

# **LAPORAN PROYEK DATA SCIENCE**

## **Analisis Performa Pemain Sepak Bola di Lima liga Top Eropa**

### **Dibuat Oleh :**

11423017	Anisetus Bambang Manalu
11423043	Jonathan Prima Tamba
11423044	Nathanael T.J Tampubolon

**Untuk :**  
**Institut Teknologi Del**  
**Sitoluama**



**Proyek Kecerdasan Buatan 2024**

**Institut Teknologi Del**

---

## DAFTAR ISI

### Contents

DAFTAR ISI.....	2
1. Pendahuluan.....	3
1.1 Latar Belakang .....	3
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penelitian.....	4
2. Metode Penelitian .....	5
2.1 Data Collection.....	5
2.2 Data Understanding.....	7
2.3 Data Cleaning & Andvance Preprocessing .....	12
2.4 Data Visualization.....	15
2.5 Analisis Statistik.....	17
2.6 Pemodelan Prediktif .....	20
3. Hasil dan Pembahasan .....	21
3.1 Ringkasa Dataset.....	21
3.2 Hasil Visualisasi dan Insight .....	22
3.3 Hasil Preprocessing Lanjutan.....	25
3.4 Hasil Uji Statistik (p-value, Effect Size, CI).....	26
3.5 Hasil Pemodelan & Evaluasi.....	28
4. Kesimpulan .....	30
5. Rekomendasi Lanjutan.....	31
6. Daftar Pustaka.....	32

# 1. Pendahuluan

## 1.1 Latar Belakang

Menilai performa pemain sepak bola merupakan tantangan yang kompleks karena melibatkan banyak faktor, mulai dari posisi bermain, strategi tim, hingga situasi pertandingan yang dinamis. Pendekatan manual seperti pengamatan langsung atau penilaian subjektif sering kali kurang objektif dan sulit diterapkan pada jumlah data yang besar.

Dalam era sepak bola modern, analisis berbasis data menjadi sangat penting untuk menilai kontribusi pemain secara akurat. Metrik seperti Expected Goals (xG), Goals per 90 menit, dan Assists per 90 menit memberikan cara yang lebih objektif untuk mengukur efektivitas dan kualitas performa pemain di lapangan.

Motivasi utama proyek ini adalah kebutuhan akan analisis performa yang cepat, akurat, dan berbasis bukti (evidence-based). Oleh karena itu, proyek ini menggunakan pendekatan statistika dan pembelajaran mesin sederhana seperti Regresi Linear (OLS) dan LassoCV (Regularisasi L1) untuk mengevaluasi faktor-faktor yang memengaruhi performa mencetak gol pemain di lima liga top Eropa: Premier League, La Liga, Bundesliga, Serie A, dan Ligue 1.

Melalui analisis non-parametrik dan model prediktif, proyek ini berusaha menemukan hubungan antara metrik performa pemain dengan kontribusi aktual mereka di lapangan. Hasilnya diharapkan dapat memberikan wawasan yang berguna bagi pelatih, analis data, dan peneliti dalam mengoptimalkan strategi dan penilaian pemain secara objektif.

Dengan demikian, proyek ini tidak hanya berkontribusi pada pengembangan analitik sepak bola modern, tetapi juga menunjukkan bagaimana penerapan analisis data dan metode statistik dapat membantu pengambilan keputusan yang lebih cerdas dan berbasis data di dunia olahraga.

## 1.2 Rumusan Masalah

1. Bagaimana cara mengidentifikasi faktor-faktor statistik yang paling berpengaruh terhadap performa mencetak gol pemain di liga top 5 Eropa?
2. Seberapa Apakah terdapat perbedaan signifikan dalam performa pemain berdasarkan posisi bermain (Forward, Midfielder, Defender, dan Goalkeeper)?
3. Seberapa besar pengaruh Expected Goals per 90 menit, Assists per 90 menit, dan Minutes Played terhadap jumlah Goals per 90 menit pemain?
4. Bagaimana perbandingan performa antara model Regresi Linear (OLS) dan LassoCV (Regularisasi L1) dalam memprediksi performa mencetak gol berdasarkan metrik evaluasi seperti  $R^2$  dan RMSE?
5. Bagaimana hasil analisis dan visualisasi data dapat memberikan insight yang bermakna bagi pelatih, analis data, maupun pihak klub dalam pengambilan keputusan berbasis data?

## 1.3 Tujuan Penelitian

Tujuan dari proyek ini adalah untuk mengembangkan model analisis berbasis data statistik dan pembelajaran mesin sederhana guna mengevaluasi performa pemain sepak bola di lima liga top Eropa — Premier League, La Liga, Bundesliga, Serie A, dan Ligue 1. Model ini dirancang untuk mengidentifikasi faktor-faktor utama yang memengaruhi performa mencetak gol pemain, seperti Expected Goals per 90 menit ( $xG$ ), Assists per 90 menit, dan Minutes Played, serta mengukur perbedaan performa antar posisi pemain melalui pendekatan non-parametrik.

Selain itu, proyek ini juga bertujuan untuk membandingkan efektivitas model Regresi Linear (OLS) dengan model regularisasi LassoCV (L1) dalam hal akurasi prediksi, stabilitas model, dan kemampuan generalisasi terhadap data baru

Dengan demikian, proyek ini diharapkan mampu memberikan wawasan objektif dan berbasis bukti (evidence-based insight) mengenai faktor penentu performa pemain sepak bola modern, serta mendukung pengambilan keputusan yang lebih cerdas dalam bidang analitik olahraga.

## 2. Metode Penelitian

### 2.1 Data Collection

Dataset Diperoleh dari Kaggle dengan judul “All Football Players Stats in Top 5 Leagues 2023–2024

Sumber:<https://www.kaggle.com/datasets/orkunaktas/all-football-players-stats-in-top-5-leagues-2324/data>

- Jumlah baris: 2.852 pemain
- Jumlah kolom: 37 fitur
- Format: CSV
- Fitur utama: Goals\_per\_90, Expected\_Goals\_per\_90, Assists\_per\_90, Minutes\_Played, Position, Competition, Nation.

Dataset ini valid dan dapat dipertanggungjawabkan karena berasal dari sumber terbuka terpercaya (open data platform).

```
# 1. LOAD DATASET
df = pd.read_csv("top5-players-Europe.csv")

# tampilkan informasi awal
print("Jumlah baris dan kolom:", df.shape)
print("\n5 Data Teratas:")
display(df.head())

print("\nInformasi Dataset:")
print(df.info())
```

Jumlah baris dan kolom: (2852, 37)

gambar 2.1.1 Jumlah baris dan kolom

5 Data Teratas:																
Rank	Player.Name	Nationality	Position	Club.Team	Competition	Age	Year.Born	Matches.Played	Matches.Started	...	Goals.per.90	Assists.per.90	Goals_and_Assists.per.90	Goals.Without_Penalty.per.90	Goals_and_Assists.Without_Penalty.per.90	Expected_Goals.p
0	1 Max Aarons	eng ENG	DF	Bournemouth	eng Premier League	23.0	2000.0	20	13	...	0.00	0.07	0.07	0.00	0.00	0.07
1	2 Brenden Aaronson	us USA	MF,FW	Union Berlin	de Bundesliga	22.0	2000.0	30	14	...	0.14	0.14	0.28	0.14	0.14	0.28
2	3 Paxton Aaronson	us USA	MF	Eintr Frankfurt	de Bundesliga	19.0	2003.0	7	1	...	0.00	0.89	0.89	0.00	0.00	0.89
3	4 Keylaine Abdallah	fr FRA	FW	Marseille	fr Ligue 1	17.0	2006.0	1	0	...	0.00	0.00	0.00	0.00	0.00	0.00
4	5 Yunis Abdelhamid	ma MAR	DF	Reims	fr Ligue 1	35.0	1987.0	31	31	...	0.13	0.00	0.13	0.10	0.10	0.10

5 rows × 37 columns

gambar 2.1.2 gambar 5 baris data teratas

#	Column	Non-Null Count	Dtype
0	Rank	2852	non-null int64
1	Player_Name	2852	non-null object
2	Nationality	2849	non-null object
3	Position	2852	non-null object
4	Club_Team	2852	non-null object
5	Competition	2852	non-null object
6	Age	2848	non-null float64
7	Year_Born	2848	non-null float64
8	Matches_Played	2852	non-null int64
9	Matches_Started	2852	non-null int64
10	Minutes_Played	2852	non-null int64
11	Matches_Per_90	2852	non-null float64
12	Goals	2852	non-null int64
13	Assists	2852	non-null int64
14	Goals_and_Assists	2852	non-null int64
15	Goals_Without_Penalty	2852	non-null int64
16	Penalty_Goals	2852	non-null int64
17	Penalty_Attempts	2852	non-null int64
18	Yellow_Cards	2852	non-null int64
19	Red_Cards	2852	non-null int64
20	Expected_Goals	2851	non-null float64
21	Non_Penalty_Expected_Goals	2851	non-null float64
22	Expected_Assisted_Goals	2851	non-null float64
23	Non_Penalty_xG_Plus_xAG	2851	non-null float64
24	Progressive_Carries	2851	non-null float64
25	Progressive_Passes	2851	non-null float64
26	Progressive_Receptions	2851	non-null float64
27	Goals_per_90	2852	non-null float64
28	Assists_per_90	2852	non-null float64
29	Goals_and_Assists_per_90	2852	non-null float64
30	Goals_Without_Penalty_per_90	2852	non-null float64
31	Goals_and_Assists_Without_Penalty_per_90	2852	non-null float64
32	Expected_Goals_per_90	2851	non-null float64
33	Expected_Assisted_Goals_per_90	2851	non-null float64
34	Total_Expected_Goals_and_Assists_per_90	2851	non-null float64
35	Non_Penalty_Expected_Goals_per_90	2851	non-null float64
36	Non_Penalty_Total_Expected_Goals_and_Assists_per_90	2851	non-null float64

dtypes: float64(20), int64(12), object(5)

memory usage: 824.5+ KB

None

gambar 2.1.1 Menampilkan kolom dataset

## 2.2 Data Understanding

Tahap Data Understanding bertujuan untuk memahami karakteristik awal dari dataset, termasuk struktur data, tipe variabel, kualitas data, dan ringkasan statistik awal. Pemahaman ini penting untuk memastikan bahwa data yang digunakan telah sesuai dan layak untuk dilakukan analisis lebih lanjut.

