Nama

NIM

Jurusan : MTI

Mata Kuliah : Tools and Technique for Computational Analysis

Analisis Kelangsungan Hidup Penumpang Titanic

1. **Mendefinisikan pernyataan masalah**

   Pada Analysis Kelangsungan hidup penumpang tragedi titanic kita akan mendefinisikan pernyataan masalah dengan memprediksi penumpang mana yang akan selamat dari tragedi Titanic.

   Contoh :

   Setelah kita melakukan perintah pada *https://www.kaggle.com/febrisugandi/titanic-passenger-survival-analysis* kita mengisi syntax seperti dibawah ini:

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

   Maka output yg dihasilkan adalah berupa gambar seperti dibawah ini:

Langkah ke 2 adalah memasukan atau mengimpor File CSV , CSV adalah format file umum untuk mentransfer dan menyimpan data. Kemampuan untuk membaca, memanipulasi, dan menulis data ke dan dari file CSV menggunakan Python adalah keterampilan utama yang harus dikuasai dalam menganalisis data.

Contoh :

```
import os
import pandas as pd
import numpy as np

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

Setelah langkah diatas tulis perintah train.head() kemudian akan muncul outputnya seperti dibawah ini :

	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

Maka data id penumpang,selamat,kelas penumpang,nama,jenis kelamin,umur,nomor ticket,tarif,kabin dan akan ditampilkan seperti diatas. Selanjutnya kita menulis perintah seperti dibawah ini :

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

Maka outputnya menjadi :

```
Train Shape: (891, 9)
Test Shape: (418, 9)
```

Setelah kita melakukan perintah diatas kita menulis perintah train.info() maka hasil outputnya seperti dibawah ini :

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
PassengerId    891 non-null int64
Survived       891 non-null int64
Pclass         891 non-null int64
Name           891 non-null object
Jenis Kelamin            891 non-null object
Age            714 non-null float64
SibSp          891 non-null int64
Parch          891 non-null int64
Ticket         891 non-null object
Fare           891 non-null float64
Cabin          204 non-null object
Embarked       889 non-null object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.6+ KB
```

Dari hasil data diatas didapat data sebanyak 12 kolom dimana dataset 891 memilikki item intenger sebanyak 5, object 3 dan floating 1,data 714 memilikki floating 1 kemudian 204 dan 889 memiliki 1 object. Selanjutnya kita tuliskan perintah test.info() maka hasil outputnya seperti dibawah ini :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
PassengerId    418 non-null int64
Pclass         418 non-null int64
Name           418 non-null object
Jenis Kelamin            418 non-null object
Age            332 non-null float64
SibSp          418 non-null int64
Parch          418 non-null int64
Ticket         418 non-null object
Fare           417 non-null float64
Cabin          91 non-null object
Embarked       418 non-null object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

Dari data diatas kita mendapat floating sebanyak 2,integer sebanyak 4 dan object sebanyak 5

2. Kamus Data

Pada tahap ini kita dapat melihat bahwa ada 12 kolom dalam kumpulan dataset selanjutnya disini kita akan menuliskan perintah train.head (10) maka output yang didapat akan seperti dibawah ini :

	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
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	21.0750	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

Dari hasil diatas dijelaskan bahwa data memilikki 10 kolom diantaranya berisi item passenger id,survived,Pclass,name,Sex,age,sib sp,parch,ticket,fare,cabin dan embarked. Pada survived jumlah survived 0 memilikki 5 dan 1 memilikki 5,pada item Pclass ticket class 3 mempunyai jumlah 6, ticket class 2 mempunyai 1 dan Ticket class 1 memilikki 3,pada item Jenis Kelamin didapat Laki-Laki5 dan Wanita5, kemudian pada Embarked jumlah penumpang S (Southampton) memilikki 7,C (Cherbourg) memilikki 2 dan Q (Queenstown) memilikki 1.

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

	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

Dari data diatas kita mendapat passengerid count 891,mean 446,dengan min 1dan max 891sedangkan survived rata-rata 0,38,min 0 dan max 1.

Setelah melakukan perintah diatas kemudian kita mengetik perintah test.describe() maka hasil seperti dibawah ini :

	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

Dari data diatas kita dapat melihat jumlah item count,mean.std,min,25%,50%,75% dan max passengerid pada mean didapat 1100.500000 dan max 1309.000000.

Selanjutnya kita tulis perintah train.isnull().max() dan didapat hasilnya seperti dibawah ini :

```
PassengerId            False
Survived               False
Pclass                 False
Name                   False
Jenis Kelamin                      False
Age                    True
SibSp                  False
Parch                  False
Ticket                 False
Fare                   False
Cabin                  True
Embarked               True
dtype: bool
```

Maka hasil yang dari data diatas didapat false sebanyak 9 dan True sebanyak 3

Setelah itu ketik perintah test.isnull().sum() dan hasil outputnya seperti dibawah ini :

```
PassengerId      0
Pclass           0
Name             0
Jenis Kelamin              0
Age             86
SibSp            0
Parch            0
Ticket           0
```

```
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

Dari hasil data diatas didapat 0 sebanyak 7,86 sebanyak 1,327 sebnyak 1 dan 1 sebanyak 1 sedangkan data type adalah integer 64.

3. Visualisasi Data menggunakan paket Matplotlib dan Seaborn

Pada visualisasi data kita akan mevisualisasi Bagan Batang untuk Fitur Kategorikal,Pclass,Seks ,SibSp (# saudara kandung dan pasangan),Parch (# orang tua dan anak),Memulai dan Kabin dengan mengetikan perintah seperti dibawah ini maka kita akan mendapatkan data Jenis Kelamin survived dan data Jenis Kelamin dead :

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

maka outputnya :

