BAB 3

METODOLOGI PENELITIAN

3.1 Pendekatan Penelitian

Penelitian ini melihat kinerja metode LGBM 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 Tempat dan Waktu Penelitian

3.2.1 Tempat Penelitian

Penelitian ini dilakukan menggunakan *open-data* dari StatsBomb melalui repositori GitHub dengan proses analisis yang dilakukan menggunakan Python dan platform Google Colaboratory.

3.2.2 Waktu Penelitian

Rencana waktu pelaksanaan penelitian ditunjukkan pada Tabel 3.1.

Tabel 3.1 Waktu Pelaksanaan Penelitian

No.	Tahapan	Februari 2025	Maret 2025	April 2025	Mei 2025	Juni 2025	Juli 2025
1	Landasan Teori						

2	Pengumpulan Data						
3	Analisis Data						
4	Interpretasi						
5	Pembuatan Laporan						

3.3 Metodologi Pengumpulan Data

3.3.1 Studi Literatur

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.

3.3.2 Pengambilan Data

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.

3.4 Pengembangan Model

3.4.1 Metode KDD

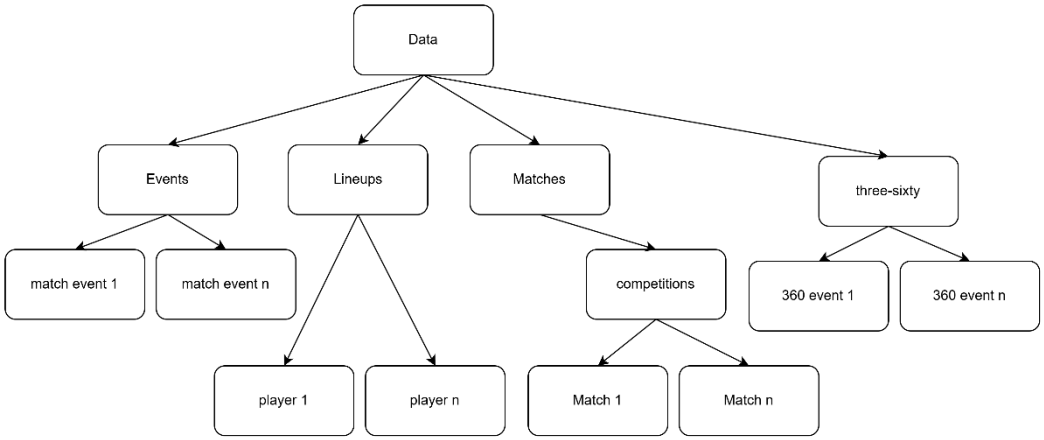
Penelitian ini menggunakan pendekatan *Knowledge Discovery in Databases* (KDD) dalam proses pengembangan model. Metode KDD memiliki keunggulan dalam membantu mengidentifikasi pola tersembunyi dari kumpulan data yang kompleks sehingga dapat menghasilkan informasi yang lebih mudah dipahami. Proses KDD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

a. *Data Selection*

Data dari StatsBomb *open-data* diambil dengan mengakses repositori resmi di GitHub. Pertama, kita perlu mengidentifikasi kompetisi apa saja yang tersedia dalam *dataset*. Setiap kompetisi kemudian terdiri dari beberapa musim (edisi), dan masing-masing musim ini mewakili rentang waktu berlangsungnya pertandingan yang terdokumentasi. Di dalam setiap musim terdapat fase-fase pertandingan: untuk kompetisi sistem gugur biasanya meliputi babak perempat final, semi final, final, dan seterusnya, sedangkan untuk liga reguler umumnya hanya ada satu fase liga utama, dengan beberapa kompetisi seperti, Piala FA yang juga memiliki babak *play-off*. Setelah fase-fase ditentukan, barulah kita mengakses data pertandingan. Dalam konteks StatsBomb, satu pertandingan terdiri dari serangkaian *event*, dan masing-masing *event* ini dapat memiliki *event* terkait.

