J.62DMI00.005.1

Modul 2: Menelaah Data Menggunakan Python

Modul ini memperkenalkan teknik analisis dan telaah data menggunakan Python. Penelaahan data merupakan langkah krusial setelah pengumpulan data untuk memahami karakteristik, pola, dan hubungan dalam dataset. Melalui modul ini, Anda akan mempelajari cara mengidentifikasi tipe data, menganalisis karakteristik data, dan membuat laporan telaah data yang komprehensif.

Perangkat Pendukung

No.	Sistem	Keterangan
1	Python versi 3.7 atau lebih tinggi	Untuk kebutuhan kompilasi kode
2	Jupyter Notebook atau Google Colab	Sebagai alat kompilasi python berbasis cloud
3	Library Python: pandas, numpy, matplotlib, seaborn	Sebagai library untuk analisis dan visualisasi data

Dataset yang Digunakan

Dalam modul ini, kita akan menggunakan dataset berikut yang tersedia melalui repository:

1. Data Mahasiswa

   ```
   https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Mahasiswa/Data%20mahasiswa.csv
   ```

2. Data Student Depression Dataset

   ```
   https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dataset/student_depression_dataset.csv
   ```

3. Data SPMB Politeknik Statistika STIS

   ```
   https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Hasil%20SPMB%20Politeknik%20Statistika%20STIS%202024/spmb2024_1_skd_polstatstis.csv
   ```

Pengenalan Library Visualisasi dan Analisis Data

Sebelum masuk ke materi utama, mari kita pahami beberapa library dan fungsi yang akan sering digunakan dalam telaah data:

1. Seaborn (sns)

Seaborn adalah library visualisasi data berbasis matplotlib yang menyediakan antarmuka tingkat tinggi untuk membuat grafik statistik yang menarik dan informatif. Berikut beberapa fungsi utama yang akan kita gunakan:

a. sns.histplot()

```
sns.histplot(x=None, data=None, kde=False, bins='auto', color=None)
```

Parameter	Tipe Data	Deskripsi	Wajib/Opsional	Default
`x`	array_like, str	Data untuk sumbu x atau nama kolom dalam dataframe	Opsional	`None`
`data`	DataFrame	DataFrame yang berisi variabel untuk diplot	Opsional	`None`
`kde`	bool	Apakah menampilkan estimasi kepadatan kernel	Opsional	`False`
`bins`	int, str, array	Jumlah bin atau strategi penentuan bin	Opsional	`'auto'`
`color`	str, tuple	Warna untuk semua elemen plot	Opsional	`None`

Contoh: `sns.histplot(df_mahasiswa['Nilai Mapel UN'], kde=True, bins=20)` akan membuat histogram dengan 20 bins dan menampilkan kurva distribusi dari variabel Nilai Mapel UN.

Dokumentasi: https://seaborn.pydata.org/generated/seaborn.histplot.html

b. sns.boxplot()

```
sns.boxplot(x=None, y=None, hue=None, data=None, palette=None, orient=None)
```

Parameter	Tipe Data	Deskripsi	Wajib/Opsional	Default
`x`	str	Nama variabel di data atau posisi di array untuk sumbu x	Opsional	`None`
`y`	str	Nama variabel di data atau posisi di array untuk sumbu y	Opsional	`None`
`hue`	str	Nama variabel untuk pengelompokan warna	Opsional	`None`
`data`	DataFrame	DataFrame yang berisi variabel untuk diplot	Opsional	`None`
`palette`	str, list	Nama pallete seaborn atau daftar warna	Opsional	`None`
`orient`	str	Orientasi box: "v" (vertical) atau "h" (horizontal)	Opsional	Ditentukan dari input

Contoh: `sns.boxplot(x='JK', y='Nilai Mapel UN', data=df_mahasiswa)` membuat boxplot yang membandingkan distribusi Nilai Mapel UN berdasarkan Jenis Kelamin.

