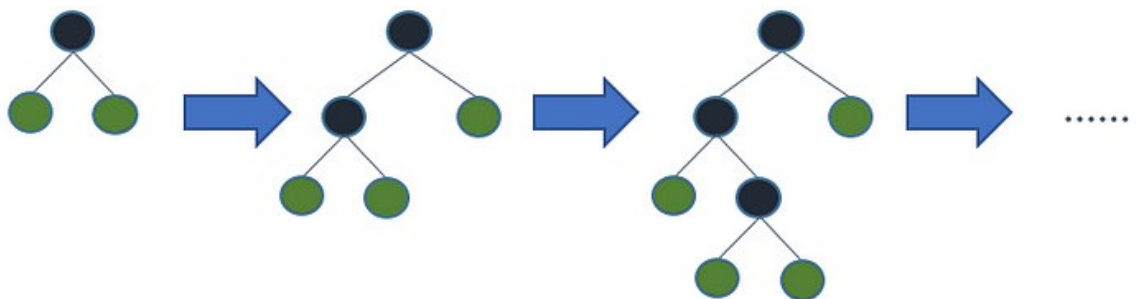
Sementara itu, GOSS mengatasi tantangan jumlah sampel yang besar dengan secara selektif mempertahankan instansi berdasarkan gradiennya. Sampel dengan gradien besar (*top $\alpha\%$*) dipertahankan secara utuh karena mereka berkontribusi paling signifikan terhadap *error* model, sedangkan sampel dengan gradien kecil diambil secara acak pada laju $(1-\alpha\%)\times b\%$. Pendekatan ini memungkinkan algoritma GBDT untuk fokus pada sampel yang paling informatif, mempercepat proses pelatihan sambil menjaga akurasi model. Kombinasi EFB dan GOSS secara sinergis mengurangi dimensi fitur dan jumlah sampel, secara substansial meningkatkan efisiensi komputasi dari algoritma GBDT tanpa mengorbankan kinerja, menjadikannya sangat efektif untuk *dataset* skala besar.

LightGBM menggunakan pendekatan yang berbeda dalam *decision tree learning* dibandingkan algoritma *decision tree* tradisional yang biasanya tumbuh berdasarkan tingkat atau kedalaman pohon (*depth-wise*). Dalam metode tradisional ini, semua *node* pada tingkat yang sama dianggap sama pentingnya, dan pohon bertumbuh secara berjenjang untuk mencakup setiap *node* pada tingkat tertentu, seperti yang ditunjukkan pada Gambar 2.5 (LightGBM, 2024).



Gambar 2.5 Ilustrasi *Level-wise Tree Growth* (LightGBM, 2024)

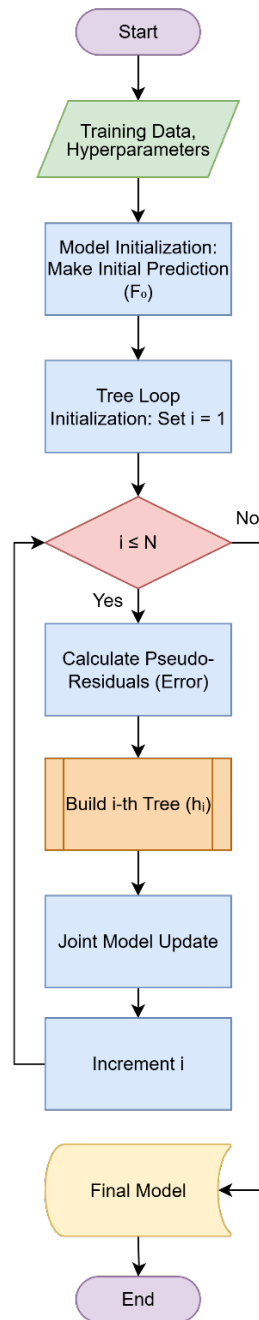
Namun, LightGBM mengadopsi strategi pertumbuhan pohon berbasis daun atau *leaf-wise*, yang hanya membagi daun yang diharapkan memberikan peningkatan terbesar terhadap akurasi model, seperti pada Gambar 2.6. Dengan fokus pada daun yang paling berpotensi untuk meningkatkan performa model, LightGBM membangun pohon secara lebih selektif dan efisien. Strategi *leaf-wise* ini bertujuan untuk memaksimalkan akurasi model dengan sumber daya yang lebih minimal, dibandingkan dengan metode tradisional yang sering kali menghasilkan cabang-cabang pohon yang tidak diperlukan dan memperlambat proses pelatihan (LightGBM, 2024).



Gambar 2.6 Ilustrasi *Leaf-wise Tree Growth* (LightGBM, 2024)

Pendekatan *leaf-wise* dalam LightGBM sering disebut juga sebagai pertumbuhan "*greedy growth*," yang memungkinkan algoritma untuk menemukan dan membagi daun dengan dampak terbesar terhadap akurasi model tanpa harus mempertimbangkan semua cabang secara merata pada setiap tingkat (LightGBM, 2024). Hal ini dapat diibaratkan seperti memangkas cabang-cabang yang tidak perlu, dengan fokus pada jalur yang paling bermanfaat. Sebagai akibat dari pendekatan yang selektif ini, struktur pohon dalam LightGBM menjadi asimetris, di mana beberapa cabang tumbuh lebih dalam daripada cabang lainnya, karena tujuan utamanya bukan simetri, melainkan peningkatan akurasi model.

Manfaat dari strategi pertumbuhan berbasis daun ini adalah dalam hal kecepatan dan akurasi (LightGBM, 2024). Dari segi kecepatan, LightGBM menjadi sangat efisien karena metode *leaf-wise* hanya membagi daun yang memberikan dampak signifikan pada model, sehingga menghindari pengembangan sub-pohon yang tidak berkontribusi banyak terhadap peningkatan akurasi. Selain itu, pertumbuhan *leaf-wise* ini cenderung menghasilkan model dengan tingkat kesalahan (*loss*) yang lebih rendah dan akurasi yang lebih tinggi, karena algoritma dapat lebih terfokus pada bagian data yang paling informatif. Hal ini menjadikan LightGBM sebagai algoritma yang unggul dalam hal efisiensi dan ketepatan dalam menangani *dataset* yang besar dan kompleks. Gambar 2.7 menunjukkan *flowchart* dari LightGBM.

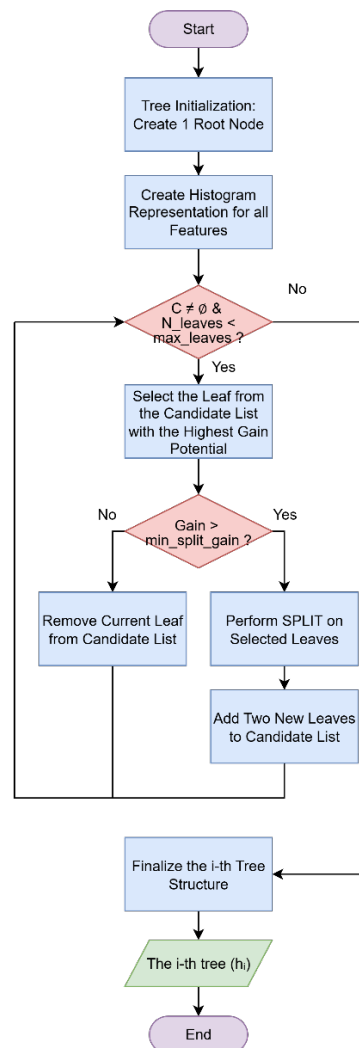


Gambar 2.7 *Flowchart* LightGBM (Ke *et al.*, 2017)

Gambar 2.7 menyajikan diagram alur kerja rinci dari algoritma LightGBM. Proses diawali dengan masukan (input) berupa data latih (*training data*) beserta konfigurasi *hyperparameter* yang telah ditentukan. Selanjutnya, algoritma

melakukan tahap inisialisasi, di mana model dasar (F_0) dibentuk dengan membuat prediksi awal dan sebuah penghitung iterasi (i) diatur untuk memulai perulangan.

Setelah inisialisasi, proses memasuki *loop* iteratif utama yang berjalan selama jumlah pohon (N) yang ditargetkan belum tercapai. Pada setiap iterasi, langkah pertama adalah menghitung nilai *pseudo-residual*, yaitu selisih antara nilai target aktual dengan hasil prediksi model dari iterasi sebelumnya. Nilai *residual* ini kemudian digunakan sebagai target baru untuk melatih sebuah model lemah (*weak learner*), yang dalam hal ini adalah satu pohon keputusan. Setelah pohon keputusan yang baru berhasil dibangun, model gabungan diperbaharui dengan menambahkan kontribusi dari pohon baru tersebut yang telah diskalakan oleh *learning rate*. Proses ini diulang secara terus-menerus hingga kondisi berhenti terpenuhi dan menghasilkan sebuah model final yang kuat. Adapun rincian mengenai alur kerja sub-proses untuk membangun satu pohon keputusan secara detail disajikan pada Gambar 2.8.



Gambar 2.8 *Flowchart* Sub-Proses Membangun Satu Pohon Keputusan

Gambar 2.8 menyajikan diagram alur kerja untuk sub-proses pembangunan satu pohon keputusan, yang merupakan langkah detail dari alur utama LightGBM. Proses ini diawali dengan inisialisasi pohon yang terdiri dari satu *root node*. Untuk efisiensi komputasi, seluruh fitur pada data kemudian diubah ke dalam representasi berbasis histogram.

Selanjutnya, algoritma memasuki *loop* utama untuk menumbuhkan pohon secara *leaf-wise*. Pada setiap iterasi, dari semua daun yang menjadi kandidat, algoritma akan memilih satu daun yang memiliki potensi *information gain* tertinggi. Setelah kandidat daun terbaik ditentukan, dilakukan pengecekan apakah nilai gain dari pemecahan (*split*) tersebut melebihi ambang batas *min_split_gain*. Jika gain mencukupi, daun tersebut akan dipecah menjadi dua daun baru, dan keduanya ditambahkan ke dalam daftar kandidat untuk iterasi selanjutnya. Sebaliknya, jika gain tidak mencukupi, daun tersebut akan dihapus dari daftar kandidat. Perulangan ini terus berjalan hingga tidak ada lagi kandidat yang bisa dipecah atau jumlah daun telah mencapai batas *max_leaves*, kemudian struktur pohon yang sudah final dikembalikan ke alur utama.

Untuk mengimplementasikan LightGBM, *library* utama yang diperlukan adalah LightGBM itu sendiri, yang dapat diinstal melalui pengelola paket sesuai bahasa pemrograman yang digunakan, seperti *Python* atau *R* (LightGBM, 2024). Selain *library* utama tersebut, ada beberapa dependensi lain yang juga dibutuhkan, seperti *CMake* untuk membangun lingkungan pengembangan dan *library* *CUDA* jika ingin memanfaatkan akselerasi GPU untuk mempercepat proses komputasi. Dengan adanya dukungan GPU, LightGBM dapat menangani data dalam jumlah besar dengan lebih efisien, mempercepat pelatihan model secara signifikan.

2.8 Data Non-Linear

Dalam analisis data dan pemodelan statistik, pemahaman terhadap sifat hubungan antar variabel adalah fundamental. Secara tradisional, banyak metode

mengasumsikan adanya hubungan linear, di mana perubahan pada satu variabel akan menghasilkan perubahan proporsional pada variabel lainnya. Namun, sebagian besar fenomena di dunia nyata, mulai dari sistem biologis, pasar keuangan, hingga interaksi sosial, menunjukkan pola yang jauh lebih kompleks dan tidak dapat dijelaskan secara akurat oleh garis lurus. Kemampuan untuk memodelkan hubungan non-linear ini menjadi inti dari banyak kemajuan dalam kecerdasan buatan modern (Liu *et al.*, 2021).

Data non-linear merujuk pada sekumpulan data di mana hubungan antara variabel independen (prediktor) dan variabel dependen (target) tidak dapat direpresentasikan sebagai fungsi linear. Jika hubungan linear dapat digambarkan sebagai garis lurus, maka hubungan non-linear dapat mengambil berbagai bentuk kurva, seperti polinomial, eksponensial, logaritmik, atau pola kompleks lainnya yang tidak beraturan (Aggarwal, 2020). Tabel 2.1 menunjukkan contoh data non-linear.

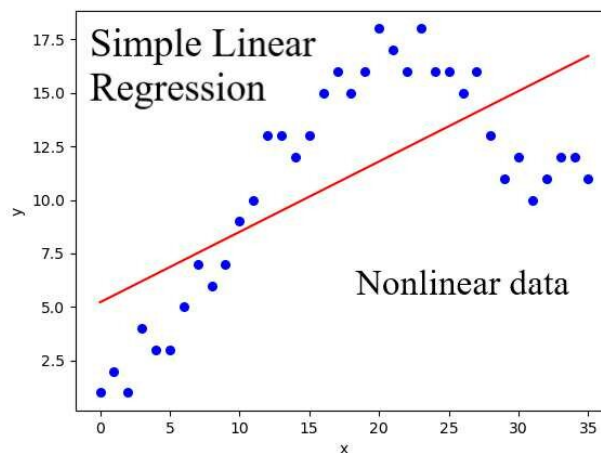
Tabel 2.1 Contoh Data Non-Linear (Ahmad *et al.*, 2021)

Usia Beton (Hari)	Kuat Tekan Rata-rata (MPa)
3	28,52
7	44,75
14	59,86
28	79,99
56	85,23
90	88,33
100	89,54

Tabel 2.1 menunjukkan bahwa peningkatan kuat tekan beton dari hari ke-3 hingga hari ke-28 sangat signifikan (lebih dari 50 MPa), sementara peningkatan

dari hari ke-56 hingga hari ke-100 jauh lebih landai (hanya sekitar 4 MPa). Pola pertumbuhan yang melambat ini secara visual membentuk sebuah kurva, bukan garis lurus, yang menegaskan sifat non-linear dari hubungan antara usia dan kekuatan beton.

Mengabaikan non-linearitas dalam data dapat menyebabkan kesimpulan yang salah dan model prediktif yang tidak akurat. Model linear yang dipaksakan pada data non-linear akan menghasilkan *underfitting*, di mana model gagal menangkap struktur fundamental dalam data, sehingga memiliki performa yang buruk baik pada data latih maupun data uji (Shwartz-Ziv & Armon, 2022). Gambar 2.9 menunjukkan visualisasi data non-linear.



Gambar 2.9 Visualisasi Data Non-linear (Senapati, 2023)

Gambar 2.9 mengilustrasikan keterbatasan model Regresi Linier Sederhana ketika diterapkan pada data yang bersifat non-linear. Sebaran titik-titik data (berwarna biru) secara jelas menunjukkan pola yang melengkung, bukan garis lurus. Meskipun model regresi (garis merah) mencoba menemukan tren garis lurus

terbaik yang "mendekati" semua titik, model tersebut gagal menangkap pola kurva yang sesungguhnya. Akibatnya, terjadi kesalahan (*error*) prediksi yang besar, di mana garis model berada jauh di bawah atau di atas titik data aktual pada beberapa bagian, yang membuktikan bahwa model linier tidak cocok untuk data dengan hubungan yang kompleks seperti ini.

Pentingnya analisis data non-linear menjadi semakin krusial seiring dengan meningkatnya volume dan kompleksitas data (Big Data). Dalam bidang-bidang seperti keuangan, model non-linear diperlukan untuk memprediksi volatilitas pasar saham yang sangat tidak menentu (Ozbayoglu *et al.*, 2020). Di bidang medis, hubungan antara faktor risiko dan penyakit seringkali bersifat sangat kompleks dan non-linear, sehingga memerlukan model canggih untuk diagnosis dini atau prediksi prognosis (Ahmad *et al.*, 2021). Demikian pula dalam pemrosesan bahasa alami (NLP), hubungan antar kata dalam sebuah kalimat bersifat sangat non-linear, yang menjadi alasan utama kesuksesan model berbasis Transformer dan deep learning lainnya (Otter *et al.*, 2020).

2.9 Sepak Bola

Sepak bola merupakan olahraga tim yang dimainkan secara global, menuntut pemain untuk menguasai berbagai kemampuan teknis, taktis, dan fisik dalam lingkungan yang dinamis dan kompetitif. Permainan ini pada dasarnya melibatkan dua tim yang saling berhadapan, di mana setiap tim berusaha untuk mencetak gol dengan memasukkan bola ke gawang lawan menggunakan bagian tubuh mana pun selain tangan atau lengan (Sarmento *et al.*, 2018). Sifat permainan

yang kompleks dan interaktif ini menjadikan sepak bola sebagai subjek yang kaya untuk dianalisis dari berbagai perspektif ilmiah, mulai dari fisiologi hingga analisis data performa.

Sebuah pertandingan sepak bola standar dimainkan dalam dua babak yang masing-masing berdurasi 45 menit, dengan tujuan utama untuk mencetak skor lebih tinggi dari tim lawan. Setiap tim terdiri atas pemain yang menempati posisi-posisi strategis, seperti penjaga gawang yang bertugas melindungi gawang, pemain bertahan yang menghalau serangan lawan, pemain tengah yang mengatur alur permainan, serta penyerang yang berfokus untuk menciptakan peluang dan mencetak gol. Keberhasilan sebuah tim tidak hanya ditentukan oleh kemampuan individu, tetapi juga oleh kohesi dan koordinasi kolektif dalam menjalankan strategi permainan di bawah kerangka aturan yang diawasi oleh wasit (Filter *et al.*, 2023).

2.10 Analisis Sepak Bola

Analisis sepak bola merupakan proses mencari, menafsirkan, dan mengolah data untuk mendapatkan keunggulan kompetitif, baik dalam bidang olahraga untuk meningkatkan kinerja dan efisiensi tim, maupun dalam bidang ekonomi yang berkaitan dengan perolehan pendapatan (Malagón-Selma, 2023). Analisis sepak bola juga diartikan sebagai kombinasi dari pengumpulan data, prediksi pertandingan, dan penggunaan alat serta teknik untuk menafsirkan strategi permainan guna meningkatkan performa pemain secara individu dan tim (Fontanive, 2021). Pendekatan ini bertujuan untuk mempelajari, memahami, dan memodelkan bagian objektif dari olahraga, sehingga mengurangi peran

subjektivitas dalam pengambilan keputusan (Malagón-Selma, 2023). Pendekatan dalam analisis ini sangat ditentukan oleh sifat data yang tersedia, yang diklasifikasikan ke dalam beberapa kategori utama (Malagón-Selma, 2023):

- a. Data Peristiwa (*Eventing*): Merupakan data yang mencatat semua aksi terukur yang berkaitan dengan bola selama pertandingan, seperti jumlah gol, asis, dan tekel. Data ini digunakan untuk menganalisis statistik yang berpengaruh pada hasil pertandingan, membedakan tim sukses dan gagal, hingga memprediksi nilai pasar pemain (Malagón-Selma, 2023).
- b. Data Pelacakan (*Tracking*): Merupakan data spatio-temporal yang merekam pergerakan setiap pemain (dengan atau tanpa bola) di lapangan. Data ini memungkinkan analisis taktis yang mendalam mengenai formasi, strategi tim, dan kontribusi individu pemain (Malagón-Selma, 2023).
- c. Data GPS: Fokus pada pengumpulan informasi aktivitas fisik pemain seperti total jarak tempuh dan lari berintensitas tinggi. Analisisnya penting untuk memahami tuntutan fisik, dampak kelelahan, dan pengaruh jadwal padat terhadap performa (Malagón-Selma, 2023).
- d. Data Cedera (*Injuries*): Mencakup informasi mengenai cedera pemain. Data ini dimanfaatkan untuk mempelajari penyebab cedera dan hubungannya dengan intensitas latihan atau kepadatan jadwal pertandingan (Malagón-Selma, 2023).

Lebih jauh, analisis dalam sepak bola tidak hanya fokus pada aspek teknis dan taktis, tetapi juga memperhatikan variabel situasional yang perlu diperhatikan seperti lokasi pertandingan, kualitas lawan, dan status pertandingan yang berpengaruh pada performa tim (Sarmiento *et al.*, 2018). Dalam upaya

meningkatkan performa pemain dan mengembangkan aktivitas pelatih, analisis sepak bola juga mengarah pada aspek-aspek mendetail seperti performa dalam situasi bola mati, perilaku sistem kolektif, komunikasi tim, dan profil aktivitas pemain (Sarmiento *et al.*, 2018). Melalui pemanfaatan berbagai jenis data ini, analisis sepak bola memberikan landasan objektif bagi pelatih dan manajemen untuk membuat keputusan strategis yang lebih terinformasi dan berbasis bukti.

2.11 *Expected Goals (xG)*

Expected Goals atau xG adalah salah satu metrik yang semakin digunakan dalam analisis sepak bola modern untuk menilai peluang terjadinya gol berdasarkan kualitas dan lokasi tembakan yang dilakukan (Mead, O'Hare, & McMenemy, 2023). Metrik ini memberikan prediksi probabilitas yang lebih akurat dibandingkan statistik konvensional dalam memperkirakan keberhasilan suatu tim di masa mendatang. Dalam hal ini, xG membantu memberikan pandangan yang lebih obyektif dan berbasis data mengenai kemungkinan pencapaian gol yang dihasilkan dari berbagai jenis tembakan selama pertandingan.

Metrik xG dirancang untuk memberikan skor probabilistik pada setiap tembakan, dengan nilai yang berkisar antara 0 dan 1, di mana 0 menunjukkan tidak ada peluang mencetak gol, dan 1 menunjukkan kepastian terjadinya gol. Penilaian ini memungkinkan xG untuk menangani unsur ketidakpastian dalam sepak bola dengan lebih baik dibandingkan metrik berbasis gol konvensional. Karena tembakan jauh lebih sering terjadi daripada gol, pendekatan ini memungkinkan

analisis yang lebih stabil dan realistis dalam memahami efektivitas tim dan pemain di lapangan (Mead, O'Hare, & McMenemy, 2023).

Perkembangan xG berdasarkan penelitian yang diteliti dapat dilihat pada Tabel 2.2.

Tabel 2.2 Perkembangan xG

Tahun	Sumber	Judul & Peneliti	Algoritma	Dataset
1997	<i>The Statistician</i>	<i>Measuring the effectiveness of playing strategies at soccer</i> (Pollard & Reep)	<i>Logistic Regression</i>	22 match, 489 tembakan
2005	<i>World congress on science and football: Conference Paper</i>	<i>Applications of Logistic Regression to Shots at Goal in Association Football</i> (Ensum, Pollard, & Taylor)	<i>Logistic Regression</i>	48 match, 1.099 tembakan
2015	ESANN 2015 proceedings	<i>Measuring scoring efficiency through goal expectancy estimation</i> (Ruiz et al.)	<i>Multilayer Perceptron (MLP)</i>	EPL 2013/14, Prozone, 10.318 tembakan
2016	<i>Master's Thesis / MLSA 2016</i>	<i>Expected Goals in Soccer Explaining Match Results Using Predictive Analytics</i> (Eggels)	<i>Logistic Regression, Decision Tree, Random Forest, AdaBoost, Isotonic</i>	>1.000 peluang, 5.020 match (Data tembakan tidak disebutkan)
2022	<i>Journal of the Operational Research Society</i> 76(1)	<i>Explainable Expected Goals</i> (Cavus & Biecek)	<i>XGBoost, Random Forest, LightGBM, CatBoost via Forester AutoML</i>	Understat event data (315.430 tembakan dari 7 musim Top 5 Eropa)
2025	<i>Frontiers in Sports & Active Living</i>	<i>Toward Interpretable Expected Goals Modeling Using Bayesian Mixed Models</i> (Iapteff et al., 2025)	<i>Bayesian Generalized Linear Mixed Model (GLMM) + Transfer Learning</i>	StatsBomb 2003–22: 460 match, 63.177 tembakan

Berdasarkan Tabel 2.2, terlihat evolusi signifikan dalam penelitian xG dari tahun 1997 hingga 2025, yang ditandai oleh dua tren utama yaitu peningkatan

kompleksitas algoritma dan skala *dataset* yang digunakan. Pada awalnya, penelitian seperti oleh Pollard & Reep (1997) dan Ensum, Pollard, & Taylor (2005) bergantung pada model fundamental *Logistic Regression*. Metodologi kemudian berkembang dengan penerapan model jaringan saraf seperti *Multilayer Perceptron* (MLP) pada tahun 2015 dan eksplorasi beragam algoritma *machine learning* termasuk *Random Forest* dan AdaBoost pada tahun 2016. Puncak kecanggihan metode terlihat pada penelitian tahun 2025 oleh Iapteff *et al*, yang menerapkan model statistik canggih *Bayesian Generalized Linear Mixed Model* (GLMM) yang dikombinasikan dengan *Transfer Learning*. Perkembangan ini didukung oleh peningkatan skala data secara eksponensial, mulai dari 489 tembakan pada tahun 1997 meningkat menjadi 10.318 tembakan pada 2015 dan mencapai puncaknya pada 63.177 tembakan dalam penelitian terbaru.

Selain berguna untuk analisis taktis yang mendukung peningkatan performa di lapangan, xG juga memainkan peran penting dalam keputusan finansial klub. Metrik ini membantu dalam keputusan seperti perekrutan pemain dan negosiasi kontrak dengan memberikan wawasan yang lebih akurat mengenai kontribusi pemain. Dengan demikian, xG tidak hanya membantu klub dalam memaksimalkan performa di lapangan tetapi juga dalam mengelola sumber daya finansial secara lebih efisien (Mead, O'Hare, & McMenemy, 2023).

Penerapan xG memberikan keuntungan strategis bagi klub sepak bola dengan memperluas pemahaman terkait kualitas peluang yang dihasilkan. Hal ini memungkinkan klub untuk mengevaluasi kinerja pemain secara lebih mendalam dan membantu dalam pengembangan strategi permainan yang berbasis pada

kualitas dan efektivitas peluang (Mead, O'Hare, & McMenemy, 2023). Oleh karena itu, xG melampaui perannya sebagai metrik statistik.

Di dalam konsepnya, perhitungan xG dapat dianggap sebagai permasalahan klasifikasi, karena melibatkan penentuan probabilitas tembakan menghasilkan gol berdasarkan berbagai faktor. Untuk menghitung probabilitas ini, metode *machine learning* dan statistika sering diterapkan, termasuk *logistic regression*, *gradient boosting*, *neural networks*, *support vector machines*, serta algoritma klasifikasi *tree-based*. Beragam pendekatan ini memungkinkan xG untuk memanfaatkan data historis dan pola dalam data tembakan untuk memodelkan kemungkinan gol secara lebih akurat, yang berguna dalam memberikan penilaian yang lebih detail tentang kualitas peluang tembakan (Herbinet, 2018).

Penggunaan fitur-fitur data dalam pengembangan model xG sangat beragam dan tidak mengikuti satu format yang baku. Setiap peneliti atau organisasi dapat memilih kombinasi variabel yang berbeda tergantung pada ketersediaan data, tujuan model, dan kompleksitas komputasi yang diinginkan. Fleksibilitas ini memungkinkan adanya inovasi berkelanjutan dalam pemodelan xG, di mana fitur-fitur baru seperti tekanan dari pemain bertahan atau posisi penjaga gawang mulai diintegrasikan untuk meningkatkan akurasi (Herbinet, 2018).

Meskipun terdapat keragaman, ada kesamaan fundamental pada fitur-fitur inti yang hampir selalu ada dalam setiap model xG. Fitur-fitur ini dianggap sebagai prediktor paling signifikan terhadap probabilitas sebuah tembakan menjadi gol. Beberapa fitur data yang secara konsisten digunakan meliputi: lokasi tembakan di lapangan (yang kemudian diterjemahkan menjadi jarak dan sudut ke gawang),

bagian tubuh yang digunakan untuk menembak (misalnya, kaki atau kepala), serta jenis aksi permainan yang mendahului tembakan (misalnya, umpan terobosan, umpan silang, atau situasi bola mati). Praktik ini menunjukkan bahwa meskipun model dapat menjadi sangat kompleks, fondasinya sering kali dibangun di atas variabel-variabel yang secara intuitif paling memengaruhi hasil tembakan (Olvera-Rojas *et al.*, 2023). Tabel 2.3 memperlihatkan fitur-fitur inti yang hampir selalu ada dalam setiap model xG.

Tabel 2.3 Daftar Fitur Data Umum dalam Model xG

Fitur Data	Deskripsi	Sifat Data
Jarak ke Gawang	Jarak dari lokasi tembakan ke titik tengah garis gawang.	Non-Linear
Sudut Tembakan	Sudut yang dibentuk oleh lokasi tembakan dengan kedua tiang gawang.	Non-Linear
Lokasi Tembakan (x,y)	Koordinat spesifik di lapangan tempat tembakan dilepaskan.	Non-Linear
Bagian Tubuh	Jenis bagian tubuh yang digunakan untuk melakukan tembakan (misalnya, kaki kanan, kaki kiri, kepala).	Non-Linear
Jenis Umpan/ <i>Assist</i>	Tipe aksi yang langsung mendahului tembakan (misalnya, umpan silang, umpan terobosan, bola liar).	Non-Linear
Jenis Permainan	Konteks permainan saat tembakan terjadi (misalnya, permainan terbuka, tendangan bebas, tendangan sudut, penalti).	Non-Linear
Tekanan Pemain Bertahan	Jumlah atau jarak pemain lawan terdekat saat tembakan dilepaskan.	Non-Linear

Berdasarkan faktor-faktor tersebut, sebuah tembakan mungkin diberi nilai 0,30 xG. namun model yang lebih presisi, seperti Statsbomb xG, mempertimbangkan informasi tambahan seperti posisi kiper, status kiper, posisi pemain bertahan dan penyerang, serta tinggi dampak tembakan. Dalam kondisi kiper yang tidak berada di posisinya, model ini mungkin memberikan nilai yang

lebih tinggi, misalnya 0,65 xG, untuk menggambarkan kualitas peluang yang lebih tinggi (Statsbomb, 2024).

Visualisasi dari model ini pada Gambar 2.10, yang merupakan visualisasi xG pada pertandingan langsung, memperlihatkan bagaimana setiap faktor dihitung untuk menghasilkan prediksi xG yang mendalam dan akurat.



