

DAFTAR PUSTAKA

- Andrean, P. (2019). *PENERAPAN METODE K-NN UNTUK MEMPREDIKSI HASIL PERTANIAN DI KABUPATEN MALANG*. 3(1).
- Bayley, J. (2024, February 27). *SEVENTEEN's FML announced by IFPI as biggest-selling Global Album of the Year*. IFPI. <https://www.ifpi.org/seventeens-fml-announced-by-ifpi-as-biggest-selling-global-album-of-the-year/>
- Data Mining and Statistics: What's the Connection?* (2024, September 28). ResearchGate. https://www.researchgate.net/publication/337324537_Data_Mining_and_Statistics_What's_the_Connection
- E Cha. (2024, May 5). *SEVENTEEN Achieves Highest 1st-Week Sales Of 2024 With "17 IS RIGHT HERE."* Soompi. <https://www.soompi.com/article/1659265wpp/seventeen-achieves-highest-1st-week-sales-of-2024-with-17-is-right-here>
- Ema Utami, Prabowo Budi Utomo, & Suwanto Raharjo. (2019). IMPLEMENTASI METODE K-NEAREST NEIGHBOR DAN REGRESI LINEAR DALAM PREDIKSI HARGA EMAS. *JURNAL INFORMASI INTERAKTIF*, 4(3). <https://journal.shantibhuana.ac.id/index.php/jifotech/article/view/519>
- Hesty Kartikasari & Arief Sudrajat. (2024). MAKNA PEMBELIAN ALBUM FISIK BAGI PENGEMAR BUDAYA POP KOREA. *Jurnal Analisa Sosiologi*. <https://doi.org/10.20961/jas.v11i3.59242>

Ihzanah, L. S., Setiawan, A., & Wijaya, R. W. N. (2023). Perbandingan Kinerja Metode Regresi K-Nearest Neighbor dan Metode Regresi Linear Berganda pada Data Boston Housing. *Jambura Journal of Probability and Statistics*, 4(1), 17–29. <https://doi.org/10.34312/jjps.v4i1.18948>

Ina Marthiani. (2024). Uji Validitas dan Reliabilitas Instrumen Penelitian Pemahaman Konsep Biologi. *Jurnal Yudistira : Publikasi Riset Ilmu Pendidikan dan Bahasa*, 2(2), 351–356. <https://doi.org/10.61132/yudistira.v2i2.727>

Inawati, W., Kurniawan, R., & Wijaya, Y. A. (2024). Penerapan Aplikasi Rapidminer pada Prediksi Data Penjualan Album K-Pop Menggunakan Metode Regresi Linear. *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, 5(3), Article 3. <https://doi.org/10.30998/jrami.v5i3.10648>

Kafil, M. (2019). PENERAPAN METODE K-NEAREST NEIGHBORS UNTUK PREDIKSI PENJUALAN BERBASIS WEB PADA BOUTIQ DEALOVE BONDOWOSO. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 3(2), 59–66. <https://doi.org/10.36040/jati.v3i2.860>

Koivukoski, I. (2023). *Marketing of K-pop group Seventeen*. Haaga-Helia University of Applied Sciences.

K-pop Events Market Trends, Demand | Industry Forecast 2031. (n.d.). Retrieved November 10, 2024, from <https://www.alliedmarketresearch.com/press-release/k-pop-events-market.html>

- Kuriakose, A. M., Kariyalil, D. P., Augusthy, M., Sarath, S., Jacob, J., & Antony, N. R. (2020). Comparison of Artificial Neural Network, Linear Regression and Support Vector Machine for Prediction of Solar PV Power. *2020 IEEE Pune Section International Conference (PuneCon)*, 1–6.
<https://doi.org/10.1109/PuneCon50868.2020.9362442>
- Kurniaty, A., Ekojono, & Rohadi, E. (2023). SISTEM PENDUKUNG KEPUTUSAN TEMPAT PEMBELIAN ALBUM KPOP MENGGUNAKAN MOORA. *Jurnal Informatika Polinema*, 9(2), Article 2.
<https://doi.org/10.33795/jip.v9i2.1157>
- Kurniawan, I., Rahaningsih, N., & Suprapti, T. (2024). IMPLEMENTASI ALGORITMA REGRESI LINIER DAN K-NEAREST NEIGHBOR UNTUK PREDIKSI HARGA RUMAH. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(1), 1187–1193. <https://doi.org/10.36040/jati.v8i1.8402>
- Kurniawan, R., & Apriliani, A. (2020). ANALISIS SENTIMEN MASYARAKAT TERHADAP VIRUS CORONA BERDASARKAN OPINI DARI TWITTER BERBASIS WEB SCRAPER. *Jurnal INSTEK (Informatika Sains Dan Teknologi)*, 5(1), Article 1.
<https://doi.org/10.24252/instek.v5i1.13686>
- Kusuma, P. D. (2020). *Machine Learning: Teori, Program, dan Studi Kasus*. Deepublish.

