MINI PROJECT

BIG DATA ANALYTICS

“Prediksi Korban Bencana Titanic dengan Metode KNN”

Disusun Oleh :

Raden Isnawan Argi Aryasatya (195410257 / kelas IF-5)

Satria Dwi Hartanto (195410229 / kelas IF-5)

Prodi Informatika

Dataset bisa didapatkan dari salah satu dari dua sumber berikut:

1. https://github.com/datasciencedojo/datasets/blob/master/titanic.csv
2. https://www.kaggle.com/c/titanic

Note*: Isi datasetnya dari kedua sumber sebenarnya sama saja. Perbedaannya adalah dataset yang di link kaggle sudah dipisah file data latih dan data uji nya.

Overview:

Tenggelamnya kapal titanic merupakan salah satu bencana paling terkenal di dalam sejarah umat manusia. Pada tanggal 15 April 1912, kapal Titanic yang dianggap tidak bisa tenggelam sedang melakukan pelayaran pertamanya. Sayang sekali, kapal Titanic tersebut justru malah menabrak bongkahan es besar dan akhirnya kapal tersebut tenggelam. Tidak semua korban bisa selamat karena jumlah *lifeboat* yang sedikit. Hasilnya, dari 2224 kru dan penumpang, hanya ada 722 yang selamat.

Walau ada elemen keberuntungan yang mempengaruhi selamat atau tidaknya korban bencana Titanic tersebut, jika dilihat-lihat beberapa kelompok lebih cenderung selamat daripada beberapa kelompok lain, dan *vice versa*. Pada mini project ini, kami kami membuat model prediktif menggunakan KNN untuk menjawab beberapa pertanyaan seputar topik tersebut.

Pada mini project ini, kita akan menggunakan dua dataset bernama ‘train.csv’ dan ‘test.csv’.

file ‘train.csv’ berisi detail dari subset milik penumpang berjumlah 891 baris. File tersebut akan menampilkan selamat atau tidaknya penumpang pada kolom ‘survived’, yang pada ilmu Machine Learning biasa dinamai dengan ‘ground truth’.

File ‘test.csv berisi informasi yang hampir sama dengan train.csv, tetapi tidak menampilkan selamat atau tidaknya penumpang.

Tugas utama kami di mini project ini adalah: dengan memproses dan menganalisis data di file train.csv, kami akan memprediksi *outcome* pada kolom ‘survived’ di file ‘test.csv’ untuk menentukan selamat atau tidaknya 418 penumpang yang ada di dalam file itu dengan metode KNN.

Langkah-langkah pengerjaan:

1. Mengimport library/pustaka yang diperlukan

```
In [25]: import matplotlib.pyplot as plt
         import seaborn as sns
         import pandas as pd
         import numpy as np

         from sklearn.model_selection import train_test_split
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.metrics import accuracy_score
```

2. Load dataset titanic dengan menggunakan metode read_csv milik library pandas. Pertama-tama kita lihat shape dataset tersebut. Angka 0 adalah untuk menghitung jumlah data secara vertikal (row/baris), angka 1 adalah untuk menghitung jumlah data secara horizontal (column/kolom).

```
In [26]: data_latih = pd.read_csv('train.csv')

         print('shape dataset di file train.csv : %s penumpang dengan %s kolom/variabel'%
               (data_latih.shape[0],data_latih.shape[1]))
```

```
shape dataset di file train.csv : 891 penumpang dengan 12 kolom/variabel
```

3. Kita lihat 10 baris pertama dengan fungsi ***head(10)***

```
data_latih.head(10)
```

	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

Keterangan:

- PassengerId = id setiap passenger yang dimulai dari angka 1 dan seterusnya
- Survived = Variabel ini menunjukan selamat atau tidaknya penumpang. Variabel ini merupakan variabel target untuk kita prediksi nilainya. Variabel ini bertipe data biner. Angka 0 berarti tidak selamat, angka 1 berarti selamat.
- Pclass = Kelas tiket setiap penumpang. 1 berarti kelas atas, 2 berarti kelas menengah, 3 berarti kelas bawah.
- Name = nama lengkap penumpang
- Sex = jenis kelamin penumpang
- Age = umur penumpang
- Sibsp = jumlah saudara kandung atau pasangan dari setiap penumpang
- Parch = jumlah orang tua atau anak dari setiap penumpang
- Ticket = nomor tiket
- Fare = tarif atau biaya tiket

- Cabin = nomor kabin
- Embarked = Tempat keberangkatan atau bisa disebut dengan pelabuhan (C = Cherbourg, Q = Queenstown, S = Southampton)

4. Dengan salah satu fungsi pandas yaitu ***info()***, cetak informasi tentang DataFrame berupa nama kolom, tipe data setiap variabel (dtype), jumlah data bernilai non-null, dan memory usage (penggunaan memori)

```
In [28]: data_latih.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
```

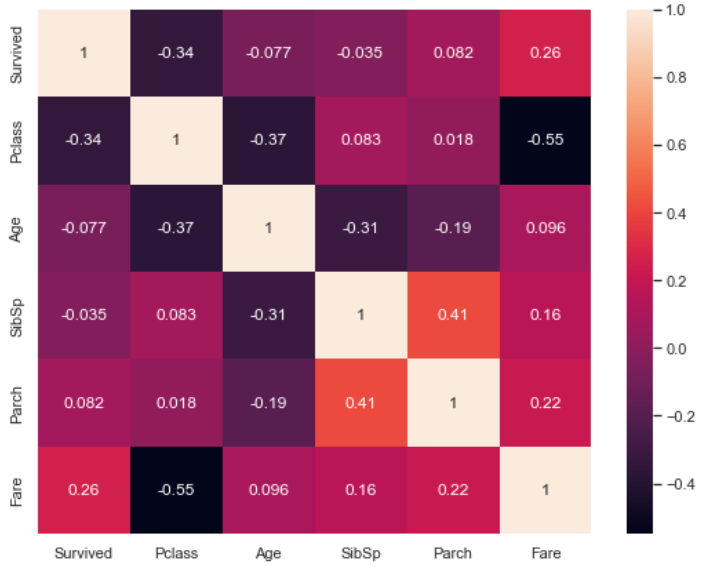
5. Bisa kita lihat di nomer 4 bahwa kita memiliki beberapa kolom yang nilainya hilang. Seharusnya, semua kolom memiliki 891 data. Tetapi di kolom Age, Cabin, dan Embarked ada beberapa *missing value* sehingga datanya tidak mencapai angka 891. Maka dari itu, kami periksa *null value* menggunakan fungsi isnull() dan sum().

```
In [29]: data_latih.isnull().sum()

Out[29]: PassengerId      0
         Survived         0
         Pclass           0
         Name             0
         Sex              0
         Age            177
         SibSp            0
         Parch            0
         Ticket           0
         Fare             0
         Cabin          687
         Embarked         2
         dtype: int64
```

6. Kita analisis data untuk melihat variabel apa saja yang memiliki efek signifikan (penting) untuk memprediksi *value* dari variabel target (survived). Untuk melakukan itu, kita menggunakan sebuah heatmap untuk melihat korelasi di antara parameter dan variabel target.

```
In [32]: heatmap = sns.heatmap(data_latih[["Survived", "Pclass", "Age", "SibSp", "Parch", "Fare"]].corr(),
                               annot = True)
         sns.set(rc={'figure.figsize':(10,8)})
```



Penjelasan:

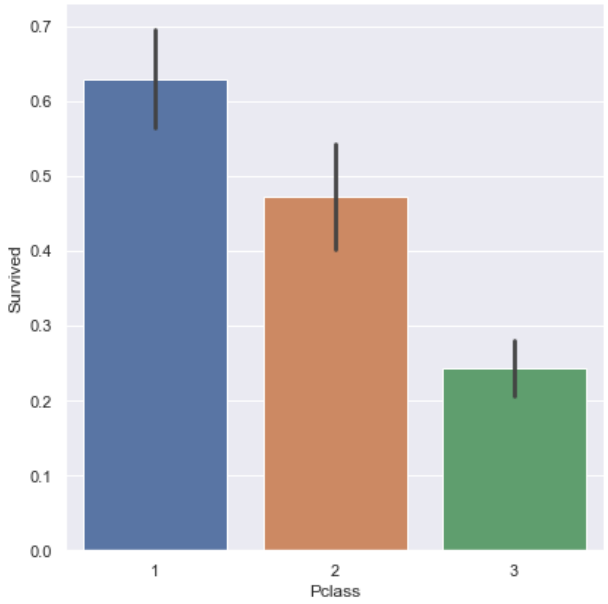
Nilai-nilai yang ditampilkan di atas bisa kita sebut dengan *correlation values*. Kami mencari korelasi di antara kolom “survived” dengan variabel-variabel lain yang berbentuk angka yang kita-kira memiliki pengaruh atas selamat atau tidaknya penumpang Titanic. Pada grafik tersebut, semakin cerah warna kotak nilai, maka semakin dekat korelasinya dengan kolom “survived”. Bisa kita lihat nilai dengan kotak paling gelap adalah Pclass yang memiliki nilai -0,34 yang berarti variabel tersebut tidak memiliki korelasi apapun dengan keselamatan penumpang. Di bawahnya, ada Age dan SibSp yang memiliki nilai masing-masing -0.077 dan -0.035 yang juga berarti tidak memiliki kaitan dengan keselamatan penumpang. Parch memiliki nilai 0.082 yang berarti memiliki sedikit katian dengan keselamatan penumpang. Terakhir, ada variabel di kotak dengan warna paling cerah bernama Fare dengan nilai 0.26 yang berarti memiliki korelasi yang erat dengan keselamatan penumpang (kolom “survived”). Dengan kata lain, semakin tinggi nilai Fare, semakin tinggi tingkat keselamatan penumpang.

7. Sekarang, kami menganalisis beberapa variabel yang di heatmap tadi memiliki nilai kecil (Pclass, Age, SibSp). Selain itu, kami juga menganalisis satu variabel yang tidak disertakan di heatmap tadi (Sex). Analisa ini untuk membuktikan jika variabel-variabel tersebut masih memiliki pengaruh atas tingkat keselamatan penumpang.

Pclass

(grafik di halaman selanjutnya)

```
pclassplot = sns.catplot(x = "Pclass", y="Survived", data = data_latih, kind="bar", height = 6)
```



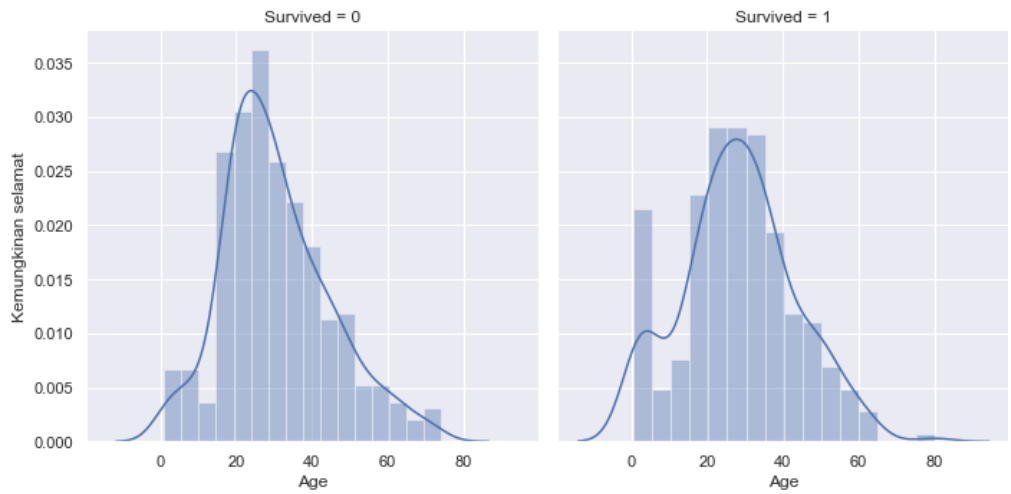
Penjelasan:

Grafik tersebut menampilkan data berupa tingkat keselamatan penumpang jika dilihat dari perspektif variabel Pclass. Penumpang di class 1 memiliki tingkat keselamatan paling tinggi jika dibandingkan dengan penumpang di class 2 dan class 3. Penumpang di class 2 memiliki tingkat keselamatan yang lebih tinggi dibandingkan dengan penumpang di class 3. Penumpang di class 3 memiliki tingkat keselamatan yang paling rendah.

Age

```
In [35]: ageplot = sns.FacetGrid(data_latih, col="Survived", height = 5)
         ageplot = ageplot.map(sns.distplot, "Age")
         ageplot = ageplot.set_ylabels("Kemungkinan selamat")
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futu
nd will be removed in a future version. Please adapt your code to use either `d
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Futu
nd will be removed in a future version. Please adapt your code to use either `d
exibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

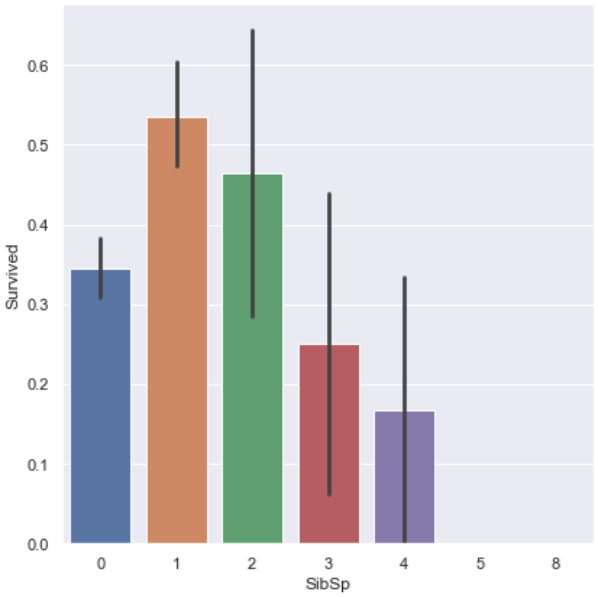


Penjelasan:

Bisa kita lihat bahwa orang yang memiliki usia lebih tua, lebih mungkin untuk tidak selamat.

SibSp

```
data_latih['SibSp'].unique()
bargraph_sibsp = sns.catplot(x = "SibSp", y = "Survived", data = data_latih, kind="bar", height = 6)
```

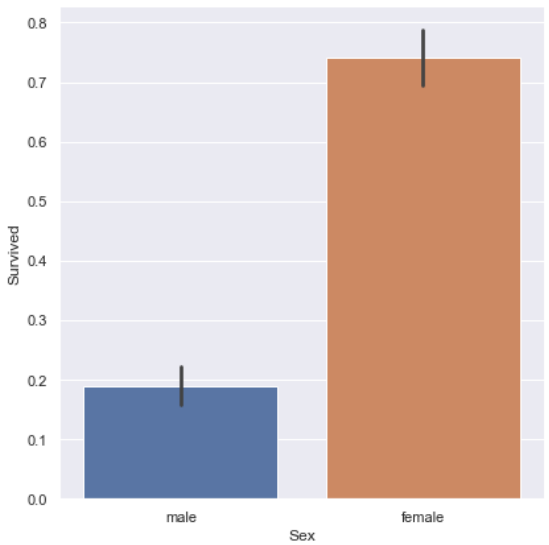


Penjelasan:

Penumpang dengan saudara berjumlah 1 dan 2 memiliki tingkat keselamatan yang paling tinggi. Sementara penumpang dengan jumlah saudara lebih dari 2 memiliki tingkat keselamatan yang lebih rendah.

Sex