Misalnya, sebuah tusukan (*dribble*) bisa jadi dipicu oleh operan rekan tim yang sebelumnya dieksekusi operan tersebut, kemudian tercatat sebagai *event* terkait. Namun, karena operan juga tercatat sebagai *event* utama, jika kita menarik semua *event* terkait tanpa seleksi, kita akan mendapati banyak duplikasi operan tercatat dua kali, sekali sebagai *event* utama dan sekali lagi sebagai *event* terkait. Sebaliknya, jika kita sama sekali mengabaikan *event* terkait, kita bisa kehilangan jejak kronologi aksi yang sebenarnya terjadi di lapangan.

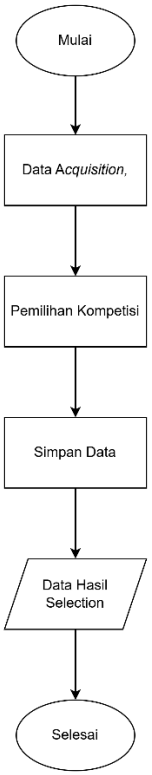
Untuk mengatasi masalah ini, saat ini hanya situasi gol dan kartu (kuning/merah) yang diikutkan sebagai *event* terkait dalam pemrosesan data StatsBomb. Dengan begitu, kita tetap menjaga konteks penting seperti *assist* sebelum gol atau pelanggaran yang berujung kartu tanpa menumpuk terlalu banyak duplikasi. Pada Gambar 3.1 dijelaskan struktur data yang dimiliki oleh StatsBomb *open-data.*



Gambar 3.1 Struktur Data StatsBomb *open-data.*

Data *event* dari StatsBomb disediakan dalam format JSON pada repositori GitHub mereka. Karena pengambilan data langsung dari GitHub juga memakan

waktu, biasanya file-file JSON tersebut diunduh sekali saja lalu dikonversi dan disimpan dalam format *Parquet* untuk penggunaan selanjutnya. Dengan cara ini, analisis bisa dilakukan lebih cepat tanpa perlu terus-menerus mengunduh data mentah. Gambar 3.2 menunjukkan *flowchart* dari *data selection.*



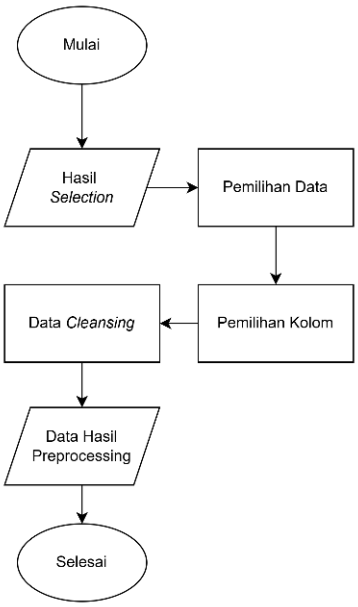
Gambar 3.2 *Flowchart Selection*

b. *Data Preprocessing*

Tahap data *preprocessing* bertujuan untuk menyiapkan data hasil seleksi agar dapat dianalisis secara optimal dan digunakan dalam proses pelatihan model. Proses ini disesuaikan dengan karakteristik data *event* sepak bola yang diperoleh dari StatsBomb, yang memiliki struktur sangat baik dan konsisten sehingga mempermudah proses pembersihan dan pengolahan data.

Langkah pertama adalah pemilihan data, yaitu dengan mengambil hanya *event* yang bertipe *Shot* dan berasal dari situasi permainan terbuka (*open play*), karena jenis tembakan ini paling relevan dalam konteks prediksi *expected goals*. Setelah itu, dilakukan pemilihan kolom dengan memilih fitur-fitur yang berpotensi mendukung prediksi, seperti posisi tembakan, bagian tubuh yang digunakan, tekanan lawan, serta pola permainan. Kolom-kolom yang bersifat administratif atau tidak relevan terhadap tujuan model, seperti nama pemain dan identifikasi pertandingan, tidak disertakan.

