

## Rangkuman Proyek Analisis Data – Dataset Titanic (Versi Parafrase)

### 1. Business Understanding (Pemahaman Bisnis)

Tahap pertama berfokus pada memahami tujuan utama proyek analisis ini. Dataset Titanic memuat berbagai informasi mengenai penumpang—mulai dari usia, jenis kelamin, kelas tiket, hingga status keselamatan mereka.

Tujuan dari analisis ini meliputi:

- Mengidentifikasi faktor-faktor yang memengaruhi peluang seorang penumpang untuk selamat.
- Menghasilkan dataset yang sudah rapi, bebas masalah data, dan siap digunakan untuk analisis lanjutan atau pemodelan prediktif.

### 2. Data Understanding (Pemahaman Data)

Pada tahap ini dilakukan eksplorasi dasar untuk memahami kondisi awal dataset. Pengecekan dilakukan melalui `df.info()`, `df.describe()`, serta beberapa visualisasi.

```
# 1. Mendapatkan ringkasan dasar data
info = df.info(verbose=False, memory_usage=False)

# 2. Membuat DataFrame baru dari informasi kolom
data_description = pd.DataFrame({
    'Nama Kolom (Atribut)': df.columns,
    'Tipe Data': df.dtypes,
    'Jumlah Non-Null': df.apply(lambda x: x.count())
})

# 3. Hitung Jumlah Nilai Hilang
# Jumlah total baris diambil dari .shape[0]
total_rows = df.shape[0]
data_description['Jumlah Nilai Hilang'] = total_rows - data_description['Jumlah Non-Null']
data_description = data_description.set_index('Nama Kolom (Atribut')

# 4. Tampilkan Laporan Deskripsi Data
print("Total Baris (Entries):", total_rows)
print("\nLaporan Deskripsi Atribut:")
print(data_description)

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Columns: 12 entries, PassengerId to Embarked
dtypes: float64(2), int64(5), object(5)
Total Baris (Entries): 891

Laporan Deskripsi Atribut:
  Tipe Data Jumlah Non-Null Jumlah Nilai Hilang
Nama Kolom (Atribut)
PassengerId      int64        891          0
Survived         int64        891          0
Pclass           int64        891          0
Name             object       891          0
Sex              object       891          0
Age              float64      714         177
SibSp            int64        891          0
Parch            int64        891          0
Ticket           object       891          0
Fare             float64      891          0
Cabin            object       284         687
Embarked         object       889          2
```

Ringkasan informasi dataset:

- Total data: 891 baris dan 12 kolom.
- Beberapa kolom penting: Survived, Pclass, Sex, Age, SibSp, Parch, Fare, Embarked.
- Label yang dianalisis: Survived (1 = selamat, 0 = tidak).

Temuan dari eksplorasi awal:

- Usia rata-rata penumpang sekitar 29.7 tahun.
- Tingkat keselamatan sekitar 38%.
- Mayoritas penumpang berada di kelas 3.
- Data kosong ditemukan di kolom Age (177 baris) dan Embarked (2 baris).
- Ditemukan nilai ekstrem (outlier) pada Fare dengan harga tertinggi mencapai 512.

### 3. Data Preparation – Pemilihan & Pembersihan Data

Tahap ini dilakukan untuk memilih fitur yang relevan dan memastikan dataset bebas dari masalah kualitas data.

Langkah-langkah yang dilakukan:

#### 1. Menghapus kolom yang tidak memberi kontribusi signifikan

Kolom PassengerId, Name, Ticket, dan Cabin dihilangkan karena tidak berhubungan langsung dengan peluang keselamatan.

```
Menghapus atribut tidak relevan
[1]: # Melihat semua kolom sebelum penghapusan
print("Daftar kolom awal:\n", df.columns.tolist())

# Menghapus kolom yang tidak relevan untuk analisis
df = df.drop(columns=['PassengerId', 'Name', 'Ticket', 'Cabin'])

# Menampilkan kolom setelah dihapus
print("\nKolom setelah penghapusan atribut tidak relevan:\n", df.columns.tolist())

*** Daftar kolom awal:
[PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked']

Kolom setelah penghapusan atribut tidak relevan:
[Survived', 'Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']
```