Gambar 2.10 Visualisasi xG pada Pertandingan Langsung (Statsbomb, 2024)

2.12 Brier Score

Brier score merupakan metrik evaluasi yang mengukur ketepatan dalam pemodelan prediksi, dengan cara membagi prediksi ke dalam beberapa kelompok atau “bins” berdasarkan kesamaan nilai prediksi (Foster & Hart, 2022). Metrik ini memadukan skor kalibrasi dan skor penyempurnaan (*refinement*) untuk mengukur keahlian dalam pemodelan prediktif. Dengan menggabungkan aspek kalibrasi, yang menunjukkan seberapa baik prediksi sejalan dengan hasil aktual, dan aspek penyempurnaan, yang melihat kemampuan model dalam memisahkan atau

membedakan hasil yang berbeda, *Brier score* memberikan gambaran komprehensif mengenai performa model dalam memberikan prediksi probabilistik.

Penggunaan *Brier score* dalam evaluasi model probabilitas penting karena metrik ini dapat mengukur kemampuan diskriminasi dan performa prediktif secara keseluruhan. Dengan kata lain, *Brier score* tidak hanya melihat akurasi dari prediksi probabilitas tetapi juga sejauh mana model dapat membedakan antara kejadian yang mungkin terjadi dengan yang tidak (Dimitriadis *et al.*, 2023). Hal ini membuat *Brier score* menjadi pilihan yang baik untuk mengevaluasi performa model probabilistik, khususnya ketika diperlukan pemahaman yang lebih dalam mengenai kualitas prediksi yang bersifat probabilistik. Fungsi Brier score ditunjukkan pada persamaan (2.6).

$$Brier\ Score = (f_t - o_t)^2 \quad (2.6)$$

Brier score digunakan untuk menghitung selisih kuadrat antara nilai prediksi dan nilai aktual, sebagaimana terlihat pada persamaan (2.6). Dalam konteks ini, f_t merepresentasikan nilai probabilitas yang diprediksi untuk suatu peristiwa, sedangkan o_t adalah nilai aktual dari peristiwa tersebut (biasanya 1 jika terjadi dan 0 jika tidak terjadi). *Brier score* memiliki rentang nilai antara 0 hingga 1, di mana nilai yang lebih rendah menunjukkan prediksi yang lebih akurat karena mendekati hasil aktual (Foster & Hart, 2022).

Brier score diperkenalkan oleh Glenn W. Brier pada tahun 1950 sebagai alat untuk menilai akurasi prediksi probabilitas (Foster & Hart, 2022). Skor ini menghitung selisih antara nilai prediksi dan realisasi aktual, di mana hasil

perhitungan *Brier score* memperlihatkan seberapa dekat prediksi tersebut dengan hasil aktual menggunakan formula *mean squared error* standar.

Sejak pertama kali diperkenalkan yaitu pada evaluasi ramalan cuaca, *Brier score* telah berkembang menjadi metode yang diakui untuk mengukur akurasi model probabilitas dalam berbagai bidang, termasuk bisnis dan aplikasi lainnya (Petroopoulos *et al.*, 2022). Penerapan awalnya pada meteorologi menunjukkan bagaimana metode ini dapat memberikan wawasan yang lebih mendalam terhadap ketepatan perkiraan, yang kemudian menjadikan *Brier Score* sebagai standar dalam penilaian akurasi probabilitas di berbagai disiplin ilmu.

2.13 *Confusion Matrix*

Evaluasi kinerja sebuah model klasifikasi dimulai dari pemahaman terhadap *confusion matrix*. *Confusion matrix* adalah alat visualisasi fundamental dalam bentuk tabel kontingensi yang merangkum dan membandingkan hasil prediksi model dengan kelas aktual dari data uji. Struktur ini memberikan gambaran yang jelas tidak hanya tentang seberapa sering model benar, tetapi juga tentang jenis kesalahan yang dibuatnya (Tharwat, 2021).

Untuk masalah klasifikasi biner, *confusion matrix* biasanya disajikan dalam format 2×2 . Matriks ini memiliki empat komponen utama yang mendeskripsikan hasil prediksi. Pertama, *True Positive* (TP), yang mewakili kasus di mana model dengan tepat memprediksi kelas positif; sebagai contoh, sebuah *email spam* berhasil diidentifikasi sebagai spam. Kedua, *True Negative* (TN), yaitu kasus di mana model secara benar memprediksi kelas negatif, seperti email penting yang tidak

diklasifikasikan sebagai spam. Dua komponen lainnya menggambarkan kesalahan model. *False Positive* (FP), atau *error* tipe I, terjadi ketika model salah memprediksi kelas positif untuk *instance* yang sebenarnya negatif, misalnya email penting yang keliru ditandai sebagai spam. Terakhir, *False Negative* (FN), atau *error* tipe II, terjadi saat model salah memprediksi kelas negatif untuk *instance* yang sebenarnya positif, seperti *email spam* yang lolos dari filter dan masuk ke kotak masuk utama. Visualisasi dari keempat komponen tersebut disajikan dalam struktur Tabel 2.2.

Tabel 2.4 Confusion Matrix

	Prediksi Positif	Prediksi Negatif
Aktual Positif	<i>True Positive</i> (TP)	<i>False Negative</i> (FN)
Aktual Negatif	<i>False Positive</i> (FP)	<i>True Negative</i> (TN)

2.14 Receiver Operating Characteristic Area Under Curve (ROC AUC)

Receiver Operating Characteristic (ROC) adalah alat statistik yang digunakan untuk menilai kinerja model klasifikasi dengan menggambarkan hubungan antara dua parameter, yaitu *True Positive Rate* (TPR) dan *False Positive Rate* (FPR). Analisis ROC dapat dilakukan dengan memanfaatkan distribusi prior dan algoritma *elicitation* untuk memilih prior yang tepat, yang selanjutnya digunakan untuk menarik inferensi mengenai AUC (*Area Under the Curve*) dan karakteristik error model (Al-Labadi *et al.*, 2022).

ROC juga digunakan untuk mengevaluasi kinerja perangkat pengujian dan algoritma klasifikasi dalam menilai kepatuhan terhadap kriteria tertentu (Pendrill *et al.*, 2023). Dengan demikian, ROC menjadi alat yang penting untuk perbandingan dan evaluasi relatif dari berbagai sistem klasifikasi dalam konteks yang berbeda.

Kurva ROC menggambarkan kinerja model klasifikasi pada berbagai ambang batas klasifikasi dengan memplot dua parameter utama, yaitu TPR dan FPR. Salah satu kelemahan dari kurva ROC adalah kesulitan dalam menginterpretasi kinerja model jika terdapat banyak titik keputusan, karena setiap titik mewakili *trade-off* antara TPR dan FPR, yang dapat membuat sulit untuk menentukan titik terbaik yang mencerminkan kinerja keseluruhan model (Chen *et al.*, 2023). ROC AUC mengukur luas dua dimensi di bawah kurva ROC, dimulai dari titik (0,0) hingga (1,1). Semakin tinggi nilai ROC AUC, semakin baik model dalam membedakan antara kelas positif dan negatif.

Secara matematis, AUC dari kurva ROC dihitung dengan mengintegalkan fungsi ROC. Mengingat kurva ROC memplot *True Positive Rate* (TPR) sebagai fungsi dari *False Positive Rate* (FPR), AUC dapat didefinisikan dengan persamaan integral (Chen *et al.*, 2023) seperti pada persamaan (2.7).

$$AUC = \int_0^1 TPR(FPR), d(FPR) \quad (2.7)$$

Dalam praktiknya, karena kurva ROC terdiri atas sejumlah titik diskrit, AUC sering dihitung menggunakan aturan trapesium. Parameter TPR dan FPR sendiri dihitung berdasarkan nilai dari *confusion matrix* dengan persamaan (2.8) dan (2.9).

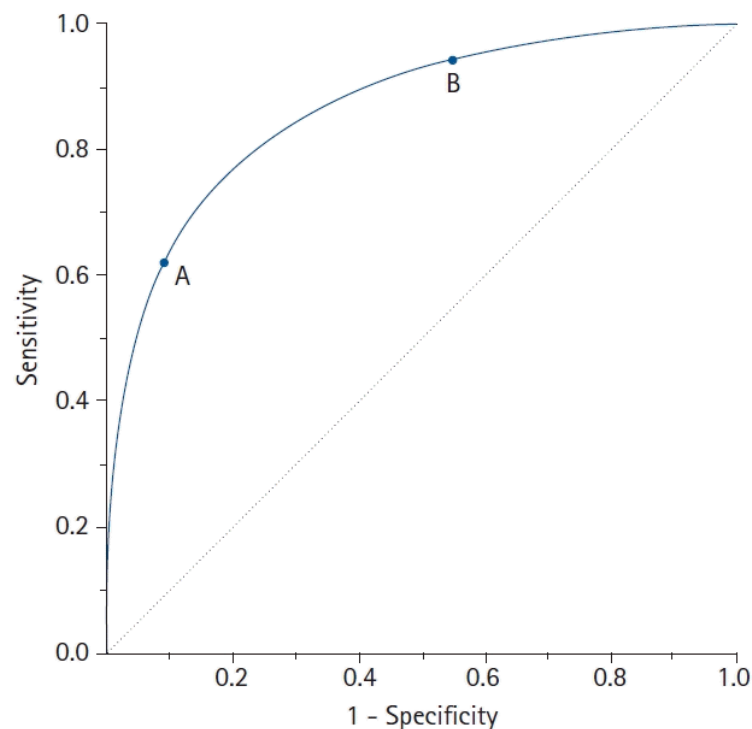
$$TPR = \frac{TP}{TP + FN} \quad (2.8)$$

$$FPR = \frac{FP}{FP + TN} \quad (2.9)$$

Di mana TP adalah *True Positive*, FN adalah *False Negative*, FP adalah *False Positive*, dan TN adalah *True Negative*. Nilai AUC juga dapat diinterpretasikan

sebagai probabilitas bahwa model akan memberikan skor yang lebih tinggi untuk sampel positif yang dipilih secara acak dibandingkan dengan sampel negatif yang dipilih secara acak (Nahm, 2022).

Pada Gambar 2.11, menampilkan kurva ROC AUC, di mana sumbu x menunjukkan nilai $1 - \text{spesifisitas}$ (*false positive rate*) dan sumbu y menunjukkan sensitivitas pada semua nilai *cut-off* yang diukur dari hasil pengujian (Nahm, 2022). Ketika nilai *cut-off* yang lebih ketat diterapkan, titik pada kurva akan bergerak ke bawah dan ke kiri (Titik A). Sebaliknya, saat *cut-off* lebih longgar diterapkan, titik pada kurva bergerak ke atas dan ke kanan (Titik B). Garis diagonal 45° pada grafik ini berfungsi sebagai garis referensi, yang merepresentasikan kurva ROC dari klasifikasi acak.



Gambar 2.11 Contoh ROC AUC (Nahm, 2022)

ROC AUC memiliki peran penting dalam evaluasi model karena mampu mengukur kinerja model dalam berbagai kelompok risiko yang diprediksi (Carrington *et al.*, 2021). Ini memberikan informasi yang lebih mendalam yang dapat digunakan dalam pengambilan keputusan, memungkinkan pemahaman yang lebih komprehensif tentang bagaimana model berperforma di berbagai titik potong dan kelompok risiko.

Lebih lanjut, ROC AUC juga memungkinkan perbandingan yang wajar antar model dan membantu mengidentifikasi batas keputusan yang optimal serta potensi peningkatan ROC AUC. Ini membuat ROC AUC sangat bermanfaat dalam seleksi model yang lebih baik dan pemahaman tentang ruang yang dapat dioptimalkan untuk meningkatkan kinerja klasifikasi (Tafvizi *et al.*, 2022).

Secara fundamental, fungsi utama dari ROC AUC adalah untuk menyediakan satu nilai tunggal yang merangkum kinerja keseluruhan model klasifikasi di semua kemungkinan ambang batas (*threshold*). Alih-alih harus menganalisis setiap titik pada kurva ROC, yang dapat menyulitkan interpretasi jika terdapat banyak titik keputusan (Chen *et al.*, 2023), ROC AUC menyederhanakan evaluasi dengan mengukur total area dua dimensi di bawah kurva tersebut. Nilai AUC ini dapat diartikan sebagai probabilitas bahwa model akan memberikan skor prediksi yang lebih tinggi untuk sampel kelas positif yang dipilih secara acak daripada sampel kelas negatif yang dipilih secara acak. Oleh karena itu, AUC berfungsi sebagai metrik yang agregat dan tidak bergantung pada ambang batas tertentu, di mana nilai yang mendekati 1.0 menunjukkan kemampuan diskriminasi

yang sangat baik antara kelas positif dan negatif, sementara nilai mendekati 0.5 mengindikasikan kinerja yang tidak lebih baik dari tebakan acak (Chen *et al.*, 2023).

2.15 Akurasi

Akurasi merupakan metrik evaluasi yang paling intuitif dan umum digunakan untuk mengukur performa sebuah model klasifikasi. Secara fundamental, akurasi mengukur proporsi dari total prediksi yang dibuat oleh model yang sesuai dengan kelas aktualnya. Dengan kata lain, metrik ini menjawab pertanyaan sederhana: "Dari keseluruhan data, berapa persen yang berhasil ditebak dengan benar oleh model?". Tingginya nilai akurasi sering kali diartikan sebagai indikator performa model yang baik, karena menunjukkan tingkat kesalahan yang rendah secara keseluruhan (Sarker, 2021).

Secara matematis, akurasi dihitung dengan menjumlahkan prediksi yang benar yaitu *True Positive* (TP) dan *True Negative* (TN), lalu membaginya dengan jumlah total seluruh sampel data. Formula untuk menghitung akurasi ditunjukkan pada persamaan (2.10) (Tharwat, 2021).

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.10)$$

Meskipun sederhana dan mudah diinterpretasikan, akurasi memiliki kelemahan yang signifikan, terutama ketika dihadapkan pada *dataset* yang tidak seimbang (*imbalanced dataset*). Dalam skenario seperti itu, di mana jumlah sampel pada satu kelas jauh lebih dominan daripada kelas lainnya, akurasi bisa menjadi metrik yang menyesatkan (Tharwat, 2021).

2.16 Presisi (*Precision*)

Setelah memahami komponen *confusion matrix*, metrik Presisi dapat didefinisikan secara spesifik. Presisi menjawab pertanyaan: "Dari semua *instance* yang diprediksi oleh model sebagai kelas positif, berapa persen yang benar-benar positif?" Metrik ini mengukur tingkat keandalan atau ketepatan dari prediksi positif yang dibuat oleh model (Tharwat, 2021).

Secara matematis, presisi dihitung dengan membagi jumlah *true positive* dengan total jumlah prediksi positif yang dihasilkan oleh model (*true positive* ditambah *false positive*), seperti yang ditunjukkan pada persamaan (2.11) (Tharwat, 2021).

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2.11)$$

Presisi menjadi metrik yang sangat krusial dalam skenario di mana biaya dari *False Positive* (FP) sangat tinggi. Sebagai contoh, dalam sistem penyaringan email, kesalahan mengklasifikasikan email penting dari atasan sebagai spam (sebuah FP) dapat menyebabkan pengguna kehilangan informasi yang sangat krusial. Dalam kasus deteksi penyakit, mendiagnosis orang sehat sebagai penderita penyakit (FP) dapat menyebabkan kecemasan, biaya pengobatan yang tidak perlu, dan tes lebih lanjut yang invasif. Oleh karena itu, model dengan presisi tinggi lebih disukai dalam situasi-situasi tersebut karena ia cenderung tidak salah dalam melabeli sesuatu sebagai positif (Tharwat, 2021).

2.17 *Recall*

Recall, yang juga dikenal sebagai sensitivitas atau *True Positive Rate* (TPR), adalah metrik yang mengukur kemampuan sebuah model untuk menemukan kembali semua sampel positif yang relevan dalam sebuah *dataset*. Dengan kata lain, *recall* merepresentasikan proporsi dari kasus positif aktual yang berhasil diidentifikasi dengan benar oleh model. Esensi dari metrik ini adalah untuk mengevaluasi tingkat kelengkapan (*completeness*) dari prediksi positif yang dihasilkan (Tharwat, 2021).

Recall sangat krusial dalam domain di mana biaya dari *false negative* sangat tinggi. Misalnya, dalam diagnosis medis, gagal mendeteksi adanya penyakit (*false negative*) pada pasien bisa berakibat fatal. Oleh karena itu, *recall* yang tinggi lebih diutamakan dalam konteks tersebut. *Recall* dapat dihitung dengan persamaan (2.12) (Tharwat, 2021).

$$Recall = \frac{TP}{TP + FN} \quad (2.12)$$

2.18 *F1-Score*

F1-Score adalah metrik yang menggabungkan presisi dan *recall* ke dalam satu skor tunggal dengan menghitung rata-rata harmonik dari keduanya. Rata-rata harmonik cenderung lebih dekat ke nilai yang lebih kecil, sehingga *F1-Score* memberikan bobot yang seimbang pada kedua metrik tersebut (Tharwat, 2021).

Metrik ini sangat berguna ketika terjadi ketidakseimbangan kelas (*imbalanced class*), di mana jumlah sampel pada satu kelas jauh lebih dominan daripada kelas lainnya. Dalam kasus seperti itu, akurasi saja bisa menyesatkan,

sedangkan *F1-Score* memberikan gambaran yang lebih representatif mengenai performa model. Nilai *F1-Score* yang tinggi menunjukkan bahwa model memiliki performa yang baik dalam hal presisi maupun *recall*, menjadikannya metrik evaluasi yang komprehensif. Persamaan untuk *F1-Score* ditunjukkan pada persamaan (2.13) (Tharwat, 2021).

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (2.13)$$

2.19 *Log-Loss (Cross-Entropy Loss)*

Log-Loss, yang secara formal dikenal sebagai *cross-entropy loss*, adalah metrik evaluasi fundamental untuk model klasifikasi yang menghasilkan *output* probabilitas (Murphy, 2022). Berakar dari teori informasi, metrik ini mengukur "jarak" antara distribusi probabilitas yang diprediksi model dengan distribusi aktualnya. Berbeda dengan akurasi yang hanya menilai kebenaran prediksi, *Log-Loss* memberikan evaluasi yang lebih mendalam dengan mengukur seberapa baik kalibrasi dan tingkat keyakinan (*confidence*) dari setiap prediksi (Sarker, 2021). Hal ini menjadikannya sangat berharga dalam aplikasi berbasis risiko, di mana mengetahui probabilitas suatu hasil jauh lebih penting daripada sekadar klasifikasi benar atau salah.

Mekanisme utama *Log-Loss* adalah memberikan penalti yang besar pada prediksi yang sangat yakin namun ternyata salah. Sebagai contoh, prediksi dengan probabilitas 0.95 untuk kelas yang salah akan dihukum jauh lebih berat daripada prediksi 0.55 untuk kasus yang sama. Dengan demikian, metrik ini mendorong model untuk tidak hanya akurat, tetapi juga menghasilkan probabilitas yang

terkalibrasi dengan baik. Nilai *Log-Loss* yang ideal adalah 0, yang menandakan model sempurna, dan nilai yang semakin tinggi menunjukkan performa model yang semakin buruk.

Untuk masalah klasifikasi biner (di mana hasil akhirnya adalah 1 atau 0), *Log-Loss* dihitung menggunakan persamaan (2.14) (Murphy, 2022).

$$LogLoss = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (2.14)$$

Dalam persamaan (2.13), N merepresentasikan jumlah total sampel dalam *dataset*. Perhitungan kerugian dilakukan secara iteratif untuk setiap sampel, dari sampel pertama ($i=1$) hingga terakhir ($i=N$), yang dilambangkan oleh operator sigma (Σ). Hasil penjumlahan total kerugian kemudian dinormalisasi dengan cara dibagi oleh N , sehingga menghasilkan nilai kerugian rata-rata. Normalisasi ini memastikan bahwa performa model dapat dibandingkan secara adil tanpa terpengaruh oleh ukuran *dataset*.

Setiap komponen dalam kurung siku menghitung kerugian untuk satu sampel individual. Di sini, y_i adalah label kelas aktual dari sampel ke- i , yang memiliki nilai 1 untuk kelas positif dan 0 untuk kelas negatif. Sementara itu, p_i adalah probabilitas yang dihasilkan oleh model, yang menunjukkan prediksi peluang sampel ke- i untuk masuk ke dalam kelas positif (nilai antara 0 dan 1). Fungsi logaritma natural, $\log()$, menjadi inti dari mekanisme penalti dalam rumus ini.

Logika perhitungan kerugian dapat dipahami dengan menganalisis dua kondisi berdasarkan nilai y_i . Pertama, ketika label aktualnya adalah 1 ($y_i=1$), suku kedua dalam penjumlahan, yaitu $(1-y_i) \log (1-p_i)$, akan menjadi nol. Dengan demikian, kerugian untuk sampel ini hanya dihitung dari suku pertama, $y_i \log (p_i)$ atau $\log (p_i)$. Jika model memprediksi probabilitas (p_i) yang mendekati 1 (sangat yakin benar), nilai $\log (p_i)$ akan mendekati 0, yang berarti kerugian sangat kecil. Sebaliknya, jika model salah prediksi dengan p_i mendekati 0, nilai $\log (p_i)$ akan menuju negatif tak terhingga, yang setelah dikalikan dengan tanda negatif di awal rumus, akan menghasilkan nilai kerugian yang sangat besar.

Kedua, ketika label aktualnya adalah 0 ($y_i=0$), suku pertama, $y_i \log (p_i)$, akan menjadi nol. Perhitungan kerugian kini bergantung pada suku kedua, $(1-y_i) \log (1-p_i)$ atau $\log (1-p_i)$. Suku $1-p_i$ merepresentasikan probabilitas sampel untuk masuk ke kelas negatif. Jika model memprediksi p_i mendekati 0 (sehingga $1-p_i$ mendekati 1), maka model sangat yakin bahwa sampel ini adalah kelas negatif. Dalam kasus ini, nilai $\log (1-p_i)$ akan mendekati 0, dan kerugiannya pun kecil. Namun, jika model salah besar dengan memprediksi p_i mendekati 1, maka $1-p_i$ akan mendekati 0, dan kerugian (*loss*) yang dihasilkan akan sangat besar.

2.20 *Feature Engineering*

Feature engineering adalah proses rekayasa data secara cerdas untuk meningkatkan kinerja model *machine learning* dengan cara meningkatkan akurasi dan kemampuan interpretasinya (Verdonck *et al.*, 2024). Proses ini dilakukan melalui penyesuaian fitur yang telah ada atau dengan mengekstraksi fitur baru yang

lebih bermakna dari berbagai sumber data. Teknik ini bertujuan untuk menciptakan representasi data yang lebih informatif, sehingga model dapat memahami hubungan yang lebih kompleks di dalam data. *Feature engineering* tidak hanya membantu dalam memperbaiki akurasi prediksi, tetapi juga memungkinkan pengguna untuk memahami bagaimana setiap fitur memengaruhi hasil akhir, menjadikannya langkah penting dalam pengembangan model *machine learning* yang lebih efektif dan dapat diandalkan.

Feature engineering memungkinkan pengguna untuk membuat fitur-fitur baru secara mandiri yang lebih relevan dengan permasalahan yang sedang dianalisis (Das *et al.*, 2022). Fitur-fitur ini kemudian dapat digunakan untuk meningkatkan proses penerapan algoritma *machine learning* dalam membuat prediksi yang lebih akurat. Dengan menciptakan fitur yang disesuaikan dengan kebutuhan analisis, pengguna dapat membantu model *machine learning* mengenali pola-pola penting yang sebelumnya tidak terdeteksi, sehingga hasil prediksi menjadi lebih optimal dan bermakna.

Teknik-teknik esensial dalam *feature engineering* berperan penting dalam meningkatkan kinerja model prediksi di berbagai bidang. Teknik-teknik ini mencakup (Katya, 2023):

a. *Feature Selection*

Feature selection merupakan proses memilih fitur-fitur yang paling relevan dan informatif dari kumpulan data yang tersedia. Dengan menyaring fitur yang tidak signifikan atau *redundant*, proses ini membantu mengurangi *noise* dan kompleksitas data. Hal tersebut sangat penting untuk mencegah

overfitting dan memastikan bahwa model hanya menggunakan informasi yang benar-benar berkontribusi terhadap variabel target. Dengan demikian, model prediksi dapat bekerja lebih efisien dan menghasilkan akurasi yang lebih tinggi.

b. *Dimensionality Reduction*

Dimensionality reduction adalah teknik yang bertujuan untuk mengurangi jumlah fitur dalam *dataset* tanpa mengorbankan informasi penting yang terkandung di dalamnya. Teknik ini menyederhanakan struktur data, sehingga memudahkan proses analisis dan meningkatkan performa model. Metode seperti *Principal Component Analysis* (PCA) mengubah fitur asli menjadi komponen baru yang lebih ringkas, tetapi tetap merepresentasikan variasi data secara keseluruhan. Pendekatan ini tidak hanya mempercepat proses pelatihan model, tetapi juga meningkatkan kemampuan interpretasi hasil.

c. *Interaction Term Creation*

Interaction term creation adalah proses menciptakan fitur baru dengan mengombinasikan dua atau lebih fitur yang ada. Teknik ini dirancang untuk menangkap interaksi atau hubungan sinergis antar fitur yang mungkin tidak terlihat saat dianalisis secara individual. Dengan menggabungkan fitur-fitur tersebut, model dapat lebih sensitif terhadap pola-pola kompleks yang berpengaruh terhadap hasil akhir, sehingga meningkatkan keakuratan prediksi.

Secara keseluruhan, penerapan teknik-teknik ini dalam *feature engineering* membantu mengoptimalkan data input sehingga algoritma *machine learning* dapat menghasilkan prediksi yang lebih akurat dan interpretasi yang lebih mendalam. Teknik-teknik tersebut berperan penting dalam menyederhanakan, menyoroti, dan memperkaya informasi yang terkandung dalam data, yang pada akhirnya berkontribusi terhadap peningkatan kinerja model di berbagai aplikasi.

2.21 Tools Penelitian

Pelaksanaan penelitian ini didukung oleh beberapa perangkat lunak esensial yang digunakan untuk pemrosesan dan analisis data. Setiap *tool* memiliki peran spesifik yang berkontribusi pada pencapaian tujuan penelitian.

2.21.1 Python

Python adalah bahasa pemrograman tingkat tinggi bersifat *object-oriented*, dikembangkan oleh Guido van Rossum, bahasa ini dirancang untuk menjadi mudah dipahami dan digunakan sehingga cocok baik untuk pemula yang sedang mempelajari dasar-dasar pemrograman maupun untuk para profesional yang mengerjakan proyek pemrograman di dunia nyata (Hur, 2025). Python menawarkan *syntax* yang sederhana dan intuitif, sehingga memungkinkan pengguna menulis kode dengan lebih cepat dan efisien. Selain itu, Python memiliki dukungan pustaka yang sangat luas serta komunitas yang aktif, menjadikannya pilihan populer untuk berbagai kebutuhan, mulai dari pengembangan web, analisis data, *machine learning*, hingga komputasi ilmiah dan otomatisasi sistem.

Python menawarkan keseimbangan antara kejelasan *syntax* dan fleksibilitas dalam pengembangan alat-alat penelitian komputasi, sehingga sangat mendukung dalam menciptakan solusi untuk berbagai jenis permasalahan yang kompleks. Bahasa ini dirancang untuk menangani beragam tantangan yang melibatkan pengolahan *dataset* berukuran besar, penerapan algoritma yang rumit, serta pengembangan sistem komputasi (Hur, 2025). Kemampuan Python untuk berintegrasi dengan berbagai pustaka dan *framework* membuatnya menjadi pilihan utama dalam penelitian berbasis data dan pengembangan teknologi inovatif. Dengan ekosistem yang luas, Python memungkinkan peneliti dan pengembang untuk membangun, menguji, serta mengimplementasikan solusi secara efisien dan *scalable*.

2.21.2 Pandas

Pandas adalah pustaka Python berperforma tinggi yang dirancang khusus untuk manipulasi, analisis, dan eksplorasi data. Pustaka ini banyak digunakan oleh peneliti data, analis, dan pengembang karena kemampuannya yang unggul dalam mengolah data secara efisien (Molin & Jee, 2021). Pandas menyediakan berbagai fungsi yang memudahkan proses pembersihan, transformasi, serta analisis data dalam berbagai format, seperti tabel, *file* CSV, dan *database*. Selain itu, Pandas juga mendukung integrasi dengan pustaka visualisasi seperti Matplotlib dan Seaborn, sehingga memungkinkan pengguna untuk membuat visualisasi data yang informatif dan menarik. Kemudahan penggunaan serta fleksibilitas Pandas menjadikannya salah satu alat utama dalam analisis data modern dan pengembangan aplikasi berbasis data.

Salah satu kekuatan utama dari pustaka ini adalah penggunaan data *frame* dan *series*, yang menjadi inti dalam proses manipulasi, perhitungan, serta analisis data (Gupta & Bagchi, 2024). Data *frame* adalah struktur data berbentuk tabel dengan label pada baris dan kolom, mirip dengan tabel pada *database* atau *spreadsheet*, sehingga memudahkan pengolahan data dalam jumlah besar. Sementara itu, *series* merupakan struktur data satu dimensi yang berfungsi seperti *array*, tetapi dilengkapi dengan indeks yang memungkinkan akses data lebih fleksibel. Kombinasi dari dua struktur data ini memungkinkan pengguna untuk melakukan berbagai operasi analisis secara efisien, seperti pengolahan data numerik, transformasi data, serta agregasi hasil analisis dengan *syntax* yang sederhana namun *powerful*.

2.21.3 Scikit-learn

Scikit-learn merupakan pustaka Python yang menyediakan antarmuka standar untuk mengimplementasikan berbagai algoritma *machine learning*. Pustaka ini dirancang agar mudah digunakan, sehingga memudahkan pengguna dari berbagai latar belakang untuk mengembangkan model *machine learning* dengan lebih efisien. Selain mendukung algoritma untuk klasifikasi, regresi, dan *clustering*, Scikit-learn juga dilengkapi dengan berbagai fungsi penting lainnya, seperti data *preprocessing*, *resampling*, evaluasi model, serta pencarian *hyperparameter*. Fungsi-fungsi tersebut membantu memastikan bahwa proses pengolahan data, pelatihan model, hingga evaluasi dapat dilakukan secara menyeluruh dan sistematis (Bisong, 2019).