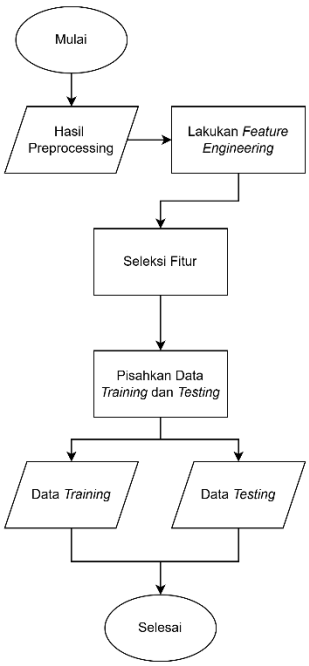
Langkah terakhir adalah data *cleansing*, yang meliputi pemeriksaan nilai kosong dan duplikat. Namun, karena data StatsBomb memiliki format yang sangat terstruktur dan tiap peristiwa dalam pertandingan bersifat unik, data yang diperoleh relatif bersih dan tidak memerlukan proses pembersihan lanjutan. Tahapan *preprocessing* ini menghasilkan *dataset* yang konsisten, bebas duplikasi, dan siap untuk dianalisis lebih lanjut pada tahap transformasi dan pemodelan. Pada Gambar 3.2 dijelaskan *flowchart* dari tahap *preprocessing.*



Gambar 3.3 *Flowchart Preprocessing*

c. *Data Transformation*

Tahap ini bertujuan untuk memperkaya representasi data agar dapat meningkatkan performa model pada tahapan *data mining*. Pertama, dilakukan proses *feature engineering* untuk menciptakan fitur-fitur baru yang merepresentasikan dinamika permainan secara lebih mendalam. Transformasi ini memungkinkan data mentah memberikan wawasan yang lebih bermakna dan relevan dalam konteks prediksi performa tembakan. Fitur-fitur seperti jarak dan sudut tembakan ke gawang serta segmentasi waktu pertandingan ditambahkan untuk memperkaya informasi spasial dan temporal. Setelah fitur baru ditambahkan, data kemudian dibagi menjadi data latih dan data uji agar proses pelatihan dan evaluasi model dapat dilakukan secara terpisah. Alur tahapan *transformation* ditunjukkan pada Gambar 3.3.

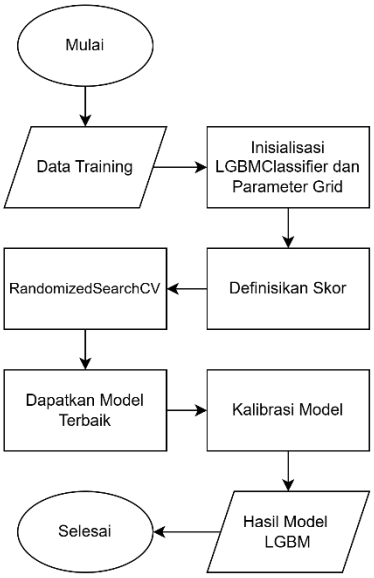


Gambar 3.4 *Flowchart Transformation*

d. Data *Mining*

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

akurat (Davis & Robberechts, 2024). Selain pelatihan dan kalibrasi, tahap ini juga mencakup analisis fitur untuk memahami kontribusi tiap variabel dalam proses prediksi. Gambar 3.4 menunjukkan alur dari tahapan data *mining* dalam penelitian ini.



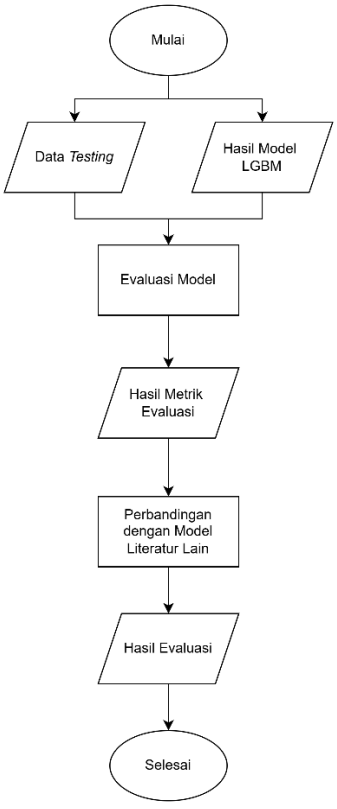
Gambar 3.5 *Flowchart* Data *Mining*

e. *Evaluation*

Setelah proses *data mining* selesai, tahap selanjutnya adalah evaluasi terhadap model yang telah dibuat. Evaluasi ini bertujuan untuk mengukur performa model secara komprehensif terhadap data uji. Sesuai dengan batasan masalah, evaluasi kinerja model akan menggunakan serangkaian metrik yang mencakup ROC AUC, *Brier Score*, presisi, *recall*, F1-Score, dan Log-Loss.

