

TUGAS MATA KULIAH DATA MINING

“MODUL 2 - DATA UNDERSTANDING”



Dosen Pengampu :

Amalia Anjani Arifiyanti, S.Kom., M.Kom.

Disusun oleh :

Kevin Yohanes Wuryanto (21082010193)

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
JAWA TIMUR
2024**

DAFTAR ISI

DAFTAR ISI.....	2
HASIL DAN PEMBAHASAN.....	4
1. Penjelasan.....	4
2. Load Data dan Library.....	4
a. Import Library dan Load Dataset.....	4
b. Cek isi dataset.....	4
3. Dokumentasi Tipe Data.....	5
a. Cek jumlah data.....	5
b. Cek tipe data.....	5
c. Cek tipe data pada salah satu variabel/atribut.....	6
4. Exploratory Data Analysis (EDA).....	6
a. Pendekatan statistik sederhana (Tugas 1).....	6
i. PassengerId.....	6
ii. Survived.....	7
iii. Pclass.....	7
iv. Age.....	8
v. SibSp (jumlah saudara atau pasangan yang ikut).....	8
vi. Parch (jumlah orang tua atau anak yang ikut):.....	9
vii. Fare (harga tiket).....	9
b. Pendekatan visual.....	10
i. Distribusi data pada suatu variabel/atribut.....	10
ii. Perbandingan antar variabel/atribut (Tugas 2).....	11
1. Buat fungsi untuk menghitung dan menampilkan diagramnya.....	11
2. Stacked bar untuk menampilkan survival status berdasarkan `sex`.....	11
3. Stacked bar untuk menampilkan survival status berdasarkan `pClass`.....	12
4. Stacked bar untuk menampilkan survival status berdasarkan `Embarked`.....	13
5. Stacked bar untuk menampilkan survival status berdasarkan `ageGroup`.....	14
5. Data Cleansing untuk missing value (Tugas 3).....	15
a. Data perlu dicek apakah terdapat data yang kosong atau tidak.....	15
b. (opsional) Menunjukkan berapa persen data yang hilang dari setiap row.....	16
c. Menghapus kolom `Cabin`.....	16
d. Mengisi data (impute) `Age` menggunakan mean.....	17
e. Mengisi data `Embarked` menggunakan.....	17
f. Mengecek data null.....	17
g. Mengimport hasil data.....	18
Tugas 3. Jelaskan kelebihan dan kekurangan dari masing-masing metode	

penanganan missing value yang telah dijelaskan diatas.....	18
LAMPIRAN.....	20

HASIL DAN PEMBAHASAN

1. Penjelasan

Pada laporan ini, akan dijelaskan hasil dari eksplorasi data yang dilakukan pada dataset menggunakan berbagai metode. Tahapan yang dilakukan meliputi pemuatan data, pendokumentasian tipe data, eksplorasi data statistik dan visual, serta verifikasi kualitas data (missing values). Dataset yang digunakan adalah Eps2-Data-train.csv.

2. Load Data dan Library

a. Import Library dan Load Dataset

```
import pandas as pd
from pandas.plotting import scatter_matrix
pd.set_option('display.width', 500)
pd.set_option('display.max_columns', 100)
pd.set_option('display.notebook_repr_html', True)

from google.colab import drive
drive.mount('/content/drive')

data = pd.read_csv('/content/drive/MyDrive/DATA MINING/Eps2-Data-train.csv')
```

Mounted at /content/drive

Kode ini mempersiapkan data untuk dianalisis dan divisualisasikan menggunakan Pandas di Google Collab.

b. Cek isi dataset.

```
[11] data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Kode data.head() menampilkan 5 baris pertama dari DataFrame data. Fungsi ini berguna untuk melihat sekilas struktur data, nama kolom, dan beberapa nilai awal dalam dataset. Kode ini dapat mengubah jumlah baris yang ditampilkan dengan memasukkan angka di dalam tanda kurung, misalnya data.head(10) untuk menampilkan 10 baris pertama.

3. Dokumentasi Tipe Data

a. Cek jumlah data

```
[12] data = pd.DataFrame(data)
      print(data.shape)
```

(891, 12)

Kode tersebut mengubah variabel data menjadi DataFrame Pandas dan menampilkan dimensinya. Kode ini membantu memahami ukuran dataset dengan menampilkan jumlah baris dan kolom yang ada.

b. Cek tipe data