<https://openlibrary.telkomuniversity.ac.id/pustaka/160514/machine-learning-teori-program-dan-studi-kasus.html>

- Lestari, S. (2023). Analisis Algoritma Regresi Linear Sederhana dalam Memprediksi Tingkat Penjualan Album KPOP. *INSOLOGI: Jurnal Sains dan Teknologi*, 2(1), 199–209. <https://doi.org/10.55123/insologi.v2i1.1692>
- M. Riziq Sirfatullah Alfarizi, Muhammad Zidan Al-farish, & Muhammad Taufiqurrahman. (2023). PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING. *Karimah Tauhid*, 2(1).
- Muharrom, M. (2023). Analisis Komparasi Algoritma Data Mining Naive Bayes, K-Nearest Neighbors dan Regresi Linier Dalam Prediksi Harga Emas. *Bulletin of Information Technology (BIT)*, 4(4), 430–438. <https://doi.org/10.47065/bit.v4i4.986>
- Nuha, H. (2023). *Mean Squared Error (MSE) dan Penggunaannya (SSRN Scholarly Paper No. 4420880)*. Social Science Research Network. <https://papers.ssrn.com/abstract=4420880>
- Nurani, A. T., Setiawan, A., & Susanto, B. (2023). Perbandingan Kinerja Regresi Decision Tree dan Regresi Linear Berganda untuk Prediksi BMI pada Dataset Asthma. *Jurnal Sains dan Edukasi Sains*, 6(1), 34–43. <https://doi.org/10.24246/juses.v6i1p34-43>

- Putri, V. K., & Reese, M. (2016). THE IMPACT OF 'KOREAN WAVE' ON YOUNG INDONESIAN FEMALES AND INDONESIAN CULTURE IN JABODETABEK AREA. *Emerging Markets : Business and Management Studies Journal*, 4(1), Article 1. <https://doi.org/10.33555/ijembm.v4i1.60>
- Safitri, W. R. (2016). ANALISIS KORELASI PEARSON DALAM MENENTUKAN HUBUNGAN ANTARA KEJADIAN DEMAM BERDARAH DENGUE DENGAN KEPADATAN PENDUDUK DI KOTA SURABAYA PADA TAHUN 2012—2014. *Jurnal Ilmiah Keperawatan (JIKEP)*, 2(2).
- Sarangpure, N., Dhamde, V., Roge, A., Doye, J., Patle, S., & Tamboli, S. (2023). Automating the Machine Learning Process using PyCaret and Streamlit. *2023 2nd International Conference for Innovation in Technology (INOCON)*, 1–5. <https://doi.org/10.1109/INOCON57975.2023.10101357>
- Service (KOCIS), K. C. and I. (n.d.). *No. of global Hallyu fans last year broke 200M mark: Survey : Korea.net : The official website of the Republic of Korea*. Retrieved November 10, 2024, from <https://www.korea.net/NewsFocus/Society/view?articleId=248349>
- Sholihah, S. M., Aditiya, N. Y., Evani, E. S., & Maghfiroh, S. (2023). KONSEP UJI ASUMSI KLASIK PADA REGRESI LINIER BERGANDA. 2(2).

