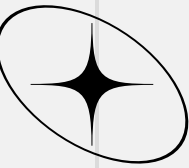


2024

✦ **STUDENT SCORE** ✦
PREDICT





DESKRIPSI DATA

- **Jam Belajar (Hours):** Waktu yang dihabiskan oleh siswa untuk belajar.
- **Nilai (Scores):** Hasil atau nilai yang diperoleh oleh siswa setelah belajar.

No	Hours	Scores
1	2.5	21
2	5.1	47
3	3.2	27
4	8.5	75
5	3.5	30
.....

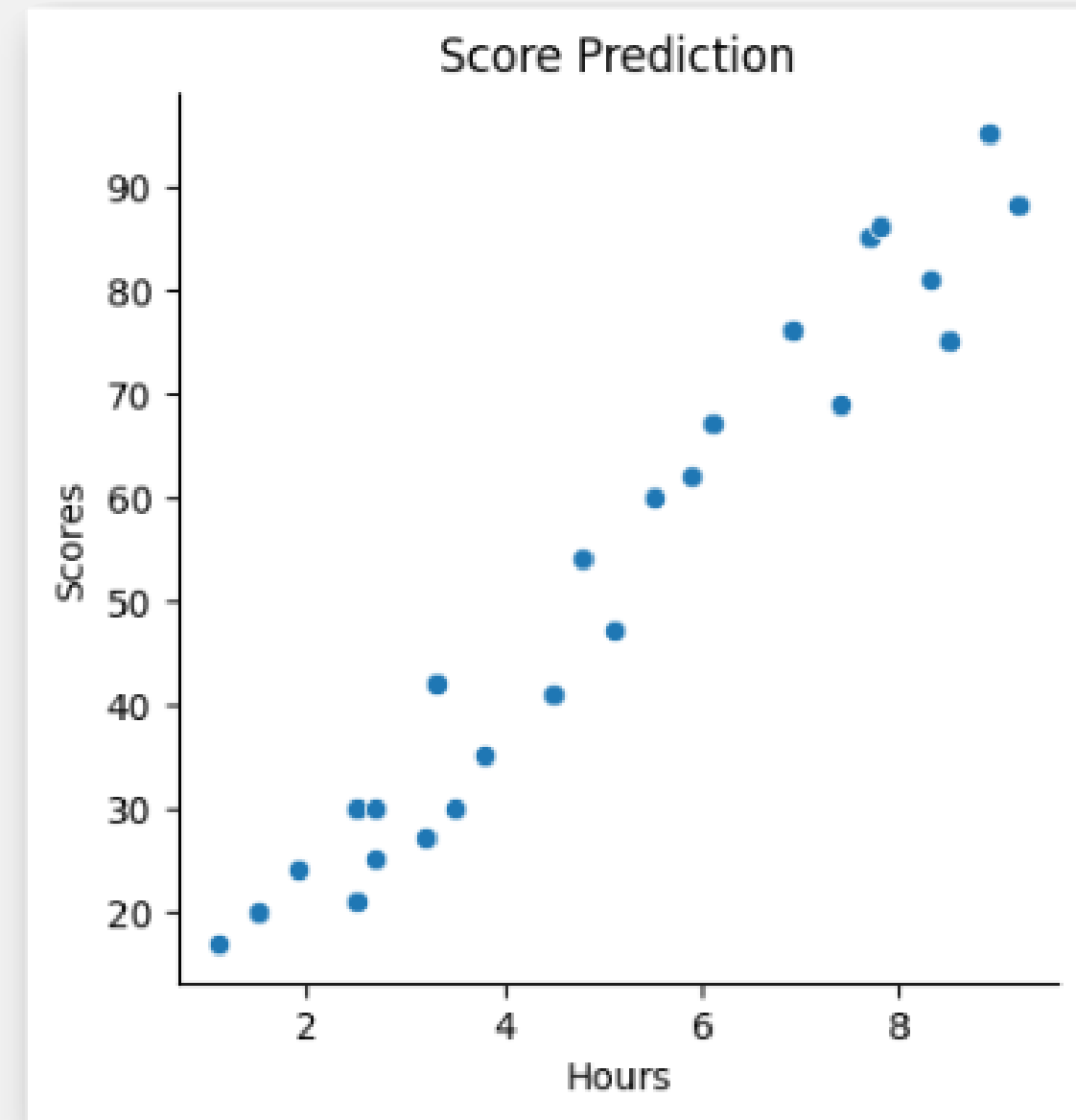
INTERPRETASI SEBARAN DATA

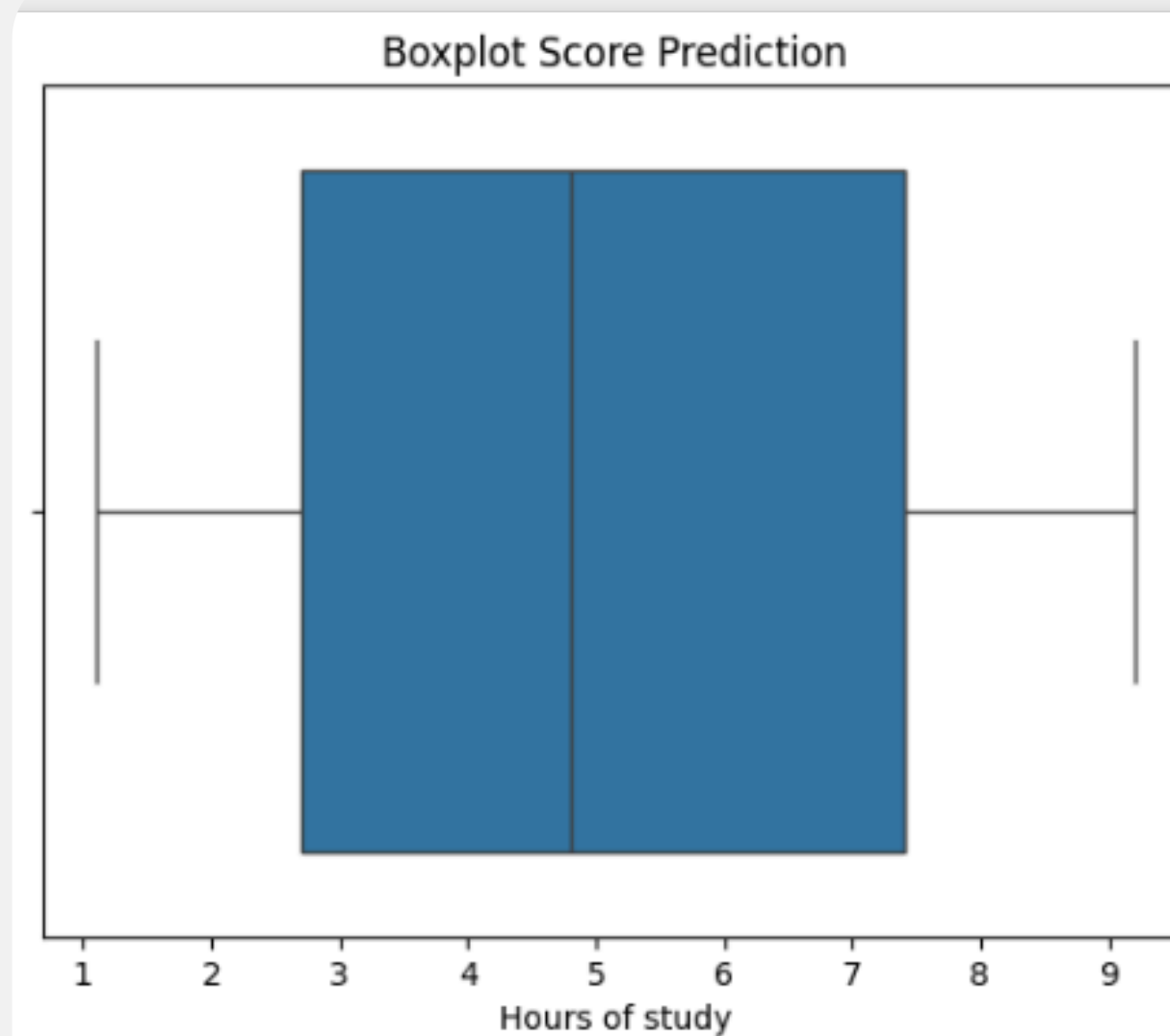
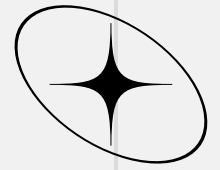
SUMBU