Dokumentasi: https://seaborn.pydata.org/generated/seaborn.boxplot.html

c. sns.pairplot()

```
sns.pairplot(data, hue=None, vars=None, corner=False, diag_kind='auto')
```

Parameter	Tipe Data	Deskripsi	Wajib/Opsional	Default
`data`	DataFrame	DataFrame yang berisi variabel untuk diplot	Wajib	-
`hue`	str	Variabel untuk pengelompokan warna	Opsional	`None`
`vars`	list	Daftar variabel yang akan diplot	Opsional	`None`
`corner`	bool	Jika True, hanya menampilkan bagian bawah matriks	Opsional	`False`
`diag_kind`	str	Jenis plot untuk diagonal: "hist" atau "kde"	Opsional	`'auto'`

Contoh: `sns.pairplot(df_mahasiswa[['Nilai Mapel UN', 'X1', 'X2']], hue='JK')` membuat matriks scatter plot untuk semua kombinasi variabel yang dipilih, dengan warna berbeda berdasarkan Jenis Kelamin.

Dokumentasi: https://seaborn.pydata.org/generated/seaborn.pairplot.html

2. Pandas

Pandas menyediakan beberapa fungsi analisis statistik yang sangat berguna:

a. df.corr()

```
DataFrame.corr(method='pearson', min_periods=1, numeric_only=None)
```

Parameter	Tipe Data	Deskripsi	Wajib/Opsional	Default
`method`	str	Metode korelasi: 'pearson', 'kendall', 'spearman'	Opsional	`'pearson'`
`min_periods`	int	Jumlah minimum observasi yang diperlukan	Opsional	`1`
`numeric_only`	bool	Hanya termasuk kolom numerik	Opsional	`None`

Contoh: `correlation = df[numerical_cols].corr()` menghitung korelasi antar semua variabel numerik menggunakan metode Pearson.

Dokumentasi: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html

b. df.quantile()

```
DataFrame.quantile(q=0.5, axis=0, numeric_only=True, interpolation='linear')
```

Parameter	Tipe Data	Deskripsi	Wajib/Opsional	Default
`q`	float, list	Kuantil atau daftar kuantil (0-1)	Opsional	`0.5`
`axis`	int	0 untuk baris, 1 untuk kolom	Opsional	`0`
`numeric_only`	bool	Hanya pertimbangkan kolom numerik	Opsional	`True`
`interpolation`	str	Metode interpolasi	Opsional	`'linear'`

Contoh: `Q1 = df[column].quantile(0.25)` mengembalikan nilai pada kuantil 25% (kuartil pertama).

Dokumentasi:
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.quantile.html

c. df.groupby()

```
DataFrame.groupby(by, axis=0, level=None, as_index=True, sort=True,
group_keys=True)
```

Parameter	Tipe Data	Deskripsi	Wajib/Opsional	Default
`by`	str, list, array	Kriteria untuk membentuk kelompok	Wajib	-
`axis`	int	0 untuk baris, 1 untuk kolom	Opsional	`0`
`as_index`	bool	Return object dengan grup labels sebagai indeks	Opsional	`True`
`sort`	bool	Mengurutkan grup	Opsional	`True`

Contoh: `df.groupby('Pilihan 1')['Nilai Mapel UN'].mean()` menghitung rata-rata Nilai Mapel UN untuk setiap kelompok Pilihan 1.