```
print(data.dtypes)
```

PassengerId	int64
Survived	int64
Pclass	int64
Name	object
Sex	object
Age	float64
SibSp	int64
Parch	int64
Ticket	object
Fare	float64
Cabin	object
Embarked	object
dtype:	object

Kode ini menampilkan tipe data dari setiap kolom dalam DataFrame data. `print(data.dtypes)` akan mencetak daftar kolom beserta tipe datanya masing-masing. Outputnya akan menunjukkan tipe data seperti `int64` untuk integer, `float64` untuk angka desimal, `object` untuk teks atau campuran tipe data, dan sebagainya. Informasi ini penting untuk memahami struktur data dan memilih metode analisis yang tepat.

c. Cek tipe data pada salah satu variabel/atribut

```
0s print(data['Age'].dtypes)
float64
```

Kode `print(data['Age'].dtypes)` digunakan untuk menampilkan tipe data dari kolom 'Age' dalam DataFrame data.

4. Exploratory Data Analysis (EDA)

a. Pendekatan statistik sederhana (Tugas 1)

```
0s data.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

Kode `data.describe()` menghasilkan ringkasan statistik deskriptif dari DataFrame data. Fungsi ini memberikan informasi tentang:

- Count : Jumlah nilai non-null.
- Mean : Rata-rata.
- Std : Standar deviasi.
- Min : Nilai minimum.
- 25% : Persentil ke-25 (kuartil bawah).
- 50% : Persentil ke-50 (median).
- 75% : Persentil ke-75 (kuartil atas).
- Max : Nilai maksimum.

Hasilnya akan berupa tabel yang menampilkan statistik-statistik tersebut untuk setiap kolom numerik dalam DataFrame. Fungsi ini berguna untuk mendapatkan gambaran umum tentang distribusi data dan mengidentifikasi potensi outlier. Berikut adalah penjelasan setiap kolom :

i. **PassengerId**

PassengerId	count	Ada 891 data (PassengerId).
	mean	Rata-rata nilai PassengerId adalah 446.

	std	Standar deviasi sebesar 257.35, menunjukkan penyebaran nilai yang cukup besar.
	min	Nilai minimum adalah 1.
	25%	Kuartil pertama adalah 223.5.
	50%	Nilai tengah adalah 446.
	75%	Kuartil ketiga adalah 668.5.
	max	Nilai maksimum adalah 891.

ii. Survived

Survived	count	Ada 891 data (survived).
	mean	Rata-rata penumpang yang selamat adalah 0.38 (38% dari total penumpang selamat).
	std	Standar deviasi sebesar 0.49, menunjukkan perbedaan besar antara selamat dan tidak selamat.
	min	Nilai minimum adalah 0 (tidak selamat).
	25%	25% dari penumpang tidak selamat (nilai 0).
	50%	50% dari penumpang tidak selamat (nilai 0).
	75%	75% penumpang tidak selamat, dan 25% penumpang selamat (nilai 1).
	max	Nilai maksimum adalah 1 (selamat).

iii. Pclass

Pclass	count	Ada 891 data (Pclass).
	mean	Rata-rata kelas penumpang adalah 2.31.

	std	Standar deviasi sebesar 0.83, menunjukkan penyebaran kelas penumpang.
	min	Nilai minimum adalah 1 (kelas pertama).
	25%	2 berarti 25% penumpang berada di kelas 2.
	50%	3 berarti 50% penumpang berada di kelas 3.
	75%	3 berarti 75% penumpang berada di kelas 3.
	max	Nilai maksimum adalah 3 (kelas ketiga).

iv. Age

Age	count	Ada 714 nilai usia yang tersedia (berarti ada 177 missing values).
	mean	Rata-rata usia penumpang adalah 29.7 tahun.
	std	Standar deviasi 14.52, menunjukkan variasi usia penumpang yang cukup besar.
	min	Usia termuda adalah 0.42 tahun (bayi).
	25%	Kuartil pertama, usia di bawah 25% dari penumpang, adalah 20.12 tahun.
	50%	Usia tengah adalah 28 tahun.
	75%	Kuartil ketiga adalah 38 tahun.
	max	Usia tertua adalah 80 tahun.

v. SibSp (jumlah saudara atau pasangan yang ikut)

