

TITANIC DATA

Explanatory Data Analysis With Python

By: Asip Kasipul Kurob

PENDAHULUAN

Gambaran Kasus

Tragedi tenggelamnya kapal Titanic pada tahun 1912 merupakan salah satu bencana maritim paling terkenal dalam sejarah, menewaskan lebih dari 1.500 orang dari total lebih dari 2.200 penumpang dan awak kapal. Peristiwa ini memunculkan pertanyaan menarik dari sisi analisis data:

1. Faktor apa saja yang memengaruhi kemungkinan seseorang untuk selamat?
2. Apakah usia, jenis kelamin, kelas sosial, atau kombinasi dari beberapa faktor?

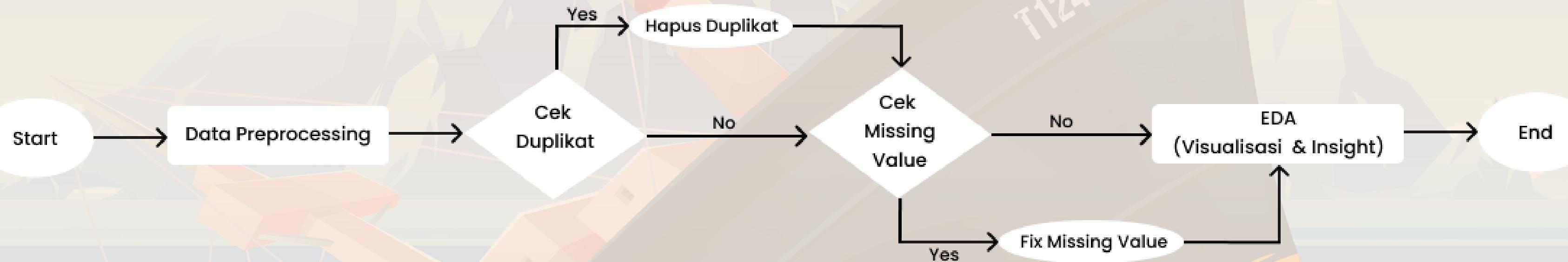
Tujuan

Proyek ini bertujuan untuk menjawab pertanyaan tersebut melalui eksplorasi data menggunakan dataset Titanic yang telah dikurasi. Proses analisis melibatkan tahapan **data preprocessing** dan **exploratory data analysis (EDA)** untuk mengidentifikasi pola-pola dan variabel-variabel yang berperan penting dalam keselamatan penumpang.

Alat yang Digunakan



FLOWCHART



DATA OVERVIEW

	survived		name	sex	age
0	1		Allen, Miss. Elisabeth Walton	female	29.0000
1	1		Allison, Master. Hudson Trevor	male	0.9167
2	0		Allison, Miss. Helen Loraine	female	2.0000
3	0		Allison, Mr. Hudson Joshua Creighton	male	30.0000
4	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	

	survived		name	sex	age
495	1	Mallet, Mrs. Albert (Antoinette Magnin)	female	24.0	
496	0	Mangiavacchi, Mr. Serafino Emilio	male	Nan	
497	0	Matthews, Mr. William John	male	30.0	
498	0	Maybery, Mr. Frank Hubert	male	40.0	
499	0	McCrae, Mr. Arthur Gordon	male	32.0	

Dataset berisi informasi penumpang Titanic, termasuk data pribadi dan status keselamatan. Analisis difokuskan pada **usia**, **jenis kelamin**, dan **status keselamatan** untuk memahami pengaruh faktor demografis terhadap peluang bertahan dalam tragedi tersebut.

