

# **TUGAS AKHIR**

## **PENERAPAN *LIGHT GRADIENT BOOSTING MACHINE* (LIGHTGBM) UNTUK PREDIKSI NILAI *EXPECTED GOALS* (xG) DALAM ANALISIS SEPAK BOLA**

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Komputer  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Syarif Hidayatullah Jakarta



Disusun oleh:

**Fadhil Raihan Akbar**  
**NIM. 11210930000101**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA  
2025 M/1447 H**

# **TUGAS AKHIR**

## **PENERAPAN *LIGHT GRADIENT BOOSTING MACHINE* (LIGHTGBM) UNTUK PREDIKSI NILAI *EXPECTED GOALS* (xG) DALAM ANALISIS SEPAK BOLA**

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Komputer

Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta



Disusun oleh:

**Fadhil Raihan Akbar**  
**NIM. 11210930000101**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA  
2025 M/1447 H**



# **LEMBAR PENGESAHAN TUGAS AKHIR**

## **PENERAPAN LIGHT GRADIENT BOOSTING MACHINE (LIGHTGBM) UNTUK PREDIKSI NILAI EXPECTED GOALS (xG) DALAM ANALISIS SEPAK BOLA**

**Disusun oleh:**

**Fadhil Raihan Akbar  
NIM. 11210930000101**

Menyetujui,

Dosen Pembimbing I

Dosen Pembimbing II

**Dr. Qurrotul Aini, M.T.**  
**NIP. 197303252009012001**

**Nida'ul Hasanati, S.T.,MMSI.**  
**NIP. 197907182014112002**

Mengetahui,

**Ketua Program Studi Sistem Informasi**

**Dr. Qurrotul Aini, M.T.**  
**NIP. 197303252009012001**



## LEMBAR PENGESAHAN UJIAN

Tugas akhir berjudul **PENERAPAN *LIGHT GRADIENT BOOSTING MACHINE (LIGHTGBM)* UNTUK PREDIKSI NILAI *EXPECTED GOALS* (xG) DALAM ANALISIS SEPAK BOLA** yang ditulis oleh Fadhil Raihan Akbar dengan NIM. 11210930000101 telah diuji dan dinyatakan LULUS dalam sidang Munaqosah Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta pada tanggal xx xxxxx 2025. Tugas akhir ini telah diterima sebagai salah satu syarat memperoleh gelar sarjana Strata Satu (S1) Program Studi Sistem Informasi.

Menyetujui,

Dosen Penguji I

Dosen Penguji II

xxx  
NIP. xxx

Dosen Pembimbing I

xxxx  
NIP. xxxx

Dosen Pembimbing II

Dr. Qurrotul Aini, M.T.  
NIP. 197303252009012001

Nida'ul Hasanati, S.T.,MMSI.  
NIP. 197907182014112002

Mengetahui,

Dekan  
Fakultas Sains dan Teknologi

Ketua  
Program Sistem Informasi

Husniteja Sukmana, ST, M.Sc, Ph.D  
NIP. 197710302001121003

Dr. Qurrotul Aini, M.T.  
NIP. 197303252009012001



## **LEMBAR PERNYATAAN**

**DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI BENAR-BENAR HASIL KARYA SENDIRI DAN BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN.**

Tangerang Selatan, xx xxxx 2025

MATERAI 10K

DAN TTD

**Fadhil Raihan Akbar  
NIM. 11210930000101**





## ABSTRAK

**Fadhil Raihan Akbar – 11210930000101**, Penerapan Light Gradient Boosting Machine (LightGBM) Untuk Prediksi Nilai Expected Goals (xG) Dalam Analisis Sepak Bola, di bawah bimbingan **Dr. Qurrotul Aini, M.T. dan Nida'ul Hasanati, S.T.,MMSI**.

Analisis sepak bola modern menghadapi tantangan dalam menghitung metrik *Expected Goals* (xG) secara akurat dan efisien, karena model linear tradisional gagal menangkap hubungan data yang non-linear, sementara model kompleks lainnya seringkali lambat secara komputasi. Penelitian ini menerapkan algoritma *Light Gradient Boosting Machine* (LightGBM) dengan kerangka kerja *Knowledge Discovery in Databases* (KDD) untuk membangun model prediksi xG menggunakan dataset publik dari StatsBomb. Tujuan penelitian adalah untuk mengevaluasi peningkatan performa akurasi dan efisiensi komputasi yang ditawarkan oleh LightGBM dalam memprediksi nilai xG. Hasil evaluasi pada data uji menunjukkan kinerja model yang sangat unggul, dengan rincian metrik, akurasi 0,9113, presisi 0,8996, *recall* 0,9113, *F1-Score* 0,9010, *Brier Score* 0,0621, ROC AUC 0,9209, dan *Log-Loss* 0,2. Dari segi efisiensi, model LightGBM mampu menyelesaikan proses pelatihan hanya dalam waktu 49.75 detik, jauh mengungguli algoritma pembandingan lainnya. Hasil ini membuktikan bahwa penerapan LightGBM dapat menghasilkan model xG yang tidak hanya sangat akurat dan terkalibrasi dengan baik, tetapi juga unggul secara komputasi untuk analisis sepak bola.

**Kata Kunci:** Analisis sepak bola, *data mining*, *expected goals* (xG), *machine learning*, LightGBM

Bab 1–5 + xxi Halaman + 202 Halaman + 41 Gambar + 34 Tabel + Lampiran  
Pustaka Acuan (108, 2020–2025)

## KATA PENGANTAR

*Assalamualaikum Warahmatullahi Wabarakatuh*

Puji dan syukur atas kehadiran Allah SWT yang telah memberikan rahmat serta hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir ini. Shalawat serta salam selalu tercurahkan kepada junjungan kita Rasulullah Muhammad SAW yang telah membawa kita keluar dari zaman kegelapan.

Pada tugas akhir ini, penulis menyadari bahwa masih banyak terdapat kekurangan. Namun demikian, tugas akhir ini dapat memenuhi salah satu syarat memperoleh gelar Sarjana Komputer strata satu (S1) Program Studi Sistem Informasi di Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta. Skripsi yang berjudul “Penerapan *Light Gradient Boosting Machine* (LightGBM) untuk Prediksi Nilai *Expected Goals* (xG) dalam Analisis Sepak Bola” akhirnya dapat diselesaikan sesuai dengan harapan penulis. Selama penyusunan tugas akhir ini, penulis menghadapi banyak kesulitan dan tantangan.

Pada kesempatan ini penulis juga hendak mengucapkan terima kasih kepada pihak-pihak yang telah membantu memberikan dukungan, sehingga penulis dapat menyelesaikan tugas akhir ini. Ucapan terima kasih secara khusus penulis berikan kepada:

1. Bapak Husni Teja Sukmana, S.T., M.Sc, Ph.D. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.
2. Ibu Dr. Qurrotul Aini, M.T. selaku Ketua Program Studi Sistem Informasi Fakultas Sains dan Teknologi. Sekaligus, selaku Dosen Pembimbing yang telah menyediakan waktunya untuk memberikan ilmu dan tentunya dukungan kepada penulis
3. Bapak Ir. Eri Rustamaji, MBA selaku Sekretaris Program Studi Sistem Informasi Fakultas Sains dan Teknologi.
4. Ibu Nida'ul Hasanati, S.T.,MMSI. selaku Dosen Pembimbing yang telah menyediakan waktunya untuk memberikan ilmu dan tentunya dukungan kepada penulis

5. Dosen-dosen Program Studi Sistem Informasi yang telah memberikan ilmu selama perkuliahan.
6. Keluarga dan teman yang selalu memberikan motivasi, semangat serta arahan selama penyusunan laporan ini dimulai hingga selesai.

Penulis menyadari bahwa dalam penyusunan tugas akhir ini masih jauh dari kata sempurna. Oleh karena itu, penulis tidak menutup diri atas segala bentuk saran dan kritik yang bersifat membangun ke depannya. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi pembaca dan penulis sendiri.

*Wassalamualaikum Warahmatullahi Wabarakatuh.*

Tangerang Selatan, xx xxxx 2025

**Fadhil Raihan Akbar**  
**NIM. 11210930000101**

## DAFTAR ISI

<b>LEMBAR PENGESAHAN TUGAS AKHIR .....</b>	<b>iii</b>
<b>LEMBAR PENGESAHAN UJIAN .....</b>	<b>v</b>
<b>LEMBAR PERNYATAAN .....</b>	<b>vii</b>
<b>ABSTRAK .....</b>	<b>ix</b>
<b>KATA PENGANTAR.....</b>	<b>x</b>
<b>DAFTAR ISI.....</b>	<b>xii</b>
<b>DAFTAR GAMBAR .....</b>	<b>xvii</b>
<b>DAFTAR TABEL .....</b>	<b>xx</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Identifikasi Masalah .....	15
1.3 Rumusan Masalah .....	15
1.4 Batasan Masalah .....	15
1.5 Tujuan Penelitian.....	16
1.6 Manfaat Penelitian .....	17
1.7 Metode Penelitian .....	18
1.8 Sistematika Penulisan .....	20
<b>BAB 2 TINJAUAN PUSTAKA .....</b>	<b>23</b>
2.1 <i>Data Mining</i> .....	23
2.2 Klasifikasi Probabilistik ( <i>Probabilistic Classification</i> ) .....	25
2.3 <i>Knowledge Discovery in Databases</i> (KDD) .....	27
2.4 <i>Exploratory Data Analysis</i> (EDA).....	30
2.5 <i>Machine Learning</i> .....	31
2.6 <i>Gradient Boosting</i> .....	34
2.7 <i>Light Gradient Boosting Machine</i> .....	41
2.8 Data Non-Linear.....	49

2.9	Sepak Bola .....	52
2.10	Analisis Sepak Bola .....	53
2.11	<i>Expected Goals</i> (xG).....	55
2.12	<i>Brier Score</i> .....	60
2.13	<i>Confusion Matrix</i> .....	62
2.14	<i>Receiver Operating Characteristic Area Under Curve</i> (ROC AUC) .....	63
2.15	<i>Isotonic Regression</i> .....	66
2.16	Akurasi .....	67
2.17	Presisi ( <i>Precision</i> ) .....	68
2.18	<i>Recall</i> .....	70
2.19	<i>F1-Score</i> .....	70
2.20	<i>Log-Loss</i> ( <i>Cross-Entropy Loss</i> ).....	71
2.21	<i>Feature Engineering</i> .....	73
2.22	<i>Tools Penelitian</i> .....	76
2.22.1	Python.....	76
2.22.2	Pandas.....	77
2.22.3	Scikit-learn .....	78
2.22.4	Matplotlib .....	79
2.22.5	Seaborn.....	79
2.23	Penelitian Sejenis .....	80
2.24	Ranah Penelitian.....	88
<b>BAB 3 METODOLOGI PENELITIAN .....</b>		<b>91</b>
3.1	Pendekatan Penelitian .....	91
3.2	Sumber Data.....	91
3.2.1	Populasi dan Sampel Penelitian .....	92
3.2.2	Kriteria Seleksi Sampel.....	92
3.3	Perangkat Penelitian.....	93
3.4	Pengumpulan Data .....	93
3.5	Pengembangan Model.....	94
3.5.1	<i>Knowledge Discovery in Databases</i> .....	94
3.5.2	Pemodelan LightGBM .....	107

3.6	Intrepretasi Hasil .....	108
3.6.1	<i>Practical Application</i> .....	109
3.6.2	<i>Academic Benchmark</i> .....	109
3.6.3	<i>Industrial Benchmark</i> .....	109
3.7	Implikasi Penelitian.....	110
3.7.1	Implikasi Teoritis.....	110
3.7.2	Implikasi Praktis.....	110
3.8	Tahapan Penelitian .....	111
3.9	Jadwal Penelitian.....	112
<b>BAB 4 HASIL DAN PEMBAHASAN .....</b>		<b>115</b>
4.1	<i>Data Selection</i> .....	115
4.1.1	<i>Data Understanding</i> .....	115
4.1.2	<i>Data Acquisition</i> .....	116
4.2	<i>Data Preprocessing</i> .....	117
4.2.1	<i>Event Data Filtering</i> .....	118
4.2.2	<i>Exploratory Data Analysis (EDA)</i> .....	122
4.2.3	<i>Data Cleansing</i> .....	136
4.3	<i>Data Transformation</i> .....	137
4.3.1	<i>Feature Engineering</i> .....	137
4.3.2	<i>Split Dataset</i> .....	145
4.3.3	<i>Feature Selection</i> .....	147
4.4	Data Mining .....	147
4.4.1	<i>Hyperparameter Tuning</i> .....	148
4.4.2	<i>Train Model</i> .....	151
4.4.3	<i>Model Probability Calibration</i> .....	153
4.5	<i>Evaluation</i> .....	156
4.5.1	<i>Performing Classification Probabilities</i> .....	157
4.5.2	<i>Evaluation Metrics</i> .....	157
4.5.3	Perbandingan Efisiensi Komputasi .....	164
4.6	Interpretasi Hasil .....	166
4.6.1	<i>Practical Application</i> .....	166

4.6.2	<i>Academic Benchmark</i> .....	176
4.6.3	<i>Industrial Benchmark</i> .....	178
4.7	Implikasi Penelitian .....	181
4.8	Keterbatasan Penelitian .....	182
<b>BAB 5 PENUTUP</b> .....		<b>185</b>
5.1	Kesimpulan .....	185
5.2	Saran .....	186
<b>DAFTAR PUSTAKA</b> .....		<b>187</b>





## DAFTAR GAMBAR

Gambar 1.2 Visualisasi Hubungan Non-Linear Data Sepak Bola (StatsBomb, 2022)	4
Gambar 1.3 Logo Statsbomb	14
Gambar 2.1 Proses KDD (Paucar & Andrade-Arenas, 2025)	28
Gambar 2.2 Contoh Implementasi <i>Supervised Learning</i> (Mahadevkar <i>et al.</i> , 2022)	33
Gambar 2.4 Arsitektur GOSS dan EFB	42
Gambar 2.5 Ilustrasi <i>Level-wise Tree Growth</i> (LightGBM, 2024)	44
Gambar 2.6 Ilustrasi <i>Leaf-wise Tree Growth</i> (LightGBM, 2024)	44
Gambar 2.10 Visualisasi xG pada Pertandingan Langsung (Statsbomb, 2024)	60
Gambar 2.11 Contoh ROC AUC (Nahm, 2022)	65
Gambar 2.12 Ranah Penelitian	89
Gambar 3.1 Struktur Data StatsBomb <i>open-data</i>	95
Gambar 3.2 <i>Flowchart Data Selection</i>	98
Gambar 3.3 <i>Flowchart Data Preprocessing</i>	99
Gambar 3.4 <i>Flowchart Data Transformation</i>	101
Gambar 3.5 <i>Flowchart Data Mining</i>	103
Gambar 3.6 <i>Flowchart Evaluation</i>	107
Gambar 3.7 Tahapan Penelitian	112
Gambar 4.1 Hasil Pemilihan Data	118
Gambar 4.2 Contoh Data Sesudah Pemilihan Variabel	122
Gambar 4.3 Distribusi Menit Pertandingan	123
Gambar 4.4 Distribusi Detik Pertandingan	124
Gambar 4.5 Distribusi Koordinat <i>Start x</i>	125
Gambar 4.6 Distribusi Koordinat <i>Start y</i>	126
Gambar 4.7 Distribusi Pola Permainan ( <i>Play Pattern</i> )	127
Gambar 4.8 Distribusi Posisi Pemain	128
Gambar 4.9 Distribusi Teknik Tembakan	129
Gambar 4.10 Distribusi Bagian Tubuh yang Digunakan untuk Menembak	130

Gambar 4.11 Hasil Tembakan Berdasarkan Jarak dan Sudut .....	132
Gambar 4.12 Tingkat Konversi Gol Berdasarkan Teknik Tembakan .....	133
Gambar 4.13 Tingkat Konversi Gol Berdasarkan Bagian Tubuh .....	134
Gambar 4.15 Visualisasi <i>distance_to_goal</i> .....	138
Gambar 4.16 Visualisasi <i>angle_to_goal</i> .....	140
Gambar 4.18 Visualisasi Distribusi Fitur <i>angle_to_goal</i> .....	141
Gambar 4.20 Grafik Konvergensi Skor Selama Optimasi .....	150
Gambar 4.21 Contoh Nilai Probabilitas .....	153
Gambar 4.22 Kurva Kalibrasi Sebelum dan Sesudah Proses Kalibrasi .....	155
Gambar 4.23 <i>Receiver Operating Characteristic (ROC) Curve</i> .....	160
Gambar 4.24 <i>Shot Map</i> Inggris vs Iran .....	172
Gambar 4.27 Tampilan Halaman Simulasi Tembakan Kustom .....	175
Gambar 4.28 Tampilan Halaman Visualisasi Simulasi Tembakan Kustom .....	176



## DAFTAR TABEL

Tabel 1.1 Perbandingan Kinerja Model Non-Linear vs. Linear Pada Pemodelan xG .....	5
Tabel 1.2 Perkembangan Model xG Saat Ini .....	5
Tabel 1.3 Perbandingan Waktu Pelatihan Berdasarkan Kompleksitas Algoritma... 7	
Tabel 1.4 Perbandingan Performa LightGBM pada Data Non-Linear .....	9
Tabel 1.5 Perbandingan Waktu Pelatihan pada Model <i>Boosting</i> (Florek & Zagdański, 2023).....	11
Tabel 1.6 Keunggulan Bisnis dari Efisiensi Komputasi LightGBM dalam Implementasi Praktis .....	12
Tabel 2.1 Contoh Data Non-Linear (Ahmad <i>et al.</i> , 2021) .....	50
Tabel 2.2 Perkembangan xG .....	56
Tabel 2.3 Daftar Fitur Data Umum dalam Model xG .....	58
Tabel 2.4 <i>Confusion Matrix</i> .....	63
Tabel 2.5 Penelitian Sejenis .....	81
Tabel 3.1 Spesifikasi <i>Hardware</i> dan <i>Software</i> .....	93
Tabel 3.2 <i>Field</i> Data Statsbomb .....	96
Tabel 3.3 Waktu Pelaksanaan Penelitian.....	112
Tabel 4.1 Daftar Kompetisi .....	115
Tabel 4.2 Nama dan Deskripsi Kolom .....	119
Tabel 4.3 Ringkasan Temuan EDA .....	135
Tabel 4.4 Deskripsi Jenis <i>type</i> dalam <i>Dataset</i> StatsBomb. ....	142
Tabel 4.5 Rincian Dimensi <i>Dataset</i> Akhir .....	146
Tabel 4.6 Himpunan Fitur Akhir untuk Pemodelan xG .....	147
Tabel 4.7 Ruang Pencarian <i>Hyperparameter</i> .....	148
Tabel 4.8 Sampel Hasil Iterasi Pencarian (5 kombinasi teratas).....	151
Tabel 4.9 Hasil Evaluasi <i>Brier Score</i> Model LightGBM .....	158
Tabel 4.10 Hasil Evaluasi ROC AUC Model LightGBM .....	159
Tabel 4.11 Hasil Evaluasi Akurasi Model LightGBM .....	161
Tabel 4.12 Hasil Evaluasi Presisi Model LightGBM.....	162

Tabel 4.13 Hasil Evaluasi <i>Recall</i> Model LightGBM .....	163
Tabel 4.14 Hasil Evaluasi <i>F1-Score</i> Model LightGBM.....	163
Tabel 4.15 Hasil Evaluasi <i>Log-Loss</i> Model LightGBM.....	164
Tabel 4.16 Tabel Perbandingan Waktu Komputasi .....	165
Tabel 4.17 Statistik Berdasarkan xG dari Model LightGBM (Non-Penalti) .....	168
Tabel 4.18 Perbandingan Model xG pada Berbagai Model <i>Boosting</i> Literatur ..	177
Tabel 4.19 Perbandingan Model dengan Penyedia Statistik Sepak Bola.....	179

# BAB 1

## PENDAHULUAN

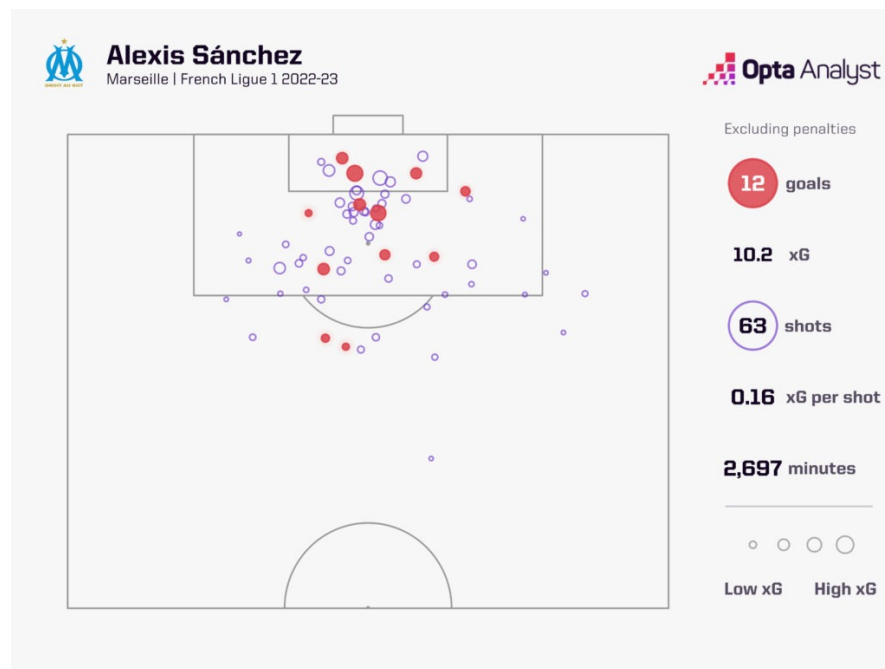
### 1.1 Latar Belakang

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)

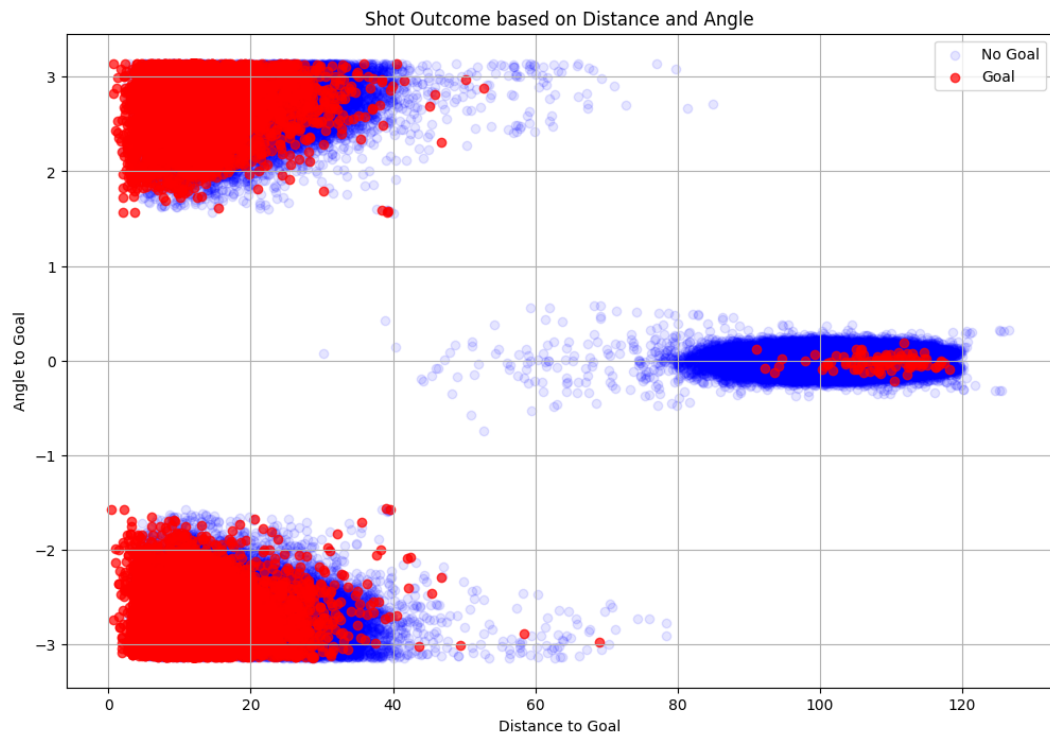
Gambar 1.1 merupakan sebuah *shot map* dari Opta yang menyajikan visualisasi data tembakan yang dilakukan oleh pemain Alexis Sánchez untuk klub Marseille pada kompetisi Liga 1 Prancis musim 2022-2023. Diagram ini memetakan semua lokasi tembakan di sepertiga akhir lapangan, dengan setiap lingkaran merepresentasikan 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. Oleh karena itu, model yang mampu menyajikan hasil akurat dengan komputasi yang cepat dapat memberikan keunggulan kompetitif yang nyata.

