Tools and Technique for Computational Analysis

Case Study: Titanic Passenger Survival Analysis

Disusun oleh:

Nama: Isadora Bougenville
NPM. 24240000000

MAGISTER TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
IIB DARMAJAYA
BANDAR LAMPUNG

2025

Analisis Kelangsungan Hidup Penumpang Titanic

Dalam menganalisis kelangsungan hidup penumpang Titanic, penulis menggunakan beberapa tools dimana kode program dan data didapat pada https://www.kaggle.com/code/ashishpatel26/titanic-passenger-survival-analysis/notebook, selanjutnya penulis menggunakan Google Colab dalam running program sehingga dapat melihat interpretasi dan identifikasi dengan baik.

1. Defining the problem statement

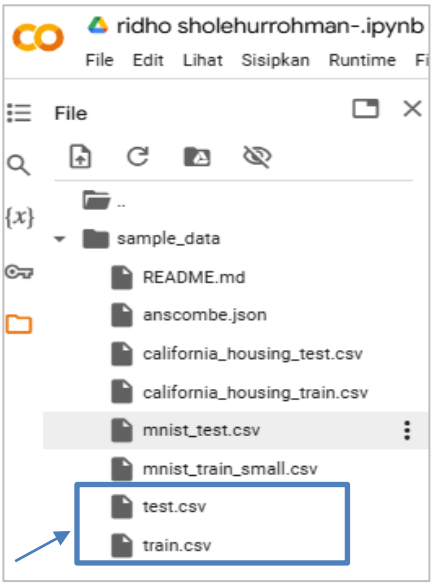
Sebelum memulai proses, tahap pertama adalah definisikan pernyataan masalah dengan mengindetifikasi penumpang yang akan selamat dimana definisi tersebut mempunyai faktor-faktor yang memengaruhi kemungkinan seseorang selamat dari tragedi Titanic.

Diberikan, contoh gambar kapal Titanic pada url sebaga berikut.

```
from IPython.display import Image
Image(url=     "https://static1.squarespace.com/static/5006453fe4b09ef2252ba068/5095eabce4b06cb305058603/5095eabce4b02d37bef4c24c/1352002236895/100_anniversary_titanic_sinking_by_esai8mellows-d4xbme8.jpg")
```

Gambar 1. Gambar Kapal Titanic

Selanjutnya, impor data File CSV-traint dan CSV.test pada sample collab, agar dapat diidentifikasi dan analisis.



Gambar 2. Gambar impor data File CSV-traint dan CSV.test

CSV merupakan format file pada Python/collab untuk mentransfer dan menyimpan data. Selanjutnya diberikan sourse program untuk melihat informasi data yang terdapat di file tersebut.

```
[3]  import os
     import pandas as pd
     import numpy as np

     # print(os.listdir("../input"))
     train = pd.read_csv('sample_data/train.csv')
     test = pd.read_csv('sample_data/test.csv')
     # Any results you write to the current directory are saved as output.
```

```
train.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

Gambar 3. Source Program dan Output Sample Data

Tabel yang diberikan mencakup data tentang penumpang Titanic dengan beberapa atribut seperti ID penumpang, status kelangsungan hidup, kelas, nama, jenis kelamin, usia, jumlah saudara/saudari, jumlah orang tua/anak, tiket, tarif, kabin, dan pelabuhan naik. Dalam data ini, terdapat penumpang dari berbagai kelas, dengan beberapa penumpang yang bepergian di kelas pertama dan sebagian besar di kelas ketiga. Kelangsungan hidup penumpang menunjukkan bahwa lebih banyak penumpang wanita yang selamat dibandingkan pria. Usia penumpang bervariasi, dengan rentang usia antara 22 hingga 38 tahun, dan mayoritas penumpang tidak bepergian dengan orang tua atau anak, meskipun beberapa penumpang memiliki saudara atau saudari yang bepergian bersama mereka. Tarif yang dibayar untuk tiket juga bervariasi, dengan beberapa penumpang membayar tarif tinggi, terutama di kelas pertama. Beberapa

penumpang memiliki informasi kabin, sementara yang lain tidak. Berdasarkan pelabuhan naik, terdapat penumpang yang naik dari pelabuhan S dan C, dengan lebih sedikit yang naik dari pelabuhan Q. Secara keseluruhan, data ini menunjukkan variasi dalam profil penumpang Titanic dan memberikan gambaran tentang faktor-faktor yang mungkin mempengaruhi kelangsungan hidup penumpang.

Untuk mengetaui

```
train.isnull().sum()
print("Train Shape:",train.shape)
test.isnull().sum()
print("Test Shape:",test.shape)
```

```
Train Shape: (891, 12)
Test Shape: (418, 11)
```

Gambar 4. Source Program dan Output Train & Test

Untuk mengetahui informasi data yang ditraining, diberikan source program berikut.

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Gambar 5. Source Program dan Output Infromasi Training

Dalam gambar 5, diberikan data tentang 891 penumpang dengan 12 kolom, yang masing-masing memiliki karakteristik berbeda. Kolom-kolom tersebut mencakup ID penumpang (PassengerId), status kelangsungan hidup (Survived), kelas tiket (Pclass), nama (Name), jenis kelamin (Sex), umur (Age), jumlah saudara/istri yang bepergian bersama (SibSp), jumlah orang tua atau anak yang bepergian bersama (Parch), nomor tiket (Ticket), tarif yang dibayar (Fare), nomor kabin (Cabin), dan pelabuhan keberangkatan (Embarked). Tipe data yang digunakan bervariasi, dengan sebagian besar kolom bertipe int64 dan object, serta dua kolom bertipe float64 untuk data numerik seperti Age dan Fare. Namun, ada beberapa isu terkait data yang hilang. Kolom Age memiliki 177 nilai kosong (sekitar 20% dari total data), sementara kolom Cabin memiliki banyak nilai kosong dengan hanya 204 entri yang terisi. Kolom Embarked memiliki 2 nilai kosong, sementara kolom Survived, Pclass, SibSp, Parch, Ticket, dan Fare tidak mengandung nilai kosong sama sekali. Penanganan terhadap nilai kosong ini penting, misalnya dengan mengisi nilai kosong pada Age menggunakan median atau rata-rata, serta menangani kolom

Cabin yang memiliki banyak data hilang—mungkin dengan menghapusnya atau menganggapnya sebagai kategori "missing". Selain itu, kolom Embarked dengan dua nilai kosong dapat diisi dengan kategori mayoritas.

Untuk mengetahui informasi data yang ditraining, diberikan source program berikut.

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Pclass       418 non-null    int64
 2   Name         418 non-null    object
 3   Sex          418 non-null    object
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object
 10  Embarked     418 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

Gambar 6. Source Program dan Output Infromasi Testing

Dalam informasi, berisi data tentang 418 penumpang dengan 11 kolom yang memiliki berbagai tipe data dan beberapa nilai kosong. Kolom-kolom tersebut sama dengan traning namun tidak memiliki data status kelangsungan hidup (Survived), karena akan menjadi prediksi (y). Untuk mengetaui status tersebut, terdapat beberapa masalah terkait data yang hilang. Kolom Age memiliki 86 nilai kosong, sekitar 21% dari total data, yang perlu diatasi dengan metode imputasi, seperti mengisi dengan median atau rata-rata usia. Kolom Fare hanya memiliki satu nilai kosong, yang dapat dengan mudah diatasi dengan imputasi sederhana. Sementara itu, kolom Cabin memiliki banyak nilai kosong (327 data kosong), yang menunjukkan bahwa banyak penumpang tidak tercatat kabinnya. Kolom ini mungkin perlu dihapus atau diberi label "missing" jika dianggap relevan untuk analisis. Kolom Embarked tidak memiliki nilai kosong, sehingga data ini lengkap dan tidak memerlukan penanganan lebih lanjut. Secara keseluruhan, data ini tampaknya berasal dari dataset Titanic, di mana analisis lebih lanjut dapat dilakukan untuk mengeksplorasi faktor-faktor yang memengaruhi kelangsungan hidup penumpang, seperti kelas tiket, jenis kelamin, umur, tarif, dan pelabuhan keberangkatan. Penanganan terhadap nilai kosong di kolom Age dan Cabin sangat penting untuk mempersiapkan data sebelum digunakan dalam analisis lebih lanjut atau pembuatan model prediktif.

2. Data Dictionary

Pada tahap ini, kita dapat melihat bahwa dataset memiliki 12 kolom. Selanjutnya, diberikan soruce program sebagai berikut:

```
train.head(10)
```

1 to 10 of 10 entries Filter

index	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	7.25	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	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	7.925	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.075	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708	NaN	C

Show 100 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

Gambar 7. Source Program dan Output Infromasi training(10)

Dari hasil di atas, dijelaskan bahwa dataset Titanic memiliki 10 kolom, yaitu PassengerId, Survived, Pclass, Name, Sex, Age, SibSp, Parch, Ticket, Fare, Cabin, dan Embarked. Kolom Survived menunjukkan bahwa 5 penumpang tidak selamat (0) dan 5 penumpang selamat (1). Pada kolom Pclass (kelas tiket), ditemukan 6 penumpang dengan kelas tiket 3, 1 penumpang dengan kelas tiket 2, dan 3 penumpang dengan kelas tiket 1. Kolom Sex menunjukkan jumlah penumpang laki-laki dan perempuan masing-masing adalah 5. Sedangkan pada kolom Embarked (pelabuhan naik kapal), terdapat 7 penumpang yang naik dari Southampton (S), 2 penumpang dari Cherbourg (C), dan 1 penumpang dari Queenstown (Q). Dataset ini memuat informasi yang lebih lengkap, termasuk usia, jumlah saudara/saudari, orang tua/anak yang ikut, nomor tiket, tarif, dan nomor kabin, meskipun beberapa data hilang, seperti pada kolom Age dan Cabin. Dataset ini berguna untuk berbagai analisis, seperti memprediksi kelangsungan hidup penumpang berdasarkan faktor-faktor tersebut dan juga memerlukan penanganan lebih lanjut untuk data yang hilang dan pengkodean data kategorikal.

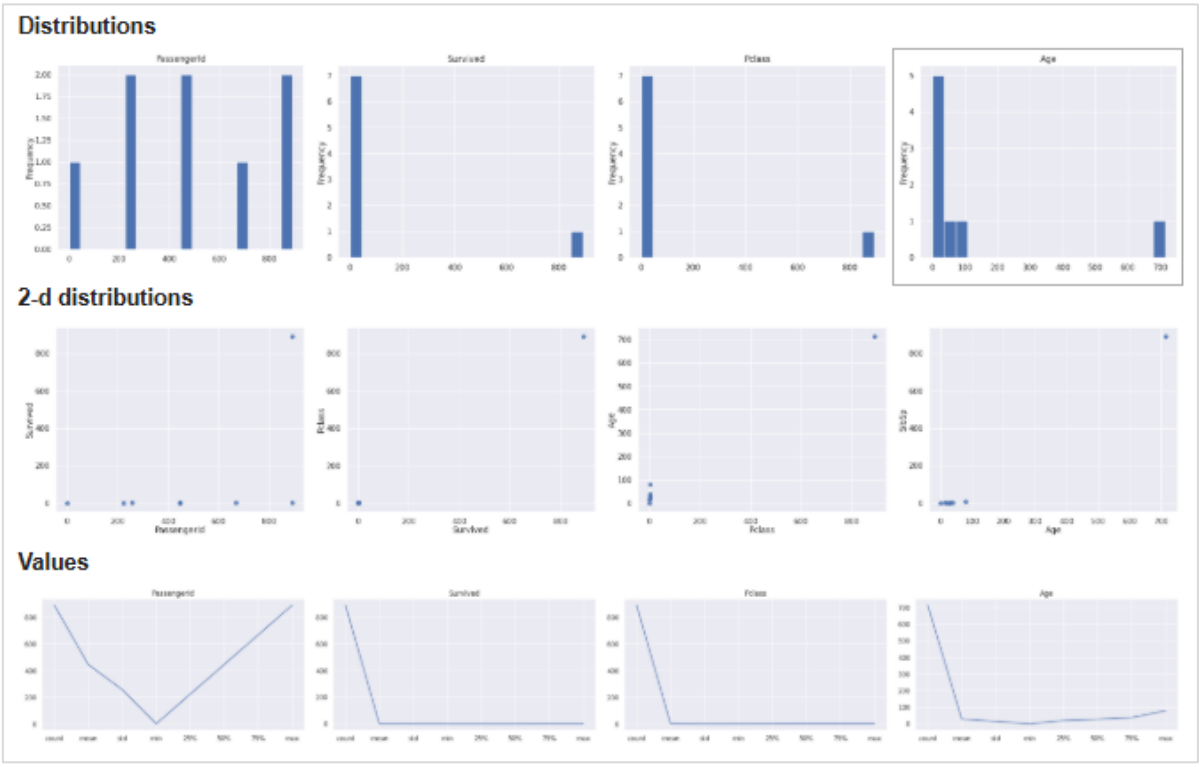
Setelah melakukan perintah diatas selanjutnya kita melakukan perintah train.describe() sehingga didapat hasil seperti dibawah ini.

