

LAPORAN PRAKTIKUM
MACHINE LEARNING BERBASIS WEB



Oleh :

Ilham Gading Pangestu

NIM. 233307102

Dosen Pengampu :
Ardian Prima Atmaja, S.Kom., M.Cs.

POLITEKNIK NEGERI MADIUN
TAHUN AKADEMIK 2024/ 2025

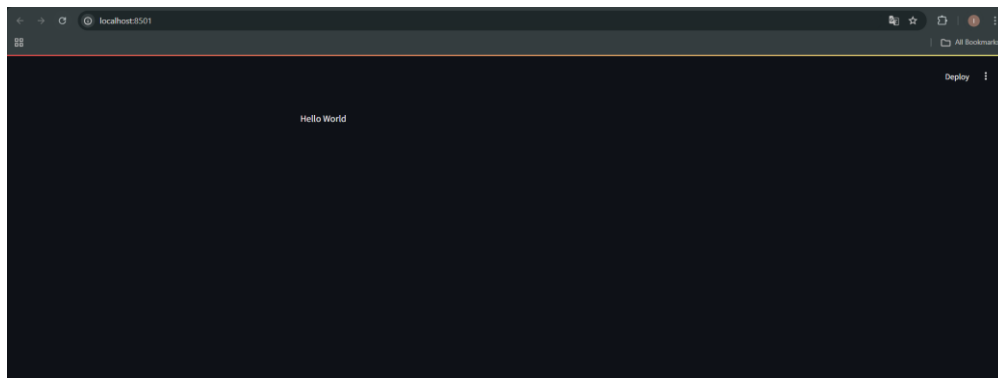
MACHINE LEARNING BERBASIS WEB PYTHON

A.

1. Mengimpor library streamlit terlebih dahulu dan jalankan kode berikut di terminal dengan cara: `streamlit run nama_file.py`. Maka browser akan terbuka pada pada alamat `http://localhost:8501` (no port bergantung komputer).

```
machinelearning.py
1  import streamlit as st
2
3  st.write("Hello World")
4
```

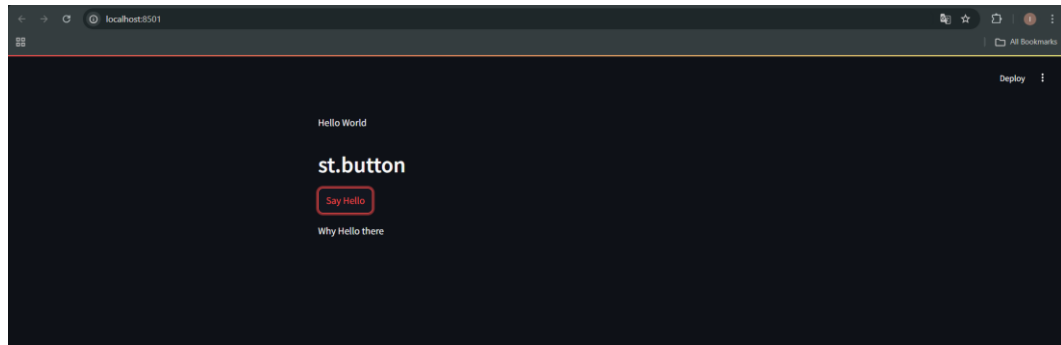
Hasil:



2. Membuat aksi dari button.

```
machinelearning.py
1  import streamlit as st
2
3  st.write("Hello World")
4
5  st.header("st.button")
6
7  if st.button("Say Hello"):
8      st.write("Why Hello there")
9  else:
10     st.write("Goodbye")
```

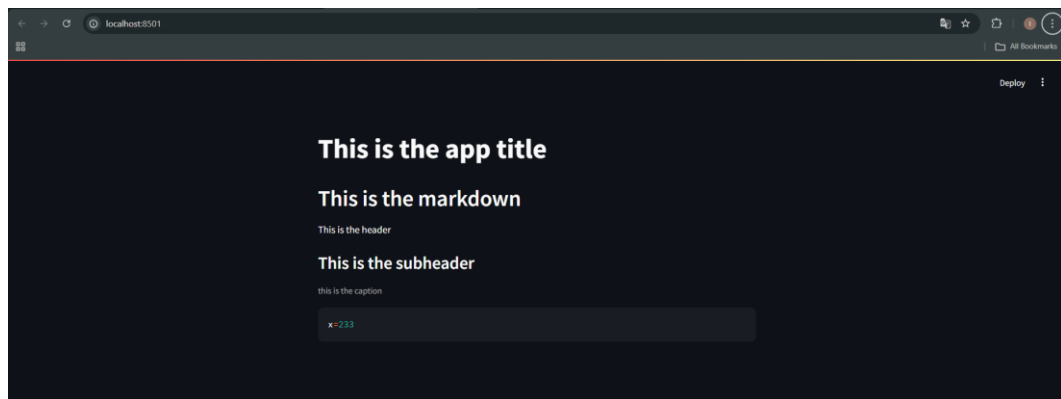
Hasil:



3. Susunlah kode untuk hasil form di bawah ini.

```
12  st.title("This is the app title")
13  st.header("This is the markdown")
14  st.markdown("This is the header")
15  st.subheader("This is the subheader")
16  st.caption("this is the caption")
17
18  st.code("x=233")
```

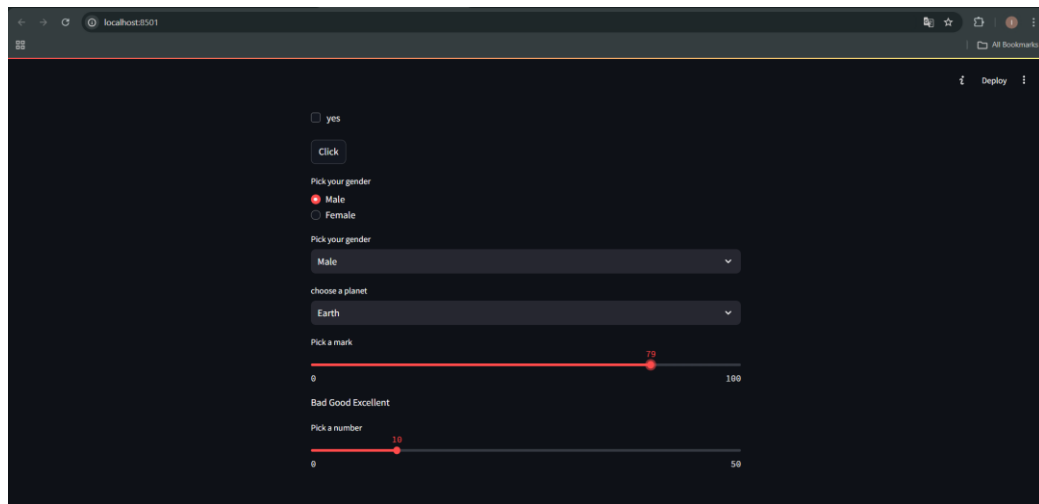
Hasil:



4. Susunlah kode untuk hasil form di bawah ini.

```
26  st.checkbox("yes")
27  st.button("Click")
28  gender = st.radio("Pick your gender", options=["Male", "Female"])
29  gender_select = st.selectbox("Pick your gender", options=["Male", "Female"])
30  planet = st.selectbox("choose a planet", options=["Mercury", "Venus", "Earth",
31  "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"])
32  mark = st.slider("Pick a mark", min_value=0, max_value=100, value=50)
33  st.write("Bad", " " * 50, "Good", " " * 50, "Excellent")
34  number = st.slider("Pick a number", min_value=0, max_value=50, value=10)
```

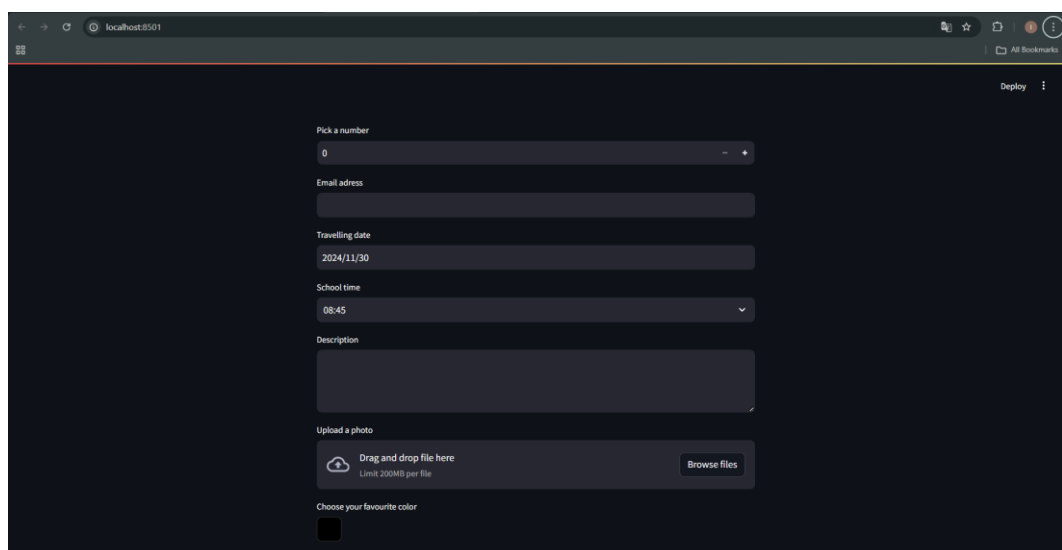
Hasil:



5. Susunlah kode untuk hasil form di bawah ini.

```
35 st.number_input('Pick a number', 0, 10)
36 st.text_input('Email adress')
37 st.date_input('Travelling date')
38 st.time_input('School time')
39 st.text_area('Description')
40 st.file_uploader('Upload a photo')
41 st.color_picker('Choose your favourite color')
```

Hasil:



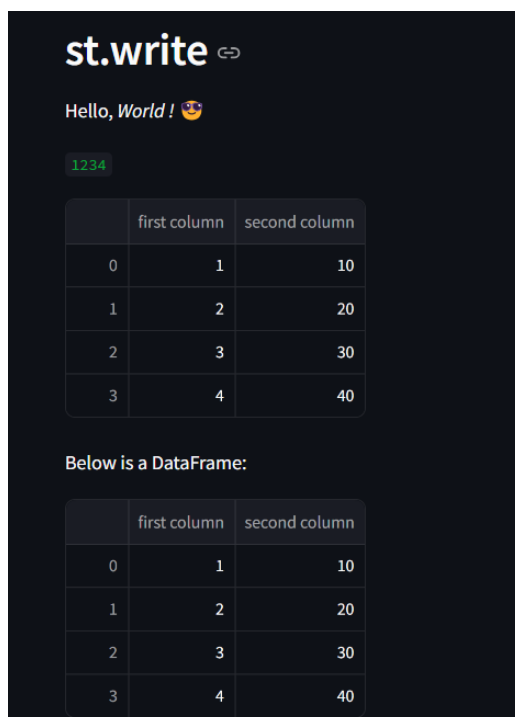
6. Memuat sebuah Dataframe yang berisi data numerik, dan membuat visualisasi data numerik random ke dalam sebuah plot chart.

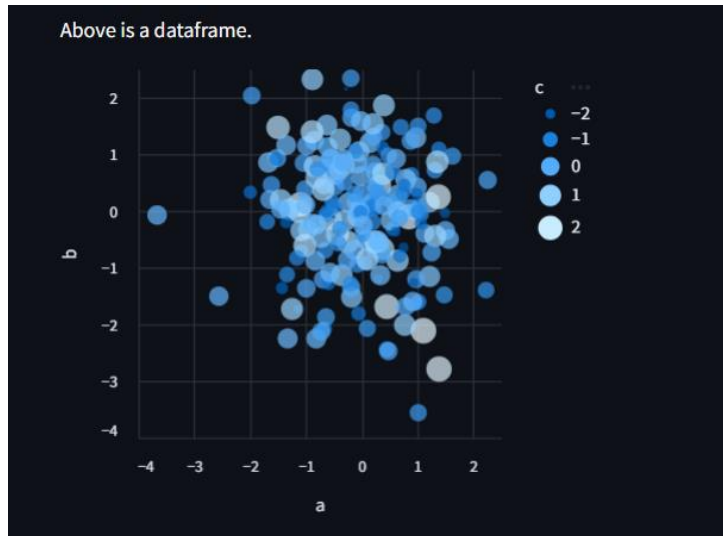
```
44 import numpy as np
45 import altair as alt
46 import pandas as pd
47 import streamlit as st
```

```
49 st.header('st.write')
50 st.write('Hello, *World !* :sunglasses:')
51 st.write(1234)
52
53 df = pd.DataFrame({
54     'first column': [1, 2, 3, 4],
55     'second column': [10, 20, 30, 40]
56 })
57
```

```
58 st.write(df)
59
60 st.write('Below is a DataFrame:', df, 'Above is a dataframe.')
61
62 df2 = pd.DataFrame(
63     np.random.randn(200, 3),
64     columns=['a', 'b', 'c'])
65 c = alt.Chart(df2).mark_circle().encode(
66     x='a', y='b', size='c', color='c', tooltip=['a', 'b', 'c'])
67 st.write(c)
```

