

	period	minute	second	location_x	...	technique
0	1	1	42	111.0	...	93
1	1	4	47	96.0	...	93
2	1	8	37	107.0	...	93
3	1	17	26	111.0	...	93
4	1	21	16	105.0	...	93
...
68863	2	65	39	106.1	...	93
68864	2	69	0	114.9	...	93
68865	2	82	41	103.6	...	93
68866	2	85	10	108.5	...	93
68867	2	85	58	117.0	...	91

Gambar 4.2 Contoh Data Sesudah Pemilihan Variabel

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.

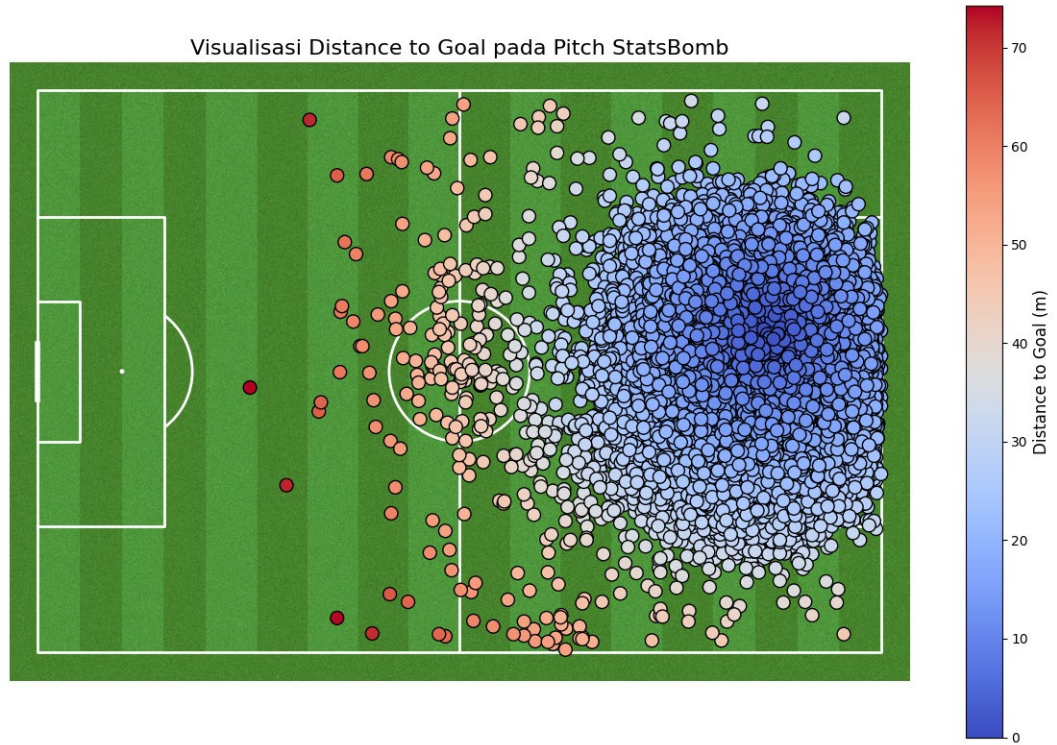
4.3 Data Transformation

4.3.1 Feature Engineering

Tahapan pertama dalam proses *transformation* pada penelitian ini adalah melakukan *feature engineering* dengan menambahkan tiga fitur baru, yaitu jarak dan sudut tembakan terhadap gawang serta kejadian sebelum terjadinya *shot*. Fitur ini ditambahkan untuk memberikan informasi spasial yang lebih kaya kepada model, mengingat lokasi dan sudut tembakan serta momentum sangat berpengaruh terhadap kemungkinan terciptanya gol.

a. Distance to Goal

Fitur pertama dalam proses *feature engineering* adalah menghitung jarak antara posisi tembakan dan pusat gawang. Informasi spasial ini penting karena jarak tembakan merupakan salah satu faktor utama yang memengaruhi kemungkinan terciptanya gol. Semakin dekat jarak tembakan ke gawang, secara umum peluang untuk mencetak gol menjadi lebih besar. Gambar 4.4 menunjukkan visualisasi fitur *distance to goal* pada lapangan pertandingan berdasarkan koordinat StatsBomb. Titik-titik pada visualisasi merepresentasikan lokasi awal tembakan, dengan warna yang menunjukkan jaraknya terhadap gawang, semakin biru berarti semakin dekat dan semakin merah berarti semakin jauh.