```
train.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

Gambar 8. Source Program dan Output Infromasi train.describe()

Berdasarkan Gambar 8, kita memperoleh informasi bahwa PassengerId memiliki total 891 data dengan nilai rata-rata (mean) sebesar 446, nilai minimum (min) 1, dan nilai maksimum (max) 891. Sedangkan untuk kolom Survived, nilai rata-rata (mean) adalah 0,38, yang menunjukkan persentase penumpang yang selamat sekitar 38%. Nilai minimum (min) untuk kolom Survived adalah 0 (penumpang yang tidak selamat), dan nilai maksimum (max) adalah 1 (penumpang yang selamat). Data ini memberi gambaran tentang distribusi kelangsungan hidup penumpang serta identifikasi terhadap pola kelangsungan hidup dan faktor-faktor lain yang mempengaruhi. Sebagai bahan



Gambar 9. Source Program dan Output Infromasi deskripsi data Training

Berdasarkan gambar 8 dan 9, kita bisa menghubungkan statistik deskriptif tersebut dengan distribusi data. PassengerId menunjukkan bahwa dataset terdiri dari 891 penumpang dengan ID yang terurut dari 1 hingga 891, dengan nilai rata-rata sekitar 446. Sedangkan pada kolom Survived, rata-rata 0,38 berarti sekitar 38% penumpang selamat, sementara sisanya tidak selamat, yang tercermin dari nilai minimum 0 dan maksimum 1. Ini memberikan gambaran bahwa mayoritas penumpang dalam dataset ini tidak selamat. Hubungan antara PassengerId dan Survived bisa menjadi dasar untuk menganalisis faktor-faktor yang mempengaruhi kelangsungan hidup penumpang berdasarkan variabel lainnya, seperti Pclass, Age, dan Embarked.

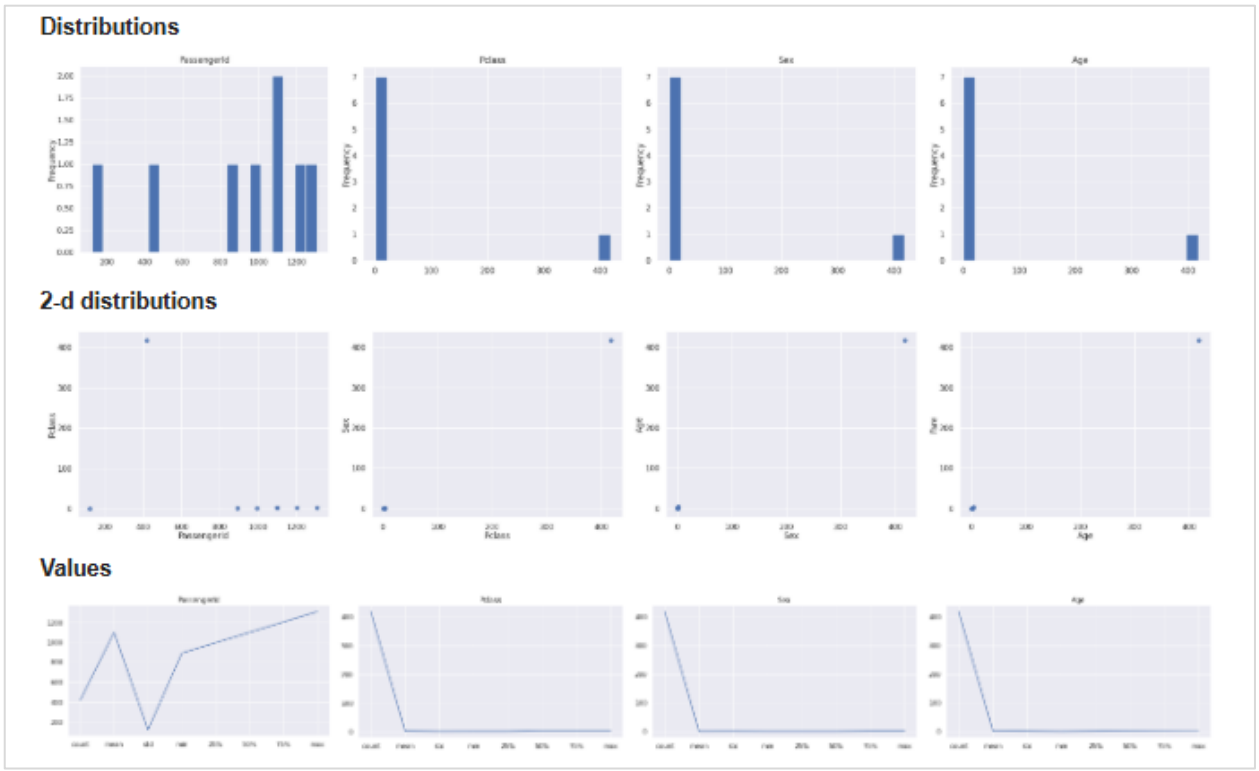
Selanjutnya diebrikan source program untuk mendeskripsikan data test seperti dibawah ini :

```
test.describe()
```

	PassengerId	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	3.000000	76.000000	8.000000	9.000000	512.329200

Gambar 10. Source Program dan Output Infromasi test.describe()

Berdasarkan Gambar 10, kita dapat melihat statistik deskriptif untuk PassengerId, yang mencakup jumlah item, rata-rata (mean), standar deviasi (std), nilai minimum (min), kuartil ke-25 (25%), median atau kuartil ke-50 (50%), kuartil ke-75 (75%), dan nilai maksimum (max). Pada kolom PassengerId, rata-rata (mean) adalah 1100,500, yang menunjukkan bahwa ID penumpang memiliki distribusi yang lebih tinggi dari nilai 446 yang terlihat sebelumnya. Nilai maksimum (max) adalah 1309, yang lebih besar dari jumlah baris data (891), yang bisa menunjukkan adanya nilai tambahan atau ID yang terpisah dari data utama. Hal ini juga mengindikasikan bahwa mungkin ada kesalahan atau kejanggalan dalam data yang perlu diperiksa lebih lanjut untuk memastikan konsistensi dan akurasi dalam dataset.



Gambar 11. Source Program dan Output Infromasi deskripsi data testing

Berdasarkan Gambar 10 dan Gambat 11, kita dapat melihat adanya perbedaan signifikan dalam statistik PassengerId. Sebelumnya, kita melihat bahwa PassengerId dalam dataset Titanic memiliki 891 data dengan rata-rata 446, yang menunjukkan distribusi ID penumpang yang terurut dari 1 hingga 891. Namun, pada Gambar 10, rata-rata PassengerId yang dihitung menjadi 1100,500, dengan nilai maksimum mencapai 1309, yang lebih tinggi dari jumlah total penumpang (891) pada dataset aslinya. Perbedaan ini bisa menunjukkan adanya masalah atau inkonsistensi dalam pengolahan data, seperti adanya duplikasi atau ID yang tidak sesuai dengan jumlah penumpang yang sebenarnya (misalnya penumpang dengan ID lebih dari 891). Kejanggalan ini perlu diteliti lebih lanjut agar dataset bisa diperbaiki dan digunakan dengan lebih tepat untuk analisis kelangsungan hidup penumpang Titanic.

Dalam melihat kualitas data, mengidentifikasi kolom dengan nilai hilang, dan menentukan langkah-langkah pembersihan data yang tepat, digunakan soruce seperti imputasi atau penghapusan data sebagai berikut.

```
train.isnull().sum()
```

	0
Survived	0
Pclass	0
Sex	0
Age	0
Fare	0
Cabin	0
Embarked	0
Title	0
FamilySize	0

dtype: int64

```
test.isnull().sum()
```

	0
PassengerId	0
Pclass	0
Name	0
Sex	0
Age	86
SibSp	0
Parch	0
Ticket	0
Fare	1
Cabin	327
Embarked	0

dtype: int64

(a) train.isnull().sum() (b) test.isnull().sum()

Gambar 12. Source Program dan Output mengidentifikasi kolom dengan nilai hilang

Hasil dari *train.isnull().sum()* menunjukkan bahwa tidak ada nilai hilang pada kolom-kolom dalam dataset train. Semua kolom, termasuk PassengerId, Survived, Pclass, Age, Fare, Cabin, Embarked, Title, dan FamilySize, memiliki nilai lengkap tanpa missing values, sehingga dataset siap untuk analisis lebih lanjut. Selanjutnya berdasarkan output *test.isnull().sum()*, terdapat 86 nilai hilang pada kolom Age, 1 nilai hilang pada Fare, dan 327 nilai hilang pada Cabin. Kolom lainnya, seperti PassengerId, Pclass, Name, Sex, SibSp, Parch, Ticket, dan Embarked, tidak memiliki nilai hilang. Kolom dengan nilai hilang perlu ditangani, seperti imputasi untuk Age dan Fare, serta mempertimbangkan penghapusan atau pengolahan khusus untuk kolom Cabin.

3. Data Visualization using Matplotlib and Seaborn packages

Untuk mempersiapkan visualisasi data dengan Matplotlib dan Seaborn. Digunakan soure %matplotlib inline memastikan grafik ditampilkan di dalam collab, sedangkan sns.set() mengatur gaya default Seaborn agar grafik lebih menarik dan konsisten. Selanjutnya, Fungsi bar_chart(feature) digunakan untuk membuat grafik batang yang membandingkan jumlah penumpang yang selamat dan tidak selamat berdasarkan nilai unik suatu fitur (kolom) pada dataset train.

```
[82] import matplotlib.pyplot as plt # Plot the graphes
     %matplotlib inline
     import seaborn as sns
     sns.set() # setting seaborn default for plots
```

```
[83] def bar_chart(feature):
         survived = train[train['Survived']==1][feature].value_counts()
         dead = train[train['Survived']==0][feature].value_counts()
         df = pd.DataFrame([survived,dead])
         df.index = ['Survived','Dead']
         df.plot(kind='bar',stacked=True, figsize=(10,5))
```

Gambar 13. Source Program %matplotlib dan bar_chart(feature)

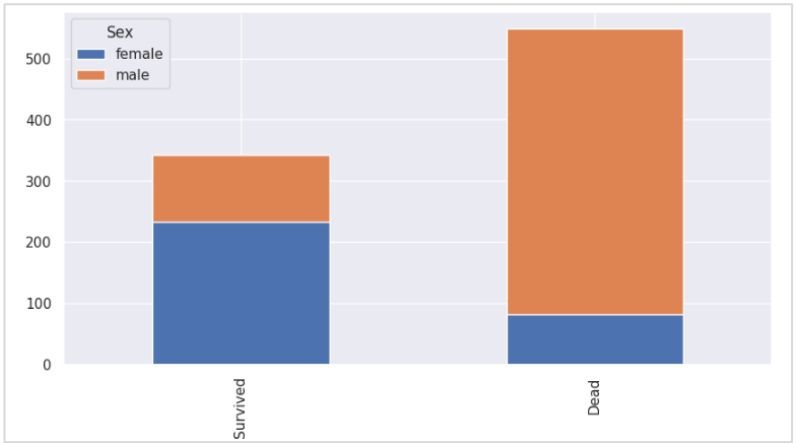
Selanjutnya, dalam mevisualisasi Bagan Batang untuk Fitur Kategorikal, Pclass, Seks, SibSp (# saudara kandung dan pasangan), Parch (# orang tua dan anak). Diberikan source program Seks sebagai berikut.

```
bar_chart('Sex')
print("Survived :\n",train[train['Survived']==1]['Sex'].value_counts())
print("Dead:\n",train[train['Survived']==0]['Sex'].value_counts())
```

```
Survived :
 Sex
female    233
male      109
Name: count, dtype: int64
Dead:
 Sex
male      468
female     81
Name: count, dtype: int64
```

Gambar 14. Source Program dan Output mengidentifikasi bagan Seks

Dari data diatas kita mendapatkan jumlah penumpang bertahan/survived Wanita233 dan Laki- Laki109 sedangkan jumlah penumpang mati/Dead Wanita81 dan Laki-Laki468. Selanjutnya, Grafik Visualisasi penumpang kapal titanic sebagi berikut



Gambar 15. Grafik Visualisasi penumpang Titanic

Diagram diatas menjelaskan bahwa data wanita pada survived mendominasi lebih banyak daripada data Laki-Laki sedangkan pada Dead data Laki-Laki lebih banyak daripada Wanita sehingga dapat dikatakan bahwa Wanita lebih mungkin bertahan daripada male.