Data non-linear merupakan sekumpulan data di mana hubungan antara variabel independen (prediktor) dan variabel dependen (target) tidak dapat direpresentasikan sebagai fungsi linear (Aggarwal, 2020). 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 (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 relevan yang berfokus pada model non-linear dan linear.

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 merangkum berbagai implementasi terkini dalam pemodelan xG, di mana para peneliti menerapkan algoritma dan fitur-fitur inovatif untuk meningkatkan kinerja pemodelan. Namun, kemajuan ini memiliki akibat berupa meningkatnya kompleksitas dan menurunnya efisiensi komputasi. Hal ini menghasilkan model yang akurat tetapi cenderung lambat dan berat untuk diimplementasikan (Manatova & Woo, 2023). Tabel 1.3 menyajikan perbandingan

efisiensi komputasi antara beberapa model *machine learning* pada *dataset* yang berbeda untuk mengilustrasikan hal ini.

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 1.3 befokus pada

potensi akurasi dari model yang kompleks dengan biaya komputasi yang lebih tinggi.

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 / Dataset	Model Pemandang	Metrik Evaluasi	Kinerja Model LightGBM	Kinerja Model Pemandang	Sumber
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>	<i>R-squared</i> ( $R^2$ )	0,76	0,71, 0,66	Garg <i>et al.</i> (2025)
<i>Benchmark 12 Dataset Publik</i>	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 pengembangan model xG, LightGBM memiliki 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>	<b>LightGBM (detik)</b>	<b>GBM (detik)</b>	<b>XGBoost (detik)</b>	<b>CatBoost (detik)</b>
<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 *tree 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 Implementasi Praktis

Skenario Aplikasi & Bisnis	Tantangan dengan Model Lain	Solusi yang Ditawarkan LightGBM	Dampak Bisnis & Finansial
Analisis Siaran Langsung	Kinerja 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.

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

- a. Pemodelan yang dapat mencapai akurasi non-linear tanpa mengorbankan efisiensi komputasi secara signifikan.
- b. Kinerja model secara spesifik melalui proses optimalisasi *hyperparameter tuning*.
- c. Fitur sekuensial untuk menangkap konteks jenis permainan yang mendahului tembakan.
- d. Kalibrasi model menggunakan metode regresi isotonik untuk memastikan nilai probabilitas xG yang dihasilkan lebih andal untuk analisis praktis.
- e. *Interface* berbasis web untuk mempermudah pengaplikasian prediksi xG secara interaktif tanpa memerlukan keahlian pemrograman.

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 kinerja 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 kinerja 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 mengenai algoritma LightGBM dan aplikasinya dalam perhitungan xG, serta dapat dijadikan sebagai referensi 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 2

### 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 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  $x$ .

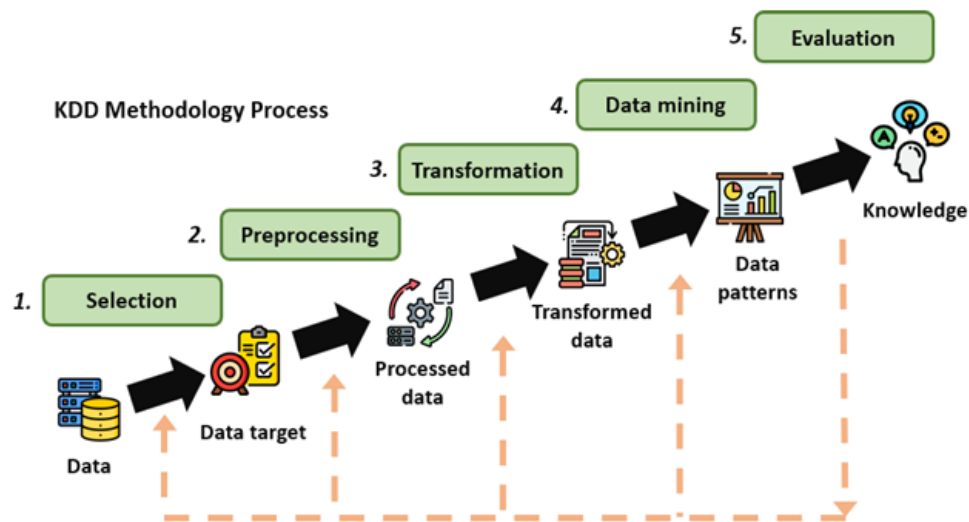


Pendekatan ini berfokus langsung pada penentuan batas keputusan antar kelas (Géron, 2020).

### **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, & Juszczuk, 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, dipelopori oleh 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, dan
- 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-

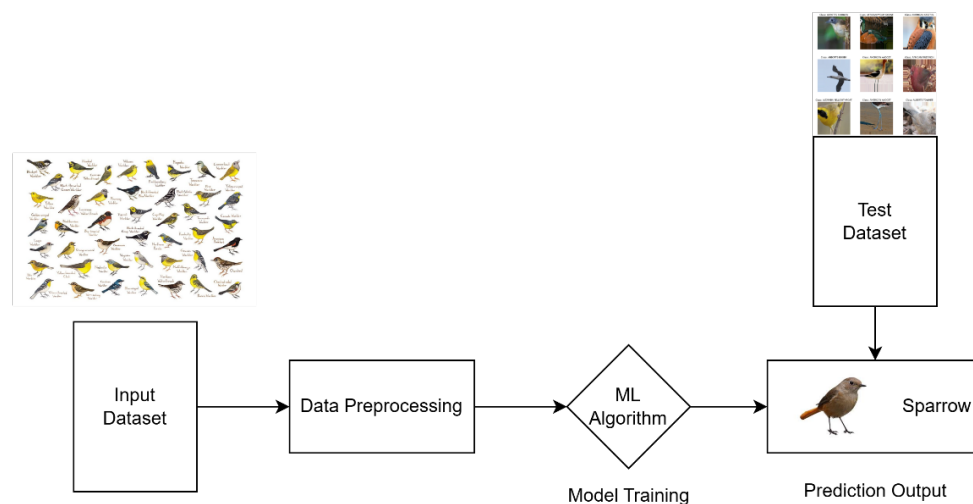
tugas terkait. Dalam konteks ini, ML memungkinkan sistem komputer untuk mengidentifikasi pola dalam data tanpa campur tangan manual yang intensif, sehingga memungkinkan solusi otomatis terhadap berbagai masalah kompleks berbasis data (Janiesch *et al.*, 2021).

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*,  $\rho_m$  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  $\alpha$  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  $\alpha$  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 dikembangkan 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).

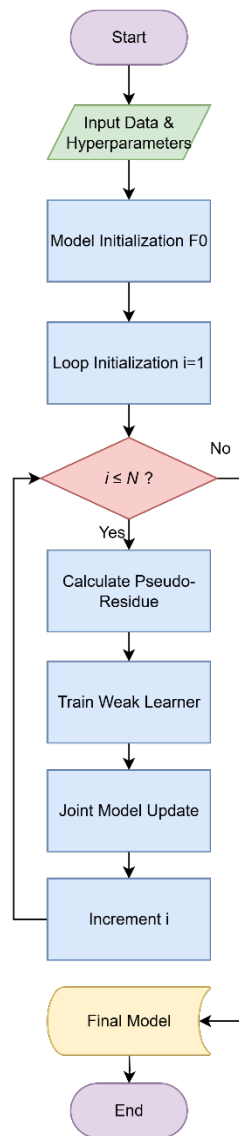
$$r_{mi} = \left[ \frac{\partial L(y_i, F(x))}{\partial L(x)} \right]_{F(x)=F_{m-1}(x)} \quad (2.5)$$

Nilai dari  $\rho_m$  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 *tree 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.

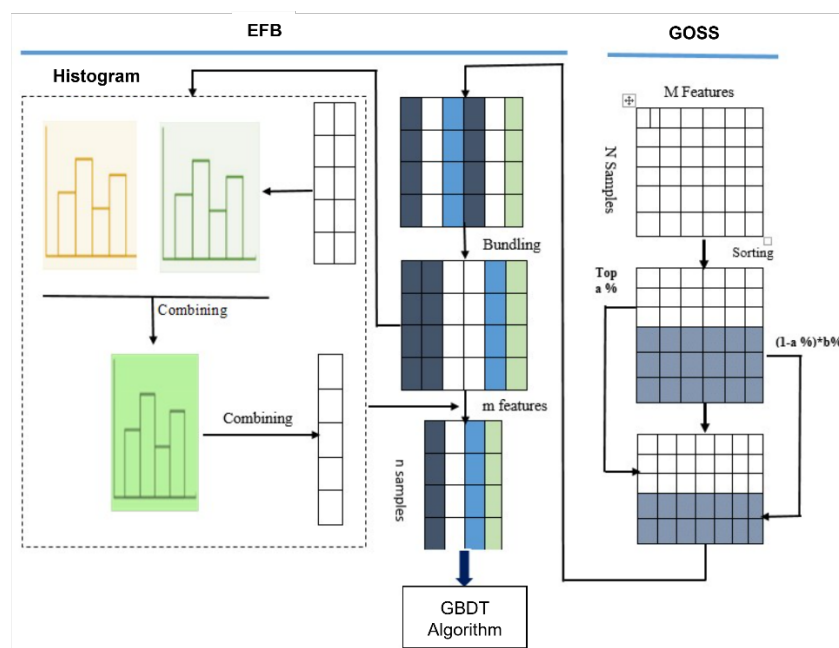
Beberapa parameter dalam *gradient boosting*, seperti jumlah *node*, kedalaman maksimum, dan tingkat pembelajaran, dapat disesuaikan berdasarkan kinerja model pada *testing* set (Hu *et al.*, 2023). Pengaturan parameter ini penting untuk memastikan model tidak hanya memberikan prediksi yang akurat, tetapi 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 (Mohammed *et al.*, 2021).



Gambar 2.4 Arsitektur GOSS dan EFB

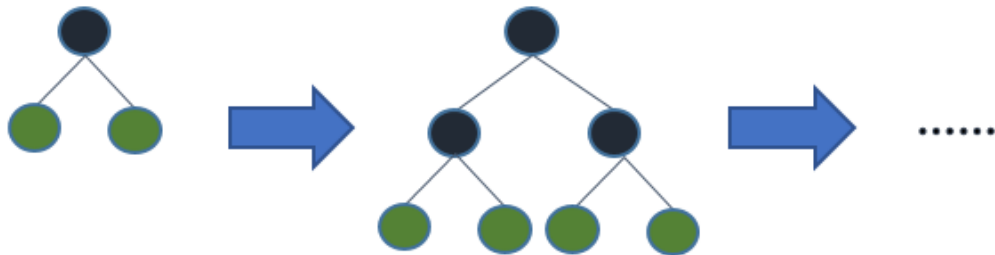
Pada Gambar 2.4 disajikan arsitektur dan aliran data dalam kerangka kerja GBDT pada LightGBM yang ditingkatkan, mengintegrasikan teknik EFB dan GOSS. EFB bertujuan untuk mengurangi dimensi fitur dengan menggabungkan fitur-fitur yang jarang aktif bersamaan ke dalam bundel tunggal, sehingga



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.

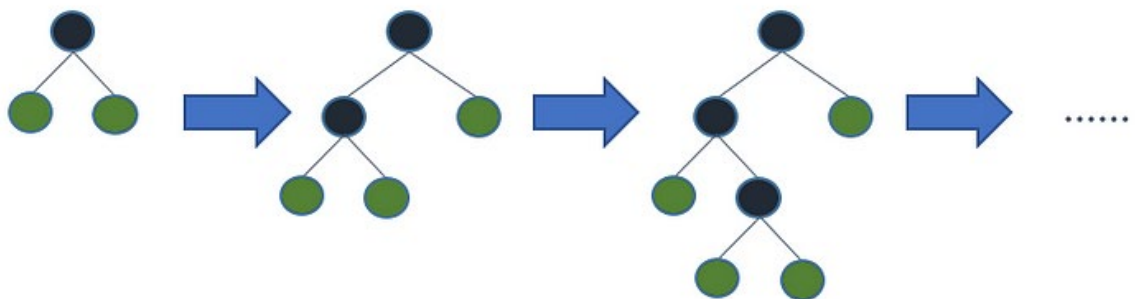
Sementara itu, GOSS mengatasi tantangan jumlah sampel yang besar dengan secara selektif mempertahankan instansi berdasarkan gradiennya. Sampel dengan gradien besar (*top  $\alpha\%$* ) dipertahankan secara utuh karena mereka berkontribusi paling signifikan terhadap *error* model, sedangkan sampel dengan 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 *tree* (*depth-wise*). Dalam metode tradisional ini, semua *node* pada tingkat yang sama dianggap sama pentingnya, dan *tree* 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)

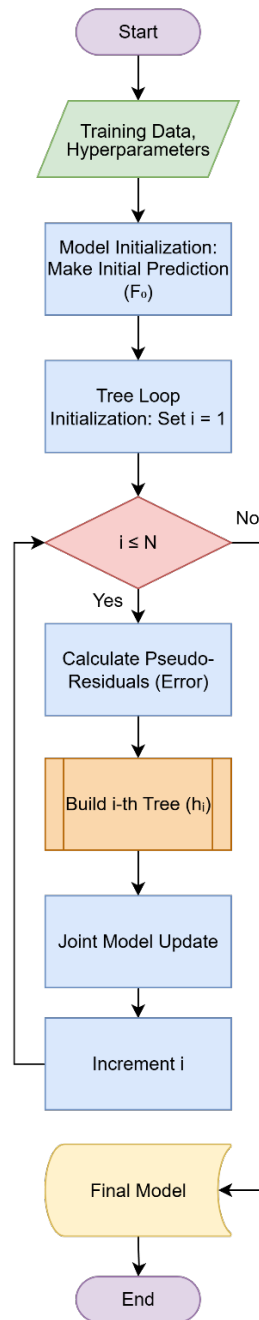
Namun, LightGBM mengadopsi strategi pertumbuhan *tree* berbasis *leaf* atau *leaf-wise*, yang hanya membagi *leaf* yang diharapkan memberikan peningkatan terbesar terhadap akurasi model, seperti pada Gambar 2.6. Dengan fokus pada *leaf* yang paling berpotensi untuk meningkatkan performa model, LightGBM membangun *tree* 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 *tree* 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 *leaf* 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 *tree* 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 *leaf* ini adalah dalam hal kecepatan dan akurasi (LightGBM, 2024). Dari segi kecepatan, LightGBM menjadi sangat efisien karena metode *leaf-wise* hanya membagi *leaf* yang memberikan dampak signifikan pada model, sehingga menghindari pengembangan sub-*tree* 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.

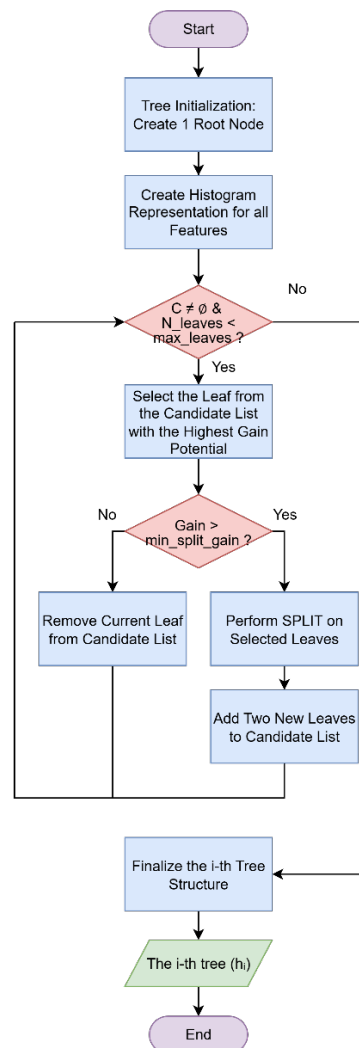


Gambar 2.7 Flowchart LightGBM (Ke *et al.*, 2017)

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 *tree* ( $N$ ) 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 *decision tree*. Setelah *decision tree* yang baru berhasil dibangun, model gabungan diperbaharui dengan menambahkan kontribusi dari *tree* 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 *decision tree* secara detail disajikan pada Gambar 2.8.



Gambar 2.8 *Flowchart* Sub-Proses Membangun Satu *Decision tree*

Gambar 2.8 menyajikan diagram alur kerja untuk sub-proses pembangunan satu *decision tree*, yang merupakan langkah detail dari alur utama LightGBM. Proses ini diawali dengan inisialisasi *tree* 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 *tree* secara *leaf-wise*. Pada setiap iterasi, dari semua *leaf* yang menjadi kandidat, algoritma akan memilih satu *leaf* yang memiliki potensi *information gain* tertinggi. Setelah kandidat *leaf* terbaik ditentukan, dilakukan pengecekan apakah nilai gain dari pemecahan (*split*) tersebut melebihi ambang batas *min\_split\_gain*. Jika gain mencukupi, *leaf* tersebut akan dipecah menjadi dua *leaf* baru, dan keduanya ditambahkan ke dalam daftar kandidat untuk iterasi selanjutnya. Sebaliknya, jika gain tidak mencukupi, *leaf* tersebut akan dihapus dari daftar kandidat. Perulangan ini terus berjalan hingga tidak ada lagi kandidat yang bisa dipecah atau jumlah *leaf* telah mencapai batas *max\_leaves*, kemudian struktur *tree* 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)

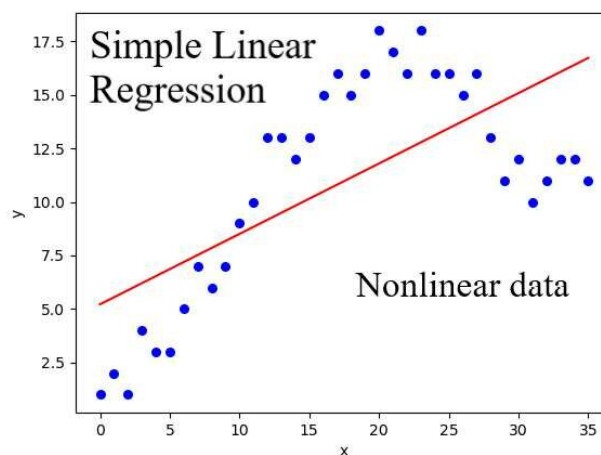
Usia Beton (Hari)	Kuat Tekan Rata-rata (MPa)
3	28,52
7	44,75
14	59,86
28	79,99
56	85,23
90	88,33
100	89,54

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*): 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*): Data 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*): 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

Peneliti	Sumber Dataset	Algoritma	Metrik Kinerja
(Pollard & Reep, 1997)	22 Pertandingan (manual)	<i>Logistic Regression</i>	-
(Ensum, Pollard, & Taylor, 2005)	48 Pertandingan (manual)	<i>Logistic Regression</i>	-
(Ruiz et al., 2015)	Prozone (EPL 2013/14)	<i>Multilayer Perceptron</i> (MLP)	p-value < 0.01
(Eggels, 2016)	ORTEC & Inmotio	<i>AdaBoost, Random Forest</i> , dll.	ROC AUC: 0.84 (AdaBoost)
(Cavus & Biecek, 2022)	Understat	LightGBM, XGBoost, dll. (AutoML)	ROC AUC: 0.875 (Random Forest)
(Iapteff et al., 2025)	StatsBomb (2003–22)	Bayesian GLMM + Transfer Learning	ROC AUC: 0.781

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 *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 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, sering kali dibangun di atas variabel-variabel yang 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 mengenai kinerja 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 kinerja 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 (Petropoulos *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  $2 \times 2$ . 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 (TP)</i>	<i>False Negative (FN)</i>
Aktual Negatif	<i>False Positive (FP)</i>	<i>True Negative (TN)</i>

#### 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).

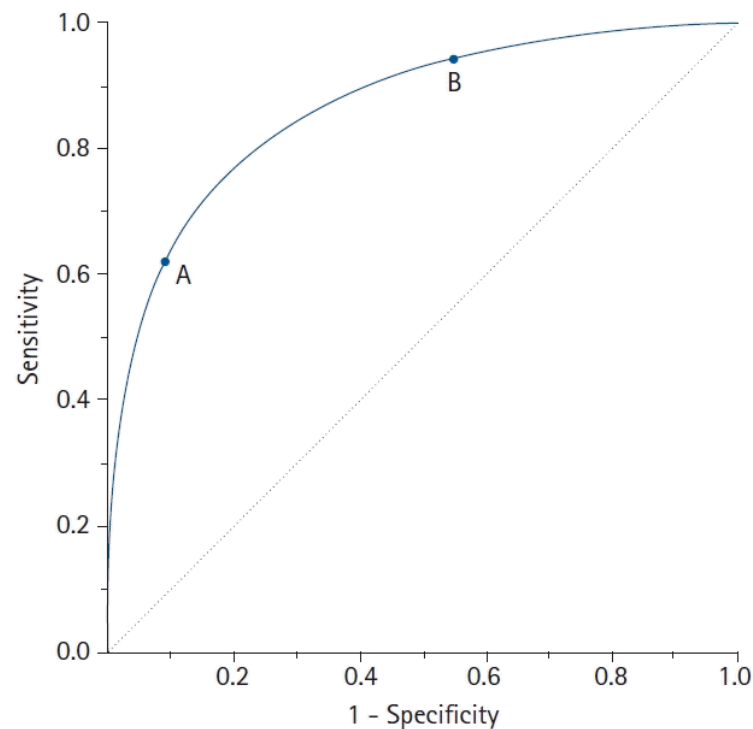
$$TPR = \frac{TP}{TP + FN} \quad (2.8)$$

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

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 - 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  $45^\circ$  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 *Isotonic Regression*

Regresi isotonik (*isotonic regression*) adalah sebuah metode regresi non-parametrik yang digunakan untuk menyesuaikan (*fit*) sebuah fungsi non-penurunan (*non-decreasing*) pada suatu urutan data. Berbeda dengan regresi linear yang



memaksakan data ke dalam bentuk garis lurus, regresi isotonik tidak membuat asumsi apapun mengenai bentuk fungsi yang mendasarinya, selain dari sifat monotoniknya (yaitu, nilainya tidak pernah menurun) (Henzi et al., 2021).

Dalam *machine learning*, regresi isotonik sangat efektif digunakan sebagai metode kalibrasi probabilitas. Tujuannya adalah untuk mengubah probabilitas mentah yang dihasilkan oleh sebuah model klasifikasi agar lebih sesuai dengan distribusi probabilitas yang sebenarnya. Dengan kata lain, regresi isotonik "memperbaiki" prediksi probabilitas model agar lebih andal dan tidak bias, tanpa mengubah kemampuan model dalam membedakan antar kelas (Gupta & Ramdas, 2021). Fleksibilitasnya dalam menangkap hubungan yang kompleks tanpa memerlukan asumsi parametrik menjadikannya pilihan yang unggul untuk kalibrasi model non-linear.

## 2.16 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).

## 2.17 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).

### 2.18 *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).

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

### 2.19 *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).

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

## 2.20 *Log-Loss (Cross-Entropy Loss)*

*Log-Loss*, yang secara formal dikenal sebagai *cross-entropy loss*, adalah metrik evaluasi 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 ( $\Sigma$ ). 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.