SibSp	count	Ada 891 data (SibSp).
	mean	Rata-rata penumpang memiliki 0.52 saudara atau pasangan.
	std	Standar deviasi 1.10, menunjukkan variasi

		besar dalam jumlah anggota keluarga.
	min	Nilai minimum adalah 0 (tidak ada saudara atau pasangan).
	25%	25% penumpang tidak memiliki saudara atau pasangan.
	50%	Setengah penumpang tidak memiliki saudara atau pasangan.
	75%	75% penumpang memiliki paling banyak 1 saudara atau pasangan.
	max	Penumpang yang paling banyak memiliki 8 saudara atau pasangan.

vi. Parch (jumlah orang tua atau anak yang ikut):

Parch	count	891 data terisi (Parch).
	mean	Rata-rata penumpang memiliki 0.38 orang tua atau anak.
	std	Standar deviasi sebesar 0.81.
	min	Nilai minimum adalah 0 (tidak ada orang tua atau anak).
	25%	25% penumpang tidak memiliki orang tua atau anak yang ikut.
	50%	50% penumpang tidak memiliki orang tua atau anak.
	75%	75% penumpang memiliki paling banyak 1 orang tua atau anak.
	max	Penumpang terbanyak memiliki 6 orang tua atau anak.

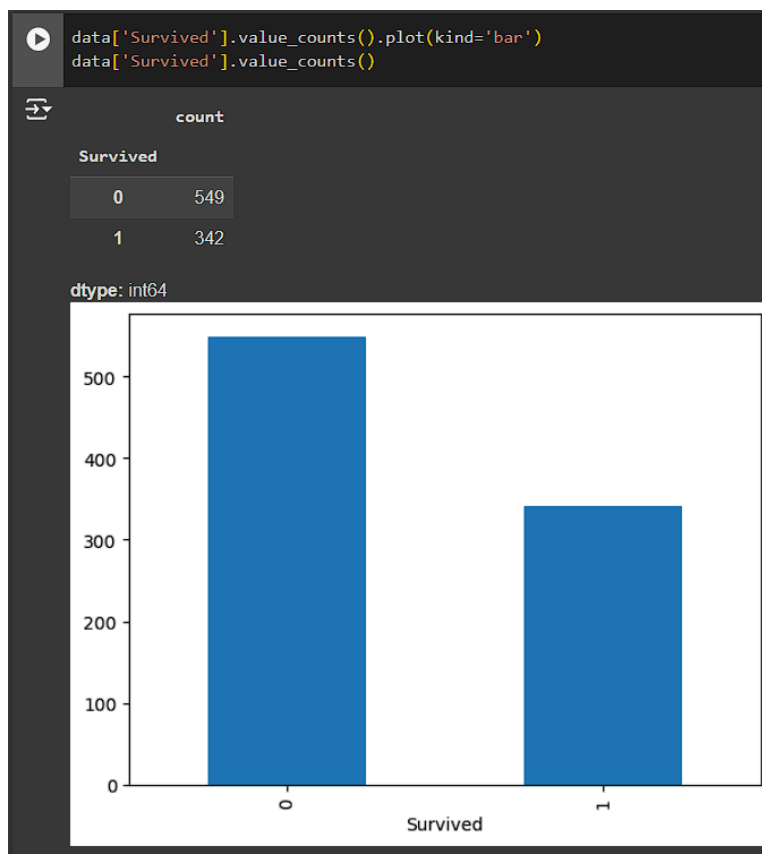
vii. Fare (harga tiket)

Fare	count	891 data terisi (Fare)
	mean	Harga rata-rata tiket adalah 32.20.

	std	Standar deviasi sebesar 49.69, menunjukkan variasi harga tiket yang signifikan.
	min	Harga tiket termurah adalah 0.
	25%	25% harga tiket dari penumpang adalah 7.91.
	50%	50% Harga tiket tengah adalah 14.45.
	75%	75% Harga tiket tengah adalah 31.00.
	max	Harga tiket termahal adalah 512.32.

b. Pendekatan visual

i. Distribusi data pada suatu variabel/atribut



Kode ini membantu memvisualisasikan distribusi antara penumpang yang selamat dan tidak selamat secara lebih mudah. Kode di atas menghasilkan diagram batang dengan dua bar. bar pertama menunjukan 549 penumpang yang tidak selamat. bar kedua menunjukan 342 penumpang yang selamat.

ii. Perbandingan antar variabel/atribut (Tugas 2)