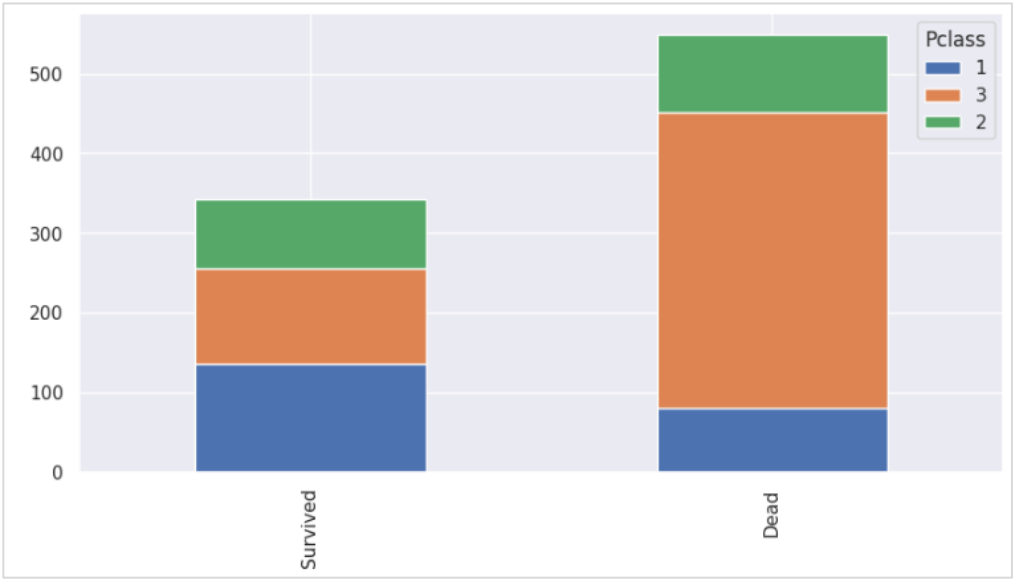
Selanjutnya, untuk memvisualisasikan distribusi kelas tiket di antara penumpang yang selamat dan yang tidak selamat digunakan source sebagai berikut.

```
bar_chart('Pclass')
print("Survived :\n",train[train['Survived']==1]['Pclass'].value_counts())
print("Dead:\n",train[train['Survived']==0]['Pclass'].value_counts())
```

```
Survived :
 Pclass
1    136
3    119
2     87
Name: count, dtype: int64
Dead:
 Pclass
3    372
2     97
1     80
Name: count, dtype: int64
```

Gambar 16. Source Program dan Output mengidentifikasi bagan distribusi Pclas

Berdasarkan Gambar 16, menunjukkan bahwa penumpang dengan kelas survived 1 sebanyak 136,kelas survived 2 sebanyak 87 dan kelas survived 3 sebanyak 119 sehingga ini menegaskan bahwa kelas 1 lebih mungkin bertahan daripada kelas lainnya dan menegaskan bahwa kelas 3 lebih mungkin mati daripada kelas lainnya. Selanjutnya untuk dapat menganalisis dengan baik, diberikan grafik visualisasi distribusi kelas tiket penumpang kapal titanic sebagi berikut.



Gambar 17. Grafik Visualisasi Distribusi Kelas Tiket Penumpang Kapal Titanic

Berdasarkan Gambar 17, dapat dilihat bahwa dead pada kelas 3 lebih banyak dari kelas 1 dan kelas 2 dan pada survived kelas 1 lebih banyak dari kelas 2 dan kelas 3.

Selanjutnya diberikan analisis hubungan antara jumlah saudara/saudari atau pasangan (SibSp) pada data

penumpang Titanic berikut.

```
bar_chart('SibSp')
print("Survived :\n",train[train['Survived']==1]['SibSp'].value_counts())
print("Dead:\n",train[train['Survived']==0]['SibSp'].value_counts())
```

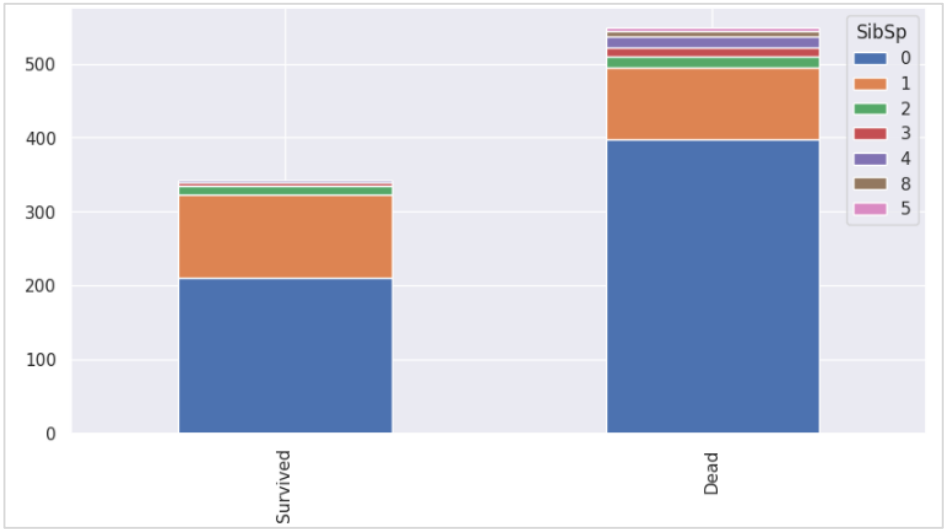
```
Survived :
 SibSp
0    210
1    112
2     13
3      4
4      3
Name: count, dtype: int64
Dead:
 SibSp
0    398
1     97
4     15
2     15
3     12
8      7
5      5
Name: count, dtype: int64
```

Gambar 18. Source Program dan Output (SibSp)

Dalam Gambar 18, menunjukan distribusi SibSp menunjukkan jumlah penumpang berdasarkan jumlah saudara/saudari atau pasangan yang mereka bawa:

- Survived = 1: Menunjukkan jumlah penumpang yang selamat, dibagi berdasarkan jumlah SibSp (saudara/saudari atau pasangan). Misalnya, jumlah penumpang yang selamat dengan SibSp = 0 lebih banyak dibandingkan dengan yang memiliki lebih banyak saudara/saudari.
- Survived = 0: Menunjukkan jumlah penumpang yang meninggal, juga dibagi berdasarkan jumlah SibSp. Misalnya, banyak penumpang yang meninggal memiliki SibSp = 1 atau SibSp = 2, tetapi lebih sedikit yang meninggal dengan SibSp = 0.

Selanjutnya, diberikan grafik visualisasi distribusi SibSp sebagai berikut.



Gambar 19. Grafik Visualisasi Distribusi SibSp Penumpang Kapal Titanic

Grafik Visualisasi Distribusi SibSp tersebut menegaskan bahwa ada lebih dari dua orang tua atau anak yang kemungkinan besar akan bertahan hidup. Sedangkan jika sendirian di atas kapal kemungkinan besar akan meninggal.

Selanjutnya diberikan analisis visualisasi distribusi jumlah orang tua atau anak (Parch) yang dibawa oleh penumpang sebagai berikut.

```
bar_chart('Parch')
print("Survived :\n",train[train['Survived']==1]['Parch'].value_counts())
print("Dead:\n",train[train['Survived']==0]['Parch'].value_counts())
```

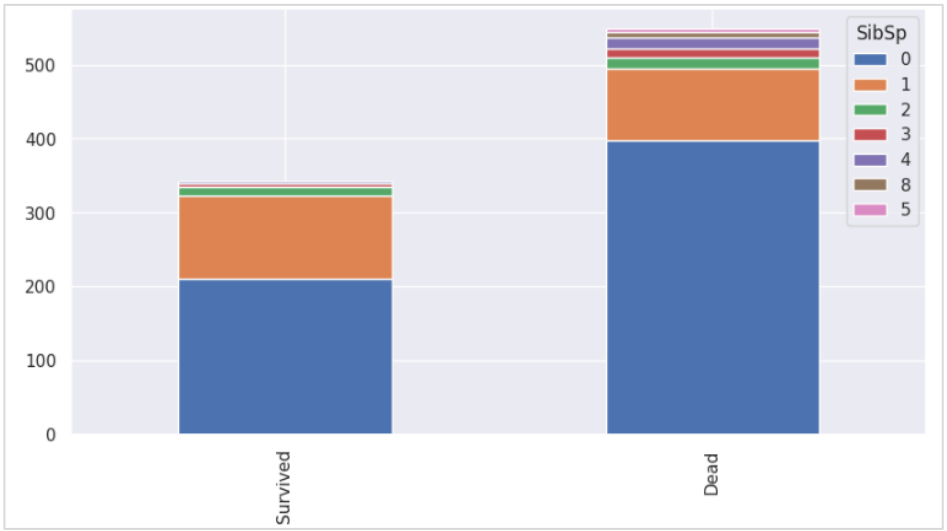
```
Survived :
 Parch
0    233
1     65
2     40
3      3
5      1
Name: count, dtype: int64
Dead:
 Parch
0    445
1     53
2     40
5      4
4      4
3      2
6      1
Name: count, dtype: int64
```

Gambar 20. Source Program dan Output (Parch)

Dalam Gambar 20, menunjukan distribusi Parch menunjukkan jumlah penumpang berdasarkan jumlah orang tua atau anak (Parch) yang dibawa:

- Survived = 1: Menunjukkan jumlah penumpang yang selamat, dibagi berdasarkan jumlah orang tua atau anak (Parch) yang mereka bawa.
- Survived = 0: Menunjukkan jumlah penumpang yang meninggal, juga dibagi berdasarkan jumlah orang tua atau anak.

Selanjutnya, diberikan grafik visualisasi distribusi (Parch) sebagai berikut



Gambar 21. Grafik Visualisasi Distribusi (Parch) Penumpang Kapal Titanic

Gambat 20 dan 21 menunjukkan bahwa penumpang yang memiliki sedikit atau tidak ada orang tua/anak (Parch = 0) lebih cenderung meninggal, sedangkan penumpang dengan lebih banyak orang tua/anak (Parch > 0) cenderung selamat.

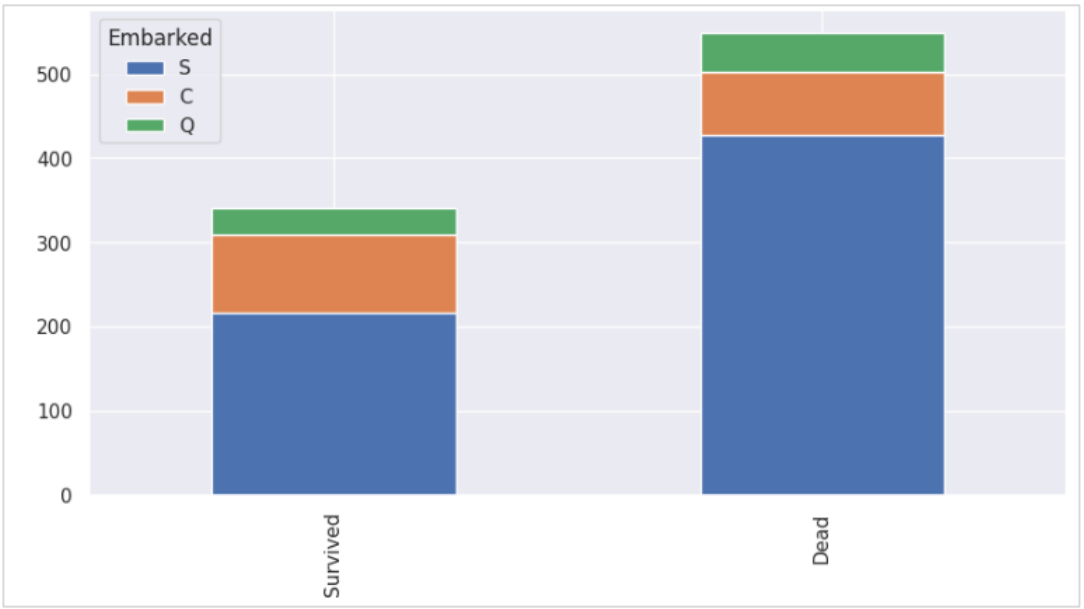
Selanjutnya diberikan analisis untuk melihat sebaran (bar_chart(Embarked)), jumlah penumpang berdasarkan pelabuhan keberangkatan mereka sebagai berikut

```
bar_chart('Embarked')
print("Survived :\n",train[train['Survived']==1]['Embarked'].value_counts())
print("Dead:\n",train[train['Survived']==0]['Embarked'].value_counts())
```

```
Survived :
 Embarked
S    217
C     93
Q     30
Name: count, dtype: int64
Dead:
 Embarked
S    427
C     75
Q     47
Name: count, dtype: int64
```

Gambar 22. Source Program dan Output (Embarked)

Dalam Gambar 22, menunjukan bahwa survived S memilikki 217, C memilikki 93 dan Q memilikki 30 dan untuk Dead S memilikki 427,C memilikki 75 dan Q memilikki 47Selanjutnya, diberikan grafik visualisasi distribusi (Embarked) sebagai berikut



Gambar 23. Grafik Visualisasi Distribusi (Embarked) Penumpang Kapal Titanic

Grafik Visualisasi Gambar 23 tersebut menegaskan bahwa orang yang bangkit dari C sedikit lebih mungkin untuk bertahan hidup. Bar chart tersebut menegaskan bahwa kapal Q kemungkinan besar akan mati. Bagan tersebut menegaskan bahwa orang yang mendaki dari S kemungkinan besar akan meninggal.