## 2.21 *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.

## **2.22 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.

### **2.22.1 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*.

### **2.22.2 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*.

### 2.22.3 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).

#### 2.22.4 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.

#### 2.22.5 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.

### **2.23 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"> <li>Persamaan probabilitas gol (regresi logistik)</li> <li>Koefisien Jarak (<math>X</math>): -0.096 (semakin jauh, peluang turun)</li> <li>Koefisien Sudut (<math>A</math>): -1.037 (semakin menyamping, peluang turun)</li> </ul>
2	Ensum, Pollard, & Taylor (2005)	Pengembangan model probabilitas gol melalui analisis multivariat kontekstual	Regresi Logistik	48 pertandingan, 1.099 tembakan ( <i>FA Premier League &amp; World Cup</i> )	Konfirmasi dan perluasan temuan awal dengan <i>dataset</i> yang lebih besar.	<ul style="list-style-type: none"> <li>Persamaan probabilitas gol (regresi logistik)</li> <li>Koefisien Jarak: -0.16</li> <li>Koefisien Sudut: -1.24</li> <li>Koefisien Sundulan: -0.73</li> <li>Koefisien Kaki Lemah: -0.63</li> <li>Koefisien Tendangan Voli: -0.27 (Semua koefisien negatif menunjukkan penurunan probabilitas gol dibandingkan kondisi ideal).</li> </ul>
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"> <li>AUC Model <i>Baseline</i> (Hanya Lokasi): 0.75</li> <li>AUC Model <i>Spatiotemporal</i> (EGV): 0.81</li> </ul>

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"> <li>• Tingkat Signifikansi Sangat Tinggi (p 0.01)</li> <li>• Tingkat Signifikansi Tinggi (p 0.05)</li> </ul> <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"> <li>• Regresi Logistik</li> <li>• <i>Decision Tree</i></li> <li>• <i>Random Forest</i></li> <li>• AdaBoost</li> <li>• Python (scikit-learn)</li> </ul>	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  <i>Random Forest</i> = 0,82            Regresi Logistik = 0,78  <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"> <li>• Regresi Logistik</li> <li>• Python</li> <li>• SciPy/Statsmodels</li> </ul>	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"> <li>• Regresi Logistik</li> <li>• XGBoost</li> <li>• ANN</li> <li>• <i>scikit-learn</i></li> <li>• Keras</li> </ul>	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"> <li>• ANN = 0,88</li> <li>• XGBoost = 0,85</li> <li>• Regresi Logistik = 0,78.</li> </ul> <p>RMSE:</p> <ul style="list-style-type: none"> <li>• ANN = 0,25</li> <li>• XGBoost = 0,27</li> <li>• Regresi Logistik = 0,32.</li> </ul>



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"> <li>Model probabilistik parametrik</li> <li>Python (SciPy <i>optimize</i>)</li> </ul>	>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"> <li>• <i>Log-Score</i> Model Mereka: -0.312</li> <li>• <i>Log-Score Baseline Naive</i>: -0.342</li> <li>• Peningkatan vs. <i>Baseline Naive</i>: +0.030</li> <li>• Peningkatan vs. <i>Baseline Pollard &amp; Reep</i>: +0.004</li> </ul>
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"> <li>• ROC AUC (<i>Random Forest</i>): 0.875</li> <li>• <i>Brier Score</i> (<i>Random Forest</i>): 0.072</li> </ul> Performa model lain: <ul style="list-style-type: none"> <li>• AUC (LightGBM): 0.873</li> <li>• AUC (CatBoost): 0.872</li> <li>• AUC (XGBoost): 0.869</li> </ul>
10	Méndez <i>et al.</i> (2023)	Peningkatan nilai xG dengan jaringan saraf	<ul style="list-style-type: none"> <li><i>Multilayer Perceptron</i> (MLP)</li> <li>Python</li> <li>Keras</li> </ul>	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"> <li>• ROC AUC: 0.87 (MLP) vs. 0.82 (LR)</li> <li>• Akurasi: 90.04% (MLP) vs. 89.28% (LR)</li> <li>• <i>Brier Score</i>: 0.076 (MLP) vs. 0.081 (LR)</li> </ul>

No	Penulis	Domain Riset	Metode dan <i>Tools</i>	<i>Dataset</i> (Populasi, Sampel)	Kontribusi	Hasil
						<ul style="list-style-type: none"> <li>• Log-Loss: 0.262 (MLP) vs. 0.280 (LR)</li> <li>• F1-Score: 0.41 (MLP) vs. 0.34 (LR)</li> </ul> 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"> <li>• Regresi Logistik,</li> <li>• RF,</li> <li>• AdaBoost,</li> <li>• XGBoost,</li> <li>• Python (scikit-learn)</li> </ul>	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"> <li>• <i>Random Forest</i> (Fitur Diperkaya): 0.910</li> <li>• <i>Random Forest</i> (Fitur Dasar): 0.891</li> <li>• Regresi Logistik (Fitur Dasar): 0.852</li> </ul>
12	Hewitt & Karakuş (2023)	Pemodelan xG kontekstual berbasis identitas dan peran pemain	<ul style="list-style-type: none"> <li>• <i>Regresi Logistik</i>,</li> <li>• <i>Gradient Boosted Trees (GBT)</i>,</li> <li>• <i>scikit-learn</i>,</li> </ul>	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"> <li>• Korelasi Pearson antara xG dan gol lebih tinggi pada GBT (0,208) dibandingkan Regresi Logistik (0,188).</li> <li>• Analisis pada 347 tembakan Lionel Messi menunjukkan model ini memberikan nilai xG +3,74 lebih tinggi daripada model dasar.</li> </ul>
13	Bandara <i>et al.</i> (2024)	Pemodelan xG sekuensial berbasis aliran <i>event</i> pra-tembakan	<ul style="list-style-type: none"> <li>• <i>Random Forest</i> (100 <i>estimators</i>),</li> <li>• scikit-learn</li> </ul>	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"> <li>• Model berbasis sekuens mencapai ROC AUC 0,833 pada set validasi.</li> <li>• Pada pengujian dengan data Euro 2020, model</li> </ul>

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"> <li>• Regresi Logistik</li> <li>• R (fungsi glm)</li> </ul>	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"> <li>• ROC AUC = 0,812</li> <li>• <i>Brier Score</i> = 0,078</li> </ul> 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"> <li>• Jaringan Saraf Konvolusional (CNN)</li> <li>• Regresi Logistik</li> <li>• Analisis <i>Pose</i> (OpenPose)</li> </ul>	<ul style="list-style-type: none"> <li>• Data Publik: 477 tembakan dari <i>dataset</i> publik.</li> <li>• Data SoccerNet-v2: 927 tembakan dari 500 pertandingan.</li> </ul>	<ul style="list-style-type: none"> <li>• Pionir dalam menggunakan data pose/kerangka tubuh pemain (<i>skeleton</i> data) secara langsung untuk estimasi xG.</li> <li>• Memperkenalkan model Skor-xG yang mengintegrasikan orientasi tubuh pemain dan posisi kiper.</li> </ul>	<ul style="list-style-type: none"> <li>• ROC AUC: Model Skor-xG mencapai 0,845, mengungguli model Regresi Logistik (0,791).</li> <li>• <i>Brier Score</i>: Skor-xG mendapatkan 0,068, lebih baik dari Regresi Logistik (0,075). (Nilai lebih rendah lebih baik).</li> </ul>

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.

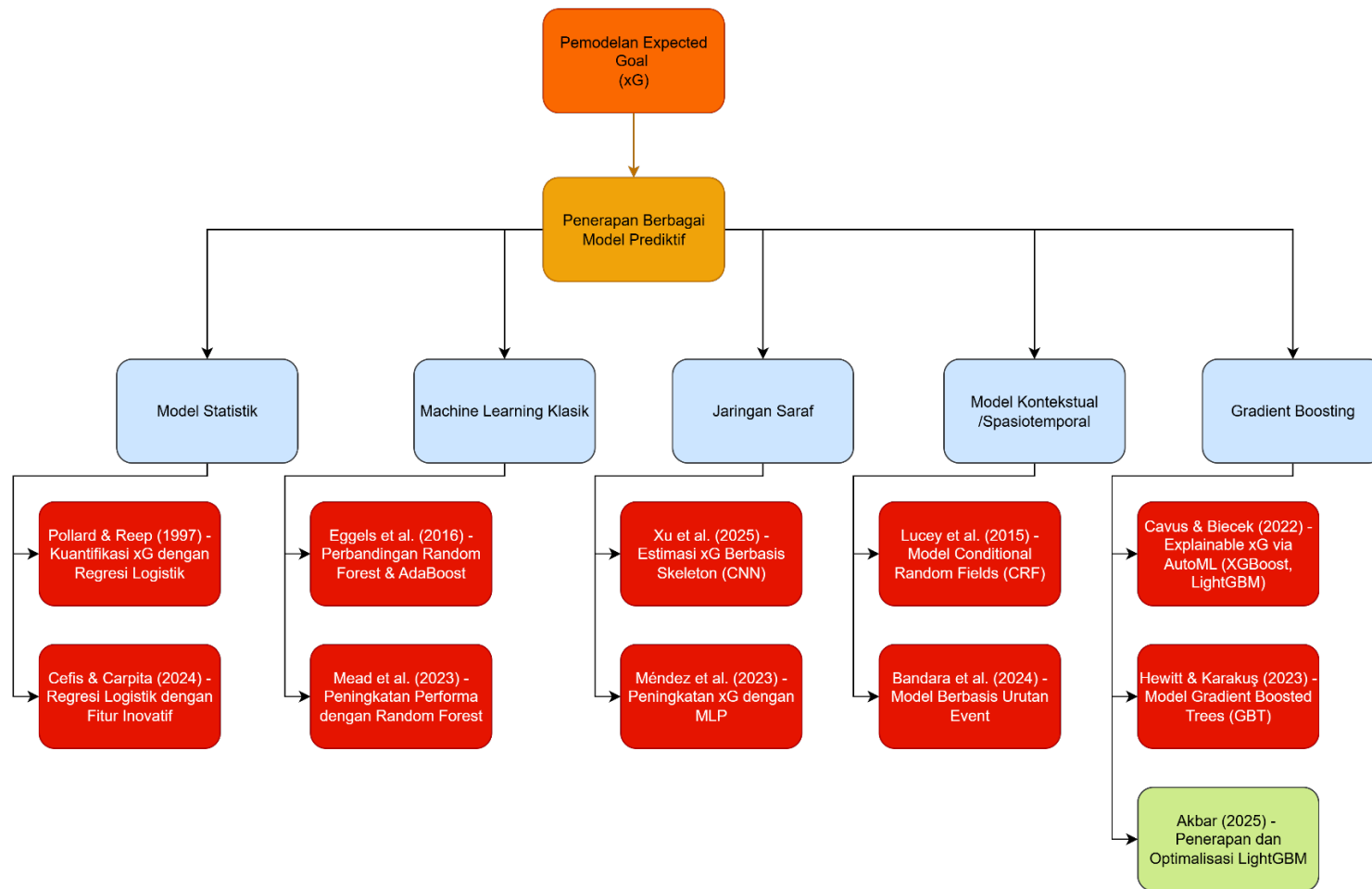
Lebih lanjut, terdapat penelitian yang mengembangkan model dengan fitur-fitur yang lebih canggih dan kontekstual. Lucey *et al.* (2015) menerapkan *Conditional Random Fields* untuk memasukkan informasi spasiotemporal, sementara Bandara *et al.* (2024) menggunakan *Random Forest* dengan fitur sekuensial 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.

## 2.24 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 3**

### **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 73.844 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).

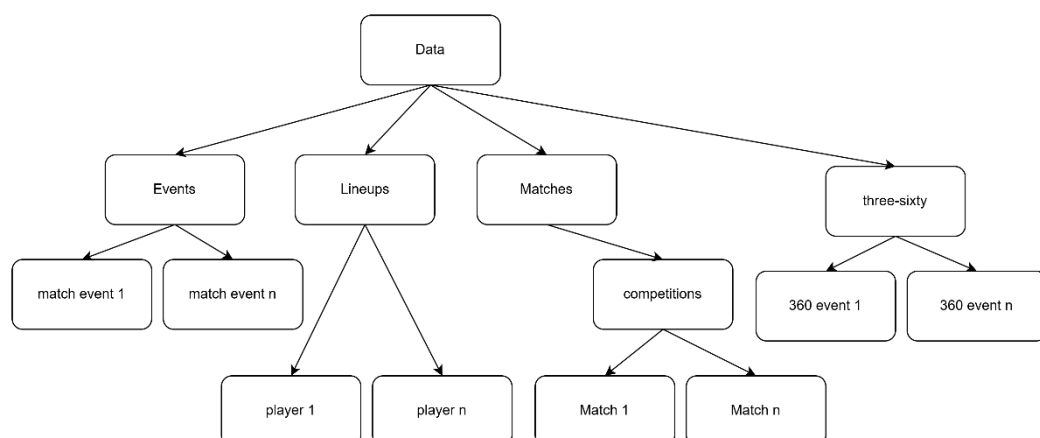
##### 3.5.1.1 *Data Selection*

Tahap *data selection* merupakan fase awal dan strategis dalam metodologi KDD yang diterapkan pada penelitian ini. Fokus utama dari tahap ini adalah untuk mengidentifikasi, memahami, dan secara teknis mengumpulkan data mentah yang

relevan dari sumber yang telah ditentukan. Proses ini memastikan bahwa data yang akan digunakan pada tahap selanjutnya benar-benar sesuai dengan tujuan penelitian. Tahapan ini terdiri dari dua langkah berurutan: *data understanding* dan *data acquisition*.

a. *Data understanding*

*Data understanding* yaitu proses eksplorasi untuk membangun pemahaman mendasar terhadap sumber data. Dalam penelitian ini, proses tersebut melibatkan penelaahan dokumentasi resmi dan struktur dari repositori StatsBomb Open Data di GitHub. Tujuannya adalah untuk memahami hierarki data (mulai dari kompetisi hingga *event* individual) , format data (JSON), serta cakupan dan makna dari setiap atribut yang tersedia. Pemahaman ini menjadi landasan untuk menentukan kelayakan data dan merumuskan kriteria spesifik untuk pengumpulan. Pada Gambar 3.1 dijelaskan struktur data yang dimiliki oleh StatsBomb *open-data*.



Gambar 3.1 Struktur Data StatsBomb *open-data*.

b. *Data acquisition*

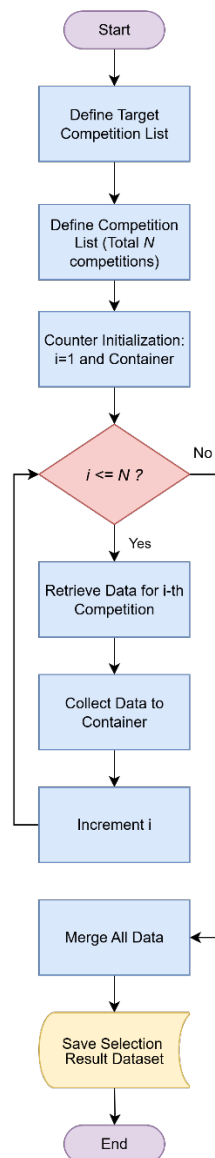
*Data acquisition* adalah tahap eksekusi teknis untuk mengumpulkan data mentah berdasarkan pemahaman yang telah diperoleh. Pengumpulan data dilakukan dengan mengunduh dataset dari repositori GitHub. Hasil akhir dari keseluruhan tahap *data selection* ini adalah repositori *master* dari StatsBomb *open-data* yang berisi kumpulan data mentah dalam format JSON, yang kemudian menjadi masukan untuk tahap *data preprocessing*. 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", ... }]

<i>Event</i>	<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. Secara detail, Gambar 3.2 menunjukkan *flowchart* dari *data selection*.



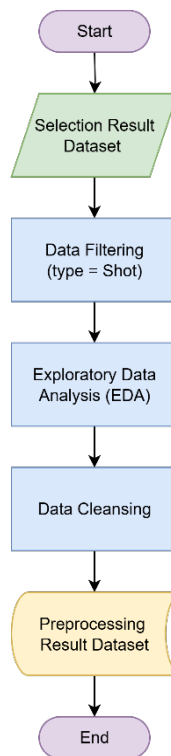
Gambar 3.2 Flowchart Data Selection

### 3.5.1.2 Data Preprocessing

Tahap *data preprocessing* merupakan langkah esensial yang bertujuan untuk mengubah data mentah (*raw data*) yang telah dikumpulkan menjadi *dataset* yang bersih, konsisten, dan optimal untuk dianalisis serta digunakan dalam pemodelan. Tahap ini terdiri dari tiga langkah utama yang berurutan: *event data filtering*,



*Exploratory Data Analysis* (EDA), dan *data cleansing*. 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

a. *Event data filtering*

Pada langkah ini, data mentah yang berisi semua jenis peristiwa (*event*) pertandingan disaring secara ketat untuk mengambil hanya baris data yang relevan dengan tujuan penelitian. Secara spesifik, *dataset* difilter untuk hanya menyimpan peristiwa dengan tipe '*Shot*' (*type.id* = 16) yang berasal dari situasi permainan terbuka (*open play*). Tindakan ini sangat penting untuk

memfokuskan analisis hanya pada data yang paling relevan dengan konteks prediksi xG dan secara signifikan mengurangi volume data yang perlu diolah.

b. *Exploratory Data Analysis* (EDA)

Pada tahap ini, dilakukan analisis mendalam untuk memahami karakteristik dan distribusi data tembakan pada data StatsBomb. Melalui teknik statistik deskriptif dan visualisasi, diidentifikasi pola-pola penting, anomali, serta hubungan antar variabel. EDA juga membantu dalam seleksi awal fitur-fitur yang paling berpotensi mendukung prediksi, seperti posisi tembakan dan bagian tubuh yang digunakan, sementara kolom yang tidak relevan dapat diidentifikasi untuk dihilangkan.

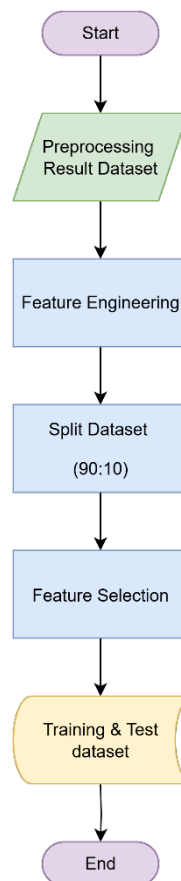
c. *Data cleansing*

Tahap ini meliputi pemeriksaan dan penanganan nilai yang hilang (*missing values*) atau data duplikat untuk memastikan integritas dan kualitas data. Sesuai dengan hasil analisis pada data StatsBomb, ditemukan bahwa *dataset* yang digunakan umumnya sudah sangat terstruktur dan bersih, sehingga tidak memerlukan intervensi pembersihan yang ekstensif. Keseluruhan proses *preprocessing* ini menghasilkan sebuah *dataset* yang konsisten dan siap untuk dilanjutkan ke tahap *data transformation*.

### 3.5.1.3 *Data Transformation*

Setelah *dataset* bersih diperoleh dari tahap *preprocessing*, penelitian dilanjutkan ke tahap *data transformation*. Tahap ini bertujuan untuk memperkaya dan mengubah representasi data agar lebih informatif dan sesuai dengan format yang dibutuhkan oleh algoritma *machine learning*. Proses ini sangat krusial untuk

meningkatkan performa model pada tahapan *data mining*. Tahap ini terdiri dari tiga langkah utama: *feature engineering*, *feature selection*, dan *split dataset*. Alur tahapan *transformation* ditunjukkan pada Gambar 3.4.



Gambar 3.4 *Flowchart Data Transformation*

a. *Feature engineering*

Proses ini berfokus pada penciptaan fitur-fitur baru (*derived features*) dari data yang sudah ada untuk menangkap dinamika permainan secara lebih mendalam. Berdasarkan analisis dan studi literatur, tiga fitur kunci direkayasa, yaitu *distance\_to\_goal* (jarak tembakan ke gawang), *angle\_to\_goal* (sudut tembakan ke gawang), dan *type\_before* (jenis peristiwa yang terjadi tepat sebelum

tembakan). Penambahan fitur-fitur ini bertujuan untuk memperkaya informasi spasial dan kontekstual yang tidak tersedia secara eksplisit pada data mentah, yang terbukti krusial untuk meningkatkan daya prediksi model.

b. *Split dataset*

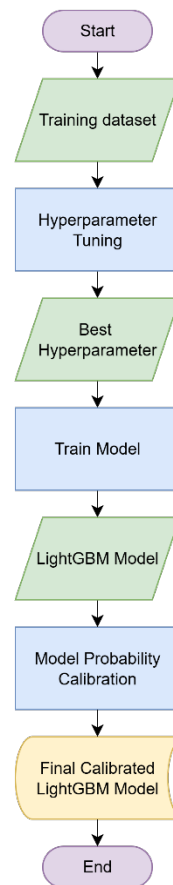
Pada tahap ini, *dataset* yang telah final dibagi menjadi dua bagian terpisah: data latih (*training set*) dan data uji (*test set*). Pembagian dilakukan dengan proporsi 90% untuk data latih dan 10% untuk data uji. Data latih akan digunakan sepenuhnya untuk proses pencarian *hyperparameter* dan pelatihan model, sementara data uji akan disimpan dan hanya digunakan sekali pada tahap evaluasi akhir untuk mengukur kinerja generalisasi model pada data yang belum pernah dilihat sebelumnya.

c. *Feature selection*

Pada tahap ini, dilakukan pemilihan himpunan fitur final yang akan digunakan untuk melatih model menjadi variabel  $X$  dan  $y$ . Berdasarkan analisis EDA dan relevansi teoritis, semua fitur yang telah diproses baik fitur asli maupun hasil rekayasa diputuskan untuk dipertahankan dalam penelitian ini. Keputusan ini diambil untuk menyediakan informasi selengkap mungkin kepada model, dengan asumsi bahwa setiap fitur memberikan konteks unik yang berharga.

#### 3.5.1.4 *Data Mining*

Setelah data melewati fase *transformation* dan siap digunakan, proses ini dilanjutkan ke tahap *data mining*. Fase ini berfokus pada pembangunan, penyetelan, dan finalisasi model prediktif LightGBM. Gambar 3.5 menunjukkan alur dari tahapan *data mining* dalam penelitian ini.

Gambar 3.5 *Flowchart Data Mining*

a. *Hyperparameter tuning*

Sebuah proses krusial karena performa model *machine learning* sangat sensitif terhadap konfigurasi awalnya, dan setelan *default* jarang memberikan hasil optimal untuk *dataset* yang spesifik. Tujuannya adalah untuk menemukan kombinasi *hyperparameter* terbaik yang memungkinkan algoritma LightGBM untuk memaksimalkan daya prediksinya dalam menangkap pola non-linear yang kompleks pada data tembakan. Untuk mencapai ini, penelitian menggunakan metode *RandomizedSearchCV* yang secara efisien mengeksplorasi 100 kombinasi parameter acak dari ruang pencarian yang telah

ditentukan. Setiap kombinasi dievaluasi menggunakan skema *5-fold cross-validation* yang berjalan pada data latih untuk memastikan estimasi performa yang dihasilkan bersifat stabil dan tidak bias terhadap pembagian data tertentu. Metrik ROC AUC dipilih sebagai acuan skor utama karena kemampuannya yang unggul dalam mengevaluasi model pada kasus klasifikasi dengan kelas yang tidak seimbang (*imbalanced*) seperti prediksi gol, di mana metrik akurasi bisa sangat menyesatkan (Tharwat, 2021).

b. *Model training*

Setelah konfigurasi *hyperparameter* terbaik berhasil ditemukan, tahap selanjutnya adalah melatih model final. Model ini akan dilatih ulang dari awal menggunakan seluruh data latih dengan konfigurasi *hyperparameter* yang sudah terpilih tadi. Tujuan pelatihan ini yaitu memastikan model belajar dari semua data yang ada. Ini berbeda dengan tahap *tuning* yang hanya menggunakan sebagian data untuk membandingkan banyak model. Dengan cara ini, model final yang terbentuk diharapkan bisa bekerja dengan baik pada data baru.

c. *Model probability calibration*

Meskipun sebuah model dengan skor ROC AUC tinggi mampu membedakan antar kelas secara efektif, nilai probabilitas yang dihasilkannya belum tentu akurat. Dalam konteks xG, di mana angka probabilitas itu sendiri adalah *output* utamanya, masalah ini menjadi sangat penting sehingga diperlukan proses kalibrasi. Kalibrasi pada dasarnya adalah proses untuk mengoreksi probabilitas model agar lebih cocok dengan kondisi di dunia nyata. Sederhananya, jika

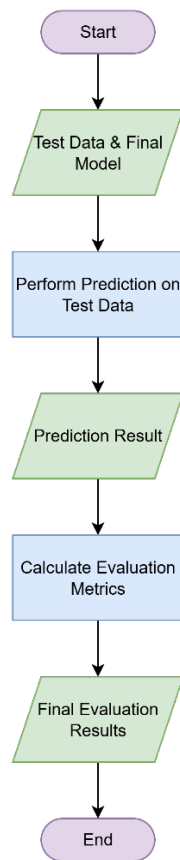
model memprediksi xG sebesar 0.3, kita ingin peluang kejadiannya benar-benar sekitar 30% (Davis & Robberechts, 2024). Untuk itu, penelitian ini akan melakukan kalibrasi menggunakan `CalibratedClassifierCV` dengan metode *isotonic regression*, yang dipilih karena fleksibel dalam menyesuaikan probabilitas prediksi dengan hasil yang sebenarnya, bahkan untuk hubungan yang kompleks sekalipun. Evaluasi keberhasilan kalibrasi akan dilakukan dengan memvisualisasikan kurva kalibrasi, di mana metodologi pembuatannya adalah dengan mengelompokkan semua prediksi ke dalam beberapa *bins* dan membandingkan dua metrik utama. Sumbu  $X$  (*mean predicted value*) adalah seberapa besar keyakinan rata-rata model bahwa tembakan di dalam bin tersebut akan menjadi gol. Misalnya, jika rata-rata prediksi probabilitas untuk semua tembakan dalam satu bin adalah 0.1, maka nilai *mean predicted value* untuk bin tersebut adalah 0.1 (atau 10%). Di sisi lain, Sumbu  $Y$  (*fraction of positives*) adalah seberapa sering tembakan di dalam bin yang sama benar-benar menghasilkan gol pada kenyataannya. Misalnya, jika dari 100 tembakan di dalam bin yang sama ternyata ada 8 yang benar-benar menjadi gol, maka *fraction of positives* adalah 0.08 (atau 8%). Dengan demikian, model yang terkalibrasi sempurna akan memiliki titik-titik plot yang berada sangat dekat dengan garis diagonal (garis kalibrasi sempurna).