- Siti Nur Asiyah & Kartika Fithriasari. (2016). Klasifikasi Berita Online Menggunakan Metode Support Vector Machine dan K- Nearest Neighbor. *JURNAL SAINS DAN SENI ITS*, 5(2), 2337–3520.
- Syafarina, G. A., & Zaenuddin, Z. (2023). Implementasi Framework Streamlit Sebagai Prediksi Harga Jual Rumah Dengan Linear Regresi. *METIK JURNAL*, 7(2), Article 2. <https://doi.org/10.47002/metik.v7i2.608>
- Uci Juwita Agustiani, Hendrayudi, & Rusidi. (2021). SISTEM INFORMASI PENDAFTARAN DI RSIA PRIMA QONITA MENGGUNAKAN PHP MYSQL BERBASIS WEB. *Jurnal Informatika dan Komputer (JIK)*, 12(2).
- Wijaya, A. R. (n.d.). *Model Prediksi Data Harga Minyak Mentah Dunia Dengan Metode Exponential Smoothing*.
- YANG Soyeong. (n.d.). *SEVENTEEN topped the boy group's brand reputation. According to the Korea Corporate Reputation Research Institute*. Retrieved November 11, 2024, from <https://www.mk.co.kr/en/hot-issues/11188308>

LAMPIRAN CODING

```
File Edit Selection View Go Run ... final thesis (Workspace)
Svt.py import statsmodels.py prediksi.py invers.py app.py x
final thesis > app.py > ...
1 import streamlit as st
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from sklearn.linear_model import LinearRegression
5 from sklearn.neighbors import KNeighborsRegressor
6 from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
7
8 # Data
9 data = {
10     'Album': [
11         "An Ode", "Heng:gara", "Semicolon", "Your Choice", "Attacca",
12         "Face The Sun", "FML", "Seventeenth Heaven", "17 Right Here", "Spill The Feels"],
13     'youtube_promotion': [13, 17, 8, 15, 28, 34, 40, 39, 36, 32],
14     'konser_tour': [5, 7, 10, 12, 15, 17, 20, 23, 25, 28],
15     'events': [3, 4, 4, 6, 5, 7, 7, 8, 6, 4],
16     'fanmeeting': [6, 4, 4, 6, 6, 5, 4, 3, 3, 5],
17     'previous_album_sales': [204860, 330494, 347568, 266669, 461665, 590147, 745374, 602312, 696380, 679880],
18     'current_album_sales': [1681971, 1979948, 1285729, 1750597, 2990573, 4574697, 6402296, 5785030, 5056172, 5406787]
19 }
20
21 df = pd.DataFrame(data)
22
23 # Title
24 st.title("SISTEM PREDIKSI PENJUALAN ALBUM SEVENTEEN")
25
26 if 'selected_album' not in st.session_state:
27     st.session_state.selected_album = None
28 if 'previous_search' not in st.session_state:
29     st.session_state.previous_search = ""
30 if 'show_update_page' not in st.session_state:
31     st.session_state.show_update_page = False
32 if 'predictions_lr' not in st.session_state:
```

```
File Edit Selection View Go Run ... final thesis (Workspace)
Svt.py import statsmodels.py prediksi.py invers.py app.py x
final thesis > app.py > ...
37 # Sidebar Menu
38 menu = st.sidebar.selectbox("Menu", ["Dashboard", "Sales Prediction", "Algorithm Comparison"])
39
40 if menu == "Dashboard":
41     st.header("Dashboard")
42
43 # Create a container for the title and search box
44 with st.container():
45     col1, col2 = st.columns([3, 1]) # Adjust the ratio as needed
46     col1.write("") # Empty space for alignment
47     with col2:
48         search_album = st.text_input(label="", placeholder="Search", key="search_album")
49 if search_album != st.session_state.previous_search:
50     st.session_state.selected_album = None # Clear previous selection
51     st.session_state.previous_search = search_album
52
53 # Filter albums based on search input
54 if search_album:
55     filtered_albums = df[df['Album'].str.contains(search_album, case=False)]
56 else:
57     filtered_albums = df
58
59 # Check if there are any filtered albums
60 if not filtered_albums.empty:
61     # Display Albums in a horizontal layout
62     cols = st.columns(len(filtered_albums))
63     for i, album in enumerate(filtered_albums['Album']):
64         if cols[i].button(album):
65             st.session_state.selected_album = album
66             st.rerun()
67 else:
68     st.write("Tidak ada album yang ditemukan.")
```

```

File Edit Selection View Go Run ... final thesis (Workspace)
Svt.py import statsmodels.py prediksi.py invers.py app.py x
final thesis > app.py > ...
70 if st.session_state.selected_album:
71     album_data = df[df['Album'] == st.session_state.selected_album].iloc[0]
72
73     # Pie Chart untuk YouTube, Konser, dan Fanmeeting
74     labels = ['YouTube Promotion', 'Concerts', 'Fanmeetings']
75     sizes = [
76         album_data['youtube_promotion'],
77         album_data['konser_tour'],
78         album_data['fanmeeting']
79     ]
80
81     # Menghitung total untuk proporsi
82     total = sum(sizes)
83
84     # Menghitung persentase
85     sizes_percentage = [size / total * 100 for size in sizes]
86
87     colors = ['#ff7f0e', '#2ca02c', '#d62728'] # Warna untuk masing-masing kategori
88
89     # Create a pie chart
90     plt.figure(figsize=(8, 8))
91     wedges, texts, autotexts = plt.pie(sizes_percentage, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90)
92
93     plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
94     plt.title(f'{st.session_state.selected_album} Chart')
95     st.pyplot(plt)
96