4. Rekayasa fitur

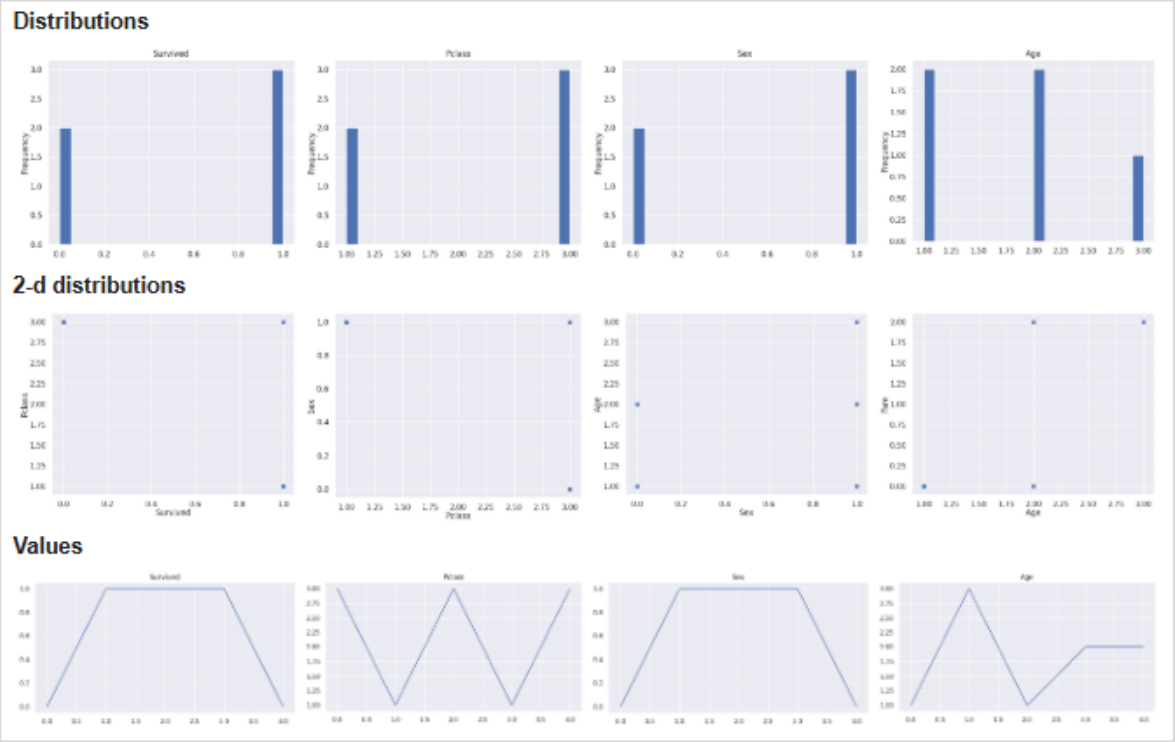
Pada rekayasa fitur, proses ini melibatkan penggunaan pengetahuan domain untuk menciptakan fitur-fitur baru (vektor fitur). Vektor fitur adalah representasi numerik dari atribut suatu objek, yang mempermudah pemrosesan dan analisis statistik. Selanjutnya, diberikan sourse program train.head() maka outputnya seperti dibawah ini.

```
train.head()
```

	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked	Title	FamilySize
0	0	3	0	1.0	0.0	2.0	0	0	0.4
1	1	1	1	3.0	2.0	0.8	1	2	0.4
2	1	3	1	1.0	0.0	2.0	0	1	0.0
3	1	1	1	2.0	2.0	0.8	0	2	0.4
4	0	3	0	2.0	0.0	2.0	0	0	0.0

Gambar 24. Source dan Output Program train.head()

Berdasarkan data di atas, kita dapat melihat bahwa terdapat 5 baris pertama dengan informasi mengenai penumpang Titanic. Usia penumpang yang tercatat berkisar antara yang paling tua yaitu 38 tahun dan yang paling muda 22 tahun. Selain itu, data ini juga menunjukkan bahwa ada 2 penumpang laki-laki dan 3 penumpang perempuan di baris pertama dataset.



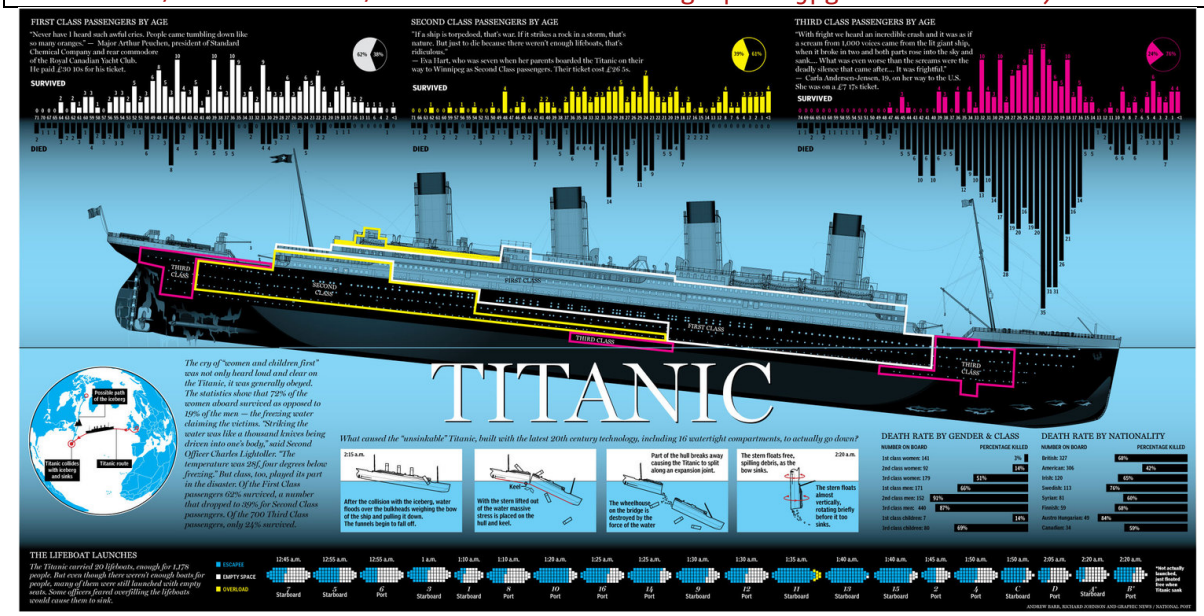
Gambar 24. Grafik Visualisasi Program train.head()

Berdasarkan gambar, data menunjukkan lima baris pertama dari penumpang Titanic. Usia penumpang berkisar antara 22 hingga 38 tahun, meskipun ada satu data yang kosong. Jenis kelamin penumpang terdiri dari 2 laki-laki dan 3 perempuan.

4.1 how titanic sank?

Dalam menganalisis tragedi tenggelamnya kapal Titanic, kita akan memulai dengan melihat data statistik penumpang yang tersedia pada source program berikut.

```
Image(url=
"https://static1.squarespace.com/static/5006453fe4b09ef2252ba068/t/5090b249e4b047ba54dfd258/1351660113175/TItanic-Survival-Infographic.jpg?format=1500w")
```



Gambar 25. Data Statistik Penumpang Kapal Titanic

Dari gambar diatas kita bisa melihat berapa banyak penumpang meninggal berdasarkan kelas jenis kelamin dan berdasarkan tingkat nasional. Selanjutnya kita akan melihat dataset ukuran keluarga korban tragedy titanic dengan source program train.head(12) berikut

```
train.head(10)
```

	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked	Title	FamilySize
0	0	3	0	1.0	0.0	2.0	0	0	0.4
1	1	1	1	3.0	2.0	0.8	1	2	0.4
2	1	3	1	1.0	0.0	2.0	0	1	0.0
3	1	1	1	2.0	2.0	0.8	0	2	0.4
4	0	3	0	2.0	0.0	2.0	0	0	0.0
5	0	3	0	2.0	0.0	2.0	2	0	0.0
6	0	1	0	3.0	2.0	1.6	0	0	0.0
7	0	3	0	0.0	1.0	2.0	0	3	1.6
8	1	3	1	2.0	0.0	2.0	0	2	0.8
9	1	2	1	0.0	2.0	1.8	1	2	0.4

Gambar 25. Data Statistik Penumpang Kapal Titanic

Berdasarkan gambar, analisis menunjukkan bahwa penumpang kelas pertama memiliki peluang lebih tinggi untuk selamat, sementara penumpang kelas ketiga lebih banyak yang meninggal. Kelas kedua menunjukkan distribusi yang lebih seimbang antara yang selamat dan yang meninggal. Hal ini mencerminkan perbedaan akses dan kemungkinan keselamatan antara kelas-kelas tiket di Titanic.

1) Title

Pada Title kita menggabungkan kumpulan data dengan menggunakan sourse berkut.

Soucre combinasi dataset

```
train_test_data = [train,test] # combine dataset

for dataset in train_test_data:
    dataset['Title'] = dataset['Name'].str.extract(' ([A-Za-z]+)\.', expand=False)
```

```
train['Title'].value_counts()
```

Title	count
Mr	517
Miss	182
Mrs	125
Master	40
Dr	7
Rev	6
Mlle	2
Major	2
Col	2
Countess	1
Capt	1
Ms	1
Sir	1
Lady	1
Mme	1
Don	1
Jonkheer	1

dtype: int64

(a) train['Title'].value_counts()

```
test['Title'].value_counts()
```

Title	count
Mr	240
Miss	78
Mrs	72
Master	21
Col	2
Rev	2
Ms	1
Dr	1
Dona	1

dtype: int64

(b) test['Title'].value_counts()

Gambar 26. Data banyak penumpang berdasarkan Title

Berdasarkan Gambar 26, kita bisa melihat berapa banyak penumpang berdasarkan Title, Title Mr sebanyak 517 merupakan jumlah terbanyak dari julah Title yang lainya. Selanjutnya, pada data test jumlah Mr sebanyak 240, Miss sebanyak 78, Mrs sebanyak 72, Master sebanyak 21, Rev dan col sebanyak 2 dan Dr,Ms,Dona sebanyak 1 dengan data type integer.

- **Title Map**

  Untuk mengetahui berdasarkan Title pertama-tama kita petakan Title (Title Map) kita tuliskan perintah seperti dibawah ini :

Soucre Title Map

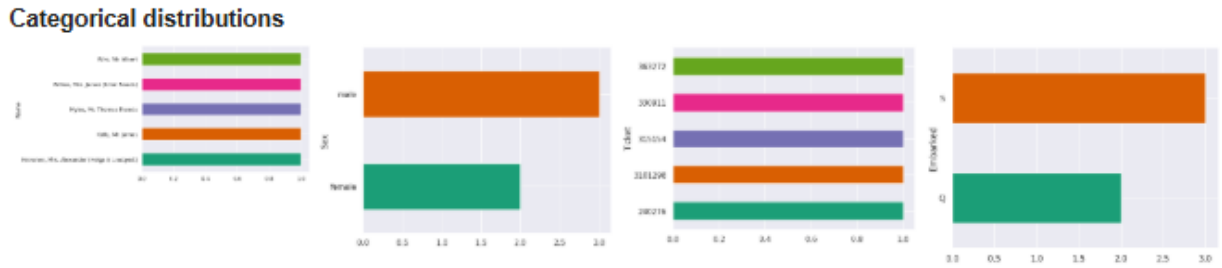
```
title_mapping = {"Mr": 0, "Miss": 1, "Mrs": 2,
                 "Master": 3, "Dr": 3, "Rev": 3, "Col": 3, "Major": 3, "Mlle": 3,"Countess": 3,
                 "Ms": 3, "Lady": 3, "Jonkheer": 3, "Don": 3, "Dona" : 3, "Mme": 3,"Capt": 3,"Sir": 3 }

for dataset in train_test_data:
    dataset['Title'] = dataset["Title"].map(title_mapping)
```