```
In [38]: sexplot = sns.catplot(x = "Sex", y="Survived", data = data_latih, kind="bar", height = 6)
```



Penjelasan:

Laki-laki (male) memiliki kemungkinan selamat yang lebih rendah jika dibandingkan dengan perempuan (female).

8. Pada tahap ini kami telah masuk ke tahap data preprocessing. Pertama, kami lihat berapa banyak *missing value* (null) pada setiap kolom/variabel. Kami gunakan fungsi isnull() dan sum().

```
In [39]: data_latih.isnull().sum()

Out[39]: PassengerId      0
         Survived         0
         Pclass           0
         Name             0
         Sex              0
         Age            177
         SibSp            0
         Parch            0
         Ticket           0
         Fare             0
         Cabin          687
         Embarked         2
         dtype: int64
```

   Penjelasan:
   Bisa kita lihat bahwa ada 177 *missing values* di kolom Age, 687 *missing values* di kolom Cabin, dan 2 *missing values* di kolom Embarked.

9. Kami isi *missing values* yang ada di dalam kolom Age. Untuk melakukan itu, kami menggunakan nilai random yang dihasilkan dari range perhitungan: rata-rata - standar deviasi, rata-rata + standar deviasi. Sehingga kita mendapatkan nilai distribusi data normal.

```
In [40]: mean = data_latih["Age"].mean()
         std = data_latih["Age"].std()

         random_age = np.random.randint(mean-std, mean+std, size = 177)
         age_slice = data_latih["Age"].copy()
         age_slice[np.isnan(age_slice)] = random_age
         data_latih["Age"] = age_slice

         data_latih.isnull().sum()

Out[40]: PassengerId      0
         Survived         0
         Pclass           0
         Name             0
         Sex              0
         Age              0
         SibSp            0
         Parch            0
         Ticket           0
         Fare             0
         Cabin          687
         Embarked         2
         dtype: int64
```

10. Kami isi *missing values* yang ada di dalam kolom Embarked dengan mengisi value NaN dengan value yang paling sering muncul di kolom Embarked tersebut

```
In [41]: data_latih = data_latih.fillna(data_latih['Embarked'].value_counts().index[0])
         data_latih.isnull().sum()
```

```
Out[41]: PassengerId    0
         Survived       0
         Pclass         0
         Name           0
         Sex            0
         Age            0
         SibSp          0
         Parch          0
         Ticket         0
         Fare           0
         Cabin          0
         Embarked       0
         dtype: int64
```

11. Variabel/kolom PassengerId, Ticket No., Name, dan Cabin tidak berpengaruh terhadap tingkat keselamatan penumpang Titanic. Maka dari itu, kita drop kolom-kolom tersebut supaya dataset terlihat lebih ‘clean’.

```
In [42]: drop_kolom = ["PassengerId", "Ticket", "Cabin", "Name"]
         data_latih.drop(drop_kolom, axis=1, inplace=True)
         data_latih.head(10)
```

Out[42]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S
5	0	3	male	21.0	0	0	8.4583	Q
6	0	1	male	54.0	0	0	51.8625	S
7	0	3	male	2.0	3	1	21.0750	S
8	1	3	female	27.0	0	2	11.1333	S
9	1	2	female	14.0	1	0	30.0708	C

12. Setelah itu kami convert nilai bertipe kategori (huruf) menjadi nilai bertipe numerik, karena model machine learning hanya mengenali nilai numerik. Kolom yang datanya kita ubah adalah kolom Sex dan Embarked. Di kolom Sex, kami mengganti male dengan nilai 0 dan femal dengan nilai 1. Di kolom Embarkedm kami mengganti S dengan 0, C dengan 1, dan Q dengan 2.