Hasil:





7. Buatlah line chart sesuai dengan kode berikut ini. Lalu tambahkan bar chart dan area chart.

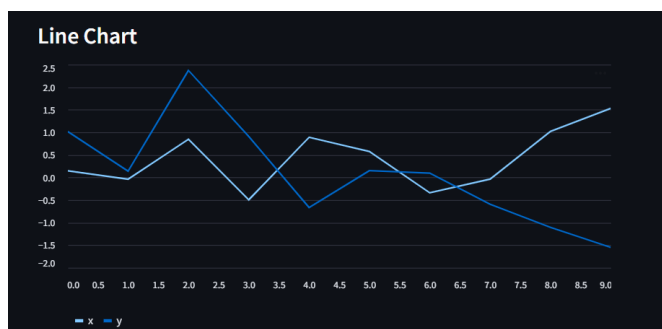
```
70 import streamlit as st
71 import pandas as pd
72 import numpy as np
73
74 df= pd.DataFrame(
75     np.random.randn(10, 2),
76     columns=['x', 'y'])
```

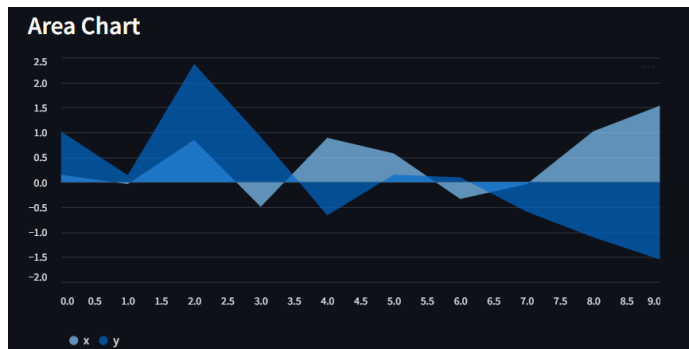
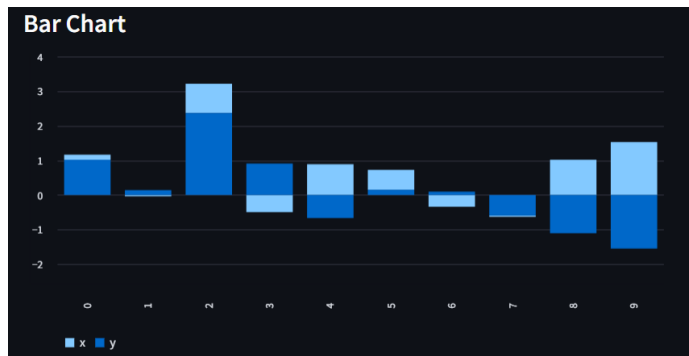
```
78 # Line Chart
79 st.write("### Line Chart")
80 st.line_chart(df)
```

```
82 # Bar Chart
83 st.write("### Bar Chart")
84 st.bar_chart(df)
```

```
86 # Area Chart
87 st.write("### Area Chart")
88 st.area_chart(df)
```

Hasil:





8. Buatlah sebuah web untuk menampilkan image, dataset (csv), dan grafik sesuai menu selectbox yang dipilih pada left sidebar seperti pada gambar di bawah ini. Sumber image dan dataset yang digunakan bebas.

```

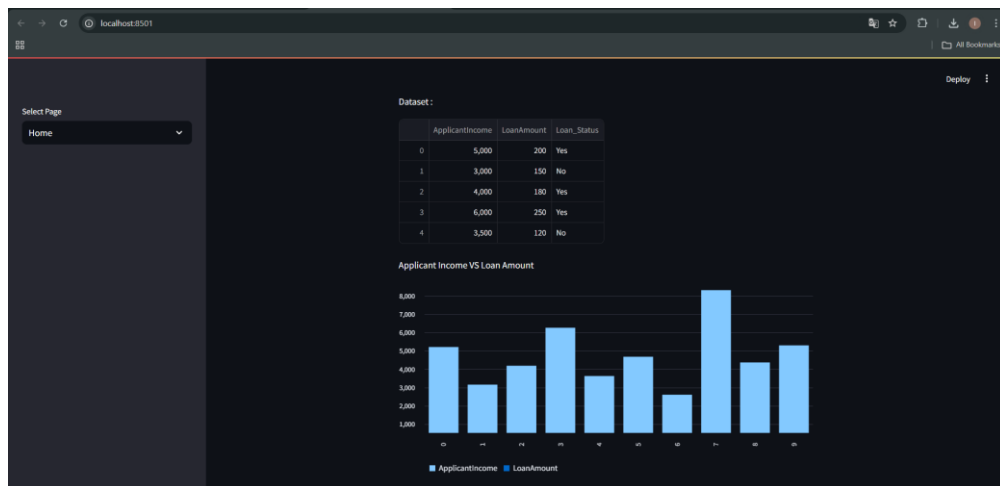
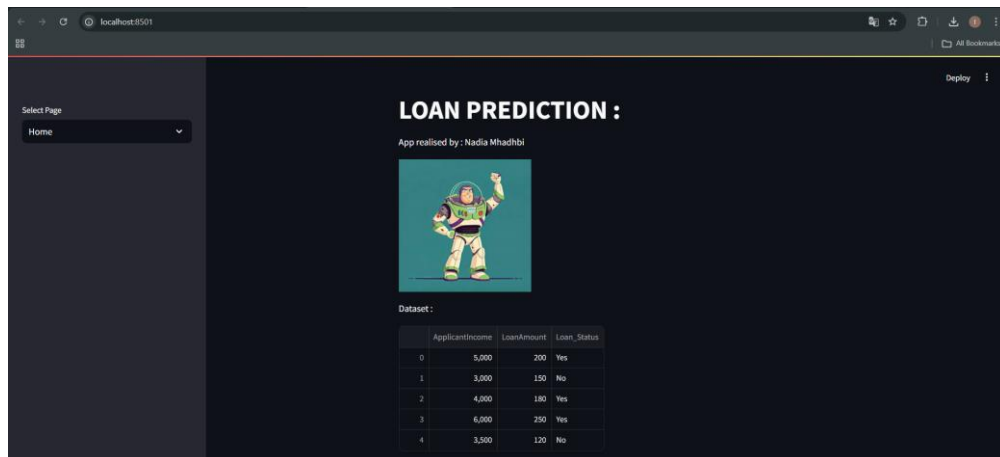
91 import streamlit as st
92 import pandas as pd
93 import numpy as np
94 import pickle
95 import base64

97 @st.cache(suppress_st_warning=True)
98 def get_fvalue(val):
99     feature_dict = {"No":1,"Yes":2}
100     for key, value in feature_dict.items():
101         if val == key:
102             return value
103
104 def get_value(val,my_dict):
105     for key,value in my_dict.items():
106         if val == key:
107             return value

109 app_mode = st.sidebar.selectbox('Select Page', ['Home','Prediction'])
110 if app_mode == 'Home':
111     st.title('LOAN PREDICTION :')
112     st.write('App realised by : Nadia Mhadhbi')
113     st.image('loan_image.jpeg')
114     st.markdown('Dataset :')
115     data=pd.read_csv('loan_dataset.csv')
116     st.write(data.head())
117     st.markdown('Applicant Income VS Loan Amount ')
118     st.bar_chart(data[['ApplicantIncome','LoanAmount']].head(20))

```

Hasil:



B.