Dari data diatas kita akan memetakan berdasarkan Title penumpang titanic, kemudian selanjutnya kita tuliskan perintah dataset.head()

```
dataset.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q	0
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S	2
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q	0
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S	0
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S	2

Gambar 27. Data Statistik Penumpang Kapal Titanic berdasarkan title dengan source dataset.head()



Gambar 28. Dsitribusi Kategori Penumpang Kapal Titanic berdasarkan title dengan source dataset.head()

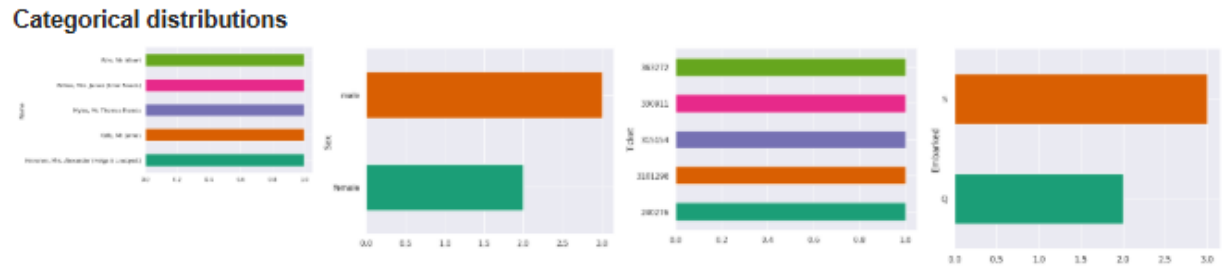
Berdasarkan data yang ditampilkan pada Gmabar 27 dan Gambar 28, kita dapat melihat informasi mengenai gelar (Title) penumpang Titanic. Pada kolom Title, gelar Mr memiliki jumlah 3, yang menunjukkan ada 3 penumpang laki-laki. Sementara itu, gelar Mrs tercatat sebanyak 2, yang menunjukkan ada 2 penumpang wanita yang menikah.

Selanjutnya kita tuliskan perintah test.head() dan outputnya adalah :

```
test.head()
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q	0
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S	2
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q	0
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S	0
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S	2

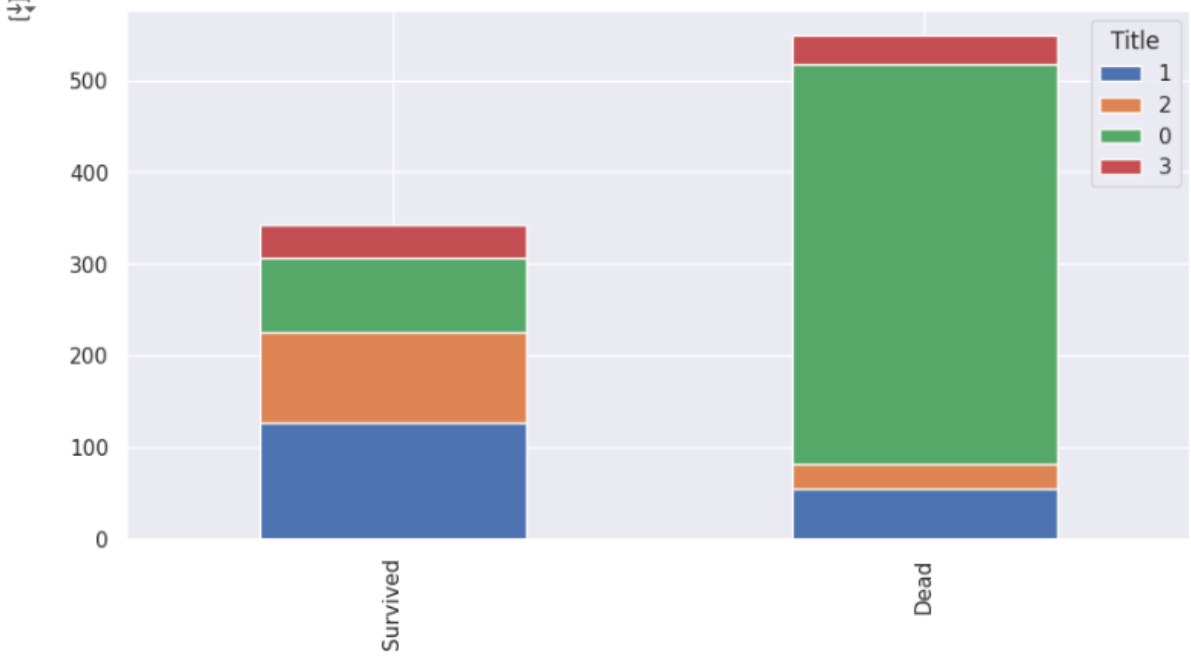
Gambar 29. Data Statistik Penumpang Kapal Titanic berdasarkan title dengan source test.head()



Gambar 30. Dsitribusi Kategori Penumpang Kapal Titanic berdasarkan title dengan source test.head()

Berdasarkan Gambar 29 dan Gambar 30, menunjukkan bahwa Mr. berjumlah 3 dan paling tua berumur 62 tahun dan Mrs paling tua berumur 47 tahun. Sehingga kita dapat membuat grafik visualisasinya seperti dibawah ini.

```
bar_chart('Title')
```



Gambar 30. Grafik Visualisasinya distribusi Kategori Penumpang Kapal Titanic berdasarkan title

Dalam gambar diatas menegaskan bahwa Mrs = 1 banyak yang bertahan dan untuk banyak yang meninggal adalah Mr = 0 dalam tragedi titanic.

Selanjutnya untuk mengurangi kompleksitas data, Kolom 'Name' mungkin tidak relevan untuk analisis atau model prediksi, sehingga dihapus untuk menyederhanakan dataset.

Soucre delete unnecessary feature from dataset

```
# delete unnecessary feature from dataset
train.drop('Name', axis=1, inplace=True)
test.drop('Name', axis=1, inplace=True)
```

```
[35] train.head()
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	1	0	3	male	22.0	1	0	A/5 21171	7.2500	NaN	S	0
1	2	1	1	female	38.0	1	0	PC 17599	71.2833	C85	C	2
2	3	1	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	1
3	4	1	1	female	35.0	1	0	113803	53.1000	C123	S	2
4	5	0	3	male	35.0	0	0	373450	8.0500	NaN	S	0

Gambar 31. Data Hasil Penyederhanakan Dataset

2) Jenis Kelamin/Sex

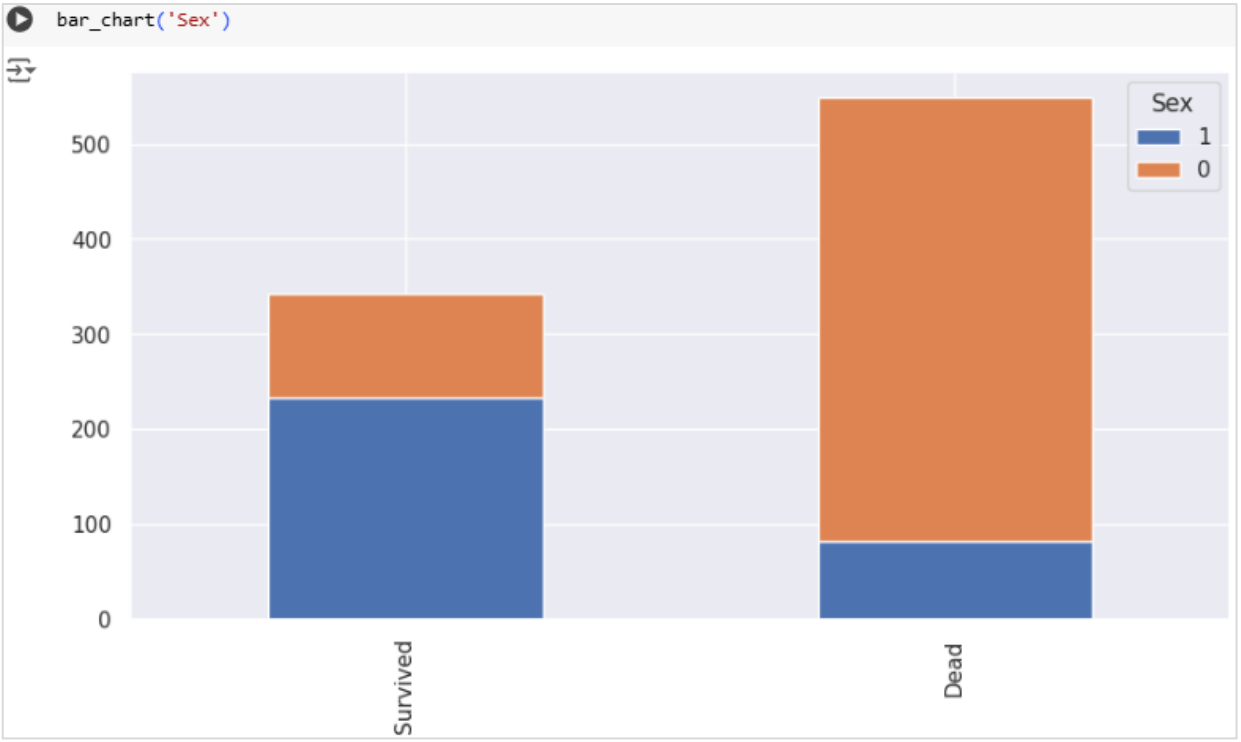
Selanjutnya diberikan analasisis distribusi Penumpang Kapal Titanic yang dipetakan berdasarkan kategori jenis kelamin penumpang tragedi titanic dengan menuliskan perintah :

Soucre sex_mapping

```
sex_mapping = {"male": 0, "female": 1}
for dataset in train_test_data:
    dataset['Sex'] = dataset['Sex'].map(sex_mapping)
```

Berdasarkan Gambar diatas, kita dapat melihat laki-laki = 0 dan wanita = 1. Kemudian visualisasikan dengan grafik agar dapat dianalisa dengan baik.

```
bar_chart('Sex')
```



Gambar 32. Grafik Visualisasinya distribusi kategori jenis kelamin Penumpang Kapal Titanic

Pada gambar grafik visualisasi diatas menegaskan bahwa jumlah Laki-laki lebih banyak yang meninggal dan jumlah wanita lebih banyak yang bertahan. Selanjutya, diberikan menganalisis pola kelangsungan hidup sesuai kelompok jenis kelamin, memberikan wawasan tentang bagaimana usia memengaruhi kemungkinan selamat atau meninggal.

```
test.head()
```

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	892	3	0	34.5	0	0	330911	7.8292	NaN	Q	0
1	893	3	1	47.0	1	0	363272	7.0000	NaN	S	2
2	894	2	0	62.0	0	0	240276	9.6875	NaN	Q	0
3	895	3	0	27.0	0	0	315154	8.6625	NaN	S	0
4	896	3	1	22.0	1	1	3101298	12.2875	NaN	S	2

Gambar 33. Grafik Visualisasinya distribusi kelangsungan hidup penumpang Titanic berdasarkan Gender

Grafik di atas menunjukkan distribusi kelangsungan hidup penumpang Titanic berdasarkan jenis kelamin, tidak ada penumpang laki-laki yang selamat, sementara ada 2 penumpang wanita yang selamat, yang mengindikasikan bahwa wanita memiliki peluang lebih tinggi untuk bertahan hidup dibandingkan laki-laki.

3) Usia/Age

Selanjutnya diberikan analasisis distribusi Penumpang Kapal Titanic yang dipetakan berdasarkan usia menggunakan median dengan menuliskan perintah

Soucre analisis Train and test “Age”

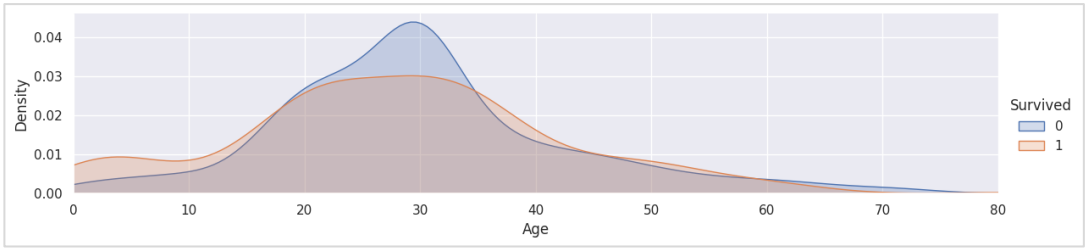
```
train["Age"].fillna(train.groupby("Title")["Age"].transform("median"), inplace= True)
test["Age"].fillna(test.groupby('Title')['Age'].transform("median"), inplace= True)
```

```
train.head()
#train.groupby("Title")["Age"].transform("median")
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	1	0	3	0	22.0	1	0	A/5 21171	7.2500	NaN	S	0
1	2	1	1	1	38.0	1	0	PC 17599	71.2833	C85	C	2
2	3	1	3	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	1
3	4	1	1	1	35.0	1	0	113803	53.1000	C123	S	2
4	5	0	3	0	35.0	0	0	373450	8.0500	NaN	S	0

Gambar 34. Grafik Visualisasinya distribusi kelangsungan hidup penumpang Titanic berdasarkan Gender

Gambar 34 diatas kita dapat melihat ada 5 kolom terdiri usia paling tua adalah 38 dan usia paling muda adalah 22. Kemudian diberikan visualisasi untuk melihat distribusi usia penumpang Titanic berdasarkan status kelangsungan hidupnya, dan memahami perbedaan antara penumpang yang selamat dan yang meninggal menurut usia mereka.