```

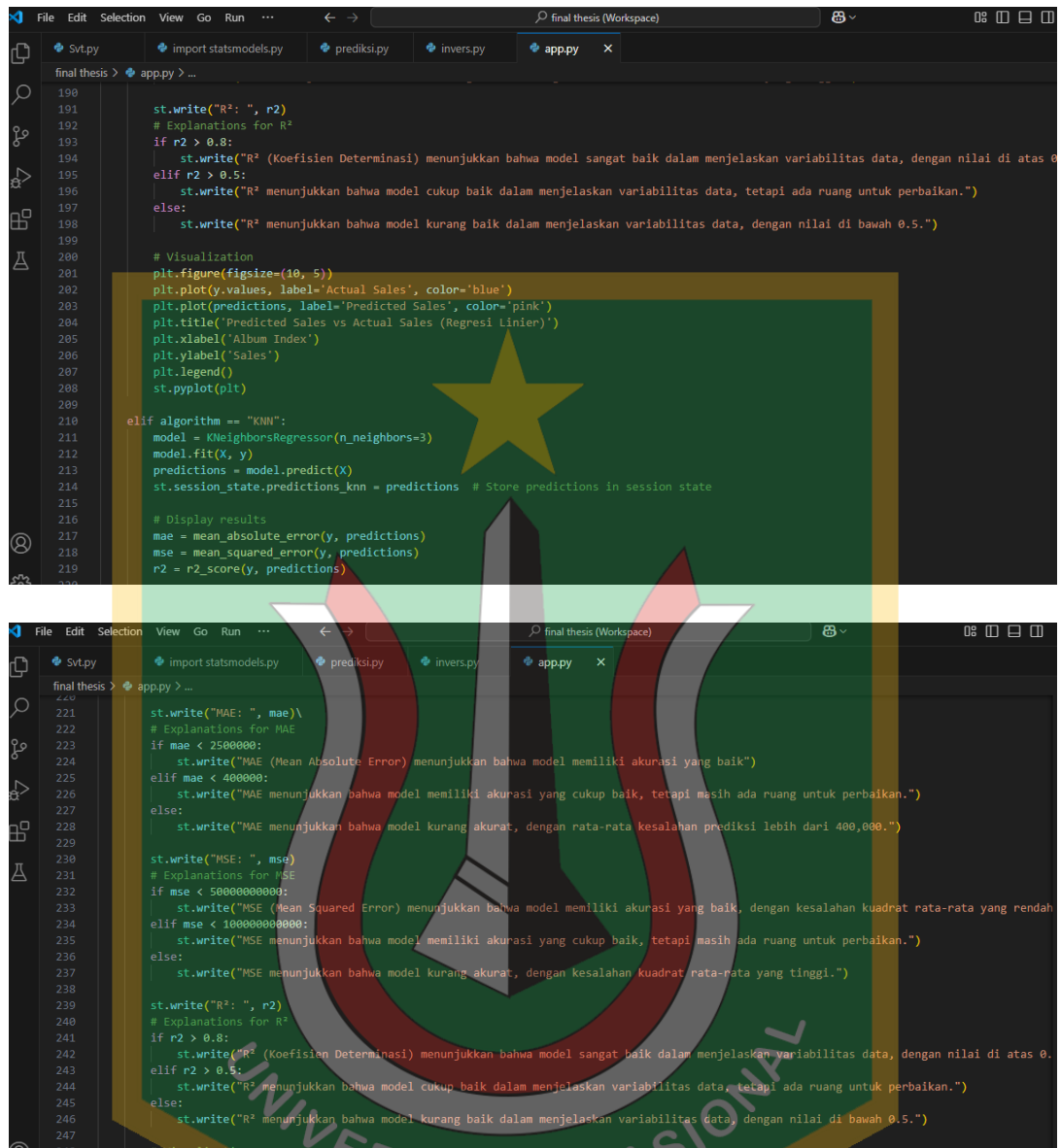
```

File Edit Selection View Go Run ... final thesis (Workspace)
Svt.py import statsmodels.py prediksi.py invers.py app.py x
final thesis > app.py > ...
97 # Sales Graph
98 selected_index = df[df['Album'] == st.session_state.selected_album].index[0]
99
100 # Ambil data untuk album yang dipilih dan album sebelumnya
101 if selected_index > 0: # Pastikan ada album sebelumnya
102     previous_album = df.iloc[selected_index - 1]
103     sales_data = {
104         'Album': [previous_album['Album'], album_data['Album']],
105         'Sales': [previous_album['current_album_sales'], album_data['current_album_sales']]
106     }
107     sales_df = pd.DataFrame(sales_data)
108
109     # Menampilkan grafik garis
110     st.line_chart(sales_df.set_index('Album'))
111
112 else:
113     st.write("Tidak ada album sebelumnya untuk ditampilkan")
114
115 # Update Button
116 if st.button("Update Data"):
117     st.session_state.show_update_page = True
118
119 # Display the update form if the update page is selected
120 if st.session_state.show_update_page:
121     st.header("Update Data")
122
123     # Dropdown for existing albums and option to add a new album
124     album_options = df['Album'].tolist() + ["Tambahkan Album Baru"]
125     selected_album = st.selectbox("Pilih Album", album_options)