2.21.4 Matplotlib

Matplotlib adalah pustaka Python yang digunakan untuk pembuatan grafik dan visualisasi data. Pustaka ini menyediakan berbagai fitur yang memungkinkan pengguna untuk membuat beragam jenis grafik dan diagram, mulai dari grafik garis (*line plot*), grafik sebar (*scatter plot*), peta panas (*heatmap*), diagram batang (*bar chart*), diagram lingkaran (*pie chart*), hingga visualisasi data dalam bentuk tiga dimensi (3D plot) (Hunt, 2019). Kemampuan Matplotlib dalam menghasilkan visualisasi yang informatif dan berkualitas tinggi menjadikannya salah satu alat utama bagi peneliti dan analis data. Selain itu, pustaka ini mendukung kustomisasi penuh pada setiap elemen grafik, seperti warna, label, dan sumbu, sehingga memudahkan pengguna untuk menyajikan data secara lebih menarik dan sesuai dengan kebutuhan analisis.

2.21.5 Seaborn

Seaborn adalah pustaka Python yang dirancang untuk membuat visualisasi grafik statistik dengan cara yang lebih mudah dan estetis. Pustaka ini menyediakan antarmuka tingkat tinggi untuk Matplotlib, sehingga memungkinkan pengguna membuat grafik kompleks dengan sedikit kode (Waskom, 2021). Seaborn juga terintegrasi erat dengan Pandas, sehingga pengguna dapat langsung memvisualisasikan data dari struktur data *frame* tanpa perlu konversi tambahan. Dengan berbagai fitur bawaan, seperti pembuatan grafik hubungan antar variabel, distribusi data, serta anotasi statistik, Seaborn membantu dalam menyajikan visualisasi data yang informatif dan menarik. Kemudahan penggunaan serta desain

visual yang lebih elegan membuat Seaborn menjadi pilihan utama bagi analis data dan ilmuwan data yang ingin meningkatkan kualitas visualisasi mereka.

2.22 Penelitian Sejenis

Penelitian sejenis yang digunakan pada penelitian ini ditunjukkan pada Tabel 2.3.

Tabel 2.5 Penelitian Sejenis

No	Penulis	Domain Riset	Metode dan Tools	Dataset (Populasi, Sampel)	Kontribusi	Hasil
1	Pollard & Reep (1997)	Pemodelan kualitas tembakan dan cikal bakal metrik xG	Regresi Logistik	22 pertandingan, 489 tembakan	Salah satu studi perintis dalam kuantifikasi xG, membuktikan pentingnya lokasi tembakan.	<ul style="list-style-type: none"> Persamaan probabilitas gol (regresi logistik) Koefisien Jarak (X): -0.096 (semakin jauh, peluang turun) Koefisien Sudut (A): -1.037 (semakin menyamping, peluang turun)
2	Ensum, Pollard, & Taylor (2005)	Pengembangan model probabilitas gol melalui analisis multivariat kontekstual	Regresi Logistik	48 pertandingan, 1.099 tembakan (<i>FA Premier League & World Cup</i>)	Konfirmasi dan perluasan temuan awal dengan <i>dataset</i> yang lebih besar.	<ul style="list-style-type: none"> Persamaan probabilitas gol (regresi logistik) Koefisien Jarak: -0.16 Koefisien Sudut: -1.24 Koefisien Sundulan: -0.73 Koefisien Kaki Lemah: -0.63 Koefisien Tendangan Voli: -0.27 (Semua koefisien negatif menunjukkan penurunan probabilitas gol dibandingkan kondisi ideal).
3	Lucey <i>et al.</i> (2015)	Analitika video <i>spatiotemporal</i> untuk pemodelan peluang gol	<i>Conditional Random Fields</i>	Data Prozone/Stats Perform (~9.732 tembakan + 10 detik video pra-tembakan)	Menggabungkan fitur strategis & <i>spatiotemporal</i> (fase permainan, interaksi pemain) dari data video.	ROC AUC: <ul style="list-style-type: none"> AUC Model <i>Baseline</i> (Hanya Lokasi): 0.75 AUC Model <i>Spatiotemporal</i> (EGV): 0.81

No	Penulis	Domain Riset	Metode dan Tools	Dataset (Populasi, Sampel)	Kontribusi	Hasil
4	Ruiz <i>et al.</i> (2015)	<i>Machine learning</i> untuk evaluasi kemampuan <i>finishing</i> pemain	<i>Multilayer Perceptron</i> (MLP)	Data Prozone (EPL 2013/14, 10.318 tembakan)	Aplikasi model non-linear (MLP) untuk mengidentifikasi efisiensi pemain secara individual.	<p><i>p-value</i>:</p> <ul style="list-style-type: none"> • Tingkat Signifikansi Sangat Tinggi (p 0.01) • Tingkat Signifikansi Tinggi (p 0.05) <p>Nilai <i>p-value</i> ini menunjukkan bahwa kemampuan individu para pemain ini dalam mencetak gol secara signifikan melebihi ekspektasi model xG standar.</p>
5	Eggels <i>et al.</i> (2016)	Analitik prediktif untuk hasil pertandingan	<ul style="list-style-type: none"> • Regresi Logistik • <i>Decision Tree</i> • <i>Random Forest</i> • AdaBoost • Python (scikit-learn) 	Data <i>event/tracking</i> ORTEC & Inmotio + atribut EA Sports (~20.000 tembakan)	Perbandingan komprehensif berbagai model <i>machine learning</i> untuk prediksi xG.	<p>ROC AUC:</p> <p>AdaBoost = 0,84</p> <p><i>Random Forest</i> = 0,82</p> <p>Regresi Logistik = 0,78</p> <p><i>Decision Tree</i> = 0,74</p>
6	Fairchild <i>et al.</i> (2018)	Analitika terapan pemodelan xG yang praktis dan terinterpretasi	<ul style="list-style-type: none"> • Regresi Logistik • Python • SciPy/Statsmodels 	1.115 tembakan non-penalti dari 99 pertandingan MLS 2016 (data <i>tag manual</i>)	Aplikasi model sederhana yang dapat diinterpretasikan pada <i>dataset</i> yang lebih kecil	Kalibrasi model yang kuat dengan validasi silang ROC AUC = 0,80.
7	Pardo (2020)	Pengayaan model xG dengan atribut pemain dari data eksternal	<ul style="list-style-type: none"> • Regresi Logistik • XGBoost • ANN • <i>scikit-learn</i> • Keras 	Data OPTA (~20.000 tembakan) + atribut pemain FIFA (740 pemain)	Integrasi atribut kualitatif pemain (dari <i>game</i> FIFA) ke dalam model xG.	<p>ROC AUC:</p> <ul style="list-style-type: none"> • ANN = 0,88 • XGBoost = 0,85 • Regresi Logistik = 0,78. <p>RMSE:</p> <ul style="list-style-type: none"> • ANN = 0,25 • XGBoost = 0,27 • Regresi Logistik = 0,32.

No	Penulis	Domain Riset	Metode dan Tools	Dataset (Populasi, Sampel)	Kontribusi	Hasil
8	Wheatcroft & Sienkiewicz (2021)	Pemodelan probabilistik kesuksesan tembakan	<ul style="list-style-type: none"> Model probabilistik parametrik Python (SciPy <i>optimize</i>) 	>1 juta tembakan dari 22 liga (football-data.co.uk)	Pengembangan model yang sangat sederhana dan cepat untuk <i>pipeline</i> prediksi.	Peningkatan <i>Log-Score (log-likelihood per shot)</i> dibandingkan model dasar: <ul style="list-style-type: none"> • <i>Log-Score</i> Model Mereka: -0.312 • <i>Log-Score Baseline Naive</i>: -0.342 • Peningkatan vs. <i>Baseline Naive</i>: +0.030 • Peningkatan vs. <i>Baseline Pollard & Reep</i>: +0.004
9	Cavus & Biecek (2022)	<i>Explainable AI</i> (XAI) untuk interpretabilitas model xG kompleks	XGBoost, RF, LightGBM, CatBoost (via AutoML)	Data <i>event</i> Understat (315.430 tembakan)	Pionir dalam menerapkan AutoML untuk eksplorasi model dan SHAP untuk interpretabilitas model kompleks.	<i>Random Forest</i> menjadi model terbaik: <ul style="list-style-type: none"> • ROC AUC (<i>Random Forest</i>): 0.875 • <i>Brier Score</i> (<i>Random Forest</i>): 0.072 Performa model lain: <ul style="list-style-type: none"> • AUC (LightGBM): 0.873 • AUC (CatBoost): 0.872 • AUC (XGBoost): 0.869
10	Méndez <i>et al.</i> (2023)	Peningkatan nilai xG dengan jaringan saraf	<ul style="list-style-type: none"> <i>Multilayer Perceptron</i> (MLP) Python Keras 	Data <i>event</i> StatsBomb (>12.000 tembakan)	Menunjukkan superioritas MLP dalam menangkap pola non-linear dibandingkan regresi logistik.	MLP secara konsisten mengungguli Regresi Logistik (LR) di semua metrik: <ul style="list-style-type: none"> • ROC AUC: 0.87 (MLP) vs. 0.82 (LR) • Akurasi: 90.04% (MLP) vs. 89.28% (LR) • <i>Brier Score</i>: 0.076 (MLP) vs. 0.081 (LR)

No	Penulis	Domain Riset	Metode dan <i>Tools</i>	<i>Dataset</i> (Populasi, Sampel)	Kontribusi	Hasil
						<ul style="list-style-type: none"> • Log-Loss: 0.262 (MLP) vs. 0.280 (LR) • F1-Score: 0.41 (MLP) vs. 0.34 (LR) Arsitektur MLP: 2 <i>hidden layers</i> (16 dan 8 neuron, aktivasi <i>ReLU</i>)
11	Mead <i>et al.</i> (2023)	Peningkatan performa & demonstrasi nilai model xG	<ul style="list-style-type: none"> • Regresi Logistik, • RF, • AdaBoost, • XGBoost, • Python (scikit-learn) 	Data Wyscout (~250.000 tembakan)	Menunjukkan peningkatan performa signifikan dengan fitur tambahan (nilai pemain, <i>rating</i> ELO).	Perbandingan ROC AUC yang menunjukkan keunggulan <i>Random Forest</i> dengan fitur yang diperkaya: <ul style="list-style-type: none"> • <i>Random Forest</i> (Fitur Diperkaya): 0.910 • <i>Random Forest</i> (Fitur Dasar): 0.891 • Regresi Logistik (Fitur Dasar): 0.852
12	Hewitt & Karakuş (2023)	Pemodelan xG kontekstual berbasis identitas dan peran pemain	<ul style="list-style-type: none"> • <i>Regresi Logistik</i>, • <i>Gradient Boosted Trees (GBT)</i>, • <i>scikit-learn</i>, 	Data <i>event</i> StatsBomb (15.574 tembakan) dari 5 liga top Eropa musim 2021/22	Mengembangkan model yang menyesuaikan nilai xG dengan kemampuan pemain dan posisi.	<ul style="list-style-type: none"> • Korelasi Pearson antara xG dan gol lebih tinggi pada GBT (0,208) dibandingkan Regresi Logistik (0,188). • Analisis pada 347 tembakan Lionel Messi menunjukkan model ini memberikan nilai xG +3,74 lebih tinggi daripada model dasar.
13	Bandara <i>et al.</i> (2024)	Pemodelan xG sekuensial berbasis aliran <i>event</i> pra-tembakan	<ul style="list-style-type: none"> • <i>Random Forest</i> (100 <i>estimators</i>), • scikit-learn 	Data dari 990 pertandingan (StatsBomb Open Data), mencakup kompetisi seperti	Inovasi dengan menggunakan fitur dari 3 urutan <i>event</i> sebelum tembakan, bukan hanya data tembakan tunggal.	<ul style="list-style-type: none"> • Model berbasis sekuens mencapai ROC AUC 0,833 pada set validasi. • Pada pengujian dengan data Euro 2020, model

No	Penulis	Domain Riset	Metode dan Tools	Dataset (Populasi, Sampel)	Kontribusi	Hasil
				Piala Dunia, Euro, Liga <i>Champions</i> , dll.		mencapai ROC AUC 0,826, mengungguli model tembakan tunggal (<i>baseline</i>).
14	Cefis & Carpita (2024)	Pemodelan statistik xG dengan pengayaan fitur multi-sumber	<ul style="list-style-type: none"> • Regresi Logistik • R (fungsi glm) 	Data dari 3 sumber: Understat, SoFIFA, Math&Sport. Total 49.872 tembakan dari 5 liga top Eropa musim 2022/23.	Mengintegrasikan fitur-fitur baru seperti tekanan pada penembak, <i>rating</i> pemain (SoFIFA), dan kekuatan lawan ke dalam model linier (Regresi Logistik).	<ul style="list-style-type: none"> • ROC AUC = 0,812 • <i>Brier Score</i> = 0,078 Hasil ini menunjukkan bahwa model linier yang lebih sederhana masih bisa mencapai performa yang kuat jika diperkaya dengan fitur-fitur yang relevan dan inovatif.
15	Xu <i>et al.</i> (2025)	<i>Computer vision</i> dan pemodelan xG berbasis analisis pose tubuh	<ul style="list-style-type: none"> • Jaringan Saraf Konvolusional (CNN) • Regresi Logistik • Analisis <i>Pose</i> (OpenPose) 	<ul style="list-style-type: none"> • Data Publik: 477 tembakan dari <i>dataset</i> publik. • Data SoccerNet-v2: 927 tembakan dari 500 pertandingan. 	<ul style="list-style-type: none"> • Pionir dalam menggunakan data pose/kerangka tubuh pemain (<i>skeleton</i> data) secara langsung untuk estimasi xG. • Memperkenalkan model Skor-xG yang mengintegrasikan orientasi tubuh pemain dan posisi kiper. 	<ul style="list-style-type: none"> • ROC AUC: Model Skor-xG mencapai 0,845, mengungguli model Regresi Logistik (0,791). • <i>Brier Score</i>: Skor-xG mendapatkan 0,068, lebih baik dari Regresi Logistik (0,075). (Nilai lebih rendah lebih baik).

Berdasarkan Tabel 2.2, terdapat lima belas penelitian yang mengkaji perhitungan metrik xG dalam analisis sepak bola. Penelitian-penelitian ini dapat dikelompokkan berdasarkan metodologi yang digunakan, mulai dari model statistik sederhana hingga pendekatan *deep learning* yang kompleks.

Kelompok pertama mencakup penelitian yang mengandalkan model statistik yang lebih mudah diinterpretasi. Studi perintis oleh Pollard & Reep (1997) dan kelanjutannya oleh Ensum, Pollard, & Taylor (2005) menerapkan Regresi Logistik untuk mengukur efektivitas tembakan. Pendekatan serupa juga digunakan oleh Fairchild *et al.* (2018) untuk analisis spasial di liga MLS, serta Cefis & Carpita (2024) yang memperkaya model Regresi Logistik dengan fitur inovatif seperti tekanan pada penembak dan kekuatan lawan. Selain itu, terdapat pendekatan model probabilistik parametrik yang efisien seperti yang ditunjukkan oleh Wheatcroft & Sienkiewicz (2021).

Sebagai alternatif, banyak penelitian memanfaatkan *machine learning* untuk menangkap pola non-linear yang lebih kompleks. Beberapa di antaranya berfokus pada perbandingan komprehensif berbagai algoritma. Contohnya, Eggels *et al.* (2016) menguji empat model berbeda dan menemukan AdaBoost memiliki performa terbaik, sementara Cavus & Biecek (2022) menggunakan AutoML yang menobatkan *Random Forest* sebagai model paling optimal. Mead *et al.* (2023) juga mengonfirmasi keunggulan *Random Forest*, terutama setelah diperkaya dengan fitur tambahan seperti nilai pemain dan rating ELO.

Model jaringan saraf juga menjadi pilihan populer. Ruiz *et al.* (2015) dan Méndez *et al.* (2023) secara efektif menggunakan *Multilayer Perceptron* (MLP)

untuk menunjukkan superioritas model non-linear dalam menangkap pola kompleks dibandingkan regresi logistik. Pardo (2020) turut membandingkan *Artificial Neural Network* (ANN) dengan XGBoost untuk menganalisis pengaruh informasi kualitatif pemain terhadap kualitas peluang.

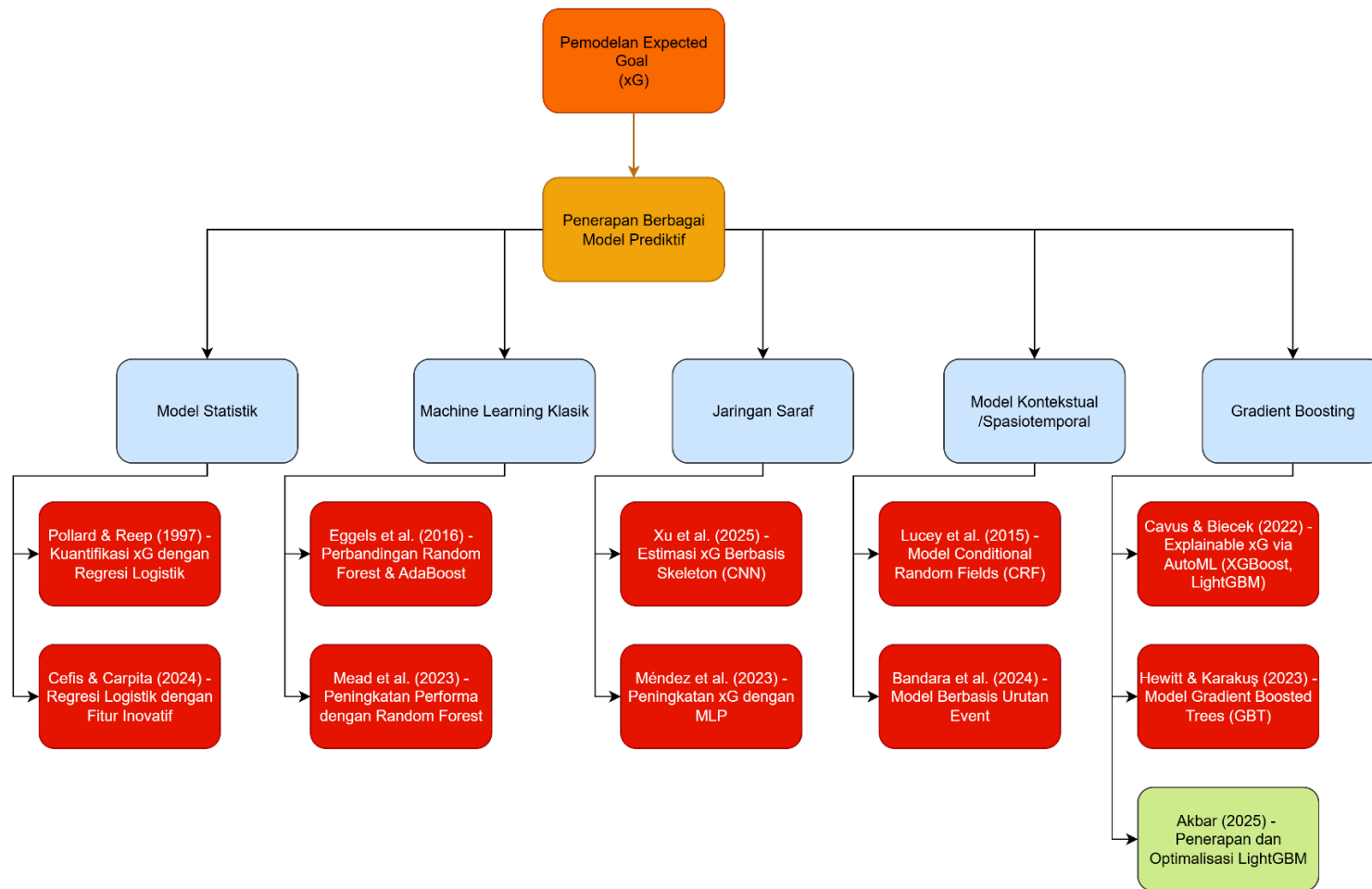
Lebih lanjut, terdapat penelitian yang mengembangkan model dengan fitur-fitur yang lebih canggih dan kontekstual. Lucey *et al.* (2015) menerapkan *Conditional Random Fields* untuk memasukkan informasi spasiotemporal, sementara Bandara *et al.* (2024) menggunakan *Random Forest* dengan fitur sekuensial dari tiga kejadian terakhir sebelum tembakan. Hewitt & Karakuş (2023) mengembangkan model *Gradient Boosted Trees* yang dapat menyesuaikan nilai xG dengan kemampuan individu pemain. Inovasi paling mutakhir datang dari Xu *et al.* (2025) yang memelopori penggunaan data pose tubuh pemain (*skeleton data*) dengan *Convolutional Neural Network* (CNN) untuk menghasilkan estimasi xG yang lebih akurat.

Penelitian ini bertujuan untuk menerapkan LightGBM untuk perhitungan metrik xG dalam analisis sepak bola. Perbedaan penelitian ini dengan penelitian sebelumnya yaitu:

1. Fokus pada penggunaan LightGBM sebagai metode utama untuk perhitungan xG, yang dikenal karena efisiensi dan kecepatan komputasinya.
2. Mengeksplorasi kemampuan LightGBM dalam menangani data sepak bola, yang sering kali memiliki interaksi fitur kompleks dan non-linear, dibandingkan dengan metode yang lebih sederhana seperti regresi logistik.

2.23 Ranah Penelitian

Pada tahap ini, menggambarkan ranah penelitian sejenis yang dilakukan penulis berdasarkan literatur yang dibandingkan, di mana berbagai metode mulai dari statistik hingga *machine learning* digunakan pada bidang analisis olahraga. Berdasarkan penelitian sejenis yang identik dengan ranah penelitian sebelumnya, maka ranah penelitian penulis berkaitan dengan bidang analisis sepak bola. Gambar 2.12 adalah ilustrasi dari ranah penelitian.



Gambar 2.12 Ranah Penelitian

Ranah pada penelitian ini adalah berfokus pada penerapan dan optimalisasi algoritma LightGBM untuk prediksi nilai xG. Data yang digunakan dalam penelitian ini adalah data *event* pertandingan sepak bola yang bersumber dari StatsBomb *Open Data*, yang mencakup berbagai kompetisi ternama di dunia. Hal yang membedakan penelitian ini dengan penelitian-penelitian sebelumnya adalah fokusnya yang mendalam pada optimalisasi algoritma LightGBM. Jika penelitian sebelumnya hanya menyertakan LightGBM sebagai salah satu pembanding dalam kerangka kerja AutoML, penelitian ini secara spesifik melakukan proses *hyperparameter tuning* dan kalibrasi untuk menggali potensi akurasi dan efisiensi komputasi yang sesungguhnya dari model. Selain itu, evaluasi model dilakukan secara komprehensif menggunakan serangkaian metrik holistik untuk memberikan gambaran kinerja yang utuh.

BAB III

METODOLOGI PENELITIAN

3.1 Pendekatan Penelitian

Penelitian ini menginvestigasi kinerja metode LightGBM dalam memprediksi nilai xG dari data tembakan pada pertandingan sepak bola dengan menggunakan pendekatan kuantitatif. Penelitian ini menggunakan bahasa pemrograman Python dan platform Google Colaboratory untuk proses pengambilan, pembersihan, dan pemodelan data. *Dataset* diambil dari repositori terbuka StatsBomb yang tersedia di GitHub. Microsoft Word digunakan untuk penyusunan laporan penelitian.

3.2 Sumber Data

Penelitian ini menggunakan pendekatan kuantitatif dengan memanfaatkan data sekunder yang bersifat publik (*publicly available*). Data yang digunakan bersumber dari StatsBomb Open Data, yang disediakan secara resmi dan gratis melalui repositori GitHub untuk mendorong riset serta inovasi di bidang analisis sepak bola.

Dalam konteks penelitian ini, populasi merujuk pada keseluruhan data peristiwa (*event data*) yang tersedia dalam repositori tersebut, sedangkan sampel adalah *subset* data yang dipilih secara spesifik untuk tujuan membangun dan mengevaluasi model prediksi xG.

3.2.1 Populasi dan Sampel Penelitian

Populasi data dalam penelitian ini adalah keseluruhan data peristiwa dari ribuan pertandingan sepak bola yang tersedia dalam StatsBomb Open Data. Populasi ini mencakup berbagai kompetisi elit di seluruh dunia:

- a. Liga-liga top Eropa (Contoh: La Liga, Serie A, Bundesliga).
- b. Kompetisi antarklub Eropa (Contoh: UEFA *Champions League*).
- c. Turnamen internasional (Contoh: FIFA *World Cup*, UEFA Euro).

Dari populasi yang luas tersebut, sampel penelitian diambil menggunakan metode *purposive sampling* (pengambilan sampel bertujuan). Metode ini diterapkan untuk memilih data yang paling relevan dengan tujuan pemodelan xG. Sampel akhir yang terkumpul adalah data peristiwa tembakan (*shot*) yang memenuhi serangkaian kriteria seleksi yang ketat.

3.2.2 Kriteria Seleksi Sampel

Proses seleksi sampel dilakukan untuk memastikan homogenitas data, karena probabilitas gol dari situasi yang berbeda memiliki karakteristik yang sangat berbeda dan sering kali dimodelkan secara terpisah. Kriteria seleksi yang diterapkan adalah sebagai berikut:

- a. Jenis Peristiwa (*event type*): Hanya peristiwa dengan jenis tembakan (*shot*) yang dimasukkan ke dalam *dataset*. Peristiwa lain seperti umpan, tekel, atau dribel akan diekstraksi untuk menghasilkan fitur turunan namun tidak menjadi bagian dari sampel utama.
- b. Situasi Permainan (*play pattern*): Sampel dibatasi hanya pada tembakan yang terjadi dari situasi permainan terbuka (*open play*). Tembakan yang berasal dari

situasi bola mati seperti penalti (*penalty*), tendangan bebas langsung (*direct free-kick*), atau situasi setelah tendangan sudut (*corner*) tidak diikutsertakan. Hal ini dilakukan untuk menjaga homogenitas data, karena probabilitas gol dari situasi bola mati memiliki karakteristik yang sangat berbeda dan sering dimodelkan secara terpisah.

Setelah melalui proses seleksi ini, terkumpul puluhan ribu data tembakan yang menjadi sampel final penelitian, yang kemudian dibagi menjadi data latih (training data) dan data uji (testing data) untuk keperluan pemodelan.

3.3 Perangkat Penelitian

Penelitian ini menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*) dengan spesifikasi yang dijelaskan pada Tabel 3.1.

Tabel 3.1 Spesifikasi *Hardware* dan *Software*

<i>Hardware</i>	Laptop Lenovo ADA 11	AMD Athlon Gold 3150U with Radeon Graphics 2.40 GHz
		12 GB RAM
		256 GB SSD
		Monitor 15 Inch
<i>Software</i>	Sistem Operasi	Windows 11 Home
	<i>Tools</i>	Google Colaboratory
	Bahasa Pemrograman	Python (Jupyter Notebook)

3.4 Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri atas data primer dan sekunder. Data primer diperoleh dari *dataset* terbuka yang disediakan oleh StatsBomb melalui repositori GitHub. Sementara itu, data sekunder diperoleh dari

berbagai jurnal ilmiah, buku, dan sumber internet yang relevan dengan topik penelitian, khususnya yang berkaitan dengan analisis xG, pemodelan prediktif, dan algoritma LightGBM.

Pengambilan data dilakukan dengan mengunduh *dataset* event pertandingan sepak bola dari repositori *open-source* StatsBomb di GitHub. Proses ini dilakukan menggunakan skrip Python di platform Google Colaboratory. *Dataset* yang digunakan mencakup data tembakan dalam pertandingan, termasuk informasi seperti lokasi, jarak, sudut tembakan, serta atribut kontekstual lainnya yang mendukung perhitungan nilai xG.

3.5 Pengembangan Model

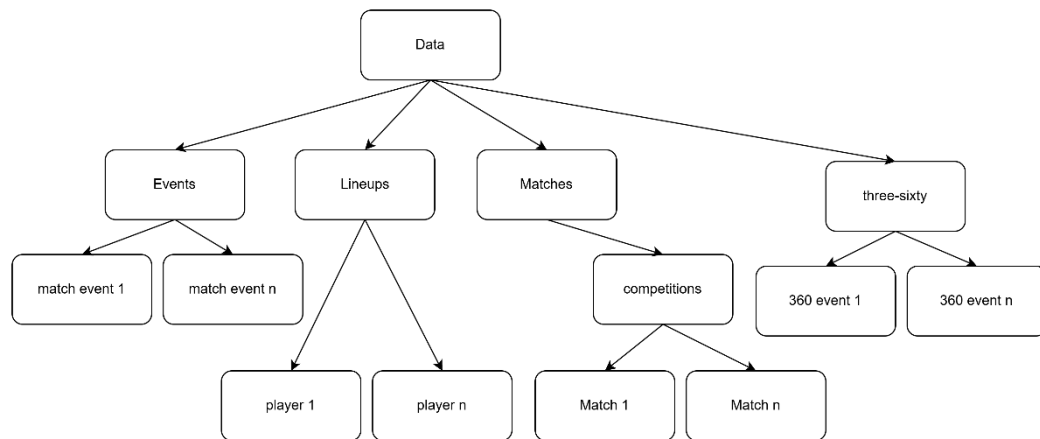
3.5.1 *Knowledge Discovery in Databases*

Penelitian ini menggunakan pendekatan *Knowledge Discovery in Databases* (KDD) dalam proses pengembangan model. Metode KDD memiliki keunggulan dalam membantu mengidentifikasi pola tersembunyi dari kumpulan data yang kompleks sehingga dapat menghasilkan informasi yang lebih mudah dipahami. Proses KDD terdiri dari beberapa tahapan, yaitu: *preprocessing* data, pemilihan data (*data selection*), transformasi data, proses *data mining*, dan evaluasi pengetahuan yang diperoleh (*knowledge evaluation*) (Ramos *et al.*, 2021).

a. *Data Selection*

Data dari StatsBomb *open-data* diambil dengan mengakses repositori resmi di GitHub. Pertama, kita perlu mengidentifikasi kompetisi apa saja yang tersedia dalam *dataset*. Setiap kompetisi kemudian terdiri dari beberapa musim (edisi),

dan masing-masing musim ini mewakili rentang waktu berlangsungnya pertandingan yang terdokumentasi. Di dalam setiap musim terdapat fase-fase pertandingan: untuk kompetisi sistem gugur biasanya meliputi babak perempat final, semi final, final, dan seterusnya, sedangkan untuk liga reguler umumnya hanya ada satu fase liga utama, dengan beberapa kompetisi seperti, Piala FA yang juga memiliki babak *play-off*. Setelah fase-fase ditentukan, barulah kita mengakses data pertandingan. Dalam konteks StatsBomb, satu pertandingan terdiri dari serangkaian *event*, dan masing-masing *event* ini dapat memiliki *event* terkait. Misalnya, sebuah tusukan (*dribble*) bisa jadi dipicu oleh operan rekan tim yang sebelumnya dieksekusi operan tersebut, kemudian tercatat sebagai *event* terkait. Namun, karena operan juga tercatat sebagai *event* utama, jika kita menarik semua *event* terkait tanpa seleksi, kita akan mendapati banyak duplikasi operan tercatat dua kali, sekali sebagai *event* utama dan sekali lagi sebagai *event* terkait. Sebaliknya, jika kita sama sekali mengabaikan *event* terkait, kita bisa kehilangan jejak kronologi aksi yang sebenarnya terjadi di lapangan. Untuk mengatasi masalah ini, saat ini hanya situasi gol dan kartu (kuning/merah) yang diikuti sebagai *event* terkait dalam pemrosesan data StatsBomb. Dengan begitu, kita tetap menjaga konteks penting seperti *assist* sebelum gol atau pelanggaran yang berujung kartu tanpa menumpuk terlalu banyak duplikasi. Pada Gambar 3.1 dijelaskan struktur data yang dimiliki oleh StatsBomb *open-data*.