```
In [43]: jenis_kelamin = {"male":0, "female":1}
         data_latih["Sex"] = data_latih["Sex"].map(jenis_kelamin)

         pelabuhan = {"S":0, "C":1, "Q":2}
         data_latih["Embarked"] = data_latih["Embarked"].map(pelabuhan)

         data_latih.head(10)
```

Out[43]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	0	22.0	1	0	7.2500	0
1	1	1	1	38.0	1	0	71.2833	1
2	1	3	1	26.0	0	0	7.9250	0
3	1	1	1	35.0	1	0	53.1000	0
4	0	3	0	35.0	0	0	8.0500	0
5	0	3	0	21.0	0	0	8.4583	2
6	0	1	0	54.0	0	0	51.8625	0
7	0	3	0	2.0	3	1	21.0750	0
8	1	3	1	27.0	0	2	11.1333	0
9	1	2	1	14.0	1	0	30.0708	1

13. Kami menggunakan fungsi Sklearn yaitu ***train_test_split*** yang berguna untuk split (membagi) array data menjadi dua subset yaitu data untuk training (latih) dan data untuk testing (uji)

```
In [44]: df_train_x = data_latih[['Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']]
         df_train_y = data_latih[['Survived']]
         x_train, x_test, y_train, y_test = train_test_split(df_train_x, df_train_y, test_size=0.20, random_state=42)
```

14. Kemudian kami aplikasikan model yang sudah kita buat tersebut ke dalam algoritma klasifikasi K-Nearest Neighbors, yang menghasilkan akurasi 83.99%

```
In [48]: knn = KNeighborsClassifier(n_neighbors = 3)
         knn.fit(x_train, y_train)
         Y_pred = knn.predict(x_test)
         knn_akurasi = round(knn.score(x_train, y_train) * 100, 2)
         print("akurasi=",knn_akurasi)

         akurasi= 83.99
```

15. Sekarang kami membuat prediksi final dengan model Machine Learning yang telah kita buat tadi. Pertama kami membuka file test.csv yang berisi data-data uji lalu menampilkan isinya dengan ***pd.read_csv***

```
In [49]: data_uji = pd.read_csv('test.csv')
         data_uji.head(10)
```

Out[49]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
5	897	3	Svensson, Mr. Johan Cervin	male	14.0	0	0	7538	9.2250	NaN	S
6	898	3	Connolly, Miss. Kate	female	30.0	0	0	330972	7.6292	NaN	Q
7	899	2	Caldwell, Mr. Albert Francis	male	26.0	1	1	248738	29.0000	NaN	S
8	900	3	Abrahim, Mrs. Joseph (Sophie Halaut Easu)	female	18.0	0	0	2657	7.2292	NaN	C
9	901	3	Davies, Mr. John Samuel	male	21.0	2	0	A/4 48871	24.1500	NaN	S

16. Cek informasi mengenai dataset tersebut dengan fungsi ***info()***

```
In [50]: data_uji.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
```

17. Cek missing value di setiap kolom/variabel

```
In [51]: data_uji.isnull().sum()

Out[51]: PassengerId      0
         Pclass           0
         Name             0
         Sex              0
         Age             86
         SibSp            0
         Parch            0
         Ticket           0
         Fare             1
         Cabin          327
         Embarked         0
         dtype: int64
```

18. Isi random *missing values* di kolom Age dan di kolom Fare

```
In [52]: mean = data_uji["Age"].mean()
         std = data_uji["Age"].std()
         random_age = np.random.randint(mean-std, mean+std, size = 86)
         age_slice = data_uji["Age"].copy()
         age_slice[np.isnan(age_slice)] = random_age
         data_uji["Age"] = age_slice

         data_uji['Fare'].fillna(data_uji['Fare'].mean(), inplace=True)
         data_uji.isnull().sum()