ROC AUC digunakan untuk menilai kemampuan diskriminatif model, yaitu kemampuannya dalam membedakan antara kelas positif dan negatif secara keseluruhan tanpa terikat pada ambang batas klasifikasi tertentu. Untuk

mengukur akurasi dari prediksi probabilistik, digunakan *Brier Score* yang menghitung rata-rata selisih kuadrat antara probabilitas prediksi dengan hasil aktual, sehingga efektif dalam menilai kalibrasi model. Serupa dengan itu, *Log-Loss* juga memberikan penalti untuk prediksi yang tingkat keyakinannya tidak sesuai dengan hasil aktual. Terakhir, untuk evaluasi yang lebih bernuansa pada tugas klasifikasi, digunakan 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 3.5.



Gambar 3.6 *Flowchart Evaluation*

3.4.2 Permodelan LightGBM

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

Teknik GOSS berfokus pada efisiensi pelatihan model dengan mempertahankan seluruh data yang memiliki nilai gradien besar yang mengandung lebih banyak informasi dan secara acak mengambil sebagian dari data dengan gradien kecil (Ke *et al*., 2017). Namun, karena proses ini dapat mengubah distribusi data asli, LightGBM memperkenalkan pengali konstan saat menghitung *information* gain untuk data dengan gradien kecil guna menyeimbangkan kontribusi antara dua kelompok data tersebut. Pendekatan ini memungkinkan model untuk tetap fokus pada sampel yang paling berpengaruh terhadap pembaruan model tanpa kehilangan akurasi secara signifikan.

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

LightGBM sangat efisien dan *scalable* dalam membangun model prediksi dari *dataset* dengan jumlah *instance* dan fitur yang sangat besar.

3.5 Analisis Data dan Intrepretasi Hasil

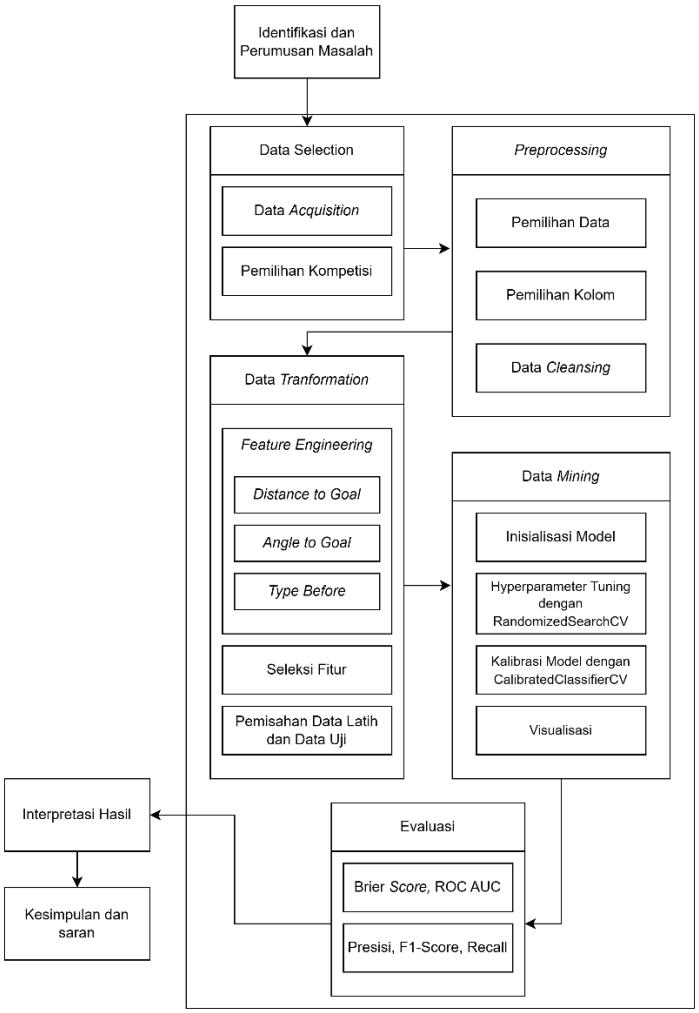
Analisis data dalam penelitian ini dilakukan berdasarkan pendekatan KDD (*Knowledge Discovery in Database*) yang mencakup lima tahapan utama: data *selection, preprocessing, transformation,* data *mining*, dan *evaluation*. Proses analisis dimulai dari tahap data *selection*, yaitu dengan menyiapkan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