Gambar 3.1 Struktur Data StatsBomb *open-data*.

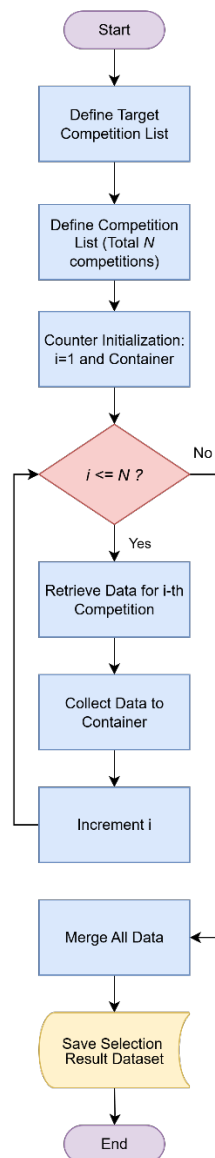
Data *event* dari StatsBomb disediakan dalam format JSON pada repositori GitHub mereka. Karena pengambilan data langsung dari GitHub juga memakan waktu, biasanya *file* JSON tersebut diunduh sekali saja lalu dikonversi dan disimpan dalam format *Parquet* untuk penggunaan selanjutnya. Dengan cara ini, analisis bisa dilakukan lebih cepat tanpa perlu terus-menerus mengunduh data mentah. Tabel 3.2 menunjukkan *field* dari setiap kategori data.

Tabel 3.2 *Field* Data Statsbomb

Kategori Data	Nama <i>Field</i>	Deskripsi	Contoh Nilai
Kompetisi	<i>competition_name</i>	Nama kompetisi.	FIFA <i>World Cup</i>
	<i>season_name</i>	Nama musim kompetisi.	2022
Pertandingan	<i>match_date</i>	Tanggal pertandingan berlangsung.	2022-12-18
	<i>home_team</i>	Objek data tim tuan rumah.	{ "home_team_name": "Argentina", ... }
	<i>away_team</i>	Objek data tim tamu.	{ "away_team_name": "France", ... }

	<i>home_score</i>	Skor akhir tim tuan rumah.	3
	<i>away_score</i>	Skor akhir tim tamu.	3
Susunan Pemain	<i>player_name</i>	Nama lengkap pemain.	Lionel Messi
	<i>jersey_number</i>	Nomor punggung pemain.	10
	<i>positions</i>	Daftar posisi yang dimainkan pemain.	[{"position": "Right Center Forward", ...}]
Event	<i>type</i>	Jenis aksi yang terjadi (Umpan, Tembakan, dll.).	{"name": "Pass" }
	<i>minute</i>	Menit beberapa aksi terjadi.	2
	<i>player</i>	Objek pemain yang melakukan aksi.	{"name": "Kylian Mbappé Lottin" }
	<i>location</i>	Koordinat [x, y] di lapangan tempat aksi dimulai.	[65,3, 22,8]
	<i>pass_outcome</i>	Hasil dari sebuah umpan (Selesai, Gagal, dll.).	{"name": "Incomplete" }
	<i>shot_statsbomb_xg</i>	Nilai xG dari sebuah tembakan.	0,087
	<i>shot_outcome</i>	Hasil dari sebuah tembakan (Gol, Diselamatkan, dll.).	{"name": "Goal" }
	<i>dribble_outcome</i>	Hasil dari sebuah dribel (Selesai, Gagal).	{"name": "Complete" }
	<i>duel_type</i>	Jenis duel yang terjadi (Tekel, Udara, dll.).	{"name": "Tackle" }

Tabel 3.2 menunjukkan sebuah arsitektur data yang hierarkis dan sangat terperinci, dirancang untuk analisis performa sepak bola yang mendalam. Data terorganisir mulai dari level makro, yaitu kompetisi dan musim, yang kemudian dipecah menjadi unit-unit pertandingan individual. Setiap pertandingan memiliki data kontekstual seperti skor, tim, dan informasi susunan pemain. Puncak kedetailan data terletak pada level *event*, yang secara granular mencatat setiap aksi signifikan di lapangan mulai dari umpan, tembakan, hingga duel lengkap dengan atribut-atribut kunci seperti lokasi spasial (koordinat x,y), waktu, pemain yang terlibat, serta hasil dari aksi tersebut. Gambar 3.2 menunjukkan *flowchart* dari *data selection*.

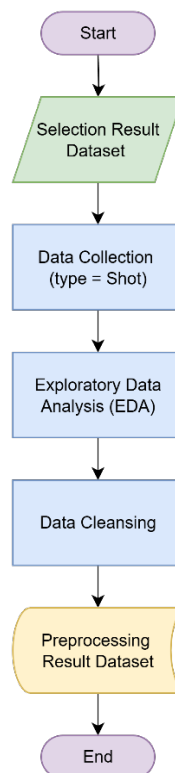


Gambar 3.2 Flowchart Data Selection

b. Data Preprocessing

Tahap *data preprocessing* bertujuan untuk menyiapkan data mentah agar optimal untuk dianalisis dan digunakan dalam pemodelan. Proses ini diawali dengan *data collection* dengan menyaring data untuk mengambil hanya *event* bertipe *shot* dari situasi permainan terbuka (*open play*) yang paling relevan

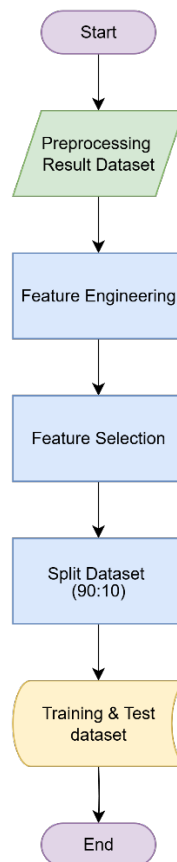
dengan konteks prediksi xG. Selanjutnya, pada tahap *Exploratory Data Analysis* (EDA), dilakukan analisis untuk memahami karakteristik data sekaligus memilih fitur-fitur yang paling berpotensi mendukung prediksi, seperti posisi tembakan dan bagian tubuh yang digunakan, sementara kolom yang tidak relevan akan dihilangkan. Langkah berikutnya adalah *data cleansing*, yang meliputi pemeriksaan nilai kosong atau duplikat untuk memastikan integritas data, meskipun data StatsBomb yang terstruktur umumnya sudah bersih. Keseluruhan proses ini menghasilkan *dataset* yang konsisten dan siap untuk tahap transformasi serta pemodelan, seperti yang diilustrasikan alurnya pada Gambar 3.2.



Gambar 3.3 *Flowchart Data Preprocessing*

c. *Data Transformation*

Tahap ini bertujuan untuk memperkaya representasi data agar dapat meningkatkan performa model pada tahapan *data mining*. Pertama, dilakukan proses *feature engineering* untuk menciptakan fitur-fitur baru yang merepresentasikan dinamika permainan secara lebih mendalam. Transformasi ini memungkinkan data mentah memberikan wawasan yang lebih bermakna dan relevan dalam konteks prediksi performa tembakan. Fitur-fitur seperti jarak dan sudut tembakan ke gawang serta segmentasi waktu pertandingan ditambahkan untuk memperkaya informasi spasial dan temporal. Setelah fitur baru ditambahkan, data kemudian dibagi menjadi data latih dan data uji agar proses pelatihan dan evaluasi model dapat dilakukan secara terpisah. Alur tahapan *transformation* ditunjukkan pada Gambar 3.4.

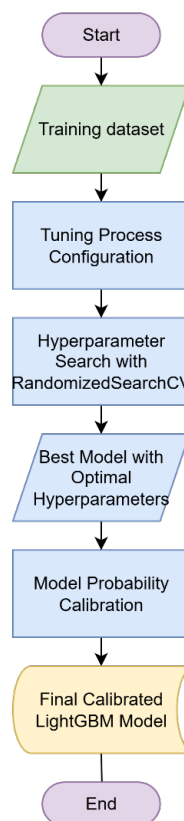


Gambar 3.4 Flowchart Data Transformation

d. *Data Mining*

Pada tahapan ini pemodelan xG dilakukan menggunakan algoritma LightGBM. Namun sebelum model dilatih, terdapat beberapa proses penting yang harus dilakukan, yaitu pencarian *hyperparameter* terbaik dan kalibrasi probabilitas. Pencarian *hyperparameter* dilakukan dengan menggunakan *RandomizedSearchCV* sebanyak 100 iterasi, yang mengevaluasi berbagai kombinasi parameter dengan *5-fold cross-validation*. Proses ini menggunakan metrik skor *roc_auc* sebagai acuan untuk menentukan kombinasi parameter terbaik dan secara otomatis melakukan *refit* pada model dengan skor tersebut.

Setelah memperoleh model dengan konfigurasi terbaik, dilakukan kalibrasi probabilitas menggunakan *CalibratedClassifierCV* untuk memastikan bahwa prediksi probabilitas dari model merefleksikan tingkat kepercayaannya secara akurat (Davis & Robberechts, 2024). Selain pelatihan dan kalibrasi, tahap ini juga mencakup analisis fitur untuk memahami kontribusi tiap variabel dalam proses prediksi. Gambar 3.5 menunjukkan alur dari tahapan data *mining* dalam penelitian ini.



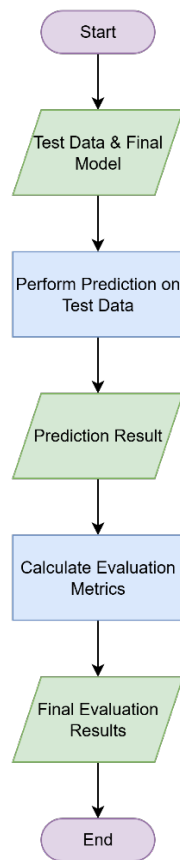
Gambar 3.5 *Flowchart Data Mining*

e. *Evaluation*

Setelah proses *data mining* selesai, tahap selanjutnya adalah evaluasi terhadap model yang telah dibuat. Evaluasi ini bertujuan untuk mengukur performa

model secara komprehensif terhadap data uji. Sesuai dengan batasan masalah, evaluasi kinerja model akan menggunakan serangkaian metrik yang mencakup ROC AUC, *Brier Score*, presisi, *recall*, F1-Score, dan *Log-Loss*.

ROC AUC digunakan untuk menilai kemampuan diskriminatif model, yaitu kemampuannya dalam membedakan antara kelas positif dan negatif secara keseluruhan tanpa terikat pada ambang batas klasifikasi tertentu. Untuk mengukur akurasi dari prediksi probabilistik, digunakan *Brier Score* yang menghitung rata-rata selisih kuadrat antara probabilitas prediksi dengan hasil aktual, sehingga efektif dalam menilai kalibrasi model. Serupa dengan itu, *Log-Loss* juga memberikan penalti untuk prediksi yang tingkat keyakinannya tidak sesuai dengan hasil aktual. Terakhir, untuk evaluasi yang lebih bernuansa pada tugas klasifikasi, digunakan akurasi, presisi, *recall*, dan F1-Score yang menganalisis keseimbangan antara keandalan prediksi positif (Presisi) dan kelengkapan dalam mengidentifikasi kasus positif (*recall*). *Flowchart* dari tahapan evaluasi model ditunjukkan pada Gambar 3.6.



Gambar 3.6 *Flowchart Evaluation*

3.5.2 Pemodelan LightGBM

Pada penelitian ini, metode yang digunakan adalah LightGBM (*Light Gradient Boosting Machine*) untuk membangun model prediksi. LightGBM dirancang untuk menangani data berukuran besar dengan efisiensi tinggi melalui dua teknik utama: *Gradient-based One-Side Sampling* (GOSS) dan *Exclusive Feature Bundling* (EFB).

Teknik GOSS berfokus pada efisiensi pelatihan model dengan mempertahankan seluruh data yang memiliki nilai gradien besar yang mengandung lebih banyak informasi dan secara acak mengambil sebagian dari data dengan

gradien kecil (Ke *et al.*, 2017). Namun, karena proses ini dapat mengubah distribusi data asli, LightGBM memperkenalkan pengali konstan saat menghitung *information gain* untuk data dengan gradien kecil guna menyeimbangkan kontribusi antara dua kelompok data tersebut. Pendekatan ini memungkinkan model untuk tetap fokus pada sampel yang paling berpengaruh terhadap pembaruan model tanpa kehilangan akurasi secara signifikan.

Sementara itu, teknik EFB dirancang untuk mengatasi tantangan ketika terdapat banyak fitur yang bersifat saling eksklusif, yaitu fitur-fitur yang tidak pernah aktif secara bersamaan. Algoritma ini menggabungkan fitur-fitur eksklusif tersebut ke dalam fitur padat (*dense feature*) dalam jumlah yang jauh lebih sedikit, sehingga mengurangi dimensi data dan beban komputasi (Ke *et al.*, 2017). Selain itu, LightGBM juga mengoptimalkan algoritma histogram dasar dengan cara mengabaikan nilai nol pada fitur, yakni dengan mencatat hanya nilai-nilai non-nol menggunakan struktur data khusus. Kombinasi dari GOSS dan EFB menjadikan LightGBM sangat efisien dan *scalable* dalam membangun model prediksi dari *dataset* dengan jumlah *instance* dan fitur yang sangat besar.

3.6 Analisis Data dan Interpretasi Hasil

Analisis data dalam penelitian ini dilakukan berdasarkan pendekatan KDD (*Knowledge Discovery in Database*) yang mencakup lima tahapan utama: *data selection*, *preprocessing*, *transformation*, *data mining*, dan *evaluation*. Proses analisis dimulai dari tahap *data selection*, yaitu dengan menyiapkan *dataset* yang relevan untuk membangun model prediksi. Tahap selanjutnya adalah *preprocessing*

yang meliputi pembersihan data, penanganan *missing value*, penghapusan duplikasi.

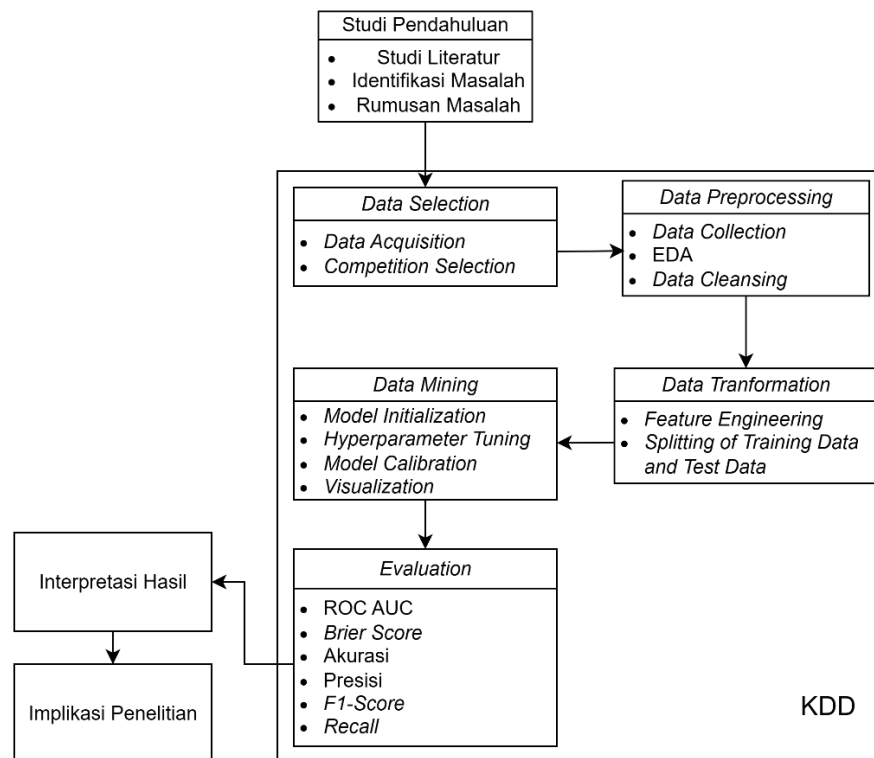
Pada tahap *transformation*, dilakukan pembagian data menjadi data latih dan data uji, serta dilakukan transformasi fitur agar sesuai dengan kebutuhan algoritma yang digunakan. Tahap data *mining* dilakukan dengan membangun model prediksi menggunakan algoritma LightGBM, serta melakukan *hyperparameter tuning* menggunakan *RandomizedSearchCV* untuk memperoleh kombinasi parameter terbaik berdasarkan nilai skor ROC AUC.

Kemudian, pada tahap evaluasi, performa model diukur secara komprehensif menggunakan serangkaian metrik. Metrik-metrik tersebut meliputi ROC AUC, *Brier Score*, akurasi, presisi, *recall*, *F1-Score*, dan *Log-Loss*, serta perbandingan kecepatan waktu komputasi terhadap model serupa. Pengujian ini bertujuan untuk menilai performa model dari berbagai aspek, mulai dari kemampuan diskriminatif, akurasi probabilistik, hingga efisiensi pemrosesan.

Interpretasi hasil evaluasi tersebut akan ditinjau dari tiga perspektif utama untuk memberikan konteks yang menyeluruh. Pertama, melalui tolok ukur akademis (*academic benchmark*). Kedua, dari sisi aplikasi praktis (*practical application*). Terakhir, performa model juga akan diukur terhadap tolok ukur industrial (*industrial benchmark*). Hasil dari seluruh tahapan analisis ini serta interpretasi terhadap performa model akan dijelaskan secara rinci pada Bab 4.

3.7 Tahapan Penelitian

Penelitian ini diawali dengan studi literatur mendalam terhadap sumber akademis untuk memahami kondisi terkini (*state-of-the-art*) dan merumuskan masalah, yang menjadi landasan bagi penerapan kerangka kerja *Knowledge Discovery in Database* (KDD). Proses KDD dimulai dengan tahap *data selection* yaitu pengumpulan *dataset* publik dari GitHub yang berisi catatan peristiwa pertandingan sepak bola, yang dilanjutkan dengan *preprocessing* komprehensif meliputi pemilihan variabel, pembersihan data dari nilai hilang dan duplikat, *encoding* fitur kategorial, serta normalisasi data numerik. Selanjutnya, pada tahap *transformation*, dilakukan rekayasa fitur (*feature engineering*), analisis korelasi, seleksi fitur penting, dan pembagian data menjadi set pelatihan dan pengujian. Tahap inti *data mining* berfokus pada penggunaan algoritma LightGBM yang dioptimalkan melalui proses *tuning hyperparameter* dengan *RandomizedSearchCV* dan diperkuat oleh kalibrasi model untuk memastikan akurasi prediksi probabilistik. Pada tahap akhir, *evaluation* dilakukan secara komprehensif menggunakan serangkaian metrik mencakup ROC AUC, *Brier Score*, presisi, *recall*, *F1-Score*, dan *Log-Loss* untuk menilai kinerja model dari aspek diskriminatif hingga akurasi probabilistik, sebelum penelitian ditutup dengan interpretasi hasil, analisis implikasi, penarikan kesimpulan, dan saran untuk riset selanjutnya. Tahapan penelitian yang dilakukan dapat dilihat pada Gambar 3.7.



Gambar 3.7 Tahapan Penelitian

3.8 Jadwal Penelitian

Rencana waktu pelaksanaan penelitian ditunjukkan pada Tabel 3.3.

Tabel 3.3 Waktu Pelaksanaan Penelitian

No.	Tahapan	Maret 2025	April 2025	Mei 2025	Juni 2025	Juli 2025	Agustus 2025
1	Studi Pendahuluan & Landasan Teori						
2	Pengumpulan & Seleksi Data						
3	Pra-pemrosesan & Transformasi Data						

4	Pemodelan, Tuning, & Kalibrasi Model						
5	Evaluasi Model & Interpretasi Hasil						
6	Penyusunan Laporan & Kesimpulan						

BAB IV

HASIL DAN PEMBAHASAN

4.1 *Data Selection*

Tahap *data selection* merupakan langkah awal dalam proses analisis di mana data yang relevan dipilih dari sumber yang tersedia untuk digunakan dalam penelitian. Pada penelitian ini, data diperoleh dari repositori terbuka StatsBomb melalui GitHub, yang menyediakan data *event* pertandingan sepak bola dalam format JSON. Dari seluruh data yang tersedia, hanya data dengan tipe *event Shot* yang dipilih karena fokus penelitian adalah memprediksi kemungkinan terciptanya gol dari sebuah tembakan. Selain itu, hanya kolom-kolom tertentu yang dipilih, seperti informasi tentang teknik tembakan, bagian tubuh yang digunakan, pola permainan, serta posisi awal tembakan, karena kolom-kolom tersebut dianggap memiliki relevansi langsung terhadap peluang mencetak gol. Proses ini bertujuan untuk menyederhanakan *dataset* dan memastikan bahwa hanya fitur yang bermakna yang digunakan dalam tahap analisis selanjutnya.

4.1.1 *Data Acquisition*

Penelitian ini menggunakan data dari repositori GitHub StatsBomb open-data yang diambil melalui proses unduhan menggunakan *tool* aria2 di Google Colab dengan bahasa pemrograman Python. Untuk mengakses data tersebut, pertama-tama dilakukan pengunduhan *file master* yang berisi data pertandingan sepak bola. Berikut adalah tahapan yang dilakukan dalam proses pengumpulan data:

- a. Mengunduh *file* master dari repositori GitHub StatsBomb open-data dengan menggunakan perintah di Google Colab dan *tool* aria2 untuk mempercepat proses unduhan.
- b. Menyusun skrip Python di Google Colab untuk mengekstrak semua *event* data yang ada pada *file* yang diunduh.
- c. Mengkonversi *event* data menjadi format *dataframe* menggunakan *pandas* untuk mempermudah pengolahan data lebih lanjut.
- d. Menyimpan *dataframe* yang telah diproses dalam format *parquet* untuk memudahkan analisis data selanjutnya. *Dataframe* yang dihasilkan mencakup berbagai kolom terkait informasi pertandingan, yang akan diseleksi dan diproses lebih lanjut untuk analisis yang lebih mendalam.

4.1.2 *Competition Selection*

Langkah ini bertujuan untuk memastikan bahwa hanya data pertandingan yang relevan dan sesuai dengan fokus penelitian yang digunakan dalam proses analisis. Data mentah yang tersedia di repositori *open-data* StatsBomb mencakup berbagai jenis kompetisi, termasuk pertandingan pria, wanita, dan kelompok usia muda. Oleh karena itu, proses seleksi dilakukan secara sistematis untuk menyaring data berdasarkan dua kriteria utama: jenis kelamin peserta dan tingkat kompetisi.

Data kompetisi difilter untuk hanya menyertakan pertandingan pria dengan memeriksa atribut *competition_gender* yang bernilai '*male*'. Selanjutnya, untuk memastikan bahwa hanya kompetisi tingkat senior yang disertakan, dilakukan pengecualian terhadap kompetisi yang mengandung kata kunci seperti 'U21', 'U23', 'U18', dan lainnya dalam nama kompetisi, yang menunjukkan kelompok usia muda.

Setelah mendapatkan daftar kompetisi yang valid, data pertandingan (*matches*) dari kompetisi tersebut dimuat dan difilter lebih lanjut untuk hanya menyertakan pertandingan antara dua tim pria. Proses ini menghasilkan kumpulan data pertandingan yang sesuai dengan fokus penelitian, yaitu analisis pertandingan sepak bola pria tingkat senior. Tabel 4.1 Menunjukkan Daftar Kompetisi yang akan digunakan.

Tabel 4.1 Daftar Kompetisi

<i>Competition ID</i>	<i>Competition Name</i>
11	<i>FIFA World Cup</i>
2	<i>Premier League</i>
37	<i>La Liga</i>
72	<i>UEFA Champions League</i>
43	<i>Bundesliga</i>
49	<i>Serie A</i>
4	<i>Ligue 1</i>
55	<i>Copa America</i>
9	<i>African Cup of Nations</i>
16	<i>Eredivisie</i>

4.2 *Data Preprocessing*

Tahap *preprocessing* adalah tahapan yang berisi serangkaian proses untuk membersihkan dan menyiapkan data agar siap digunakan dalam analisis dan pemodelan pada tahapan selanjutnya. Dengan *preprocessing* yang tepat, kualitas data meningkat dan hasil pemodelan di tahap berikutnya menjadi lebih akurat dan andal.

4.2.1 *Data Collection*

Pemilihan jenis *event* yang tepat sangat krusial untuk memastikan relevansi dan kualitas analisis. Berdasarkan dokumentasi resmi dari StatsBomb, setiap peristiwa dalam pertandingan dikategorikan dengan *identifier* unik. Event dengan

$type.id = 16$ merepresentasikan "Shot" atau tembakan, yang menjadi fokus utama dalam model xG karena langsung berkaitan dengan upaya mencetak gol.

Setelah data kompetisi didapatkan, langkah selanjutnya adalah mengambil seluruh data *event* yang relevan untuk setiap *match_id*. Untuk kepentingan *feature engineering* di tahap selanjutnya, peneliti tidak hanya mengambil *event* tembakan itu sendiri, tetapi juga data rangkaian peristiwa yang terjadi sebelumnya (*pre-shot events*) dengan menaruhnya pada kolom baru yaitu *type_before* sebagai bagian tahap *feature engineering* yang akan dijelaskan secara rinci pada sub-bab 4.3.1. Seluruh data *event* ini kemudian disimpan dalam format *parquet* dan *csv*. Dari kumpulan data tersebut, *event* tembakan ($type.id = 16$) akan diisolasi untuk menjadi *dataset* utama dalam pelatihan dan pengujian model. Hasil akhir dari data tembakan ini ditunjukkan pada Gambar 4.1.

period	minute	second	start_x	start_y	team_name	player_name	end_x	end_y	type	...
1	7	15	115.4	29.4	England	Harry Maguire	120.0	34.9	16	...
1	26	58	101.1	55.3	England	Bukayo Saka	117.5	41.9	16	...
1	29	8	113.4	49.1	England	Mason Mount	120.0	45.0	16	...
1	31	47	110.5	40.7	England	Harry Maguire	120.0	36.8	16	...
1	34	9	112.0	38.0	England	Jude Bellingham	120.0	43.0	16	...

Gambar 4.1 Hasil Pemilihan Data

4.2.2 Column Selection

Pada tahapan ini, proses awal yang dilakukan adalah memilih sub set data yang sesuai dengan tujuan penelitian. Dalam model xG, hanya data dengan tipe *event shot* yang diambil seperti yang dilakukan pada sub-bab 4.2.1. Setelah *event*

shot teridentifikasi, dilakukan proses seleksi kolom yang dianggap memiliki nilai prediktif terhadap hasil tembakan (*shot outcome*).

Data *event* mentah dari StatsBomb sangat komprehensif, mencakup lebih dari 40 kolom yang mendeskripsikan setiap aspek dari sebuah *event*. Namun, tidak semua kolom ini relevan atau memberikan nilai prediktif untuk hasil akhir tembakan (*shot outcome*). Oleh karena itu, dilakukan proses seleksi ketat untuk menyaring dan memilih 16 kolom kunci yang dianggap memiliki pengaruh paling signifikan. Kolom-kolom yang dipilih mencakup atribut spasial (seperti posisi awal dan akhir tembakan), temporal (menit dan detik), teknis (teknik tembakan, bagian tubuh yang digunakan), serta konteks permainan (tekanan lawan, pola permainan, dan tipe *event* sebelumnya).

Kriteria pemilihan kolom ini mengacu pada metodologi yang divalidasi oleh penyedia data, di mana atribut spasial, teknis, dan kontekstual seperti posisi tembakan, tekanan pada penembak, dan pola permainan adalah fitur esensial dalam membangun model xG yang akurat (Hudl, 2024). Hasil seleksi kolom ditampilkan pada Tabel 4.2.