Dataset bisa di download melalui link: <https://github.com/asipkk23/DigitalSkillFair38-FinalProject-DataScience>

IMPORT LIBRARY

```
[1] import numpy as np
    import pandas as pd
    import seaborn as sns
    import matplotlib.pyplot as plt
    pd.set_option("display.max_columns", None)
    pd.set_option("display.max_rows", None)
```

Library adalah kumpulan kode siap pakai yang berisi fungsi dan alat bantu untuk memudahkan kita dalam mengolah data, membuat visualisasi, atau melakukan analisis statistik. Berikut beberapa library yang digunakan:

- **NumPy**: Library untuk operasi numerik pada array dan matriks.
- **Pandas**: Library yang digunakan untuk manipulasi dan analisis data dengan struktur DataFrame dan Series.
- **Seaborn**: Library visualisasi statistik yang dibangun di atas matplotlib, tampilannya lebih rapi dan informatif.
- **matplotlib.pyplot**: Modul untuk membuat berbagai jenis grafik dan diagram.
- **pd.set_option()**: Digunakan untuk mengatur tampilan output DataFrame, seperti menampilkan semua kolom dan baris.

MENAMPILKAN DATA

data.head()

	survived		name	sex	age
0	1		Allen, Miss. Elisabeth Walton	female	29.0000
1	1		Allison, Master. Hudson Trevor	male	0.9167
2	0		Allison, Miss. Helen Loraine	female	2.0000
3	0		Allison, Mr. Hudson Joshua Creighton	male	30.0000
4	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	

data.tail()

	survived		name	sex	age
495	1	Mallet, Mrs. Albert (Antoinette Magnin)	female	24.0	
496	0	Mangiavacchi, Mr. Serafino Emilio	male	Nan	
497	0	Matthews, Mr. William John	male	30.0	
498	0	Maybery, Mr. Frank Hubert	male	40.0	
499	0	McCrae, Mr. Arthur Gordon	male	32.0	

data.head() Menampilkan 5 baris pertama dari dataset untuk memberikan gambaran awal tentang isi data.

Selain menggunakan **data.head()** dan **data.tail()**, kita juga bisa menggunakan **data.sample(n)** untuk mengambil n baris acak dari dataset.

Fungsi ini berguna untuk melihat keragaman data secara cepat dan acak.

data.tail() Menampilkan 5 baris terakhir dari dataset, berguna untuk melihat data di bagian akhir.

INFORMASI UMUM DATASET

```
[5] data.info()  
  
→ <class 'pandas.core.frame.DataFrame'>  
RangeIndex: 500 entries, 0 to 499  
Data columns (total 4 columns):  
 #   Column    Non-Null Count  Dtype     
---  --        --           --        
 0   survived  500 non-null   int64    
 1   name       500 non-null   object    
 2   sex        500 non-null   object    
 3   age        451 non-null   float64  
dtypes: float64(1), int64(1), object(2)  
memory usage: 15.8+ KB
```

Berdasarkan dataset Titanic, diperoleh informasi berikut:

- Dataset terdiri dari **500 baris** dan **4 kolom**
- 2 kolom **kategorikal**: name dan sex (tipe data: object)
- 2 kolom **numerik**: survived (int64) dan age (float64)
- Kolom age memiliki **49 nilai yang hilang**, karena hanya terdapat **451 nilai non-null** dari total **500 baris data**

HANDLING DUPLIKAT

```
[6] len(data)  
→ 500
```

Hasil dari **len(data)** menunjukkan bahwa dataset terdiri dari 500 baris data secara keseluruhan.

```
[7] len(data.drop_duplicates())  
→ 499
```

Saat **data.drop_duplicates()** diterapkan, jumlah baris berkurang menjadi 499. Hal ini menunjukkan bahwa terdapat 1 baris duplikat dalam dataset.

```
[8] #jika output dari code di cell ini tidak bernilai 1 maka terdapat duplikat  
len(data.drop_duplicates()) / len(data)  
→ 0.998
```

Hasil dari **len(data.drop_duplicates())/len(data)** adalah 0.998. Karena nilainya kurang dari 1, hal ini mengonfirmasi bahwa terdapat data duplikat dalam dataset.