9. Pada praktikum ke-12 sebelumnya, tambahkan sebuah kode untuk men-generate sebuah file .sav yang digunakan untuk menyimpan model machine learning yang telah selesai kita train dan test tersebut. File ini selanjutnya akan digunakan untuk mengimplementasikan sebuah model machine learning dalam sebuah website.

```
122 import pickle
123 filename = "model_prediksi_harga_mobil.sav"
124 pickle.dump(model_regresi, open(filename, "wb"))
```

10. Lengkapi kode berikut ini untuk mengimplementasikan soal praktikum pada pertemuan ke-12 sebelumnya menjadi sebuah aplikasi web berbasis Streamlit untuk memprediksi harga mobil. Kode ini akan memanggil file .sav yang baru

saja kita generate. Contoh hasil web dapat dilihat pada <https://hargamobil.streamlit.app>.

```
127 import pickle
128 import streamlit as st
129 import pandas as pd
130 import numpy as np
131 import altair as alt
132
133 # Load model yang sudah disimpan
134 model = pickle.load(open('model_prediksi_harga_mobil.sav', 'rb'))
135
136 # Judul aplikasi
137 st.title('Prediksi Harga Mobil')
138
139 # Bagian 1: Menampilkan Dataset
140 st.header("Dataset")
141 # Membuka file CSV
142 df1 = pd.read_csv('CarPrice.csv')
143 st.dataframe(df1)
144
145 # Reset index untuk grafik jika diperlukan
146 df1.reset_index(inplace=True)
```

```
148 # Bagian 2: Menampilkan Grafik
149 st.write("Grafik Highway-mpg")
150 chart_highwaympg = alt.Chart(df1).mark_line().encode(
151     x='index:Q',
152     y='highwaympg:Q'
153 )
154 st.altair_chart(chart_highwaympg, use_container_width=True)
155
156 st.write("Grafik Curbweight")
157 chart_curbweight = alt.Chart(df1).mark_line().encode(
158     x='index:Q',
159     y='curbweight:Q'
160 )
161 st.altair_chart(chart_curbweight, use_container_width=True)
162
163 st.write("Grafik Horsepower")
164 chart_horsepower = alt.Chart(df1).mark_line().encode(
165     x='index:Q',
166     y='horsepower:Q'
167 )
168 st.altair_chart(chart_horsepower, use_container_width=True)
169
170 # Bagian 3: Input untuk Prediksi
171 st.header("Input Nilai untuk Prediksi")
172 # Input variabel independent
173 highwaympg = st.number_input("Highway-mpg", min_value=0, max_value=100, step=1, value=30)
174 curbweight = st.number_input("Curbweight", min_value=0, max_value=5000, step=10, value=2000)
175 horsepower = st.number_input("Horsepower", min_value=0, max_value=1000, step=10, value=150)
```

```
177 # Bagian 4: Tombol Prediksi
178 if st.button('Prediksi'):
179     try:
180         # Prediksi variabel yang telah diinputkan
181         car_prediction = model.predict([[highwaympg, curbweight, horsepower]])
182
183         # Konversi hasil prediksi ke format string
184         harga_mobil_float = float(car_prediction[0])
185         harga_mobil_formatted = f"Rp {harga_mobil_float:,.2f}" # Format dalam bentuk mata uang
186
187         # Tampilkan hasil prediksi
188         st.success(f"Harga Mobil yang Diprediksi: {harga_mobil_formatted}")
189     except Exception as e:
190         st.error(f"Terjadi kesalahan saat melakukan prediksi: {e}")
```

Hasil:



11. Lakukan pengembangan aplikasi web machine learning tersebut sesuai kreativitas masing-masing dengan memanfaatkan komponen-komponen Streamlit.

```

194 import pickle
195 import streamlit as st
196 import pandas as pd
197 import numpy as np
198 import altair as alt
199 import base64

```

```

226 # Dataset Page
227 if menu == "Dataset":
228     st.title("Dataset Harga Mobil")
229     # Load dataset
230     df1 = pd.read_csv('CarPrice.csv')
231
232     # Opsi untuk memilih kolom yang ditampilkan
233     selected_columns = st.multiselect("Pilih Kolom untuk Ditampilkan", df1.columns.tolist(), default=df1.columns.tolist())
234     st.dataframe(df1[selected_columns])
235
236     # Filter dataset
237     st.subheader("Filter Dataset")
238     min_price = st.slider("Minimum Price", int(df1["price"].min()), int(df1["price"].max()), int(df1["price"].min()))
239     filtered_data = df1[df1["price"] >= min_price]
240     st.write(f"Menampilkan data dengan harga di atas Rp {min_price:,}")
241     st.dataframe(filtered_data)

```

```

201 # Load model yang sudah disimpan
202 model = pickle.load(open('model_prediksi_harga_mobil.sav', 'rb'))
203
204 # Sidebar untuk navigasi
205 st.sidebar.title("Menu")
206 menu = st.sidebar.radio("Pilih Halaman", ["Home", "Dataset", "Visualisasi", "Prediksi"])
207
208 # Home Page
209 if menu == "Home":
210     st.title("Aplikasi Prediksi Harga Mobil")
211     st.image("mobil_example.jpg", caption="Contoh Mobil", use_column_width=True)
212     st.markdown("""
213     ### Tentang Aplikasi
214     Aplikasi ini menggunakan model Machine Learning untuk memprediksi harga mobil berdasarkan beberapa fitur penting:
215     - **Highway-mpg**: Efisiensi bahan bakar di jalan raya.
216     - **Curbweight**: Berat kendaraan.
217     - **Horsepower**: Tenaga mesin kendaraan.
218
219     ### Cara Menggunakan
220     1. Navigasikan ke menu **Dataset** untuk melihat data.
221     2. Lihat grafik di menu **Visualisasi**.
222     3. Gunakan menu **Prediksi** untuk melakukan prediksi harga mobil.
223     """)
224     st.success("Aplikasi siap digunakan!")