Tabel 4.2 Nama dan Deskripsi Kolom

Nama Kolom	Deskripsi
<i>period</i>	Periode / babak pertandingan saat tembakan terjadi
<i>minute</i>	Menit pertandingan saat tembakan dilakukan
<i>second</i>	Detik pertandingan saat tembakan dilakukan
<i>Location</i> (dipecah menjadi <i>start_x</i> dan <i>start_y</i>)	Koordinat awal tembakan (arah serangan selalu menuju gawang lawan di sisi kanan)
<i>position</i>	Posisi pemain di dalam tim (Bek, Gelandang, Penyerang)
<i>shot_outcome</i>	Hasil tembakan (0 = tidak gol, 1 = gol)
<i>shot_body_part</i>	Bagian tubuh yang digunakan dalam menembak

<i>shot_first_time</i>	Apakah tembakan dilakukan secara langsung tanpa kontrol bola
<i>shot_one_on_one</i>	Apakah tembakan dilakukan dalam situasi satu lawan satu dengan kiper
<i>shot_open_goal</i>	Apakah tembakan dilakukan ke gawang yang kosong
<i>shot_aerial_won</i>	Apakah pemain memenangkan duel udara sebelum tembakan
<i>shot_key_pass</i>	Apakah tembakan didahului oleh umpan kunci
<i>possession</i>	Nomor penguasaan bola dari tim pada pertandingan
<i>play_pattern</i>	Pola permainan yang terjadi sebelum tembakan (<i>Regular Play, From Corner, From Free Kick</i>)
<i>under_pressure</i>	Apakah pemain berada dalam tekanan saat melakukan tembakan
<i>shot_technique</i>	Teknik tembakan yang digunakan (<i>Normal, Volley, Half Volley, Lob, Bicycle Kick</i>)

Pemilihan variabel ini bertujuan untuk menyederhanakan kompleksitas data serta meningkatkan fokus pada fitur-fitur yang relevan dalam konteks perhitungan xG. Langkah ini dilakukan untuk mengurangi redundansi informasi dan meminimalkan risiko *overfitting* akibat penggunaan variabel yang tidak informatif. Selain itu, untuk fitur-fitur yang bersifat kategorial, dilakukan pendekatan dengan mengambil langsung nilai ID atau representasi numerik yang sudah tersedia dari masing-masing kategori. Dengan demikian, proses ini menghindari kebutuhan akan transformasi tambahan seperti *one-hot encoding* atau *label encoding*. Pendekatan ini tidak hanya menjaga efisiensi pemrosesan data, tetapi juga mempertahankan struktur semantik dari variabel kategorial dalam bentuk yang lebih ringkas dan langsung digunakan oleh model. Gambar 4.2 menunjukkan contoh data setelah proses pemilihan variabel dilakukan.

	period	minute	second	start_x	start_y	end_x	end_y	position	shot_type	shot_outcome	shot_body_part
0	1	1.0	42.0	111.0	52.0	113.0	51.0	17	87	0	40
1	1	4.0	47.0	96.0	43.0	115.0	40.5	22	87	0	40
2	1	8.0	37.0	107.0	43.0	117.0	39.0	5	87	0	37
3	1	17.0	26.0	111.0	58.0	111.0	53.0	17	87	0	38
4	1	21.0	16.0	105.0	41.0	105.0	41.0	12	87	0	38

Gambar 4.2 Contoh Data Sesudah Pemilihan Variabel

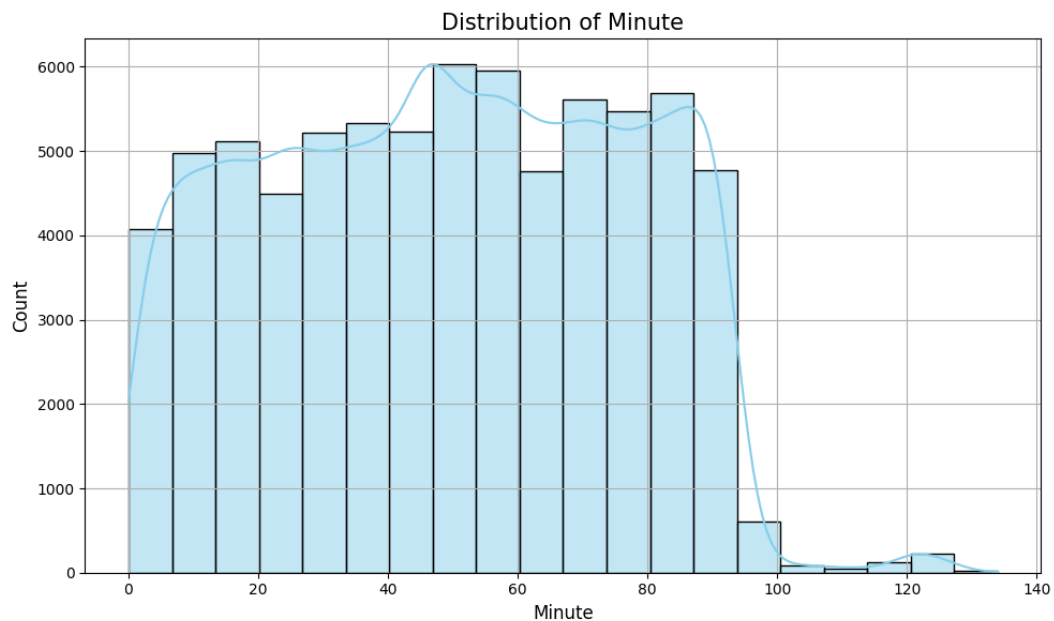
4.2.3 *Exploratory Data Analysis (EDA)*

Tahap EDA merupakan fase investigasi awal yang krusial untuk memahami karakteristik, pola, anomali, dan hubungan yang terdapat dalam *dataset*. Tujuan utama dari EDA dalam penelitian ini adalah untuk menggali wawasan dari data tembakan yang telah diproses sebelumnya. Melalui serangkaian teknik visualisasi dan statistik deskriptif, EDA membantu memvalidasi asumsi, mengidentifikasi fitur-fitur yang berpotensi memiliki nilai prediktif tinggi, serta memahami distribusi dan korelasi antar variabel. Proses ini menjadi tahapan penting sebelum melangkah ke tahap pemodelan, karena pemahaman yang mendalam terhadap data memungkinkan pemilihan strategi pemodelan yang lebih tepat dan efektif.

Analisis ini mencakup visualisasi distribusi untuk setiap variabel kunci serta analisis hubungan antara variabel-variabel tersebut dengan variabel target, yaitu hasil tembakan (*shot outcome*).

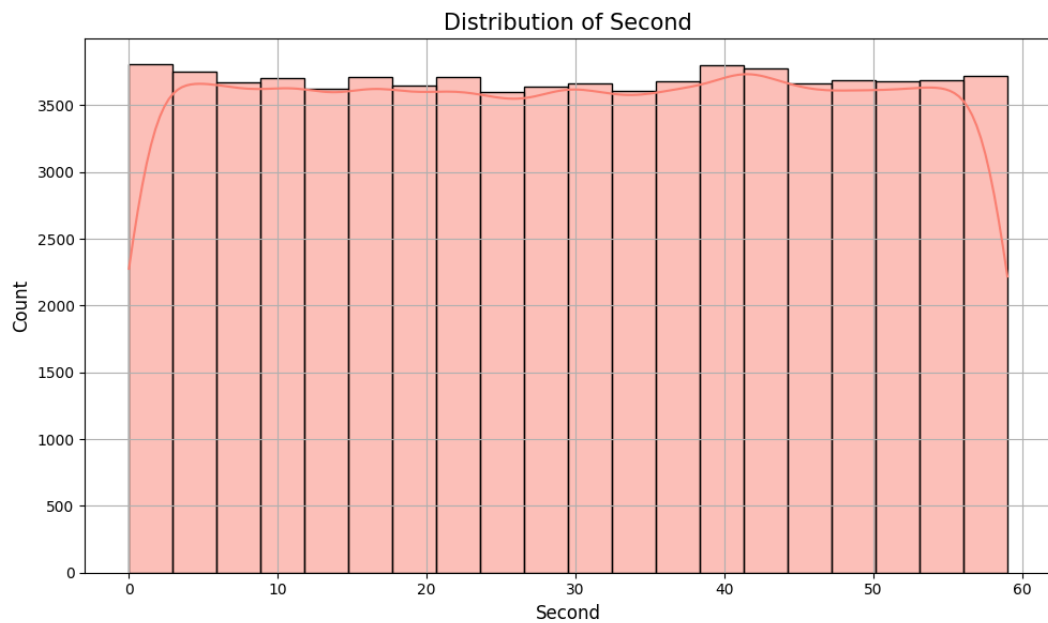
4.2.3.1 Analisis Distribusi Variabel Tunggal (Univariat)

Analisis univariat dilakukan untuk memahami distribusi dari setiap fitur individual yang akan digunakan dalam model. Gambar 4.3 menunjukkan distribusi jumlah tembakan berdasarkan menit pertandingan.



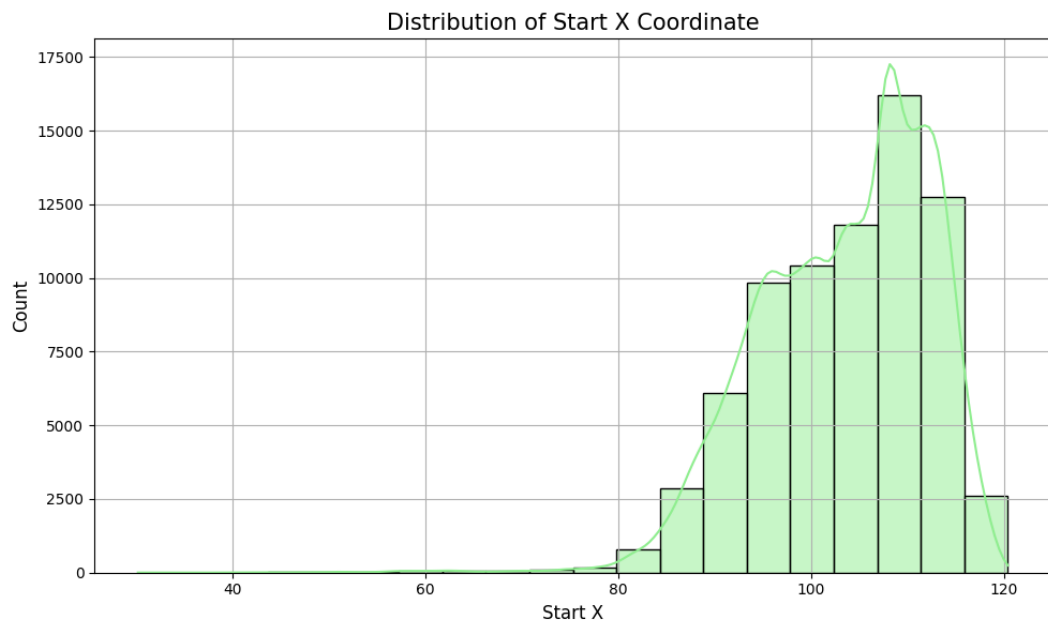
Gambar 4.3 Distribusi Menit Pertandingan

Pada Gambar 4.3, dapat diamati bahwa distribusi tembakan tidak seragam sepanjang pertandingan. Terdapat dua puncak utama yang signifikan, yaitu menjelang akhir babak pertama (sekitar menit ke-45) dan menjelang akhir babak kedua (sekitar menit ke-90). Peningkatan intensitas serangan pada periode ini, di mana tim berusaha keras untuk mencetak gol sebelum jeda atau sebelum pertandingan berakhir, menjadi penyebab utama lonjakan jumlah tembakan. Selain itu, terlihat adanya aktivitas tembakan pada periode waktu tambahan (setelah menit ke-90 dan ke-120), yang merefleksikan pertandingan yang berlanjut ke babak perpanjangan waktu. Selanjutnya, Gambar 4.4 menunjukkan distribusi tembakan berdasarkan detik dalam satu menit.



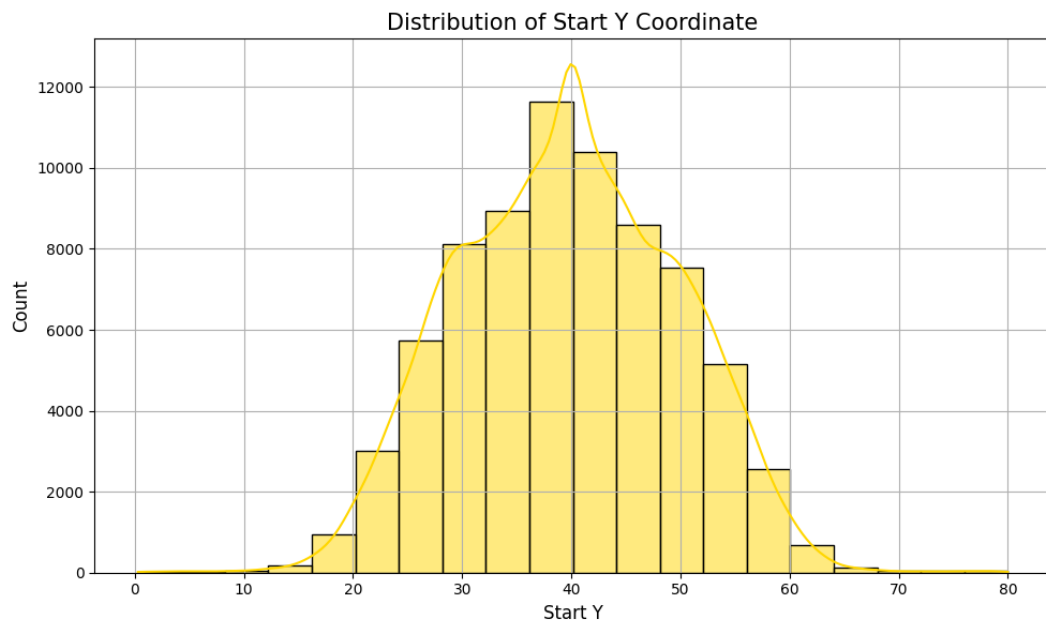
Gambar 4.4 Distribusi Detik Pertandingan

Pada Gambar 4.4, terlihat bahwa distribusi tembakan relatif seragam di sepanjang rentang 0 hingga 60 detik. Hal ini mengindikasikan bahwa tidak ada pola waktu spesifik dalam skala detik di mana sebuah tembakan lebih mungkin terjadi. Dengan kata lain, peluang terjadinya tembakan tersebar secara merata di setiap detik dalam satu menit permainan, yang sesuai dengan sifat dinamis dan acak dari momen-momen di dalam pertandingan sepak bola. Gambar 4.5 memvisualisasikan distribusi lokasi awal tembakan pada sumbu x (panjang lapangan).



Gambar 4.5 Distribusi Koordinat *Start x*

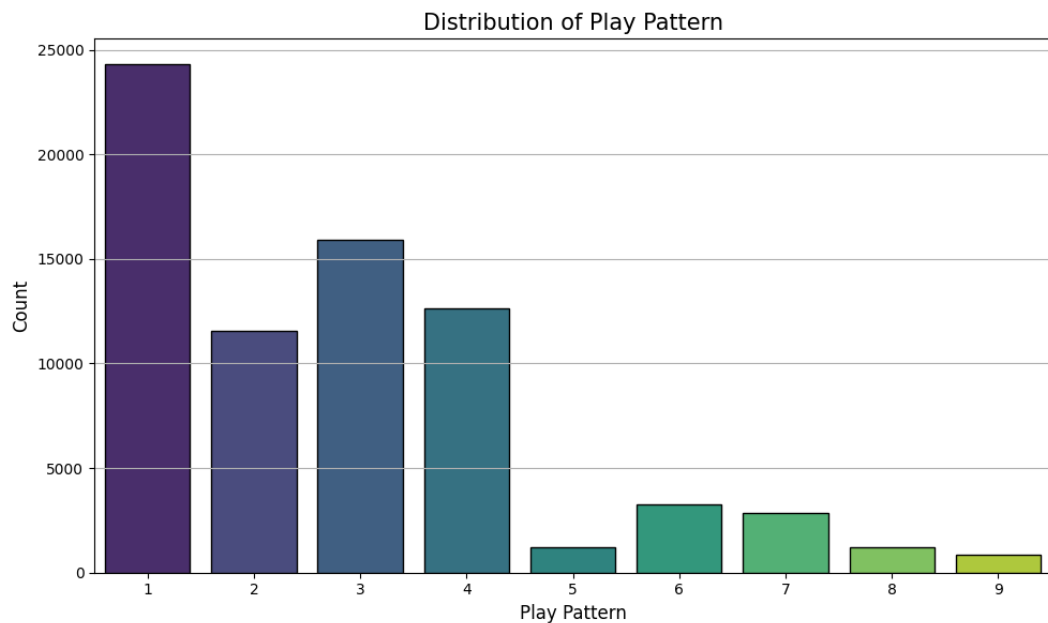
Pada Gambar 4.5, distribusi koordinat *start_x* menunjukkan konsentrasi tembakan yang sangat tinggi pada area sepertiga akhir lapangan (nilai x antara 90 hingga 120). Puncak distribusi berada di sekitar nilai $x = 110$, yang berdekatan dengan area penalti. Pola ini sangat logis, karena tim cenderung melepaskan tembakan saat berada sedekat mungkin dengan gawang lawan untuk memaksimalkan peluang gol. Distribusi yang condong ke kiri (*left-skewed*) ini menegaskan bahwa mayoritas aksi ofensif yang berujung pada tembakan terjadi di wilayah pertahanan lawan. Gambar 4.6 memvisualisasikan distribusi lokasi awal tembakan pada dan sumbu y (lebar lapangan).



Gambar 4.6 Distribusi Koordinat *Start y*

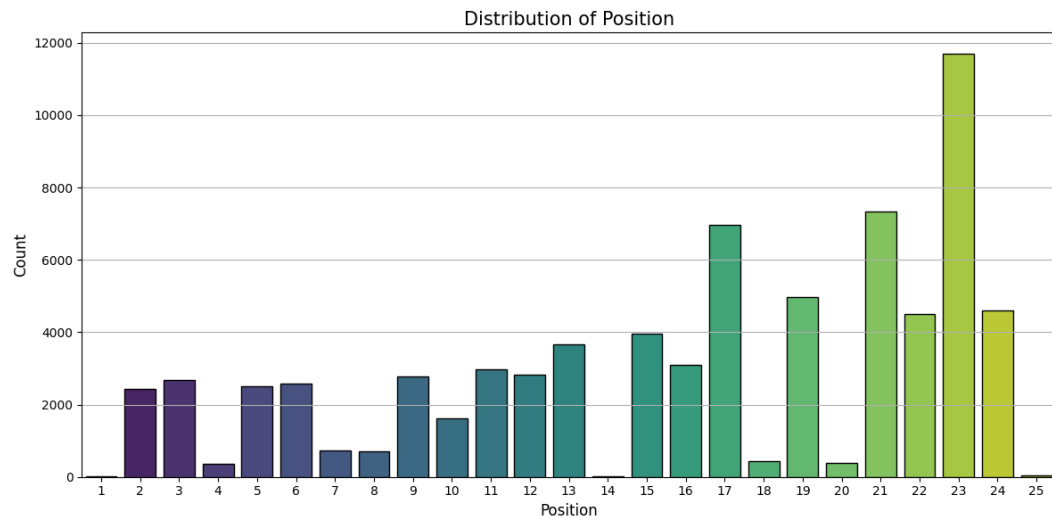
Pada Gambar 4.6, distribusi koordinat *start_y* menunjukkan pola *unimodal* yang simetris dengan puncak berada di sekitar nilai $y = 40$, yang merupakan titik tengah dari lebar lapangan. Hal ini mengindikasikan bahwa sebagian besar tembakan dilepaskan dari area tengah lapangan, yang secara strategis merupakan posisi paling ideal untuk mendapatkan sudut tembak yang lebar dan pandangan yang jelas ke arah gawang. Jumlah tembakan menurun secara signifikan saat bergerak ke arah sisi sayap lapangan (nilai y mendekati 0 atau 80), di mana sudut tembakan menjadi lebih sempit dan sulit.

Analisis juga dilakukan terhadap variabel-variabel kategorikal untuk memahami frekuensi dari setiap kategori. Gambar 4.7 menunjukkan distribusi pada variabel pola permainan.



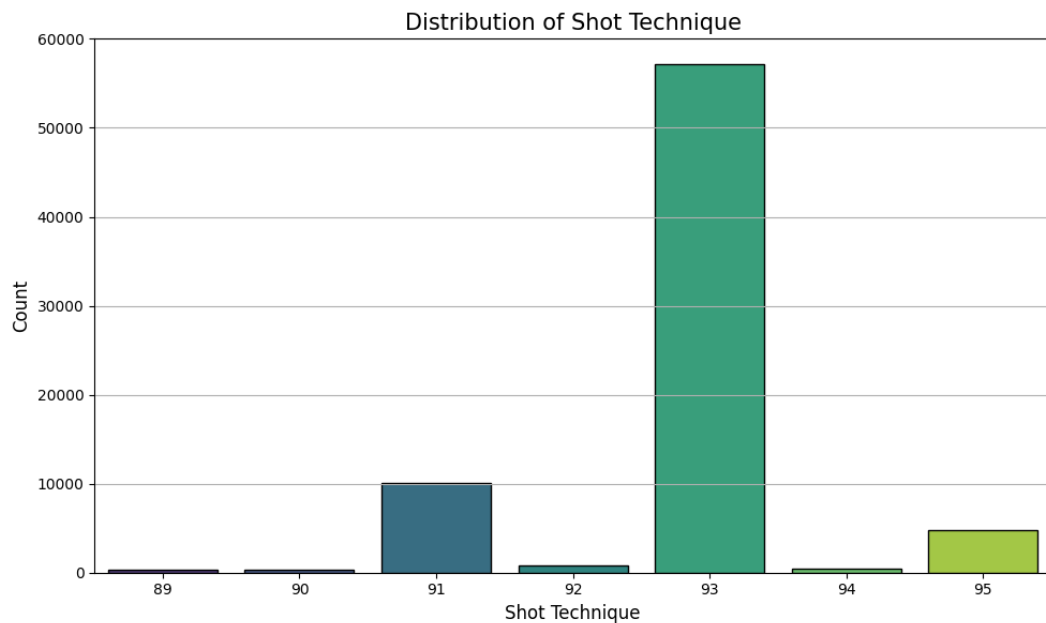
Gambar 4.7 Distribusi Pola Permainan (*Play Pattern*)

Pada Gambar 4.7, distribusi pola permainan menunjukkan bahwa mayoritas tembakan berasal dari situasi permainan reguler atau terbuka (*Regular Play / Open Play*), yang direpresentasikan oleh kategori ID 1. Kategori-kategori lain seperti yang berasal dari tendangan sudut (*From Corner*, ID 3) dan tendangan bebas (*From Free Kick*, ID 4) juga menyumbang jumlah tembakan yang signifikan, namun tidak sebanyak dari permainan terbuka. Pola ini menegaskan bahwa sebagian besar peluang diciptakan melalui alur permainan yang dinamis, bukan dari situasi bola mati. Gambar 4.8 menunjukkan distribusi pada variabel posisi pemain.



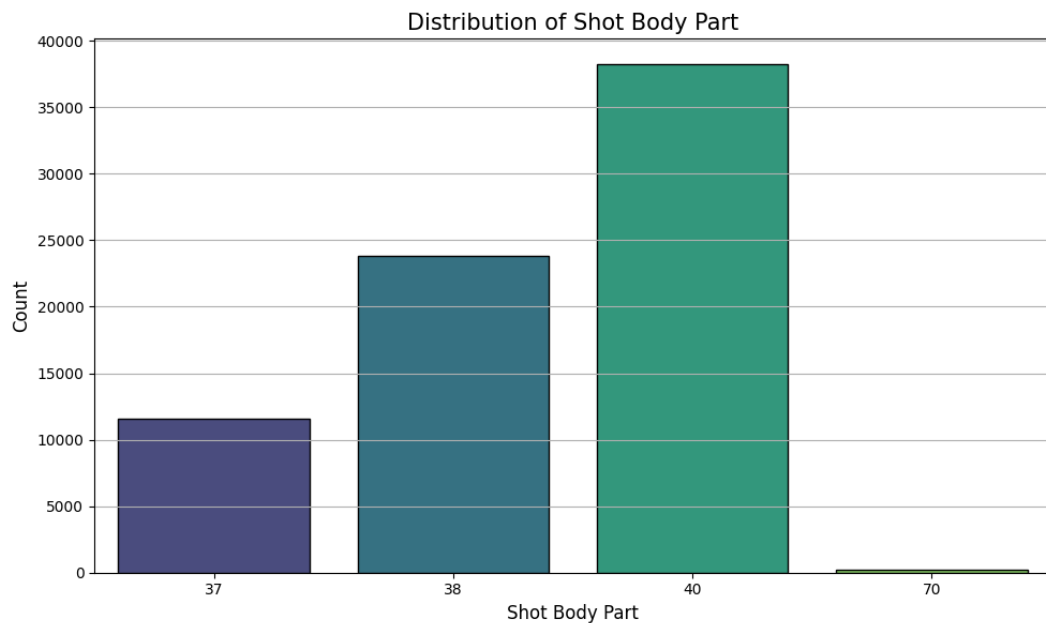
Gambar 4.8 Distribusi Posisi Pemain

Pada Gambar 4.8, distribusi tembakan berdasarkan posisi pemain menunjukkan bahwa pemain dengan peran menyerang secara dominan melepaskan lebih banyak tembakan. Kategori ID 23 dan 24, yang merepresentasikan posisi penyerang tengah (*Center Forward*) dan penyerang sayap (*Winger*), memiliki frekuensi tembakan tertinggi. Diikuti oleh pemain gelandang serang (ID 17, 21, 22). Sebaliknya, pemain dengan posisi bertahan (ID 2, 3, 5, 6) memiliki jumlah tembakan yang jauh lebih sedikit. Hal ini sesuai dengan peran dan tanggung jawab taktis masing-masing posisi di lapangan. Gambar 4.9 menunjukkan distribusi pada variabel teknik tembakan.



Gambar 4.9 Distribusi Teknik Tembakan

Pada Gambar 4.9, terlihat bahwa teknik tembakan yang paling umum digunakan adalah teknik '*Normal*' (ID 93), yang mencakup sebagian besar dari total tembakan. Teknik lain seperti '*Volley*' (ID 91) dan '*Half Volley*' (ID 95) juga digunakan, meskipun dengan frekuensi yang jauh lebih rendah. Teknik-teknik yang lebih sulit dan situasional seperti '*Bicycle Kick*' atau '*Lob*' sangat jarang terjadi. Distribusi ini memberikan gambaran tentang variasi teknis dalam upaya mencetak gol. Gambar 4.10 menunjukkan distribusi pada variabel bagian tubuh yang digunakan untuk menembak.

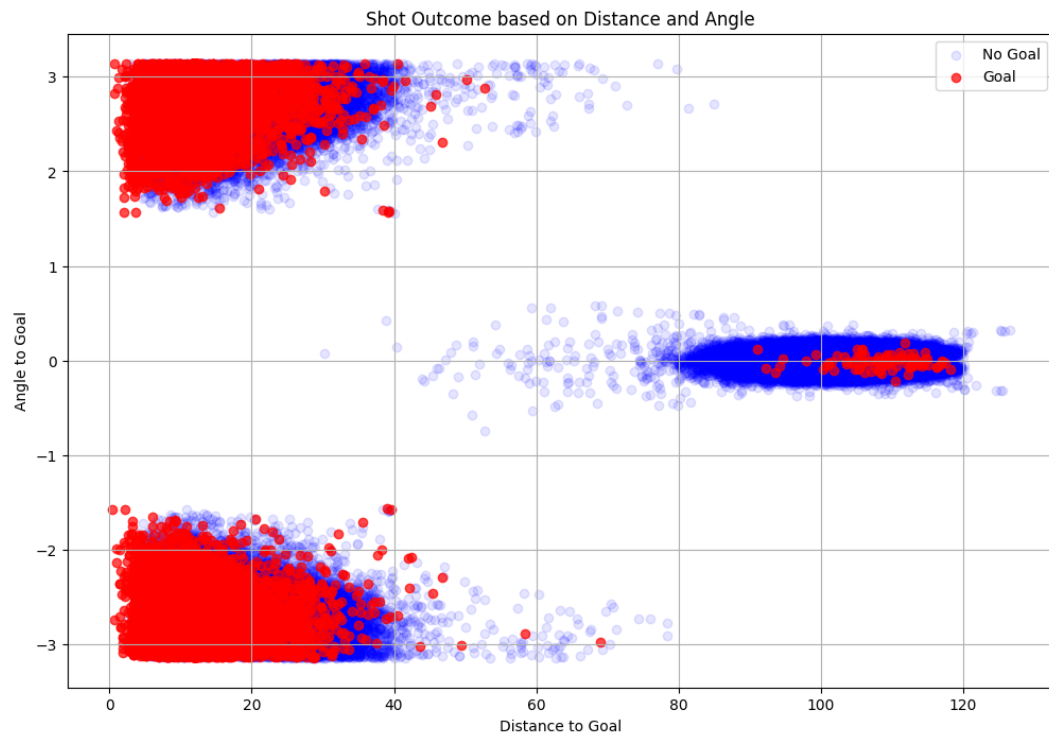


Gambar 4.10 Distribusi Bagian Tubuh yang Digunakan untuk Menembak

Pada Gambar 4.10, distribusi menunjukkan bahwa kaki kanan (ID 40) adalah bagian tubuh yang paling sering digunakan untuk menembak, diikuti oleh kaki kiri (ID 38) dan kepala (ID 37). Ini mencerminkan populasi pemain sepak bola di mana mayoritas adalah pengguna kaki kanan. Penggunaan kepala untuk menembak juga cukup signifikan, terutama dari situasi umpan silang atau bola mati. Kategori 'Lainnya' (*Other*, ID 70) sangat jarang, menunjukkan bahwa gol dengan bagian tubuh yang tidak biasa adalah sebuah anomali.

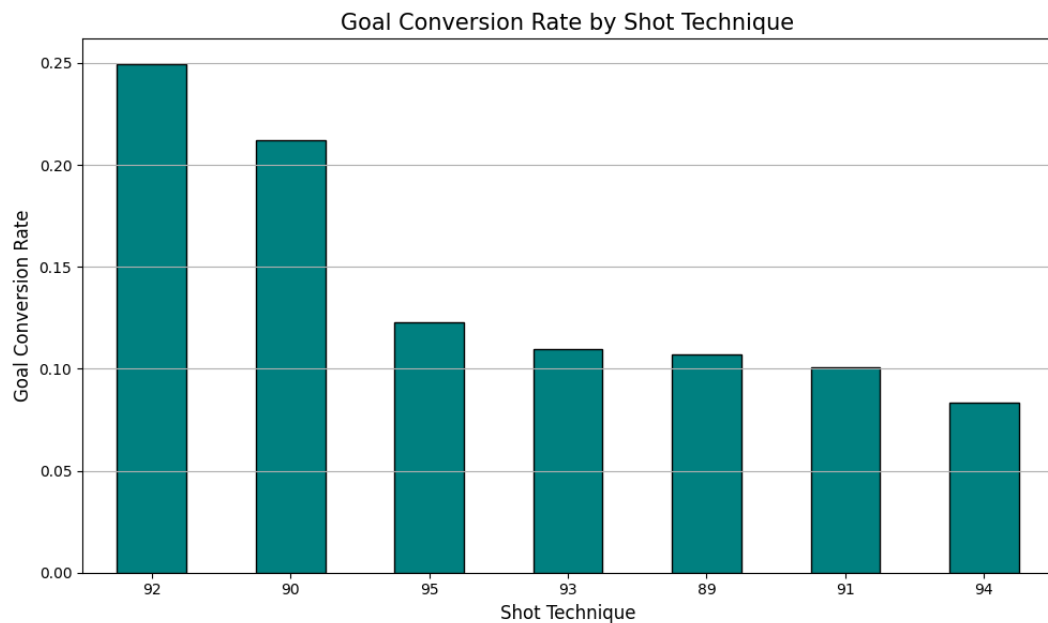
4.2.3.2 Analisis Hubungan Antar Variabel (Bivariat)

Analisis bivariat dilakukan untuk memahami hubungan antara fitur-fitur prediktor dengan hasil tembakan (gol atau tidak gol). Gambar 4.11 menunjukkan distribusi hasil tembakan berdasarkan jarak dan sudut.