HANDLING DUPLIKAT

```
[13] # Ambil baris duplikat (termasuk yang asli)
duplicates = data[data.duplicated(keep=False)]
duplicates
```

	survived	name	sex	age
104	1	Eustis, Miss. Elizabeth Mussey	female	54.0
349	1	Eustis, Miss. Elizabeth Mussey	female	54.0

```
[14] #Handling Drop duplicate
data = data.drop_duplicates()
```

```
[15] #jika output dari code di cell ini tidak bernilai 1 maka terdapat duplikat
len(data.drop_duplicates()) / len(data)
```

```
→ 1.0
```

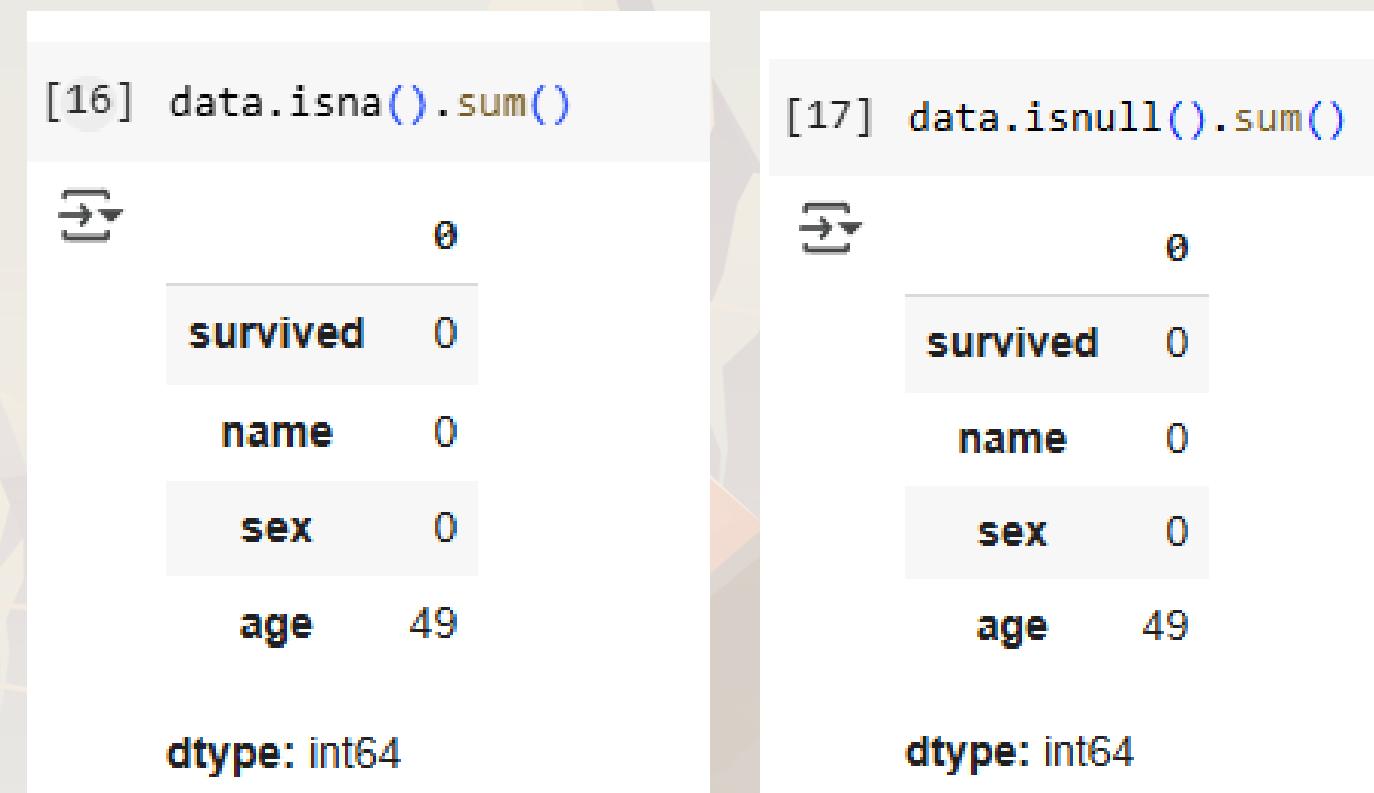
Perintah ini menampilkan semua baris yang terduplikasi, termasuk data aslinya. Hasilnya menunjukkan terdapat **2 baris identik** dalam dataset, yaitu:

- **Baris 104** dan **baris 349**, Keduanya berisi informasi yang sama.