1. Buat fungsi untuk menghitung dan menampilkan diagramnya.

```
[17] def survival_stacked_bar(data, variable):  
    # Filter data based on survival status  
    died = data[data['Survived'] == 0][variable].value_counts() / len(data[data['Survived'] == 0])  
    survived = data[data['Survived'] == 1][variable].value_counts() / len(data[data['Survived'] == 1])  
  
    # Create a DataFrame to store the results  
    dataset = pd.DataFrame([died, survived])  
    dataset.index = ['Meninggal', 'Selamat']  
  
    # Create the stacked bar chart  
    dataset.plot(kind='bar', stacked=True, title='Prosentase')  
  
    return dataset.head()
```

Fungsi ini membantu untuk memvisualisasikan proporsi survival (selamat/tidak selamat) berdasarkan variabel tertentu, seperti jenis kelamin, kelas, atau pelabuhan keberangkatan.

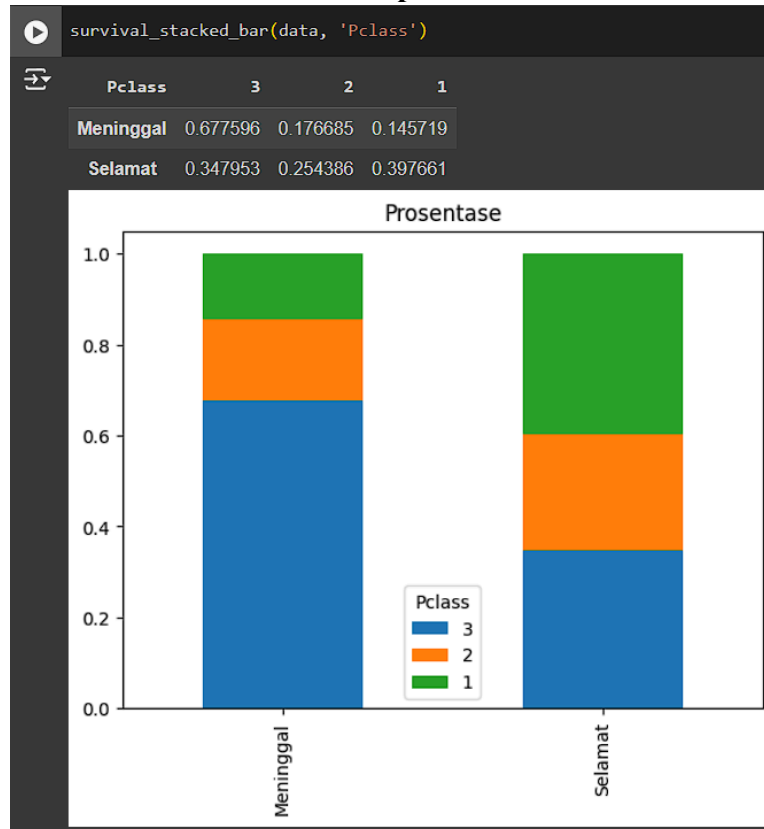
2. Stacked bar untuk menampilkan survival status berdasarkan `sex`



Kode `survival_stacked_bar(data, 'Sex')` memanggil sebuah fungsi yang membuat diagram batang bertumpuk (stacked bar chart) untuk menunjukkan status keselamatan penumpang berdasarkan jenis kelamin mereka. Hasilnya menunjukkan proporsi laki-laki dan perempuan yang meninggal atau selamat. Berikut adalah penjelasannya :

- 1) Dari mereka yang meninggal, 85,2% adalah laki-laki dan 14,8% adalah perempuan.
- 2) Dari mereka yang selamat, 31,9% adalah laki-laki dan 68,1% adalah perempuan.