```
Survived :
Wanita   233
Laki-Laki     109
Name: Jenis Kelamin, dtype: int64
Dead:
Laki-Laki     468
Wanita    81
Name: Jenis Kelamin, dtype: int64
```

Dari data diatas kita mendapatkan jumlah penumpang bertahan/survived Wanita233 dan Laki-Laki109 sedangkan jumlah penumpang mati/Dead Wanita81 dan Laki-Laki468. Maka hasil bagan batang adalah :

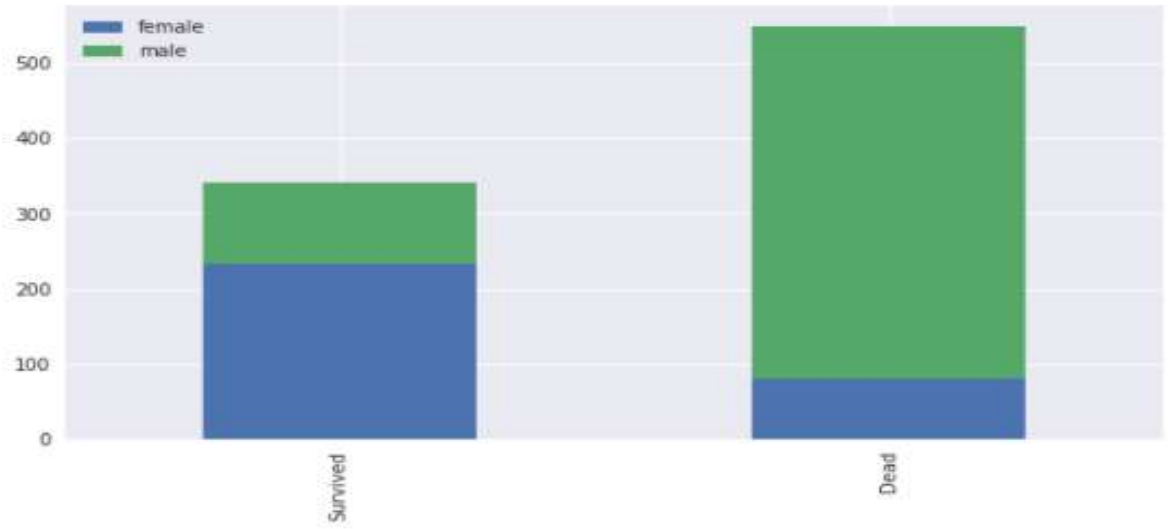


Diagram diatas menjelaskan bahwa data wanita pada survived mendominasi lebih banyak daripada data Laki-Laki sedangkan pada Dead data Laki-Laki lebih banyak daripada Wanita sehingga dapat dikatakan bahwa Wanita lebih mungkin bertahan daripada male.

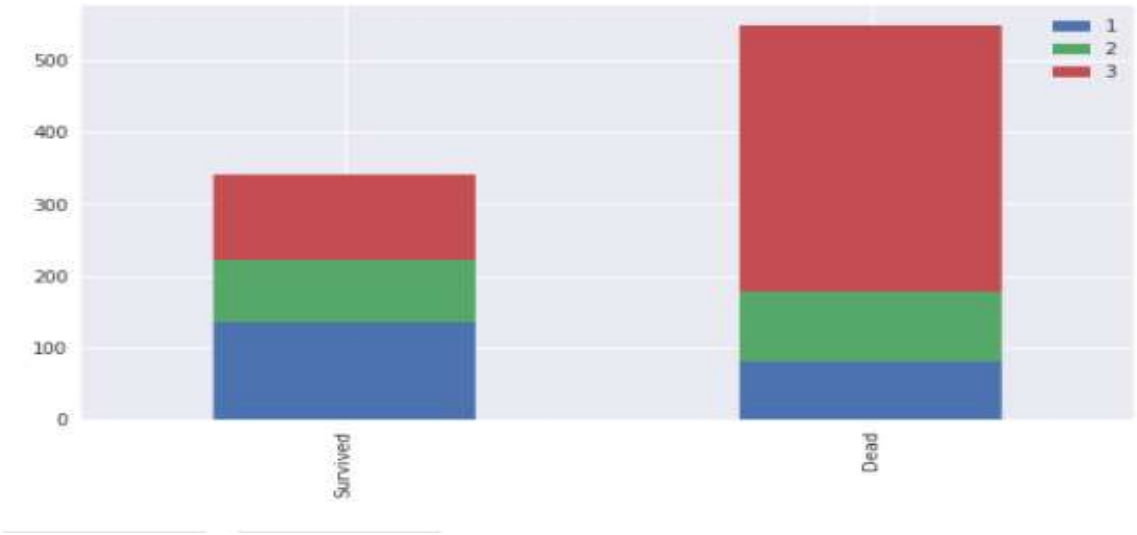
Selanjutnya kita akan menvisualisasikan data Pclass dengan menulis perintah :

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

Maka hasil ouputnya adalah

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

Pada hasil diatas didapat 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. Maka didapat Bar chart seperti dibawah ini :



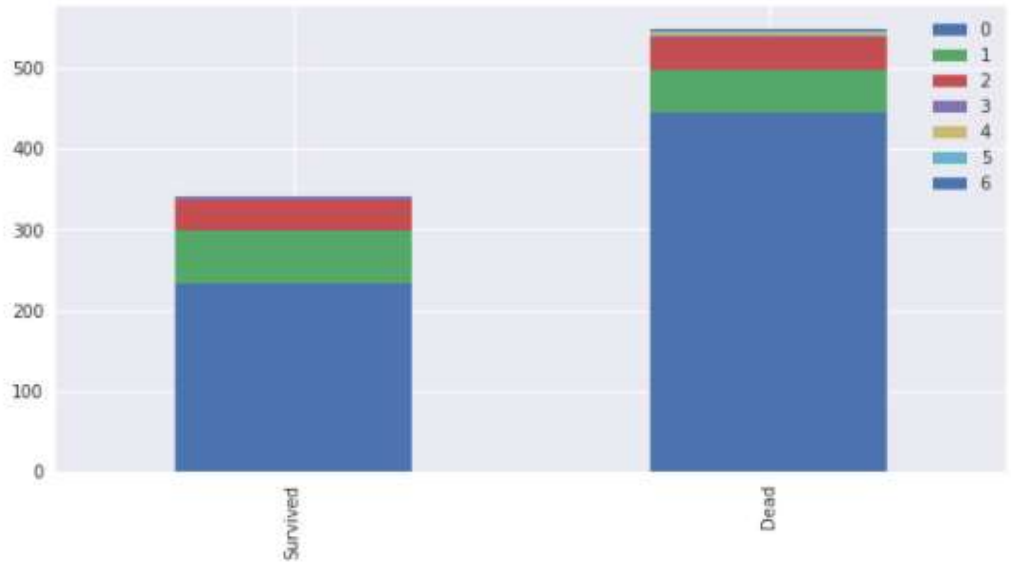
Dari bar chart diatas kita dapat melihat dead pada kelas 3 lebih banyak dari kelas 1 dan kelas 2 dan pada survived kelas 1 lebih banyak dari kelas 2 dan kelas 3. Setelah kita menvisualisasikan diatas selanjutnya kita akan melihat SibSp dengan menulis perintah :

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

Dan hasil outputnya seperti dibawah ini :

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

Hasil diatas menunjukan data survived 0 mempunyai jumlah besar yaitu 210 sedangkan Dead 0 memilikki 398. Maka bar chart dapat divisualisasikan seperti dibawah ini :

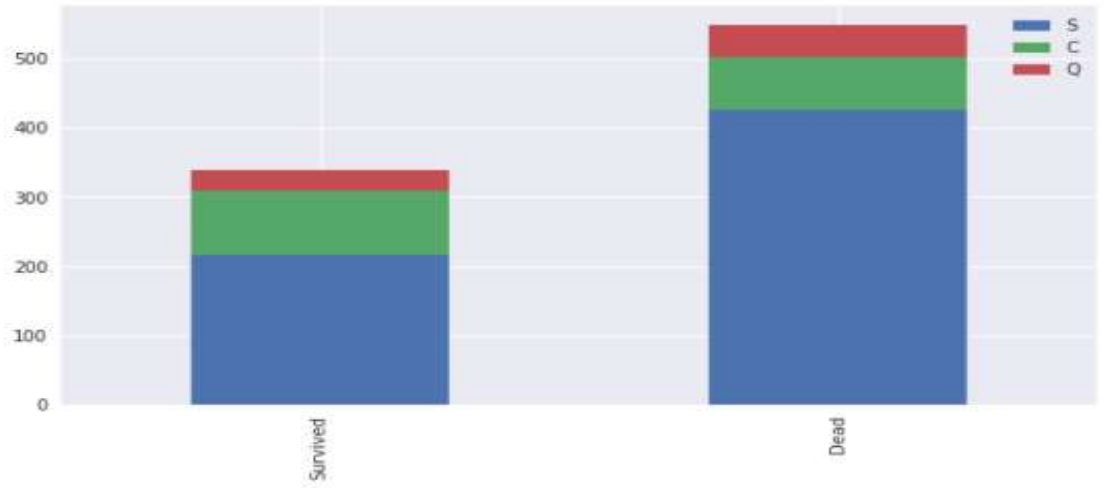


Bar Chart tersebut menegaskan bahwa ada lebih dari dua orang tua atau anak yang kemungkinan besar akan bertahan hidup. Bar Chart tersebut menegaskan bahwa seseorang yang sendirian di atas kapal kemungkinan besar akan meninggal. Selanjutnya kita akan melihat data embarked dengan menulis perintah:

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

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

Dari output diatas didapat bahwa survived S memilikki 217, C memilikki 93 dan Q memilikki 30 dan untuk Dead S memilikki 427,C memilikki 75 dan Q memilikki 47 sehingga bar chart nya seperti dibawah ini:



Bar chart 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 menggunakan pengetahuan domain data untuk membuat fitur (vektor fitur) . Vektor fitur adalah vektor numerik yang merepresentasikan fitur numerik suatu objek karena representasi seperti itu memfasilitasi pemrosesan dan analisis statistik. Seperti contoh dibawah ini dengan menulis perintah train.head() maka outputnya seperti dibawah ini :

	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

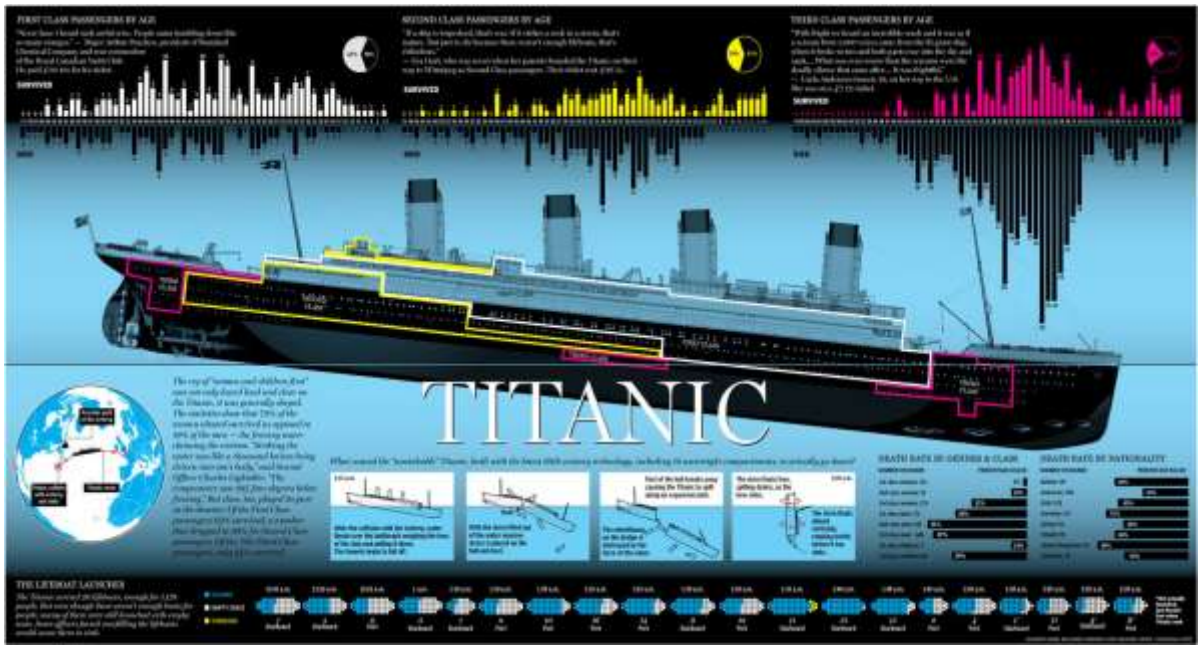
Dari data diatas kita dapat mengetahui bahwa jumlah kolom ada 5 baris dan dimana usia penumpang titanic paling tua 38 dan paling muda 22, namun tidak hanya itu disitu juga kita dapat melihat Jenis kelamin penumpang yaitu Laki-laki ada 2 dan Wanita ada 3.

4.1 Bagaimana Titanic tenggelam?

Disini kita akan menganilisis bagaimana kapal titanic bisa tenggelam dengan menggunakan perintah :

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

Maka outputnya :



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 menulis perintah train.head(12), maka outputnya adalah :

	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
10	1	3	1	0.0	0.0	2.4	0	1	0.8
11	1	1	1	3.0	1.0	0.8	0	1	0.0

Dari data diatas kita dapat kesimpulan bahwa jumlah ukuran kelurga paling besar mencapai 1.6 dan 0.0

1) Title

Pada Title kita menggabungkan kumpulan data dengan menggunakan perintah :

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

Kemudian kita menuliskan perintah train['Title'].value_counts() maka hasil outputnya :