```



```
File Edit Selection View Go Run ... final thesis (Workspace)
Svt.py import statsmodels.py prediksi.py invers.py app.py x
final thesis > app.py > ...
127 if selected_album == "Tambahkan Album Baru":
128     new_album_name = st.text_input("Nama Album Baru")
129 else:
130     new_album_name = None
131
132 total_sales = st.number_input("Total Sales", min_value=0)
133 total_youtube = st.number_input("Total YouTube Views", min_value=0)
134 total_konser = st.number_input("Total Konser", min_value=0)
135 total_fanmeeting = st.number_input("Total Fanmeeting", min_value=0)
136
137 if st.button("Update"):
138     if new_album_name:
139         # Add new album to the DataFrame
140         df.loc[len(df)] = [new_album_name, total_youtube, total_konser, 0, total_fanmeeting, 0, total_sales]
141         st.success(f"Album '{new_album_name}' telah ditambahkan!")
142     else:
143         # Update existing album
144         df.loc[df['Album'] == selected_album, ['current_album_sales', 'youtube_promotion', 'konser_tour', 'fanmeeting']] = [total_sales,
145                                                                                                                total_youtube,
146                                                                                                                total_konser,
147                                                                                                                total_fanmeeting]
148         st.success(f"Data untuk album '{selected_album}' telah diperbarui!")
149
150     # Reset the update page
151     st.session_state.show_update_page = False
152     st.rerun()
153
154 elif menu == "Sales Prediction":
155     st.header("Sales Prediction")
156     algorithm = st.selectbox("Pilih Algoritma", ["Regresi Linier", "KNN"])
157
158     # Prepare data for prediction
159     X = df[['youtube_promotion', 'konser_tour', 'events', 'fanmeeting', 'previous_album_sales']]
160     y = df['current_album_sales']
161
162     # Check if an algorithm has been selected
163     if algorithm == "Regresi Linier":
164         model = LinearRegression()
165         model.fit(X, y)
166         predictions = model.predict(X)
167         st.session_state.predictions_lr = predictions # Store predictions in session state
168
169     # Display results
170     st.write("Persamaan Regresi:  $Y = (0.2f) + (0.2f)*X1 + (0.2f)*X2 + (0.2f)*X3 + (0.2f)*X4 + (0.2f)*X5$ ".format(
171         -model.intercept_, *model.coef_))
172     mae = mean_absolute_error(y, predictions)
173     mse = mean_squared_error(y, predictions)
174     r2 = r2_score(y, predictions)
175
176     st.write("MAE: ", mae)
177     # Explanations for MAE
178     if mae < 2500000:
179         st.write("MAE (Mean Absolute Error) menunjukkan bahwa model memiliki akurasi yang baik")
180     elif mae < 4000000:
181         st.write("MAE menunjukkan bahwa model memiliki akurasi yang cukup baik, tetapi masih ada ruang untuk perbaikan.")
182     else:
183         st.write("MAE menunjukkan bahwa model kurang akurat, dengan rata-rata kesalahan prediksi lebih dari 400,000.")
184
185     st.write("MSE: ", mse)
186     # Explanations for MSE
187     if mse < 500000000000:
188         st.write("MSE (Mean Squared Error) menunjukkan bahwa model memiliki akurasi yang baik, dengan kesalahan kuadrat rata-rata yang rendah")
189     elif mse < 1000000000000:
190         st.write("MSE menunjukkan bahwa model memiliki akurasi yang cukup baik, tetapi masih ada ruang untuk perbaikan.")
191     else:
192         st.write("MSE menunjukkan bahwa model kurang akurat, dengan kesalahan kuadrat rata-rata yang tinggi.")
```



```
190 st.write("R²: ", r2)
191 # Explanations for R²
192 if r2 > 0.8:
193     st.write("R² (Koefisien Determinasi) menunjukkan bahwa model sangat baik dalam menjelaskan variabilitas data, dengan nilai di atas 0.8")
194 elif r2 > 0.5:
195     st.write("R² menunjukkan bahwa model cukup baik dalam menjelaskan variabilitas data, tetapi ada ruang untuk perbaikan.")