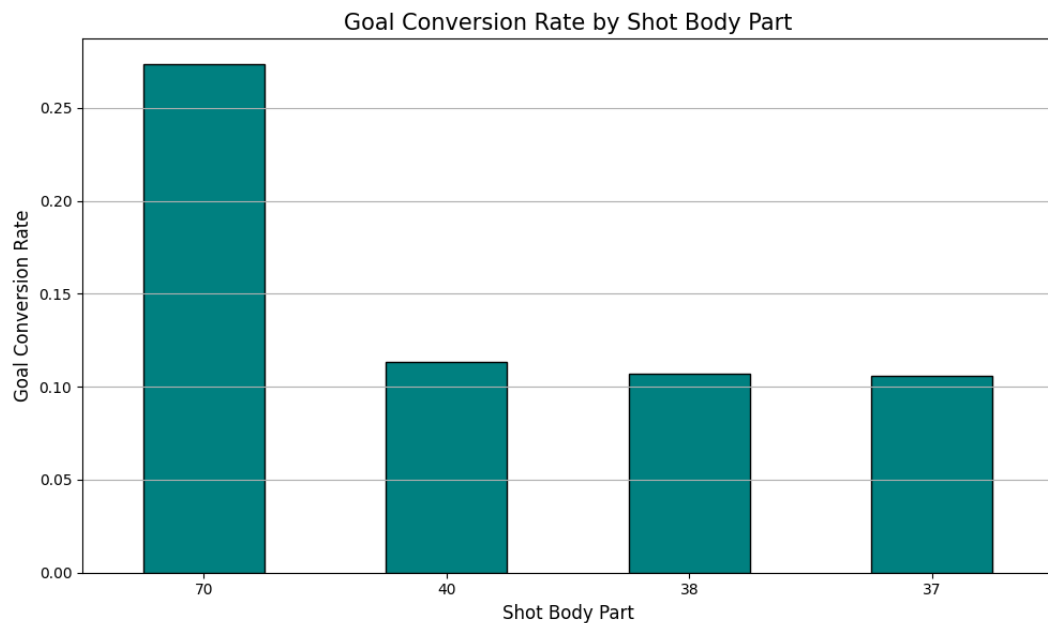
Gambar 4.11 Hasil Tembakan Berdasarkan Jarak dan Sudut

Pada Gambar 4.11, visualisasi ini secara jelas mengilustrasikan hubungan non-linear antara fitur spasial dengan hasil tembakan. Titik-titik merah yang merepresentasikan gol terkonsentrasi secara padat pada area dengan jarak ke gawang yang rendah (sumbu x kecil) dan sudut tembakan yang lebar (sumbu y mendekati nol). Sebaliknya, titik-titik biru yang merepresentasikan tembakan yang gagal mencetak gol tersebar di seluruh area, namun menjadi sangat dominan pada jarak yang jauh dan sudut yang sempit. Gambar 4.12 menunjukkan distribusi tingkat konversi gol berdasarkan teknik tembakan.



Gambar 4.12 Tingkat Konversi Gol Berdasarkan Teknik Tembakan

Pada Gambar 4.13, terlihat bahwa teknik tembakan tertentu memiliki tingkat keberhasilan yang lebih tinggi. Teknik dengan ID 92 dan 90 memiliki tingkat konversi tertinggi, masing-masing sekitar 25% dan 21%. Teknik ini adalah *lob* atau *diving header* yang meskipun sulit dilakukan, biasanya terjadi sangat dekat dengan gawang sehingga peluang golnya tinggi. Teknik '*Normal*' (ID 93), meskipun paling umum, memiliki tingkat konversi yang lebih rendah (sekitar 11%). Hal ini menunjukkan adanya *trade-off* antara frekuensi penggunaan teknik dengan efektivitasnya. Gambar 4.13 menunjukkan distribusi tingkat konversi gol berdasarkan bagian tubuh.



Gambar 4.13 Tingkat Konversi Gol Berdasarkan Bagian Tubuh

Pada Gambar 4.14, analisis tingkat konversi gol berdasarkan bagian tubuh menunjukkan hasil yang menarik. Kategori 'Lainnya' (Other, ID 70) memiliki tingkat konversi tertinggi, mencapai lebih dari 27%. Meskipun sangat jarang, tembakan dengan bagian tubuh yang tidak biasa (misalnya *knee* atau *chest*) cenderung terjadi dalam situasi kemelut di depan gawang pada jarak yang sangat dekat, sehingga probabilitas golnya menjadi sangat tinggi. Sementara itu, kaki kanan (ID 40), kaki kiri (ID 38), dan kepala (ID 37) memiliki tingkat konversi yang relatif serupa, yaitu sekitar 11%. Ini mengindikasikan bahwa pada dasarnya, efektivitas tembakan tidak terlalu bergantung pada bagian tubuh yang umum digunakan, melainkan lebih pada kualitas situasi saat tembakan itu terjadi.

4.2.3.3 Kesimpulan Hasil EDA

Dari serangkaian analisis univariat dan bivariat yang telah dilakukan pada tahap EDA, dapat ditarik beberapa kesimpulan kunci yang menjadi landasan untuk proses pemodelan selanjutnya. Temuan-temuan ini dirangkum dalam Tabel 4.3.

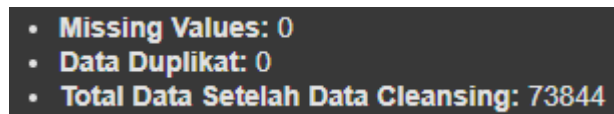
Tabel 4.3 Ringkasan Temuan EDA

Temuan Utama	Analisis	Implikasi untuk Penelitian
Pentingnya Konteks Spasial & Hubungan Non-Linear	Lokasi tembakan terkonsentrasi di dekat gawang. Hubungan antara jarak, sudut, dan hasil tembakan terbukti non-linear.	Memvalidasi kebutuhan penggunaan model non-linear dan menegaskan pentingnya fitur rekayasa spasial.
Perbedaan antara Frekuensi dan Efektivitas	Situasi yang sering terjadi tidak selalu paling efektif. Situasi langka memiliki tingkat konversi gol tertinggi.	Menekankan perlunya fitur-fitur kontekstual untuk menangkap nuansa kualitas peluang yang tidak dapat diwakili oleh frekuensi saja.
Dominasi Peran Taktis dalam Menciptakan Peluang	Pemain dengan peran menyerang secara signifikan lebih sering melepaskan tembakan dibandingkan pemain bertahan.	Mengonfirmasi bahwa fitur <i>position</i> adalah prediktor yang relevan dan harus disertakan dalam model untuk merefleksikan peran taktis pemain.
Validasi Kualitas Data dan Keselarasan Pola	Pola yang teridentifikasi (misal, peningkatan intensitas tembakan di akhir babak, dominasi penggunaan kaki kanan) selaras dengan pengetahuan umum sepak bola.	Memberikan keyakinan bahwa <i>dataset</i> yang digunakan berkualitas, konsisten, dan cukup representatif untuk membangun model prediksi xG yang andal dan valid.

4.2.4 Data Cleansing

Tahap data *cleansing* dilakukan untuk memeriksa kelengkapan dan keunikan data dengan tujuan memastikan bahwa tidak terdapat nilai kosong (*missing values*) maupun data duplikat yang dapat memengaruhi proses analisis. Pada penelitian ini, proses pembersihan data menunjukkan bahwa data yang digunakan telah bersih secara struktural. Hal ini disebabkan oleh karakteristik data sepak bola yang cenderung unik di mana setiap peristiwa dalam pertandingan memiliki identitas dan konteks yang berbeda serta karena data yang disediakan oleh StatsBomb telah tersusun secara rapi dan konsisten. Struktur data yang baik ini

sangat membantu dalam mempercepat proses *preprocessing* dan meningkatkan kualitas hasil analisis, karena tidak memerlukan upaya koreksi data secara signifikan. Hasil akhir dari proses *data cleansing* ini ditunjukkan pada Gambar 4.14.



```
• Missing Values: 0
• Data Duplikat: 0
• Total Data Setelah Data Cleansing: 73844
```

Gambar 4.14 Hasil *Data Cleansing*

4.3 *Data Transformation*

Pada tahap ini, akan dijelaskan serangkaian proses persiapan data yang dilakukan setelah tahap *preprocessing* awal. Proses ini bertujuan untuk memastikan *dataset* yang digunakan relevan dan memiliki format yang optimal untuk dilatih menggunakan algoritma LightGBM. Tahapan yang dilakukan meliputi rekayasa fitur untuk menciptakan variabel prediktif baru, seleksi dan deskripsi fitur akhir yang digunakan, dan diakhiri dengan pembagian *dataset* menjadi data latih dan data uji.

Tahapan *feature scaling* numerik tidak dilakukan dalam penelitian ini. Keputusan ini didasarkan pada karakteristik algoritma LightGBM yang merupakan model berbasis pohon (*tree-based*). Model jenis ini tidak sensitif terhadap perbedaan skala pada fitur-fitur numerik, sehingga ketiadaan proses *scaling* tidak akan memengaruhi performa model (Ke *et al.*, 2017).

4.3.1 Feature Engineering

Dalam pemodelan xG, proses *feature engineering* mencakup dua langkah utama, yaitu mengubah data koordinat menjadi metrik spasial yang bermakna dan menambahkan fitur kontekstual berdasarkan urutan peristiwa.

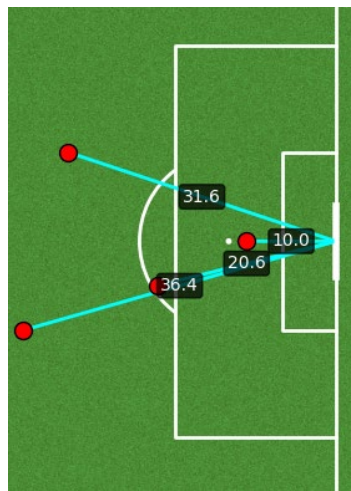
a. Fitur Spasial

Mengubah data koordinat ($start_x$, $start_y$) menjadi metrik jarak dan sudut tembakan adalah langkah yang paling fundamental dan terbukti efektif dalam meningkatkan daya prediksi model. Dalam *dataset* ini, koordinat gawang yang digunakan sebagai referensi adalah: pusat gawang berada pada titik ($x = 120,0$, $y = 40,0$), tiang bawah pada ($y = 36,34$), dan tiang atas pada ($y = 43,66$).

- i. Jarak ke Gawang ($distance_to_goal$): Fitur ini dihitung menggunakan rumus jarak *Euclidean* antara titik tembakan (x_{start} , y_{start}) dan pusat gawang (x_{goal} , y_{goal}) seperti yang ditunjukkan pada Persamaan 4.1 (Bandara *et al.* 2024).

$$Distance = \sqrt{(x_{goal} - x_{start})^2 + (y_{goal} - y_{start})^2} \quad (4.1)$$

Ilustrasi geometris dari perhitungan fitur ini dapat dilihat pada Gambar 4.15.

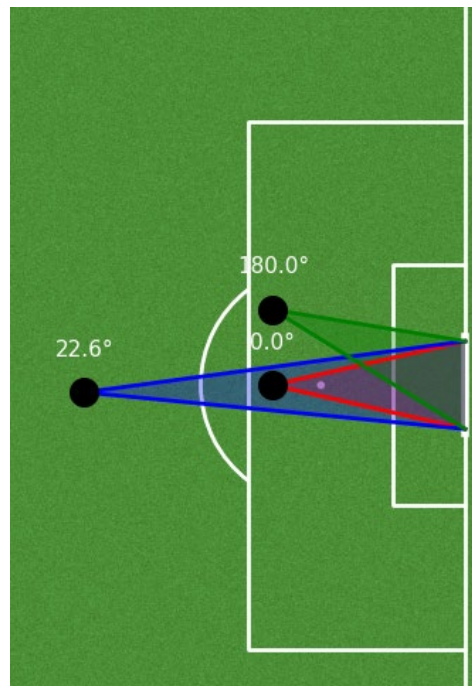


Gambar 4.15 Visualisasi *distance_to_goal*

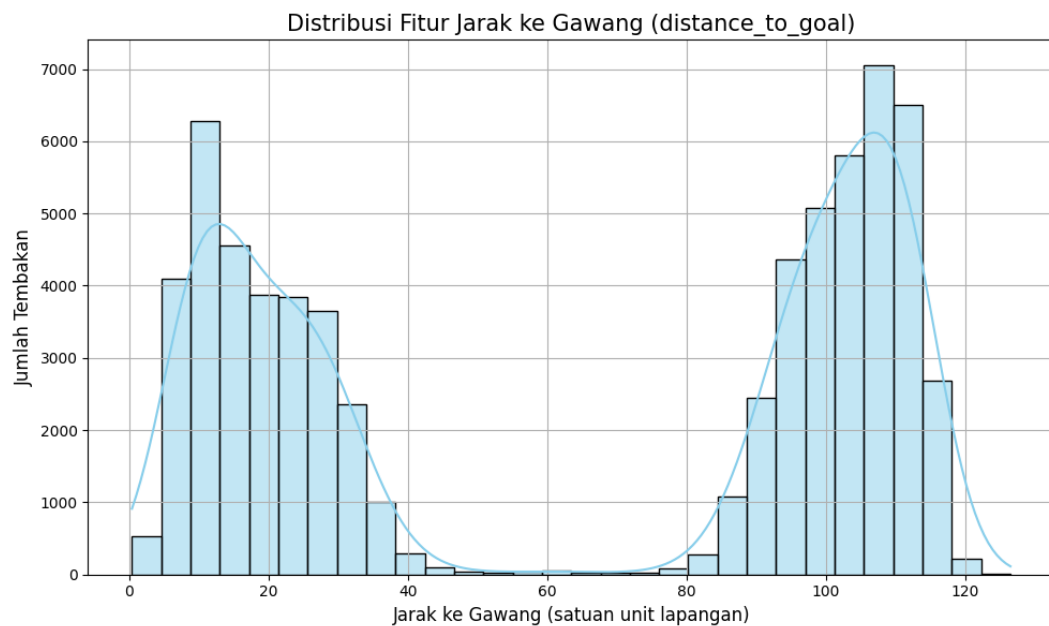
- ii. Sudut Tembakan (*angle_to_goal*): Fitur ini merepresentasikan besar sudut pandang penendang terhadap gawang, dihitung menggunakan Hukum Kosinus pada segitiga yang dibentuk oleh titik tembakan dan kedua tiang gawang, sesuai dengan Persamaan 4.2 (Bandara *et al.* 2024).

$$\theta = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \quad (4.2)$$

Dimana a dan b adalah jarak dari titik tembakan ke masing-masing tiang gawang, dan c adalah lebar gawang. Ilustrasi geometris dari perhitungan fitur ini dapat dilihat pada Gambar 4.16.

Gambar 4.16 Visualisasi *angle_to_goal*

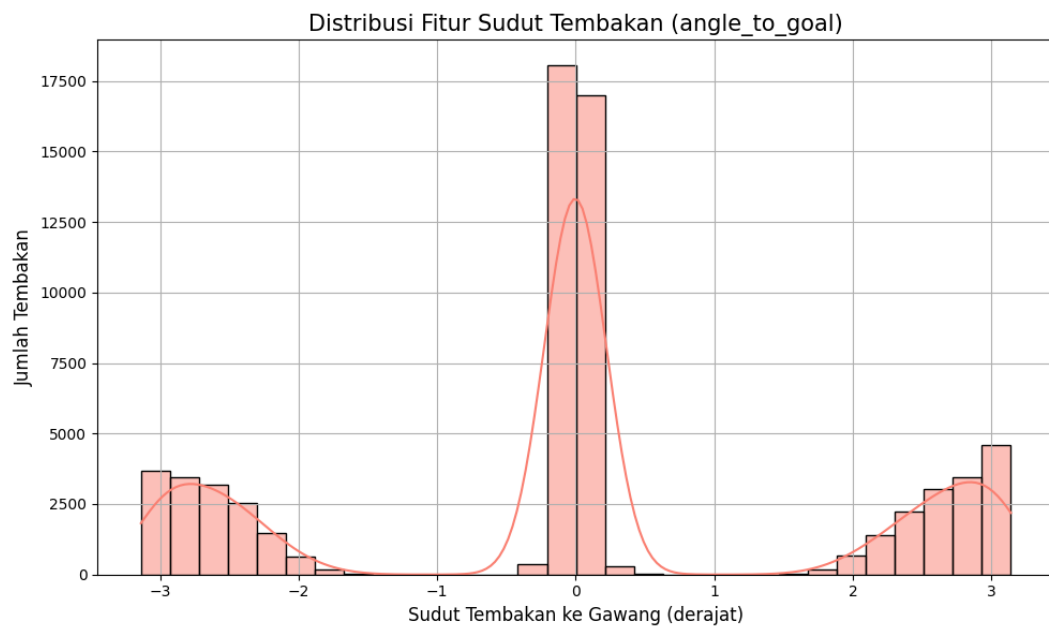
Setelah fitur *distance_to_goal* dan *angle_to_goal* dihitung, distribusinya divisualisasikan untuk memahami karakteristiknya. Gambar 4.17 menunjukkan distribusi jarak tembakan.



Gambar 4.17 Visualisasi Distribusi Fitur *distance_to_goal*

Gambar 4.17 menyajikan distribusi frekuensi dari fitur *distance_to_goal*. Distribusi ini menunjukkan pola bimodal yang jelas, dengan dua puncak (modus) utama. Puncak pertama terletak pada rentang jarak yang relatif dekat (sekitar 5 hingga 25 unit), merepresentasikan mayoritas tembakan yang dilakukan dari dalam atau sekitar area penalti. Puncak kedua yang signifikan berada pada rentang jarak yang sangat jauh (sekitar 95 hingga 115 unit). Puncak ini secara teknis merepresentasikan tembakan-tembakan yang terjadi di paruh lapangan yang berlawanan dengan gawang referensi, sehingga menghasilkan

nilai jarak yang besar. Pola bimodal ini mengilustrasikan bahwa frekuensi tembakan terkonsentrasi pada dua zona utama di lapangan. Sementara Gambar 4.18 menunjukkan visualisasi distribusi sudut tembakan.



Gambar 4.18 Visualisasi Distribusi Fitur *angle_to_goal*

Gambar 4.18 menyajikan distribusi fitur *angle_to_goal*. Distribusi ini menunjukkan pola trimodal, dengan puncak tertinggi yang sangat dominan berada di sekitar nilai sudut 0.0 derajat. Hal ini mengindikasikan bahwa frekuensi tertinggi dari tembakan terjadi pada posisi yang relatif lurus di depan gawang. Dua puncak sekunder yang lebih kecil terlihat di kedua sisi ekor distribusi, mendekati nilai -3.0 dan +3.0 derajat. Puncak-puncak ini merepresentasikan tembakan-tembakan yang dilakukan dari sudut yang sangat sempit atau dari posisi yang sangat melebar. Pola distribusi ini memberikan

informasi penting bagi model mengenai variasi sudut tembakan yang umum terjadi dalam pertandingan.

b. Fitur Kontekstual (*type_before*)

Fitur *type_before* ditambahkan sebagai bagian dari proses rekayasa fitur untuk memberikan konteks temporal terhadap peristiwa tembakan. Fitur ini merepresentasikan jenis peristiwa yang terjadi tepat sebelum tembakan dilakukan, dengan mengambil nilai *type.id* dari peristiwa sebelumnya dalam urutan kronologis pertandingan. Informasi ini bertujuan untuk menangkap dinamika permainan yang mendahului tembakan, seperti apakah tembakan tersebut terjadi setelah dribel, operan, atau intersepsi. Dengan menambahkan konteks ini, model dapat memahami alur permainan yang berujung pada tembakan. Tabel 4.4 menyajikan deskripsi lengkap untuk setiap *type.id* yang digunakan dalam data StatsBomb.

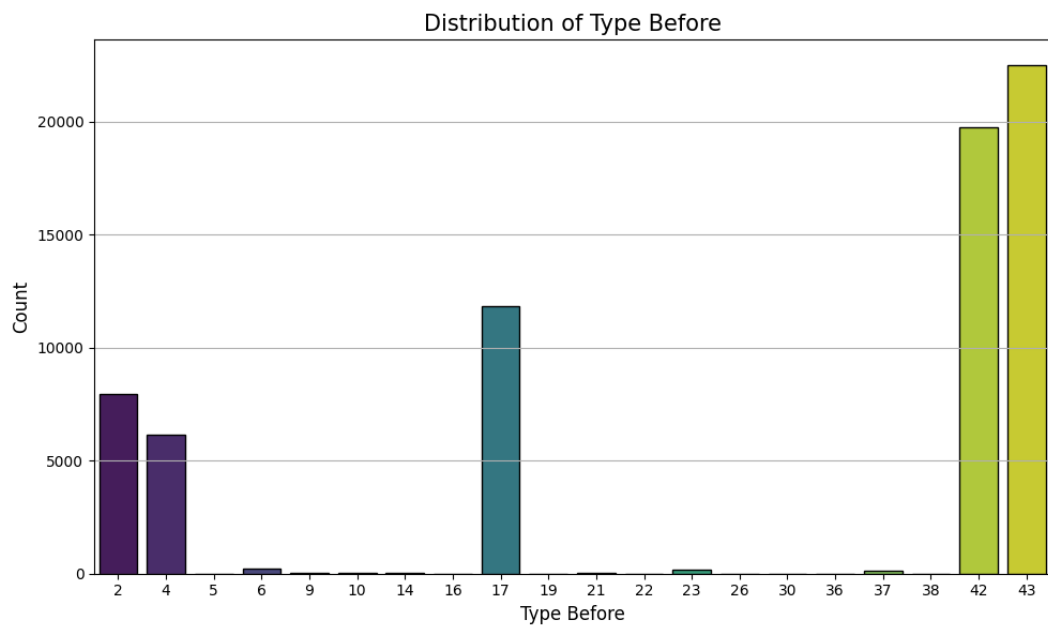
Tabel 4.4 Deskripsi Jenis *type* dalam *Dataset* StatsBomb.

<i>Event Type</i>	<i>Type ID</i>	Deskripsi Singkat
<i>50/50</i>	33	Dua pemain dari tim berbeda berebut bola lepas.
<i>Bad Behaviour</i>	24	Pelanggaran di luar permainan.
<i>Ball Receipt*</i>	42	Momen penerimaan atau usaha menerima operan.
<i>Ball Recovery</i>	2	Usaha merebut kembali bola lepas.
<i>Block</i>	6	Pemain menghalangi bola dengan tubuhnya.
<i>Carry</i>	43	Pemain menguasai bola saat bergerak atau diam.
<i>Clearance</i>	9	Menghalau bola dari area bahaya tanpa niat mengoper ke rekan.
<i>Dispossessed</i>	3	Pemain kehilangan bola karena ditekel tanpa mencoba dribel.
<i>Dribble</i>	14	Usaha pemain melewati lawan dengan menggiring bola.
<i>Dribbled Past</i>	39	Pemain dilewati oleh lawan saat dribel.

<i>Duel</i>	4	Duel 1v1 antara pemain dari tim berbeda.
<i>Error</i>	37	Kesalahan pemain.
<i>Foul Committed</i>	22	Pelanggaran yang dilakukan terhadap lawan (tidak termasuk <i>offside</i>).
<i>Foul Won</i>	21	Pelanggaran yang diterima dan menghasilkan tendangan bebas atau penalti.
<i>Goal Keeper</i>	23	Segala aksi penjaga gawang (penyelamatan, <i>smother</i> , <i>punch</i> , dll).
<i>Half End</i>	34	Peluit akhir babak pertandingan oleh wasit.
<i>Half Start</i>	18	Peluit awal babak pertandingan oleh wasit.
<i>Injury Stoppage</i>	40	Penghentian permainan karena cedera.
<i>Interception</i>	10	Pemain memotong jalur operan lawan untuk mencegah bola sampai ke target.
<i>Miscontrol</i>	38	Kehilangan kontrol bola karena sentuhan yang buruk.
<i>Offside</i>	8	Pelanggaran posisi <i>offside</i> .
<i>Own Goal Against</i>	20	Gol bunuh diri oleh tim sendiri.
<i>Own Goal For</i>	25	Gol bunuh diri yang menguntungkan tim.
<i>Pass</i>	30	Umpan dari satu pemain ke pemain lain.
<i>Player Off</i>	27	Pemain keluar lapangan tanpa pergantian (misalnya karena cedera).
<i>Player On</i>	26	Pemain kembali masuk ke lapangan setelah <i>Player Off</i> .
<i>Pressure</i>	17	Aksi menekan pemain lawan di area tertentu, direkam bersama durasi tekanan.
<i>Referee Ball-Drop</i>	41	Wasit menjatuhkan bola untuk melanjutkan pertandingan setelah jeda (misalnya cedera).
<i>Shield</i>	28	Pemain melindungi bola agar keluar lapangan tanpa dikejar lawan.

<i>Shot</i>	16	Upaya mencetak gol dengan bagian tubuh legal.
<i>Starting XI</i>	35	Informasi awal pemain yang bermain dan formasi tim.
<i>Substitution</i>	19	Pergantian pemain saat pertandingan berlangsung.
<i>Tactical Shift</i>	36	Perubahan posisi pemain atau formasi taktik dalam pertandingan.

Gambar 4.19 menampilkan distribusi jenis peristiwa yang terjadi tepat sebelum sebuah tembakan dilepaskan.



Gambar 4.19 Distribusi Jenis Peristiwa Sebelum Tembakan (*Type Before*)

Gambar 4.19 menampilkan bahwa dua peristiwa paling dominan yang mendahului tembakan adalah '*Carry*' (ID 43) dan '*Ball Receipt**' (ID 42), yang menunjukkan bahwa sebagian besar tembakan terjadi setelah pemain membawa bola atau menerima operan. Peristiwa '*Pressure*' (ID 17) juga memiliki frekuensi yang cukup tinggi, mengindikasikan bahwa banyak tembakan dilepaskan saat pemain berada di bawah tekanan lawan. Peristiwa defensif seperti '*Ball Recovery*'

(ID 2) dan 'Duel' (ID 4) juga muncul, menandakan tembakan yang berasal dari situasi perebutan bola. Distribusi ini memberikan konteks penting tentang dinamika permainan yang mengarah pada sebuah peluang tembakan.

4.3.2 Feature Selection

Setelah proses rekayasa fitur, langkah selanjutnya adalah menetapkan himpunan fitur final yang akan digunakan untuk melatih model. Dalam penelitian ini, semua fitur yang tersedia, baik fitur asli maupun hasil rekayasa, dipertahankan. Keputusan ini diambil dengan pertimbangan bahwa setiap fitur memberikan konteks unik dan berharga mengenai keadaan pertandingan saat tembakan terjadi. Dengan memberikan informasi selengkap mungkin, diharapkan model dapat mempelajari pola-pola yang lebih kompleks dan menghasilkan prediksi yang lebih akurat. Tabel 4.5 merinci himpunan fitur akhir yang dipilih untuk digunakan dalam pemodelan xG.

Tabel 4.5 Himpunan Fitur Akhir untuk Pemodelan xG

Kategori Fitur	Nama Fitur
Dimensi Waktu	<i>minute, second</i>
Dimensi Spasial	<i>start_x, start_y</i>
Konteks Tembakan	<i>shot_body_part, shot_technique, shot_type, play_pattern</i>
Konteks Situasi	<i>shot_first_time, shot_open_goal, shot_one_on_one, shot_aerial_won, shot_key_pass</i>
Fitur Hasil Rekayasa	<i>distance_to_goal, angle_to_goal, type_before</i>
Variabel Target	<i>shot_outcome</i>

4.3.3 *Splitting Dataset*

Tahap akhir dari persiapan data adalah partisi *dataset* menjadi dua himpunan yang independen: himpunan data latih (*training set*) dan himpunan data uji (*test set*). Pemilihan proporsi pembagian 90% untuk data latih dan 10% untuk data uji merupakan sebuah keputusan metodologis yang strategis. Alokasi sebesar 90% (66.459 sampel) bertujuan untuk memaksimalkan volume data yang dapat dipelajari oleh algoritma LightGBM. Mengingat sifat data sepak bola yang memiliki interaksi fitur non-linear yang sangat kompleks, ketersediaan data latih yang besar menjadi fundamental. Model yang gagal menangkap struktur fundamental pada data non-linear berisiko mengalami *underfitting*, di mana performa prediktifnya akan sangat buruk baik pada data latih maupun data uji (Shwartz-Ziv & Armon, 2022). Oleh karena itu, volume data latih yang besar esensial untuk memastikan model dapat mempelajari pola-pola rumit secara efektif.

Di sisi lain, meskipun hanya 10%, himpunan data uji yang tersisa mencakup 7.385 sampel tembakan. Jumlah ini secara statistik sudah sangat representatif dan lebih dari cukup untuk memberikan evaluasi kinerja generalisasi model yang andal, objektif, dan tidak bias terhadap data baru. Proses pembagian ini juga menggunakan parameter *random_state* untuk memastikan bahwa hasilnya dapat direproduksi secara konsisten. Rincian jumlah data pada masing-masing himpunan disajikan pada Tabel 4.6.

Tabel 4.6 Rincian Dimensi *Dataset* Akhir

<i>Dataset</i>	Jumlah Baris (Sampel Tembakan)
Data Latih	66,459
Data Uji	7,385
Total	73,844

4.4 Data Mining

Tahap *data mining* dalam penelitian ini bertujuan untuk membangun sebuah model prediktif. Prosesnya meliputi pembangunan model klasifikasi probabilistik menggunakan algoritma LightGBM, optimasi untuk menemukan konfigurasi terbaik, dan diakhiri dengan kalibrasi untuk menyempurnakan hasil.

4.4.1 Tuning Process Configuration

Proses pemodelan diawali dengan inisialisasi *LGBMClassifier*. Untuk mendapatkan performa yang optimal, dilakukan proses *tuning* terhadap sejumlah *hyperparameter*. Ruang pencarian (*search space*) yang digunakan dalam penelitian ini secara rinci disajikan pada Tabel 4.7.