```
Mr          517
Miss        182
Mrs         125
Master       40
Dr            7
Rev           6
Col           2
Major         2
Mlle          2
Capt          1
Sir           1
Don           1
Countess      1
Ms            1
Lady          1
Mme           1
Jonkheer      1
Name: Title, dtype: int64
```

Dari data diatas kita bisa melihat berapa banyak penumpang berdasarkan Title, Title Mr sebanyak 517 merupakan jumlah terbanyak dari julah Title yang lainya.

Untuk selanjutnya kita akan menghitung Title dengan menuliskan perintah test['Title'].value_counts(), maka ouput yang didapat adalah :

```
Mr        240
Miss       78
Mrs        72
Master     21
Rev         2
Col         2
Dr          1
Ms          1
Dona        1
Name: Title, dtype: int64
```

Maka disini kita dapat melihat 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 :

```
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()

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title	FamilySize
0	892	3	0	2.0	0	0	330911	0.0	2.0	2	0	0.0
1	893	3	1	3.0	1	0	363272	0.0	2.0	0	2	0.4
2	894	2	0	3.0	0	0	240276	0.0	2.0	2	0	0.0
3	895	3	0	2.0	0	0	315154	0.0	2.0	0	0	0.0
4	896	3	1	1.0	1	1	3101298	0.0	2.0	0	2	0.8