196 else:
197     st.write("R² menunjukkan bahwa model kurang baik dalam menjelaskan variabilitas data, dengan nilai di bawah 0.5.")
198
199 # Visualization
200 plt.figure(figsize=(10, 5))
201 plt.plot(y.values, label='Actual Sales', color='blue')
202 plt.plot(predictions, label='Predicted Sales', color='pink')
203 plt.title('Predicted Sales vs Actual Sales (Regresi Linier)')
204 plt.xlabel('Album Index')
205 plt.ylabel('Sales')
206 plt.legend()
207 st.pyplot(plt)
208
209 elif algorithm == "KNN":
210     model = KNeighborsRegressor(n_neighbors=3)
211     model.fit(X, y)
212     predictions = model.predict(X)
213     st.session_state.predictions_knn = predictions # Store predictions in session state
214
215 # Display results
216 mae = mean_absolute_error(y, predictions)
217 mse = mean_squared_error(y, predictions)
218 r2 = r2_score(y, predictions)
219
220
221 st.write("MAE: ", mae)\
222 # Explanations for MAE
223 if mae < 2500000:
224     st.write("MAE (Mean Absolute Error) menunjukkan bahwa model memiliki akurasi yang baik")
225 elif mae < 4000000:
226     st.write("MAE menunjukkan bahwa model memiliki akurasi yang cukup baik, tetapi masih ada ruang untuk perbaikan.")
227 else:
228     st.write("MAE menunjukkan bahwa model kurang akurat, dengan rata-rata kesalahan prediksi lebih dari 400,000.")
229
230 st.write("MSE: ", mse)
231 # Explanations for MSE
232 if mse < 500000000000:
233     st.write("MSE (Mean Squared Error) menunjukkan bahwa model memiliki akurasi yang baik, dengan kesalahan kuadrat rata-rata yang rendah")
234 elif mse < 1000000000000:
235     st.write("MSE menunjukkan bahwa model memiliki akurasi yang cukup baik, tetapi masih ada ruang untuk perbaikan.")
236 else:
237     st.write("MSE menunjukkan bahwa model kurang akurat, dengan kesalahan kuadrat rata-rata yang tinggi.")
238
239 st.write("R²: ", r2)
240 # Explanations for R²
241 if r2 > 0.8:
242     st.write("R² (Koefisien Determinasi) menunjukkan bahwa model sangat baik dalam menjelaskan variabilitas data, dengan nilai di atas 0.8")
243 elif r2 > 0.5:
244     st.write("R² menunjukkan bahwa model cukup baik dalam menjelaskan variabilitas data, tetapi ada ruang untuk perbaikan.")
245 else:
246     st.write("R² menunjukkan bahwa model kurang baik dalam menjelaskan variabilitas data, dengan nilai di bawah 0.5.")
247
248 # Visualization
```

```

File Edit Selection View Go Run ... final thesis (Workspace)
Svt.py import statsmodels.py prediksi.py invers.py app.py x
final thesis > app.py > ...
248 # Visualization
249 plt.figure(figsize=(10, 5))
250 plt.plot(y.values, label='Actual Sales', color='blue')
251 plt.plot(predictions, label='Predicted Sales', color='orange')
252 plt.title('Predicted Sales vs Actual Sales (KNN)')
253 plt.xlabel('Album Index')
254 plt.ylabel('Sales')
255 plt.legend()
256 st.pyplot(plt)
257
258 elif menu == "Algorithm Comparison":
259     st.header("Algorithm Comparison")
260     # Prepare data for prediction
261     X = df[['youtube_promotion', 'konser_tour', 'events', 'fanmeeting', 'previous_album_sales']]
262     y = df['current_album_sales']
263
264     X = df[['youtube_promotion', 'konser_tour', 'events', 'fanmeeting', 'previous_album_sales']]
265     y = df['current_album_sales']
266
267     # Check if predictions exist before plotting
268     if st.session_state.predictions_lr is not None and st.session_state.predictions_knn is not None:
269         plt.figure(figsize=(10, 5))
270         plt.plot(y.values, label='Actual Sales', color='blue')
271         plt.plot(st.session_state.predictions_lr, label='Linear Regression Predicted Sales', color='orange')
272         plt.plot(st.session_state.predictions_knn, label='KNN Predicted Sales', color='green')
273         plt.title('Comparison of Predicted Sales')
274         plt.xlabel('Album Index')
275         plt.ylabel('Sales')
276         plt.legend()
277         st.pyplot(plt)