Tabel 4.7 Ruang Pencarian *Hyperparameter*

Nama <i>Hyperparameter</i>	Nilai yang Diuji
<i>min_child_samples</i>	Distribusi integer acak dari 0-200
<i>num_leaves</i>	Distribusi integer acak dari 2-500
<i>reg_lambda</i>	Distribusi uniform acak dari 0-1
<i>reg_alpha</i>	Distribusi uniform acak dari 0-1
<i>max_depth</i>	Distribusi integer acak dari 0-500

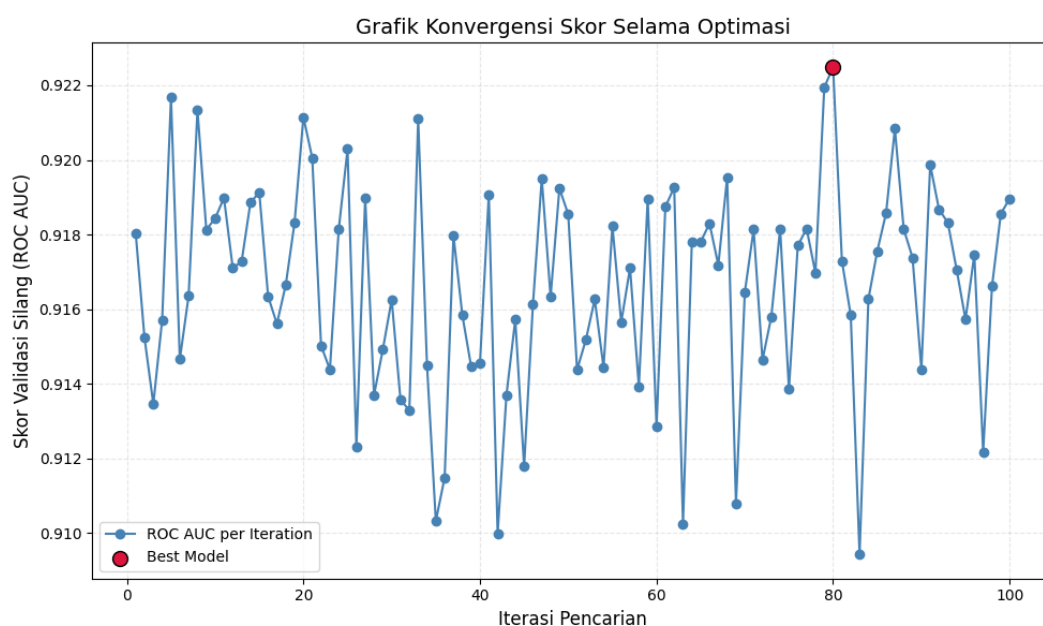
Metrik utama yang digunakan sebagai acuan skor (*scoring*) untuk proses optimasi adalah ROC AUC. Metrik ini dipilih karena kemampuannya dalam mengevaluasi performa model pada *dataset* yang tidak seimbang (*imbalanced*), seperti kasus klasifikasi probabilistik gol dalam penelitian ini.

4.4.2 Hyperparameter Search

Pencarian kombinasi *hyperparameter* terbaik dari ruang pencarian pada Tabel 4.7 dilakukan menggunakan metode *RandomizedSearchCV* dari *scikit-learn*. Proses pencarian dijalankan dengan 100 iterasi acak ($n_iter=100$) dan

menggunakan skema *cross validation* 5-lipat ($cv=5$). Selain itu, ROC AUC juga akan digunakan pada tahap evaluasi untuk memberikan analisis pembandingan kinerja model mana yang lebih komprehensif.

Proses pencarian ini melalui banyak iterasi untuk menemukan performa terbaik. Gambar 4.20 menunjukkan fluktuasi skor ROC AUC pada setiap iterasi. Dari grafik tersebut, dapat diamati bahwa proses pencarian telah menjelajahi berbagai tingkat performa dan berhasil menemukan titik optimalnya, yang ditandai sebagai skor terbaik.



Gambar 4.20 Grafik Konvergensi Skor Selama Optimasi

Hasil dari 5 kombinasi *hyperparameter* teratas yang ditemukan selama proses pencarian disajikan pada Tabel 4.8.

Tabel 4.8 Sampel Hasil Iterasi Pencarian (5 kombinasi teratas)

Peringkat	Skor ROC AUC	Konfigurasi <i>Hyperparameter</i>
1	0.922487	<i>max_depth</i> : 3 <i>min_child_samples</i> : 38

		<i>num_leaves</i> : 247 <i>reg_alpha</i> : 0.00374 <i>reg_lambda</i> : 0.08957
2	0.921943	<i>max_depth</i> : 214 <i>min_child_samples</i> : 29 <i>num_leaves</i> : 9 <i>reg_alpha</i> : 0.39692 <i>reg_lambda</i> : 0.69268
3	0.921676	<i>max_depth</i> : 277 <i>min_child_samples</i> : 140 <i>num_leaves</i> : 16 <i>reg_alpha</i> : 0.46764 <i>reg_lambda</i> : 0.344755
4	0.921338	<i>max_depth</i> : 415 <i>min_child_samples</i> : 118 <i>num_leaves</i> : 4 <i>reg_alpha</i> : 0.896714 <i>reg_lambda</i> : 0.16111
5	0.921151	<i>max_depth</i> : 241 <i>min_child_samples</i> : 151 <i>num_leaves</i> : 24 <i>reg_alpha</i> : 0.80678 <i>reg_lambda</i> : 0.491033

Berdasarkan hasil yang disajikan pada Tabel 4.8, model dengan konfigurasi pada Peringkat 1 ditetapkan sebagai model terbaik. Pemilihan ini didasarkan pada pencapaian skor ROC AUC rata-rata tertinggi selama proses validasi silang, yaitu sebesar 0.922487. Konfigurasi *hyperparameter* dari model inilah yang akan digunakan untuk tahap selanjutnya.

4.4.3 Model Probability

Model *LGBMClassifier* yang telah dioptimalkan pada dasarnya adalah sebuah alat klasifikasi. Namun, untuk tujuan penelitian ini, fokusnya bukanlah pada hasil klasifikasi akhir (prediksi biner gol atau tidak gol), melainkan pada estimasi probabilitas yang mendasari prediksi tersebut.

Dalam ekosistem scikit-learn, model klasifikasi yang telah dilatih (fit) memiliki dua metode utama untuk prediksi:

- a. *predict()* : Metode ini memberikan hasil prediksi kelas akhir. Ia bekerja dengan cara menghitung probabilitas internal, lalu menerapkan ambang batas (secara *default* 0.5) untuk memutuskan hasilnya. Sebagai contoh, jika probabilitas sebuah tembakan menjadi gol adalah 0.51, *predict()* akan menghasilkan 1 (gol). Sebaliknya, jika probabilitasnya 0.49, ia akan menghasilkan 0 (tidak gol). Metode ini tidak digunakan karena menghilangkan informasi granular mengenai seberapa besar peluang sebuah tembakan.
- b. *predict_proba()* : Metode inilah yang menjadi kunci dalam penelitian ini. Alih-alih memberikan hasil akhir, *predict_proba()* mengembalikan sebuah *array* yang berisi estimasi probabilitas untuk setiap kelas. Untuk masalah biner seperti ini, *output*-nya akan memiliki format (*n_sampel*, 2), di mana kolom pertama berisi probabilitas untuk kelas negatif (kelas 0, atau "tidak gol") kemudian, kolom kedua berisi probabilitas untuk kelas positif (kelas 1, atau "gol"). Nilai pada kolom kedua inilah yang diekstrak dan digunakan sebagai nilai xG untuk setiap tembakan.

Untuk memberikan gambaran yang lebih jelas, Gambar 4.21 menampilkan contoh di mana nilai probabilitas dari *predict_proba()* telah diekstrak menjadi kolom xG dan disandingkan dengan hasil tembakan yang sebenarnya (*shot_outcome*). Terlihat pada baris ke-3, sebuah tembakan dengan nilai xG yang relatif tinggi (0.452387) memang sesuai dengan hasil akhirnya yaitu gol (*shot_outcome* = 1).

	xG	shot_outcome
0	0.077743	0
1	0.095051	0
2	0.008634	0
3	0.452387	1
4	0.007224	0
5	0.003641	0

Gambar 4.21 Contoh Nilai Probabilitas

Meskipun skor ROC AUC yang tinggi membuktikan bahwa model andal dalam memeringkat tembakan (kemampuannya untuk memberikan skor probabilitas lebih tinggi pada tembakan yang memang gol), nilai mentah dari *predict_proba()* ini belum tentu terkalibrasi dengan baik. Hal ini menjadi dasar mengapa tahap kalibrasi probabilitas menjadi sangat penting.

4.4.4 *Model Probability Calibration*

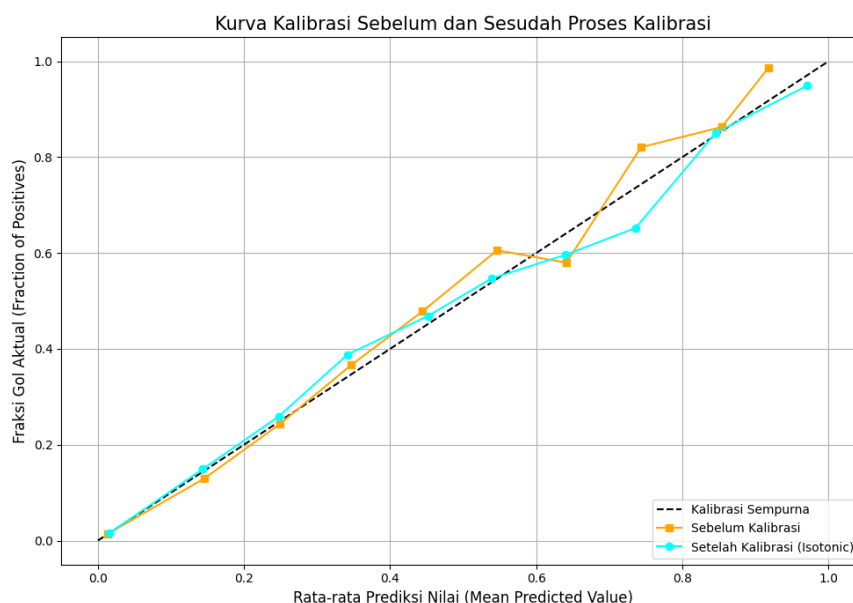
Model terbaik yang telah ditetapkan selanjutnya melalui tahap kalibrasi probabilitas. Proses ini bertujuan untuk meningkatkan keandalan nilai probabilitas (nilai xG) yang dihasilkan oleh model agar lebih akurat dan tidak bias. Kalibrasi dilakukan menggunakan metode *CalibratedClassifierCV* dari *library scikit-learn* dengan teknik *isotonic regression*.

Proses ini dimulai dengan mengambil model LightGBM dengan performa terbaik hasil dari pencarian *hyperparameter*. *CalibratedClassifierCV* kemudian membungkus model ini dan menerapkan prosedur kalibrasi menggunakan validasi silang 3-lipat (*cv=3*) pada data latih. Dalam setiap lipatan, model dasar LightGBM

dilatih pada sebagian data, lalu digunakan untuk memprediksi probabilitas pada sisa data (data validasi). Probabilitas mentah ini kemudian dibandingkan dengan hasil sebenarnya (gol atau tidak gol) untuk melatih sebuah model kalibrasi terpisah.

Setelah proses validasi silang selesai, model dasar LightGBM dilatih kembali menggunakan seluruh data latih, dan fungsi kalibrasi isotonik yang telah dipelajari diterapkan padanya. Hasil akhir dari proses ini adalah sebuah model baru yang telah terkalibrasi dan siap untuk digunakan.

Keberhasilan proses kalibrasi ini dibuktikan secara visual melalui Gambar 4.22 menampilkan perbandingan kurva sebelum (garis oranye) dan sesudah (garis cyan) proses kalibrasi. Dapat dianalisis bahwa kurva model yang telah terkalibrasi posisinya jauh lebih mendekati garis diagonal "kalibrasi sempurna" (garis putus-putus hitam). Hal ini menandakan bahwa *output* probabilitas (nilai xG) dari model kini lebih andal dan sesuai dengan proporsi gol yang sebenarnya terjadi di lapangan.



Gambar 4.22 Kurva Kalibrasi Sebelum dan Sesudah Proses Kalibrasi

Dengan terbuktinya keberhasilan proses kalibrasi melalui kurva visual, model *LGBMClassifier* kini tidak hanya andal dalam memeringkat, tetapi juga dalam menghasilkan nilai probabilitas (xG) yang akurat dan dapat dipercaya. Sebagai langkah finalisasi, keseluruhan objek model yang telah terlatih dan terkalibrasi ini kemudian diserialisasi untuk disimpan ke dalam sebuah *file*.

Proses penyimpanan ini dilakukan menggunakan *library joblib*, yang sangat efisien untuk objek Python yang berisi *array* NumPy besar seperti model *scikit-learn*. Model tersebut disimpan sebagai *calibrated_model.joblib*, yang nantinya siap untuk dimuat kembali dan digunakan untuk melakukan prediksi pada data baru tanpa perlu melalui proses pelatihan ulang.

4.5 *Evaluation*

Tahap evaluasi dilakukan terhadap data uji yang telah dipisahkan sebelumnya pada proses transformasi data. Evaluasi ini bertujuan untuk mengukur sejauh mana model LightGBM yang telah dibangun mampu memberikan prediksi probabilistik yang akurat terhadap kemungkinan terciptanya gol (*expected goals*).

Untuk mendapatkan gambaran kinerja yang komprehensif, penilaian model dilakukan menggunakan serangkaian metrik evaluasi. Setiap metrik dipilih untuk mengukur aspek performa yang berbeda-beda, mulai dari akurasi nilai probabilitas hingga kemampuan model dalam membedakan antara tembakan yang menghasilkan gol dan yang tidak. Pemilihan metrik-metrik ini selaras dengan tujuan utama model xG, yaitu untuk menghasilkan prediksi dalam bentuk probabilitas, bukan sekadar klasifikasi biner.

4.5.1 *Performing Classification Probabilities*

Langkah pertama dalam tahap evaluasi adalah menjalankan prediksi menggunakan model final yang telah dilatih dan dikalibrasi. Proses ini dimulai dengan memuat kembali model yang telah disimpan (*calibrated_model.joblib*) ke dalam lingkungan kerja.

Selanjutnya, model tersebut diaplikasikan pada himpunan data uji (X_{test}), yaitu data yang sama sekali belum pernah dilihat oleh model selama proses pelatihan maupun *tuning*. Proses prediksi ini secara spesifik bertujuan untuk menghasilkan nilai probabilitas, bukan klasifikasi biner. Dengan kata lain, untuk setiap data tembakan pada data uji, model akan menghitung dan mengeluarkan estimasi probabilitas tembakan tersebut akan menjadi gol.

Hasil dari proses ini adalah sebuah rangkaian nilai xG baru, di mana setiap nilai merepresentasikan prediksi probabilitas untuk setiap pengamatan dalam data uji.

4.5.2 *Evaluation Metrics*

Pada tahap ini, kinerja model dievaluasi dari dua perspektif utama. Pertama adalah evaluasi kualitas probabilitas, yang mengukur seberapa baik dan akurat nilai xG mentah yang dihasilkan. Metrik yang digunakan untuk tujuan ini adalah *Brier Score*, ROC AUC, dan *Log-Loss*. Ketiga metrik ini secara langsung menggunakan *output* probabilitas dari model untuk menilai aspek kalibrasi, kemampuan diskriminatif, dan tingkat kesalahan prediksi secara umum.

Perspektif kedua adalah evaluasi kinerja klasifikasi. Untuk analisis ini, nilai probabilitas xG terlebih dahulu dikonversi menjadi prediksi biner (gol atau tidak

gol) dengan menerapkan ambang batas standar 0.5. Berdasarkan hasil prediksi biner ini, kinerja model kemudian diukur menggunakan metrik klasifikasi klasik, yaitu *Precision*, *Recall*, dan *F1-Score*. Analisis ganda ini memberikan gambaran yang lengkap, tidak hanya tentang akurasi probabilitas model, tetapi juga tentang efektivitas praktisnya sebagai alat klasifikasi.

4.5.2.1 Brier Score

Dalam penelitian ini, perhitungan *Brier Score* dilakukan menggunakan pustaka *scikit-learn* (*sklearn*) dengan memanggil fungsi *brier_score_loss* dari modul *sklearn.metrics*. Tabel 4.9 berikut menunjukkan hasil evaluasi model LightGBM berdasarkan *Brier Score*.

Tabel 4.9 Hasil Evaluasi *Brier Score* Model LightGBM

Metrik	Nilai
<i>Brier Score</i>	0.0626

Nilai *Brier Score* sebesar 0.0626 yang diperoleh model ini tergolong sangat rendah, yang mengindikasikan tingkat akurasi probabilitas yang tinggi. Secara praktis, skor ini menunjukkan bahwa prediksi nilai xG yang dihasilkan oleh model sangat andal dan terkalibrasi dengan baik. Artinya, jika model memberikan nilai xG sebesar 0.1 (10%) pada sekelompok tembakan, maka secara rata-rata memang sekitar 10% dari tembakan tersebut yang benar-benar menjadi gol di dunia nyata.

4.5.2.2 ROC AUC

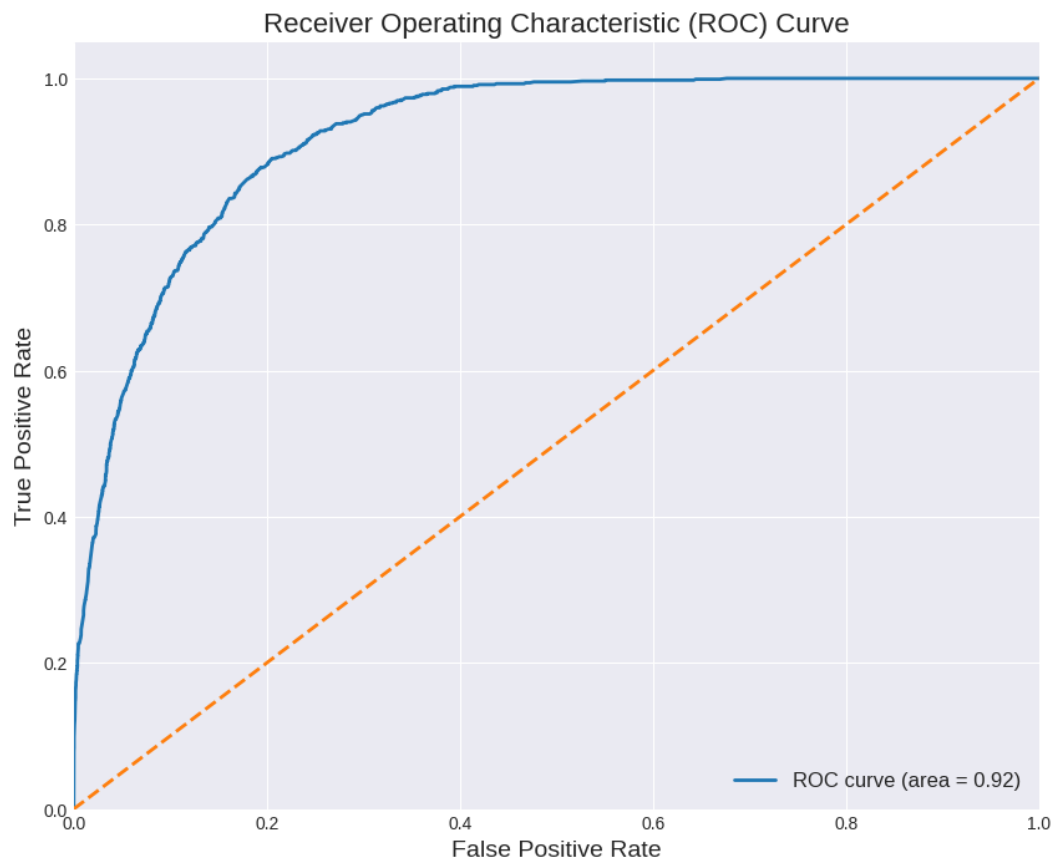
Nilai ROC AUC berkisar dari 0.5 (kinerja setara tebakan acak) hingga 1.0 (kinerja sempurna). Perhitungan dilakukan menggunakan pustaka *scikit-learn*

dengan fungsi *roc_auc_score*. Tabel 4.10 berikut menunjukkan hasil evaluasi model LightGBM berdasarkan ROC AUC.

Tabel 4.10 Hasil Evaluasi ROC AUC Model LightGBM

Metrik	Nilai
ROC AUC	0.9209

Nilai ROC AUC sebesar 0.9209 menunjukkan bahwa model memiliki kemampuan diskriminatif yang sangat baik. Ini berarti, model sangat andal dalam memberikan peringkat pada tembakan, di mana tembakan berbahaya secara konsisten mendapatkan nilai xG yang lebih tinggi daripada tembakan dengan peluang rendah. Visualisasi dari skor ini dapat dilihat pada kurva ROC, di mana kurva model menjauh secara signifikan dari garis diagonal acak. Visualisasi kinerja diskriminatif model ditampilkan pada Gambar 4.23, yang memperlihatkan perbandingan antara ROC *curve* model dengan *baseline random guess*.



Gambar 4.23 Receiver Operating Characteristic (ROC) Curve

Gambar 4.23 secara visual menggambarkan kemampuan diskriminatif model dalam membedakan antara tembakan yang menghasilkan gol dan yang tidak. Kurva biru (model) yang menjulang jauh di atas garis putus-putus oranye (model tembakan acak) dan melengkung tajam ke arah sudut kiri atas menunjukkan performa yang sangat baik. Posisi ini mengindikasikan bahwa model mampu mencapai Tingkat Positif Benar (*True Positive Rate*) yang tinggi sambil mempertahankan Tingkat Positif Palsu (*False Positive Rate*) yang rendah di berbagai ambang batas probabilitas.

4.5.2.3 Akurasi

Akurasi mengukur proporsi total prediksi yang benar (baik gol maupun tidak gol) dari keseluruhan sampel pada data uji. Untuk menghitung metrik ini, nilai probabilitas xG yang berkelanjutan terlebih dahulu dikonversi menjadi prediksi kelas biner (0 atau 1) dengan menerapkan ambang batas klasifikasi standar, yaitu 0,5. Proses perhitungan kemudian dilakukan menggunakan fungsi `accuracy_score` dari pustaka `scikit-learn`. Tabel 4.11 berikut menyajikan hasil evaluasi model LightGBM berdasarkan metrik akurasi.

Tabel 4.11 Hasil Evaluasi Akurasi Model LightGBM

Metrik	Nilai
Akurasi	0,91

Nilai akurasi yang dicapai oleh model adalah 0,9101 (atau 91,01%), yang menunjukkan bahwa model mampu mengklasifikasikan hasil akhir tembakan dengan benar pada sebagian besar data uji. Angka ini mengindikasikan tingkat kesesuaian yang tinggi antara prediksi model dengan hasil aktual di lapangan. Meskipun demikian, perlu dicatat bahwa dalam konteks *dataset* yang tidak seimbang (*imbalanced*) seperti prediksi gol, di mana jumlah tembakan yang tidak menghasilkan gol jauh lebih dominan, metrik akurasi bisa menjadi kurang representatif jika dianalisis secara terpisah. Oleh karena itu, performa ini perlu divalidasi bersama metrik lain seperti ROC AUC dan F1-Score untuk mendapatkan pemahaman yang lebih komprehensif mengenai kemampuan model dalam menangani kelas minoritas (gol).

4.5.2.4 Presisi

Untuk menghitung presisi, probabilitas xG terlebih dahulu diubah menjadi label biner (0 atau 1) dengan ambang batas 0,5. Perhitungan kemudian dilakukan menggunakan fungsi *precision_score* dari *scikit-learn*. Tabel 4.12 berikut menunjukkan hasil evaluasi model LightGBM berdasarkan presisi.

Tabel 4.12 Hasil Evaluasi Presisi Model LightGBM

Metrik	Nilai
Presisi	0,8986

Nilai presisi sebesar 0,8986 (atau 89,86%) tergolong sangat tinggi. Ini mengindikasikan bahwa ketika model memprediksi sebuah tembakan akan menjadi gol (dengan $xG > 0,5$), prediksinya benar hampir 90% dari waktu. Dengan kata lain, model memiliki tingkat false positive yang rendah.

4.5.2.5 Recall

Seperti presisi, *recall* dihitung berdasarkan label biner menggunakan fungsi *recall_score* dari *scikit-learn*. Tabel 4.13 berikut menunjukkan hasil evaluasi model LightGBM berdasarkan *recall*.

Tabel 4.13 Hasil Evaluasi *Recall* Model LightGBM

Metrik	Nilai
<i>Recall</i>	0,9106

Nilai *recall* sebesar 0,9106 (atau 91,06%) menunjukkan tingkat cakupan yang sangat baik. Ini berarti model mampu mengidentifikasi lebih dari 91% dari total gol yang terjadi di dalam data uji. Kemampuan ini menandakan bahwa model tidak banyak melewatkan kejadian-kejadian gol yang sesungguhnya.

4.5.2.6 *F1-Score*

Perhitungan dilakukan menggunakan fungsi *f1_score* dari *scikit-learn* pada label prediksi biner. Tabel 4.14 berikut menunjukkan hasil evaluasi model LightGBM berdasarkan *F1-Score*.

Tabel 4.14 Hasil Evaluasi *F1-Score* Model LightGBM

Metrik	Nilai
<i>F1-Score</i>	0,8998

Nilai *F1-Score* sebesar 0,8998 menunjukkan adanya keseimbangan yang sangat baik antara presisi dan *recall*. Skor yang tinggi ini mengonfirmasi bahwa model tidak hanya akurat saat memprediksi gol, tetapi juga komprehensif dalam menangkap sebagian besar gol yang terjadi, menjadikannya model klasifikasi yang efektif secara keseluruhan.

4.5.2.7 *Log-Loss*

Nilai *Log-Loss* berkisar dari 0 hingga tak terhingga, di mana skor yang lebih rendah adalah lebih baik. Skor 0 menandakan model yang sempurna. Perhitungan dilakukan menggunakan fungsi *log_loss* dari *scikit-learn*. Tabel 4.15 berikut menunjukkan hasil evaluasi model LightGBM berdasarkan *Log-Loss*.

Tabel 4.15 Hasil Evaluasi *Log-Loss* Model LightGBM

Metrik	Nilai
<i>Log-Loss</i>	0,2

Nilai *Log-Loss* sebesar 0,2 tergolong rendah. Hasil ini selaras dengan *Brier Score* yang rendah, yang kembali mengonfirmasi bahwa model tidak hanya membuat prediksi yang benar secara umum, tetapi juga memiliki tingkat

'keyakinan' yang sesuai pada setiap prediksinya. Ini menunjukkan bahwa model mampu memberikan probabilitas yang mencerminkan ketidakpastian secara akurat.

4.5.3 Perbandingan Efisiensi Komputasi

Selain akurasi prediktif, efisiensi komputasi merupakan aspek krusial dalam evaluasi model, karena hal ini memengaruhi waktu yang dibutuhkan untuk pelatihan. Untuk itu, dilakukan perbandingan waktu eksekusi secara menyeluruh antara model LightGBM yang digunakan dalam penelitian ini dengan beberapa algoritma alternatif, yaitu XGBoost, *Random Forest*, dan AdaBoost. Pengujian dilakukan dengan metodologi yang konsisten, di mana total waktu eksekusi untuk setiap model diukur dari awal proses pelatihan. dapat dilihat pada Tabel 4.16

Tabel 4.16 Tabel Perbandingan Waktu Komputasi

Model	Total Waktu (detik)
LightGBM	49,751553
AdaBoost	106,569891
XGBoost	119,356985
Random Forest	136,376029

Tabel 4.16 menyajikan hasil perbandingan efisiensi komputasi antara empat algoritma model yang berbeda, dengan mengukur total waktu eksekusi dalam satuan detik. Hasil dari perbandingan ini secara jelas menunjukkan keunggulan superior dari LightGBM dalam hal kecepatan, yang mampu menyelesaikan pelatihan hanya dalam 49,75 detik. Waktu ini secara signifikan lebih cepat dibandingkan dengan para pesaingnya. Model AdaBoost membutuhkan waktu 106,57 detik, sementara XGBoost yang sering dianggap sebagai pesaing utama mencatatkan waktu 119,36 detik. Random Forest menjadi model yang paling

intensif secara komputasi dengan waktu 136,38 detik, hampir tiga kali lebih lama dibandingkan LightGBM. Temuan ini secara kuantitatif mengonfirmasi bahwa LightGBM tidak hanya unggul dalam performa prediktif, tetapi juga merupakan pilihan yang paling efisien dari segi sumber daya komputasi.

4.6 Interpretasi Hasil

Setelah performa model diukur secara kuantitatif melalui metrik-metrik evaluasi, bab ini berfokus pada interpretasi hasil secara lebih kontekstual untuk memberikan pemahaman yang holistik. Interpretasi akan disajikan dari tiga sudut pandang utama: mendemonstrasikan kemampuannya dalam aplikasi praktis (*Practical Application*) pada sebuah pertandingan nyata, membandingkan kinerja model dengan standar yang ada di literatur akademis (*Academic Benchmark*), dan terakhir, mengukur kualitas prediksinya terhadap standar industri (*Industry Benchmark*) yang sudah mapan.

4.6.1 *Practical Application*

Setelah kinerja model divalidasi melalui metrik kuantitatif, penting untuk mendemonstrasikan utilitas praktisnya dalam menganalisis skenario pertandingan yang sesungguhnya. Aplikasi ini bertujuan untuk mengubah data abstrak menjadi wawasan taktis yang dapat dipahami. Dengan menerapkan model pada pertandingan nyata, kita dapat melampaui skor akhir dan menganalisis narasi permainan secara lebih mendalam, tim mana yang benar-benar menciptakan peluang lebih berkualitas, seberapa efisien penyelesaian akhir mereka, dan bagaimana dinamika permainan berfluktuasi.

4.6.1.1 Real Match Application

Tabel 4.17 menyajikan visualisasi distribusi nilai xG yang dihasilkan oleh model yang dikembangkan dalam penelitian ini untuk setiap peluang yang tercipta dalam pertandingan antara tim nasional Inggris melawan Iran pada Piala Dunia FIFA 2022. Pemilihan pertandingan ini sebagai pengaplikasian didasarkan pada karakteristiknya yang ideal untuk pengujian model. Sebagai anomali statistik (skor 6-2), laga ini tidak hanya menyediakan volume data yang kaya, tetapi juga menyajikan disparitas ekstrem antara xG dan gol aktual, sehingga memungkinkan evaluasi fenomena *finishing overperformance*. Selain itu, kontras taktis yang tajam antara kedua tim menjadikannya laboratorium sempurna untuk menunjukkan bagaimana model dapat secara objektif membedakan kualitas peluang yang dihasilkan dari dua pendekatan strategis yang berlawanan.