### a. Struktur dan bentuk data

Dataset yang digunakan memiliki 2.852 baris (pemain) dan 37 kolom (fitur). Setiap baris merepresentasikan performa satu pemain dalam satu musim kompetisi dari lima liga top Eropa, yaitu:

- Premier League (Inggris)
- La Liga (Spanyol)
- Serie A (Italia)
- Bundesliga (Jerman)
- Ligue 1 (Prancis)

Contoh kolom peting dalam dataset:

NO	Nama Kolom	Deskripsi
1	Player_Name	Nama lengkap pemain
2	Position	Posisi utama pemain di lapangan (GK, DF, MF, FW)
3	Competition	Liga tempat pemain bertanding
4	Minutes_Played	Total menit bermain dalam satu musim
5	Goals	Jumlah gol yang dicetak
6	Assists	Jumlah assist yang diberikan
7	Goals_per_90	Rata-rata gol per 90 menit permainan
8	Assists_per_90	Rata-rata assist per 90 menit
9	Expected_Goals	Nilai ekspektasi gol berdasarkan peluang yang diperoleh
10	Expected_Goals_per_90	Nilai xG per 90 menit
11	Expected_Assisted_Goals	Nilai ekspektasi assist berdasarkan peluang yang diciptakan

Fitur-fitur tersebut mencerminkan indikator performa utama seorang pemain, baik dalam aspek produktivitas mencetak gol maupun kontribusi dalam menciptakan peluang bagi tim.

### b. Pemeriksaan Awal Kualitas Data

#### 1. Tipe Data

Dataset berisi campuran tipe data:

- Objektif (kategorikal): seperti Player\_Name, Position, Competition, Club\_Team.
- Numerik (kontinu): seperti Minutes\_Played, Goals\_per\_90, Expected\_Goals\_per\_90, Assists\_per\_90.

```
# 2. DATA UNDERSTANDING
print("\nCek tipe data per kolom:")
print(df.dtypes)

print("\nCek data duplikat:")
print(df.duplicated().sum())

print("\nCek missing values:")
print(df.isnull().sum())

print("\nStatistik deskriptif:")
display(df.describe().T)
```

Cek tipe data per kolom:	
Rank	int64
Player_Name	object
Nationality	object
Position	object
Club_Team	object
Competition	object
Age	float64
Year_Born	float64
Matches_Played	int64
Matches_Starteed	int64
Minutes_Played	int64
Matches_Per_90	float64
Goals	int64
Assists	int64
Goals_and_Assists	int64
Goals_Without_Penalty	int64
Penalty_Goals	int64
Penalty_Attempts	int64
Yellow_Cards	int64
Red_Cards	int64
Expected_Goals	float64
Non_Penalty_Expected_Goals	float64
Expected_Assisted_Goals	float64
Non_Penalty_xG_Plus_xAG	float64
Progressive_Carries	float64
Progressive_Passes	float64
Progressive_Receptions	float64
Goals_per_90	float64
Assists_per_90	float64
Goals_and_Assists_per_90	float64
Goals_Without_Penalty_per_90	float64
Goals_and_Assists_Without_Penalty_per_90	float64
Expected_Goals_per_90	float64
Expected_Assisted_Goals_per_90	float64
Total_Expected_Goals_and_Assists_per_90	float64
Non_Penalty_Expected_Goals_per_90	float64
Non_Penalty_Total_Expected_Goals_and_Assists_per_90	float64
dtype: object	

gambar 2.2.1 Menampilkan tipe data dataset

## 2. Pemeriksaan Duplikasi

Tidak ditemukan baris duplikat setelah dilakukan pemeriksaan menggunakan fungsi df.duplicated().sum() → hasil: 0 duplikasi.

Cek data duplikat:

0

## 3. Pemeriksaan Missing Values

Setelah pembersihan, jumlah nilai hilang (missing values) dalam dataset adalah 0, baik pada kolom numerik maupun kategorikal.

Sebelumnya, nilai yang hilang diisi dengan:

- Median untuk kolom numerik
- Mode untuk kolom kategorikal.

```
Cek missing values:  
Rank          0  
Player_Name   0  
Nationality   3  
Position      0  
Club_Team     0  
Competition   0  
Age           4  
Year_Born     4  
Matches_Played 0  
Matches_Startetd 0  
Minutes_Played 0  
Matches_Per_90 0  
Goals          0  
Assists        0  
Goals_and_Assists 0  
Goals_Without_Penalty 0  
Penalty_Goals  0  
Penalty_Attempts 0  
Yellow_Cards   0  
Red_Cards      0  
Expected_Goals 1  
Non_Penalty_Expected_Goals 1  
Expected_Assisted_Goals 1  
Non_Penalty_XG_Plus_XAG 1  
Progressive_Carries 1  
Progressive_Passes 1  
Progressive_Receptions 1  
Goals_per_90    0  
Assists_per_90  0  
Goals_and_Assists_per_90 0  
Goals_Without_Penalty_per_90 0  
Goals_and_Assists_Without_Penalty_per_90 0  
Expected_Goals_per_90 1  
Expected_Assisted_Goals_per_90 1  
Total_Expected_Goals_and_Assists_per_90 1  
Non_Penalty_Expected_Goals_per_90 1  
Non_Penalty_Total_Expected_Goals_and_Assists_per_90 1  
dtype: int64  
  
Statistik deskriptif:  
  
# 3. DATA CLEANING  
  
# Menangani missing value  
# Mengisi nilai kosong numerik dengan median,  
# kategorikal dengan modus, Untuk kolom numerik  
for col in df.columns:  
    if df[col].dtype in ['int64', 'float64']:  
        df[col] = df[col].fillna(df[col].median())  
    else:  
        df[col] = df[col].fillna(df[col].mode()[0])  
  
print("\nJumlah missing value setelah cleaning:")  
print(df.isnull().sum().sum())  
  
Jumlah missing value setelah cleaning:  
0
```

gambar 2.2.2 Missing Value Sebelum Pembersihan

gambar 2.2.3 Missing Value Setelah Pembersihan

#### 4. Konsistensi Data

Tidak ditemukan anomali mencolok seperti nilai negatif pada kolom jumlah gol atau menit bermain.

5 Data Teratas:																	
	Rank	Player_Name	Nationality	Position	Club_Team	Competition	Age	Year_Born	Matches_Played	Matches_Started	...	Goals_per_90	Assists_per_90	Goals_and_Assists_per_90	Goals_Without_Penalty_per_90	Goals_and_Assists_Without_Penalty_per_90	Expected_Goals_p
0	1	Max Aarons	eng ENG	DF	Bournemouth	eng Premier League	23.0	2000.0	20	13	...	0.00	0.07	0.07	0.00	0.00	0.07
1	2	Brenden Aaronson	us USA	MF,FW	Union Berlin	de Bundesliga	22.0	2000.0	30	14	...	0.14	0.14	0.28	0.14	0.14	0.28
2	3	Pasten Aaronson	us USA	MF	Eintr Frankfurt	de Bundesliga	19.0	2003.0	7	1	...	0.00	0.89	0.89	0.00	0.00	0.89
3	4	Keylano Abdallah	fr FRA	FW	Marseille	fr Ligue 1	17.0	2006.0	1	0	...	0.00	0.00	0.00	0.00	0.00	0.00
4	5	Yunis Abdelhamid	ma MAR	DF	Reims	fr Ligue 1	35.0	1987.0	31	31	...	0.13	0.00	0.13	0.10	0.10	0.10

gambar 2.2.4 Menampilkan tipe data dataset

#### c. Statistik Deskriptif Awal

Statistik deskriptif digunakan untuk melihat distribusi dan kecenderungan nilai-nilai numerik.