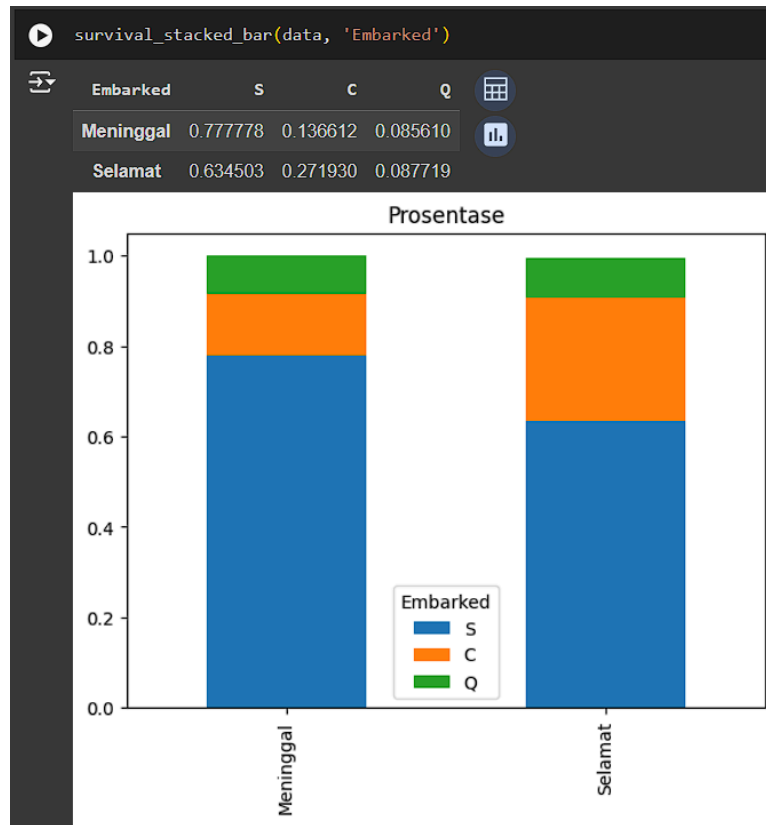
3. Stacked bar untuk menampilkan survival status berdasarkan `pClass`



Kode `survival_stacked_bar(data, 'Pclass')` memanggil sebuah fungsi yang membuat diagram batang bertumpuk (stacked bar chart) untuk menunjukkan status keselamatan penumpang berdasarkan kelas mereka (Pclass). Hasilnya menunjukkan proporsi penumpang dari kelas 1, 2, dan 3 yang meninggal atau selamat. Berikut adalah penjelasannya:

- 1) Dari mereka yang meninggal:
 - a) 67,8% berasal dari kelas 3.
 - b) 17,7% berasal dari kelas 2.
 - c) 14,6% berasal dari kelas 1.
- 2) Dari mereka yang selamat:
 - a) 34,8% berasal dari kelas 3.
 - b) 25,4% berasal dari kelas 2.
 - c) 39,8% berasal dari kelas 1.

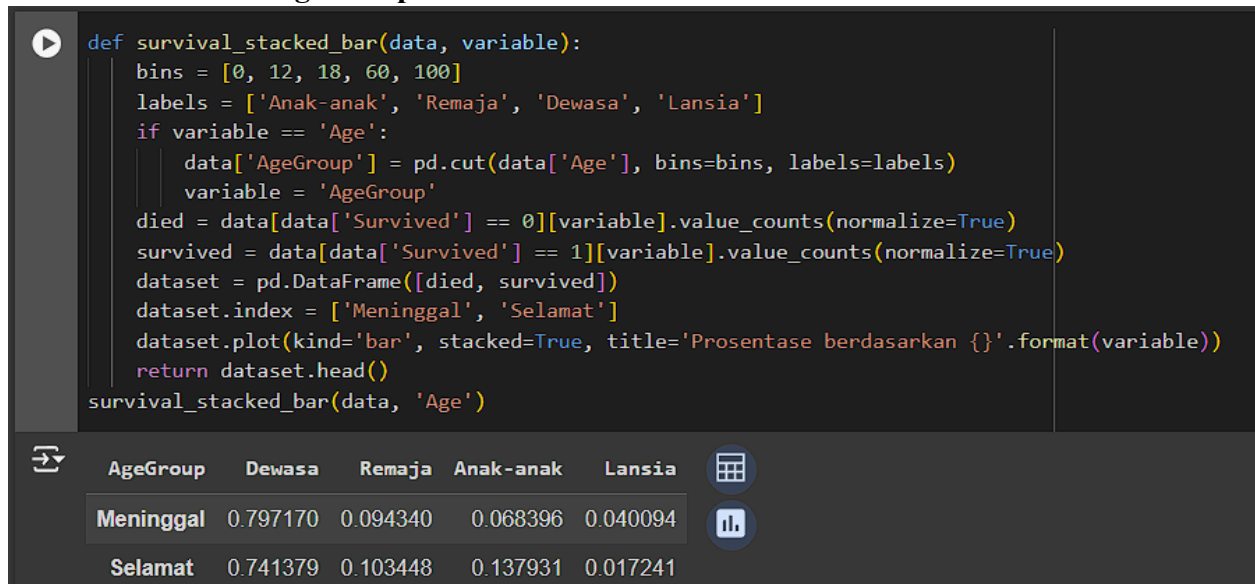
4. Stacked bar untuk menampilkan survival status berdasarkan 'Embarked'