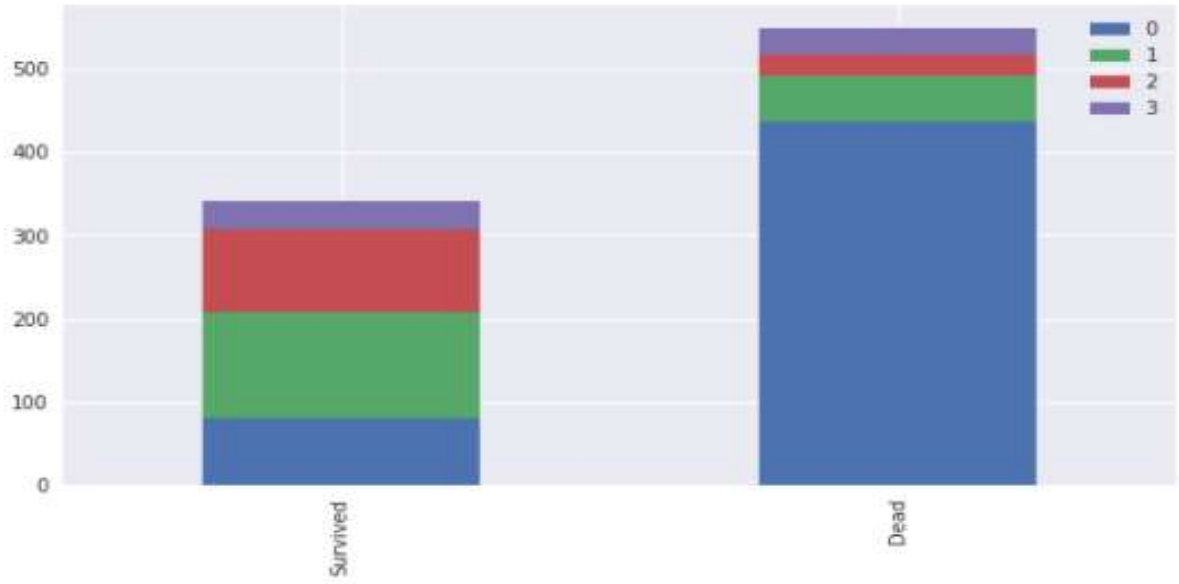
Dari data diatas kita bisa melihat di Title bahwa Mr = 0 mempunyai jumlah 3 dan Mrs = 2 mempunyai jumlah 2.

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

	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

Dari data diatas menegaskan bahwa Mr. berjumlah 3 dan paling tua berumur 62 tahun dan Mrs paling tua berumur 47 tahun. Sehingga kita dapat membuat bar chart nya seperti dibawah ini dengan menuliskan perintah

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



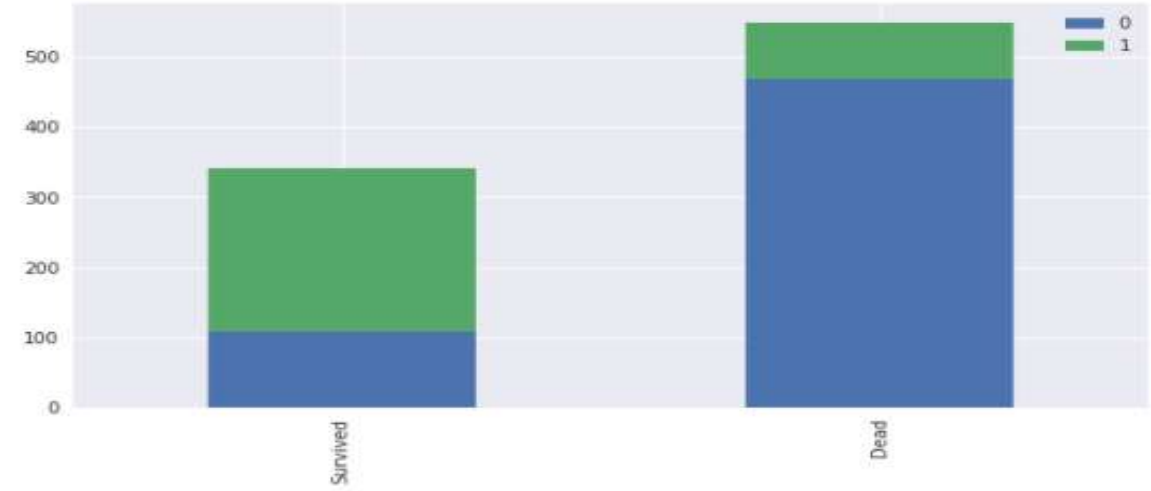
Dari Bar chart diatas menegaskan bahwa Mrs = 1 banyak yang bertahan dan untuk banyak yang meninggal adalah Mr = 0 dalam tragedi titanic.

2) Jenis Kelamin/Sex

Disini kita akan memetakan berdasarkan jenis kelamin penumpang tragedi titanic dengan menuliskan perintah :

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

dari data diatas kita dapat melihat laki-laki = 0 dan wanita = 1. Kemudian kita melihat dengan menggunakan bar Chart dengan menulis perintah bar_chart ('Sex') maka outputnya adalah :



Dari Bar chart menegaskan bahwa jumlah Laki-laki lebih banyak yang meninggal dan jumlah wanita lebih banyak yang bertahan. Setelah itu kita tuliskan perintah test.head() dan outputnya dapat dilihat seperti dibawah ini :

	PassengerId	Pclass	Sex	Age	Fare	Cabin	Embarked	Title	FamilySize
0	892	3	0	2.0	0.0	2.0	2	0	0.0
1	893	3	1	3.0	0.0	2.0	0	2	0.4
2	894	2	0	3.0	0.0	2.0	2	0	0.0
3	895	3	0	2.0	0.0	2.0	0	0	0.0
4	896	3	1	1.0	0.0	2.0	0	2	0.8

Dari data diatas kita dapat melihat jumlah Laki-laki = 0 adalah 3 sedangkan Wanita berjumlah 2.