```

```

226 # Dataset Page
227 if menu == "Dataset":
228     st.title("Dataset Harga Mobil")
229     # Load dataset
230     df1 = pd.read_csv('CarPrice.csv')
231
232     # Opsi untuk memilih kolom yang ditampilkan
233     selected_columns = st.multiselect("Pilih Kolom untuk Ditampilkan", df1.columns.tolist(), default=df1.columns.tolist())
234     st.dataframe(df1[selected_columns])
235
236     # Filter dataset
237     st.subheader("Filter Dataset")
238     min_price = st.slider("Minimum Price", int(df1["price"].min()), int(df1["price"].max()), int(df1["price"].min()))
239     filtered_data = df1[df1["price"] >= min_price]
240     st.write(f"Menampilkan data dengan harga di atas Rp {min_price:,}")
241     st.dataframe(filtered_data)

```

```

243 # Visualisasi Page
244 if menu == "Visualisasi":
245     st.title("Visualisasi Data")
246     df1 = pd.read_csv('CarPrice.csv')
247
248     # Pilihan grafik
249     st.write("Pilih jenis grafik untuk ditampilkan:")
250     chart_type = st.radio(
251         "Pilih Grafik",
252         ["Line Chart", "Bar Chart", "Scatter Plot"]
253     )
254
255     if chart_type == "Line Chart":
256         st.line_chart(df1[['highwaympg', 'curbweight', 'horsepower']])
257     elif chart_type == "Bar Chart":
258         st.bar_chart(df1[['highwaympg', 'curbweight', 'horsepower']])
259     elif chart_type == "Scatter Plot":
260         scatter_chart = alt.Chart(df1).mark_circle(size=60).encode(
261             x='horsepower:Q',
262             y='price:Q',
263             color='fueltype:N',
264             tooltip=['horsepower', 'price', 'fueltype']
265         ).interactive()
266         st.altair_chart(scatter_chart, use_container_width=True)
267

```

```

268 # Prediksi Page
269 if menu == "Prediksi":
270     st.title("Prediksi Harga Mobil")
271     st.write("Masukkan nilai untuk fitur berikut:")
272
273     # Input untuk prediksi
274     highwaympg = st.slider("Highway-mpg", min_value=0, max_value=100, value=30)
275     curbweight = st.slider("Curbweight", min_value=0, max_value=5000, value=2000)
276     horsepower = st.slider("Horsepower", min_value=0, max_value=1000, value=150)
277

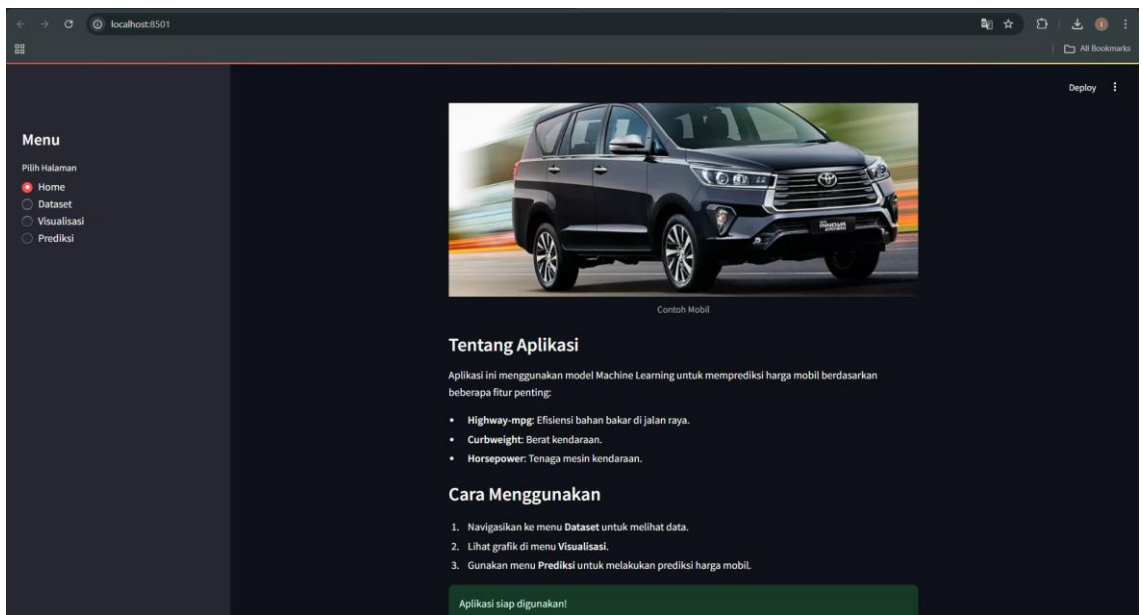
```

```

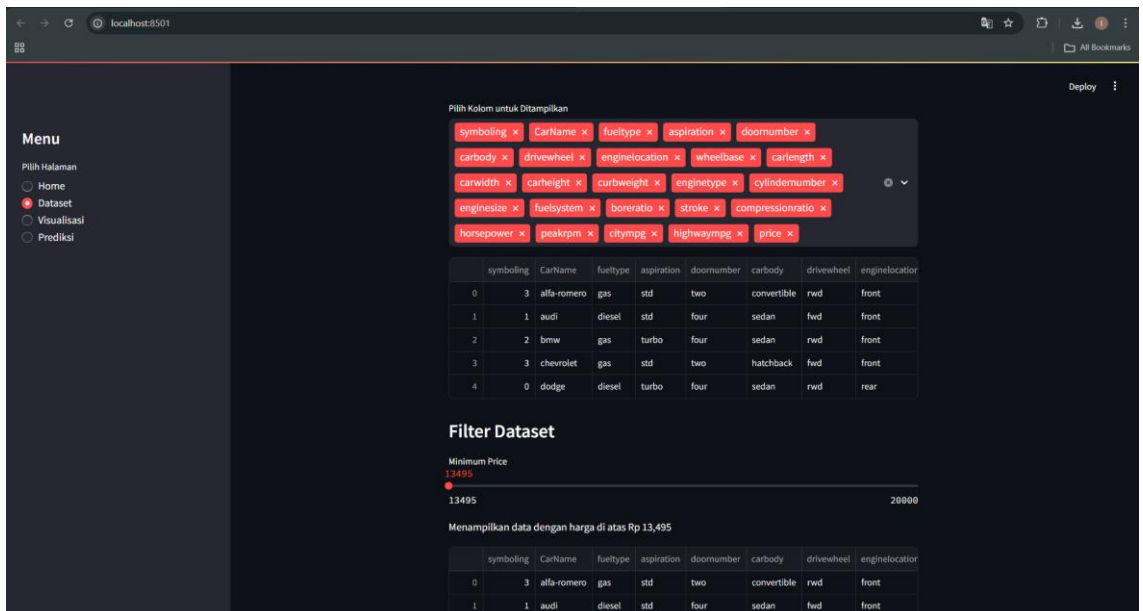
278 # Prediksi berdasarkan input
279 if st.button("Prediksi"):
280     car_prediction = model.predict([[highwaympg, curbweight, horsepower]])
281     harga_mobil_float = float(car_prediction[0])
282     harga_mobil_formatted = f"Rp {harga_mobil_float:,.2f}" # Format mata uang
283     st.success(f"Harga Mobil yang Diprediksi: {harga_mobil_formatted}")
284
285 # Opsi untuk mengunduh hasil prediksi
286 hasil_prediksi = pd.DataFrame({
287     "Highway-mpg": [highwaympg],
288     "Curbweight": [curbweight],
289     "Horsepower": [horsepower],
290     "Predicted Price": [harga_mobil_float]
291 })
292 csv = hasil_prediksi.to_csv(index=False)
293 b64 = base64.b64encode(csv.encode()).decode() # encode to base64
294 href = f"<a href='data:file/csv;base64,{b64}' download='hasil_prediksi.csv'>Klik di sini untuk mengunduh hasil prediksi</a>"
295 st.markdown(href, unsafe_allow_html=True)