Gambar 4.3 Visualisasi *Distance to Goal*

Dalam *dataset* ini, koordinat pusat gawang StatsBomb berada pada titik ($x = 104.0$, $y = 34.0$), yang merepresentasikan titik tengah di antara dua tiang gawang. Jarak dihitung menggunakan rumus *Euclidean distance*, yaitu akar kuadrat dari jumlah kuadrat selisih antara koordinat tembakan dan koordinat pusat gawang. Secara matematis, perhitungan ini dinyatakan sebagai berikut:

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

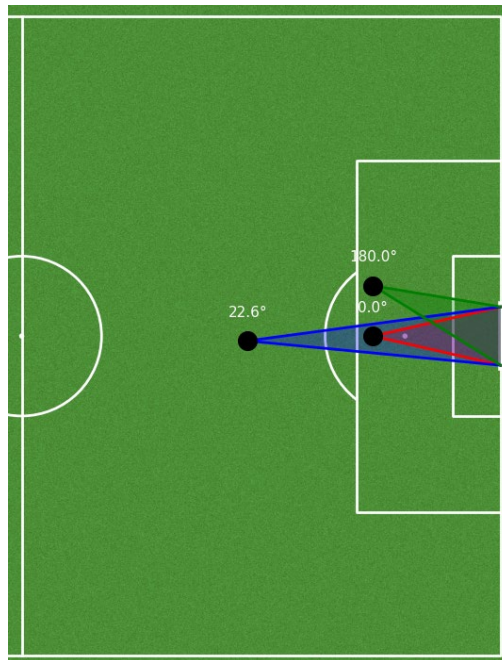
Fungsi ini diimplementasikan dalam kode Python yang akan mengembalikan nilai jarak dalam satuan relatif terhadap sistem koordinat StatsBomb. Hasil perhitungan disimpan dalam kolom baru bernama *distance_to_goal* dan digunakan sebagai salah satu fitur masukan dalam model prediksi. Gambar 4.7 menunjukkan contoh hasil dari proses penambahan fitur ini.

distance_to_goal
15.448625
14.905368
26.828716
10.837435
19.568598

Gambar 4.4 *Distance to Goal*

b. *Angle to Goal*

Selain jarak, fitur penting lainnya yang ditambahkan dalam proses *feature engineering* adalah sudut tembakan terhadap gawang, atau dikenal sebagai *open play angle*. Fitur ini merepresentasikan seberapa besar ruang terbuka yang tersedia bagi penembak untuk mengarahkan bola ke area di antara kedua tiang gawang. Semakin lebar sudut yang terbuka, semakin besar peluang tembakan untuk menghasilkan gol.



Gambar 4.5 Visualisasi Sudut Tembakan

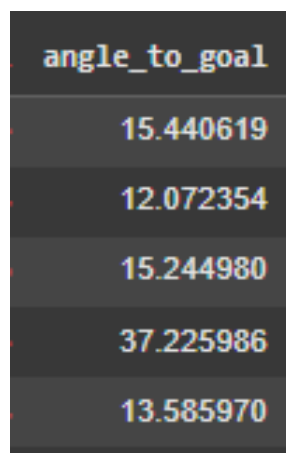
Perhitungan sudut dilakukan dengan mengacu pada tiga titik, posisi tembakan dan dua titik tiang gawang (kanan dan kiri). Dalam sistem koordinat StatsBomb, gawang terletak pada posisi horizontal tetap yaitu $x = 120$, dengan tiang bawah (kanan) berada pada $y = 43,66$ dan tiang atas (kiri) pada $y = 36,34$. Fungsi python digunakan untuk menghitung besar sudut terbuka menggunakan hukum cosinus. Langkah-langkahnya meliputi:

- i) Hitung jarak dari posisi tembakan ke masing-masing tiang gawang (A dan B).
- ii) Hitung panjang sisi antara kedua tiang (C).
- iii) Gunakan hukum cosinus untuk mencari sudut di antara kedua sisi tersebut.