Langkah selanjutnya adalah **menghapus baris duplikat** yang ditemukan dalam dataset untuk memastikan kualitas dan keakuratan data.

Hasil yang diperoleh adalah **1.0**, yang menunjukkan bahwa **tidak ada data duplikat yang tersisa** setelah proses pembersihan. Dengan demikian, dapat disimpulkan bahwa penanganan data duplikat telah berhasil dilakukan.

HANDLING MISSING VALUE



[16] data.isna().sum()	
survived	0
name	0
sex	0
age	49
dtype: int64	

[17] data.isnull().sum()	
survived	0
name	0
sex	0
age	49
dtype: int64	

Untuk mengecek nilai yang hilang (Missing Value) , dapat digunakan perintah:

- **data.isna().sum()** atau **data.isnull().sum()**

Hasil output menunjukkan:

- Kolom **survived**, **name**, dan **sex** tidak memiliki nilai yang hilang (**missing value= 0**)
- Kolom **age** memiliki **49 nilai yang hilang**, nilai yang hilang ini perlu ditangani agar hasil analisis lebih akurat dan tidak bias.

Dengan **asumsi** bahwa **penanganan missing value telah disetujui** oleh **atasan**, maka langkah selanjutnya adalah melakukan imputasi untuk kolom age menggunakan metode yang sesuai.

HANDLING MISSING VALUE

```
[18] # percentage version
total_rows = len(data)

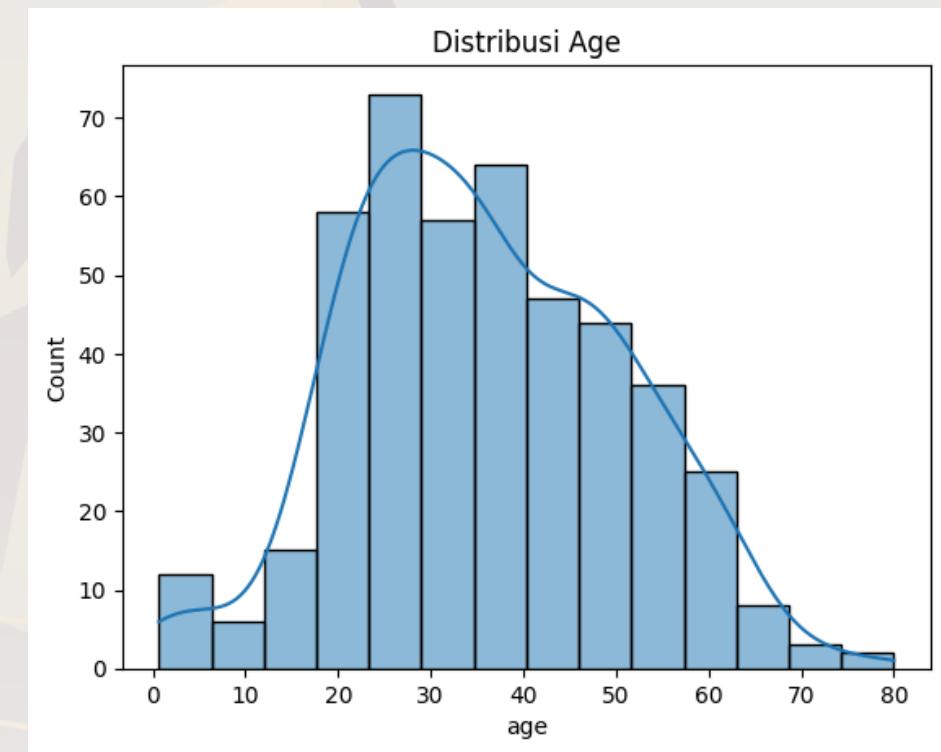
# Menghitung dan menampilkan persentase missing values di setiap kolom satu per satu
for column in data.columns:
    missing_count = data[column].isna().sum()
    missing_percentage = (missing_count / total_rows) * 100
    print(f"Column '{column}' Has {missing_count} missing values ({missing_percentage:.2f}%)") # .2f means 2 decimal

→ Column 'survived' Has 0 missing values (0.00%)
Column 'name' Has 0 missing values (0.00%)
Column 'sex' Has 0 missing values (0.00%)
Column 'age' Has 49 missing values (9.82%)
```