Dokumentasi:
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.groupby.html

Materi Pembelajaran

1. Analisis Awal dengan Fungsi Dasar

Langkah pertama dalam menelaah data adalah memahami gambaran umum dataset melalui fungsi-fungsi dasar pandas:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Memuat data
df_mahasiswa =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Mahasiswa/Data%20mahasiswa.csv')

# Melihat 5 baris pertama
print(df_mahasiswa.head())

# Informasi struktur dataset
print(df_mahasiswa.info())

# Statistik deskriptif
print(df_mahasiswa.describe())
```

Output: Kode di atas akan menampilkan:

1. Lima baris pertama dari dataset mahasiswa yang memuat informasi seperti nomor pendaftaran, jenis kelamin, status bidikmisi, sekolah asal, dan berbagai fitur akademik lainnya.
2. Ringkasan struktur dataset seperti jumlah total baris, nama kolom, jumlah nilai non-null, dan tipe data masing-masing kolom.
3. Statistik deskriptif untuk kolom numerik, termasuk jumlah data, nilai rata-rata, standar deviasi, nilai minimum, kuartil, dan nilai maksimum.

2. Analisis Tipe Data dan Relasi

Penting untuk memahami tipe data dalam dataset dan bagaimana data-data ini saling berhubungan:

```
# Menampilkan tipe data setiap kolom
print(df_mahasiswa.dtypes)

# Mengidentifikasi kolom kategorikal vs numerik
categorical_cols =
df_mahasiswa.select_dtypes(include=['object']).columns.tolist()
numerical_cols = df_mahasiswa.select_dtypes(include=['int64',
'float64']).columns.tolist()

print("Kolom kategorikal:", categorical_cols)
print("Kolom numerik:", numerical_cols)

# Analisis korelasi untuk kolom numerik
correlation = df_mahasiswa[numerical_cols].corr()
print(correlation)
```

Output: Kode ini akan menghasilkan:

1. Daftar tipe data untuk setiap kolom dalam dataset (misalnya: 'JK' mungkin bertipe object/string, 'Nilai Mapel UN' bertipe float, dll).
2. Pemisahan kolom menjadi dua kategori: kategorikal (seperti 'JK', 'Provinsi', 'Sekolah') dan numerik (seperti 'Nilai Mapel UN', 'Ranking Sekolah').
3. Matriks korelasi yang menunjukkan hubungan antar variabel numerik. Nilai mendekati 1 menunjukkan korelasi positif kuat (misalnya, mungkin ada korelasi positif antara 'IP Sem 1' dan 'IP Sem 2'), nilai mendekati -1 menunjukkan korelasi negatif, dan nilai mendekati 0 menunjukkan tidak ada korelasi yang signifikan.

3. Visualisasi Data untuk Analisis Karakteristik

Visualisasi membantu mengidentifikasi pola dan karakteristik dalam data:

```
# Histogram untuk melihat distribusi nilai mahasiswa
plt.figure(figsize=(10, 6))
sns.histplot(df_mahasiswa['Nilai Mapel UN'], kde=True)
plt.title('Distribusi Nilai UN Mahasiswa')
plt.xlabel('Nilai')
plt.ylabel('Frekuensi')
plt.show()
```

Output: Histogram ini menampilkan distribusi nilai Ujian Nasional mahasiswa. Visualisasi akan menunjukkan bentuk distribusi nilai UN - apakah nilai cenderung terpusat di tengah (normal), condong ke kanan (positively skewed), atau condong ke kiri (negatively skewed). Kurva KDE (garis biru) memperhalus distribusi dan membantu mengidentifikasi puncak-puncak dalam distribusi nilai.

```
# Boxplot untuk identifikasi outlier
plt.figure(figsize=(10, 6))
sns.boxplot(x='JK', y='Nilai Mapel UN', data=df_mahasiswa)
plt.title('Perbandingan Nilai UN Berdasarkan Jenis Kelamin')
plt.show()
```

Output: Boxplot ini membandingkan distribusi nilai UN berdasarkan jenis kelamin. Dari visualisasi ini dapat terlihat:

1. Median nilai untuk masing-masing jenis kelamin (garis horizontal di tengah box)
2. Rentang interkuartil (tinggi box) yang menunjukkan sebaran nilai tengah
3. Nilai minimum dan maksimum yang masih dalam batas normal (whiskers)
4. Outlier (titik-titik di luar whiskers) yang menunjukkan nilai yang jauh di atas atau di bawah nilai umumnya
5. Perbedaan distribusi nilai antara laki-laki dan perempuan, termasuk potensi perbedaan performa akademik berdasarkan jenis kelamin