#### **3.5.1.5 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).

#### 3.5.2.1 GOSS

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.

### 3.5.2.2 EFB

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 Interpretasi Hasil

Setelah kinerja model diukur secara kuantitatif pada tahap evaluasi, interpretasi terhadap hasil akan dilakukan untuk memberikan konteks yang

menyeluruh. Interpretasi akan ditinjau dari tiga perspektif utama yaitu *practical application*, *academic benchmark* dan *industrial benchmark*.

### **3.6.1 Practical Application**

Interpretasi praktis akan dilakukan melalui dua cara. Pertama, hasil prediksi model akan diterapkan pada studi kasus pertandingan nyata, yaitu pertandingan Inggris melawan Iran pada Piala Dunia 2022, untuk mendemonstrasikan utilitasnya dalam analisis taktis. Total xG, rata-rata xG per tembakan, dan visualisasi *shot map* akan dihasilkan dan dianalisis. Kedua, akan dikembangkan sebuah antarmuka aplikasi web interaktif menggunakan *framework* Streamlit. Metodologi pengembangan antarmuka ini bertujuan untuk mempermudah pengaplikasian prediksi xG oleh pengguna akhir tanpa memerlukan keahlian pemrograman, baik untuk prediksi berdasarkan *dataset* maupun simulasi tembakan kustom.

### **3.6.2 Academic Benchmark**

Untuk mengukur signifikansi penelitian dalam konteks akademis, kinerja model akan dibandingkan secara sistematis dengan hasil dari penelitian-penelitian sejenis yang relevan. Metodologi perbandingan ini melibatkan metrik kinerja utama (ROC AUC, Brier *Score*, akurasi, presisi, *recall*, F1-*Score*, dan Log-Loss) dari model ini dan model dari literatur yang telah dipublikasikan, khususnya yang menggunakan algoritma *boosting*.

### **3.6.3 Industrial Benchmark**

Untuk memvalidasi konsistensi model dengan standar industri, akan dilakukan perbandingan estimasi xG. Metodologi yang digunakan yaitu dengan memilih beberapa pertandingan tingkat tinggi (Piala Dunia 2022 dan EURO 2024),

kemudian membandingkan nilai xG yang dihasilkan oleh model penelitian ini dengan nilai xG yang dilaporkan oleh penyedia data statistik ternama seperti Opta dan FBref dan penyedia data statistik lain untuk pertandingan yang sama. Hasil dari seluruh tahapan analisis ini serta interpretasi mendalam terhadap kinerja model akan dijelaskan secara rinci pada Bab 4.

### **3.7 Implikasi Penelitian**

Setelah hasil dianalisis dan diinterpretasikan, implikasi dari penelitian ini akan dibahas untuk menjelaskan kontribusi dan dampak potensialnya. Pembahasan implikasi akan dibagi menjadi dua aspek utama.

#### **3.7.1 Implikasi Teoritis**

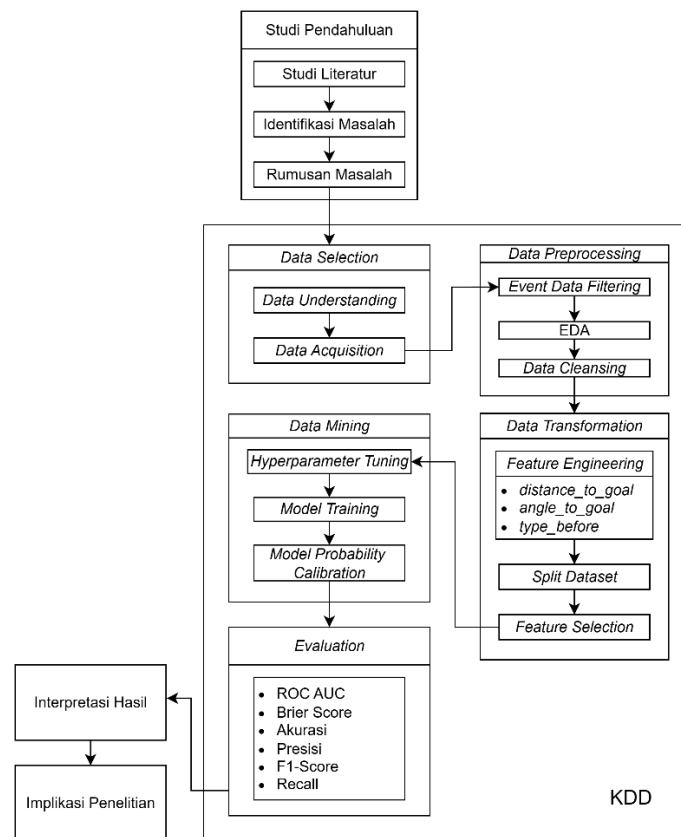
Pada bagian ini, akan dianalisis bagaimana hasil penelitian memberikan kontribusi terhadap khazanah ilmu pengetahuan, khususnya dalam bidang analisis data olahraga dan *machine learning*. Ini akan mencakup pembahasan mengenai validasi efektivitas algoritma LightGBM dalam konteks data sepak bola yang non-linear dan bagaimana temuan ini mendukung atau menantang teori-teori yang ada dalam pemodelan prediktif.

#### **3.7.2 Implikasi Praktis**

Bagian ini akan menguraikan bagaimana hasil penelitian dapat diterapkan dalam dunia nyata. Akan dijelaskan potensi pemanfaatan model xG yang dikembangkan bagi para pemangku kepentingan di industri sepak bola, seperti klub, analis performa, media, dan pencari bakat, dalam pengambilan keputusan strategis yang lebih berbasis data.

### 3.8 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*, akurasi, 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

### 3.9 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	September 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 4

### HASIL DAN PEMBAHASAN

#### 4.1 *Data Selection*

Hasil dari tahap seleksi data adalah *dataset* yang relevan untuk tujuan penelitian, yang diperoleh dari repositori *open-data* milik StatsBomb di GitHub.

##### 4.1.1 *Data Understanding*

Hasil dari tahap pemahaman data adalah identifikasi struktur, format, dan cakupan data yang tersedia melalui penelaahan dokumentasi resmi StatsBomb open-data (tersedia di <https://github.com/statsbomb/open-data/tree/master/doc>). Dari proses ini, dipahami bahwa data diorganisasi secara hierarkis (kompetisi, musim, pertandingan, *event*) dan tersedia dalam format JSON. Pemahaman ini memungkinkan identifikasi kompetisi pria tingkat senior yang relevan untuk diakuisisi. Tabel 4.1 menunjukkan daftar kompetisi yang berhasil diidentifikasi dari dokumentasi sebagai target untuk diakuisisi.

Tabel 4.1 Daftar Kompetisi

No.	Competition ID	Competition Name
1	11	FIFA World Cup
2	2	Premier League
3	37	La Liga
4	72	UEFA Champions League
5	43	Bundesliga
6	49	Serie A
7	4	Ligue 1
8	55	Copa America
9	9	African Cup of Nations
10	16	Eredivisie

#### 4.1.2 Data Acquisition

Berdasarkan pemahaman yang diperoleh, proses akuisisi data dilakukan dengan mengunduh seluruh data *event* dari 10 kompetisi yang telah diidentifikasi pada Tabel 4.1. Proses teknis akuisisi data menggunakan *tool* aria2 di Google Colab dengan bahasa pemrograman Python untuk efisiensi. Tahapan yang dilakukan adalah sebagai berikut:

- Mengunduh *file* master dari repositori GitHub StatsBomb open-data menggunakan perintah di Google Colab dan *tool* aria2 untuk mempercepat proses unduhan.
- Menyusun skrip Python untuk mengekstrak semua data *event* yang relevan dari *file* yang telah diunduh.
- Mengonversi data *event* yang diekstrak ke dalam format *dataframe* menggunakan pustaka pandas untuk mempermudah manipulasi dan analisis data.
- Menyimpan *dataframe* yang telah diproses dalam format *parquet* untuk efisiensi penyimpanan dan kecepatan akses pada tahap selanjutnya.

Hasil akhir dari tahap ini adalah kumpulan data mentah dalam format terstruktur yang siap untuk diproses pada tahap *preprocessing*.

```
!apt-get update -qq && apt-get install -qq aria2
!aria2c -x 16 -s 16 \
  https://github.com/statsbomb/open-
  data/archive/refs/heads/master.zip \
  -d . -o open-data.zip
!unzip -q open-data.zip 'open-data-master/data/*' -d open-data
!mv open-data/open-data-master/data open-data/data
!rm -rf open-data/open-data-master open-data.zip
with open(os.path.join(data_path, 'competitions.json'), 'r',
encoding='utf-8') as f:
```

```

    competitions = json.load(f)
    competitions_df = pd.DataFrame(competitions)
    competitions_df =
    competitions_df[competitions_df['competition_gender'] ==
    'male']
    youth_keywords = ['U21', 'U20', 'U23', 'U18', 'U19', 'U17',
    'U16', 'U15', 'U14', 'Youth']
    is_youth =
    competitions_df['competition_name'].str.contains('|'.join(youth_
    h_keywords), case=False, na=False)
    competitions_df = competitions_df[~is_youth]
    valid_competitions = competitions_df[['competition_id',
    'season_id']].drop_duplicates()
    valid_match_ids = set()
    for _, row in valid_competitions.iterrows():
        comp_id = row['competition_id']
        season_id = row['season_id']
        matches_file = os.path.join(data_path, 'matches',
    str(comp_id), f'{season_id}.json')
        if os.path.exists(matches_file):
            with open(matches_file, 'r', encoding='utf-8') as f:
                matches = json.load(f)
                for match in matches:
                    home_gender = match.get('home_team',
    {}).get('home_team_gender')
                    away_gender = match.get('away_team',
    {}).get('away_team_gender')
                    if home_gender == 'male' and away_gender ==
    'male':
                        valid_match_ids.add(match['match_id'])

```

## 4.2 Data Preprocessing

Tahap *preprocessing* bertujuan untuk membersihkan dan menyiapkan data agar siap digunakan dalam pemodelan. Proses ini memastikan kualitas data meningkat sehingga hasil pemodelan menjadi lebih akurat dan andal.

### 4.2.1 Event Data Filtering

Hasil dari tahap penyaringan adalah *dataset* yang difokuskan hanya pada peristiwa tembakan (*shot*). Data mentah yang berisi semua jenis *event* disaring untuk hanya menyimpan baris data dengan *type.id* = 16 (*Shot*) yang berasal dari situasi permainan terbuka (*open play*). Proses ini secara signifikan mengurangi volume data dan memfokuskan analisis hanya pada data yang paling relevan dengan konteks prediksi xG. Hasil akhir dari data tembakan ini ditunjukkan pada Gambar 4.1.

```
all_shots = []
for file_name in tqdm(event_files, desc='Processing event
files'):
    try:
        match_id = int(file_name.replace('.json', ''))
    except:
        continue
    if match_id not in valid_match_ids:
        continue
    file_path = os.path.join(events_path, file_name)
    with open(file_path, 'r', encoding='utf-8') as f:
        events = json.load(f)
        for idx, event in enumerate(events):
            if event.get('type', {}).get('id') == 16:
                all_shots.append(shot_data)
shot_df = pd.DataFrame(all_shots)
```

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

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 kolom 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.

```
location = event.get('location', [None, None])
    end_location = event.get('shot',
{ }).get('end_location', [None, None])
    x = location[0] if location and len(location)
> 0 else None
```

```

        y = location[1] if location and len(location)
> 1 else None
        end_x = end_location[0] if end_location and
len(end_location) > 0 else None
        end_y = end_location[1] if end_location and
len(end_location) > 1 else None
        shot_data = {
            'period': event.get('period'),
            'minute': event.get('minute'),
            'second': event.get('second'),
            'start_x': x,
            'start_y': y,
            'end_x': end_x,
            'end_y': end_y,
            'position' : event.get('position',
{ }).get('id'),
            'shot_type' : event.get('shot',
{ }).get('type', { }).get('id'),
            'shot_outcome': event.get('shot',
{ }).get('outcome', { }).get('id'),
            'shot_body_part':
event.get('shot', { }).get('body_part', { }).get('id'),
            'shot_first_time': 1 if
event.get('shot', { }).get('first_time') else 0,
            'shot_one_on_one': 1 if
event.get('shot', { }).get('one_on_one') else 0,
            'shot_open_goal': 1 if
event.get('shot', { }).get('open_goal') else 0,
            'shot_aerial_won': 1 if
event.get('shot', { }).get('aerial_won') else 0,
            'shot_key_pass': 1 if
event.get('shot', { }).get('key_pass_id') else 0,
            'play_pattern':
event.get('play_pattern', { }).get('id'),
            'under_pressure': 1 if
event.get('under_pressure') else 0,
            'shot_technique':
event.get('shot', { }).get('technique', { }).get('id'),
        }
        all_shots.append(shot_data)

```

	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

### 4.2.2 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*).

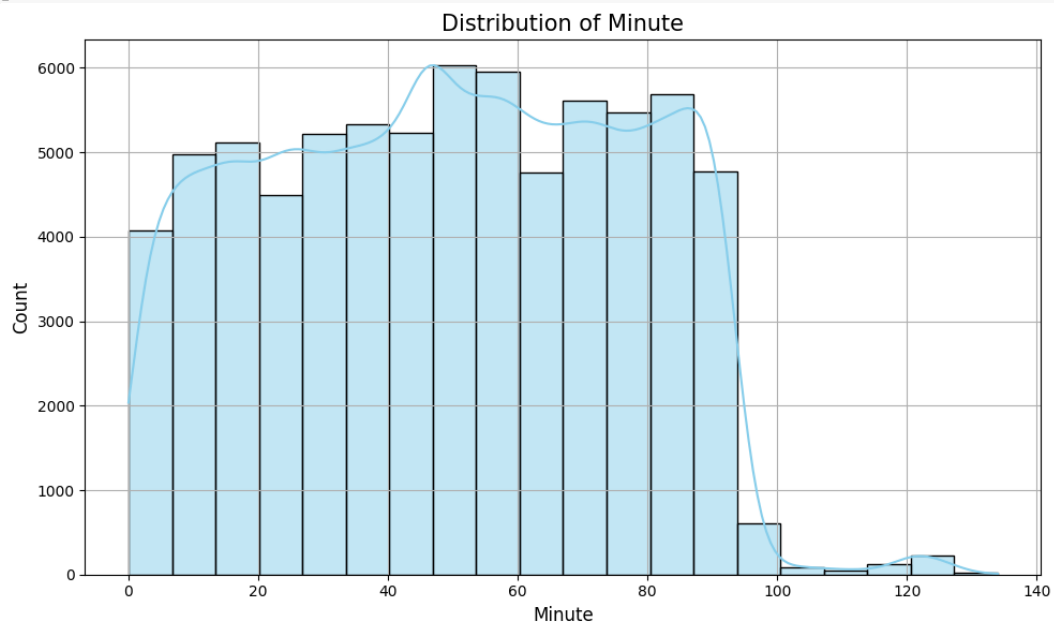
#### 4.2.2.1 Analisis Distribusi Variabel Tunggal (Univariat)

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.

```
plt.figure(figsize=(10, 6))
```



```
sns.histplot(df['minute'], bins=20, kde=True, color='skyblue',
edgecolor='black')
plt.title('Distribution of Minute', fontsize=15)
plt.xlabel('Minute', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()
```

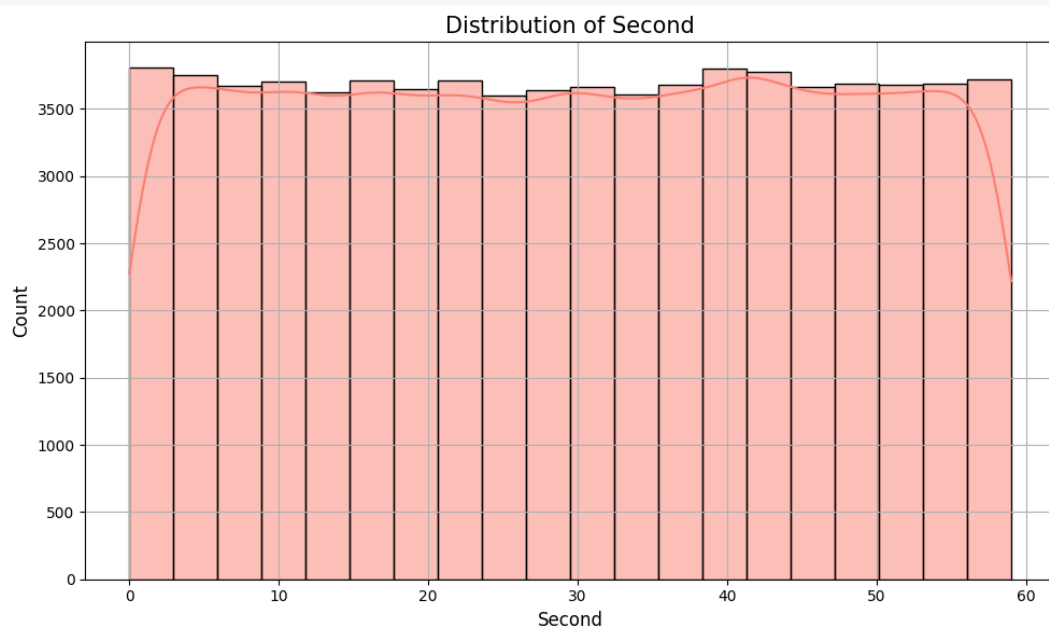


Gambar 4.3 Distribusi Menit Pertandingan

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.

```
plt.figure(figsize=(10, 6))
sns.histplot(df['second'], bins=20, kde=True, color='salmon',
edgecolor='black')
plt.title('Distribution of Second', fontsize=15)
plt.xlabel('Second', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()
```

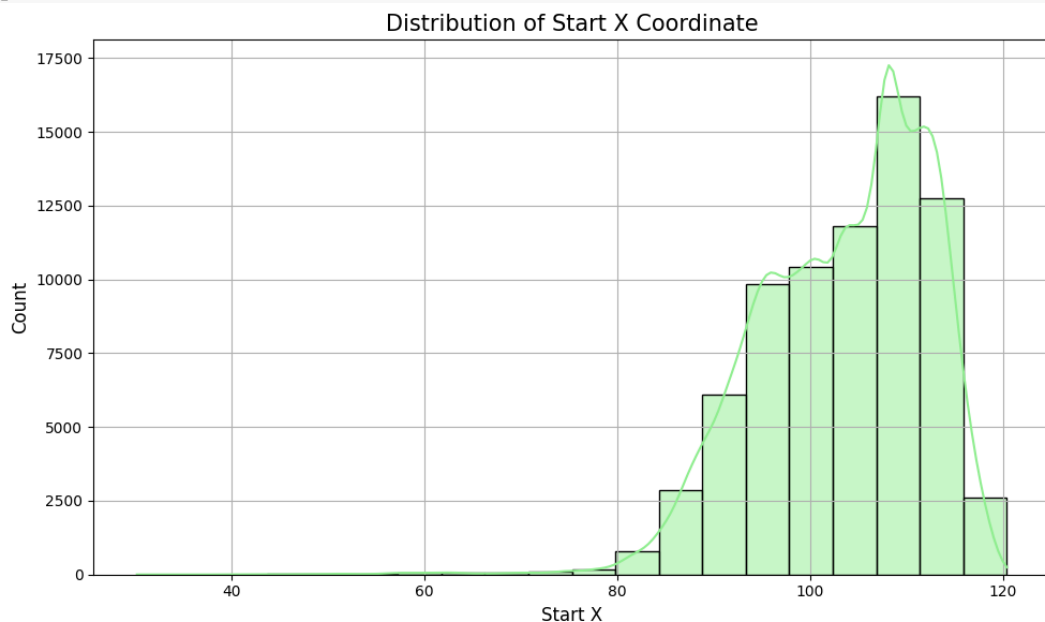


Gambar 4.4 Distribusi Detik Pertandingan

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).

```
plt.figure(figsize=(10, 6))
sns.histplot(df['start_x'], bins=20, kde=True,
color='lightgreen', edgecolor='black')
plt.title('Distribution of Start X Coordinate', fontsize=15)
plt.xlabel('Start X', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()
```

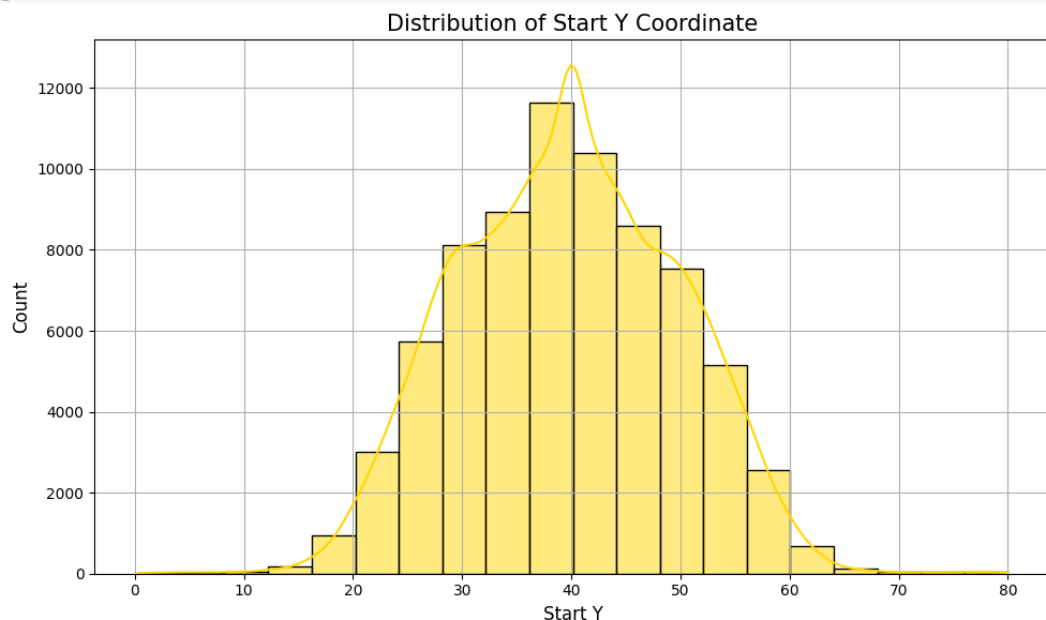


Gambar 4.5 Distribusi Koordinat *Start x*

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).

```
plt.figure(figsize=(10, 6))
sns.histplot(df['start_y'], bins=20, kde=True, color='gold',
edgecolor='black')
plt.title('Distribution of Start Y Coordinate', fontsize=15)
plt.xlabel('Start Y', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()
```



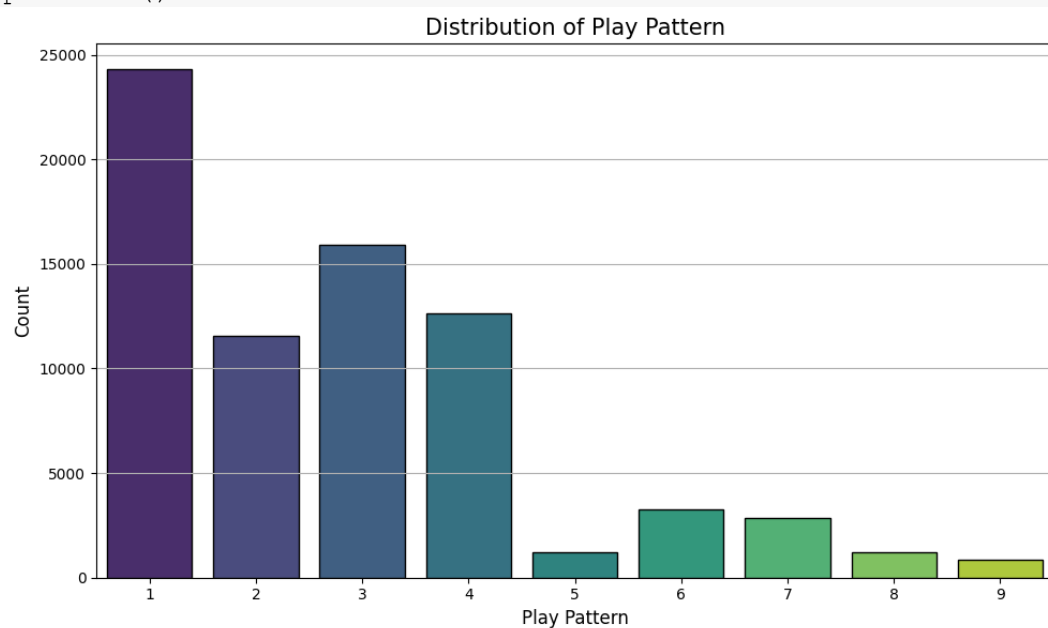
Gambar 4.6 Distribusi Koordinat *Start y*

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.