- Sumbu X: Hours (Jam Belajar)
- Sumbu Y: Scores (Nilai)

ANALISIS GRAFIK

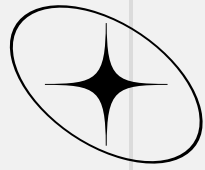
- Data tersebar cukup merata dan memiliki pola yang hampir linear
- Korelasi Positif dalam sebaran data, semakin tinggi jam belajar cenderung berkaitan dengan peningkatan nilai





INTERPRETASI BOXPLOT

Boxplot menunjukkan bahwa sebagian besar siswa belajar antara 3,5 hingga 7,5 jam, dengan median sekitar 5 jam. Rentang belajar cukup lebar dan tidak ada nilai yang ekstrem, menunjukkan distribusi data yang merata.



MODELING

Berikut ini tahapan modeling menggunakan algoritma Random Forest dan Linear Regression

```
from sklearn.ensemble import RandomForestRegressor

# Inisialisasi model Random Forest
model_rf = RandomForestRegressor(n_estimators=100, random_state=42)

# Latih model
model_rf.fit(X_train, y_train)

# Prediksi pada data uji
predictions_rf = model_rf.predict(X_test)

# Evaluasi model
mse_rf = mean_squared_error(y_test, predictions_rf)
print("Mean Squared Error (MSE) for Random Forest:", mse_rf)
r_squared_rf = model_rf.score(X_test, y_test)
print("R-Squared for Random Forest:", r_squared_rf)
```

Mean Squared Error (MSE) for Random Forest: 43.03697569444442
R-Squared for Random Forest: 0.9360862307020844

Random Forest

Proses dasar dari penggunaan model Random Forest untuk regresi, yang mencakup inisialisasi model, pelatihan model dengan data pelatihan, dan prediksi menggunakan data uji.

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Inisialisasi model regresi linear
model_lr = LinearRegression()

# Latih model
model_lr.fit(X_train, y_train)

# Prediksi pada data uji
predictions_lr = model_lr.predict(X_test)

# Evaluasi model
mse_lr = mean_squared_error(y_test, predictions_lr)
print("Mean Squared Error (MSE) for Linear Regression:", mse_lr)

intercept_lr = model_lr.intercept_
print("Intercept for Linear Regression:", intercept_lr)