```
# Pairplot untuk melihat hubungan antar variabel numerik
sns.pairplot(df_mahasiswa[['Nilai Mapel UN', 'X1', 'X2', 'X3']])
plt.suptitle('Hubungan antar Variabel Numerik', y=1.02)
plt.show()
```

Output: Pairplot ini menampilkan matriks scatter plot untuk semua kombinasi dari empat variabel numerik ('Nilai Mapel UN', 'X1', 'X2', 'X3'). Diagonal utama menampilkan distribusi masing-masing variabel, sedangkan plot lainnya menunjukkan hubungan antar dua variabel. Dari visualisasi ini, kita dapat melihat:

1. Pola hubungan antar variabel (linear, kuadratik, atau tidak ada pola)
2. Kekuatan hubungan (seberapa erat titik-titik data mengelompok di sekitar pola)
3. Adanya outlier atau kelompok (cluster) dalam data
4. Distribusi masing-masing variabel (pada diagonal)

4. Analisis Kelompok Menggunakan Groupby

Fungsi groupby memungkinkan analisis berdasarkan kategori tertentu:

```
# Analisis nilai berdasarkan jurusan
nilai_per_prodi = df_mahasiswa.groupby('Pilihan 1')['Nilai Mapel
UN'].agg(['mean', 'median', 'std', 'count'])
print(nilai_per_prodi.sort_values(by='mean', ascending=False))
```

Output: Tabel yang menampilkan statistik nilai UN untuk setiap program studi pilihan pertama, diurutkan dari rata-rata tertinggi ke terendah. Tabel ini berisi kolom:

1. 'mean' - rata-rata nilai UN untuk setiap prodi
2. 'median' - nilai tengah UN untuk setiap prodi
3. 'std' - standar deviasi yang menunjukkan sebaran nilai
4. 'count' - jumlah mahasiswa di masing-masing prodi

Data ini memungkinkan perbandingan antar program studi, seperti prodi mana yang menarik mahasiswa dengan nilai tertinggi atau prodi mana yang memiliki variasi nilai paling besar.

```
# Visualisasi perbandingan nilai antar prodi
plt.figure(figsize=(12, 6))
sns.barplot(x='Pilihan 1', y='Nilai Mapel UN', data=df_mahasiswa)
plt.title('Perbandingan Nilai UN Berdasarkan Pilihan Prodi')
plt.xticks(rotation=90)
plt.show()
```

Output: Diagram batang yang menampilkan rata-rata nilai UN untuk setiap program studi pilihan pertama. Visualisasi ini mempermudah perbandingan visual antar program studi, dan memungkinkan identifikasi prodi mana yang memiliki rata-rata nilai tertinggi dan terendah. Error bar pada setiap batang menunjukkan interval kepercayaan untuk rata-rata nilai di setiap prodi.

5. Analisis Outlier dan Nilai Ekstrem

Identifikasi outlier penting untuk memastikan kualitas analisis:

```
# Identifikasi outlier menggunakan metode IQR
def detect_outliers(df, column):
```

```
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)]
    return outliers, lower_bound, upper_bound

outliers, lower, upper = detect_outliers(df_mahasiswa, 'Nilai Mapel UN')
print(f"Jumlah outlier: {len(outliers)}")
print(f"Batas bawah: {lower}, Batas atas: {upper}")
print(outliers.head())
```

Output: Kode ini mengidentifikasi outlier dalam kolom 'Nilai Mapel UN' menggunakan metode IQR (Interquartile Range). Hasilnya berupa:

1. Jumlah total outlier yang ditemukan
2. Batas bawah dan atas untuk identifikasi outlier (nilai di luar rentang ini dianggap outlier)
3. Tampilan beberapa baris pertama dari data outlier, yang menunjukkan detail mahasiswa dengan nilai UN yang sangat tinggi atau sangat rendah dibandingkan dengan mayoritas populasi

Identifikasi outlier ini penting untuk analisis statistik yang akurat, karena outlier dapat mempengaruhi hasil perhitungan rata-rata, standar deviasi, dan analisis lainnya.

6. Membuat Laporan Telaah Data

Hasil telaah data perlu didokumentasikan dalam laporan yang komprehensif:

```
def generate_analysis_report(df, numerical_columns, categorical_columns,
title="Laporan Analisis Data"):
    """
    Menghasilkan laporan analisis data yang komprehensif
    """
    print("="*80)
    print(f"{title:^80}")
    print("="*80)

    # Informasi umum dataset
    print("\n1. INFORMASI UMUM DATASET")
    print(f"   Jumlah baris: {df.shape[0]}")
    print(f"   Jumlah kolom: {df.shape[1]}")
    print(f"   Missing values: {df.isnull().sum().sum()}")
    print(f"   Duplikasi: {df.duplicated().sum()}")

    # Analisis kolom numerik
    print("\n2. ANALISIS KOLOM NUMERIK")
    for col in numerical_columns:
```

```
        print(f"\n   Kolom: {col}")
        print(f"   Min: {df[col].min():.2f}")
        print(f"   Max: {df[col].max():.2f}")
        print(f"   Mean: {df[col].mean():.2f}")
        print(f"   Median: {df[col].median():.2f}")
        print(f"   Std Dev: {df[col].std():.2f}")

    # Analisis kolom kategorikal
    print("\n3. ANALISIS KOLOM KATEGORIKAL")
    for col in categorical_columns[:3]:  # Batasi 3 kolom saja
        print(f"\n   Kolom: {col}")
        print(f"   Jumlah kategori unik: {df[col].nunique()}")
        print(f"   Kategori teratas:
{df[col].value_counts().head(3).to_dict()}")

    # Kesimpulan
    print("\n4. KESIMPULAN")
    print("   • Dataset memiliki kualitas yang cukup baik dengan jumlah
missing values minimal")
    print("   • Terdapat variasi nilai yang signifikan pada kolom numerik")
    print("   • Perlu analisis lebih lanjut untuk kolom kategorikal dengan
kardinalitas tinggi")

# Contoh penggunaan
generate_analysis_report(
    df_mahasiswa,
    numerical_columns=['Nilai Mapel UN', 'X1', 'X2'],
    categorical_columns=['JK', 'Pilihan 1', 'Provinsi']
)
```

Output: Fungsi ini menghasilkan laporan analisis data yang terstruktur dan komprehensif, terdiri dari:

1. Header dengan judul laporan
2. Informasi umum dataset (jumlah baris, kolom, missing values, duplikasi)
3. Analisis kolom numerik (nilai minimum, maksimum, rata-rata, median, dan standar deviasi)
4. Analisis kolom kategorikal (jumlah kategori unik dan kategori yang paling sering muncul)
5. Kesimpulan umum tentang kualitas dataset

Laporan ini memberikan gambaran menyeluruh tentang dataset, memungkinkan pembuat keputusan untuk mendapatkan insight cepat tentang karakteristik data tanpa harus melakukan analisis terpisah. Format tersusun memudahkan pembacaan dan perbandingan antar kolom.

Pertanyaan

1. **Judul**: Mengenali fungsi dasar analisis data
   Deskripsi: Tulislah kode untuk menampilkan informasi umum dan ringkasan statistik dari dataset student depression menggunakan fungsi-fungsi dasar pandas.
   Contoh Kode:

```
import pandas as pd

# Load dataset student depression
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dataset/student_depression_dataset.csv')