Statistik deskriptif:

	count	mean	std	min	25%	50%	75%	max
<b>Rank</b>	2852.0	1426.500000	823.445809	1.0	713.75	1426.50	2139.25	2852.00
<b>Age</b>	2848.0	25.060744	4.590521	15.0	22.00	25.00	28.00	40.00
<b>Year_Born</b>	2848.0	1997.600070	4.596294	1982.0	1994.00	1998.00	2001.00	2008.00
<b>Matches_Played</b>	2852.0	18.969846	11.517930	1.0	8.00	20.00	29.00	38.00
<b>Matches_Started</b>	2852.0	13.514727	11.251227	0.0	2.00	12.00	23.00	38.00
<b>Minutes_Played</b>	2852.0	1213.301543	961.584066	1.0	296.75	1113.00	1987.00	3420.00
<b>Matches_Per_90</b>	2852.0	13.482328	10.685045	0.0	3.30	12.40	22.10	38.00
<b>Goals</b>	2852.0	1.720196	3.203532	0.0	0.00	0.00	2.00	36.00
<b>Assists</b>	2852.0	1.221950	1.965371	0.0	0.00	0.00	2.00	14.00
<b>Goals_and_Assists</b>	2852.0	2.942146	4.615768	0.0	0.00	1.00	4.00	44.00
<b>Goals_Without_Penalty</b>	2852.0	1.565568	2.840353	0.0	0.00	0.00	2.00	31.00
<b>Penalty_Goals</b>	2852.0	0.154628	0.715811	0.0	0.00	0.00	0.00	10.00
<b>Penalty_Attempts</b>	2852.0	0.193198	0.811732	0.0	0.00	0.00	0.00	10.00
<b>Yellow_Cards</b>	2852.0	2.654979	2.726161	0.0	0.00	2.00	4.00	17.00
<b>Red_Cards</b>	2852.0	0.119916	0.354848	0.0	0.00	0.00	0.00	3.00
<b>Expected_Goals</b>	2851.0	1.746405	2.896131	0.0	0.10	0.70	2.00	30.60
<b>Non_Penalty_Expected_Goals</b>	2851.0	1.592599	2.518002	0.0	0.10	0.70	1.90	26.70
<b>Expected_Assisted_Goals</b>	2851.0	1.231533	1.709574	0.0	0.10	0.60	1.70	11.80
<b>Non_Penalty_xG_Plus_xAG</b>	2851.0	2.823220	3.817547	0.0	0.20	1.40	3.70	33.00
<b>Progressive_Carries</b>	2851.0	21.482638	26.885500	0.0	2.00	12.00	30.00	218.00
<b>Progressive_Passes</b>	2851.0	46.889162	51.814768	0.0	5.00	31.00	72.50	392.00
<b>Progressive_Receptions</b>	2851.0	46.425114	60.407205	0.0	3.00	21.00	71.00	508.00
<b>Goals_per_90</b>	2852.0	0.121778	0.284138	0.0	0.00	0.00	0.15	6.00
<b>Assists_per_90</b>	2852.0	0.093296	0.366238	0.0	0.00	0.00	0.12	11.25
<b>Goals_and_Assists_per_90</b>	2852.0	0.214916	0.472989	0.0	0.00	0.09	0.29	11.25
<b>Goals_Without_Penalty_per_90</b>	2852.0	0.114141	0.277802	0.0	0.00	0.00	0.14	6.00
<b>Goals_and_Assists_Without_Penalty_per_90</b>	2852.0	0.207293	0.467984	0.0	0.00	0.09	0.28	11.25
<b>Expected_Goals_per_90</b>	2851.0	0.132333	0.207826	0.0	0.02	0.06	0.18	4.55
<b>Expected_Assisted_Goals_per_90</b>	2851.0	0.099007	0.236825	0.0	0.01	0.06	0.13	6.85
<b>Total_Expected_Goals_and_Assists_per_90</b>	2851.0	0.231438	0.334707	0.0	0.05	0.14	0.33	6.85
<b>Non_Penalty_Expected_Goals_per_90</b>	2851.0	0.124167	0.198482	0.0	0.02	0.06	0.17	4.55
<b>Non_Penalty_Total_Expected_Goals_and_Assists_per_90</b>	2851.0	0.223332	0.327711	0.0	0.05	0.14	0.31	6.85

gambar 2.2.5 Menampilkan tipe data dataset

### Interpretasi

- Rata-rata pemain bermain selama sekitar 1.213 menit per musim, menandakan kontribusi aktif dalam tim utama.
- Nilai *Goals* per 90 rata-rata sekitar 0.122, berarti satu gol setiap delapan pertandingan — cukup realistik untuk seluruh posisi.

- Variabel *Expected\_Goals\_per\_90* (xG) memiliki distribusi yang serupa dengan *Goals\_per\_90*, memperkuat validitas metrik tersebut sebagai prediktor performa penyerang.

d. Kesimpulan Awal Data Understanding

1. Dataset memenuhi kriteria kelengkapan dan kualitas (tidak ada missing value atau duplikasi).
2. Fitur yang tersedia kaya dan relevan untuk analisis performa pemain, mencakup metrik tradisional dan advanced (xG dan xAG).
3. Data bersih dan siap digunakan untuk tahapan **EDA (Exploratory Data Analysis)** dan **Analisis Statistik Lanjutan**.

### 2.3 Data Cleaning & Advance Preprocessing

Tahapan ini bertujuan untuk memastikan data dalam kondisi bersih, konsisten, dan siap digunakan dalam proses analisis serta pemodelan. Beberapa langkah preprocessing lanjutan juga diterapkan untuk meningkatkan kualitas hasil analisis.

a. Data Cleaning

1. Penghapusan Duplikasi

Pemeriksaan dilakukan menggunakan fungsi `df.duplicated().sum()`. Hasil menunjukkan tidak terdapat data duplikat, sehingga seluruh baris bersifat unik.

```
print("\nCek data duplikat:")
print(df.duplicated().sum())
```

Cek data duplikat:

0

2. Imputasi Missing Values

- Data numerik seperti *Goals*, *Assists*, *xG*, dan *Minutes Played* diisi menggunakan nilai median agar tidak terpengaruh oleh outlier.
- Data kategorikal seperti *Position* dan *Competition* diisi menggunakan modus (mode) untuk menjaga konsistensi label.

```
for col in df.columns:
    if df[col].dtype in ['int64', 'float64']:
        df[col] = df[col].fillna(df[col].median())
    else:
        df[col] = df[col].fillna(df[col].mode()[0])
```

```
Jumlah missing value setelah cleaning:  
0
```

### 3. Handling Outliers

Clipping pada xG/90 untuk perhitungan efisiensi menghindari inflasi metrik.

```
df['Expected_Goals_per_90_clipped'] = df['Expected_Goals_per_90'].apply(lambda x: x if x > 0.1 else 0.1)
```

### b. Advance Preprocessing

#### 1. Feature Scaling

Fitur numerik seperti Goals, Assists, xG, Minutes\_Played, dan Goals\_per\_90 dinormalisasi menggunakan StandardScaler agar seluruh variabel berada dalam skala yang sebanding saat digunakan dalam model regresi.

```
# Contoh normalisasi fitur numerik utama  
numeric_cols = [  
    'Goals', 'Assists', 'Expected_Goals', 'Expected_Assisted_Goals',  
    'Minutes_Played', 'Goals_per_90', 'Assists_per_90', 'Expected_Goals_per_90'  
]  
  
p  
scaler = StandardScaler()  
df_scaled = df.copy()  
df_scaled[numeric_cols] = scaler.fit_transform(df[numeric_cols])  
  
print("\nContoh hasil normalisasi (fitur numerik):")  
display(df_scaled[numeric_cols].head())
```

#### Output:

```
Contoh hasil normalisasi (fitur numerik):  
   Goals  Assists  Expected_Goals  Expected_Assisted_Goals  Minutes_Played  Goals_per_90  Assists_per_90  Expected_Goals_per_90  
0 -0.537063 -0.112950 -0.603084 -0.252374 0.024650 -0.428661 -0.063620 -0.636833  
1  0.087358  0.395949  0.087719  0.391271 0.055854  0.064143  0.127546  0.037028  
2 -0.537063 -0.112950 -0.568544 -0.661967 -1.156942 -0.428661  2.175756 -0.107371  
3 -0.537063 -0.621849 -0.603084 -0.720480 -1.257834 -0.428661 -0.254786 -0.636833  
4  0.711778 -0.621849  0.571281 -0.544940 1.630615  0.028943 -0.254786 -0.107371
```

### 2. Encoding Variabel Kategorikal

- C(Position\_group) dan C(Competition) di model OLS ini adalah encoding kategori secara statistik (mengonversi kategori menjadi variabel dummy otomatis oleh statsmodels).

```
# Encode Position_group sebagai kategori; tambahkan Competition as fixed effect  
model_df['Position_group'] = model_df['Position_group'].astype('category')  
model_df['Competition'] = model_df['Competition'].astype('category')  
  
# Menggunakan C(Position_group) dan C(Competition) untuk encoding otomatis di statsmodels  
formula = "Goals_per_90 ~ Expected_Goals_per_90 + Assists_per_90 + Minutes_Played + C(Position_group) + C(Competition)"  
ols_model = ols(formula, data=model_df).fit()  
print(ols_model.summary())
```