#### 2. Menangani data kosong

- Age diisi menggunakan nilai median karena lebih stabil terhadap outlier.
- Embarked diisi menggunakan nilai yang paling sering muncul (modus).

```
Cek missing values
[1]: Ditemukan missing values yang signifikan. Kolom Cabin memiliki 687 nilai hilang, Age memiliki 177 nilai hilang, dan Embarked memiliki 2 nilai hilang.

[2]: 
  import pandas as pd # Pastikan Pandas sudah diimpor

  # 1. Hitung missing values
  missing_data = df.isnull().sum()

  # 2. Ubah hasil hitungan menjadi DataFrame
  missing_df = missing_data.to_frame(name='Jumlah Nilai Hilang')

  # 3. Ubah nama kolom (index) menjadi 'Nama Kolom (Atribut)'
  missing_df.index.name = 'Nama Kolom (Atribut)'

  # 4. Tampilkan DataFrame (Ini akan menghasilkan output tabel)
  print(missing_df)

  ...
  Jumlah Nilai Hilang
  Nama Kolom (Atribut)
  PassengerId          0
  Survived              0
  Pclass                 0
  Name                  0
  Sex                   0
  Age                  177
  SibSp                 0
  Parch                 0
  Ticket                0
  Fare                  0
  Cabin                687
  Embarked              2
```

### 3. Menghapus data duplikat

Terdapat 116 baris duplikat yang kemudian dihapus untuk menjaga keakuratan analisis.

```
Cek duplikasi
hasil pengecekan nya tidak ada baris yang duplikat.

# Cek duplikasi
jumlah_duplikat = df.duplicated().sum()

if jumlah_duplikat == 0:
    print("✅ Tidak ada baris data duplikat ditemukan.")
else:
    print("⚠️ Ditemukan {} baris data duplikat.".format(jumlah_duplikat))
...
✅ Tidak ada baris data duplikat ditemukan.
```

### 4. Menangani outlier

Outlier pada Fare dan Age ditangani menggunakan metode IQR sehingga distribusi data menjadi lebih wajar.

```
detect_and_handle_outliers()
mendeteksi dan menghapus outlier (data ekstrem) dari dataset Titanic, khususnya pada kolom Age (umur) dan Fare (harga tiket).

import seaborn as sns
import matplotlib.pyplot as plt

# Fungsi untuk menghapus outlier
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - 1.5 * IQR
    upper = Q3 + 1.5 * IQR
    return df[(df[column] >= lower) & (df[column] <= upper)]

# Deteksi & hapus outlier pada kolom numerik
print("Jumlah data sebelum hapus outlier: ", df.shape[0])
df = remove_outliers(df, 'Age')
df = remove_outliers(df, 'Fare')
print("Jumlah data setelah hapus outlier: ", df.shape[0])

# Visualisasi distribusi setelah pembersihan
sns.boxplot(x=df['Fare'])
plt.title('Boxplot Harga Tiket Setelah Outlier Dihapus')
plt.show()

Jumlah data sebelum hapus outlier: 775
Jumlah data setelah hapus outlier: 649
Boxplot Harga Tiket Setelah Outlier Dihapus
```

Hasil dari tahap pembersihan:

- Dataset kini berisi 596 baris yang valid.
- Tidak ada nilai hilang.
- Distribusi data lebih stabil dan bersih dari nilai ekstrem.

## 4. Data Preparation – Konstruksi Fitur, Labeling & Integrasi

Tahap ini bertujuan menambah informasi pada dataset serta mempersiapkan elemen yang diperlukan untuk analisis atau model prediksi.