# Menampilkan informasi umum dataset
print("Informasi Dataset:")
df.info()

# Menampilkan ringkasan statistik
print("\nRingkasan Statistik:")
print(df.describe())

# Memeriksa 5 baris pertama data
print("\nSample Data:")
print(df.head())
```

   Test Case:

   - self.assertIn("describe()", student_code)
   - self.assertIn("info()", student_code)

2. **Judul**: Mengidentifikasi tipe data dan relasinya
   Deskripsi: Tulislah kode untuk mengidentifikasi tipe data dalam dataset mahasiswa dan menampilkan korelasi antar variabel numerik seperti 'CGPA', 'Academic Pressure', dan 'Work Pressure'.
   Contoh Kode:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dataset/student_depression_dataset.csv')

# Identifikasi tipe data
print("Tipe data per kolom:")
```

```
print(df.dtypes)

# Memisahkan kolom numerik dan kategorikal
numerical_cols = df.select_dtypes(include=['int64',
'float64']).columns.tolist()
categorical_cols =
df.select_dtypes(include=['object']).columns.tolist()

print("\nKolom numerik:", numerical_cols)
print("Kolom kategorikal:", categorical_cols)

# Analisis korelasi
correlation = df[['CGPA', 'Academic Pressure', 'Work Pressure']].corr()
print("\nMatriks Korelasi:")
print(correlation)

# Visualisasi heatmap korelasi
plt.figure(figsize=(10, 6))
sns.heatmap(correlation, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Korelasi Antar Variabel Akademik')
plt.tight_layout()
plt.show()
```

Test Case:

- self.assertIn("corr", student_code)

3. **Judul**: Membuat visualisasi data untuk analisis karakteristik
Deskripsi: Buat histogram untuk menampilkan distribusi CGPA mahasiswa dan boxplot untuk membandingkan Academic Pressure berdasarkan Gender dari dataset student depression.
Contoh Kode:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codea
sy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dat
aset/student_depression_dataset.csv')

# Set style
sns.set(style="whitegrid")

# Plot histogram untuk CGPA
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='CGPA', kde=True, bins=20)
plt.title('Distribusi CGPA Mahasiswa')
```

```
plt.xlabel('CGPA')
plt.ylabel('Frekuensi')
plt.show()

# Plot boxplot untuk Academic Pressure berdasarkan Gender
plt.figure(figsize=(10, 6))
sns.boxplot(x='Gender', y='Academic Pressure', data=df)
plt.title('Perbandingan Academic Pressure Berdasarkan Gender')
plt.xlabel('Gender')
plt.ylabel('Academic Pressure')
plt.show()

# Tambahan: Scatterplot untuk melihat hubungan CGPA dengan Academic
Pressure
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Academic Pressure', y='CGPA', hue='Gender', data=df)
plt.title('Hubungan Academic Pressure dan CGPA')
plt.xlabel('Academic Pressure')
plt.ylabel('CGPA')
plt.legend(title='Gender')
plt.show()
```

Test Case:

- self.assertIn("sns.histplot", student_code)

Pre-code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codea
sy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dat
aset/student_depression_dataset.csv')

# Tulis kode visualisasi Anda di sini
```

4. **Judul**: Analisis data berdasarkan kelompok
   Deskripsi: Analisis tingkat depresi mahasiswa berdasarkan kelompok Gender, Profession, dan Degree menggunakan fungsi groupby.
   Contoh Kode:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
```

```
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dataset/student_depression_dataset.csv')

# Analisis tingkat depresi berdasarkan Gender
gender_depression = df.groupby('Gender')['Depression'].agg(['mean', 'median', 'std', 'count'])
print("Analisis Depresi Berdasarkan Gender:")
print(gender_depression)

# Analisis tingkat depresi berdasarkan Profession
profession_depression =
df.groupby('Profession')['Depression'].agg(['mean', 'median', 'std', 'count'])
print("\nAnalisis Depresi Berdasarkan Profession:")
print(profession_depression.sort_values(by='mean', ascending=False))