```

Hasil:

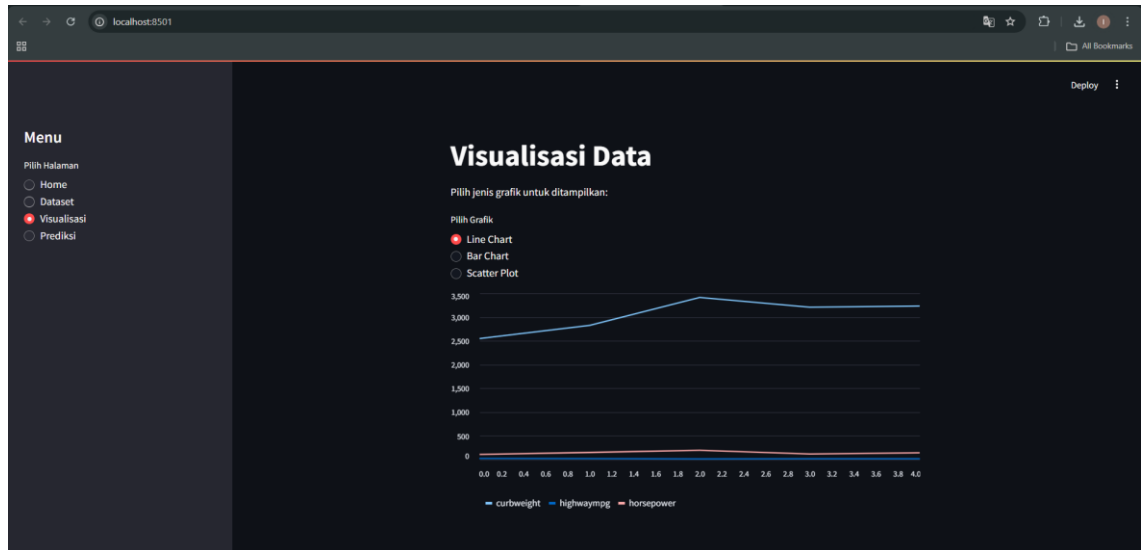


The screenshot shows a web application running on localhost:8501. On the left is a sidebar menu with options: Home (selected), Dataset, Visualisasi, and Prediksi. The main content area features a large image of a dark-colored Toyota Innova. Below the image is a section titled "Tentang Aplikasi" (About the Application) which states that the app uses a Machine Learning model to predict car prices based on key features. It lists three features: Highway-mpg (fuel efficiency), Curbweight (vehicle weight), and Horsepower (engine power). Below this is a section titled "Cara Menggunakan" (How to Use) with three steps: 1. Navigate to the Dataset menu to view data, 2. Look at the graph in the Visualisasi menu, and 3. Use the Prediksi menu to perform car price predictions. At the bottom of the main content area, a green box says "Aplikasi siap digunakan!" (Application ready for use!).



The screenshot shows the same web application with the "Dataset" menu item selected in the sidebar. The main content area displays a section titled "Pilih Kolom untuk Ditampilkan" (Select columns to display) with a grid of buttons for various car features: symboling, CarName, fueltype, aspiration, doornumber, carbody, drivewheel, enginelocation, carwidth, carheight, curbweight, enginetype, cylindernumber, enginesize, fuelsystem, boreratio, stroke, compressionratio, horsepower, peakrpm, citympg, highwaympg, and price. Below this grid is a table with 8 columns (symboling, CarName, fueltype, aspiration, doornumber, carbody, drivewheel, enginelocation) and 5 rows of car data. Below the table is a section titled "Filter Dataset" showing a "Minimum Price" filter set to 13495, with a range from 13495 to 28000. It states "Menampilkan data dengan harga di atas Rp 13,495" (Displaying data with price above Rp 13,495). Below this is a filtered table showing only the first two rows of the original data.

symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation
0	3 alfa-romero	gas	std	two	convertible	rwd	front
1	1 audi	diesel	std	four	sedan	fwd	front



Prediksi Harga Mobil

Masukkan nilai untuk fitur berikut:

Highway-mpg: 39

Curbweight: 3622

Horsepower: 277

Prediksi

Harga Mobil yang Diprediksi: Rp 51,949.38

[Klik di sini untuk mengunduh hasil prediksi](#)

HOTEL PRICE PREDICT

A. Import Library

```
1 # Import library yang diperlukan
2 import streamlit as st
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 from sklearn.model_selection import train_test_split
7 from sklearn.ensemble import RandomForestRegressor
8 from sklearn.metrics import mean_absolute_error
```

Fungsi dari baris kode ini adalah untuk mengimpor pustaka atau library yang diperlukan untuk aplikasi. Library ini mencakup alat untuk pengolahan data, pembuatan grafik, dan machine learning.

- Streamlit (st): Digunakan untuk membangun aplikasi web interaktif.
- Pandas (pd): Untuk mengolah dan memanipulasi data dalam bentuk DataFrame.
- Matplotlib (plt) dan Seaborn (sns): Untuk visualisasi data dalam bentuk grafik.
- Scikit-learn: Untuk model machine learning, termasuk pembagian data (train_test_split), algoritma regresi (RandomForestRegressor), dan metrik evaluasi (mean_absolute_error).