```

```

File Edit Selection View Go Run ... final thesis (Workspace)
Svt.py import statsmodels.py prediksi.py invers.py app.py
final thesis > app.py > ...
279 # calculate metrics for both models
280 mae_lr = mean_absolute_error(y, st.session_state.predictions_lr)
281 mae_knn = mean_absolute_error(y, st.session_state.predictions_knn)
282 r2_lr = r2_score(y, st.session_state.predictions_lr)
283 r2_knn = r2_score(y, st.session_state.predictions_knn)
284
285 # Provide explanations based on metrics
286 st.write("Analisis dan Interpretasi:")
287
288 if mae_lr < mae_knn:
289     st.write("Model Regresi Linier memiliki MAE yang lebih rendah dibandingkan KNN, menunjukkan bahwa Regresi Linier lebih akurat dalam m")
290 else:
291     st.write("Model KNN memiliki MAE yang lebih rendah dibandingkan Regresi Linier, menunjukkan bahwa KNN lebih akurat dalam memprediksi")
292
293 if r2_lr > r2_knn:
294     st.write("Model Regresi Linier memiliki R² yang lebih tinggi, menunjukkan bahwa model ini lebih baik dalam menjelaskan variabilitas d")
295 else:
296     st.write("Model KNN memiliki R² yang lebih tinggi, menunjukkan bahwa model ini lebih baik dalam menjelaskan variabilitas data penjual")
297
298 st.write("Dari grafik di atas, dapat dilihat perbandingan antara hasil prediksi dari kedua algoritma. Akurasi dari algoritma dapat diliha")
299
300 else:
301     st.write("Predictions are not available. Please run the Sales Prediction first.")

```