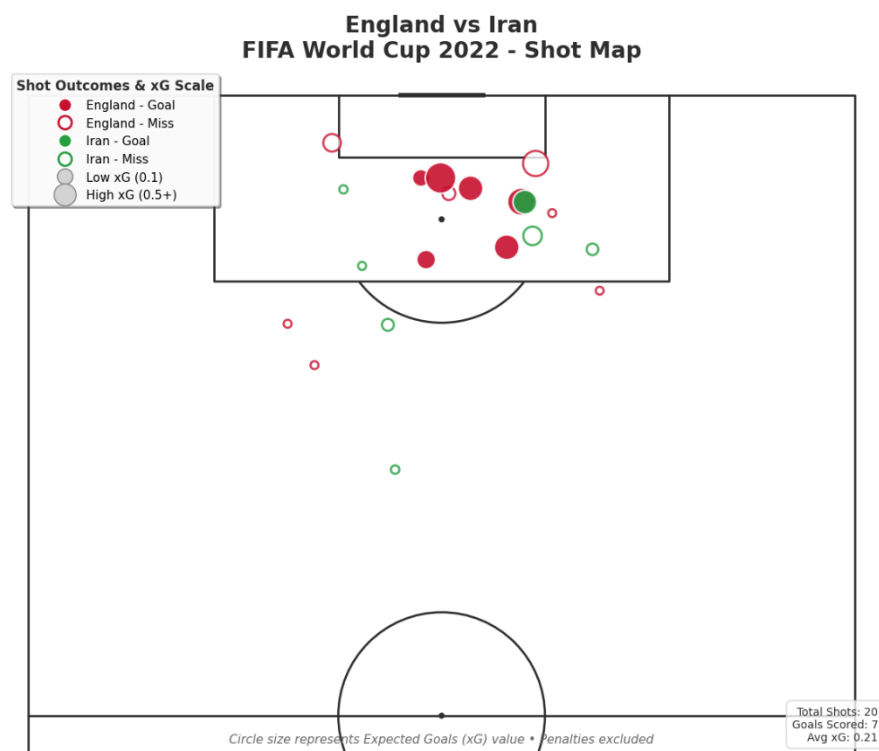
Tabel 4.17 Statistik Berdasarkan xG dari Model LightGBM (Non-Penalti)

Tim Nasional	Total xG	Jumlah Tembakan	Jumlah Gol Aktual	Rata-rata xG per Tembakan	Diferensial Gol (Gol – xG)
Inggris	3,09	13	6	0,238	+2,91
Iran	0,87	7	1	0,125	+0,13

Dari Tabel 4.17, terlihat bahwa tim nasional Inggris mendominasi pertandingan dengan menciptakan total 3,09 xG dari 13 tembakan, menunjukkan kualitas peluang yang sangat tinggi dengan rata-rata 0,238 xG per tembakan. Performa penyelesaian akhir mereka luar biasa, terbukti dengan keberhasilan mencetak 6 gol, menghasilkan diferensial positif sebesar +2,91, yang berarti mereka mencetak hampir tiga gol lebih banyak dari yang diharapkan. Di sisi lain, Iran hanya mampu menghasilkan 0,87 xG dari 7 tembakan, dengan rata-rata kualitas peluang

yang lebih rendah (0,125 xG per tembakan). Dengan 1 gol yang dicetak, performa mereka sesuai ekspektasi dengan diferensial +0,13. Analisis mendalam seperti pada kasus pertandingan ini menjadi bukti nyata akan nilai praktis dari model yang dikembangkan.

Lebih lanjut, model ini juga dapat diintegrasikan sebagai alat ukur performa yang tervisualisasi secara intuitif dan informatif. Gambar 4.24 menyajikan visualisasi sebaran tembakan (*shot map*) beserta nilai xG masing-masing peluang yang terjadi pada pertandingan antara Inggris melawan Iran di Piala Dunia 2022.



Gambar 4.24 *Shot Map* Inggris vs Iran

Gambar 4.24 menyajikan visualisasi *shot map* dari pertandingan Inggris melawan Iran. Setiap lingkaran pada grafik merepresentasikan satu tembakan, dengan posisinya menunjukkan lokasi dari mana tembakan itu dilepaskan. Warna

lingkaran membedakan hasil dari tembakan tersebut: merah untuk gol yang dicetak oleh Inggris, hijau untuk gol oleh Iran, dan abu-abu untuk tembakan yang gagal menjadi gol (misalnya, meleset atau diselamatkan). Ukuran dari setiap lingkaran juga memiliki makna penting, di mana semakin besar lingkaran, semakin tinggi nilai xG-nya, yang menandakan peluang gol yang lebih besar. Sebaliknya, lingkaran yang lebih kecil merepresentasikan tembakan dengan nilai xG yang rendah. Dari visualisasi ini, terlihat jelas bahwa Inggris berhasil mencetak banyak gol dari posisi-posisi yang sangat menguntungkan di dalam kotak penalti, yang direpresentasikan oleh lingkaran berukuran sedang hingga besar.

4.6.1.2 xG Prediction Application Interface

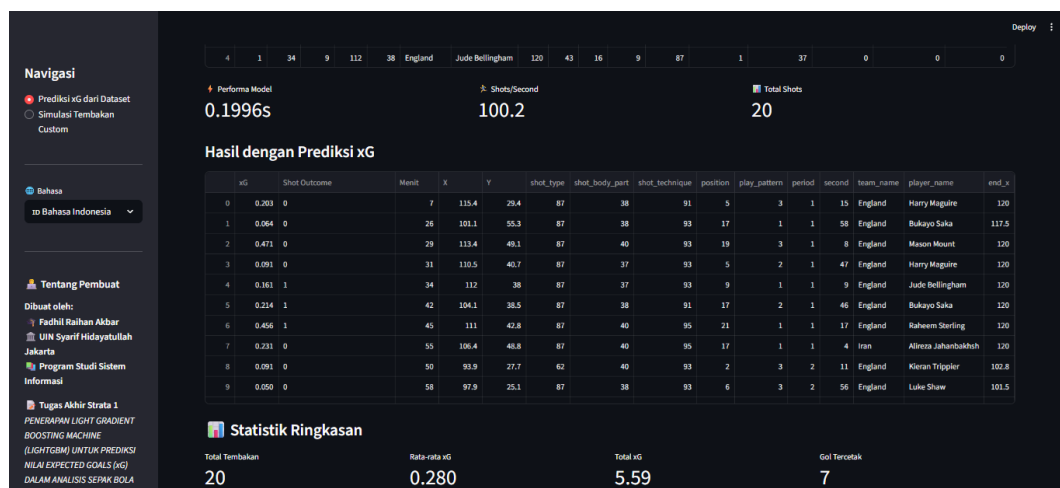
Untuk menjembatani kesenjangan antara model *machine learning* yang kompleks dengan penerapan praktis oleh pengguna akhir, sebuah antarmuka aplikasi web interaktif dikembangkan menggunakan *framework* Streamlit. Aplikasi ini, yang dinamakan "*xG Prediction Interface*", dirancang untuk menyediakan akses yang mudah dan intuitif terhadap fungsionalitas model prediksi xG tanpa memerlukan keahlian teknis atau pemrograman dari pengguna. Tujuannya adalah untuk mengubah model yang telah dilatih menjadi sebuah alat analisis yang fungsional, visual, dan dapat ditindaklanjuti.

Aplikasi ini dibangun dengan arsitektur modular yang memisahkan antara logika antarmuka, manajemen model, dan pemrosesan data. Teknologi inti yang digunakan meliputi Streamlit untuk membangun antarmuka web, Pandas untuk manipulasi data, Scikit-learn dan LightGBM untuk menjalankan prediksi, serta

Matplotlib yang dikombinasikan dengan mplsoccer untuk menghasilkan visualisasi lapangan sepak bola yang akurat dan profesional.

Antarmuka ini menawarkan dua fitur utama yang dirancang untuk memenuhi kebutuhan analisis yang berbeda:

- Prediksi Berdasarkan *Dataset (Batch Prediction)*: Fitur ini memungkinkan pengguna untuk mengunggah *file* CSV yang berisi data tembakan dalam jumlah besar. Setelah diunggah, aplikasi akan memproses setiap baris data, menerapkan model prediksi, dan menampilkan hasilnya dalam bentuk tabel yang interaktif. Hasil prediksi nilai xG untuk setiap tembakan kemudian dapat diunduh kembali untuk analisis lebih lanjut. Fungsionalitas ini sangat berguna bagi analis yang perlu mengevaluasi performa tembakan dari satu atau beberapa pertandingan secara sekaligus. Tampilan antarmuka untuk fitur ini disajikan pada Gambar 4.25 dan Gambar 4.26.

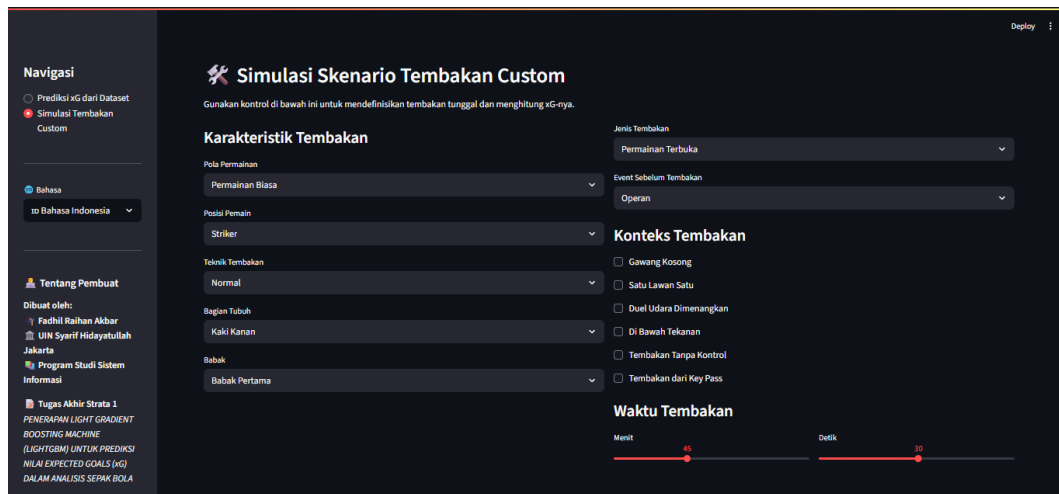


Gambar 4.25 Tampilan Halaman Prediksi Berdasarkan *Dataset*

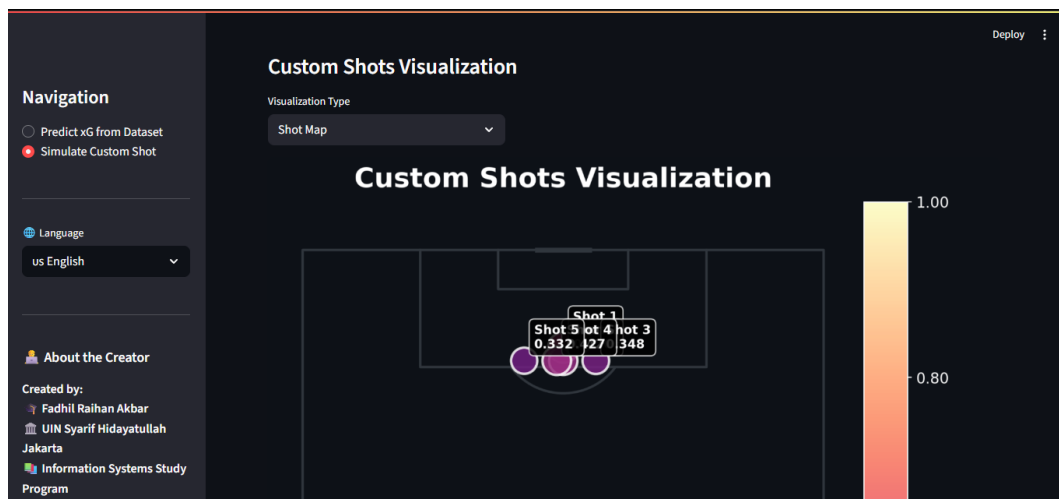


Gambar 4.26 Tampilan Halaman Visualisasi Prediksi Berdasarkan *Dataset*

- b. Simulasi Tembakan Kustom (*Custom Shot Simulation*): Fitur ini menyediakan sebuah simulator interaktif di mana pengguna dapat secara manual mengatur berbagai parameter dari sebuah skenario tembakan. Pengguna dapat menyesuaikan variabel seperti jarak ke gawang, sudut tembakan, bagian tubuh yang digunakan, teknik tembakan, dan ada atau tidaknya tekanan dari lawan melalui serangkaian *slider* dan menu pilihan. Berdasarkan input tersebut, model akan menghitung nilai xG secara *real-time* dan menampilkannya bersama dengan visualisasi *shot map* yang dinamis. Fitur ini berfungsi sebagai alat edukasi dan eksplorasi yang sangat baik untuk memahami bagaimana faktor-faktor yang berbeda memengaruhi kualitas sebuah peluang. Antarmuka simulator ini ditunjukkan pada Gambar 4.27 dan Gambar 4.28.



Gambar 4.27 Tampilan Halaman Simulasi Tembakan Kustom



Gambar 4.28 Tampilan Halaman Visualisasi Simulasi Tembakan Kustom

Secara keseluruhan, pengembangan antarmuka aplikasi ini mentransformasi model prediktif dari sebuah artefak komputasi menjadi sebuah perangkat analisis praktis. Hal ini memungkinkan pelatih, analis, atau bahkan penggemar untuk berinteraksi langsung dengan wawasan berbasis data, mengeksplorasi skenario

hipotetis, dan memperoleh pemahaman yang lebih dalam mengenai dinamika penciptaan peluang dalam sepak bola.

4.6.2 Academic Benchmark

Untuk menilai kinerja model yang dikembangkan dalam penelitian ini secara lebih komprehensif, dilakukan perbandingan terhadap hasil evaluasi dari beberapa model xG pada studi sebelumnya. Untuk menciptakan tolok ukur akademis yang relevan, perbandingan kinerja difokuskan secara khusus pada keluarga algoritma *boosting* yang dikenal sebagai metode berkinerja tinggi dalam literatur. Rangkuman perbandingan hasil evaluasi terhadap beberapa studi terdahulu, termasuk model-model tersebut, dapat dilihat pada Tabel 4.18.

Tabel 4.18 Perbandingan Model xG pada Berbagai Model *Boosting* Literatur

Penulis & Tahun	Model	Brier Score	ROC AUC	Akurasi	Presisi	Recall	F1-Score	Log-Loss
Eggels <i>et al.</i> (2016)	<i>Ada-Boost</i>	–	0,670	–	0,624	0,773	0,688	–
Anzer & Bauer (2021)	<i>Gradient Boosting Machine</i>	–	0,822	–	0,646	0,181	–	–
Haaren (2021)	<i>Boosting Machine</i>	0,082	0,793	–	–	–	–	–
Cavus & Biecek (2022)	LightGBM	0,173	0,818	0,904	0,748	0,721	0,734	0,520
ElHabr (2023)	XGBoost (Opta npxG)	0,0715	–	–	–	–	–	–
Mead <i>et al.</i> (2023)	XGBoost	0,0799	0,8	–	–	–	–	0,28184
Senn (2024)	XGBoost	–	0,718	0,891	0,536	0,071	0,125	–
Model Penelitian	LightGBM (with Calibration)	0,0626	0,9209	0,9101	0,8986	0,9106	0,8998	0,2

Tabel 4.18 secara komprehensif menunjukkan bahwa model LightGBM dengan kalibrasi yang dikembangkan dalam penelitian ini memiliki kinerja yang

superior secara signifikan di semua metrik evaluasi jika dibandingkan dengan model-model xG dari studi-studi terdahulu, termasuk yang menggunakan algoritma *boosting* populer. Dari sisi akurasi probabilitas, model ini mencatatkan *Brier Score* terendah yaitu 0,0626 dan *Log-Loss* terendah 0,2, mengungguli model-model seperti XGBoost dari ElHabr (2023) dan Mead *et al.* (2023). Dalam hal kemampuan diskriminatif, skor ROC AUC sebesar 0,9209 adalah yang tertinggi di antara semua pembandingan, menunjukkan kemampuan luar biasa dalam membedakan peluang gol. Lebih lanjut, pada metrik klasifikasi, model ini juga mendominasi dengan akurasi 0,9101, presisi 0,8986, *recall* 0,9106, dan *F1-Score* 0,8998, yang secara keseluruhan membuktikan bahwa model tidak hanya akurat dalam memprediksi gol, tetapi juga komprehensif dalam mengidentifikasi hampir semua kejadian gol yang sebenarnya, sekaligus menjaga keseimbangan yang sangat baik antara kedua aspek tersebut.

4.6.3 Industrial Benchmark

Untuk mengevaluasi sejauh mana performa model xG ini dalam konteks prediksi pertandingan secara langsung, dilakukan perbandingan hasil prediksi dengan data xG yang disediakan oleh beberapa penyedia statistik sepak bola ternama seperti Opta, *Pro Football Focus* (PFF), FBref, dan xGScore.

Perbandingan ini dilakukan pada tiga pertandingan kunci di ajang Piala Dunia 2022, yaitu Inggris vs Iran, Inggris vs Prancis, dan Argentina vs Kroasia, serta satu pertandingan final UEFA Euro 2024. Keempat laga ini dipilih secara strategis untuk menguji dan memvalidasi ketahanan model dalam menganalisis

spektrum skenario pertandingan yang sangat bervariasi, mulai dari dominasi taktis sepihak hingga duel seimbang bertekanan tinggi pada level turnamen puncak.

Analisis ini bertujuan untuk menilai konsistensi dan validitas model dalam konteks aplikatif, serta menakar sejauh mana model yang dikembangkan mampu menghasilkan estimasi yang kompetitif dibandingkan dengan standar industri dalam bidang analisis sepak bola berbasis data.

Tabel 4.19 menyajikan nilai xG yang dihasilkan oleh model ini pada masing-masing pertandingan tersebut, beserta perbandingannya dengan estimasi dari penyedia statistik lainnya.

Tabel 4.19 Perbandingan Model dengan Penyedia Statistik Sepak Bola

Pertandingan	Skor	Sumber	xG Tim A	xG Tim B
England vs Iran – WC 2022	6 – 2	Penelitian ini	England: 3,09	Iran: 0,87
		Opta	England: 2,1	Iran: 1,751
		xGScore.io	England: 2,14	Iran: 1,42
		FBref	England: 2,1	Iran: 1,4
		PFF	England: 2,14	Iran: 1,62
England vs France – WC 2022	1 – 2	Penelitian ini	England: 1,98	France: 0,64
		PFF	England: 2,4	France: 0,73
		xGScore.io	England: 2,55	France: 1,21
		FBref	England: 2,4	France: 0,9
		Opta	England: 2,4	France: 1,012
Argentina vs Croatia – WC 2022	3 – 0	Penelitian ini	Argentina: 2,01	Croatia: 0,95
		PFF	Argentina: 2,12	Croatia: 0,30
		xGScore.io	Argentina: 2,76	Croatia: 0,57
		Opta	Argentina: 2,33	Croatia: 0,52

		FBref	Argentina: 2,3	Croatia: 0,5
Spain vs England – Final EURO 2024	2 – 1	Penelitian ini	England: 0,67	Spain: 1,63
		xGScore.io	England: 0,63	Spain: 1,9
		FBref	England: 0,5	Spain: 1,9
		Opta	England: 0,527	Spain: 1,953
		PFF	–	–

Pada Tabel 4.19, pertandingan antara Inggris melawan Iran di fase grup Piala Dunia 2022 yang berakhir dengan skor 6–2, model yang dikembangkan dalam penelitian ini menghasilkan estimasi nilai xG sebesar 3,09 untuk Inggris dan 0,87 untuk Iran. Jika dibandingkan dengan data dari penyedia statistik lainnya, terdapat perbedaan yang cukup signifikan. Opta mencatat xG sebesar 2,109 (Inggris) dan 1,751 (Iran), sementara xGScore.io melaporkan nilai 2,14 (Inggris) dan 1,42 (Iran). FBref memberikan estimasi serupa yaitu 2,1 untuk Inggris dan 1,4 untuk Iran, sedangkan PFF mencatat 2,14 untuk Inggris dan 1,62 untuk Iran. Meskipun terdapat variasi antar penyedia, model ini menunjukkan kecenderungan yang lebih tinggi dalam memperkirakan dominasi Inggris, dengan nilai xG yang mencerminkan secara lebih jelas disparitas kualitas peluang yang tercipta di antara kedua tim.

Pada pertandingan perempat final antara Inggris dan Prancis (1–2), model ini memprediksi xG sebesar 1,98 untuk Inggris dan 0,64 untuk Prancis. Angka ini mengindikasikan bahwa Inggris menciptakan peluang dengan kualitas lebih tinggi dibanding Prancis, meskipun hasil akhir menunjukkan sebaliknya. Bila dibandingkan dengan penyedia data lainnya, PFF mencatat 2,4 (Inggris) dan 0,73 (Prancis), xGScore.io memberikan 2,55 dan 1,21, sementara FBref dan Opta

masing-masing memperkirakan 2,4 dan 0,9 serta 2,407 dan 1,012. Secara umum, model ini memberikan estimasi yang lebih konservatif untuk Prancis, namun tetap sejalan dengan kesimpulan bahwa Inggris memiliki dominasi peluang dalam pertandingan tersebut.

Selanjutnya, pada laga semifinal antara Argentina dan Kroasia yang berakhir dengan kemenangan Argentina 3–0, model ini memperkirakan nilai xG sebesar 2,01 untuk Argentina dan 0,95 untuk Kroasia. Estimasi ini relatif sejalan dengan hasil observasi dan mendekati beberapa penyedia data resmi. PFF mencatat 2,12 untuk Argentina dan hanya 0,30 untuk Kroasia. Sementara itu, xgscore.io dan FBref memberikan nilai yang sedikit lebih tinggi untuk Argentina, yakni masing-masing 2,76 dan 2,3, dan nilai lebih rendah untuk Kroasia, yaitu 0,57 dan 0,5. Opta memberikan estimasi sebesar 2,336 (Argentina) dan 0,520 (Kroasia). Secara umum, model ini menampilkan prediksi yang stabil dan mencerminkan keseimbangan realistis antara dominasi Argentina dan ketidakmampuan Kroasia menciptakan peluang berkualitas.

Terakhir, pada pertandingan final Euro 2024 antara Spanyol dan Inggris yang berakhir dengan kemenangan Spanyol 2–1, model ini menghasilkan nilai xG sebesar 1,63 untuk Spanyol dan 0,67 untuk Inggris. Estimasi ini mendekati angka dari beberapa penyedia statistik. xGScore.io mencatat nilai sebesar 1,90 (Spanyol) dan 0,63 (Inggris), sementara FBref dan Opta memberikan hasil serupa, yakni 1,9 dan 0,5 (FBref), serta 1,953 dan 0,527 (Opta). Sayangnya, data dari PFF untuk pertandingan ini tidak tersedia. Perbandingan ini menunjukkan bahwa model yang dikembangkan mampu memberikan prediksi yang sejalan dengan tren umum yang

tercermin dalam data statistik publik, sehingga memperkuat validitas model sebagai alat analisis performa pertandingan sepak bola tingkat tinggi.

4.7 Keterbatasan Penelitian

Penelitian ini memiliki beberapa keterbatasan yang perlu diperhatikan. Pertama, keterbatasan utama terletak pada kelengkapan dan cakupan data. *Dataset* yang digunakan berasal dari sumber open-data StatsBomb yang hanya mencakup liga dan turnamen tertentu, seperti Liga Inggris, La Liga, dan Piala Dunia. Hal ini membatasi kemampuan generalisasi model terhadap kompetisi lain yang memiliki karakteristik permainan berbeda, baik dari segi level kompetisi, gaya bermain, maupun kualitas pemain. Selain itu, meskipun data StatsBomb dikenal kaya akan detail teknis, sejumlah fitur krusial dalam analisis xG seperti posisi kiper atau intensitas tekanan dari pemain bertahan tidak selalu tersedia atau hanya tersedia dalam jumlah terbatas (misalnya *freeze frame* data). Kekosongan ini dapat mengurangi kemampuan model dalam merepresentasikan konteks situasional dari suatu tembakan secara menyeluruh.

Kedua, keberlakuan model yang dibangun secara spesifik pada data dari satu liga atau turnamen tertentu dapat membatasi performanya ketika diterapkan pada kompetisi lain. Perbedaan gaya bermain antar liga, taktik dominan, tingkat kemampuan teknis pemain, serta kondisi permainan yang kontekstual dapat memengaruhi performa model secara signifikan. Dengan demikian, validitas eksternal dari model ini masih perlu diuji secara lebih luas sebelum dapat digunakan secara *general*.

Ketiga, keterbatasan dalam pemahaman domain atau domain *knowledge* turut menjadi tantangan dalam eksplorasi fitur. Tanpa pemahaman mendalam mengenai peran spesifik pemain, strategi taktis, dan pola permainan, terdapat kemungkinan bahwa beberapa fitur bersifat terlalu dangkal atau bahkan mengarah pada interpretasi yang menyesatkan. Hal ini menunjukkan pentingnya kolaborasi antara peneliti data dan praktisi atau analis sepak bola untuk memperkaya proses *feature engineering* dan interpretasi model.

BAB V

PENUTUP

5.1 Kesimpulan

Penelitian ini melakukan penerapan model *Light Gradient Boosting Machine* (LGBM) untuk melakukan prediksi metrik *Expected Goals* (xG) dalam konteks analisis performa tembakan pada pertandingan sepak bola. Data yang digunakan merupakan data tembakan yang telah melalui proses pembersihan dan *preprocessing*, serta dilakukan rekayasa fitur berbasis konteks spasial, temporal, dan teknikal. Selanjutnya, dilakukan pelatihan model menggunakan algoritma LGBM, evaluasi performa dengan metrik seperti *Brier Score* dan ROC AUC, serta analisis interpretasi model menggunakan visualisasi SHAP dan distribusi nilai prediksi xG. Berdasarkan hasil pembahasan penerapan model LGBM untuk prediksi xG dalam pertandingan sepak bola, dapat ditarik kesimpulan:

- a. Penelitian ini menerapkan algoritma LightGBM untuk meningkatkan akurasi dan efisiensi dalam perhitungan xG dalam analisis sepak bola menggunakan *open-data* dari StatsBomb. Proses dimulai dengan tahapan *feature engineering* yang mencakup variabel-variabel penting seperti *distance_to_goal*, *angle_to_goal*, dan *type_before* yang berperan signifikan dalam menentukan probabilitas terjadinya gol. Untuk meningkatkan kalibrasi prediksi probabilistik model, digunakan metode *CalibratedClassifierCV* dengan teknik *isotonic regression* dan *3-fold cross-validation*. Parameter model disetel secara spesifik guna mengoptimalkan performa, antara lain *boosting_type = gbdt*, *num_leaves*

= 15, $max_depth = 84$, $learning_rate = 0,1$, $n_estimators = 100$, serta regulasi melalui $reg_alpha = 0,513$ dan $reg_lambda = 0,971$. Model juga dirancang dengan kontrol terhadap *overfitting* melalui $min_child_samples = 146$, $subsample = 1,0$, dan $colsample_bytree = 1,0$. Hasil konfigurasi ini menunjukkan bahwa LightGBM dapat digunakan secara efisien dan akurat untuk memodelkan metrik xG dalam domain sepak bola, dengan mempertimbangkan kontribusi fitur-fitur relevan dan teknik kalibrasi prediktif.

- b. Performa algoritma LightGBM dalam perhitungan xG dievaluasi menggunakan dua metrik utama, yaitu *Area Under Curve* (AUC) dan *Brier Score*. Berdasarkan hasil evaluasi, model LightGBM menunjukkan nilai *Brier Score* sebesar 0,0626, yang mengindikasikan tingkat kalibrasi probabilistik yang sangat baik dan kesalahan prediksi yang rendah. Selain itu, nilai ROC AUC mencapai 0,9209, yang menunjukkan bahwa model memiliki kemampuan diskriminatif yang sangat tinggi dalam membedakan antara peluang yang berujung pada gol dan yang tidak. Jika dibandingkan dengan model-model lain yang digunakan dalam studi ini, LightGBM menunjukkan performa yang relatif unggul berdasarkan kedua metrik evaluasi tersebut. Validasi tambahan terhadap hasil prediksi juga telah dilakukan pada data pertandingan nyata sebagaimana dijelaskan pada bagian interpretasi hasil. Model ini mampu menghasilkan estimasi xG yang selaras dengan konteks situasi pertandingan, sehingga menunjukkan potensi yang baik dalam penerapan nyata untuk analisis sepak bola.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, berikut beberapa saran yang dapat dijadikan dan dipertimbangkan untuk penelitian selanjutnya:

- a. Pada penelitian ini hanya menggunakan satu algoritma pembelajaran mesin, yaitu LightGBM, karena mempertimbangkan efisiensi dan kompleksitas model. Untuk penelitian selanjutnya disarankan untuk mengeksplorasi dan membandingkan beberapa algoritma lain, seperti XGBoost, CatBoost, atau model berbasis neural network, guna memperoleh perspektif yang lebih komprehensif terkait performa dalam perhitungan xG.
- b. Fitur-fitur yang digunakan dalam model ini masih terbatas pada variabel yang tersedia dari open-data StatsBomb. Penelitian selanjutnya disarankan untuk melakukan pengayaan fitur, seperti memasukkan variabel taktis, posisi pemain bertahan lawan, atau kondisi pertandingan (misalnya skor sementara atau menit ke berapa dalam pertandingan), guna meningkatkan konteks spasial dan temporal dalam prediksi xG.
- c. Proses kalibrasi dilakukan menggunakan metode *isotonic* melalui *CalibratedClassifierCV*, namun belum dilakukan evaluasi terhadap metode kalibrasi alternatif. Penelitian selanjutnya dapat mempertimbangkan untuk membandingkan beberapa pendekatan kalibrasi, seperti *Platt scaling* atau *beta calibration*, untuk melihat dampaknya terhadap probabilitas prediktif model.