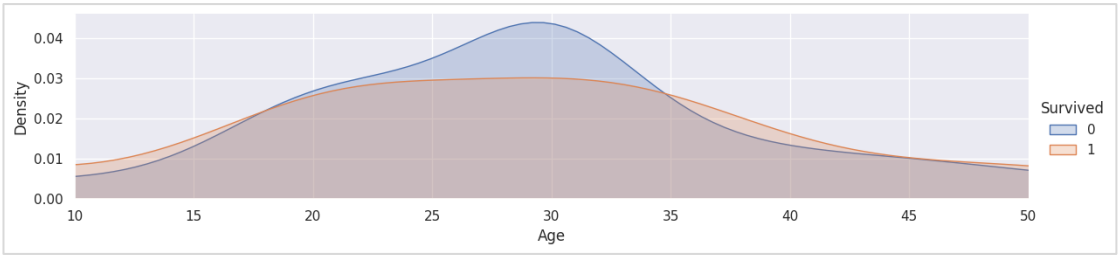


Gambar 34. Grafik Visualisasinya distribusi kelangsungan hidup penumpang Titanic berdasarkan Gender

Dari data diatas menegaskan bahwa jumlah rata-rata penumpang tragedi titanic laki-laki dan wanita yang paling banyak bertahan dan meninggal adalah berkisar berusia 20 sampai 30 an. Setelah itu kita akan melihat jumlah penumpang yang bertahan pada tragedi titanic yang berumur dari 10 sampai 50 tahun dengan sources berikut.

```
facet = sns.FacetGrid(train, hue="Survived",aspect=4)
facet.map(sns.kdeplot,'Age',shade= True)
facet.set(xlim=(0, train['Age'].max()))
facet.add_legend()
plt.xlim(10,50)
```

Selanjutnya, diberikan grafik visualisasi source tersebut.



Dari data diatas bahwa usia 20 sampai 30 an adalah penumpang yang banyak bertahan dan meninggal pada tragedi titanic

4.2 Pengelompokan/Binning

Sebagai contoh kita akan mengelompokan tragedi titanic berdasarkan usia ke variable katagori anak- anak, muda, dewasa, paruh baya dan senior dengan source berikut.

```
train.head()
```

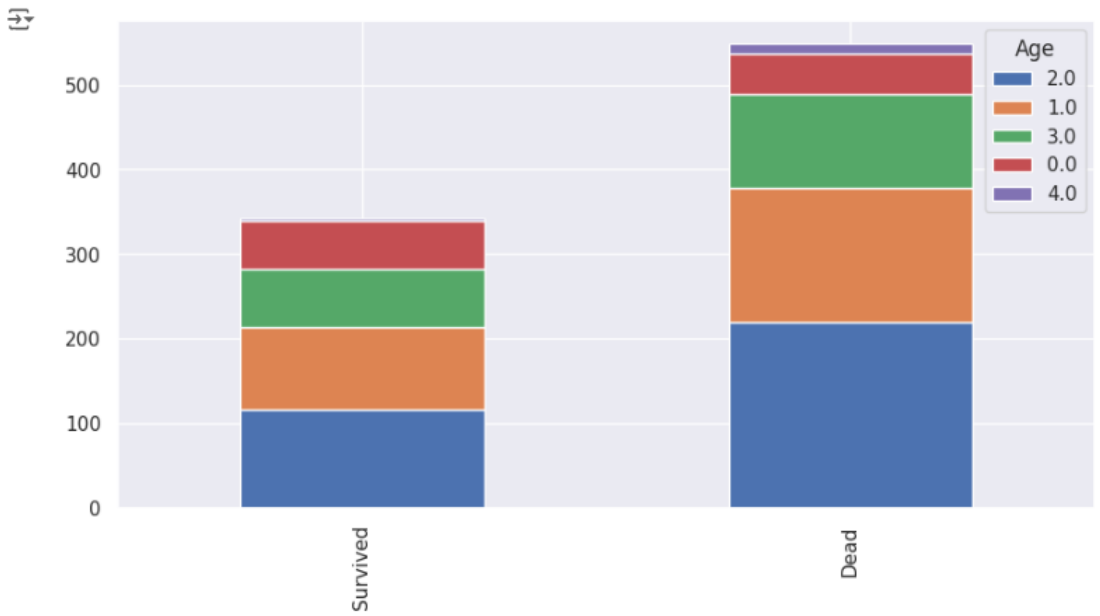
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	1	0	3	0	22.0	1	0	A/5 21171	7.2500	NaN	S	0
1	2	1	1	1	38.0	1	0	PC 17599	71.2833	C85	C	2
2	3	1	3	1	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	1
3	4	1	1	1	35.0	1	0	113803	53.1000	C123	S	2
4	5	0	3	0	35.0	0	0	373450	8.0500	NaN	S	0

Gambar 35. Source dan Output Pengelompokan Tragedi Titanic Berdasarkan Pengelompokan/Binning

Dari data diatas bahwa usia paling muda adalah 22 tahun dan usia paling tua adalah 38 tahun. Kemudian kita kan memetakan berdasarkan usia dengan menulis perintah seperti berikut.

```
for dataset in train_test_data:
    dataset.loc[dataset['Age'] <= 16, 'Age'] = 0
    dataset.loc[(dataset['Age'] > 16) & (dataset['Age'] <= 26), 'Age'] = 1
    dataset.loc[(dataset['Age'] > 26) & (dataset['Age'] <= 36), 'Age'] = 2
    dataset.loc[(dataset['Age'] > 36) & (dataset['Age'] <= 62), 'Age'] = 3
    dataset.loc[dataset['Age'] > 62, 'Age'] = 4
```

```
train.head()
bar_chart('Age')
```



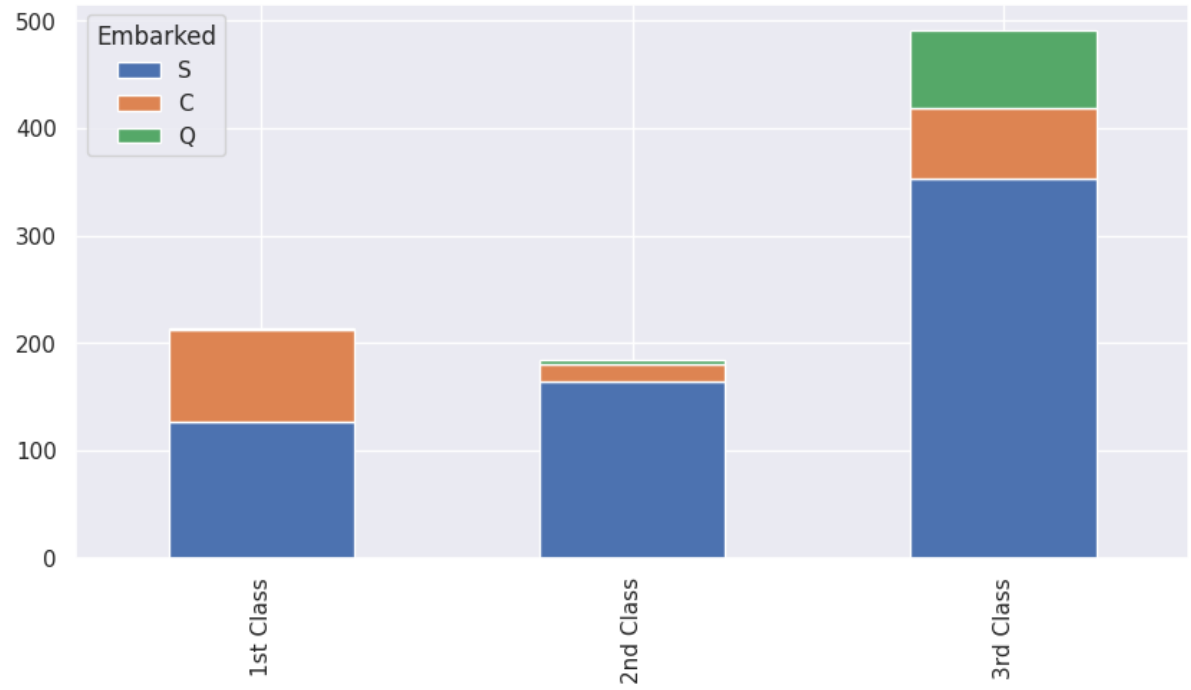
Gambar 36. Grafik Visualisasi Pengelompokan Tragedi Titanic Berdasarkan Pengelompokan/Binning

Dalam data tersebut, kita membagi penumpang menjadi 5 kelompok berdasarkan usia: 0 hingga 16 tahun untuk kategori anak-anak, 16 hingga 26 tahun untuk kategori muda, 26 hingga 36 tahun untuk kategori dewasa, 36 hingga 62 tahun untuk kategori paruh baya, dan 62 tahun ke atas untuk kategori senior. Hal ini menegaskan bahwa menegaskan bahwa jumlah usia yang selamat dan meninggal dari tragedi titanic yang paling banyak adalah berkisar antara 26 sampai 36 tahun.

Selanjutnya, data di atas dapat dipetakan berdasarkan Pclass (kelas tiket) agar analisis menjadi lebih rinci. Dengan membagi data menurut kelas tiket, kita dapat melihat bagaimana kelangsungan hidup penumpang berbeda antara kelas 1, 2, dan 3.

```
Pclass1 = train[train['Pclass'] == 1]['Embarked'].value_counts()
Pclass2 = train[train['Pclass'] == 2]['Embarked'].value_counts()
Pclass3 = train[train['Pclass'] == 3]['Embarked'].value_counts()
df = pd.DataFrame([Pclass1,Pclass2,Pclass3])
df.index = ['1st Class','2nd Class','3rd Class']
df.plot(kind = 'bar', stacked =  True, figsize=(10,5))
plt.show()
print("Pclass1:\n",Pclass1)
print("Pclass2:\n",Pclass2)
print("Pclass3:\n",Pclass3)
```

Berdasarkan source diatas kita mebagi Pclass menjadi 3 bagian yaitu Pclass 1, Pclass 2 dan Pclass 3 untuk melihat penumpang berasal dari S (Southampton),C (Cherbourg) dan Q (Queenstown). Maka didaptkan output hasil grafik yang dihasilkan adalah :



Gambar 37. Grafik Visualisasi Pengelompokan Tragedi Titanic Berdasarkan Pclass

Dari bar chart di atas, dapat dilihat bahwa sebagian besar penumpang Titanic, baik yang berasal dari Pclass 1, 2, maupun 3, berangkat dari pelabuhan S (Southampton). Hal ini menunjukkan bahwa Southampton merupakan pelabuhan utama yang digunakan oleh penumpang Titanic, terlepas dari kelas tiket yang mereka miliki.

Pclass	S (Southampton)	C (Cherbourg)	Q (Queenstown)
1	127	85	2
2	164	17	3
3	353	66	72

Dari data diatas disimpulkan jumlah penumpang terbanyak Pclass 1,2 dan 3 adalah berasal dari S (Southampton) dengan masing-masing jumlah Pclass 1 = 127, Pclass 2 = 164 dan Pclass 3 = 353.

Selanjutnya kita akan memetakan berdasarkan tariff penumpang tragedi titanic dengan sourse berikut.

```
for dataset in train_test_data:
    dataset['Embarked'] =  dataset['Embarked'].fillna('S')
train.head()
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	1	0	3	0	1.0	1	0	A/5 21171	7.2500	NaN	S	0
1	2	1	1	1	3.0	1	0	PC 17599	71.2833	C85	C	2
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	7.9250	NaN	S	1
3	4	1	1	1	2.0	1	0	113803	53.1000	C123	S	2
4	5	0	3	0	2.0	0	0	373450	8.0500	NaN	S	0

Gambar 38. Source dan Output Pengelompokan Tragedi Titanic Berdasarkan Embarked

Dari data di atas, dapat dilihat bahwa nilai tarif (Fare) tertinggi adalah 71.2833, sementara tarif terendah adalah 7.2500. Hal ini menunjukkan adanya perbedaan yang signifikan dalam harga tiket yang dibayar oleh penumpang Titanic, yang kemungkinan terkait dengan kelas tiket yang mereka pilih dan fasilitas yang tersedia.

Selanjutnya, agar memudahkan pemrosesan data dalam model machine learning diberikan source program sebagai berikut.