Kode `survival_stacked_bar(data, 'Embarked')` memanggil fungsi untuk membuat diagram batang bertumpuk (stacked bar chart) yang menunjukkan status keselamatan penumpang berdasarkan pelabuhan embarkasi mereka (Embarked). Pelabuhan embarkasi diwakili oleh S (Southampton), C (Cherbourg), Q (Queenstown). Hasilnya menunjukkan proporsi penumpang dari tiap pelabuhan yang meninggal atau selamat. Berikut adalah penjelasannya:

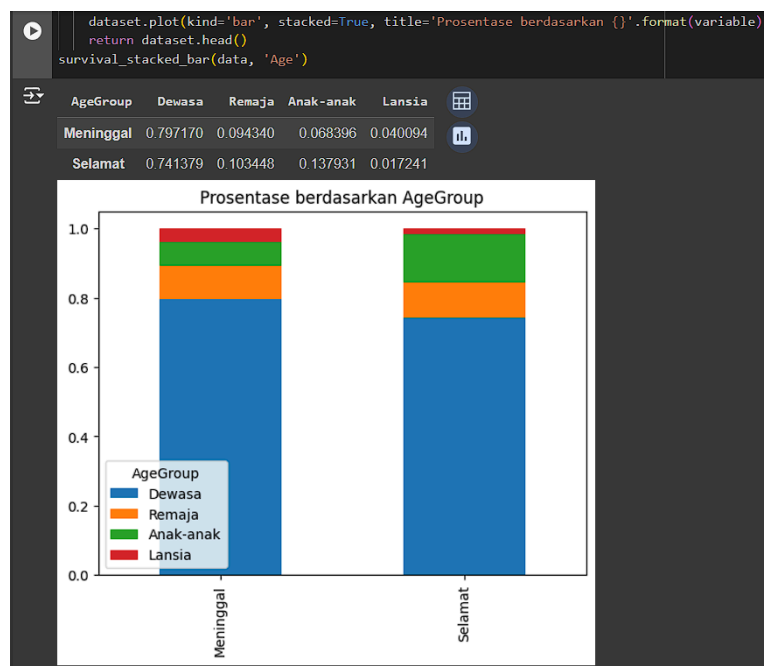
- 1) Dari mereka yang meninggal:
 - a) 77,8% berangkat dari pelabuhan Southampton (S).
 - b) 13,7% berangkat dari pelabuhan Cherbourg (C).
 - c) 8,6% berangkat dari pelabuhan Queenstown (Q).
- 2) Dari mereka yang selamat:
 - a) 63,5% berangkat dari pelabuhan Southampton (S).
 - b) 27,2% berangkat dari pelabuhan Cherbourg (C).
 - c) 8,8% berangkat dari pelabuhan Queenstown (Q).

5. Stacked bar untuk menampilkan survival status berdasarkan 'ageGroup'



Kode `survival_stacked_bar(data, 'Age')` membuat diagram batang bertumpuk (stacked bar chart) yang menunjukkan status keselamatan penumpang berdasarkan kelompok usia mereka (AgeGroup). Di sini, data usia penumpang dibagi menjadi empat kelompok:

- 1) Anak-anak (0–12 tahun)
- 2) Remaja (12–18 tahun)
- 3) Dewasa (18–60 tahun)
- 4) Lansia (60–100 tahun)

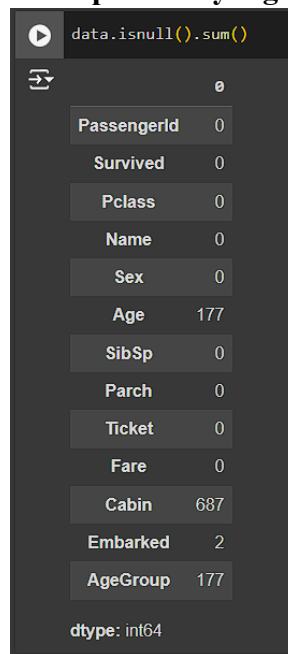


Hasilnya menunjukkan proporsi penumpang dari setiap kelompok usia yang meninggal atau selamat. Berikut adalah penjelasannya:

- 1) Dari mereka yang meninggal:
 - a) 79,7% adalah Dewasa.
 - b) 9,4% adalah Remaja.
 - c) 6,8% adalah Anak-anak.
 - d) 4,0% adalah Lansia.
- 2) Dari mereka yang selamat:
 - a) 74,1% adalah Dewasa.
 - b) 10,3% adalah Remaja.
 - c) 13,8% adalah Anak-anak.
 - d) 1,7% adalah Lansia.