Sudut dalam radian dikonversi ke derajat menggunakan fungsi `np.degrees()`.

Jika tembakan dilakukan tepat dari titik tengah gawang ($x = 120$ dan y berada

antara dua tiang), maka sudut maksimal akan diberikan sebesar 180 derajat. Sebaliknya, jika posisi tembakan berada sejajar secara horizontal dengan gawang tetapi tidak dalam rentang vertikal antara tiang, maka sudut dianggap 0 derajat. Hasil perhitungan ini disimpan dalam kolom *angle_to_goal*, yang menjadi input penting dalam proses pemodelan. Gambar 4.5 menunjukkan hasil dari *feature engineering* tersebut.



angle_to_goal
15.440619
12.072354
15.244980
37.225986
13.585970

Gambar 4.6 *Angle to Goal*

c. *Type Before*

Fitur *type_before* ditambahkan sebagai bagian dari proses *feature engineering* untuk memberikan konteks temporal terhadap peristiwa tembakan yang dianalisis. Fitur ini merepresentasikan jenis *event* yang terjadi tepat sebelum tembakan dilakukan, dengan mengambil nilai *type.id* dari *event* sebelumnya dalam urutan kronologis pertandingan. Informasi ini bertujuan untuk menangkap dinamika permainan yang mendahului tembakan, seperti apakah tembakan tersebut terjadi setelah dribel, operan, intersepsi, atau aksi defensif lawan. Tabel 4.3 beberapa *type.id* umum dalam data, yang digunakan untuk mengidentifikasi jenis peristiwa dalam pertandingan sepak bola.

Tabel 4.3 Deskripsi Jenis *type* dalam Pertandingan Sepak Bola.

Event Type	Type ID	Deskripsi Singkat
<i>50/50</i>	33	Dua pemain dari tim berbeda berebut bola lepas.
<i>Bad Behaviour</i>	24	Pelanggaran di luar permainan yang berujung kartu.
<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 yang mengarah pada tembakan lawan.
<i>Foul Committed</i>	22	Pelanggaran yang dilakukan terhadap lawan (tidak termasuk <i>offside</i>).
<i>Foul Won</i>	21	Pelanggaran yang diterima dan menghasilkan tendangan bebas atau penalti.
<i>Goal Keeper</i>	23	Segala aksi penjaga gawang (penyelamatan, <i>smother</i> , <i>punch</i> , dll).
<i>Half End</i>	34	Peluit akhir babak pertandingan oleh wasit.
<i>Half Start</i>	18	Peluit awal babak pertandingan oleh wasit.
<i>Injury Stoppage</i>	40	Penghentian permainan karena cedera.
<i>Interception</i>	10	Pemain memotong jalur operan lawan untuk mencegah bola sampai ke target.
<i>Miscontrol</i>	38	Kehilangan kontrol bola karena sentuhan yang buruk.
<i>Offside</i>	8	Pelanggaran posisi <i>offside</i> .

<i>Own Goal Against</i>	20	Gol bunuh diri oleh tim sendiri.
<i>Own Goal For</i>	25	Gol bunuh diri yang menguntungkan tim.
<i>Pass</i>	30	Umpan dari satu pemain ke pemain lain.
<i>Player Off</i>	27	Pemain keluar lapangan tanpa pergantian (misalnya karena cedera).
<i>Player On</i>	26	Pemain kembali masuk ke lapangan setelah <i>Player Off</i> .
<i>Pressure</i>	17	Aksi menekan pemain lawan di area tertentu, direkam bersama durasi tekanan.
<i>Referee Ball-Drop</i>	41	Wasit menjatuhkan bola untuk melanjutkan pertandingan setelah jeda (misalnya cedera).
<i>Shield</i>	28	Pemain melindungi bola agar keluar lapangan tanpa dikejar lawan.
<i>Shot</i>	16	Upaya mencetak gol dengan bagian tubuh legal.
<i>Starting XI</i>	35	Informasi awal pemain yang bermain dan formasi tim.
<i>Substitution</i>	19	Pergantian pemain saat pertandingan berlangsung.
<i>Tactical Shift</i>	36	Perubahan posisi pemain atau formasi taktik dalam pertandingan.