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='play_pattern', palette='viridis',
edgecolor='black')
plt.title('Distribution of Play Pattern', fontsize=15)
plt.xlabel('Play Pattern', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

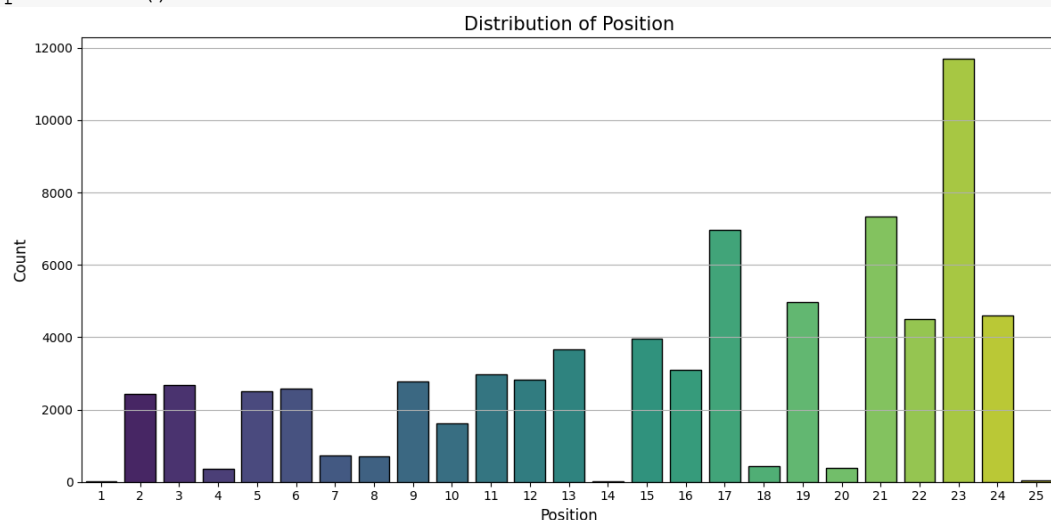


Gambar 4.7 Distribusi Pola Permainan (*Play Pattern*)

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.

```
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='position', palette='viridis',
edgecolor='black')
plt.title('Distribution of Position', fontsize=15)
plt.xlabel('Position', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

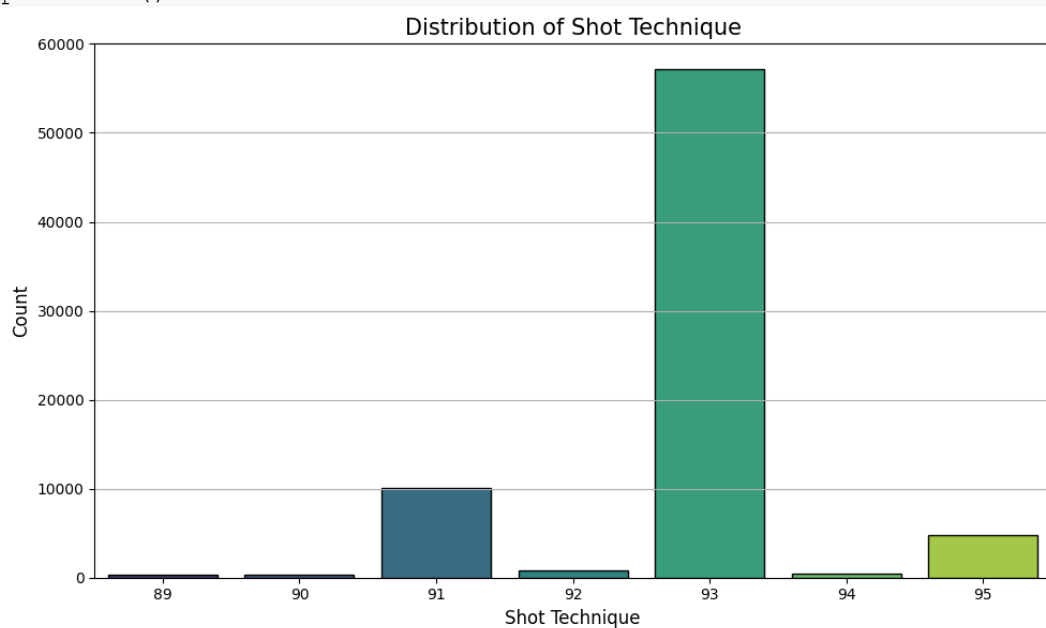


Gambar 4.8 Distribusi Posisi Pemain

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.

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='shot_technique', palette='viridis',
edgecolor='black')
plt.title('Distribution of Shot Technique', fontsize=15)
plt.xlabel('Shot Technique', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

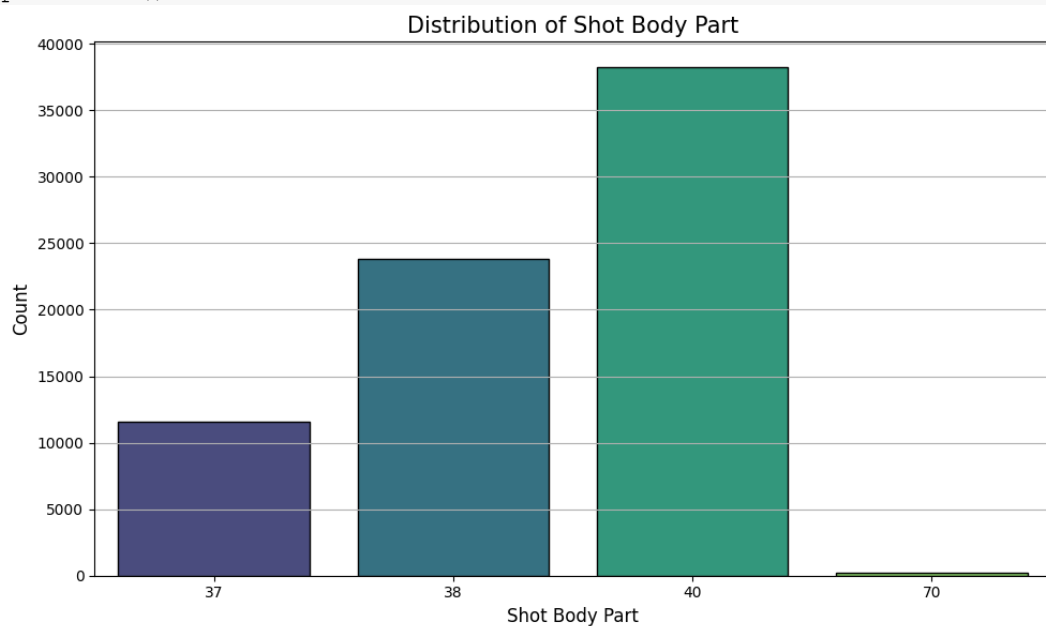


Gambar 4.9 Distribusi Teknik Tembakan

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.

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='shot_body_part', palette='viridis',
edgecolor='black')
plt.title('Distribution of Shot Body Part', fontsize=15)
plt.xlabel('Shot Body Part', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```



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

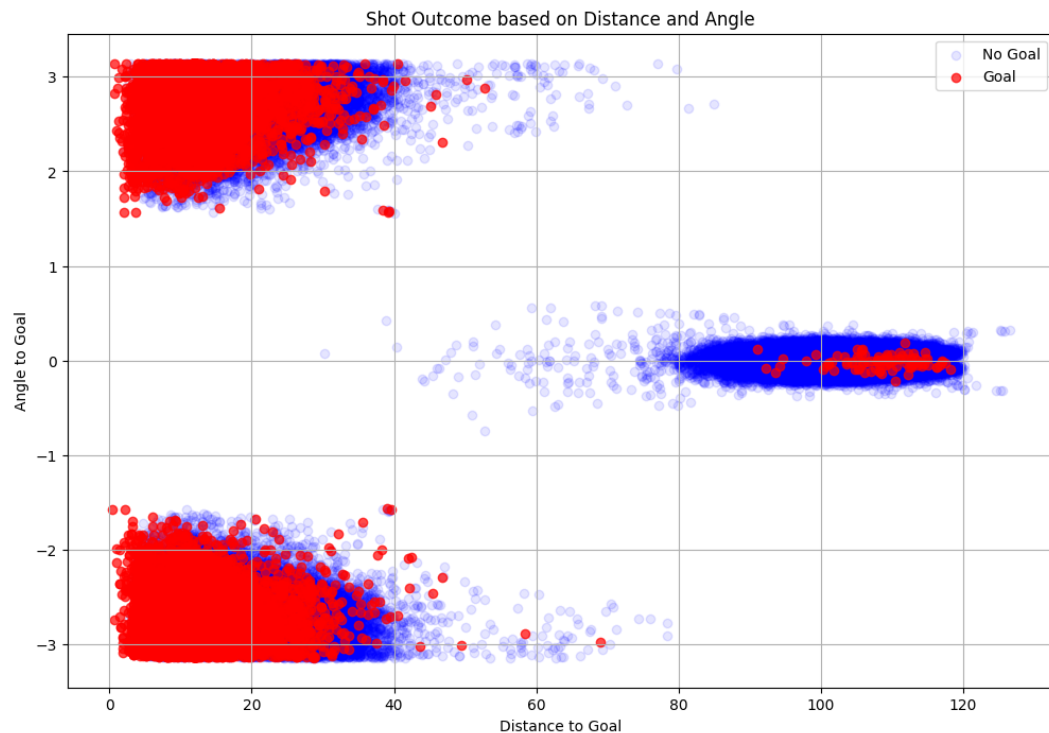


kaki kiri (ID 38) dan kepala (ID 37). Ini mencerminkan populasi pemain sepak bola di mana mayoritas adalah pengguna kaki kanan. Penggunaan kepala untuk menembak juga cukup signifikan, terutama dari situasi umpan silang atau bola mati. Kategori 'Lainnya' (*Other*, ID 70) sangat jarang, menunjukkan bahwa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

#### 4.2.2.2 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.

```
plt.figure(figsize=(12, 8))
goals = df[df['shot_outcome'] == 1]
no_goals = df[df['shot_outcome'] == 0]
plt.scatter(no_goals['distance_to_goal'],
            no_goals['angle_to_goal'], color='blue', alpha=0.1, label='No
            Goal')
plt.scatter(goals['distance_to_goal'], goals['angle_to_goal'],
            color='red', alpha=0.7, label='Goal')
plt.xlabel('Distance to Goal')
plt.ylabel('Angle to Goal')
plt.title('Shot Outcome based on Distance and Angle')
plt.legend()
plt.grid(True)
plt.show()
```



Gambar 4.11 Hasil Tembakan Berdasarkan Jarak dan Sudut

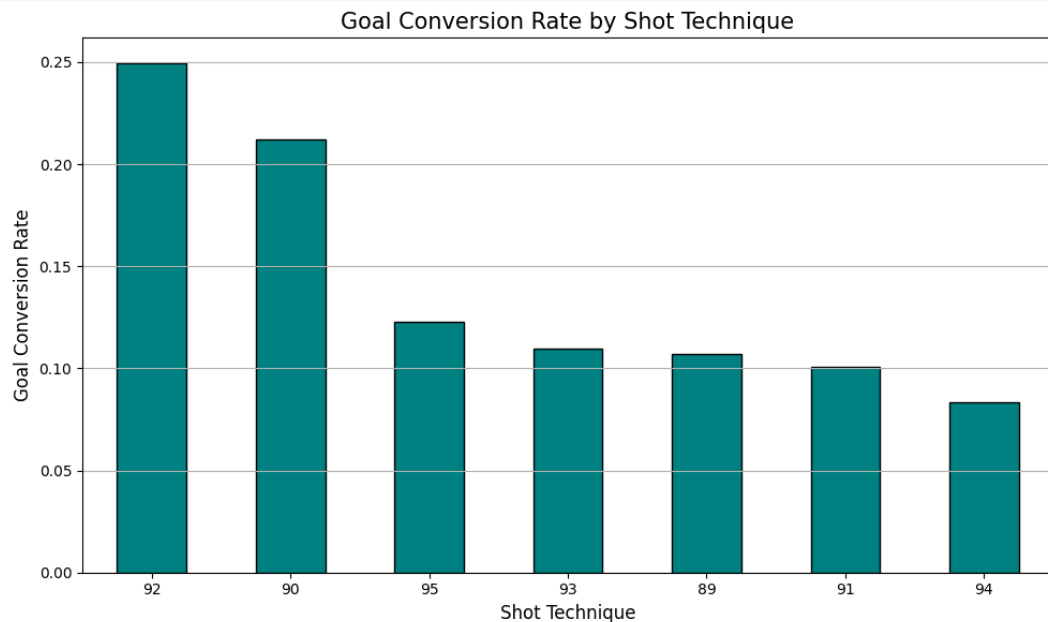
Pada Gambar 4.11, visualisasi ini secara jelas mengilustrasikan hubungan non-linear antara fitur spasial dengan hasil tembakan. Titik-titik merah yang merepresentasikan gol terkonsentrasi secara padat pada area dengan jarak ke gawang yang rendah (sumbu  $x$  kecil) dan sudut tembakan yang lebar (sumbu  $y$  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.

```
shot_technique_outcome =
df.groupby('shot_technique')['shot_outcome'].value_counts(normalize=True).unstack()
```

```

shot_technique_outcome['Goal Conversion Rate'] =
shot_technique_outcome[1] / (shot_technique_outcome[0] +
shot_technique_outcome[1])
plt.figure(figsize=(10, 6))
shot_technique_outcome['Goal Conversion
Rate'].sort_values(ascending=False).plot(kind='bar',
color='teal', edgecolor='black')
plt.title('Goal Conversion Rate by Shot Technique',
fontsize=15)
plt.xlabel('Shot Technique', fontsize=12)
plt.ylabel('Goal Conversion Rate', fontsize=12)
plt.xticks(rotation=0)
plt.grid(axis='y')
plt.tight_layout()
plt.show()

```

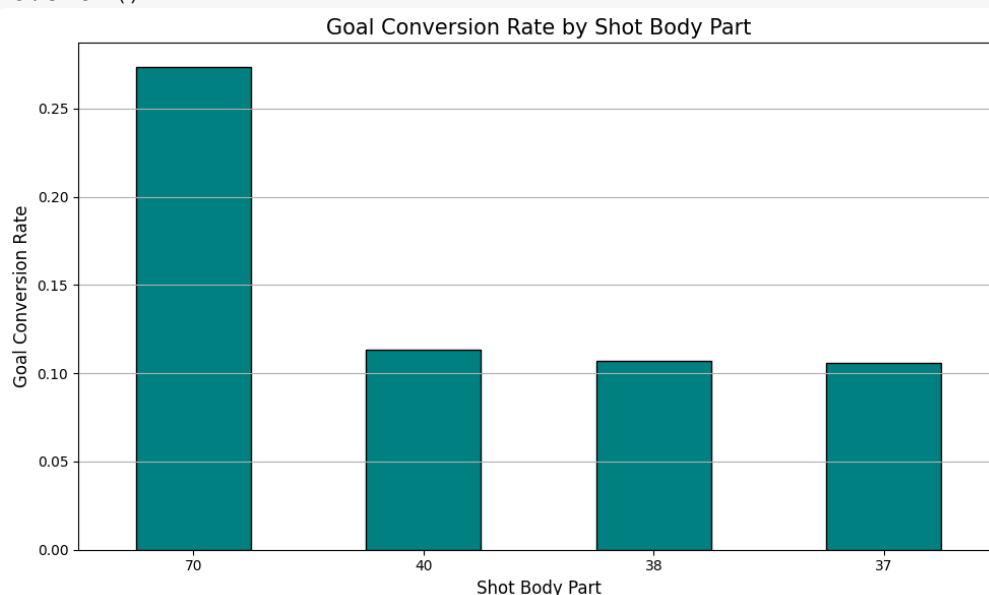


Gambar 4.12 Tingkat Konversi Gol Berdasarkan Teknik Tembakan

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.

```
shot_body_part_outcome =
df.groupby('shot_body_part')['shot_outcome'].value_counts(normalize=True).unstack()
shot_body_part_outcome['Goal Conversion Rate'] =
shot_body_part_outcome[1] / (shot_body_part_outcome[0] +
shot_body_part_outcome[1])
plt.figure(figsize=(10, 6))
shot_body_part_outcome['Goal Conversion
Rate'].sort_values(ascending=False).plot(kind='bar',
color='teal', edgecolor='black')
plt.title('Goal Conversion Rate by Shot Body Part',
fontsize=15)
plt.xlabel('Shot Body Part', fontsize=12)
plt.ylabel('Goal Conversion Rate', fontsize=12)
plt.xticks(rotation=0)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```



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.

#### 4.2.2.3 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	Memberikan keyakinan bahwa <i>dataset</i> yang digunakan berkualitas, konsisten, dan cukup representatif

	kaki kanan) selaras dengan pengetahuan umum sepak bola.	untuk membangun model prediksi xG yang andal dan valid.
--	---	---

### 4.2.3 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.

```
df.dropna(inplace=True)
print(f"Missing Values: {df.shape}")
df.drop_duplicates(inplace=True)
print(f>Data Duplikat: {df.shape}")
print(f"Total Data Setelah Data Cleansing: {df.shape[0]}")
```

- 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 *tree* (*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.

##### 4.3.1.1 **Fitur Spasial**

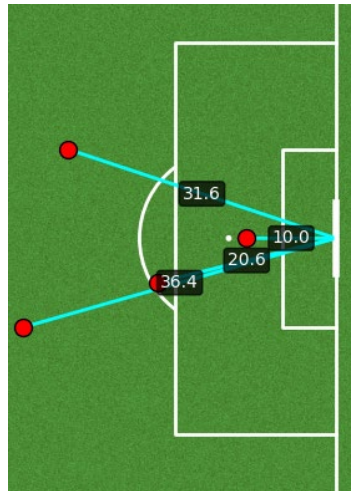
Mengubah data koordinat (*start\_x*, *start\_y*) menjadi metrik jarak dan sudut tembakan adalah langkah yang paling fundamental dan terbukti efektif dalam meningkatkan daya prediksi model. Dalam *dataset* ini, koordinat gawang yang digunakan sebagai referensi adalah: pusat gawang berada pada titik (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

- a. 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.

```
goal_center_x = 120.0
goal_center_y = 40.0
goal_post_bottom_y = 36.34
goal_post_top_y = 43.66
def distance_bw_coordinates(x1, y1, x2=goal_center_x,
y2=goal_center_y):
    return math.hypot(x2 - x1, y2 - y1)
```



Gambar 4.15 Visualisasi *distance\_to\_goal*

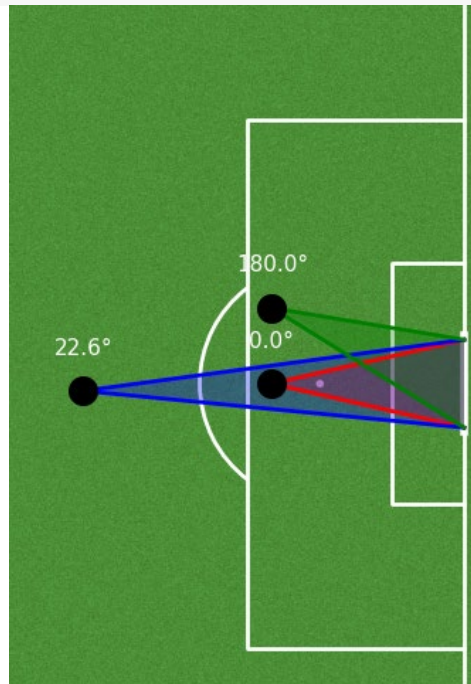
- b. Sudut Tembakan (*angle\_to\_goal*): Fitur ini merepresentasikan besar sudut pandang penendang terhadap gawang, dihitung menggunakan Hukum Kosinus pada segitiga yang dibentuk oleh titik tembakan dan kedua tiang gawang, sesuai dengan Persamaan 4.2 (Bandara *et al.* 2024).



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

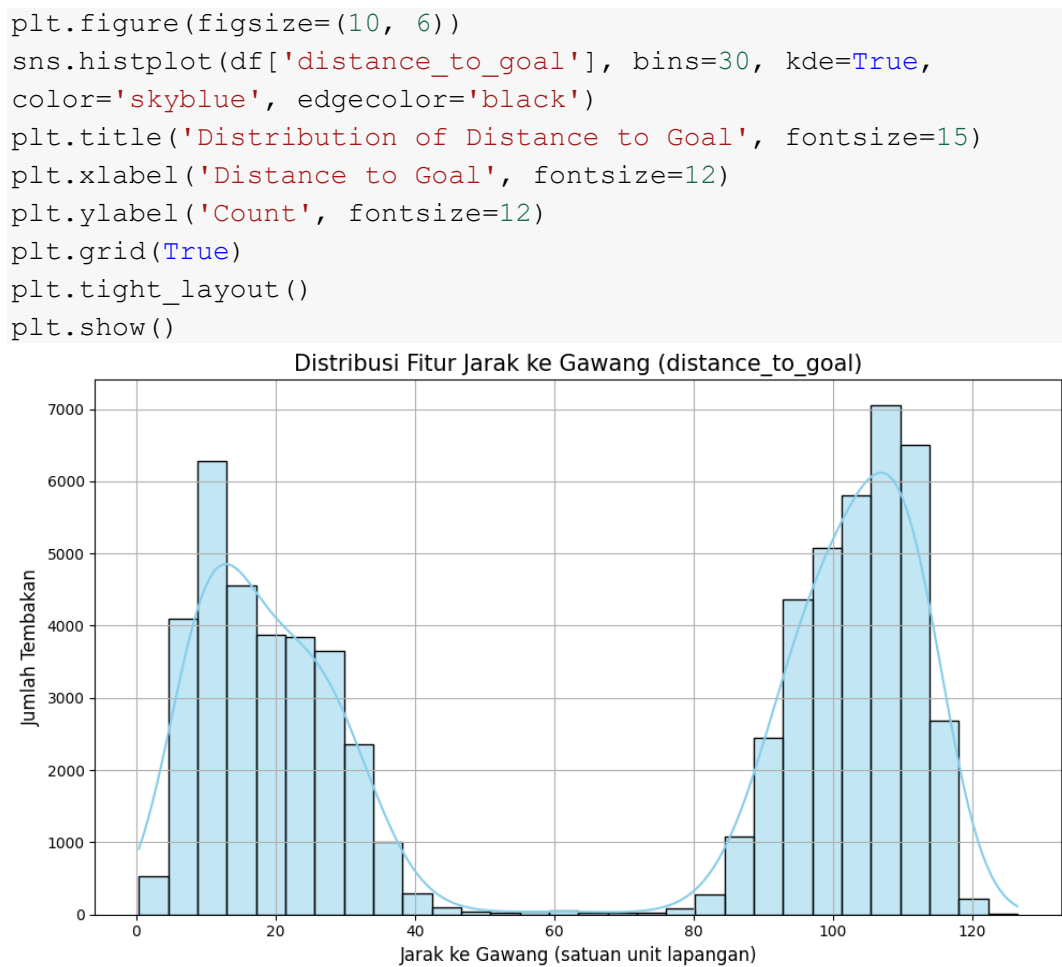
Dimana  $a$  dan  $b$  adalah jarak dari titik tembakan ke masing-masing tiang gawang, dan  $c$  adalah lebar gawang. Ilustrasi geometris dari perhitungan fitur ini dapat dilihat pada Gambar 4.16.

```
def post_angle(x, y,
               g1_x=goal_center_x, g1_y=goal_post_bottom_y,
               g2_x=goal_center_x, g2_y=goal_post_top_y):
    if x == goal_center_x and g1_y <= y <= g2_y:
        return 180.0
    if x == goal_center_x and (y < g1_y or y > g2_y):
        return 0.0
    A = distance_bw_coordinates(x, y, g1_x, g1_y)
    B = distance_bw_coordinates(x, y, g2_x, g2_y)
    C = distance_bw_coordinates(g1_x, g1_y, g2_x, g2_y)
    try:
        cos_value = (A**2 + B**2 - C**2) / (2 * A * B)
        angle_rad = np.arccos(np.clip(cos_value, -1.0, 1.0))
        return np.degrees(angle_rad)
    except:
        return 0.0
```



Gambar 4.16 Visualisasi *angle\_to\_goal*

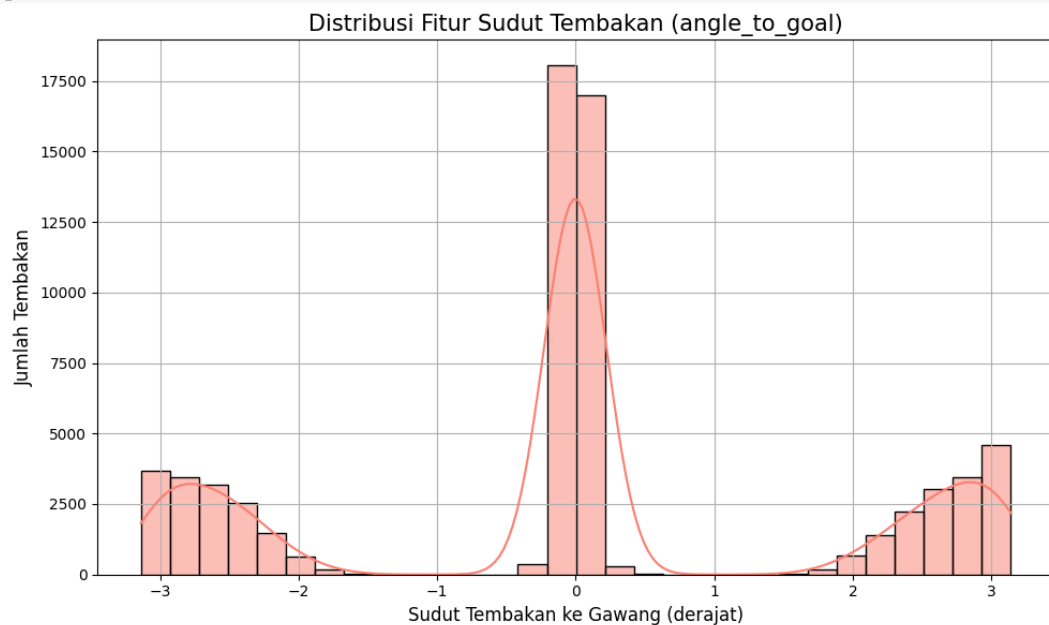
Setelah fitur *distance\_to\_goal* dan *angle\_to\_goal* dihitung, distribusinya divisualisasikan untuk memahami karakteristiknya. Gambar 4.17 menunjukkan distribusi jarak tembakan.