3) Usia/Age

Disini kita akan menggroupkan berdasarkan usia menggunakan median dengan menuliskan perintah

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

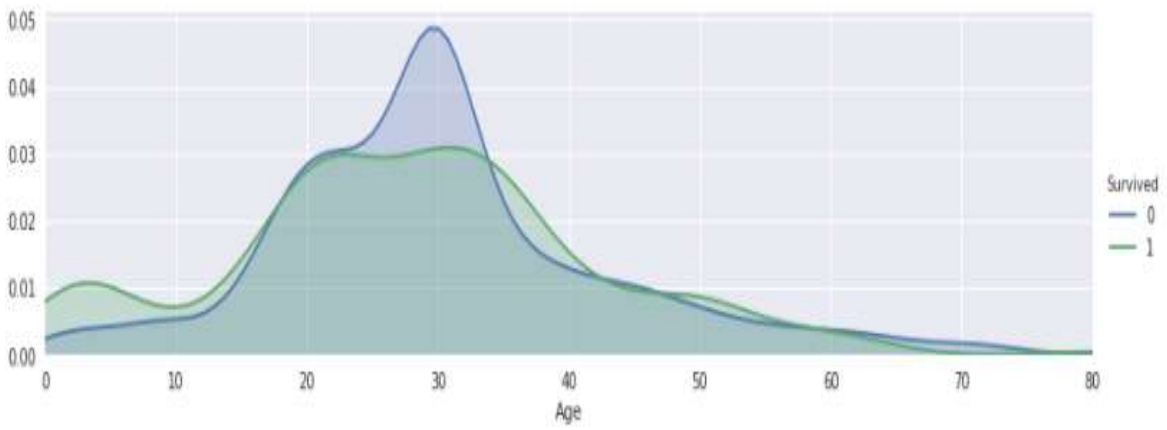
kemudian kita tulis perintah :

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

Dari data diatas kita dapat melihat ada 5 kolom terdiri usia paling tua adalah 38 dan usia paling muda adalah 22.Kemudian kita tuliskan perintah seperti dibawah ini :

```
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.show()
```

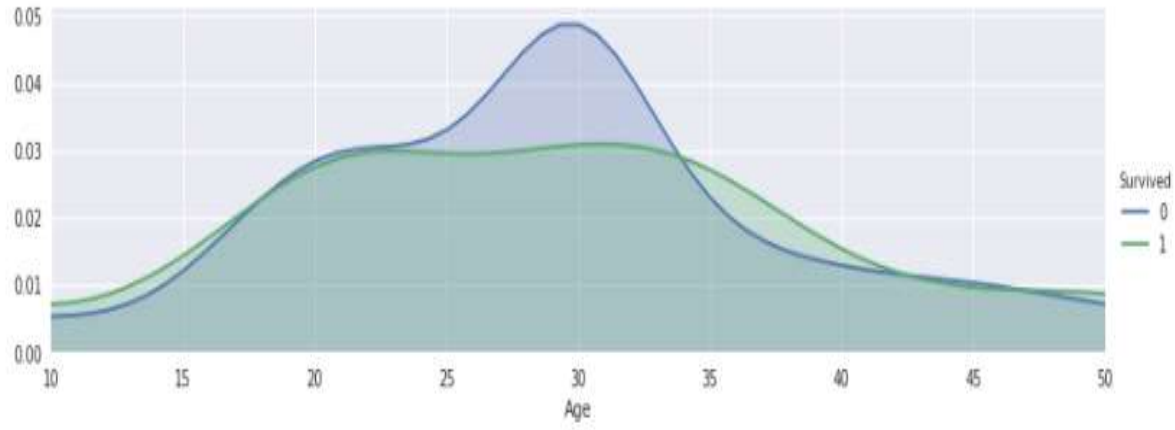


Dari data diatas menegaskan bahwa jumlah rata-rata penumpang tragedi titanic laki-laki dan wanita yang paling banyak bertahan dan meninggal adalah berkisar berusia 20 sampai 30 an.

Setelah itu kita akan melihat jumlah penumpang yang bertahan pada tragedi titanic yang berumur dari 10 sampai 50 tahun dengan menuliskan perintah :

```
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)
```

```
(10, 50)
```



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

4.2 Pengelompokan/Binning

Selanjutnya kita akan mengelompokan tragedi titanic berdasarkan usia ke variable katagori anak-anak,muda,dewasa,paruh baya dan senior dengan menulis perintah train.head() maka hasil outputny adalah :

	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

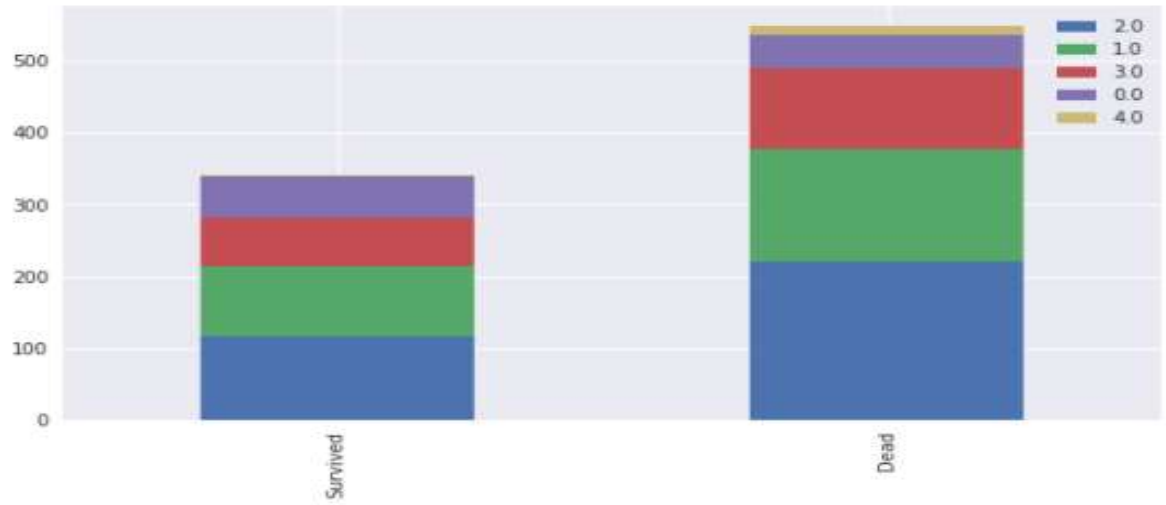
Dari data diatas bahwa usia paling muda adalah 22 tahun dan usia paling tua adalah 38 tahun.
Kemudian kita kan memetakan berdasarkan usia dengan menulis perintah seperti dibawah ini :

```
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
# for dataset in train_test_data:
# dataset.loc[]
# train[train['Age'].isin([23])]
```