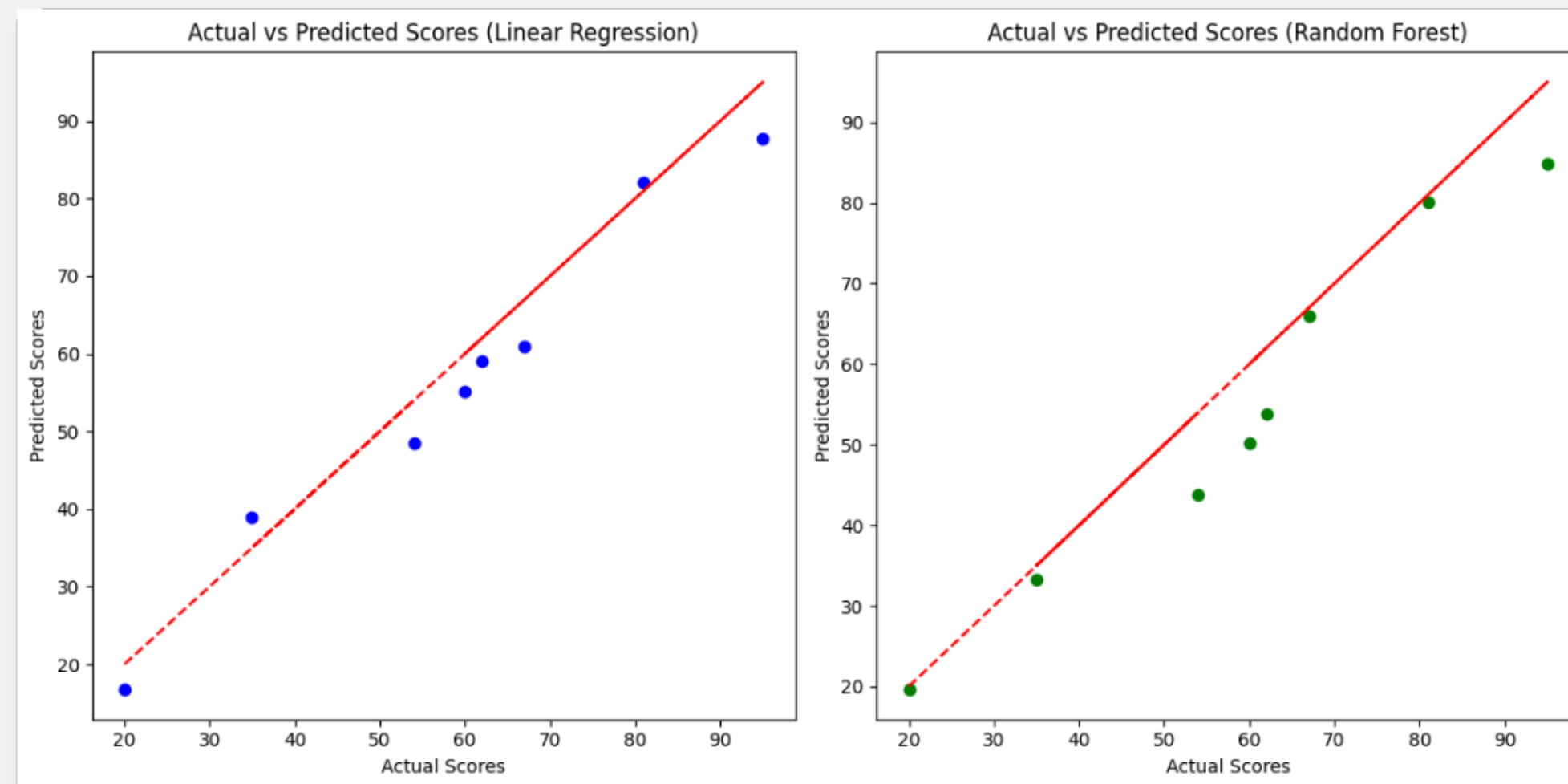
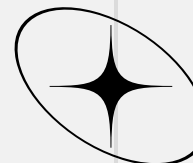
coef_lr = model_lr.coef_
print("Coefficients for Linear Regression:", coef_lr)

r_squared_lr = model_lr.score(X_test, y_test)
print("R-Squared for Linear Regression:", r_squared_lr)
```

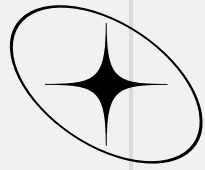
Mean Squared Error (MSE) for Linear Regression: 26.582796488881115
Intercept for Linear Regression: 2.695388922345728
Coefficients for Linear Regression: [9.60171878]
R-Squared for Linear Regression: 0.9605221627526418

Linear Regression

Proses dasar dari penggunaan model Linear Regression untuk regresi, yang mencakup inisialisasi model, pelatihan model dengan data pelatihan, dan prediksi menggunakan data uji.



ACTUAL VS PREDICT SCORES PLOT



R-SCORE MODEL

Kesimpulannya, nilai R-Squared untuk Linear Regression sedikit lebih tinggi dibandingkan dengan Random Forest. Ini menunjukkan bahwa model Linear Regression memiliki kemampuan yang sedikit lebih baik dalam memprediksi skor siswa dengan melihat jam belajar dalam data yang ada diatas daripada model Random Forest.

▼ Membandingkan R-Score Model

```
✓  
0s [48] print("R-Score Linear Regression:", r_squared_lr)  
      print("R-Score Random Forest:", r_squared_rf)
```

```
⇒ R-Score Linear Regression: 0.9549410139157178  
  R-Score Random Forest: 0.9056526081624953
```