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.

```
plt.figure(figsize=(10, 6))
sns.histplot(df['angle_to_goal'], bins=30, kde=True,
color='salmon', edgecolor='black')
plt.title('Distribution of Angle to Goal', fontsize=15)
plt.xlabel('Angle to Goal', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()
```



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.

#### 4.3.1.2 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.

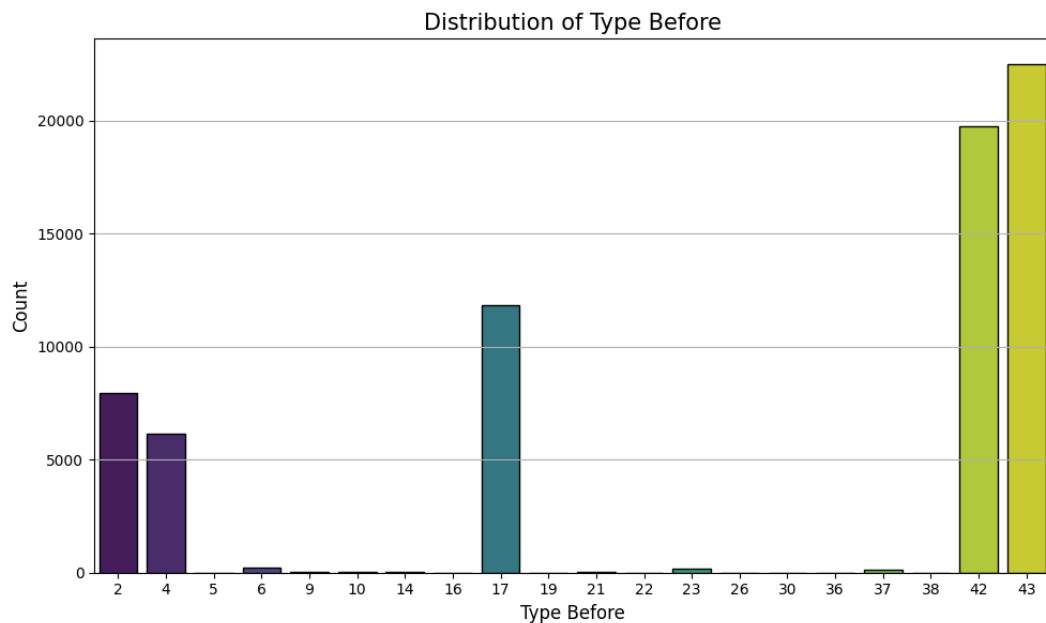
<i>Event Type</i>	<i>Type ID</i>	Deskripsi Singkat
50/50	33	Dua pemain dari tim berbeda berebut bola lepas.

<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 ( <i>penyelamatan, smother, punch, dll</i> ).
<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.

```
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='type_before', palette='viridis',
edgecolor='black')
plt.title('Distribution of Type Before', fontsize=15)
plt.xlabel('Type Before', fontsize=12)
plt.ylabel('Count', fontsize=12)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```



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 *Split Dataset*

Tahap akhir dari persiapan data adalah membagi *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 4.5.

```
train_df, test_df = train_test_split(df, test_size=0.1,
random_state=seed)
```

Tabel 4.5 Rincian Dimensi *Dataset* Akhir

<i>Dataset</i>	Jumlah Baris (Sampel Tembakan)
Data Latih	66.459
Data Uji	7.385
<b>Total</b>	<b>73.844</b>



### 4.3.3 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 dengan total 17 fitur dengan 1 variabel target. 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 4.6 merinci himpunan fitur akhir yang dipilih untuk digunakan dalam pemodelan xG.

```
X_train = train_df.drop(['shot_outcome', 'end_x', 'end_y'],
axis=1)
y_train = train_df['shot_outcome']
X_test = test_df.drop(['shot_outcome', 'end_x', 'end_y'],
axis=1)
y_test = test_df['shot_outcome']
```

Tabel 4.6 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>

## 4.4 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 *Hyperparameter Tuning*

Proses pemodelan diawali dengan inisialisasi *LGBMClassifier*. Kemudian, 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.

```
lgbm_param_grid = {
    'min_child_samples': sp_randint(0, 200),
    'num_leaves': sp_randint(2, 500),
    'reg_lambda': sp_uniform(0, 1),
    'reg_alpha': sp_uniform(0, 1),
    'max_depth': sp_randint(0, 500),
}
```

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.

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.

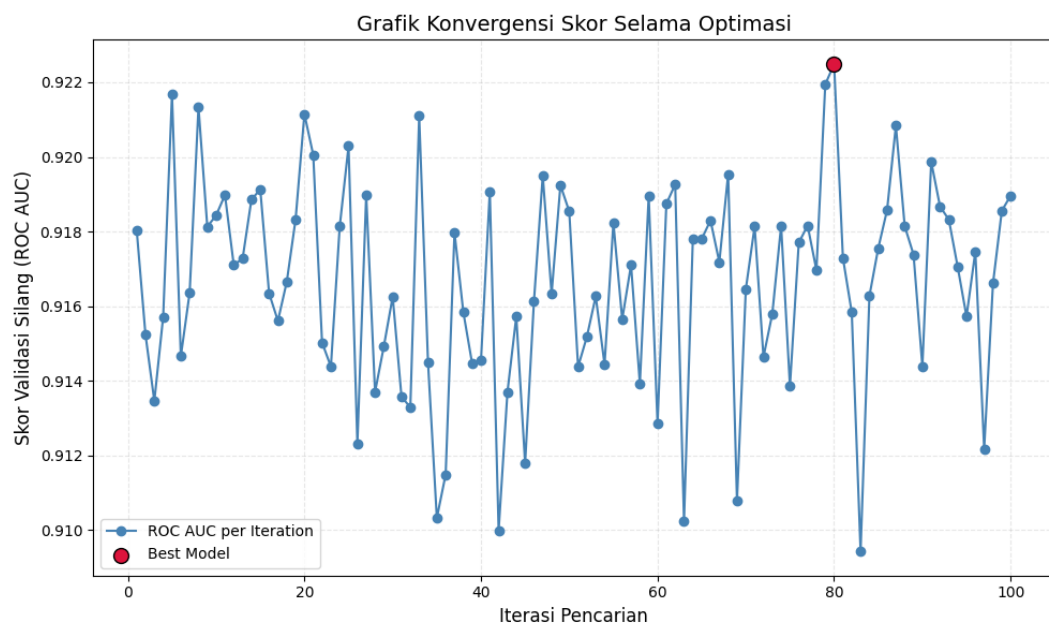
Proses pencarian ini melalui banyak iterasi untuk menemukan performa terbaik. Gambar 4.20 menunjukkan fluktuasi skor ROC AUC pada setiap iterasi. Dari grafik tersebut, dapat diamati bahwa proses pencarian telah menjelajahi berbagai tingkat performa dan berhasil menemukan titik optimalnya, yang ditandai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

```
searchcv = RandomizedSearchCV(estimator=lgbm,
                              n_iter=100,
                              param_distributions=lgbm_param_g
rid,
                              cv=5,
                              scoring=scoring,
                              refit='roc_auc',
                              n_jobs=-1)
results_df = pd.DataFrame(searchcv.cv_results_)
scores = results_df['mean_test_roc_auc'].values
best_index = results_df['rank_test_roc_auc'].idxmin()
plt.figure(figsize=(10, 6))
plt.plot(
    range(1, len(scores) + 1),
    scores,
    marker='o',
    linestyle='-',
    color='steelblue',
    label='ROC AUC per Iteration'
)
plt.scatter(
    best_index + 1,
    scores[best_index],
    color='crimson',
    edgecolor='black',
    s=100,
    label='Best Model',
```

```

        zorder=5
    )
    plt.title("Grafik Konvergensi Skor Selama Optimasi",
             fontsize=14, color='black')
    plt.xlabel("Iterasi Pencarian", fontsize=12, color='black')
    plt.ylabel("Skor Validasi Silang (ROC AUC)", fontsize=12,
              color='black')
    plt.tick_params(colors='black')
    plt.grid(True, linestyle='--', alpha=0.3)
    plt.legend()
    plt.tight_layout()
    plt.show()

```



Gambar 4.20 Grafik Konvergensi Skor Selama Optimasi

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

```

top_5_results = results_df.nlargest(5, 'mean_test_roc_aug')
top_5_results = top_5_results[[col for col in
top_5_results.columns if col.startswith('param_')] +
['mean_test_roc_aug']]
print("Top 5 Hyperparameter Configurations by ROC AUC:")
display(top_5_results)

```

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.

#### 4.4.2 Train Model

Hasil dari tahap ini adalah sebuah model LightGBM final yang telah dilatih secara optimal. Proses pelatihan ini secara teknis dieksekusi saat memanggil

metode *fit* pada objek `RandomizedSearchCV`. Setelah proses pencarian dan validasi silang selesai, `RandomizedSearchCV` secara otomatis melatih ulang model terbaik (*best estimator*) menggunakan seluruh data latih (*training set*) dengan konfigurasi *hyperparameter* yang telah ditemukan.

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*).

```
searchcv.fit(X_train, y_train)
y_pred = searchcv.predict_proba(X_test)[: , 1]
new_data['xG'] = y_pred
new_data[['xG', 'shot_outcome']]
```

	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.

#### 4.4.3 Model Probability Calibration

Model terbaik yang telah dilatih (*fit*) 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 uji). 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.

Proses ini dievaluasi dengan memvisualisasikan kurva kalibrasi. Kurva ini dibuat dengan mengelompokkan semua prediksi ke dalam beberapa *bins*. Untuk setiap *bins*, dilakukan perbandingan antara prediksi model dengan realitas di lapangan.

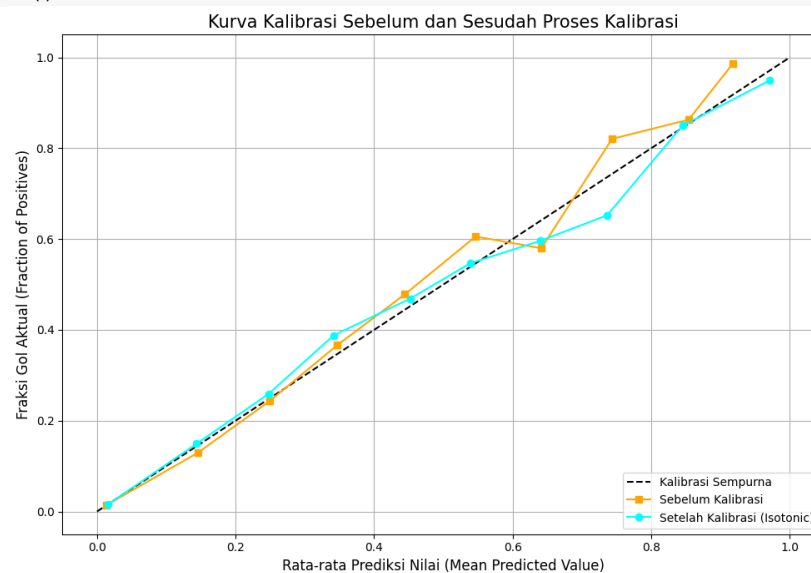
Keberhasilan proses kalibrasi ini dibuktikan secara visual melalui Gambar 4.22 menampilkan perbandingan kurva sebelum (garis oranye) dan sesudah (garis cyan) proses kalibrasi. Dapat dianalisis bahwa kurva model yang telah terkalibrasi posisinya jauh lebih mendekati garis diagonal "kalibrasi sempurna" (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.



```

best_lgbm = searchcv.best_estimator_
calibrated_lgbm = CalibratedClassifierCV(best_lgbm,
method='isotonic', cv=3)
calibrated_lgbm.fit(X_train, y_train)
best_model = calibrated_lgbm
fraction_of_positives, mean_predicted_value =
calibration_curve(y, prob_pos, n_bins=10)
cal_fraction_of_positives, cal_mean_predicted_value =
calibration_curve(y, calibrated_prob_pos, n_bins=10)
plt.style.use('seaborn-v0_8-whitegrid')
fig, ax = plt.subplots(figsize=(8, 6))
ax.plot([0, 1], [0, 1], "k--", label="Kalibrasi Sempurna")
ax.plot(mean_predicted_value, fraction_of_positives, "s-",
color="orange", label="Sebelum Kalibrasi")
ax.plot(cal_mean_predicted_value, cal_fraction_of_positives,
"o-", color="cyan", label="Setelah Kalibrasi (Isotonic)")
ax.set_title("Kurva Kalibrasi Sebelum dan Sesudah Proses
Kalibrasi", fontsize=14)
ax.set_xlabel("Rata-rata Prediksi Nilai (Mean Predicted
Value)")
ax.set_ylabel("Fraksi Gol Aktual (Fraction of Positives)")
ax.legend(loc="lower right")
ax.grid(True)
plt.show()

```



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.

```
y_pred_lgbm_calibrated = best_model.predict_proba(X_test)[:,\n1]\nfraction_of_positives_lgbm, mean_predicted_value_lgbm =\ncalibration_curve(y_test, y_pred_lgbm_calibrated, n_bins=10)
```

### 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*.

```
print('Brier score, Light GBM:', brier_score_loss(y_test,
y_pred_lgbm_calibrated, pos_label=y_test.max()))
```

Tabel 4.9 Hasil Evaluasi *Brier Score* Model LightGBM

Metrik	Nilai
<i>Brier Score</i>	0,0621

Nilai *Brier Score* sebesar 0,0621 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.

```
print('ROC AUC, Light GBM:', roc_auc_score(y_test,
y_pred_lgbm_calibrated))
```

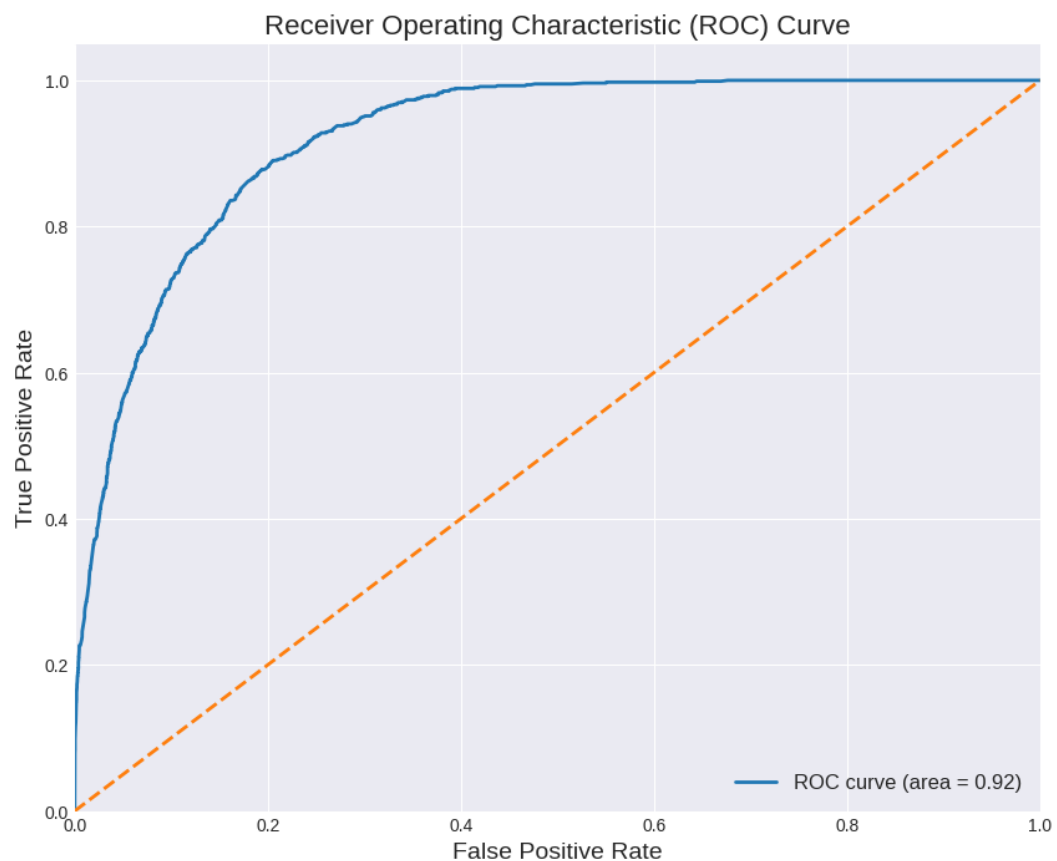
Tabel 4.10 Hasil Evaluasi ROC AUC Model LightGBM

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*.

```
fpr, tpr, thresholds = roc_curve(y_test,
y_pred_lgbm_calibrated)
roc_auc = auc(fpr, tpr)
plt.style.use('seaborn-v0_8-darkgrid')
plt.figure(figsize=(10, 8))
```

```
plt.plot(fpr, tpr, color='#1f77b4', lw=2, label='ROC curve
(area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='#ff7f0e', lw=2, linestyle='--
')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate', fontsize=14)
plt.ylabel('True Positive Rate', fontsize=14)
plt.title('Receiver Operating Characteristic (ROC) Curve',
fontsize=16)
plt.legend(loc="lower right", fontsize=12)
plt.grid(True)
plt.show()
```



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.

```
accuracy = accuracy_score(y_test, y_pred_labels)
```

Tabel 4.11 Hasil Evaluasi Akurasi Model LightGBM

Metrik	Nilai
Akurasi	0,9113

Nilai akurasi yang dicapai oleh model adalah 0,9113 (atau 91,13%), 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.

```
precision = precision_score(y_test, y_pred_labels,
                           average='weighted')
```

Tabel 4.12 Hasil Evaluasi Presisi Model LightGBM

Metrik	Nilai
Presisi	0,8996

Nilai presisi sebesar 0,8996 (atau 89,96%) 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*.

```
recall = recall_score(y_test, y_pred_labels,
average='weighted')
```

Tabel 4.13 Hasil Evaluasi *Recall* Model LightGBM

Metrik	Nilai
<i>Recall</i>	0,9113

Nilai *recall* sebesar 0,9113 (atau 91,13%) 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*.

```
f1 = f1_score(y_test, y_pred_labels, average='weighted')
```

Tabel 4.14 Hasil Evaluasi *F1-Score* Model LightGBM

Metrik	Nilai
<i>F1-Score</i>	0,9010

Nilai F1-Score sebesar 0,9010 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*.

```
logloss = log_loss(y_test, y_pred_lgbm_calibrated)
```

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

```
for name, model in models_to_compare.items():
    print(f"Training {name}...")
    cv_results = cross_validate(model, X_train, y_train, cv=5,
                                scoring=scoring,
                                n_jobs=-1,
                                return_train_score=False)
    total_fit_time = np.sum(cv_results['fit_time'])
    results[name] = {
        'Total Fit Time (s)': total_fit_time
    }
    print(f"Selesai {name}. Total Waktu Training:
{total_fit_time:.4f} detik")
results_df = pd.DataFrame(results).T
print("\n--- Hasil Perbandingan Kecepatan Training ---")
display(results_df.sort_values('Total Train Time (s)'))
```

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.

```
team_xg_summary = new_data.groupby('team_name').agg(
    Total_xG=('xG', 'sum'),
    Shots=('xG', 'count'),
    Goals=('shot_outcome', 'sum')
).reset_index()
team_xg_summary['Average_xG_per_Shot'] =
team_xg_summary['Total_xG'] / team_xg_summary['Shots']
team_xg_summary['Goal_Difference'] = team_xg_summary['Goals']
- team_xg_summary['Total_xG']
team_xg_summary = team_xg_summary.rename(columns={
    'team_name': 'Nama Tim',
    'Total_xG': 'Total Expected Goals (xG)',
    'Shots': 'Jumlah Tembakan',
    'Goals': 'Jumlah Gol Sebenarnya',
    'Average_xG_per_Shot': 'Rata-rata xG per Tembakan',
    'Goal_Difference': 'Diferensial Gol (Gol - xG)'
})
team_xg_summary['Total Expected Goals (xG)'] =
team_xg_summary['Total Expected Goals (xG)'].round(2)
team_xg_summary['Rata-rata xG per Tembakan'] =
team_xg_summary['Rata-rata xG per Tembakan'].round(3)
```

```
team_xg_summary['Diferensial Gol (Gol - xG)'] =
team_xg_summary['Diferensial Gol (Gol - xG)'].round(2)
team_xg_summary
```

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.