Dataset ini mengandung **nilai yang hilang** hanya pada kolom **age**, dengan **49 nilai yang hilang (9.82%)**. Karena tingkat missing value < 20%, maka penanganannya dilakukan sebagai berikut:

1. **Cek terlebih dahulu jenis distribusi data** untuk menentukan metode imputasi yang tepat:
 - Jika distribusi data **normal**, imputasi dapat dilakukan menggunakan **mean**.
 - Jika distribusi data **skewed** atau **ada outliers**, imputasi lebih baik menggunakan **median**.
 - Jika terdapat banyak **nilai yang terulang** dalam data numerik, **modus** bisa dipertimbangkan.
2. Kolom **age** (numerik) → Imputasi bisa menggunakan nilai mean, median atau modus
3. Kolom **sex** dan **survived** (kategorikal) → **Tidak ada missing values**, jadi tidak perlu imputasi

HANDLING MISSING VALUE



Berdasarkan visualisasi, distribusi data usia **tidak mengikuti distribusi normal**, melainkan menunjukkan pola **right-skewed (kemiringan positif)**, dengan ciri-ciri sebagai berikut:

- **Puncak distribusi** (modus) berada pada usia sekitar **25-30 tahun**.
- **Ekor distribusi memanjang ke arah kanan**, yaitu pada usia yang lebih tinggi — menandakan bahwa jumlah data pada kelompok usia tua lebih sedikit.
- **Distribusi tidak simetris**, melainkan **miring ke kanan**, yang merupakan ciri khas distribusi dengan skewness positif.

Dengan distribusi seperti ini, **median** merupakan metode imputasi yang lebih tepat dibandingkan mean, karena lebih tahan terhadap pengaruh outlier.

