

Progres Project Praktik Aplikasi Web

Analisis Saham dengan Regresi Berganda IDX 10

Prabatha Nandhi Wardhana // 20537144017

- Pendahuluan :
Saya membuat sebuah platform sederhana yang berfungsi menampilkan berbagai insight 10 saham dengan kapitalisasi terbesar di Indonesia dan yang utama mencoba mengembangkan metode regresi berganda untuk membantu memprediksi harga penutupan saham.

- Progres Pengembangan :

Sudah membuat web streamlit

Kode Program :

```
• import streamlit as st
• import pandas as pd
• import matplotlib.pyplot as plt
• import seaborn as sns
• import mplfinance as mpf
• from sklearn.model_selection import train_test_split
• from sklearn.linear_model import LinearRegression
• from sklearn.metrics import r2_score
•
• def load_css():
•     css = """
•     <style>
•         /* Warna dasar */
•         body {
•             color: #333;
•             background-color: #f4f4f2;
•         }
•         /* Warna tema untuk saham */
•         .rising {
•             color: #4caf50; /* hijau */
•         }
•         .falling {
•             color: #f44336; /* merah */
•         }
•         /* Styling judul dan subjudul */
•         h1 {
•             color: #007bff;
•         }
•         h2, h3, h4 {
•             color: #0056b3;
•         }
•     """
```

```

•         /* Styling button */
•         .stButton>button {
•             color: white;
•             background-color: #007bff;
•             border: none;
•             border-radius: 4px;
•             padding: 10px 24px;
•             margin: 10px 0;
•             cursor: pointer;
•         }
•         .stButton>button:hover {
•             background-color: #0056b3;
•         }
•         /* Styling tabel dan grafik */
•         .stDataFrame, .stPlotlyChart {
•             border: 1px solid #ddd;
•             border-radius: 5px;
•         }
•     </style>
•     """
•     st.markdown(css, unsafe_allow_html=True)
•
• load_css()
•
• def load_stock_data(stock_path):
•     data = pd.read_csv(stock_path, index_col='Date', parse_dates=True)
•     return data
•
• def plot_time_series(data, stock_name):
•     plt.figure(figsize=(10, 6))
•     sns.set_style("whitegrid")
•     plt.plot(data[['Open', 'High', 'Low', 'Close']])
•     plt.title(f'Pergerakan Harga Saham {stock_name} (Time Series)',
•               fontsize=16)
•     plt.xlabel('Tanggal', fontsize=14)
•     plt.ylabel('Harga', fontsize=14)
•     plt.legend(['Open', 'High', 'Low', 'Close'], loc='upper left')
•     st.pyplot(plt)
•     plt.clf()
•
• def plot_volume(data, stock_name):
•     plt.figure(figsize=(10, 6))
•     sns.set_style("whitegrid")
•     plt.plot(data['Volume'], color='blue')
•     plt.title(f'Volume Perdagangan Saham {stock_name}', fontsize=16)
•     plt.xlabel('Tanggal', fontsize=14)
•     plt.ylabel('Volume', fontsize=14)
•     st.pyplot(plt)

```

```

• plt.clf()
•
• def plot_heatmap(data, stock_name):
•     plt.figure(figsize=(10, 6))
•     sns.heatmap(data[['Open', 'High', 'Low', 'Close',
• 'Volume']].corr(), annot=True, cmap='coolwarm')
•     plt.title(f'Heatmap Korelasi Atribut Saham {stock_name}',
• fontsize=16)
•     st.pyplot(plt)
•     plt.clf()
•
• def plot_scatter(data, stock_name):
•     plt.figure(figsize=(6, 6))
•     sns.scatterplot(data=data, x='High', y='Close')
•     plt.title(f'Hubungan Harga Tertinggi vs Harga Penutupan Saham
• {stock_name}', fontsize=14)
•     st.pyplot(plt)
•     plt.clf()
•
•     plt.figure(figsize=(6, 6))
•     sns.scatterplot(data=data, x='Low', y='Close')
•     plt.title(f'Hubungan Harga Terendah vs Harga Penutupan Saham
• {stock_name}', fontsize=14)
•     st.pyplot(plt)
•     plt.clf()
•
•     plt.figure(figsize=(6, 6))
•     sns.scatterplot(data=data, x='Volume', y='Close')
•     plt.title(f'Hubungan Volume Perdagangan vs Harga Penutupan Saham
• {stock_name}', fontsize=14)
•     st.pyplot(plt)
•     plt.clf()
•
• def plot_candlestick(data, stock_name):
•     mpf.plot(data, type='candle', style='charles', title=f'Candlestick
• Chart Saham {stock_name}', ylabel='Harga', mav=(20, 50), volume=True,
• show_nontrading=True)
•     st.pyplot(plt)
•     plt.clf()
•
• def show_regression_model(data, stock_name):
•     features = data[['Open', 'High', 'Low', 'Volume']]
•     target = data['Close']
•     x_train, x_test, y_train, y_test = train_test_split(features,
• target, test_size=0.3, random_state=0)
•     model = LinearRegression()
•     model.fit(x_train, y_train)
•     y_pred_train = model.predict(x_train)