Langkah-langkah yang dilakukan:

## 1. Feature Engineering (Membuat fitur baru)

### Membangun 2 Fitur Baru (Feature Engineering)

Dibuat dua fitur baru bernama "FamilySize" dan "AgeGroup". Fitur FamilySize merupakan hasil penjumlahan antara jumlah saudara/kakak/adik (SibSp) dan jumlah orang tua/anak (Parch) yang berada di kapal. Fitur ini digunakan untuk melihat pengaruh ukuran keluarga terhadap kemungkinan penumpang untuk selamat (misalnya, keluarga kecil mungkin lebih mudah diselamatkan).

Sementara itu, fitur AgeGroup dibuat dengan mengelompokkan usia penumpang ke dalam beberapa kategori, seperti Child, Teen, Adult, dan Senior, agar dapat menganalisis hubungan antara kelompok usia dengan tingkat keselamatan.

```
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1 # +1 untuk dirinya sendiri  
df['AgeGroup'] = pd.cut(df['Age'], bins=[0, 12, 18, 40, 60, 100],  
                        labels=['Child', 'Teen', 'Adult', 'Middle-aged', 'Senior'])
```

- **FamilySize** = jumlah keluarga di kapal (SibSp + Parch + 1).
- **AgeGroup** = kategori umur seperti Child, Teen, Adult, Middle-Aged, dan Senior.

Fitur ini membantu melihat apakah usia atau ukuran keluarga berpengaruh pada keselamatan.

## 2. Menentukan Label

### Menentukan Target (Label Dataset)

Target kolom Survived, yang menunjukkan apakah penumpang selamat atau tidak.

```
#Menentukan Target (Label Dataset)  
target = df['Survived']  
features = df.drop(columns=['Survived'])
```

Kolom Survived tetap digunakan sebagai variabel target untuk keperluan analisis atau machine learning.

## 3. Integrasi Dataset Tambahan

### Integrasi dengan Dataset Lain

Tidak ada dataset eksternal yang relevan, sehingga integrasi tidak dilakukan. Namun, dataset utama telah ditambah fitur baru (FamilySize) untuk memperkaya analisis.

```
# Dataset tambahan: keterangan pelabuhan  
port_info = pd.DataFrame({  
    'Embarked': ['C', 'Q', 'S'],  
    'Port_Name': ['Cherbourg', 'Queenstown', 'Southampton']  
})  
  
# Gabungkan dengan dataset utama  
df = pd.merge(df, port_info, on='Embarked', how='left')
```

Ditambahkan informasi mengenai pelabuhan keberangkatan berdasarkan kode Embarked (C, Q, S), kemudian digabungkan menggunakan fungsi merge() agar dataset memiliki konteks lokasi yang lebih detail.

Hasil dari tahap konstruksi dan integrasi:

- Total kolom bertambah menjadi 15.
- Semua data sudah lengkap tanpa missing values.
- Dua fitur baru meningkatkan wawasan analisis.