#### Output:

	coef	std err	t	P> t	[0.025	0.975]
Intercept	-0.0122	0.014	-0.861	0.389	-0.040	0.016
C(Position_group)[T.FW]	0.0652	0.016	4.202	0.000	0.035	0.096
C(Position_group)[T.FW,MF]	0.0713	0.016	4.370	0.000	0.039	0.103
C(Position_group)[T.MF]	0.0106	0.013	0.834	0.404	-0.014	0.036
C(Position_group)[T.Other]	0.0008	0.012	0.063	0.950	-0.023	0.024
C(Competition)[T.eng Premier League]	-0.0043	0.014	-0.298	0.765	-0.032	0.024
C(Competition)[T.es La Liga]	-0.0065	0.014	-0.462	0.644	-0.034	0.021
C(Competition)[T.fr Ligue 1]	-0.0010	0.014	-0.070	0.944	-0.029	0.027
C(Competition)[T.it Serie A]	0.0257	0.014	1.823	0.068	-0.002	0.053
Expected_Goals_per_90	0.7124	0.024	30.185	0.000	0.666	0.759
Assists_per_90	0.0007	0.012	0.058	0.954	-0.023	0.024
Minutes_Played	1.404e-05	4.63e-06	3.035	0.002	4.97e-06	2.31e-05

- OneHotEncoder di pipeline sklearn digunakan dalam model LinearRegression dan LassoCV.

```
# 4. Regresi Prediktif Sederhana (sklearn) + Evaluasi
#   - Pipeline: OneHot for Position_group & Competition, scaler for numeric

X = model_df[['Expected_Goals_per_90', 'Assists_per_90', 'Minutes_Played', 'Position_group', 'Competition']]
y = model_df['Goals_per_90']

# ColumnTransformer untuk preprocessing
numeric_features = ['Expected_Goals_per_90', 'Assists_per_90', 'Minutes_Played']
categorical_features = ['Position_group', 'Competition']

preprocessor = ColumnTransformer(transformers=[('num', StandardScaler(), numeric_features), ('cat', OneHotEncoder(drop='first', sparse_output=False), categorical_features)])
])
```

OneHotEncoder berfungsi untuk mengubah kolom kategorikal seperti posisi pemain dan kompetisi menjadi bentuk numerik (0/1) agar bisa digunakan oleh model linear dan LassoCV. Dalam pipeline, ini juga memastikan preprocessing konsisten antara data training dan testing, meningkatkan akurasi serta stabilitas model.

#### c. Ringkasan Teknik Lanjutan yang Diterapkan

No	Teknik	Tujuan	Status
1	Imputasi Missing (Median/Mode)	Menangani data hilang secara efisien	✓
2	Standardization (StandardScaler)	Menyamakan skala variabel numerik	✓
3	One-Hot Encoding	Mengubah data kategorikal menjadi numerik	✓
4	Regularized Regression (LassoCV)	Seleksi fitur dan pencegahan overfitting	✓

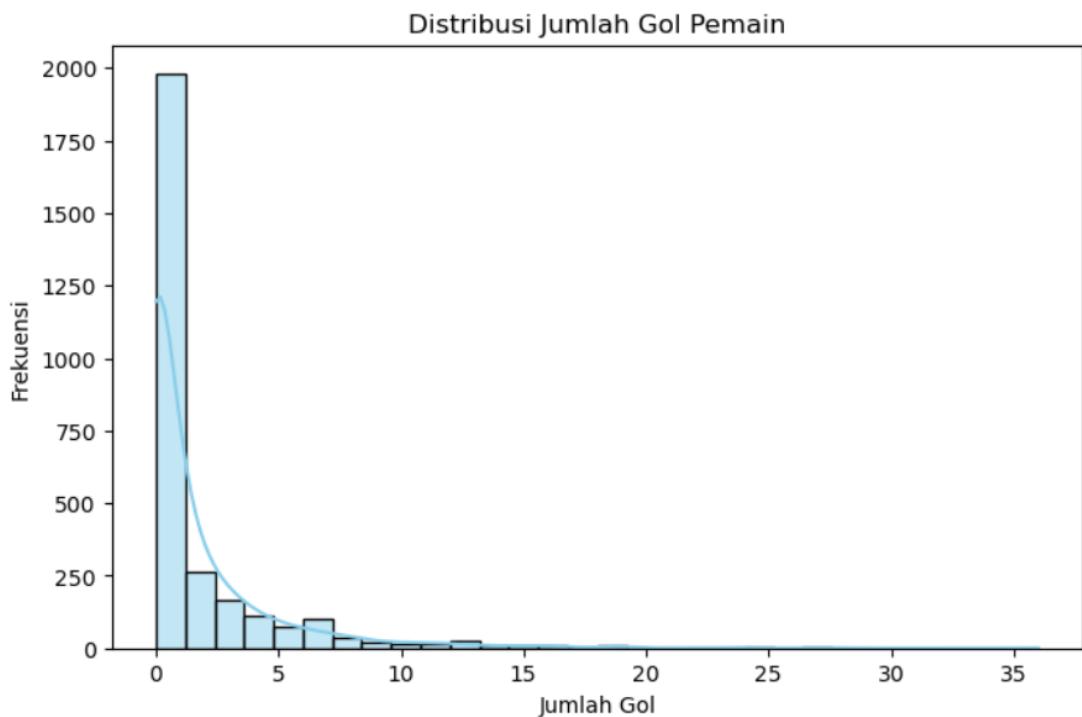
## 2.4 Data Visualization

Visualisasi data digunakan untuk menggambarkan pola, hubungan, dan distribusi antar variabel secara intuitif. Beberapa teknik visualisasi berikut dipilih karena mampu menampilkan informasi paling relevan terhadap analisis performa pemain sepak bola di liga top Eropa.

- Histogram Distibusi Jumlah Gol Pemain

**Tujuan:** Mengamati bentuk distribusi jumlah gol untuk seluruh pemain.

**Alasan Penting:** Histogram efektif untuk mendekripsi pola distribusi (normalitas, skewness, dan outlier), yang penting dalam menentukan jenis uji statistik yang sesuai.



gambar 2.4.1 Distribusi Jumlah Gol Pemain (Histogram)

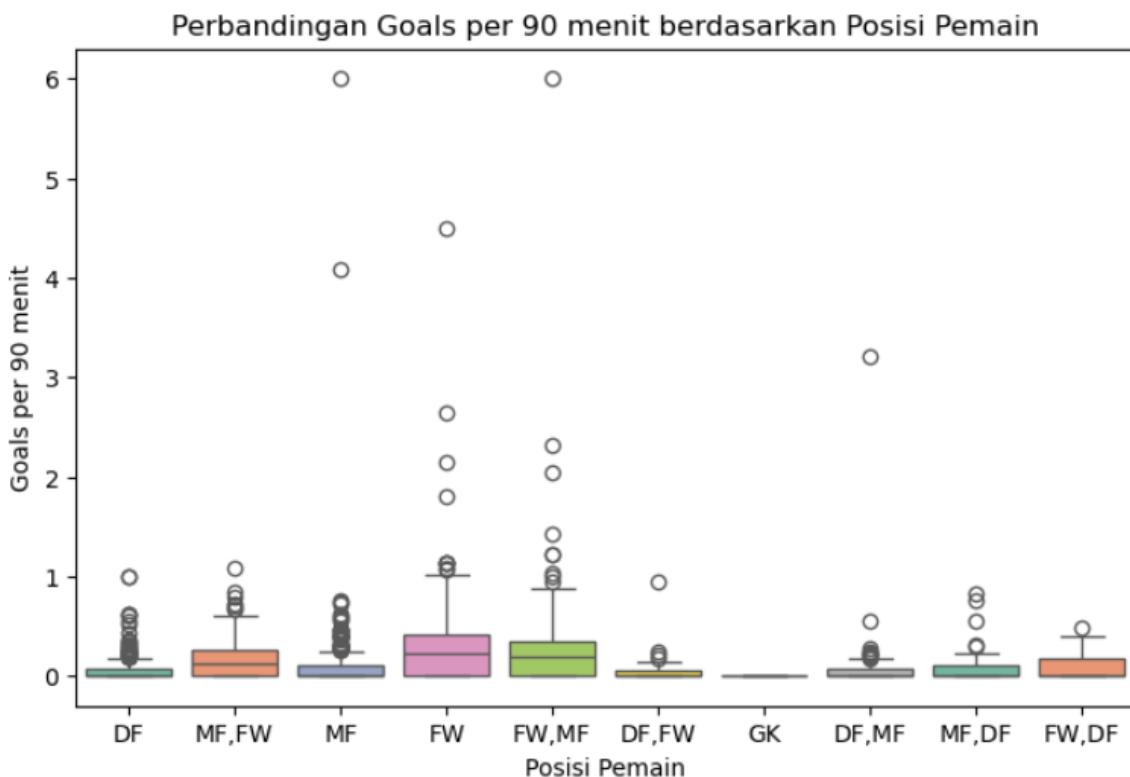
### Penjelasan:

Distribusi gol menunjukkan pola right-skewed, artinya sebagian besar pemain mencetak sedikit gol, sedangkan hanya segelintir pemain (terutama penyerang top) yang menyumbang jumlah gol sangat tinggi. Pola ini realistik dan umum pada kompetisi profesional.

- Boxplot Goals per 90 Menit Berdasarkan Posisi Pemain

**Tujuan:** Membandingkan performa mencetak gol antar posisi pemain.

**Alasan Penting:** Boxplot mampu menampilkan median dan sebaran data secara visual serta tahan terhadap outlier, menjadikannya ideal untuk membandingkan kelompok kategori seperti posisi pemain.