```
pitch = VerticalPitch(
    pitch_type='statsbomb',
```

```

    half=True,
    pitch_color='#f8f8f8',
    line_color='#2d2d2d',
    linewidth=2
)
fig, ax = pitch.draw(figsize=(12, 10))
team_colors = {
    'England': {'goal': '#C8102E', 'miss': '#C8102E'}
    'Iran': {'goal': '#239F40', 'miss': '#239F40'}
}
teams = new_data['team_name'].unique()
for team in teams:
    team_data = new_data[new_data['team_name'] == team]
    team_goals = team_data[team_data['shot_outcome'] == 1]
    team_misses = team_data[team_data['shot_outcome'] != 1]
    if len(team_misses) > 0:
        pitch.scatter(
            team_misses['start_x'], team_misses['start_y'],
            s=team_misses['xG'] * 1000 + 50,
            facecolors='white',
            edgecolors=team_colors[team]['miss'],
            linewidth=2,
            ax=ax,
            alpha=0.8,
            label=f'{team} - Miss'
        )
    if len(team_goals) > 0:
        pitch.scatter(
            team_goals['start_x'], team_goals['start_y'],
            s=team_goals['xG'] * 1000 + 50,
            color=team_colors[team]['goal'],
            edgecolors='white',
            linewidth=1.5,
            ax=ax,
            alpha=0.9,
            label=f'{team} - Goal'
        )
fig.suptitle(
    "England vs Iran\nFIFA World Cup 2022 - Shot Map",
    fontsize=20,
    fontweight='bold',
    y=0.95,
    color='#2d2d2d'
)
ax.text(

```

```

    0.5, 0.02,
    "Circle size represents Expected Goals (xG) value •
    Penalties excluded",
    transform=ax.transAxes,
    ha='center',
    fontsize=11,
    style='italic',
    color='#666666'
)
from matplotlib.patches import Circle
from matplotlib.lines import Line2D
legend_elements = []
for team in teams:
    legend_elements.append(
        Line2D([0], [0], marker='o', color='w',
                markerfacecolor=team_colors[team]['goal'],
                markeredgecolor='white', markeredgewidth=1.5,
                markersize=12, label=f'{team} - Goal')
    )
    legend_elements.append(
        Line2D([0], [0], marker='o', color='w',
                markerfacecolor='white',
                markeredgecolor=team_colors[team]['miss'],
                markeredgewidth=2, markersize=12,
label=f'{team} - Miss')
    )
legend_elements.extend([
    Line2D([0], [0], marker='o', color='w',
markerfacecolor='lightgrey',
                markeredgecolor='grey', markersize=14, label='Low
xG (0.1)'),
    Line2D([0], [0], marker='o', color='w',
markerfacecolor='lightgrey',
                markeredgecolor='grey', markersize=20, label='High
xG (0.5+)')
])
legend = ax.legend(
    handles=legend_elements,
    loc='upper left',
    bbox_to_anchor=(0.02, 0.98),
    frameon=True,
    fancybox=True,
    shadow=True,
    ncol=1,
    fontsize=11,

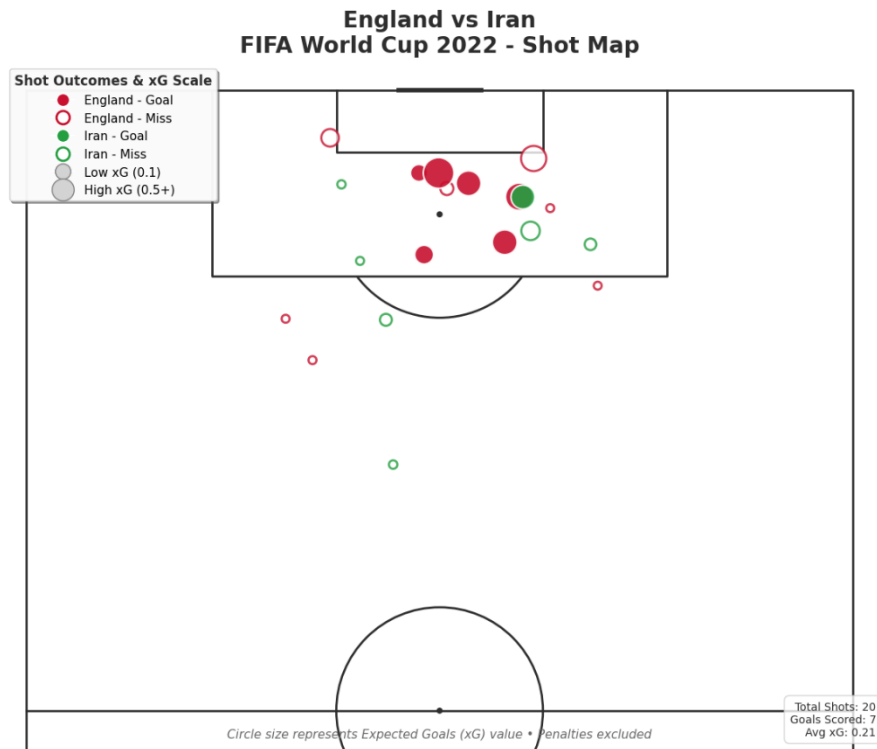
```



```

        title="Shot Outcomes & xG Scale",
        title_fontsize=12,
        framealpha=0.95,
        edgecolor='#cccccc'
    )
    legend.get_frame().set_facecolor('white')
    legend.get_title().set_fontweight('bold')
    legend.get_title().set_color('#2d2d2d')
    ax.set_facecolor('#f8f8f8')
    for spine in ax.spines.values():
        spine.set_visible(False)
    ax.axis('off')
    total_shots = len(new_data)
    total_goals = len(new_data[new_data['shot_outcome'] == 1])
    avg_xg = new_data['xG'].mean()
    stats_text = f"Total Shots: {total_shots}\nGoals Scored: {total_goals}\nAvg xG: {avg_xg:.2f}"
    ax.text(
        0.98, 0.02,
        stats_text,
        transform=ax.transAxes,
        ha='right',
        va='bottom',
        bbox=dict(boxstyle="round,pad=0.5", facecolor='white',
        alpha=0.8, edgecolor='#cccccc'),
        fontsize=10,
        color='#2d2d2d'
    )
    plt.tight_layout()
    plt.subplots_adjust(top=0.9)
    plt.show()

```



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*

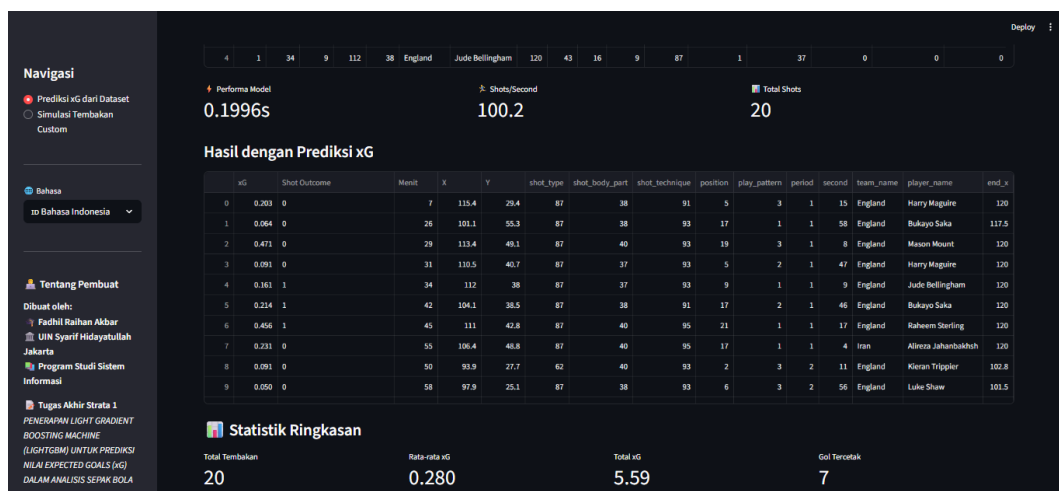
Untuk menjembatani kesenjangan antara model *machine learning* yang kompleks dengan penerapan praktis oleh pengguna akhir, sebuah antarmuka aplikasi web interaktif dikembangkan menggunakan *framework* Streamlit. Aplikasi ini, yang dinamakan "*xG Prediction Interface*", dirancang untuk menyediakan akses 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.

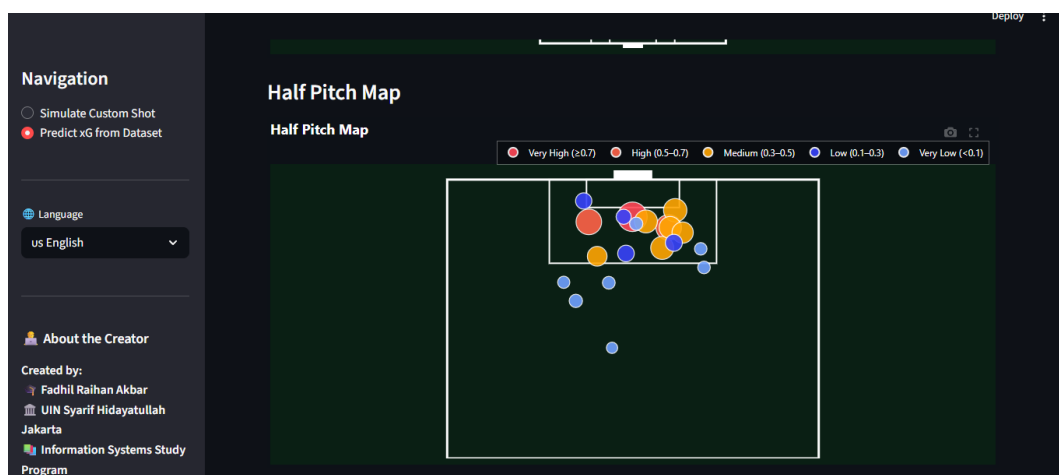
Antarmuka ini menawarkan dua fitur utama yang dirancang untuk memenuhi kebutuhan analisis yang berbeda:

- a. *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.

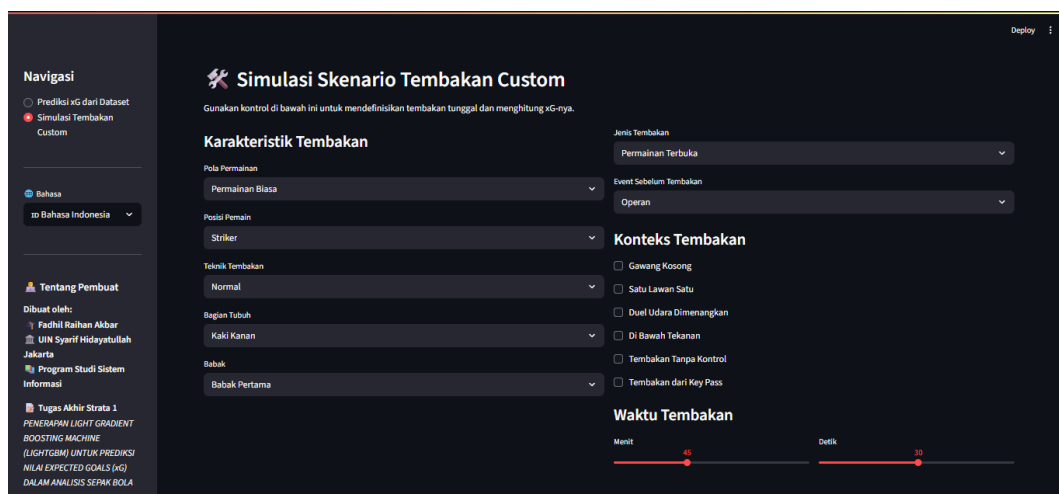


Gambar 4.25 Tampilan Halaman Prediksi Berdasarkan *Dataset*

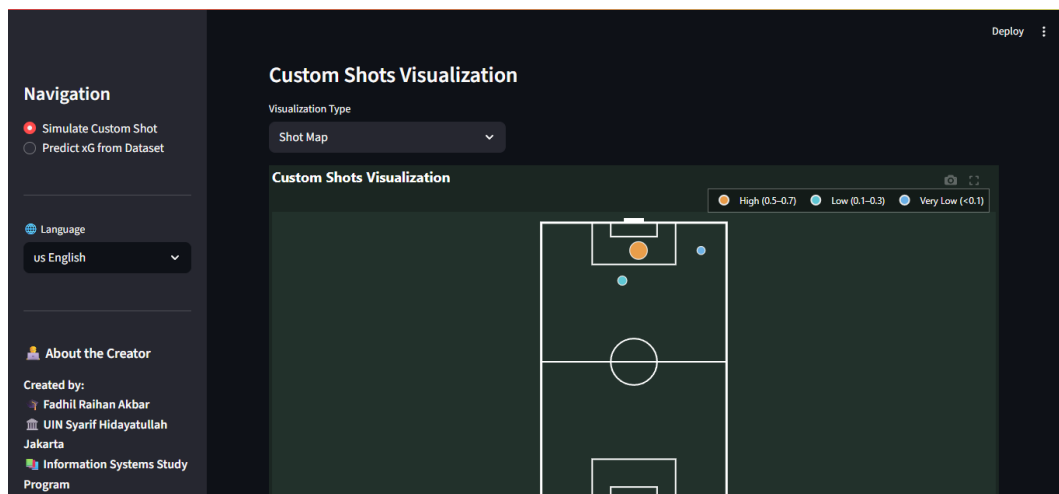


Gambar 4.26 Tampilan Halaman Visualisasi Prediksi Berdasarkan *Dataset*

- 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	<i>Brier Score</i>	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	–
<b>Model Penelitian</b>	LightGBM (with Calibration)	<b>0,0621</b>	<b>0,9209</b>	<b>0,9113</b>	<b>0,8996</b>	<b>0,9113</b>	<b>0,9010</b>	<b>0,21</b>

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
<b>England vs Iran – WC 2022</b>	6 – 2	<b>Penelitian ini</b>	England: <b>3,09</b>	Iran: <b>0,87</b>
		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
<b>England vs France – WC 2022</b>	1 – 2	<b>Penelitian ini</b>	England: <b>1,98</b>	France: <b>0,64</b>
		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
<b>Argentina vs Croatia – WC 2022</b>	3 – 0	<b>Penelitian ini</b>	Argentina: <b>2,01</b>	Croatia: <b>0,95</b>
		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
<b>Spain vs England – Final EURO 2024</b>	2 – 1	<b>Penelitian ini</b>	England: <b>0,67</b>	Spain: <b>1,63</b>
		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 Implikasi Penelitian**

Berdasarkan hasil analisis dan interpretasi yang telah dipaparkan, penelitian ini menghasilkan sejumlah implikasi penting, baik dari segi teoritis maupun praktis, dalam bidang analisis data olahraga.

Secara teoritis, penelitian ini memberikan kontribusi dengan memvalidasi efektivitas dan superioritas algoritma LightGBM untuk memodelkan data sepak bola yang bersifat kompleks dan non-linear. Kinerja model yang unggul

dibandingkan dengan tolok ukur akademis memperkuat argumen bahwa metode *gradient boosting* merupakan pendekatan yang sangat cocok untuk menangani interaksi variabel yang rumit dalam data olahraga. Lebih dari itu, penelitian ini juga menegaskan bahwa untuk menghasilkan model prediktif probabilistik yang andal seperti xG, proses tidak berhenti pada pelatihan model saja. Diperlukan sebuah kerangka kerja yang holistik, yang mencakup optimalisasi *hyperparameter tuning* untuk memaksimalkan daya diskriminatif, serta proses kalibrasi seperti regresi isotonik untuk memastikan akurasi nilai probabilitas yang dihasilkan.

Dari sisi praktis, model xG dan antarmuka aplikasi yang dikembangkan memiliki potensi pemanfaatan yang luas bagi para pemangku kepentingan di industri sepak bola. Bagi klub dan analis performa, model ini dapat berfungsi sebagai alat bantu objektif untuk evaluasi pasca-pertandingan, analisis lawan, hingga proses rekrutmen pemain. Bagi media dan penyiaran, antarmuka web yang dikembangkan menunjukkan bagaimana metrik xG dapat disajikan secara interaktif dan mudah dipahami, sehingga berpotensi meningkatkan keterlibatan penggemar dengan wawasan yang lebih mendalam. Selain itu, fitur simulasi pada antarmuka aplikasi dapat digunakan sebagai media edukasi bagi pemain muda untuk melatih pengambilan keputusan (*shot selection*) yang lebih baik di lapangan dengan menunjukkan secara visual bagaimana berbagai faktor memengaruhi probabilitas gol.

#### **4.8 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 5

### PENUTUP

#### 5.1 Kesimpulan

Setelah melalui serangkaian proses penelitian, mulai dari pengolahan data hingga evaluasi model, maka dapat ditarik kesimpulan sebagai berikut:

- a. Penggunaan algoritma LightGBM terbukti berhasil meningkatkan akurasi prediksi nilai xG pada data sepak bola yang polanya kompleks. Melalui beberapa tahapan seperti rekayasa fitur, pengaturan *hyperparameter*, dan kalibrasi, hasil yang didapatkan sangat baik. Ini terlihat dari pengujian pada data uji, di mana model ini mencapai akurasi 0,9113, presisi 0,8996, *recall* 0,9113, dan F1-Score 0,9010. Untuk pengujian probabilitasnya, model ini juga mendapatkan hasil yang memuaskan dengan *Brier Score* 0,0621, ROC AUC 0,9209, dan *Log-Loss* 0,2. Hasil ini menunjukkan bahwa model yang dibuat tidak hanya bisa membedakan peluang bagus dan tidak, tapi juga mampu memberikan nilai probabilitas yang akurat.
- b. Dari segi kecepatan proses, algoritma LightGBM terbukti lebih unggul. Waktu yang dibutuhkan untuk melatih model hanya 49,75 detik, jauh lebih cepat dibandingkan algoritma lain seperti XGBoost, AdaBoost, dan *Random Forest*. Keunggulan kecepatan ini sangat penting dalam aplikasi praktis, karena memungkinkan model untuk dilatih berulang kali dengan data baru, mempercepat proses prediksi untuk analisis pertandingan secara *real-time*, dan

memberikan fleksibilitas untuk bereksperimen dengan berbagai set fitur tanpa memakan waktu lama.

## 5.2 Saran

Meskipun penelitian ini telah mencapai tujuannya, masih terdapat ruang untuk pengembangan di masa depan. Berdasarkan temuan dan keterbatasan yang ada, beberapa saran untuk penelitian selanjutnya adalah sebagai berikut:

- a. Menambah fitur model untuk penelitian selanjutnya, disarankan untuk mencoba fitur yang lebih lengkap, seperti data posisi pemain bertahan dan kiper saat ada tendangan (*freeze frame data*). Penambahan data seperti ini diharapkan bisa membuat model lebih paham situasi di lapangan dan hasilnya lebih akurat.
- b. Menguji model pada liga yang berbeda agar tahu apakah model ini bisa diandalkan secara umum, penelitian selanjutnya bisa mencoba melakukan pengujian antar liga. Misalnya, model dilatih dengan data dari satu liga, lalu diuji pada data liga lain untuk melihat apakah hasilnya tetap bagus walaupun gaya permainannya berbeda.
- c. Mengembangkan aplikasi menjadi lebih baik, aplikasi yang sudah dibuat bisa dikembangkan lagi dengan menambahkan fitur-fitur baru. Contohnya, analisis performa untuk tiap pemain atau perbandingan taktik tim berdasarkan data xG, agar aplikasinya lebih lengkap dan berguna bagi para pelatih atau analis.



## DAFTAR PUSTAKA

- Aggarwal, A. (2020). Limit shapes and local statistics for the stochastic six-vertex model. *Communications in Mathematical Physics*, 376(1), 681-746.
- Ahmad, A., Ahmad, W., Chaiyasarn, K., Ostrowski, K. A., Aslam, F., Zajdel, P., & Joyklad, P. (2021). Prediction of Geopolymer Concrete Compressive Strength Using Novel Machine Learning Algorithms. *Polymers*, 13(19), 3389. <https://doi.org/10.3390/polym13193389>
- Al-Labadi, L., Evans, M., & Liang, Q. (2022). ROC Analyses Based on Measuring Evidence Using the Relative Belief Ratio. *Entropy*, 24(12), 1710. <https://doi.org/10.3390/e24121710>
- Anzer, G., & Bauer, P. (2021). A goal scoring probability model for shots based on synchronized positional and event data in football (SOCCER). *Frontiers in Sports and Active Living*, 3. <https://doi.org/10.3389/fspor.2021.624475>
- Artzi, N. S., Shilo, S., Hadar, E., Rossman, H., Barbash-Hazan, S., Ben-Haroush, A., Balicer, R. D., Feldman, B., Wiznitzer, A., & Segal, E. (2020). Prediction of gestational diabetes based on nationwide electronic health records. *Nature Medicine*, 26(1), 71–76. <https://doi.org/10.1038/s41591-019-0724-8>
- Bache-Mathiesen, L. K., Andersen, T. E., Dalen-Loretsen, T., Clarsen, B., & Fagerland, M. W. (2021). Not straightforward: Modelling non-linearity in training load and injury research. *BMJ Open Sport & Exercise Medicine*, 7(3), e001119. <https://doi.org/10.1136/bmjsem-2021-001119>
- Balkir, N., & A. El-Mouadib, F. (2021). Money Laundering Detection System Using Data Mining Functionalities. *The 7th International Conference on*

*Engineering & MIS* 2021. 12, 1–7.  
<https://doi.org/10.1145/3492547.3492583>

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2020). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54(3), 1937–1967. <https://doi.org/10.1007/s10462-020-09896-5>

Bill, G., Him, W., & Elly, B. (2024). *Benchmarking machine learning algorithms: A framework for evaluating efficiency in AI applications* [Unpublished article]. ResearchGate.  
<https://www.researchgate.net/publication/386451870>

Bisong, E. (2019). Introduction to Scikit-learn. *Building machine learning and deep learning models on Google Cloud Platform: A comprehensive guide for beginners* (pp. 215–229). Apress. [https://doi.org/10.1007/978-1-4842-4470-8\\_18](https://doi.org/10.1007/978-1-4842-4470-8_18)

Cavus, M., & Biecek, P. (2022). Explainable expected goal models for performance analysis in football analytics. *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)* (Vol. 45, pp. 1–9). IEEE.  
<https://doi.org/10.1109/dsaa54385.2022.10032440>

Chen, J., Muscoloni, A., Abdelhamid, I., Wu, Y., & Cannistraci, C. V. (2023). Generalizing the AUC-ROC for Unbalanced Data, Early Retrieval And Link Prediction Evaluation. *Preprints.org*.  
<https://doi.org/10.20944/preprints202209.0277.v2>

Chen, T., Xu, J., Ying, H., Chen, X., Feng, R., Fang, X., Zheng, Z., & Wu, J. (2019). Prediction of Extubation Failure For Intensive Care Unit Patients Using

- Light Gradient Boosting Machine. *IEEE Access*, 7, 150960–150968.  
<https://doi.org/10.1109/ACCESS.2019.2947260>
- Das, L., Ahuja, L., Chauhan, V., & Singh, A. (2022). A Review Of Data Warehousing Using Feature Engineering. *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, pp. 690–696.  
<https://doi.org/10.1109/ICIPTM54933.2022.9754137>
- Davis, J., & Robberechts, P. (2024). Biases In Expected Goals Models Confound Finishing Ability. *arXiv preprint arXiv:2401.09940*.
- Decroos, T., & Davis, J. (2019). Interpretable Prediction Of Goals In Soccer. *Proceedings of the AAAI-20 workshop on artificial intelligence in team sports*.
- Dimitriadis, T., Gneiting, T., Jordan, A. I., & Vogel, P. (2023). Evaluating Probabilistic Classifiers: The triptych. *International Journal of Forecasting*, 40(3), 1101–1122. <https://doi.org/10.1016/j.ijforecast.2023.09.007>
- Eggels, H., Van Elk, R., & Pechenizkiy, M. (2016). Explaining Soccer Match Outcomes With Goal Scoring Opportunities Predictive Analytics. *3rd Workshop on Machine Learning and Data Mining for Sports Analytics (MLSA 2016)*. CEUR-WS.org
- ElHabr, T. (2023). *XG model calibration*. Tony ElHabr – Tony’s Blog. Retrieved May 11, 2025, from <https://tonyelhabr.rbind.io/posts/opta-xg-model-calibration/>
- Ensum, J., Pollard, R., & Taylor, S. (2005). Applications Of Logistic Regression To Shots At Goal In Association Football. In *World congress on science and football* (Issue 5, pp. 211-218). Routledge.

- Fairchild, A., Pelechrinis, K., & Kokkodis, M. (2018). Spatial Analysis Of Shots In MLS: A Model For Expected Goals And Fractal Dimensionality. *Journal of Sports Analytics* (Vol. 4, Issue 3, pp. 165–174). IOS Press.  
<https://doi.org/10.3233/jsa-170207>
- Fikri, E., Sulistiawan, I. A., Riyanto, A., & Saputra, A. E. (2023). Neutralization of Acidity (Ph) And Reduction Of Total Suspended Solids (TSS) by Solar-Powered Electrocoagulation System. *Civil Engineering Journal*, 9(5), 1160–1172. <https://doi.org/10.28991/cej-2023-09-05-09>
- Filter, A., Olivares-Jabalera, J., Dos' Santos, T., Madruga, M., Lozano, J., Molina, A., ... & Loturco, I. (2023). High-Intensity Actions In Elite Soccer: Current Status And Future Perspectives. *International journal of sports medicine*, 44(08), 535-544.
- Florek, P., & Zagdański, A. (2023). Benchmarking State-Of-The-Art Gradient Boosting Algorithms For Classification. *arXiv preprint arXiv:2305.17094*.
- Fontanive, L. M. S. (2021). *Football analytics: A literature analysis from 2010 to 2020* [Disertasi Master, Universidade Nova de Lisboa].
- Foster, D. P., & Hart, S. (2022). “Calibeating”: Beating Forecasters At Their Own Game. *Theoretical Economics*, 18(4), 1441–1474.  
<https://doi.org/10.3982/te5330>
- García, S., Luengo, J., & Herrera, F. (2016). Tutorial On Practical Tips Of The Most Influential Data Preprocessing Algorithms In Data Mining . *Knowl. Based Syst.*, 98, 1-29. <https://doi.org/10.1016/j.knosys.2015.12.006>.

- Garg, A., Ayaan, M., Parekh, S., & Udandaraao, V. (2025). *Food Delivery Time Prediction In Indian Cities Using Machine Learning Models*. arXiv.org.  
<https://arxiv.org/abs/2503.15177>
- Gelman, A., Hill, J., & Vehtari, A. (2020). *Regression and other stories*. Cambridge: Cambridge University Press. <https://doi.org/10.1017/9781139161879>
- Géron, A. (2020). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition*. O'Reilly Online Library.  
<https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>
- Głównia, S., Kozak, J., & Juszczuk, P. (2023). Knowledge Discovery in Databases for a Football Match Result. *Electronics*, 12(12), 2712.  
<https://doi.org/10.3390/electronics12122712>
- Griesbach, C., Säfken, B. & Waldmann, E. (2021). Gradient Boosting For Linear Mixed Models. *The International Journal of Biostatistics*, 17(2), 317-329.  
<https://doi.org/10.1515/ijb-2020-0136>
- Guan, Y., & Haran, M. (2018). A Computationally Efficient Projection-Based Approach for Spatial Generalized Linear Mixed Models. *Journal of Computational and Graphical Statistics*, 27(4), 701–714.  
<https://doi.org/10.1080/10618600.2018.1425625>
- Gupta, C., & Ramdas, A. (2021). Distribution-free calibration guarantees for histogram binning without sample splitting. *Proceedings of the 38th International Conference on Machine Learning*, PMLR 139, 3942-3952.
- Gupta, P., & Bagchi, A. (2024). Introduction to Pandas. *Synthesis Lectures On Engineering, Science, and Technology* (pp. 161–196).  
[https://doi.org/10.1007/978-3-031-43725-0\\_5](https://doi.org/10.1007/978-3-031-43725-0_5)

- Han, J., Pei, J., & Tong, H. (2024). *Data mining: Concepts and techniques* (4th ed.). Morgan Kaufmann.
- Hartanto, A. D., Nur Kholik, Y., & Pristyanto, Y. (2023). Stock Price Time Series Data Forecasting Using the Light Gradient Boosting Machine (LightGBM) Model. *JOIV: International Journal on Informatics Visualization*, 7(4), 2270–2279. <https://doi.org/10.30630/joiv.7.4.1740>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer. <https://doi.org/10.1007/978-0-387-84858-7>
- Henzi, A., Ziegel, J. F., & Gneiting, T. (2021). Isotonic distributional regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(5), 963-993.
- Hu, X., Yin, S., Zhang, X., Menon, C., Fang, C., Chen, Z., Elgendi, M., & Liang, Y. (2023). Blood pressure stratification using photoplethysmography and light gradient boosting machine. *Frontiers in Physiology*, 14. <https://doi.org/10.3389/fphys.2023.1072273>
- Huang, J., & Chen, W. (2023). A study on category classification based on LightGBM for signal feature extraction and K-Means clustering. *2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, 858–862. <https://doi.org/10.1109/icpics58376.2023.10235522>
- Hudl. (2024, January 26). *Expected Goals (xG) Explained*. Hudl Blog. <https://www.hudl.com/blog/expected-goals-xg-explained>

- Hunt, J. (2019). Introduction to Matplotlib. *Advanced guide to Python 3 programming* (pp. 35–42). Springer International Publishing.  
[https://doi.org/10.1007/978-3-030-25943-3\\_5](https://doi.org/10.1007/978-3-030-25943-3_5)
- Hur, B. (2025), *Python-based Microcontroller Architecture and Microcontroller Application Education in Engineering Technology*. ASEE -GSW Annual Conference, Arlington, TX, Texas. <https://doi.org/10.18260/1-2--55073>
- Iapteff, L., Le Coz, S., Rioland, M., Houde, T., Carling, C., & Imbach, F. (2025). Toward Interpretable Expected Goals Modeling Using Bayesian Mixed Models. *Frontiers in Sports and Active Living*, 7, 1504362.
- Idris, H. I., Mohammed, A., Salisu, U. F., Balansana, K. I., Abdulazeez, D., & Danrimi, N. H. (2024). Evaluating The Performances Of Robust Logistic Regression Models In The Presence Of Outliers. *African Journal of Mathematics and Statistics Studies*, 7(4), 320–327.  
<https://doi.org/10.52589/AJMSS-YKDFCYQS>
- Indah, Y. M., Aristawidya, R., Fitrianto, A., Erfiani, E., & Jumansyah, L. R. D. (2025). Comparison of Random Forest, XGBOOST, and LightGBM methods for the Human Development Index Classification. *Jambura Journal of Mathematics*, 7(1), 14–18.  
<https://doi.org/10.37905/jjom.v7i1.28290>
- Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine Learning And Deep Learning. *Electronic Markets*, 31(3), 685–695.  
<https://doi.org/10.1007/s12525-021-00475-2>
- Katya, E. (2023). Exploring Feature Engineering Strategies for Improving Predictive Models in Data Science. *Research Journal of Computer Systems and Engineering*, 4(2), 201–215. <https://doi.org/10.52710/rjcse.88>

- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30).
- Kelly, Ryan (2019). *What is xG in football? How to calculate & what is a good score?* Diakses dari <https://www.goal.com/en/news/what-is-xg-in-football-how-to-calculate-what-is-a-good-score/1je6n626il5a7183n099j32t9t>
- Khrapach, V., & Siryi, O. (2024). Statistical Metric xG in Football and Its Impact on Scoring Performance: a Review Article. *Health Technologies*. <https://doi.org/10.58962/ht.2024.2.3.47-54>.
- Kodama, M., Takeuchi, A., Uesugi, M., & Hirai, S. (2023). Machine Learning Super-Resolution Of Laboratory CT Images In All-Solid-State Batteries Using Synchrotron Radiation CT As Training Data. *Energy and AI*, 14, 100305. <https://doi.org/10.1016/j.egyai.2023.100305>
- Kodati, S., & Selvaraj, J. (2021). Analysis of Heart Disorder by Using Machine Learning Methods and Data Mining Techniques. *Deep Learning Applications and Intelligent Decision Making in Engineering* (pp. 212-221). IGI Global Scientific Publishing.
- Liu, J., Wang, B., & Xiao, L. (2021). Non-Linear Associations Between Built Environment And Active Travel For Working And Shopping: An Extreme Gradient Boosting Approach. *Journal of Transport Geography*, 92, 103034.



- Lucey, P., Bialkowski, A., Monfort, M., Carr, P., & Matthews, I. (2015). Quality Vs Quantity: Improved Shot Prediction In Soccer Using Strategic Features From Spatiotemporal Data. *SPORTS ANALYTICS CONFERENCE*, 9. Disney Research. <http://disneyresearch.s3.amazonaws.com>
- Malagón-Selma, P. (2023). *Machine learning and multivariate statistical tools for football analytics* [Disertasi Doktoral, Universitat Politècnica de València].
- Malikov, D., & Kim, J. (2024). Beyond xG: A Dual Prediction Model for Analyzing Player Performance Through Expected and Actual Goals in European Soccer Leagues. *Applied Sciences*, 14(22), 10390.
- Manatova, D., & Woo, J. (2023). Machine Learning vs. Deep Learning on a Tabular Traffic Dataset in Big Data.
- McLean, S., Salmon, P. M., Gorman, A. D., Read, G. J. M., & Solomon, C. (2017). What's In A Game? A Systems Approach To Enhancing Performance Analysis In Football. *PLoS ONE*, 12(2), e0172565. <https://doi.org/10.1371/journal.pone.0172565>
- Mead, J., O'Hare, A., & McMenemy, P. (2023). Expected Goals In Football: Improving Model Performance And Demonstrating Value. *PLoS ONE*, 18(4), e0282295. <https://doi.org/10.1371/journal.pone.0282295>
- Méndez, M., Montero, C., & Núñez, M. (2023). Improving the Expected Goal Value in Football Using Multilayer Perceptron Networks. *Communications in Computer and Information Science* (pp. 352–363). Springer Nature Switzerland. [https://doi.org/10.1007/978-3-031-42430-4\\_29](https://doi.org/10.1007/978-3-031-42430-4_29)
- Mirkhan, A. (2022, March 8). *Image Recognition with Machine Learning: How and Why?* Kili Technology. <https://kili-technology.com/data-labeling/computer->

[vision/image-annotation/image-recognition-with-machine-learning-how-and-why](#)

Mishra, V. N., Kumar, V., Prasad, R., & Punia, M. (2021). Geographically Weighted Method Integrated With Logistic Regression For Analyzing Spatially Varying Accuracy Measures Of Remote Sensing Image Classification. *Journal of the Indian Society of Remote Sensing*, 49(5), 1189–1199. <https://doi.org/10.1007/s12524-020-01286-2>

Mohammed, M. A., Kadhem, S. M., & Maisa'a, A. A. (2021). Insider attacker detection using light gradient boosting machine. *Tech-Knowledge*, 1(1), 67-76.

Mohammed, M. A., Kadhem, S. M., & Maisa'a, A. A. (2021). Insider Attacker Detection Using Light Gradient Boosting Machine. *Tech-Knowledge*, 1(1), 67-76.

Molin, S., & Jee, K. (2021). *Hands-on data analysis with Pandas: A Python data science handbook for data collection, wrangling, analysis, and visualization*. Packt Publishing.

Mowbray, M., Del Rio Chanona, E., Harun, I., L, W. J., Hellgardt, K., & Zhang, D. (2020). Ensemble Learning For Bioprocess Dynamic Modelling And Prediction [Dataset]. In *Authorea*. <https://doi.org/10.22541/au.158456506.69710259>

Murphy, K. P. (2022). Probabilistic Machine Learning: An Introduction. MIT Press.

Nahm, F. S. (2022). Receiver Operating Characteristic Curve: Overview And Practical Use For Clinicians. *Korean Journal of Anesthesiology*, 75(1), 25–36. <https://doi.org/10.4097/kja.21209>

- Nemeth, M., Borkin, D., & Michalconok, G. (2019). The Comparison of Machine-Learning Methods XGBOOST and LightGBM to Predict Energy Development. In *Advances in Intelligent Systems And Computing* (pp. 208–215). [https://doi.org/10.1007/978-3-030-31362-3\\_21](https://doi.org/10.1007/978-3-030-31362-3_21)
- Olvera-Rojas, M., Femia-Marzo, P., & Castillo-Rodríguez, A. (2023). Scoring First Relevance In Knockout Promotion To Spanish Laliga Smartbank. *Journal of Human Sport and Exercise*, 18(1), 11-20. <https://doi.org/10.14198/jhse.2023.181.02>
- Otter, D. W., Medina, J. R., & Kalita, J. K. (2021). A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2), 604–624. <https://doi.org/10.1109/TNNLS.2020.2979670>
- Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep Learning For Financial Applications : A Survey. *Applied Soft Computing*, 93, 106384. <https://doi.org/10.1016/j.asoc.2020.106384>
- PanditFootball. (2022). *Mengupas Statistik Expected Goals (xG)*. Diakses dari <https://panditfootball.com/sains-bola/213838/PFB/220813/mengupas-statistik-expected-goals-xg>
- Patel, H. I., & Patel, D. (2024). Exploratory Data Analysis And Feature Selection For Predictive Modeling Of Student Academic Performance Using A Proposed Dataset. *International Journal of Engineering Trends and Technology*, 72(11), 131–143. <https://doi.org/10.14445/22315381/ijett-v72i11p116>

- Paucar, I. R., & Andrade-Arenas, L. (2025). Public Health Challenges In The Cuzco Region: A Decade Of Anemia In Vulnerable Populations Applying Data Mining. *Bulletin of Electrical Engineering and Informatics*, 14(3), 2171-2187.
- Pendrill, L. R., Melin, J., Stavelin, A., & Nordin, G. (2023). Modernising Receiver Operating Characteristic (ROC) Curves. *Algorithms*, 16(5), 253. <https://doi.org/10.3390/a16050253>
- Petropoulos, A., Siakoulis, V., & Stavroulakis, E. (2022). Towards An Early Warning System For Sovereign Defaults Leveraging On Machine Learning Methodologies. *Intelligent Systems in Accounting Finance & Management*, 29(2), 118–129. <https://doi.org/10.1002/isaf.1516>
- Ramadanti, E. ., Aprilya Dinathi, D. ., Christianskaditya, & Chandranegara, D. R. . (2024). Diabetes Disease Detection Classification Using Light Gradient Boosting (LightGBM) With Hyperparameter Tuning. *Sinkron: Jurnal Dan Penelitian Teknik Informatika*, 8(2), 956-963. <https://doi.org/10.33395/sinkron.v8i2.13530>
- Ramos, S., Soares, J., Cembranel, S. S., Tavares, I., Foroozandeh, Z., Vale, Z., & Fernandes, R. (2021). Data Mining Techniques For Electricity Customer Characterization. *Procedia Computer Science*, 186(3), 475–488. <https://doi.org/10.1016/j.procs.2021.04.168>
- Regin, R., & Rajest, S. S. (2024). Comprehensive Exploratory Data Analysis of the Netflix Dataset: Uncovering Viewer Preferences and Content Trends. *CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES*, 5(4), 388-400.

- Ruiz, H., Lisboa, P., Neilson, P., & Gregson, W. (2015). Measuring Scoring Efficiency Through Goal Expectancy Estimation. In *ESANN 2015 proceedings of the European symposium on artificial neural networks, computational intelligence and machine learning* (pp. 149-154).
- Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
- Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications And Research Directions. *SN Computer Science*, 2(3), 1-21. <https://doi.org/10.1007/s42979-021-00592-x>
- Sarmiento, H., Clemente, F. M., Araújo, D., Davids, K., McRobert, A., & Figueiredo, A. (2018). What Performance Analysts Need To Know About Research Trends in Association Football (2012–2016): a systematic review. *Sports Medicine*, 48(4), 799–836. <https://doi.org/10.1007/s40279-017-0836-6>
- Scholtes, A., & Karakuş, O. (2024). Bayes-xG: Player And Position Correction On Expected Goals (Xg) Using Bayesian Hierarchical Approach. *Frontiers In Sports And Active Living*, 6, 1348983. <https://doi.org/10.3389/fspor.2024.1348983>
- Senapati, A. (2023). Correlation Coefficient-Based Breakpoint Detection Piecewise Linear Regression. *Research Square (Research Square)*. <https://doi.org/10.21203/rs.3.rs-2917422/v1>
- Senn, W. (2024). *Addressing Evaluation Challenges On The Expected Goals (xG) Metric In Football Analysis* [Bachelor's Thesis, Haute école de gestion de Genève]. SONAR. <https://sonar.ch/global/documents/331251>

- Sheridan, R. P., Liaw, A., & Tudor, M. (2021). Light Gradient Boosting Machine as a Regression Method For Quantitative Structure-Activity Relationships. *arXiv preprint*. <https://arxiv.org/abs/2105.08626>
- Shwartz-Ziv, R., & Armon, A. (2021). Tabular Data: Deep Learning Is Not All You Need. *Information Fusion*, 81, 84–90. <https://doi.org/10.1016/j.inffus.2021.11.011>
- Solanki, A., & Sharma, S. K. (2023). Analytical Study Of Machine Learning Techniques On The Smart Home Energy Consumption. *AIP Conference Proceedings*, 2977, 020008. <https://doi.org/10.1063/5.0184775>
- StatsBomb. (2022). *StatsBomb open data*. GitHub. Diakses pada 28 Oktober, 2024, dari <https://github.com/statsbomb/open-data>
- StatsBomb. (2024). *Who are we?*. Diakses pada 28 Oktober, 2024, dari <https://statsbomb.com/who-we-are>
- Tan, P.-N., Steinbach, M., & Kumar, V. (2019). *Introduction to Data Mining* (2nd ed.). Pearson. <https://www.pearson.com/en-us/subject-catalog/p/introduction-to-data-mining/P2000000003204/>
- Tharwat, A. (2021). Classification assessment methods. *Applied Computing and Informatics*, 17(1/2), 168-192. <https://doi.org/10.1016/j.aci.2018.08.003>
- Tureen, T., & Olthof, S. (2022, October 5). *Estimated Player Impact (Epi): Quantifying The Effects Of Individual Players On Football (Soccer) Actions Using Hierarchical Statistical Models* [Paper presentation]. StatsBomb Conference 2022, London, England. <https://statsbomb.com/wp-content/uploads/2022/10/Estimated-Player-Impact-EPI.pdf>

- Türkmen, G., & Sezen, A. (2024). A Comparative Analysis of XGBOOST and LightGBM Approaches for Human Activity Recognition: Speed and Accuracy evaluation. *International Journal of Computational and Experimental Science and Engineering*, 10(2).  
<https://doi.org/10.22399/ijcesen.329>
- Ustimenko, A., Prokhorenkova, L.. (2021). SGLB: Stochastic Gradient Langevin Boosting. *Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research* 139:10487-10496  
<https://proceedings.mlr.press/v139/ustimenko21a.html>.
- Vaddella, R., & Hosseinzadeh, M. (2021). Naive Bayes: Applications, Variations And Vulnerabilities: A Review Of Literature With Code Snippets For Implementation. *Journal of Big Data*, 8(1), 1-21.  
<https://doi.org/10.1186/s40537-021-00529-6>
- Verdonck, T., Baesens, B., Óskarsdóttir, M., & Broucke, S. (2021). Special Issue On Feature Engineering Editorial. *Mach Learn*, 113(7), 3917-3928.  
<https://doi.org/10.1007/s10994-021-06042-2>.
- Waskom, M. L. (2021). Seaborn: Statistical Data Visualization. *Journal of Open Source Software*, 6(60), 1–4. <https://doi.org/10.21105/joss.03021>
- Whitmore, J. (2023). *What Is Expected Goals (xG)?* Opta Analyst. Retrieved October 27, 2024, <https://theanalyst.com/eu/2023/08/what-is-expected-goals-xg/>

- Wozniakowski, A., Thompson, J., Gu, M., & Binder, F. C. (2021). A New Formulation of Gradient Boosting. *Machine Learning Science and Technology*, 2(4), 045022. <https://doi.org/10.1088/2632-2153/ac1ee9>
- Xie, B., Zhu, C., Zhao, L., & Zhang, J. (2022). A Gradient Boosting Machine-Based Framework For Electricity Energy Knowledge Discovery. *Frontiers in Environmental Science*, 10. <https://doi.org/10.3389/fenvs.2022.1031095>
- Zaqy, M. J., Mentari, R., & Iqbal, M. (2023). Implementasi Data Mining Untuk Memprediksi Penjualan Toko Roti Mawar Menggunakan Regresi Linier Berganda. *J-SISKO TECH*, 6(2). <https://doi.org/10.53513/jsk.v6i2.8534>
- Zhang, Y., Song, Z., Lin, Y., Shi, Q., Hao, Y., Fu, Y., Wu, J., & Zhang, Z. (2023). Predicting Mechanical Properties of CO<sub>2</sub> Hydrates: Machine Learning Insights from Molecular Dynamics Simulations. *Journal of Physics Condensed Matter*, 36(1), 015101. <https://doi.org/10.1088/1361-648x/acfa55>
- Zhao, X., Liu, Y., & Zhao, Q. (2023). Cost Harmonization LightGBM-Based Stock Market Prediction. *IEEE Access*, 11, 105009–105026. <https://doi.org/10.1109/access.2023.3318478>



## LAMPIRAN