```
embarked_mapping = {'S':0,'C':1,'Q':2}
for dataset in train_test_data:
    dataset['Embarked'] = dataset['Embarked'].map(embarked_mapping)
```

Untuk menangani nilai hilang pada kolom Fare di dataset Titanic, kita mengisinya dengan median tarif berdasarkan Pclass. Pendekatan ini mempertimbangkan perbedaan tarif antar kelas tiket, memastikan data yang hilang terisi dengan cara yang relevan dan menjaga kualitas data untuk analisis lebih lanjut.

```
# fill missing Fare with median fare for each Pclass
train["Fare"].fillna(train.groupby("Pclass")["Fare"].transform("median"), inplace=True)
test["Fare"].fillna(test.groupby("Pclass")["Fare"].transform("median"), inplace=True)
train.head(10)
```

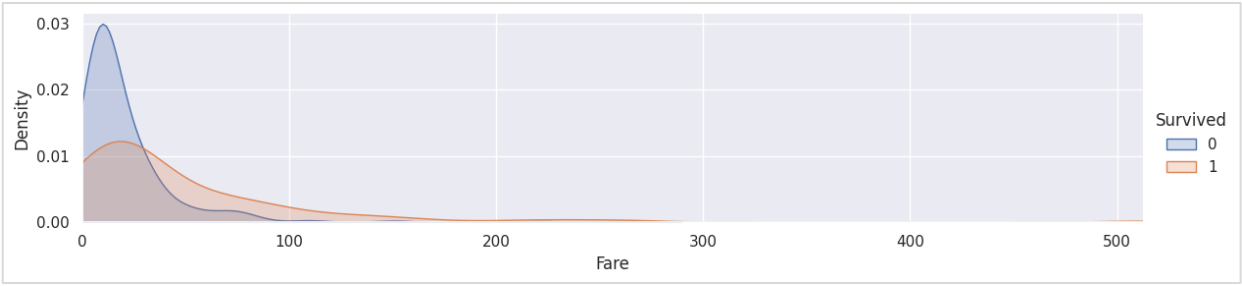
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	1	0	3	0	1.0	1	0	A/5 21171	7.2500	NaN	0	0
1	2	1	1	1	3.0	1	0	PC 17599	71.2833	C85	1	2
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	7.9250	NaN	0	1
3	4	1	1	1	2.0	1	0	113803	53.1000	C123	0	2
4	5	0	3	0	2.0	0	0	373450	8.0500	NaN	0	0
5	6	0	3	0	2.0	0	0	330877	8.4583	NaN	2	0
6	7	0	1	0	3.0	0	0	17463	51.8625	E46	0	0
7	8	0	3	0	0.0	3	1	349909	21.0750	NaN	0	3
8	9	1	3	1	2.0	0	2	347742	11.1333	NaN	0	2
9	10	1	2	1	0.0	1	0	237736	30.0708	NaN	1	2

Gambar 39. Source dan Output fill missing Fare with median fare for each Pclass

Berdasarkan hasil analisis dan grafik di atas, kita dapat melihat distribusi kelangsungan hidup penumpang Titanic berdasarkan Pclass setelah nilai Fare yang hilang diisi dengan median tarif untuk setiap kelas tiket. Dengan mengisi nilai yang hilang menggunakan pendekatan ini, data yang diperoleh menjadi lebih representatif dan relevan, memperhitungkan perbedaan tarif antara kelas 1, 2, dan 3. Pendekatan ini memungkinkan analisis yang lebih akurat mengenai kelangsungan hidup penumpang, yang menunjukkan bahwa penumpang kelas pertama memiliki peluang lebih tinggi untuk selamat, sementara penumpang kelas ketiga memiliki tingkat kematian yang lebih tinggi.

Selanjutnya, untuk menganalisis distribusi tarif tiket (Fare) penumpang Titanic berdasarkan kelangsungan hidup mereka, kita menggunakan visualisasi Kernel Density Estimate (KDE). Dengan menggunakan sns.FacetGrid, kita dapat memetakan distribusi Fare untuk penumpang yang selamat dan yang meninggal.

```
facet = sns.FacetGrid(train, hue="Survived",aspect=4 )
facet.map(sns.kdeplot, 'Fare', shade = True)
facet.set(xlim = (0, train['Fare'].max()))
facet.add_legend()
plt.show()
```

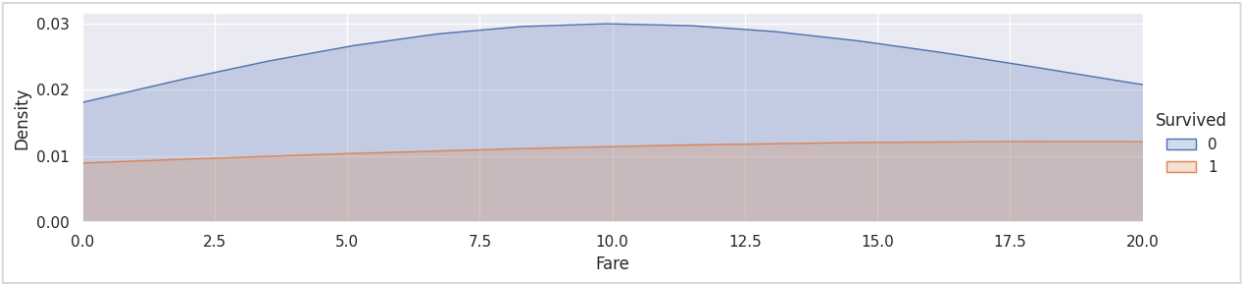


Gambar 39. Source dan Output Analisis Distribusi Tarif Tiket (Fare) Penumpang Titanic Berdasarkan Kelangsungan Hidup Mereka

Berdasarkan grafik, penumpang yang selamat (hue=1) cenderung memiliki tarif tiket yang lebih tinggi, sementara penumpang yang meninggal (hue=0) lebih banyak berada pada tarif tiket rendah. Hal ini menunjukkan bahwa tarif tiket yang lebih tinggi, biasanya di kelas 1 dan 2, berhubungan dengan peluang kelangsungan hidup yang lebih besar.

Untuk menganalisis distribusi tarif tiket (Fare) pada penumpang Titanic yang selamat dan yang meninggal, kita menggunakan visualisasi Kernel Density Estimate (KDE) dengan sns.FacetGrid. Dengan membatasi sumbu x antara 0 hingga 20, kita dapat lebih fokus pada kisaran tarif tiket yang lebih rendah.

```
facet = sns.FacetGrid(train, hue="Survived",aspect=4)
facet.map(sns.kdeplot,'Fare',shade= True)
facet.set(xlim=(0, train['Fare'].max()))
facet.add_legend()
plt.xlim(0, 20)
```



Gambar 39. Source dan Output Analisis distribusi tarif tiket pada penumpang Titanic dengan KDE

Kemudian, dianalisis kembali data tersebut dengan membagi tarif menjadi 4 kelompok dengan source berikut.

```
for dataset in train_test_data:
    dataset.loc[dataset['Fare'] <= 17, 'Fare'] = 0
    dataset.loc[(dataset['Fare'] > 17) & (dataset['Fare'] <= 30), 'Fare'] = 1
    dataset.loc[(dataset['Fare'] > 30) & (dataset['Fare'] <= 100), 'Fare'] = 2
    dataset.loc[dataset['Fare'] >= 100, 'Fare'] = 3
```

Dari data diatas kita melihat data 0 = tarif kurang dari 17, data 1 = tarif 17 sampai 30, 2 = tarif 30 sampai 100 dan 3 = lebih dari 100. Kemudian kita tulis perintah train.head() maka outputnya adalah :

```
train.head()
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title
0	1	0	3	0	1.0	1	0	A/5 21171	0.0	NaN	0	0
1	2	1	1	1	3.0	1	0	PC 17599	2.0	C85	1	2
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	0.0	NaN	0	1
3	4	1	1	1	2.0	1	0	113803	2.0	C123	0	2
4	5	0	3	0	2.0	0	0	373450	0.0	NaN	0	0

Gambar 40. Source dan Output tarif pngelompokan tarif menjadi 4 kelompok

Dari hasil output di atas, dapat dilihat bahwa penumpang dengan tarif tiket kurang dari 17 (kode 0) berjumlah 3 orang, sementara penumpang dengan tarif tiket antara 30 hingga 100 (kode 2) berjumlah 2 orang..

Selanjutnya kita akan memetakan tragedi titanic dengan cara melihat dataset cabin dengan menggunakan source berikut.

```
train.Cabin.value_counts().head()
```

Cabin	count
B96 B98	4
G6	4
C23 C25 C27	4
C22 C26	3
F33	3

dtype: int64

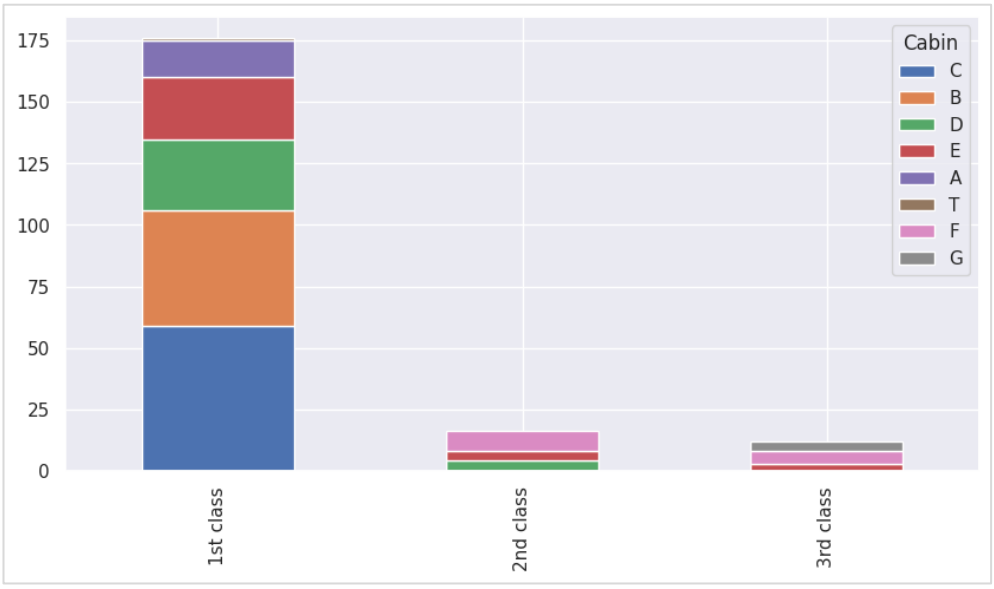
Gambar 41. Source dan Output distribusi tarif tiket (Fare) berdasarkan kelangsungan hidup penumpang Titanic

Gambar doiatas menunjukkan bahwa penumpang yang selamat (hue=1) cenderung membayar tarif tiket yang lebih tinggi, sementara penumpang yang meninggal (hue=0) lebih banyak membayar tarif tiket yang lebih rendah.

```
for dataset in train_test_data:
    dataset['Cabin'] =  dataset['Cabin'].str[:1]
```

```
Pclass1 = train[train['Pclass']==1]['Cabin'].value_counts()
Pclass2 = train[train['Pclass']==2]['Cabin'].value_counts()
Pclass3 = train[train['Pclass']==3]['Cabin'].value_counts()
df = pd.DataFrame([Pclass1, Pclass2, Pclass3])
df.index = ['1st class','2nd class', '3rd class']
df.plot(kind='bar',stacked=True, figsize=(10,5))
```

Dari dataset diatas kita membagi cabin menjadi 3 Pclass kategori sehingga menghasilkan output brikut.



Gambar 41. Grafik Visualisasi Pengelompokan 3 Pclass kategori Tragedi Titanic Berdasarkan Pclass

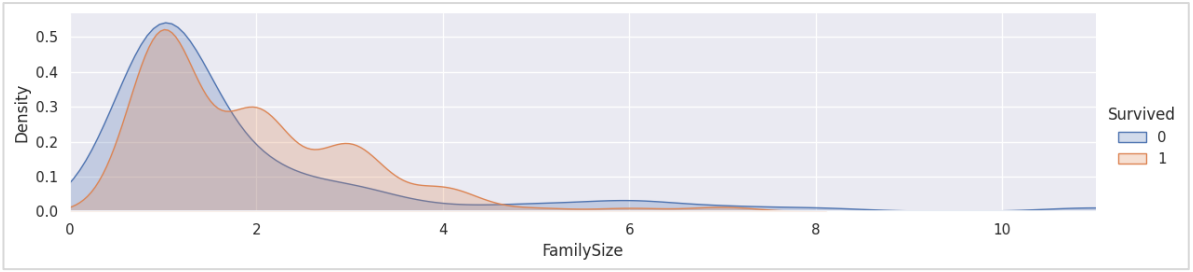
```
cabin_mapping = {"A": 0, "B": 0.4, "C": 0.8, "D": 1.2, "E": 1.6, "F": 2, "G": 2.4, "T": 2.8}
for dataset in train_test_data:
    dataset['Cabin'] = dataset['Cabin'].map(cabin_mapping)
```

Dari hasil Outpu menjelaskan bahwa jumlah cabin dibagi menjadi 3 class dimana 3 class itu terdiri dari A sampai T disini kata dapat melihat juga bahwa penumpang 1 Class C lebih banyak dari pada penumpang lainya pada tragedi titanic.