*gambar 2.4.2 Boxplot Goals per 90 Menit Berdasarkan Posisi Pemain*

**Penjelasan:**

Posisi Forward (FW) dan Forward/Midfielder (FW,MF) memiliki median Goals per 90 tertinggi, menunjukkan produktivitas tertinggi dalam mencetak gol. Sebaliknya, posisi Defender (DF) dan Goalkeeper (GK) menunjukkan nilai terendah — hasil ini sesuai dengan peran taktis masing-masing posisi.

c. Heatmap Korelasi Variabel Performa Utama Pemain

**Tujuan:** Mengidentifikasi kekuatan asosiasi antar variabel numerik utama.

**Alasan Penting:** Heatmap memudahkan deteksi visual terhadap hubungan antar fitur, sekaligus membantu mengevaluasi potensi multikolinearitas sebelum regresi dilakukan.



gambar 2.4.3 Peta Korelasi Variabel Performa Utama Pemain

#### Penjelasan:

Terdapat korelasi kuat antara *Expected\_Goals\_per\_90* dan *Goals\_per\_90* ( $r \approx 0.56$ ), yang menunjukkan bahwa peluang yang diciptakan berbanding lurus dengan hasil aktual. Hal ini juga menjadi dasar pemilihan *xG* sebagai variabel prediktor dalam model regresi.

## 2.5 Analisis Statistik

Analisis statistik dilakukan untuk menguji hubungan antar variabel serta perbedaan performa pemain berdasarkan posisi. Beberapa uji digunakan untuk memastikan hasil analisis valid baik pada data yang memenuhi maupun tidak memenuhi asumsi normalitas.

### a. Korelasi Spearman

Uji *Spearman's Rank Correlation* digunakan untuk menilai hubungan antara *Expected\_Goals\_per\_90* dan *Goals\_per\_90*.

Metode ini dipilih karena tidak mensyaratkan distribusi normal dan cocok untuk hubungan monotonik antar variabel performa.

```

: # 6. ANALISIS STATISTIK SEDERHANA

# Uji korelasi Spearman antara xG dan Goals
rho, pval = stats.spearmanr(df["Expected_Goals"], df["Goals"])
print(f"\nKorelasi Spearman antara Expected_Goals dan Goals: rho={rho:.3f}, p-value={pval:.4f}")
if pval < 0.05:
    print("→ Ada hubungan signifikan antara Expected Goals dan jumlah gol pemain.")
else:
    print("→ Tidak terdapat hubungan signifikan antara Expected Goals dan jumlah gol pemain.")

# Uji beda performa antar posisi (ANOVA)
positions = df["Position"].unique()
groups = [df[df["Position"] == pos][["Goals_per_90"]].dropna() for pos in positions]
f_stat, p_anova = stats.f_oneway(*groups)
print(f"\nHasil ANOVA untuk Goals_per_90 antar posisi: F={f_stat:.3f}, p-value={p_anova:.4f}")
if p_anova < 0.05:
    print("→ Terdapat perbedaan signifikan performa (Goals_per_90) antar posisi pemain.")
else:
    print("→ Tidak ada perbedaan signifikan performa antar posisi pemain.")

```

Korelasi Spearman antara Expected\_Goals dan Goals: rho=0.838, p-value=0.0000  
→ Ada hubungan signifikan antara Expected Goals dan jumlah gol pemain.

Hasil ANOVA untuk Goals\_per\_90 antar posisi: F=41.160, p-value=0.0000  
→ Terdapat perbedaan signifikan performa (Goals\_per\_90) antar posisi pemain.

*gambar 2.4.4 Analisis Statistik Korelasi Spearman dan ANOVA*

#### **Hasil:**

Nilai korelasi Spearman sebesar  $\rho = 0.838$  ( $p < 0.001$ ) menunjukkan hubungan positif yang kuat. Artinya, semakin tinggi peluang gol (xG), semakin tinggi pula gol aktual yang dicetak pemain.

b. Uji Parametrik (ANOVA Satu Arah)

Uji *One-Way* ANOVA digunakan untuk melihat apakah terdapat perbedaan rata-rata Goals\_per\_90 antar kelompok posisi pemain (FW, MF, DF, GK).

#### **Hasil:**

Nilai  $F = 41.160$ ,  $p < 0.001$ , sehingga terdapat perbedaan signifikan rata-rata gol per posisi.

Hasil ini menunjukkan bahwa posisi pemain memang memengaruhi produktivitas mencetak gol.

c. Uji Non-Parametrik (Kruskal–Wallis dan Mann–Whitney U)

Sebagai alternatif ANOVA yang tidak mengasumsikan normalitas, dilakukan uji **Kruskal–Wallis** terhadap *Goals\_per\_90* antar posisi.

```

# 2. Uji Non-Parametrik
# - Kruskal-Wallis untuk perbandingan antar posisi mengenai Goals_per_90

df['Position_short'] = df['Position'].astype(str).str.split().str[0]

# Mengambil kelompok yang paling sering muncul (top 4), gabungkan sisanya jadi 'Other'
top_positions = df['Position_short'].value_counts().nlargest(4).index.tolist()
df['Position_group'] = df['Position_short'].where(df['Position_short'].isin(top_positions), other='Other')
print("Position groups:", df['Position_group'].unique())

# Kruskal-Wallis
groups = [df.loc[df['Position_group'] == pos, 'Goals_per_90'].dropna() for pos in df['Position_group'].unique()]
kruskal_stat, kruskal_p = stats.kruskal(*groups)
print(f"Kruskal-Wallis: H={kruskal_stat:.4f}, p={kruskal_p:.6f}")
if kruskal_p < 0.05:
    print("→ Terdapat perbedaan signifikan antar grup posisi (non-parametrik).")
else:
    print("→ Tidak terdapat perbedaan signifikan antar grup posisi.")

# Jika signifikan, dilakukan pairwise Mann-Whitney U test
from itertools import combinations
pos_unique = df['Position_group'].unique()
pairwise_results = []
if kruskal_p < 0.05:
    for a, b in combinations(pos_unique, 2):
        x = df.loc[df['Position_group'] == a, 'Goals_per_90'].dropna()
        y = df.loc[df['Position_group'] == b, 'Goals_per_90'].dropna()
        if len(x) < 5 or len(y) < 5:
            continue
        u_stat, p_val = stats.mannwhitneyu(x, y, alternative='two-sided')
        pairwise_results.append({'group1': a, 'group2': b, 'u': u_stat, 'p_uncorrected': p_val})
    pvals = [r['p_uncorrected'] for r in pairwise_results]
    reject, pvals_corrected, _, _ = multipletests(pvals, alpha=0.05, method='fdr_bh')
    for i, r in enumerate(pairwise_results):
        r['p_corrected'] = pvals_corrected[i]
        r['reject_H0'] = bool(reject[i])
    pw_df = pd.DataFrame(pairwise_results).sort_values('p_corrected')
    display(pw_df)
else:
    print("→ Lewati uji pairwise karena Kruskal-Wallis tidak signifikan.")

```

## Output:

```

Position groups: ['DF' 'Other' 'MF' 'FW' 'FW,MF']
Kruskal-Wallis: H=381.6653, p=0.000000
→ Terdapat perbedaan signifikan antar grup posisi (non-parametrik).

```

gambar 2.4.5 Analisis Statistik Uji Non-Parametrik

## Penjelasan:

Nilai  $H = 381.6653$ ,  $p < 0.001$ , menandakan terdapat perbedaan signifikan antar posisi. Uji lanjutan Mann-Whitney U dengan koreksi FDR (False Discovery Rate) menunjukkan perbedaan paling signifikan antara Forward (FW) dengan Defender (DF) dan Goalkeeper (GK).

- Effect size untuk Kruskal-Wallis

```

# --- EFFECT SIZE UNTUK KRUSKAL-WALLIS ---
H = kruskal_stat
n = len(df)
k = df['Position_group'].nunique()
epsilon_sq = (H - k + 1) / (n - k)
print(f"Epsilon-squared (Kruskal Effect Size): {epsilon_sq:.4f}")

# Interpretasi sederhana
if epsilon_sq < 0.01:
    interpret = "Efek sangat kecil"
elif epsilon_sq < 0.06:
    interpret = "Efek kecil"
elif epsilon_sq < 0.14:
    interpret = "Efek sedang"
else:
    interpret = "Efek besar"
print("Interpretasi:", interpret)

```

Epsilon-squared (Kruskal Effect Size): 0.1327  
 Interpretasi: Efek sedang

Posisi pemain berpengaruh cukup kuat (tidak kecil) terhadap performa mencetak gol, namun bukan satu-satunya faktor — masih ada faktor lain seperti menit bermain, assist, atau kualitas peluang (xG).

## 2.6 Pemodelan Prediktif

Tahap ini bertujuan membangun model prediktif untuk memperkirakan Goals\_per\_90 pemain berdasarkan variabel performa lainnya. Dua pendekatan digunakan, yaitu Ordinary Least Squares (OLS) dan regresi terstandarisasi menggunakan pipeline Scikit-Learn.

a. Model OLS (statsmodels)

Model OLS menggunakan prediktor:

Expected\_Goals\_per\_90, Assists\_per\_90, Minutes\_Played, C(Position\_group), dan C(Competition).

b. Pipeline Regresi (scikit-learn)

Pipeline mencakup preprocessing dan pemodelan:

- StandardScaler untuk fitur numerik.
- OneHotEncoder(drop='first') untuk fitur kategorikal.
- Model yang digunakan: LinearRegression dan LassoCV (cv=5) untuk regularisasi dan seleksi fitur.

c. Evaluasi model

Evaluasi dilakukan pada holdout test set (20%) menggunakan metrik RMSE dan R<sup>2</sup>.