# Analisis tingkat depresi berdasarkan Degree
degree_depression = df.groupby('Degree')['Depression'].agg(['mean', 'median', 'std', 'count'])
print("\nAnalisis Depresi Berdasarkan Degree:")
print(degree_depression.sort_values(by='mean', ascending=False))

# Visualisasi
plt.figure(figsize=(12, 6))
sns.barplot(x='Profession', y='Depression', data=df)
plt.title('Rata-rata Tingkat Depresi Berdasarkan Profession')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

plt.figure(figsize=(10, 6))
sns.barplot(x='Degree', y='Depression', data=df)
plt.title('Rata-rata Tingkat Depresi Berdasarkan Degree')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Test Case:

- self.assertIn("groupby", student_code)

Pre-code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
```

```
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dataset/student_depression_dataset.csv')

# Tulis kode analisis kelompok Anda di sini
```

5. **Judul**: Identifikasi outlier dalam dataset
Deskripsi: Tuliskan kode untuk mengidentifikasi outlier pada kolom 'Academic Pressure', 'Work Pressure', dan 'CGPA' menggunakan metode quantile pada dataset student depression.
Contoh Kode:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dataset/student_depression_dataset.csv')

# Fungsi untuk identifikasi outlier
def identify_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    outliers = df[(df[column] < lower_bound) | (df[column] >
upper_bound)]

    print(f"\nOutliers di kolom {column}:")
    print(f"  Jumlah: {len(outliers)}")
    print(f"  Persentase: {len(outliers) / len(df) * 100:.2f}%")
    print(f"  Batas bawah: {lower_bound:.2f}")
    print(f"  Batas atas: {upper_bound:.2f}")

    return outliers

# Identifikasi outlier pada beberapa kolom
academic_outliers = identify_outliers(df, 'Academic Pressure')
work_outliers = identify_outliers(df, 'Work Pressure')
cgpa_outliers = identify_outliers(df, 'CGPA')
```

```
# Visualisasi outlier dengan boxplot
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
sns.boxplot(y=df['Academic Pressure'])
plt.title('Outliers: Academic Pressure')

plt.subplot(1, 3, 2)
sns.boxplot(y=df['Work Pressure'])
plt.title('Outliers: Work Pressure')

plt.subplot(1, 3, 3)
sns.boxplot(y=df['CGPA'])
plt.title('Outliers: CGPA')

plt.tight_layout()
plt.show()
```

Test Case:

- self.assertIn("quantile", student_code)

Pre-code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dataset/student_depression_dataset.csv')

# Definisikan fungsi untuk identifikasi outlier
def identify_outliers(df, column):
    # Implementasi fungsi Anda di sini
    pass