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

Kemudian, pada tahap *evaluation*, performa model diukur secara komprehensif menggunakan serangkaian metrik. Metrik-metrik tersebut meliputi ROC AUC, Brier Score, presisi, *recall*, F1-*Score*, dan *Log-Loss* untuk menilai performa model dari berbagai aspek, mulai dari kemampuan diskriminatif hingga akurasi probabilistik. Hasil dari seluruh tahapan analisis ini serta interpretasi terhadap performa model akan dijelaskan secara rinci pada Bab 4.

3.6 Tahapan Penelitian

Tahapan penelitian yang dilakukan dapat dilihat pada Gambar 3.7.



Gambar 3.7 Tahapan Penelitian

Tahapan pertama dalam penelitian ini adalah identifikasi dan perumusan masalah yang dilakukan dengan mengkaji literatur yang relevan untuk memahami permasalahan aktual dan peluang riset di bidang analisis pertandingan sepak bola menggunakan teknik pembelajaran mesin. Literatur yang digunakan berasal dari

jurnal ilmiah, artikel konferensi, serta laporan penelitian terkini. Setelah permasalahan dirumuskan, penelitian dilanjutkan dengan mengikuti alur metode KDD yang terdiri dari tahapan data *selection, preprocessing, transfor*mation, data *mining*, dan *evaluation*.

Pada tahap data *selection*, data dikumpulkan dengan mengunduh *dataset* publik dari GitHub yang berisi catatan pertandingan sepak bola, lalu difokuskan pada data *event* yang relevan untuk keperluan prediksi. Selanjutnya, dilakukan *preprocessing* berupa pemilihan variabel yang akan digunakan, pembersihan data dari nilai yang hilang dan duplikat, proses *encoding* untuk fitur kategorial, serta normalisasi data numerik.

Tahap berikutnya adalah *transformation* yang mencakup rekayasa fitur (*feature engineering*), analisis korelasi antar variabel, seleksi fitur penting, dan pembagian data menjadi set pelatihan dan pengujian. Pada tahap data *mining*, digunakan algoritma LightGBM dengan proses *tuning hyperparameter* melalui *RandomizedSearchCV*, serta dilanjutkan dengan proses kalibrasi model agar hasil prediksi probabilistik menjadi lebih akurat.

Tahap terakhir adalah *evaluation*, yaitu evaluasi performa model secara komprehensif menggunakan serangkaian metrik yang mencakup ROC AUC, *Brier Score*, presisi, *recall*, F1-Score, dan *Log-Loss* untuk menilai berbagai aspek kinerja model mulai dari kemampuan diskriminatif hingga akurasi probabilistik. Setelah seluruh tahapan metode KDD selesai, penelitian diakhiri dengan analisis hasil dan penarikan kesimpulan serta saran untuk penelitian selanjutnya.

3.7 Perangkat Penelitian

Penelitian ini menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*) dengan spesifikasi yang dijelaskan pada Tabel 3.2.

Tabel 3.2 Spesifikasi Hardware dan Software

Hardware	Laptop Lenovo ADA 11	AMD Athlon Gold 3150U *with* Radeon Graphics 2.40 GHz
		12.0 GB RAM
		256 GB SSD
		Monitor 15 Inch
Software	Sistem Operasi	Windows 11 Home
	Tools	Google Colaboratory
	Bahasa Pemrograman	Python