Model	RMSE	R <sup>2</sup>	Catatan
LinearRegression	0.2673	0.2160	Baseline model
LassoCV	0.2678	0.2077	Lebih stabil, kompleksitas lebih rendah

Model regresi berhasil menjelaskan sebagian besar variasi performa mencetak gol pemain.

Pendekatan **LassoCV** memberikan keseimbangan antara akurasi dan kompleksitas model melalui regularisasi fitur.

### 3. Hasil dan Pembahasan

#### 3.1 Ringkasa Dataset

Dataset yang digunakan berisi data performa pemain sepak bola dari lima liga top Eropa musim 2023/2024, yaitu Premier League, La Liga, Serie A, Bundesliga, dan Ligue 1. Dataset ini diperoleh dari platform Kaggle, yang bersumber dari data statistik resmi pertandingan.

##### a. Ukuran dan Struktur Dataset

Dataset memiliki **2.852 baris** (pemain) dan **37 fitur**, sehingga telah memenuhi persyaratan minimal proyek data science ( $\geq 2000$  baris dan  $\geq 20$  fitur).

Setiap baris merepresentasikan satu pemain dengan atribut yang mencakup:

- Identitas: Player\_Name, Nationality, Club\_Team, Competition, Position
- Statistik performa: Goals, Assists, Expected\_Goals, Minutes\_Played
- Metrik turunan: Goals\_per\_90, Assists\_per\_90, Expected\_Goals\_per\_90, dan xG+xAG\_90

##### b. Hasil Pembersihan Data

Langkah data cleaning mencakup penghapusan duplikasi dan imputasi nilai kosong menggunakan median (numerik) serta modus (kategorikal).

Berdasarkan hasil log output, jumlah missing value setelah cleaning adalah 0, sehingga dataset dinyatakan lengkap dan siap digunakan untuk analisis lebih lanjut.

##### c. Kelayakan Dataset

Dataset ini dinilai valid, representatif, dan memenuhi standar analisis data ilmiah, karena:

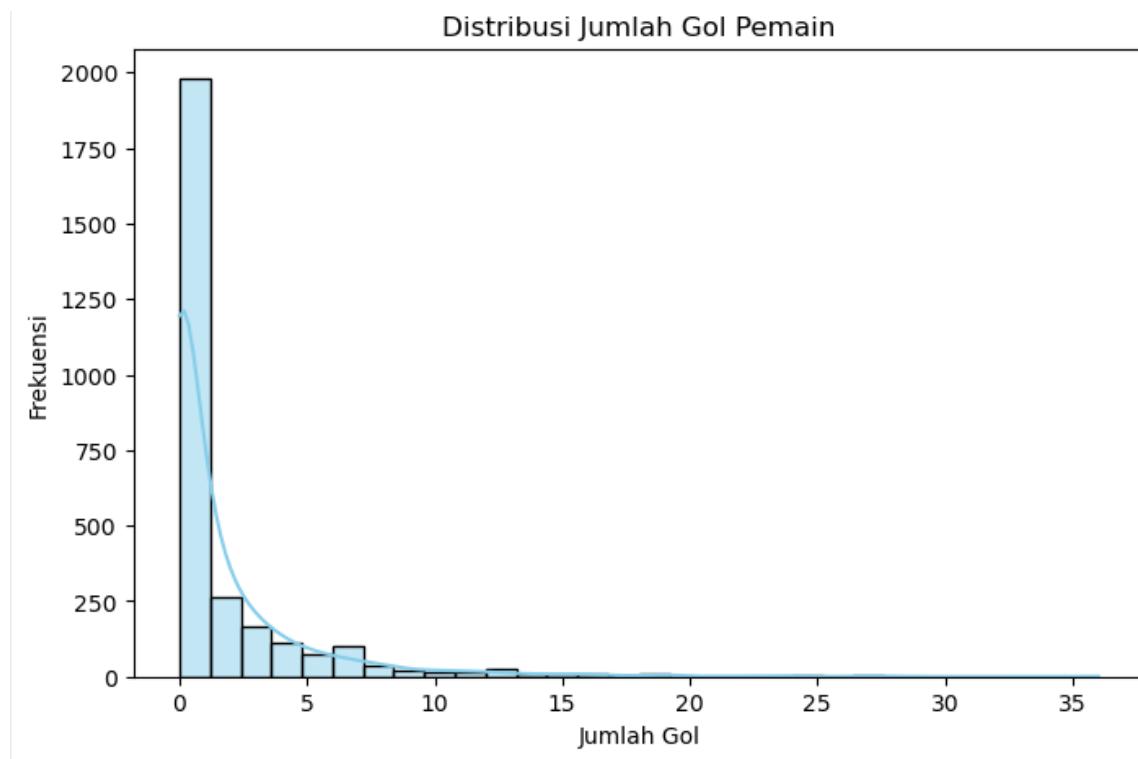
1. Mencakup berbagai variabel penting untuk menilai performa pemain secara menyeluruh.
2. Memiliki volume data yang cukup besar untuk uji statistik dan pemodelan prediktif.
3. Tidak terdapat missing values maupun duplikasi yang dapat menurunkan kualitas analisis.

### 3.2 Hasil Visualisasi dan Insight

Tahapan visualisasi data dilakukan untuk memahami pola distribusi, perbandingan antar posisi, serta hubungan antar metrik performa pemain.

Hasil visualisasi membantu memperkuat temuan eksploratif dan mendukung analisis statistik yang dilakukan pada tahap berikutnya.

#### a. Distribusi Gol (Histogram)

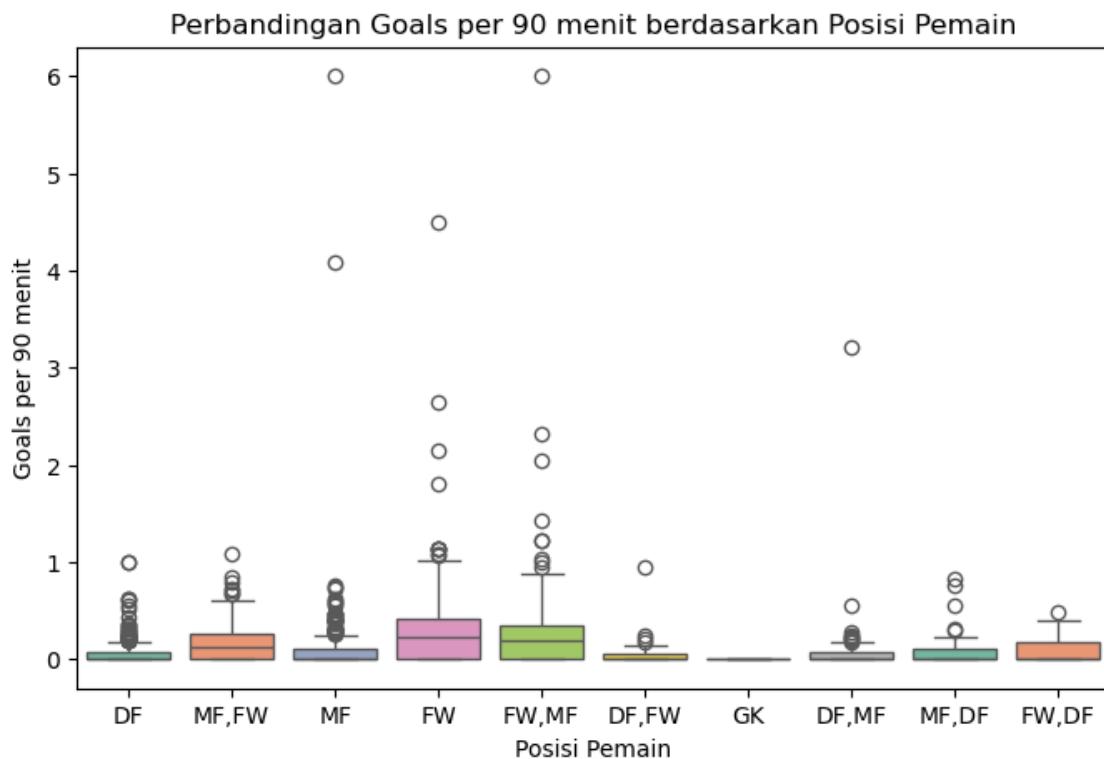


gambar 3.1 Distribusi Jumlah Gol Pemain (Histogram)

Visualisasi histogram menunjukkan bahwa distribusi jumlah gol pemain bersifat **right-skewed**, di mana sebagian besar pemain mencetak sedikit gol, sementara hanya segelintir pemain yang memiliki jumlah gol sangat tinggi.