# Gunakan fungsi untuk mengidentifikasi outlier
```

6. **Judul**: Membuat fungsi laporan analisis data
   Deskripsi: Buatlah fungsi bernama generate_analysis_report yang menerima dataframe dan menghasilkan laporan analisis komprehensif untuk dataset student depression, meliputi statistik umum, distribusi variabel, dan korelasi antar faktor depresi.
   Contoh Kode:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

def generate_analysis_report(df, title="Laporan Analisis Data Depression Mahasiswa"):
    """
    Fungsi untuk menghasilkan laporan analisis komprehensif dari dataset depresi mahasiswa

    Parameters:
    df (DataFrame): Dataset yang akan dianalisis
    title (str): Judul laporan

    Returns:
    None: Hasil analisis dicetak ke console
    """
    print("="*80)
    print(f"{title:^80}")
    print("="*80)

    # 1. Informasi Umum Dataset
    print("\n1. INFORMASI UMUM DATASET")
    print(f"   Jumlah observasi: {df.shape[0]}")
    print(f"   Jumlah variabel: {df.shape[1]}")
    print(f"   Missing values: {df.isnull().sum().sum()}")
    print(f"   Duplikasi data: {df.duplicated().sum()}")

    # 2. Statistik Deskriptif
    print("\n2. STATISTIK DESKRIPTIF VARIABEL NUMERIK")
    numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns
    print(df[numeric_cols].describe().round(2).to_string())

    # 3. Distribusi Tingkat Depresi
    print("\n3. DISTRIBUSI TINGKAT DEPRESI")
    depression_counts = df['Depression'].value_counts().sort_index()
    print(depression_counts)

    # 4. Analisis Faktor-Faktor Depresi
    print("\n4. FAKTOR-FAKTOR YANG BERKORELASI DENGAN DEPRESI")
    depression_corr = df[numeric_cols].corr()['Depression'].sort_values(ascending=False)
    print(depression_corr)

    # 5. Analisis Depresi Berdasarkan Demografis
    print("\n5. ANALISIS DEPRESI BERDASARKAN DEMOGRAFIS")
```

```
    print("\n   5.1 Berdasarkan Gender")
    gender_stats = df.groupby('Gender')['Depression'].agg(['mean',
'count']).round(2)
    print(gender_stats)

    print("\n   5.2 Berdasarkan Profession")
    profession_stats =
df.groupby('Profession')['Depression'].agg(['mean',
'count']).sort_values(by='mean', ascending=False).round(2)
    print(profession_stats)

    # 6. Kesimpulan
    print("\n6. KESIMPULAN DAN REKOMENDASI")
    print("   • Tingkat depresi rata-rata dalam populasi mahasiswa:",
round(df['Depression'].mean(), 2))

    highest_depression_group = profession_stats.index[0]
    print(f"   • Kelompok dengan tingkat depresi tertinggi:
{highest_depression_group}")

    top_correlated_factor = depression_corr.index[1]  # Index 0 is
Depression itself
    corr_value = depression_corr.iloc[1]
    print(f"   • Faktor yang paling berkorelasi dengan depresi:
{top_correlated_factor} (r={corr_value:.2f})")

    print("   • Rekomendasi: Fokus intervensi pada kelompok berisiko
tinggi dan menyediakan dukungan khusus")

# Contoh penggunaan
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codea
sy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dat
aset/student_depression_dataset.csv')
generate_analysis_report(df)

# Visualisasi tambahan untuk laporan
plt.figure(figsize=(15, 10))

# Plot distribusi depresi
plt.subplot(2, 2, 1)
sns.histplot(df['Depression'], kde=True, bins=10)
plt.title('Distribusi Tingkat Depresi')

# Plot korelasi faktor dengan depresi
plt.subplot(2, 2, 2)
corr_data = df[['Depression', 'Academic Pressure', 'Work Pressure',
'CGPA']].corr()['Depression'].sort_values(ascending=False)[1:]
sns.barplot(x=corr_data.index, y=corr_data.values)
```

```
plt.title('Korelasi Faktor dengan Depresi')
plt.xticks(rotation=45)

# Plot depresi berdasarkan gender
plt.subplot(2, 2, 3)
sns.boxplot(x='Gender', y='Depression', data=df)
plt.title('Depresi Berdasarkan Gender')

# Plot depresi berdasarkan profession
plt.subplot(2, 2, 4)
top_professions =
df.groupby('Profession')['Depression'].mean().nlargest(5).index
sns.barplot(x='Depression', y='Profession',
data=df[df['Profession'].isin(top_professions)], estimator=np.mean)
plt.title('Profesi dengan Tingkat Depresi Tertinggi')

plt.tight_layout()
plt.show()
```

Test Case:

- self.assertIn("generate_analysis_report", student_code)

Pre-code:

```
import pandas as pd
import numpy as np

def generate_analysis_report(df, title="Laporan Analisis"):
    # Implementasi fungsi Anda di sini
    pass

# Load dataset
df =
pd.read_csv('https://media.githubusercontent.com/media/evanightly/codeasy/refs/heads/main/fastapi/datasource/Data%20Student%20Depression%20Dataset/student_depression_dataset.csv')

# Panggil fungsi dengan dataset
generate_analysis_report(df)
```