```

import os
import pandas as pd
import numpy as np
import lightgbm as lgb
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
from sklearn.model_selection import GridSearchCV,
cross_validate, RandomizedSearchCV
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import FunctionTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_auc_score, log_loss,
make_scorer, brier_score_loss
from sklearn.metrics import precision_score, recall_score,
f1_score, log_loss, mean_absolute_error
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from lightgbm import LGBMClassifier
from joblib import dump, load
from sklearn.calibration import calibration_curve
from sklearn.calibration import CalibratedClassifierCV
from sklearn.inspection import permutation_importance
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import spearmanr, randint as sp_randint,
uniform as sp_uniform
from scipy.cluster import hierarchy
df.dropna(inplace=True)
print(f"Missing Values: {df.shape}")
df.drop_duplicates(inplace=True)
print(f"Data Duplikat: {df.shape}")
print(f"Total Data Setelah Data Cleansing: {df.shape[0]}")
df['goal_direction'] = np.where(df['end_x'] >= 120,
'right_goal', 'left_goal')
def calculate_distance_and_angle(row):
    if row['goal_direction'] == 'right_goal':
        distance = np.sqrt((row['start_x'] - 120) ** 2 +
(row['start_y'] - 40) ** 2)
        angle = np.arctan2((row['start_y'] - 40),
(row['start_x'] - 120))

```

```

        else:
            distance = np.sqrt((row['start_x'] - 0) ** 2 +
                                (row['start_y'] - 40) ** 2)
            angle = np.arctan2((row['start_y'] - 40),
                                (row['start_x'] - 0))

            return pd.Series([distance, angle])
df[['distance_to_goal', 'angle_to_goal']] =
df.apply(calculate_distance_and_angle, axis=1)
numerical_cols = ['minute', 'second', 'start_x', 'start_y',
                  'distance_to_goal', 'angle_to_goal']
categorical_cols = [
    'play_pattern', 'position', 'shot_technique',
    'shot_body_part',
    'shot_type'
]
binary_cols = ['shot_open_goal',
               'shot_one_on_one', 'shot_aerial_won', 'shot_outcome']

df[numerical_cols] = df[numerical_cols].astype('float32')
df[categorical_cols] = df[categorical_cols].astype('int32')
df[binary_cols] = df[binary_cols].astype('int32')
train_df, test_df = train_test_split(df, test_size=0.1,
                                      random_state=seed)
X_train = train_df.drop(['shot_outcome', 'end_x', 'end_y'],
                        axis=1)
y_train = train_df['shot_outcome']
X_test = test_df.drop(['shot_outcome', 'end_x', 'end_y'],
                      axis=1)
y_test = test_df['shot_outcome']
lgbm = LGBMClassifier(boosting_type='goss', random_state=42)
lgbm_param_grid = {
    'min_child_samples': sp_randint(0, 200),
    'num_leaves': sp_randint(2, 500),
    'reg_lambda': sp_uniform(0, 1),
    'reg_alpha': sp_uniform(0, 1),
    'max_depth': sp_randint(0, 500),
}
searchcv = RandomizedSearchCV(estimator=lgbm,
                              n_iter=100,
                              param_distributions=lgbm_param_g
rid,
                              cv=5,
                              scoring=scoring,
                              refit='roc_auc',

```

```

n_jobs=-1)

start_time = time.time()
searchcv.fit(X_train, y_train)
end_time = time.time()
execution_time = end_time - start_time
print(f"Waktu eksekusi RandomizedSearchCV:
{execution_time:.2f} detik")
calibration_start = time.time()
best_lgbm = searchcv.best_estimator_
calibrated_lgbm = CalibratedClassifierCV(searchcv,
method='isotonic', cv='prefit')
calibrated_lgbm.fit(X_train, y_train)
best_model = calibrated_lgbm
calibration_end = time.time()
calibration_time = calibration_end - calibration_start
print(f"Waktu kalibrasi model: {calibration_time:.2f} detik")
total_time = execution_time + calibration_time
print(f"Total waktu dari pencarian hingga kalibrasi:
{total_time:.2f} detik")
print('Brier score, Light GBM:', brier_score_loss(y_test,
y_pred_lgbm_calibrated, pos_label=y_test.max()))
print('ROC AUC, Light GBM:', roc_auc_score(y_test,
y_pred_lgbm_calibrated))
print('Pseudo R-squared, Light GBM:', mcfadden_r2(y_test,
y_pred_lgbm_calibrated))
y_pred_labels = (y_pred_lgbm_calibrated >= 0.5).astype(int)
precision = precision_score(y_test, y_pred_labels,
average='weighted')
recall = recall_score(y_test, y_pred_labels,
average='weighted')
f1 = f1_score(y_test, y_pred_labels, average='weighted')
logloss = log_loss(y_test, y_pred_lgbm_calibrated)
mae = mean_absolute_error(y_test, y_pred_lgbm_calibrated)
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1 Score: {f1:.4f}")
print(f"Log-Loss: {logloss:.4f}")
print(f"Mean Absolute Error: {mae:.4f}")
accuracy = accuracy_score(y_test, y_pred_labels)
print(f"Accuracy: {accuracy:.4f}")

```