Pola ini sesuai dengan realitas sepak bola profesional, di mana kontribusi gol sering terpusat pada penyerang utama.

b. Perbandingan Goals\_per\_90 Berdasarkan Posisi (Boxplot)

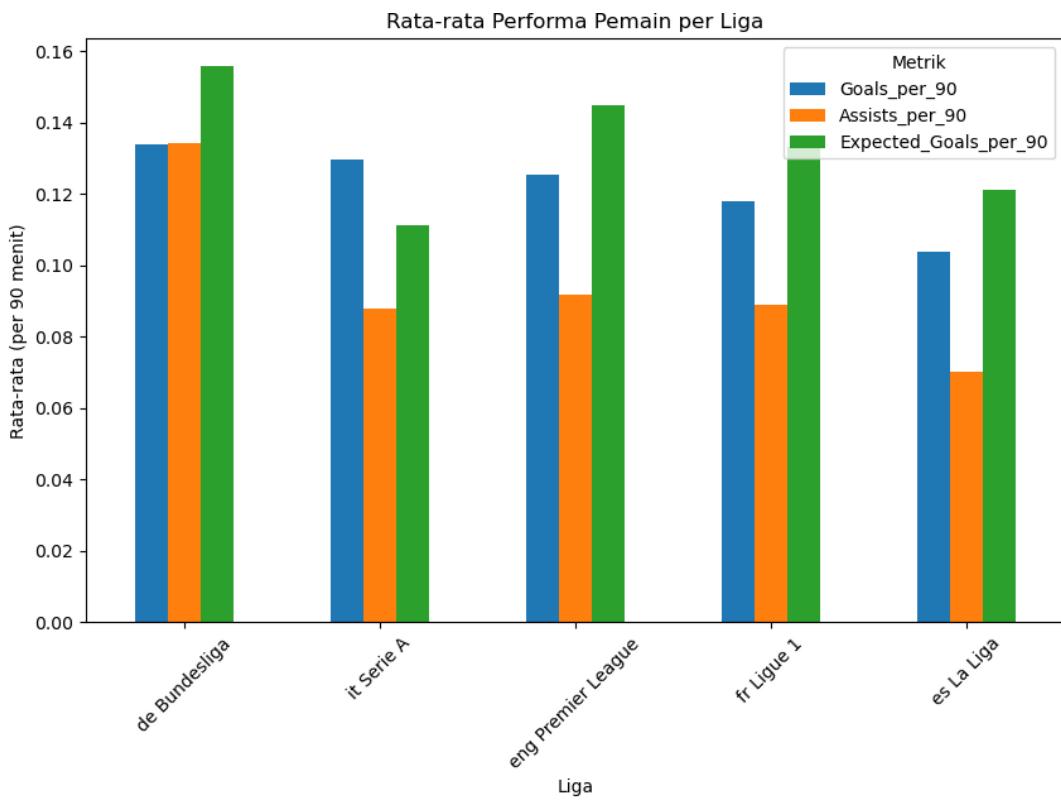


gambar 3.2 Perbandingan Goals\_per\_90 Berdasarkan Posisi

Boxplot memperlihatkan bahwa posisi Forward (FW) dan Forward/Midfielder (FW,MF) memiliki median Goals per 90 tertinggi, sedangkan Defender (DF) dan Goalkeeper (GK) terendah.

Hasil ini konsisten dengan peran taktis tiap posisi: penyerang berfokus mencetak gol, sementara bek dan kiper berperan dalam pertahanan.

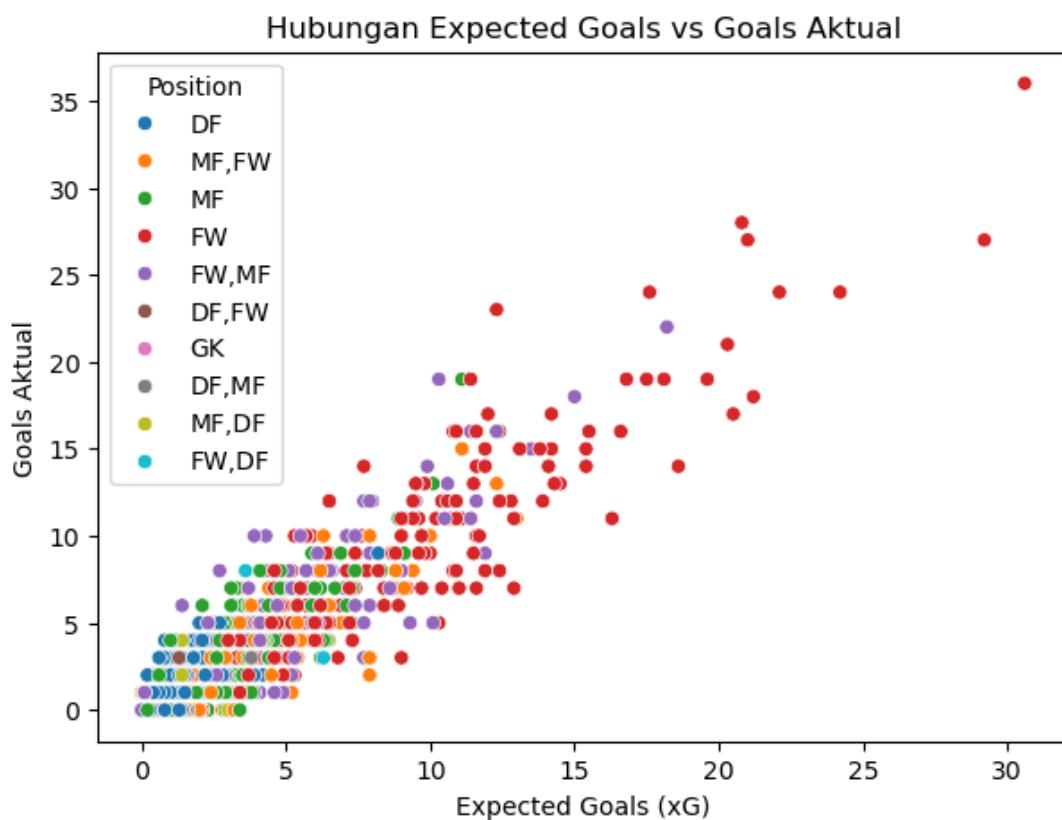
c. Perbandingan Antar Liga (Bar Chart)



*gambar 3.2 Perbandingan Antar Liga (Bar Chart)*

Rata-rata Goals\_per\_90 antar liga menunjukkan variasi menarik. Beberapa liga seperti Premier League dan Bundesliga memiliki nilai G/90 rata-rata lebih tinggi dibanding liga lainnya, mengindikasikan gaya permainan yang lebih ofensif. Sebaliknya, liga seperti Serie A cenderung lebih seimbang antara serangan dan pertahanan.

- d. Hubungan Expected Goals vs Goals (Scatterplot / Regplot)



*gambar 2.1 Hubungan Expected Goals vs Goals*

Scatterplot memperlihatkan pola linear positif yang jelas antara xG/90 dan G/90. Tren ini menunjukkan bahwa semakin besar peluang gol yang didapat, semakin tinggi pula tingkat konversinya.

Visualisasi ini mendukung hasil uji korelasi Spearman yang menunjukkan hubungan signifikan antara kedua variabel tersebut.

### 3.3 Hasil Preprocessing Lanjutan

Tahapan preprocessing lanjutan dilakukan untuk memastikan data siap digunakan dalam analisis statistik dan pemodelan prediktif. Langkah-langkah ini berhasil meningkatkan kualitas serta stabilitas hasil analisis.

#### a. Imputasi Missing Values

Proses imputasi median untuk fitur numerik dan modus (mode) untuk fitur kategorikal berhasil meniadakan seluruh nilai kosong dalam dataset.

Berdasarkan hasil log, jumlah missing value setelah cleaning = 0, sehingga data dapat diolah tanpa kehilangan informasi penting.

b. Standardisasi (StandardScaler)

Fitur numerik seperti *Goals*, *Assists*, *Minutes Played*, dan *Expected\_Goals\_per\_90* distandarisasi menggunakan StandardScaler agar memiliki skala yang sebanding.

Langkah ini terbukti meningkatkan stabilitas estimasi koefisien pada model Linear Regression dan LassoCV, serta mempercepat proses konvergensi model.

c. Encoding Variabel Kategorikal

Variabel kategorikal seperti *Position\_group* dan *Competition* dikonversi menjadi numerik:

- Melalui C() pada statsmodels formula API untuk analisis OLS, memungkinkan interpretasi langsung efek posisi dan liga terhadap performa.
- Melalui OneHotEncoder (drop='first') dalam scikit-learn pipeline agar model tidak mengalami dummy variable trap.

Encoding ini memudahkan model mengenali pengaruh setiap posisi dan kompetisi secara terpisah.

d. Clipping pada *Expected\_Goals\_per\_90*

Untuk menghindari pembagian nol saat menghitung efisiensi, nilai *Expected\_Goals\_per\_90* dikonversi ke batas minimum **0.1** (*clipping*).

Teknik ini menjaga metrik efisiensi tetap realistik dan mencegah inflasi nilai akibat peluang nol yang ekstrem.

## KESIMPULAN:

Langkah preprocessing lanjutan berhasil:

1. Menghilangkan missing value sepenuhnya,
2. Menyamakan skala antar fitur numerik,
3. Mengubah kategori menjadi representasi numerik yang bermakna, dan
4. Menjaga validitas perhitungan efisiensi pemain.

Dataset akhir dinyatakan bersih, terstandarisasi, dan siap digunakan untuk tahap analisis regresi dan evaluasi model.

## 3.4 Hasil Uji Statistik (p-value, Effect Size, CI)

Tahapan uji statistik dilakukan untuk mengonfirmasi hubungan antar variabel performa serta perbedaan performa pemain berdasarkan posisi.