Out[52]: PassengerId      0
         Pclass           0
         Name             0
         Sex              0
         Age              0
         SibSp            0
         Parch            0
         Ticket           0
         Fare             0
         Cabin          327
         Embarked         0
         dtype: int64
```

19. Drop kolom-kolom yang tidak digunakan (PassengerId, Ticket, Cabin, Name)

In [53]:
```
drop_kolom = ["PassengerId", "Ticket", "Cabin", "Name"]
data_uji.drop(drop_kolom, axis=1, inplace=True)
data_uji.head(10)
```

Out[53]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	male	34.5	0	0	7.8292	Q
1	3	female	47.0	1	0	7.0000	S
2	2	male	62.0	0	0	9.6875	Q
3	3	male	27.0	0	0	8.6625	S
4	3	female	22.0	1	1	12.2875	S
5	3	male	14.0	0	0	9.2250	S
6	3	female	30.0	0	0	7.6292	Q
7	2	male	26.0	1	1	29.0000	S
8	3	female	18.0	0	0	7.2292	C
9	3	male	21.0	2	0	24.1500	S

20. Ubah data huruf menjadi data numerik

In [54]:
```
jenis_kelamin = {"male":0, "female":1}
data_uji["Sex"] = data_uji["Sex"].map(jenis_kelamin)

pelabuhan = {"S":0, "C":1, "Q":2}
data_uji["Embarked"] = data_uji["Embarked"].map(pelabuhan)

data_uji.head(10)
```

Out[54]:

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	0	34.5	0	0	7.8292	2
1	3	1	47.0	1	0	7.0000	0
2	2	0	62.0	0	0	9.6875	2
3	3	0	27.0	0	0	8.6625	0
4	3	1	22.0	1	1	12.2875	0
5	3	0	14.0	0	0	9.2250	0
6	3	1	30.0	0	0	7.6292	2
7	2	0	26.0	1	1	29.0000	0
8	3	1	18.0	0	0	7.2292	1
9	3	0	21.0	2	0	24.1500	0

21. Prediksi penumpang mana saja yang selamat dengan model KNN yang tadi sudah dibuat

In [57]:
```
x_test = data_uji
y_pred = knn.predict(x_test)
datauji = pd.read_csv('test.csv')
hasil = pd.DataFrame({
        "PassengerId": datauji["PassengerId"],
        "Survived": y_pred
    })
hasil.head(30)
```

Output: (di halaman selanjutnya)

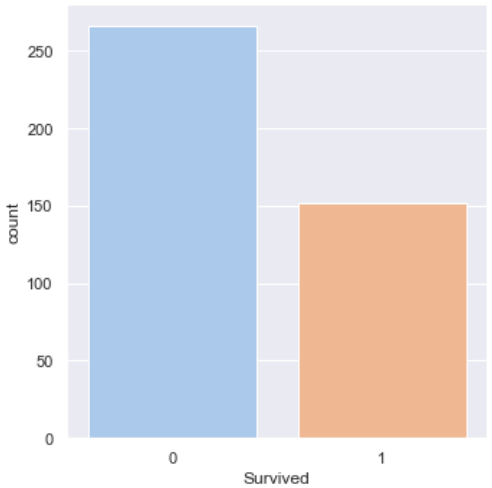
Out[57]:

	PassengerId	Survived
0	892	0
1	893	0
2	894	1
3	895	1
4	896	0
5	897	0
6	898	0
7	899	1
8	900	0
9	901	1
10	902	0
11	903	0
12	904	1
13	905	0
14	906	0
15	907	1
16	908	0
17	909	1
18	910	1
19	911	0
20	912	0
21	913	0
22	914	1
23	915	1
24	916	1
25	917	0
26	918	1
27	919	0
28	920	0
29	921	0

22. Visualisasi penumpang selamat dan tidak selamat di data uji

```
In [62]: sns.catplot(x="Survived", kind="count", palette="pastel", data=hasil)
```

```
Out[62]: <seaborn.axisgrid.FacetGrid at 0x148e53ec610>
```



Kesimpulannya, adalah lebih banyak penumpang yang tidak selamat (260+) daripada penumpang yang selamat (150).