B. Konfigurasi Tampilan Halaman

```
10 # Konfigurasi tampilan halaman
11 st.set_page_config(layout="wide", page_title="Prediksi Harga Hotel")
```

Baris kode ini mengatur pengaturan tampilan halaman aplikasi Streamlit, seperti judul halaman dan layout.

- layout="wide": Mengatur layout halaman menjadi lebih lebar agar konten bisa lebih tersebar dan nyaman dilihat.
- page_title="Prediksi Harga Hotel": Menetapkan judul halaman yang ditampilkan pada tab browser.

C. Fungsi untuk Memuat dan Membersihkan Data

```
13 # Fungsi untuk memuat dan membersihkan data
14 def muat_data():
15     data = pd.read_csv("booking_hotel.csv", encoding="latin1")
16     data.columns = data.columns.str.strip()
17     data['Room Price'] = data['Room Price (in BDT or any other currency)'].str.replace("[^\\d]", "", regex=True).astype(float)
18     return data.dropna(subset=['Room Price'])
```

Fungsi muat_data() digunakan untuk membaca dataset dan membersihkan kolom tertentu.

- pd.read_csv("booking_hotel.csv", encoding="latin1"): Membaca file CSV yang berisi data hotel.
- data.columns.str.strip(): Menghapus spasi yang tidak diperlukan dari nama kolom.
- data['Room Price'].str.replace("[^\\d]", "", regex=True).astype(float): Menghapus karakter selain angka dalam kolom "Room Price" dan mengonversinya menjadi angka.
- data.dropna(subset=['Room Price']): Menghapus baris yang memiliki nilai kosong pada kolom harga kamar.

D. Memuat Dataset

```
20 # Memuat dataset
21 dataset = muat_data()
```

Baris kode ini memanggil fungsi muat_data() dan menyimpan hasilnya dalam variabel dataset.

- Dataset yang telah dibersihkan kemudian digunakan untuk analisis, visualisasi, dan pelatihan model.

E. Konfigurasi Sidebar

```

34 # Informasi di sidebar
35 st.sidebar.markdown("---")
36 st.sidebar.markdown("### 📄 Informasi Aplikasi")
37 st.sidebar.info("Aplikasi ini menggunakan machine learning untuk memprediksi harga kamar hotel.")
38 st.sidebar.markdown("### 📊 Statistik Data")
39 st.sidebar.metric("Total Data", f"{len(dataset):,} baris")

```

Bagian ini mengatur sidebar aplikasi untuk menampilkan logo, menu navigasi, dan informasi tentang aplikasi.

- `st.sidebar.image("image.jpg", width=200)`: Menampilkan gambar logo di sidebar.
- `st.sidebar.selectbox()`: Menyediakan menu dropdown untuk memilih halaman mana yang ingin ditampilkan.
- `st.sidebar.info()` dan `st.sidebar.metric()`: Menyediakan informasi tambahan seperti deskripsi aplikasi dan statistik jumlah data.

F. Halaman Beranda

```

41 # Fungsi untuk halaman beranda
42 def beranda():
43     st.title("🏠 Aplikasi Prediksi Harga Hotel")
44     st.markdown("---")
45
46     col1, col2 = st.columns([2,1])
47     with col1:
48         st.markdown("""
49             ## Selamat Datang di Aplikasi Prediksi Harga Hotel!
50
51             Aplikasi ini membantu memperkirakan harga kamar hotel berdasarkan:
52             - 📍 Lokasi Hotel
53             - 🛏 Jenis Kamar
54             - 🛏 Tipe Tempat Tidur
55             """)
56
57         st.info("""
58             ## 🌟 Fitur Utama:
59             1. **Dataset**: Melihat data hotel yang tersedia
60             2. **Visualisasi**: Analisis visual data harga hotel
61             3. **Prediksi**: Prediksi harga berdasarkan preferensi
62             """)
63     with col2:
64         with col2:
65             st.image("image.jpg", caption="Prediksi Harga Hotel", use_container_width=True)

```

Menampilkan informasi dasar dan pengantar aplikasi di halaman beranda.

- `st.title()`: Menampilkan judul halaman beranda.
- `st.markdown()`: Menampilkan teks penjelasan aplikasi.
- `st.columns([2,1])`: Membagi tampilan halaman menjadi dua kolom dengan lebar yang berbeda.
- `st.image()`: Menampilkan gambar di kolom kanan halaman beranda.

G. Halaman Dataset

```

67 # Fungsi untuk halaman dataset
68 def dataset_view():
69     st.title("🏨 Dataset Hotel")
70     st.markdown("---")
71
72     col1, col2 = st.columns([3,1])
73     with col1:
74         st.dataframe(dataset, use_container_width=True)
75     with col2:
76         st.metric("Jumlah Data", f"{dataset.shape[0]:,}")
77         st.metric("Jumlah Kolom", f"{dataset.shape[1]:,}")
78
79         st.markdown("### 📊 Ringkasan Statistik")
80         st.write("Harga Terendah:", f"Rp {dataset['Room Price'].min():,.2f}")
81         st.write("Harga Tertinggi:", f"Rp {dataset['Room Price'].max():,.2f}")
82         st.write("Harga Rata-rata:", f"Rp {dataset['Room Price'].mean():,.2f}")

```

Menampilkan dataset lengkap dan ringkasan statistik tentang data hotel.

- `st.dataframe()`: Menampilkan dataset dalam bentuk tabel interaktif.
- `st.metric()`: Menampilkan metrik berupa jumlah data dan jumlah kolom.
- `st.write()`: Menampilkan harga terendah, tertinggi, dan rata-rata dari kolom harga kamar.

H. Halaman Visualisasi

```

84 # Fungsi untuk halaman visualisasi
85 def visualisasi():
86     st.title("📊 Visualisasi Data")
87     st.markdown("---")
88
89     tab1, tab2, tab3, tab4 = st.tabs(["📊 Distribusi Harga", "📍 Harga per Lokasi", "🏠 Analisis Kamar", "★ Rating Hotel"])
90
91     with tab1:
92         st.markdown("### Distribusi Harga Kamar Hotel")
93         fig1, ax1 = plt.subplots(figsize=(10, 6))
94         sns.histplot(dataset["Room Price"], kde=True, ax=ax1)
95         ax1.set_title("Distribusi Harga Kamar")
96         st.pyplot(fig1)
97
98     with tab2:
99         st.markdown("### Perbandingan Harga Berdasarkan Lokasi")
100        fig2, ax2 = plt.subplots(figsize=(12, 6))
101        sns.boxplot(data=dataset, x="Location", y="Room Price", ax=ax2)
102        plt.xticks(rotation=45)
103        ax2.set_title("Harga Kamar Berdasarkan Lokasi")
104        st.pyplot(fig2)