Dengan menambahkan konteks ini, model dapat memahami alur permainan yang berujung pada tembakan dan mengenali pola peristiwa yang secara statistik lebih mungkin menghasilkan gol. Fitur *type_before* diisi hanya jika terdapat *event* sebelumnya, jika tembakan merupakan *event* pertama dalam urutan, maka fitur ini dikosongkan. Gambar 4.7 menunjukkan hasil dari *feature engineering* tersebut.

type_before
2
43
42
4
4

Gambar 4.7 Type Before

4.3.2 Seleksi Fitur

Tahap ini bertujuan untuk memilih fitur-fitur yang paling relevan dan berpengaruh terhadap prediksi model, sehingga dapat meningkatkan efisiensi dan akurasi pemodelan. Seleksi fitur dilakukan setelah proses *feature engineering* selesai, dengan mempertimbangkan konteks domain serta performa masing-masing fitur dalam mendukung prediksi *shot_outcome*. Fitur yang memiliki kontribusi kecil atau *redundan* dapat dihilangkan untuk menghindari kompleksitas berlebih dan mengurangi risiko *overfitting*. Proses ini membantu model fokus pada informasi yang benar-benar penting. Tabel 4.1 menunjukkan fitur-fitur yang dipertahankan setelah melalui tahap seleksi.

Tabel 4.4 Fitur-Fitur Pada Tahap Seleksi

No.	Fitur
1	<i>minute</i>
2	<i>second</i>
3	<i>play_pattern</i>
4	<i>position</i>
5	<i>shot_technique</i>

6	<i>shot_body_part</i>
7	<i>shot_type</i>
8	<i>shot_first_time</i>
9	<i>shot_open_goal</i>
10	<i>shot_one_on_one</i>
11	<i>shot_aerial_won</i>
12	<i>under_pressure</i>
13	<i>distance_to_goal</i>
14	<i>angle_to_goal</i>
15	<i>shot_key_pass</i>
16	<i>start_x</i>
17	<i>start_y</i>
18	<i>possession</i>

4.3.3 Pemisahan Data Uji dan Data Latih

Salah satu tahapan penting dalam proses *transformation* adalah pemisahan data menjadi data latih dan data uji. Tujuan dari proses ini adalah untuk mengevaluasi kinerja model secara objektif terhadap data yang belum pernah digunakan dalam proses pelatihan. Pemisahan data dilakukan menggunakan fungsi *train_test_split* dari *library scikit-learn*, dengan proporsi 90% data sebagai data latih dan 10% sebagai data uji. Parameter *random_state* disetel ke angka 42 untuk menjamin konsistensi hasil pemisahan saat kode dijalankan ulang. Setelah proses ini dilakukan, diperoleh 61.981 baris data untuk pelatihan dan 6.887 baris data untuk pengujian. Gambar 4.6 menunjukkan jumlah baris dan kolom data latih dan uji.

	Train	Test
Rows	61981	6887
Columns	17	17

Gambar 4.8 Jumlah Baris dan Kolom Data Latih dan Uji.

4.4 Data Mining

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

4.4.1 Pembangunan dan Optimasi Model

Proses ini merupakan tahap inti untuk menemukan arsitektur model dengan performa terbaik melalui serangkaian eksperimen yang sistematis untuk *hyperparameter tuning*.

a. Inisialisasi dan Ruang Pencarian *Hyperparameter*

Proses pemodelan diawali dengan inisialisasi LGBMClassifier. Untuk mendapatkan performa yang optimal, dilakukan proses *tuning* terhadap sejumlah *hyperparameter*. Ruang pencarian *hyperparameter* yang digunakan dalam penelitian ini, yang didefinisikan dalam metode *RandomizedSearchCV*, disajikan pada Tabel 4.5.

Tabel 4.5 Ruang Pencarian *Hyperparameter*

Nama <i>Hyperparameter</i>	Nilai yang Diuji
<i>min_child_samples</i>	Distribusi integer acak dari 0-200