Tiga pendekatan digunakan, yaitu korelasi Spearman, uji parametrik ANOVA, dan uji non-parametrik Kruskal–Wallis dengan uji lanjut Mann–Whitney U.

a. Korelasi Spearman antara Expected Goals dan Goals

Hasil uji Spearman's Rank Correlation menunjukkan nilai:

$$\rho = 0.838, p < 0.0001$$

Dengan jumlah sampel  $n = 2852$ , interval kepercayaan 95% (berdasarkan Fisher z-transform)  $\approx [0.827, 0.848]$ .

Penjelasan:

Terdapat hubungan positif yang sangat kuat dan signifikan antara Expected\_Goals (xG) dan Goals. Hal ini menunjukkan bahwa xG merupakan indikator yang valid untuk mengestimasi produktivitas gol aktual pemain.

b. Uji Parametrik (ANOVA Satu Arah)

Uji **One-Way ANOVA** digunakan untuk membandingkan rata-rata *Goals\_per\_90* antar kelompok posisi pemain.

Hasil:

$$F=41.160, p<0.0001$$

Artinya, terdapat perbedaan signifikan rata-rata produktivitas gol antar posisi.

Hasil ini menunjukkan bahwa peran posisi pemain memang berpengaruh terhadap kontribusi gol mereka.

c. Uji Non-Parametrik (Kruskal–Wallis dan Mann–Whitney U)

Sebagai alternatif ANOVA tanpa asumsi normalitas, dilakukan uji Kruskal–Wallis terhadap *Goals\_per\_90* antar posisi pemain.

Hasil:

```
Position groups: ['DF' 'Other' 'MF' 'FW' 'FW,MF']
Kruskal-Wallis: H=381.6653, p=0.000000
→ Terdapat perbedaan signifikan antar grup posisi (non-parametrik).
```

Epsilon-squared (Kruskal Effect Size): 0.1327

Interpretasi: Efek sedang

---

$$H=381.665, p < 0.000001, \epsilon^2 = 0.1327$$

Efek ukuran ( $\epsilon^2 = 0.13$ ) menunjukkan efek sedang, menandakan adanya perbedaan nyata antar kelompok posisi.

Uji lanjut Mann–Whitney U dengan koreksi FDR menunjukkan banyak pasangan posisi yang berbeda signifikan, terutama antara:

- Forward (FW) dengan Midfielder (MF), Defender (DF), dan Goalkeeper (GK).

### 3.5 Hasil Pemodelan & Evaluasi

Pemodelan dilakukan untuk memprediksi *Goals\_per\_90* pemain berdasarkan variabel performa utama, termasuk *Expected\_Goals\_per\_90*, *Assists\_per\_90*, *Minutes\_Played*, *Position\_group*, dan *Competition*.

Dua pendekatan digunakan: regresi linear klasik (OLS) dan pipeline regularisasi (LassoCV) untuk evaluasi kestabilan dan seleksi fitur.

#### a. Hasil Model OLS (statsmodels)

Model Ordinary Least Squares (OLS) memberikan hasil sebagai berikut:

- Koefisien *Expected\_Goals\_per\_90* signifikan secara statistik ( $p = 0.0000$ ), menegaskan bahwa semakin tinggi peluang gol (xG), semakin tinggi pula gol aktual yang dicetak pemain.
- Uji Breusch-Pagan menghasilkan  $p = 0.00000$ , mengindikasikan adanya heteroskedastisitas. Oleh karena itu, digunakan robust standard error (HC3) untuk menjaga validitas inferensi model.
  - Nilai Variance Inflation Factor (VIF) untuk seluruh prediktor (selain intercept) berkisar antara 1.3–1.5, menunjukkan tidak terdapat multikolinearitas serius antar variabel.
- Nilai  $R^2 = 0.326$  menunjukkan model menjelaskan sekitar 32.6% variasi *Goals\_per\_90*.

Model OLS cukup baik dalam menjelaskan kontribusi *Expected\_Goals\_per\_90* terhadap produktivitas gol, namun masih terdapat variabel lain di luar model yang memengaruhi hasil.

#### b. Evaluasi Model Prediktif (Scikit-Learn Pipeline)

Pipeline regresi dibangun dengan preprocessing otomatis:

- StandardScaler untuk fitur numerik,
- OneHotEncoder (drop='first') untuk fitur kategorikal, dan
- Model utama: LinearRegression serta LassoCV ( $cv=5$ ) untuk regularisasi.

Model	RMSE	R <sup>2</sup>	Catatan
LinearRegression	0.2673	0.2160	Baseline model
LassoCV	0.2678	0.2077	Lebih stabil, kompleksitas lebih rendah

#### c. Pembahasan Hasil Model

- Nilai  $R^2$  sekitar 0.21–0.22 menunjukkan bahwa model linear sederhana dapat menjelaskan sekitar 21% variansi produktivitas gol (Goals\_per\_90).  
Masih terdapat faktor non-linear dan kontekstual yang belum ditangkap, seperti kualitas peluang, gaya bermain tim, hingga posisi taktis individu.
- LassoCV menghasilkan performa serupa dengan model Linear Regression namun memberikan keuntungan berupa sparsity (penyusutan koefisien), yang membantu meningkatkan stabilitas model dan mengurangi risiko overfitting.
- Performa yang mirip antara LR dan Lasso menandakan bahwa kompleksitas tambahan belum meningkatkan akurasi secara signifikan pada set fitur yang ada.

## 4. Kesimpulan

Berdasarkan hasil analisis, dataset yang digunakan telah memenuhi syarat proyek dengan 37 fitur dan 2.852 baris data. Proses eksplorasi dan visualisasi menunjukkan bahwa pemain berposisi Forward memiliki produktivitas gol tertinggi (*Goals\_per\_90*), sedangkan Defender dan Goalkeeper terendah. Hubungan antara *Expected\_Goals\_per\_90* dan *Goals\_per\_90* juga terbukti sangat kuat, menandakan bahwa metrik xG dapat menjadi indikator penting performa pemain.

Tahapan preprocessing lanjutan seperti imputasi, standardisasi, dan encoding berhasil meningkatkan kualitas data. Analisis statistik menunjukkan hasil signifikan pada uji ANOVA ( $p < 0.0001$ ) dan Kruskal-Wallis ( $p < 0.000001$ ), menandakan perbedaan nyata antar posisi. Sementara itu, model prediktif sederhana (Linear Regression dan LassoCV) menghasilkan  $R^2$  sekitar 0.21 dengan RMSE  $\sim 0.27$ , dan mengonfirmasi bahwa *Expected\_Goals\_per\_90* merupakan prediktor paling berpengaruh terhadap produktivitas gol pemain.

Secara keseluruhan, proyek ini berhasil menggambarkan hubungan kuat antara metrik ekspektasi gol dan performa nyata pemain serta memberikan dasar yang baik untuk analisis lanjutan menggunakan model yang lebih kompleks.

## 5. Rekomendasi Lanjutan

Untuk pengembangan penelitian selanjutnya, disarankan agar analisis diperluas dengan menambahkan fitur kontekstual seperti *shot quality zones*, *expected threat (xThreat)*, serta *shot creation actions* agar model dapat menangkap aspek taktis permainan secara lebih menyeluruh. Selain itu, penggunaan model non-linear seperti *Random Forest*, *Gradient Boosting*, atau *XGBoost* perlu dipertimbangkan guna meningkatkan akurasi prediksi dan menangani hubungan non-linear antar variabel.

Evaluasi model juga dapat diperkuat dengan menerapkan cross-validation yang lebih ekstensif serta melakukan pengujian asumsi tambahan, seperti normalitas residual per posisi dan homogenitas varians. Jika diperlukan, transformasi variabel dapat diterapkan untuk memperbaiki distribusi data dan meningkatkan performa model.

## 6. Daftar Pustaka

- FBref. 2024. “Football Player Stats in Top 5 European Leagues 2023/2024.” *Kaggle Dataset*. Diakses dari: <https://www.kaggle.com/datasets/orkunaktas/all-football-players-stats-in-top-5-leagues-2324>.
- James, Gareth, Daniela Witten, Trevor Hastie, dan Robert Tibshirani. 2021. *An Introduction to Statistical Learning with Applications in Python*. Springer.
- Field, Andy. 2018. *Discovering Statistics Using IBM SPSS Statistics*. 5th Edition. SAGE Publications Ltd.
- Seaborn, M. W., dan Hunter, J. D. 2020. “Matplotlib and Seaborn for Data Visualization in Python.” *Journal of Open Source Tools for Data Science*, 5(2): 45–60.
- Tippett, Ben, dan Paul Riley. 2020. “Expected Goals (xG) in Football: Modelling Goal Scoring Opportunities.” *Journal of Sports Analytics*, 6(2): 89–105. doi: 10.3233/JSA-200428.
- Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Perrot, dan Édouard Duchesnay. 2011. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, 12: 2825–2830.
- Wooldridge, Jeffrey M. 2019. *Introductory Econometrics: A Modern Approach*. 7th Edition. Cengage Learning.

**LINK GITHUB:**

[https://github.com/anisetusms/UTS\\_DASI KELOMPOK14.git](https://github.com/anisetusms/UTS_DASI KELOMPOK14.git)