5. Data Cleansing untuk missing value (Tugas 3)

a. Data perlu dicek apakah terdapat data yang kosong atau tidak.



data.isnull().sum()
0
PassengerId 0
Survived 0
Pclass 0
Name 0
Sex 0
Age 177
SibSp 0
Parch 0
Ticket 0
Fare 0
Cabin 687
Embarked 2
AgeGroup 177
dtype: int64

Berikut penjelasan hasil dari kode `data.isnull().sum()` yang digunakan untuk menghitung jumlah missing values (nilai kosong) di setiap kolom pada dataset:

- 1) PassengerId : Tidak ada nilai yang hilang (0).
- 2) Survived : Tidak ada nilai yang hilang (0).
- 3) Pclass : Tidak ada nilai yang hilang (0).
- 4) Name : Tidak ada nilai yang hilang (0).
- 5) Sex : Tidak ada nilai yang hilang (0).
- 6) Age : Ada 177 nilai yang hilang di kolom Age (usia).
- 7) SibSp : Tidak ada nilai yang hilang (0).
- 8) Parch : Tidak ada nilai yang hilang (0).
- 9) Ticket : Tidak ada nilai yang hilang (0).

- 10) Fare : Tidak ada nilai yang hilang (0).
- 11) Cabin : Ada 687 nilai yang hilang di kolom Cabin.
- 12) Embarked : Ada 2 nilai yang hilang di kolom Embarked (pelabuhan keberangkatan).
- 13) AgeGroup : Ada 177 nilai yang hilang di kolom AgeGroup, karena nilai ini bergantung pada kolom Age (nilai kosong pada kolom Age menyebabkan nilai kosong pada AgeGroup).

b. (opsional) Menunjukkan berapa persen data yang hilang dari setiap row

```
[ ] # data.isnull().mean() * 100
missing_percentage = data.isnull().mean() * 100
missing_percentage = missing_percentage.apply(lambda x: '{:.2f}%'.format(x))
print(missing_percentage)
```

PassengerId	0.00%
Survived	0.00%
Pclass	0.00%
Name	0.00%
Sex	0.00%
Age	19.87%
SibSp	0.00%
Parch	0.00%
Ticket	0.00%
Fare	0.00%
Cabin	77.10%
Embarked	0.22%
AgeGroup	19.87%
dtype:	object

Kode di atas menghitung persentase missing values di setiap kolom pada dataset. Hasilnya menunjukkan:

- 1) Kolom **Cabin** memiliki 77.10% missing values.
- 2) Kolom **Age** dan **AgeGroup** masing-masing memiliki 19.87% missing values.
- 3) Kolom **Embarked** memiliki 0.22% missing values.
- 4) Kolom lain tidak memiliki missing values (0.00%).

Ini membantu mengidentifikasi kolom mana yang perlu ditangani dalam proses data cleansing.

c. Menghapus kolom 'Cabin'

```
[17] data_cabin_cleaned = data.drop('Cabin', axis=1)
```

Kode `data = data.drop(['Cabin'], axis=1)` digunakan untuk menghapus kolom 'Cabin' dari DataFrame data. Parameter `axis=1` menunjukkan bahwa operasi ini dilakukan pada kolom (bukan baris). Penghapusan kolom 'Cabin' bisa dilakukan karena beberapa alasan:

- 1. **Banyak Data Hilang:** Jika kolom 'Cabin' memiliki banyak nilai yang hilang, kolom ini mungkin tidak memberikan informasi berguna dan bisa mengganggu analisis.
- 2. **Tidak Relevan:** Kolom 'Cabin' mungkin tidak relevan dengan tujuan analisis, seperti faktor-faktor yang mempengaruhi tingkat keselamatan.
- 3. **Menghindari Overfitting:** Terlalu banyak fitur bisa menyebabkan overfitting pada

model machine learning. Menghapus kolom yang tidak relevan dapat membantu mencegah masalah ini.