```



```

106     with tab3:
107         st.markdown("### Analisis Harga Berdasarkan Tipe Kamar dan Tempat Tidur")
108
109         col1, col2 = st.columns(2)
110
111         with col1:
112             avg_room_price = dataset.groupby("Room Type")["Room Price"].mean().sort_values(ascending=True)
113             fig3, ax3 = plt.subplots(figsize=(10, 6))
114             avg_room_price.plot(kind='barh', ax=ax3)
115             ax3.set_title("Rata-rata Harga per Jenis Kamar")
116             ax3.set_xlabel("Harga (Rp)")
117             plt.tight_layout()
118             st.pyplot(fig3)
119
120         with col2:
121             avg_bed_price = dataset.groupby("Bed Type")["Room Price"].mean().sort_values(ascending=True)
122             fig4, ax4 = plt.subplots(figsize=(10, 6))
123             avg_bed_price.plot(kind='barh', ax=ax4)
124             ax4.set_title("Rata-rata Harga per Tipe Tempat Tidur")
125             ax4.set_xlabel("Harga (Rp)")
126             plt.tight_layout()
127             st.pyplot(fig4)
128
129         st.markdown("### 📊 Ringkasan Statistik")
130         col3, col4, col5 = st.columns(3)
131
132         with col3:
133             st.metric("Harga Tertinggi", f"Rp {dataset['Room Price'].max():,.2f}")
134         with col4:
135             st.metric("Harga Terendah", f"Rp {dataset['Room Price'].min():,.2f}")
136         with col5:
137             st.metric("Harga Rata-rata", f"Rp {dataset['Room Price'].mean():,.2f}")
138
139     with tab4:
140         st.markdown("### ⭐ Line Chart Rating Hotel")
141         if "Rating" in dataset.columns: # Pastikan kolom Rating ada
142             sorted_dataset = dataset.sort_values(by="Hotel Name") # Urutkan berdasarkan nama hotel
143             fig5, ax5 = plt.subplots(figsize=(12, 6))
144             sns.lineplot(data=sorted_dataset, x="Hotel Name", y="Rating", ax=ax5, marker="o")
145             ax5.set_title("Rating Hotel")
146             ax5.set_xlabel("Nama Hotel")
147             ax5.set_ylabel("Rating")
148             plt.xticks(rotation=45)
149             plt.tight_layout()
150             st.pyplot(fig5)
151         else:
152             st.error("Kolom 'Rating' tidak ditemukan dalam dataset!")

```

Halaman ini menghasilkan visualisasi distribusi harga, harga berdasarkan lokasi, dan analisis harga kamar serta tempat tidur.

- `st.tabs()`: Membagi halaman visualisasi menjadi beberapa tab untuk memudahkan akses.
- Visualisasi menggunakan `matplotlib` dan `seaborn`: Menampilkan grafik distribusi harga kamar, boxplot per lokasi, dan rata-rata harga per jenis kamar dan tempat tidur.

I. Halaman Prediksi

```

154 # Fungsi untuk halaman prediksi
155 def prediksi():
156     st.title("🏨 Prediksi Harga Hotel")
157     st.markdown("---")
158
159     # Penjelasan cara kerja prediksi
160     st.markdown("""
161     ### 📋 Cara Kerja Prediksi:
162     1. Model mempelajari pola dari dataset yang berisi data historis harga hotel
163     2. Prediksi dilakukan berdasarkan 3 faktor utama:
164         - Lokasi hotel yang dipilih
165         - Jenis kamar yang diinginkan
166         - Tipe tempat tidur yang tersedia
167     3. Model akan menganalisis data historis untuk menemukan harga yang paling sesuai
168     """)
169
170     col1, col2 = st.columns([1,1])
171     col1, col2 = st.columns([1,1])
172
173     with col1:
174         st.markdown("### Parameter Input")
175         lokasi = st.selectbox("📍 Lokasi", dataset["Location"].unique())
176         jenis_kamar = st.selectbox("🛏️ Jenis Kamar", dataset["Room Type"].unique())
177         jenis_tempat_tidur = st.selectbox("🛏️ Jenis Tempat Tidur", dataset["Bed Type"].unique())
178
179     with col2:
180         st.markdown("### Hasil Prediksi")
181
182         X = dataset[["Location", "Room Type", "Bed Type"]]
183         X = pd.get_dummies(X, drop_first=True)
184         y = dataset["Room Price"]
185
186         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
187         model = RandomForestRegressor(random_state=42)
188         model.fit(X_train, y_train)
189
190         data_input = pd.DataFrame([[lokasi, jenis_kamar, jenis_tempat_tidur]],
191                                   columns=["Location", "Room Type", "Bed Type"])
192         data_input = pd.get_dummies(data_input)
193         data_input = data_input.reindex(columns=X.columns, fill_value=0)
194         hasil_prediksi = model.predict(data_input)
195
196         st.metric("🏠 Prediksi Harga", f"Rp {hasil_prediksi[0]:.2f}")
197
198         y_pred = model.predict(X_test)
199         mae = mean_absolute_error(y_test, y_pred)
200         st.metric("📊 Tingkat Akurasi Prediksi", f"±Rp {mae:.2f}")
201
202         st.info("""
203         💡 Informasi Tingkat Akurasi:
204         - Angka ini menunjukkan rata-rata selisih antara harga prediksi dengan harga sebenarnya
205         - Contoh: Jika tingkat akurasi ±Rp 100.000, prediksi harga bisa lebih tinggi atau lebih rendah sekitar Rp 100.000
206         - Semakin kecil nilai ini, semakin akurat prediksi yang dihasilkan
207         """)

```

Halaman ini memungkinkan pengguna untuk memilih lokasi, jenis kamar, dan tempat tidur, kemudian menggunakan model **RandomForestRegressor** untuk memprediksi harga kamar hotel.

- `st.selectbox()`: Menyediakan dropdown untuk memilih lokasi, jenis kamar, dan tempat tidur.
- `train_test_split()`: Membagi data menjadi data latih dan data uji.
- `RandomForestRegressor`: Model machine learning yang digunakan untuk memprediksi harga.
- `model.predict()`: Melakukan prediksi berdasarkan input pengguna.

J. Router Halaman

```
208 # Router halaman
209 if selected == "Beranda":
210     beranda()
211 elif selected == "Dataset":
212     dataset_view()
213 elif selected == "Visualisasi":
214     visualisasi()
215 elif selected == "Prediksi":
216     prediksi()
```

Bagian ini mengarahkan aplikasi untuk menampilkan halaman yang sesuai dengan menu yang dipilih di sidebar.

- `if selected == ...`: Berdasarkan pilihan menu, aplikasi menampilkan halaman yang sesuai (beranda, dataset, visualisasi, atau prediksi).