HANDLING MISSING VALUE

```
[29] # Imputasi hanya untuk kolom numerik dengan median
for column in data.columns:
    if data[column].dtype != 'object':
        data[column] = data[column].fillna(data[column].median())

# Cek jumlah missing value setelah diimputasi
print("\nJumlah missing value setelah imputasi:")
print(data.isnull().sum())

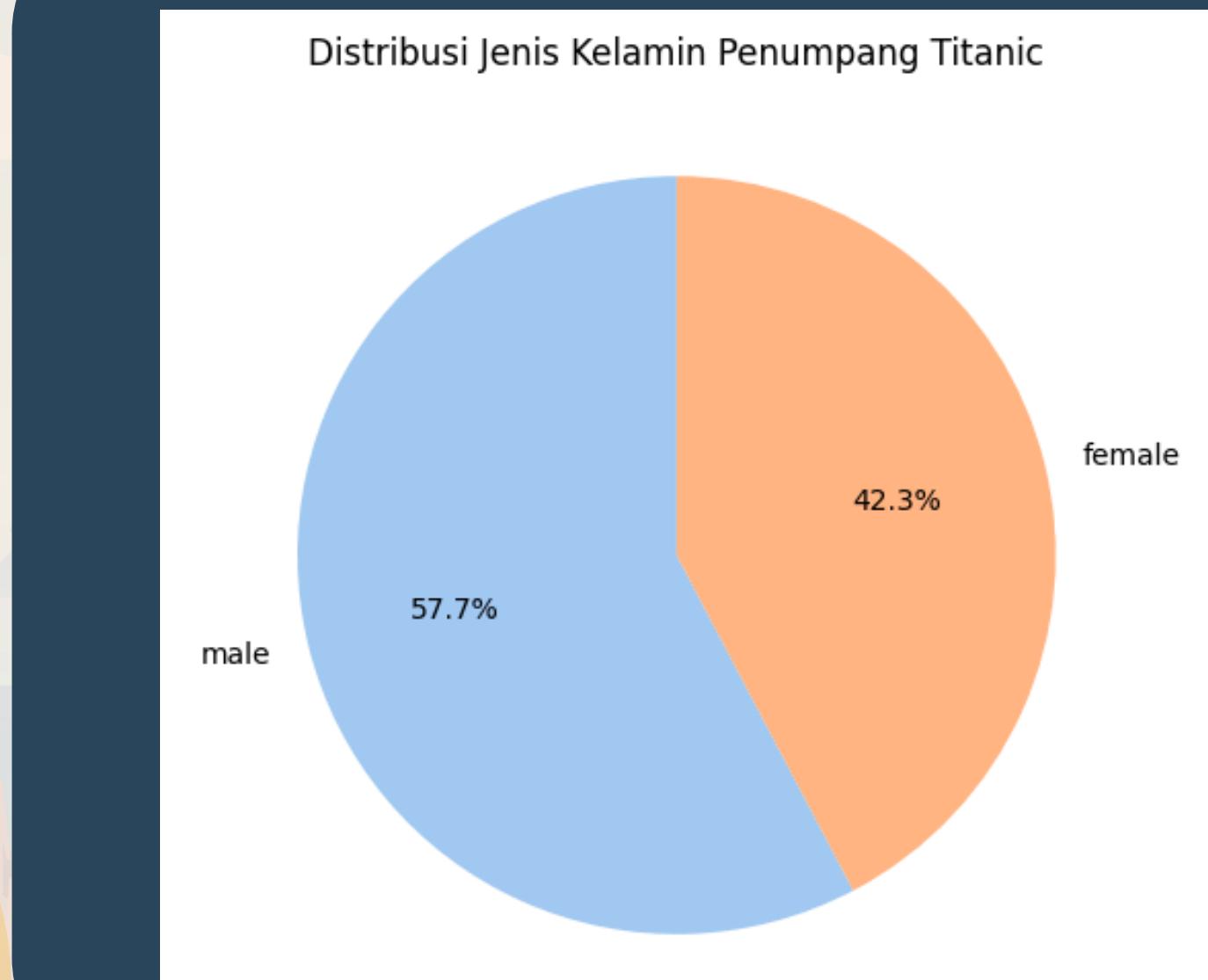
→ Jumlah missing value setelah imputasi:
survived     0
name         0
sex          0
age          0
dtype: int64
```

```
[30] data.info()

→ <class 'pandas.core.frame.DataFrame'>
Index: 499 entries, 0 to 499
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
---  --          -----          ----- 
 0   survived    499 non-null    int64  
 1   name        499 non-null    object 
 2   sex         499 non-null    object 
 3   age         499 non-null    float64 
dtypes: float64(1), int64(1), object(2)
memory usage: 19.5+ KB
```

Penanganan nilai yang hilang (missing value) dilakukan dengan metode **imputasi menggunakan median**, karena kolom yang memiliki missing value (**age**) bertipe **numerik** dan memiliki **distribusi yang tidak normal (right-skewed)**. Setelah proses imputasi dilakukan dan jumlah missing value dicek ulang, hasilnya menunjukkan bahwa **tidak ada lagi nilai yang hilang pada dataset**. Proses penanganan missing value telah berhasil diselesaikan dan dataset siap digunakan untuk proses analisis selanjutnya.