```

```

•     y_pred_test = model.predict(x_test)
•
•     st.write(f'Model Regresi untuk {stock_name}')
•     st.write(f'Train R2 Score: {r2_score(y_train, y_pred_train)}')
•     st.write(f'Test R2 Score: {r2_score(y_test, y_pred_test)}')
•
•     # Penjelasan R-squared
•     st.write("R-squared adalah ukuran yang menggambarkan seberapa baik
model ini cocok dengan data.")
•     st.write("Nilainya berkisar antara 0 hingga 1, di mana:")
•     st.write("- Nilai Mendekati 0: Menunjukkan bahwa model regresi
tidak menjelaskan variabilitas data dengan baik.")
•     st.write("  Dengan kata lain, model tersebut tidak cocok dengan
data secara efektif.")
•     st.write("- Nilai Mendekati 1: Menunjukkan bahwa model regresi
menjelaskan proporsi yang besar dari variabilitas dalam data.")
•     st.write("  Ini berarti model tersebut memberikan prediksi yang
baik terhadap data yang diberikan.")
•
•     return model
•
• def predict_price(model, stock_name):
•     st.header(f"Prediksi Harga Saham - {stock_name}")
•     open_price = st.number_input('Harga Pembukaan', min_value=0.0,
format='%f')
•     high_price = st.number_input('Harga Tertinggi', min_value=0.0,
format='%f')
•     low_price = st.number_input('Harga Terendah', min_value=0.0,
format='%f')
•     volume = st.number_input('Volume', min_value=0, format='%d')
•
•     if st.button('Prediksi'):
•         predicted_price = model.predict([[open_price, high_price,
low_price, volume]])[0]
•         st.write(f"Harga Penutupan yang Diprediksi: {predicted_price}")
•
• stock_paths = {
•     "AMMN":
"D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/AMMN.JK.csv",
•     "ASII":
"D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/ASII.JK.csv",
•     "BBCA":
"D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/BBCA.JK.csv",
•     "BBRI":
"D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/BBRI.JK.csv",
•     "BMRI":
"D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/BMRI.JK.csv",

```

```

•     "BYAN":
•     "D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/BYAN.JK.csv",
•     "BBNI":
•     "D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/BBNI.JK.csv",
•     "BYAN":
•     "D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/BYAN.JK.csv",
•     "BREN":
•     "D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/BREN.JK.csv",
•     "TPIA":
•     "D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/TPIA.JK.csv",
•     "TLKM":
•     "D:/UNY/semester5/praktik_aplikasi_web/Project/PRAKTIK/TLKM.JK.csv"
•
• }
•
• def analysis_page():
•     selected_stock = st.sidebar.selectbox("Pilih Saham",
• list(stock_paths.keys()))
•     data = load_stock_data(stock_paths[selected_stock])
•
•     st.title("ANALISIS SAHAM DENGAN REGRESI BERGANDA IDX 10")
•     page = st.sidebar.selectbox("Pilih Halaman", ["Grafik Time Series",
"Volume Perdagangan", "Heatmap Korelasi", "Grafik Scatter",
"Candlestick Chart", "Model Regresi", "Prediksi Harga Saham"])
•
•     if page == "Grafik Time Series":
•         plot_time_series(data, selected_stock)
•     elif page == "Volume Perdagangan":
•         plot_volume(data, selected_stock)
•     elif page == "Heatmap Korelasi":
•         plot_heatmap(data, selected_stock)
•     elif page == "Grafik Scatter":
•         plot_scatter(data, selected_stock)
•     elif page == "Candlestick Chart":
•         plot_candlestick(data, selected_stock)
•     elif page == "Model Regresi":
•         trained_model = show_regression_model(data, selected_stock)
•         st.session_state['trained_model'] = trained_model
•     elif page == "Prediksi Harga Saham":
•         if 'trained_model' in st.session_state:
•             predict_price(st.session_state['trained_model'],
selected_stock)
•         else:
•             st.write("Model belum dilatih, silakan kembali ke halaman
'Model Regresi'.")
•
•     if st.sidebar.button("Kembali ke Halaman Utama"):
•         st.session_state['page'] = 'Landing'

```

```

•         return
•
•     def landing_page():
•         st.title("ANALISIS SAHAM DENGAN REGRESI BERGANDA IDX 10")
•
•         st.write("""
•         Platform ini dirancang untuk memberikan analisis tentang sepuluh
•         saham dengan kapitalisasi terbesar di Indonesia (IDX 10).
•         Dengan menggunakan data historis terkini, kami mencoba menyediakan
•         berbagai insight yang menarik tentang saham tersebut.
•         """)
•
•         st.subheader("Fitur Utama:")
•         st.markdown("""
•         - Grafik Time Series
•         - Grafik Volume Perdagangan
•         - Grafik Scatter
•         - Grafik Candlestick
•         - Heatmap Korelasi
•         - Prediksi Harga Saham
•         """)
•
•
•         if st.button("Mulai Analisis"):
•             st.session_state['page'] = 'Analisis'
•
•     if 'page' not in st.session_state:
•         st.session_state['page'] = 'Landing'
•
•     if st.session_state['page'] == "Landing":
•         landing_page()
•     else:
•         analysis_page()
•

```

- Hasil Screenshot :

Landing page :



Halaman Analisis :