Dari data diatas kita mengelompokan menjadi 5 kelompok dimana 0 sampai 16 anak-anak, 16 sampai 26 adalah muda 26 sampai 36 adalah dewasa, 36 sampai 62 adalah paruh baya dan 62 keatas adalah senior.
Setelah itu kita membuat bar chart dengan menuliskan perintah

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

Maka outputnya :

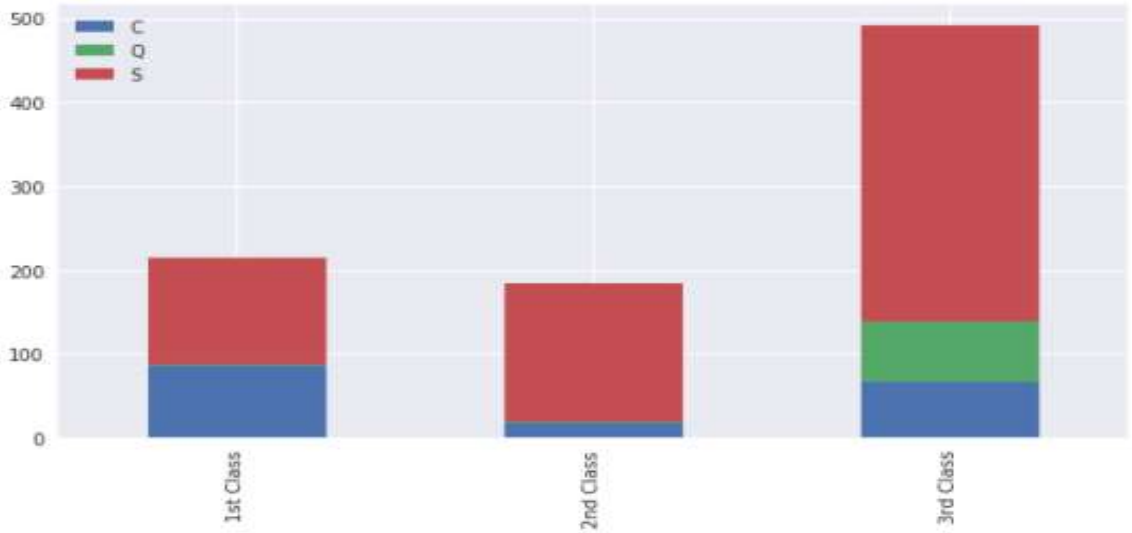


Dari bar chart diatas menegaskan bahwa jumlah usia yang selamat dan meninggal dari tragedi titanic yang paling banyak adalah berkisar antara 26 sampai 36 tahun.

Setelah langkah diatas kemudian kita memetakan berdasarkan Pclass dengan menuliskan perintah seperti dibawah ini :

```
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)
```

Dari perintah 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 output bar chart yang dihasilkan adalah :



Dari bar chart diatas dijelaskan bahwa penumpang tragedi titanic Pclass 1,2 dan 3 banyak berasal dari S (Southampton).

```
Pclass1:
S    127
C     85
Q      2
Name: Embarked, dtype: int64
Pclass2:
S    164
C     17
Q      3
Name: Embarked, dtype: int64
Pclass3:
S    353
Q     72
C     66
Name: Embarked, dtype: int64
```

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 menuliskan perintah seperti dibawah ini :

```
# 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)
```

maka outputnya adalah :

	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

Dari data diatas dapat dilihat bahwa nilai tarif paling besar adalah 71.2833 dan paling rendah adalah 7.2500. kemudian kita akan membagi tarif menjadi 4 kelompok dengan menuliskan perintah seperti dibawah ini :

```
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 :

	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

Dari hasil output diatas dijelaskan bahwa 0 = tarif kurang dari 17 ada 3 dan 2 = tarif 30 sampai 100 ada 2.

Selanjutnya kita akan memetakan tragedi titanic dengan cara melihat dataset cabin dengan menggunakan perintah :

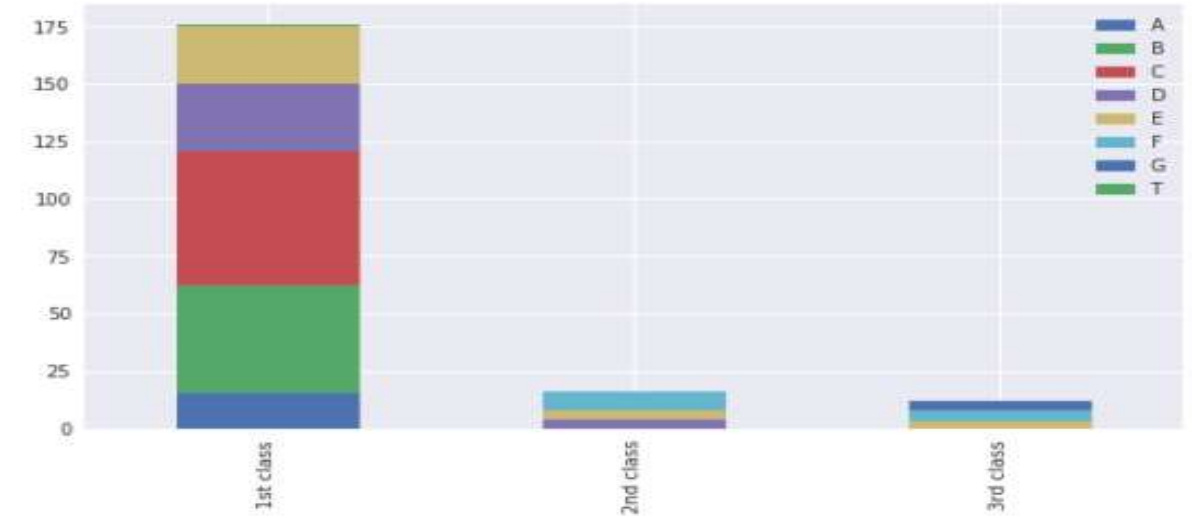
```
for dataset in train_test_data:
    dataset['Cabin'] =  dataset['Cabin'].str[:1]
```

Selanjutnya kita ketikan perintah seperti dibawah ini :