VISUALISASI DATA

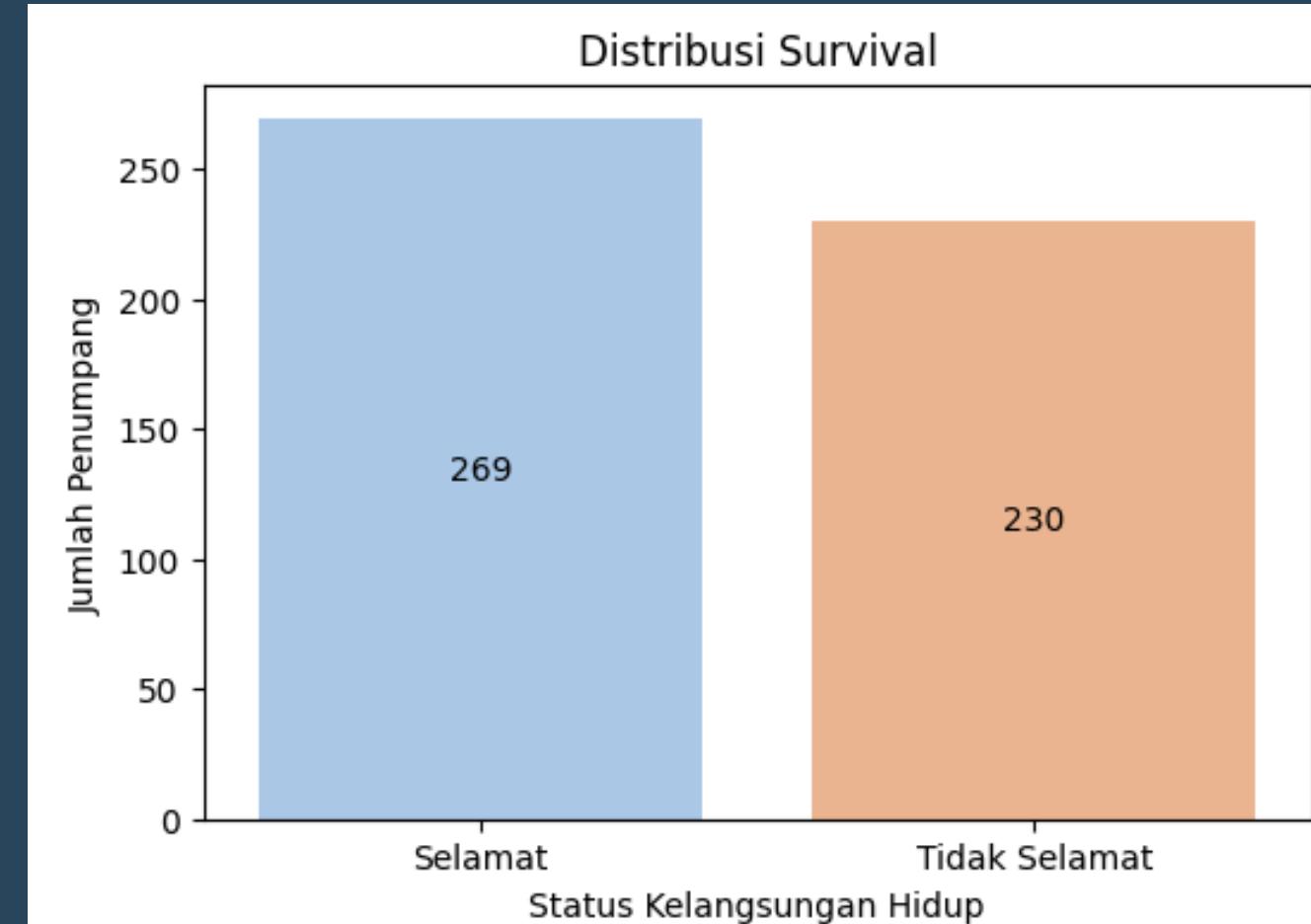


Berdasarkan visualisasi tersebut:

- **57,6%** penumpang merupakan **laki-laki**
- **42,4%** penumpang merupakan **perempuan**

Hal Ini menunjukkan bahwa jumlah penumpang **laki-laki** **lebih banyak** dibandingkan penumpang perempuan di kapal Titanic.

VISUALISASI DATA



Berdasarkan visualisasi tersebut:

- **269 penumpang selamat**
- **230 penumpang tidak selamat**

Hal ini menunjukkan bahwa jumlah penumpang yang **selamat sedikit lebih banyak**, namun masih ada sebagian besar penumpang yang tidak selamat dalam tragedi Titanic.

KESIMPULAN DAN SARAN

Kesimpulan

Faktor yang memengaruhi kemungkinan seseorang untuk selamat pada studi kasus dataset Titanic adalah usia dan jenis kelamin

Saran

Analisis ini dapat diperdalam lebih lanjut menggunakan machine learning untuk memperoleh wawasan yang lebih mendalam dan akurat.



TERIMAKASIH

CONTACT PERSON



asip.kasipul91@gmail.com



linkedin.com/in/asipkasipulkurob/