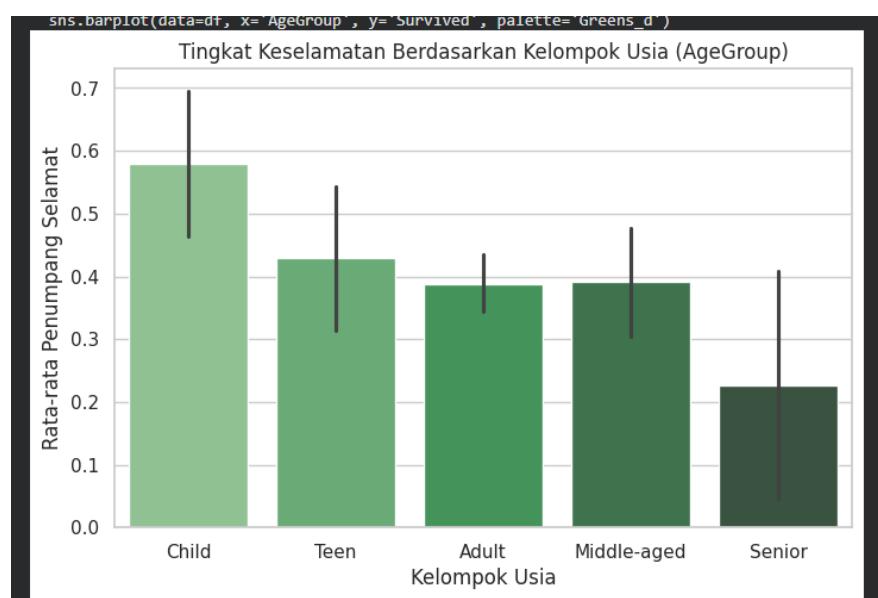
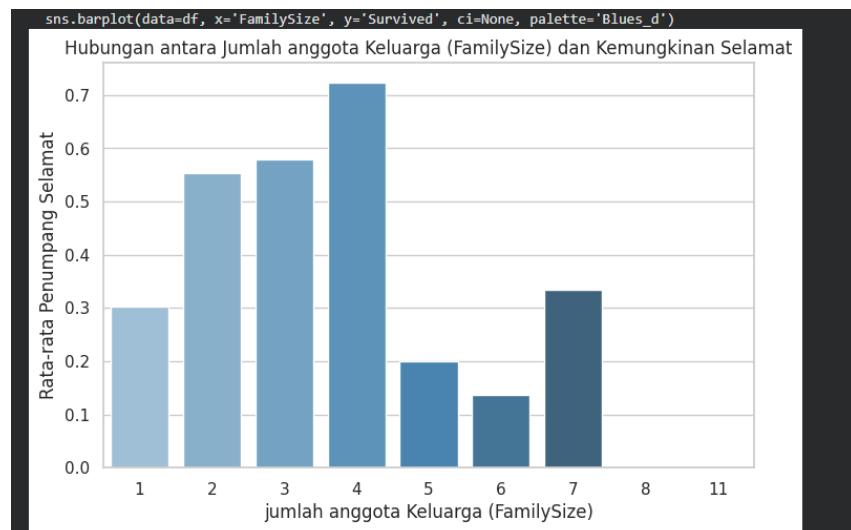
## Hasil akhir setelah konstruksi & integrasi:

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

# Atur gaya visual
sns.set(style="whitegrid")

# --- Visualisasi 1: Hubungan FamilySize dengan kemungkinan selamat ---
plt.figure(figsize=(8,5))
sns.barplot(data=df, x='FamilySize', y='Survived', ci=None, palette='Blues_d')
plt.title('Hubungan antara Jumlah anggota Keluarga (FamilySize) dan Kemungkinan Selamat', fontsize=12)
plt.xlabel('jumlah anggota Keluarga (FamilySize)')
plt.ylabel('Rata-rata Penumpang Selamat')
plt.show()

# --- Visualisasi 2: Tingkat keselamatan berdasarkan kelompok usia ---
plt.figure(figsize=(8,5))
sns.barplot(data=df, x='AgeGroup', y='Survived', palette='Greens_d')
plt.title('Tingkat Keselamatan Berdasarkan Kelompok Usia (AgeGroup)', fontsize=12)
plt.xlabel('Kelompok Usia')
plt.ylabel('Rata-rata Penumpang Selamat')
plt.show()
```



## **Kesimpulan Akhir**

Secara keseluruhan, proyek ini melibatkan proses lengkap mulai dari memahami permasalahan bisnis, mengenali struktur data, membersihkan dataset, hingga menambahkan fitur tambahan yang lebih informatif.

Dataset Titanic yang telah diproses kini siap digunakan untuk analisis lanjutan ataupun model machine learning. Hasil akhir mencakup:

- 596 baris data yang valid dan bersih.
- 15 fitur setelah integrasi dan penambahan.
- Tidak ada data kosong maupun duplikasi.
- Dua fitur baru—FamilySize dan AgeGroup—yang memberikan insight tambahan dalam analisis keselamatan penumpang.