Penting untuk memastikan bahwa penghapusan kolom tersebut tidak mempengaruhi hasil analisis secara signifikan.

d. Mengisi data (impute) `Age` menggunakan mean

```
[22] mean_age = data['Age'].mean()  
      data['Age'] = data['Age'].fillna(mean_age)
```

Kode ini digunakan untuk mengisi nilai yang hilang (missing values) pada kolom 'Age' di DataFrame data dengan nilai rata-rata (mean) dari kolom tersebut.

e. Mengisi data `Embarked` menggunakan

```
[21] mode_embarked = data['Embarked'].mode()[0]  
      data['Embarked'] = data['Embarked'].fillna(mode_embarked)
```

Kode ini bertujuan untuk mengisi nilai yang hilang (missing values) pada kolom 'Embarked' dalam DataFrame data menggunakan modus (nilai yang paling sering muncul).

f. Mengecek data null

```
[25] data.isnull().sum()  
  
PassengerId    0  
Survived        0  
Pclass          0  
Name            0  
Sex             0  
Age             0  
SibSp           0  
Parch           0  
Ticket          0  
Fare            0  
Embarked        0  
AgeGroup       177  
  
dtype: int64
```

Kode `data.isnull().sum()` digunakan untuk memeriksa jumlah nilai yang hilang (null) di setiap kolom DataFrame data setelah pembersihan. Ini membantu memastikan bahwa semua nilai yang hilang telah ditangani dan memberikan gambaran tentang kolom mana yang masih memiliki nilai null.

g. Mengimport hasil data

```
[ ] from google.colab import files  
  
data.to_csv('data_train_cleaned.csv')  
  
files.download('data_train_cleaned.csv')
```

Kode ini digunakan untuk mengunduh DataFrame data yang sudah dibersihkan sebagai file CSV dari Google Colab.

Tugas 3. Jelaskan kelebihan dan kekurangan dari masing-masing metode penanganan missing value yang telah dijelaskan diatas

1. Menghapus Atribut yang Memiliki Nilai Null

Kelebihan:

- Metode ini mudah diterapkan dan cepat dilakukan, terutama jika atribut yang dihapus tidak memiliki banyak informasi penting.
- Tidak perlu melakukan perhitungan tambahan atau pengisian nilai yang mungkin memperkenalkan bias.

Kekurangan:

- Menghapus atribut dapat menghilangkan informasi berharga yang mungkin relevan dengan analisis atau model.
- Jika banyak atribut yang dihapus, data bisa menjadi terlalu sedikit, yang dapat mempengaruhi kualitas analisis.

2. Menghapus Baris yang Memiliki Nilai Null

Kelebihan:

- Memudahkan analisis karena semua data yang digunakan sudah bersih dari nilai null.
- Tidak memperkenalkan bias yang bisa terjadi jika nilai null diisi dengan nilai lain.

Kekurangan:

- Menghapus baris dapat menyebabkan hilangnya banyak data, terutama jika nilai null terdapat di banyak baris.
- Baris yang dihapus mungkin berisi informasi yang berharga.

3. Mengisi dengan Nilai Baru

a. Mengisi Manual

Kelebihan:

- Memungkinkan pengisian nilai yang sesuai dengan konteks spesifik dari data.
- Memberikan kontrol penuh atas nilai yang dimasukkan, memungkinkan penyesuaian berdasarkan pengetahuan domain.

Kekurangan:

- Memerlukan waktu dan usaha untuk menentukan nilai pengganti yang tepat.
- Risiko pengenalan bias atau kesalahan manusia dalam penentuan nilai pengganti.

b. Mengisi dengan Metode Imputation (Mean, Median, Modus, dll)

Kelebihan:

- Mempermudah proses pengisian nilai tanpa perlu mengisi manual satu per satu.
- Metode imputation seperti mean atau median memberikan pendekatan yang konsisten dan dapat diulang.

Kekurangan:

- Pengisian dengan mean, median, atau modus dapat mempengaruhi distribusi data dan mungkin tidak mencerminkan variasi asli.
- Metode ini bisa menghilangkan informasi penting yang terkait dengan nilai yang hilang, terutama jika data tidak terdistribusi normal.

LAMPIRAN

LINK GOOGLE COLAB :

<https://colab.research.google.com/drive/1t4cSh-ZT4urCfRnKhhkJFgPonRLGBEhm?usp=sharing>

LINK GITHUB :

<https://github.com/yohaneskohl/Eps2---Data-Understanding>