```
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 seperti dibawah ini :

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb25e1504e0>
```




```
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.

Setelah itu kita akan memetakan berdasarkan familysize dengan mengetikan perintah sebagai berikut:

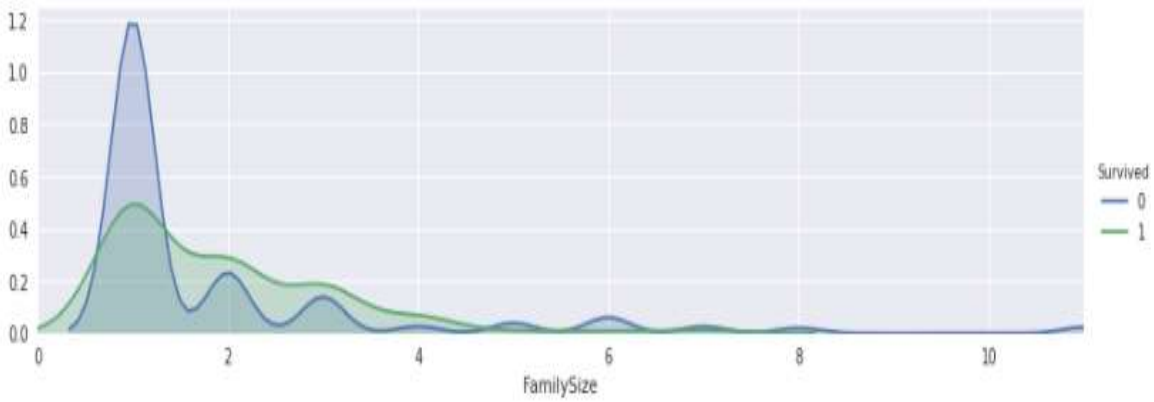
```
train["FamilySize"] = train["SibSp"] + train["Parch"] + 1
test["FamilySize"] = test["SibSp"] + test["Parch"] + 1
```

Dan kita tuliskan lagi perintahseperti dibawah ini :

```
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)
```

Maka hasil outputnya adalah :

(0, 11.0)




```
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 :

	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 diatas menjelaskan bahwa cabin 2 berjumlah 3 dan cabin 0.8 berjumlah 2 dan Pclass 3 ada 3 dan Pclass 1 ada 2.

5. Modelling

Setelah itu kita impor modul Pengklasifikasi dari lalu ketikan perintah :

```
# 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
```

Kemudian ketikan perintah train.info() maka hasil outputnya adalah :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
Survived      891 non-null int64
Pclass        891 non-null int64
Sex           891 non-null int64
Age           891 non-null float64
Fare          891 non-null float64
Cabin         891 non-null float64
Embarked      891 non-null int64
Title         891 non-null int64
FamilySize    891 non-null float64
dtypes: float64(4), int64(5)
memory usage: 62.7 KB
```

Dari hasil data diatas didapatkan dari 9 kolom data masukan atau data yang diinput 891 kita dapat melihat bahwa data integer ada 5 dan floating ada 4.

6. Cross Validation(k-fold)

Pada tahap ini kita akan mengimpor sklearn.model_selection ke kfold dengan mengetikan perintah sebagai berikut :

```
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)
```

Kemudian tuliskan perintah seperti dibawah ini :

```
#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()
```

Maka outputnya adalah :

```
Score of Model 0 : 82.6
Score of Model 1 : 79.58
Score of Model 2 : 80.7
Score of Model 3 : 78.78
Score of Model 4 : 83.5
Score of Model 5 : 79.01
Score of Model 6 : 81.25
Score of Model 7 : 81.03
Score of Model 8 : 80.47
```

Maka hasil nilai model support Machinenya adalah nilai score model 4 : 83.5

7. **Testing**

Pada langkah testing ini tuliskan perintah seperti dibawah ini :

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

Lalu ketikan perintah sebagai berikut :

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

Maka outputnya adalah :

	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

Maka hasil Survived Column yang telah diprediksi menggunakan Algoritma Support Vector Machine ini terdapat 5 kolom dimana Pclass yan memilikki nilai 3 ada 4 dan 2 ada 1 sedangkan Title yang memilikki 2 ada 2 dan 0 ada 3.