Untuk menganalisis pengaruh ukuran keluarga (FamilySize) terhadap kelangsungan hidup penumpang Titanic, kita membuat kolom baru yang menjumlahkan jumlah saudara/kakak dan orangtua/anak (SibSp dan Parch) dan menambahkan 1 untuk menghitung diri sendiri. Setelah itu, kita menggunakan visualisasi Kernel Density Estimate (KDE) untuk melihat distribusi ukuran keluarga berdasarkan kelangsungan hidup. Grafik ini memberikan wawasan apakah ukuran keluarga berpengaruh terhadap kemungkinan bertahan hidup penumpang.

```
train["FamilySize"] = train["SibSp"] + train["Parch"] + 1
test["FamilySize"] = test["SibSp"] + test["Parch"] + 1
```

```
facet = sns.FacetGrid(train, hue="Survived",aspect=4)
facet.map(sns.kdeplot,'FamilySize',shade= True)
facet.set(xlim=(0, train['FamilySize'].max()))
facet.add_legend()
plt.xlim(0)
```



Gambar 42. Source dan Output (FamilySize) terhadap kelangsungan hidup penumpang Titanic

```
family_mapping = {1: 0, 2: 0.4, 3: 0.8, 4: 1.2, 5: 1.6, 6: 2, 7: 2.4, 8: 2.8, 9: 3.2, 10: 3.6, 11: 4}
for dataset in train_test_data:
    dataset['FamilySize'] = dataset['FamilySize'].map(family_mapping)
```

Dari hasil diatas dijelaskan bahwa dataset family ditambah 1 dan dibagi kelipatan 0.4 kemudian tuliskan perintah train.head() maka hasil outputnya adalah :

```
train.head()
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title	FamilySize
0	1	0	3	0	1.0	1	0	A/5 21171	0.0	2.0	0	0	0.4
1	2	1	1	1	3.0	1	0	PC 17599	2.0	0.8	1	2	0.4
2	3	1	3	1	1.0	0	0	STON/O2. 3101282	0.0	2.0	0	1	0.0
3	4	1	1	1	2.0	1	0	113803	2.0	0.8	0	2	0.4
4	5	0	3	0	2.0	0	0	373450	0.0	2.0	0	0	0.0

Dari data di atas, dapat dijelaskan bahwa terdapat distribusi jumlah penumpang pada beberapa kategori. Penumpang dengan Cabin 2 berjumlah 3 orang, sedangkan Cabin 0.8 berjumlah 2 orang. Selain itu, untuk Pclass 3, terdapat 3 penumpang, dan untuk Pclass 1, terdapat 2 penumpang.

5. Modelling

Pada tahap ini, kita akan melakukan importing classifier modules untuk membangun model prediksi berdasarkan data Titanic. Modul-modul classifier yang akan diimpor memungkinkan kita untuk menggunakan beberapa algoritma pembelajaran mesin berikut.

```
# Importing Classifier Modules
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier,ExtraTreeClassifier
from sklearn.ensemble import RandomForestClassifier,ExtraTreesClassifier,BaggingClassifier,AdaBoostClassifier,GradientBoostingClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

import numpy as np
```

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Survived    891 non-null    int64
 1   Pclass      891 non-null    int64
 2   Sex         891 non-null    int64
 3   Age         891 non-null    float64
 4   Fare        891 non-null    float64
 5   Cabin       891 non-null    float64
 6   Embarked    891 non-null    int64
 7   Title       891 non-null    int64
 8   FamilySize  891 non-null    float64
dtypes: float64(4), int64(5)
memory usage: 62.8 KB
```

Gambar 43. Source dan Hasil train.info()

Dari hasil data di atas, dapat dijelaskan bahwa terdapat 9 kolom data masukan (input data) yang terdiri dari 5 kolom bertipe integer dan 4 kolom bertipe floating. Kolom-kolom bertipe integer umumnya berisi data kategorikal atau angka bulat yang mewakili kategori, seperti Pclass dan SibSp, sedangkan kolom floating berisi data numerik dengan angka desimal, seperti Age, Fare, dan beberapa kolom lainnya yang menggambarkan nilai kontinu.

6. Cross Validation(k-fold)

Pada tahap ini, kita akan mengimpor modul sklearn.model_selection untuk menggunakan teknik K-Fold Cross Validation dalam mengevaluasi model. K-Fold Cross Validation adalah metode yang umum digunakan untuk memvalidasi kinerja model pembelajaran mesin dengan membagi dataset menjadi beberapa bagian (folds) dan melatih serta menguji model pada setiap bagian tersebut..

```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
k_fold = KFold(n_splits=10, shuffle=True, random_state=0)
```

```
#learning_rates = [0.05, 0.1, 0.25, 0.5, 0.75, 1]
clf = [KNeighborsClassifier(n_neighbors = 13),DecisionTreeClassifier(),
       RandomForestClassifier(n_estimators=13),GaussianNB(),SVC(),ExtraTreeClassifier(),
      GradientBoostingClassifier(n_estimators=10, learning_rate=1,max_features=3, max_depth =3, random_state = 10),AdaBoostClassifier(),ExtraTreesClassifier()]
def model_fit():
    scoring = 'accuracy'
    for i in range(len(clf)):
        score = cross_val_score(clf[i], train_data, target, cv=k_fold, n_jobs=1, scoring=scoring)
        print("Score of Model",i,":",round(np.mean(score)*100,2))
#     round(np.mean(score)*100,2)
#     print("Score of :\n",score)
model_fit()
```

Berdasarkan evaluasi menggunakan teknik K-Fold Cross Validation, berikut hasil performa dari setiap model

```
Score of Model 0 : 81.93
Score of Model 1 : 79.69
Score of Model 2 : 81.04
Score of Model 3 : 78.78
Score of Model 4 : 83.5
Score of Model 5 : 78.23
Score of Model 6 : 81.25
Score of Model 7 : 81.03
Score of Model 8 : 80.59
```

Berdasarkan hasil skor model yang diberikan, Model 4 menunjukkan performa terbaik dengan skor 83.5, yang mengindikasikan bahwa model ini mampu memberikan prediksi yang lebih akurat dibandingkan dengan model lainnya. Ini menunjukkan bahwa model tersebut lebih baik dalam menangkap pola data dan memiliki kemampuan generalisasi yang lebih baik. Di sisi lain, Model 0 (81.93), Model 2 (81.04), Model 6 (81.25), dan Model 7 (81.03) memiliki skor yang hampir setara, yang berarti model-model ini juga cukup baik dalam memprediksi data, namun tidak seunggul Model 4. Sementara itu, Model 1 (79.69) dan Model 3 (78.78) memiliki skor yang sedikit lebih rendah, menunjukkan bahwa meskipun mereka masih memberikan hasil yang cukup baik, mereka kurang optimal dibandingkan model lainnya. Model 5, dengan skor terendah 78.23, menunjukkan performa yang paling rendah di antara semua model, yang menandakan bahwa model ini perlu diperbaiki atau di-tune lebih lanjut untuk meningkatkan akurasi prediksinya. Secara keseluruhan, **Model 4 merupakan pilihan terbaik (nilai score : 83.5)**, sementara model-model lainnya, khususnya yang memiliki skor serupa, dapat dipertimbangkan sebagai alternatif yang cukup baik, namun Model 5 sebaiknya diperhatikan lebih lanjut untuk meningkatkan kinerjanya.

7. Testing

Pada langkah ini, kita akan melakukan proses testing untuk mengukur performa model yang telah dibangun. Hasil model akan diuji dengan menggunakan cross-validation untuk mendapatkan gambaran yang lebih akurat mengenai kemampuan model dalam mengklasifikasikan data yang tidak terlihat sebelumnya.

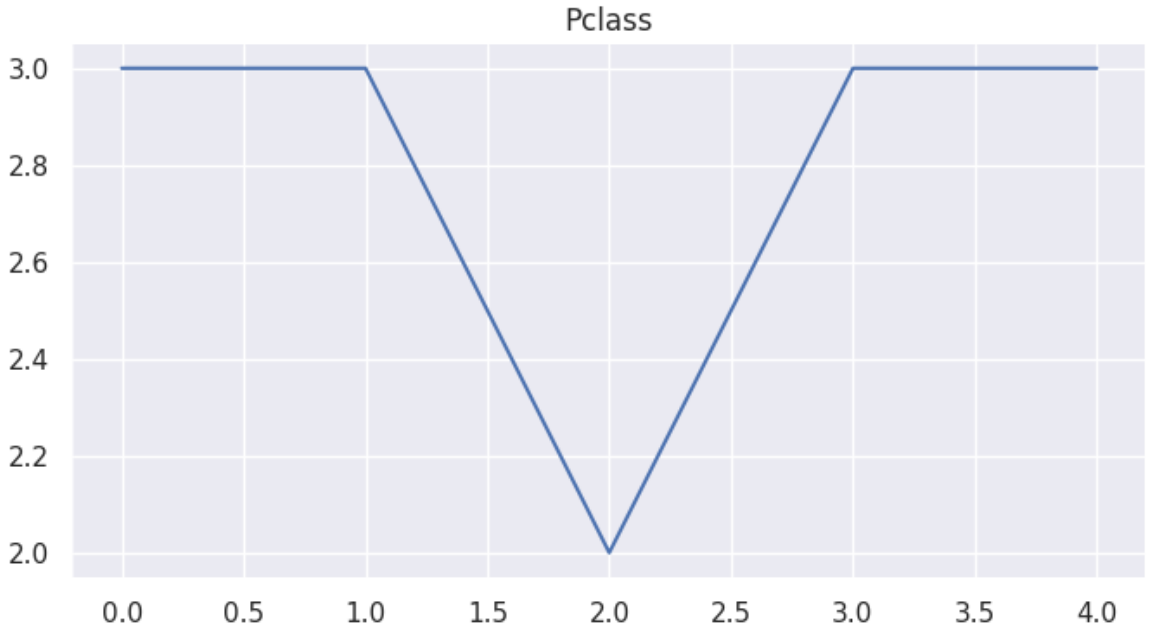
```
clf1 = SVC()
clf1.fit(train_data, target)
test
test_data = test.drop(['PassengerId'], axis=1)
test_data
prediction = clf1.predict(test_data)
```

```
test_data['Survived'] = prediction
test_data.head()
```

	Pclass	Sex	Age	Fare	Cabin	Embarked	Title	FamilySize	Survived
0	3	0	2.0	0.0	2.0	2	0	0.0	0
1	3	1	3.0	0.0	2.0	0	2	0.4	1
2	2	0	3.0	0.0	2.0	2	0	0.0	0
3	3	0	2.0	0.0	2.0	0	0	0.0	0
4	3	1	1.0	0.0	2.0	0	2	0.8	1

Gambar 44. Source dan Output Prediksi Penumbang Selamat pada Kapal Titanic

Berdasarkan hasil prediksi pada kolom Survived, terdapat beberapa hasil yang menarik. Pada kolom Pclass, terlihat bahwa 4 penumpang yang diprediksi selamat berada pada kelas 3, sementara 1 penumpang berada pada kelas 2 dibuktikan pada Gambar 44 dan Gambar 45. Hal ini menunjukkan bahwa penumpang dari kelas 3 cenderung memiliki peluang lebih besar untuk bertahan hidup dibandingkan dengan penumpang dari kelas 2, meskipun faktor lain seperti usia, ukuran keluarga, dan keputusan penyelamatan juga dapat berpengaruh. Selanjutnya, pada kolom Title, ditemukan bahwa 2 penumpang yang diprediksi selamat memiliki Title 2, dan 3 penumpang memiliki Title 0. Hal ini mengindikasikan bahwa penumpang dengan Title 2 (mungkin Mrs., Miss., atau Mr.) memiliki peluang lebih tinggi untuk bertahan hidup, sementara penumpang dengan Title 0 (kemungkinan kategori seperti "Master" atau lainnya) menunjukkan variasi yang lebih luas dan membutuhkan analisis lebih lanjut. Secara keseluruhan, hasil ini menunjukkan bahwa Pclass dan Title bisa menjadi faktor yang berhubungan dengan kelangsungan hidup, namun analisis lebih mendalam masih diperlukan untuk menggali faktor-faktor lain yang mempengaruhi hasil tersebut.



Gambar 45. Grafik Hasil prediksi survevied pada Pclass