

**LAPORAN TUGAS 2**  
**PENAMBANGAN DATA**



Laporan ini dibuat untuk memenuhi tugas 1 mata kuliah Penambangan Data

Disusun oleh :

Edgar Vigo 1301180149

Muhammad Raihan Muhith 1301184245

M Wanda Wibisono 1301184339

M Tegar Zharfan H. S 1301184354

M Dwiantara Mahardika 1301184467

**Telkom University**

**Bandung**

**2021**

# DAFTAR ISI

1.	Dataset Census Income .....	1
1.1	Handling Outlier .....	1
1.2	Handling Missing Value .....	5
1.3	Features subset selection.....	6
1.4	Transformasi Data.....	7
1.5	Sampling .....	8
1.6	Data Splitting .....	9
1.7	Decision Tree Model.....	9
1.8	Accurasy .....	9
1.9	Naive Bayes Model.....	10
2.	Dataset Dictionaries .....	12
A.	Formulasi Masalah.....	12
B.	Data Pre-Processing.....	13
1.	Handling Outlier .....	14
2.	Handling Missing Value .....	15
3.	Features subset selection .....	16
4.	Transformasi Data.....	16
C.	Proses Classification .....	19
D.	Eksperimen dan Analisis.....	19
E.	Kesimpulan .....	22



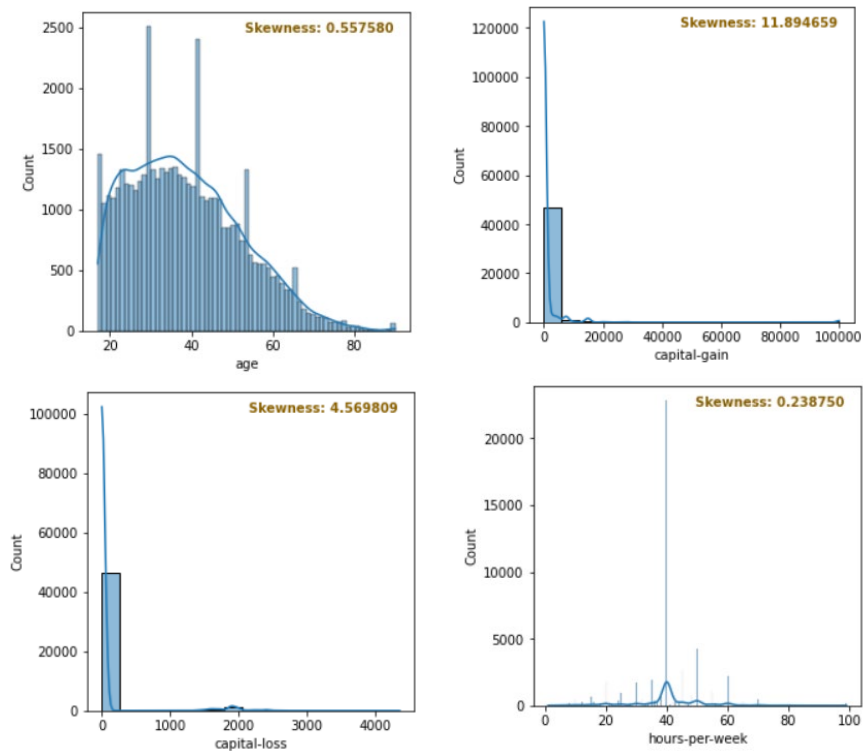
## 1. Dataset Census Income

### Data Cleaning

*Data cleaning* adalah salah satu tahapan dalam *data preparation*. Dilakukan analisis dengan cara menghapus atau memodifikasi data salah, tidak relevan, duplikat, dan tidak terformat. Proses *data cleaning* merupakan proses yang penting dilakukan karena akan mempengaruhi hasil *modelling machine learning*.

Untuk melakukan Data Cleaning pertama yang dilakukan adalah Handling Outlier terlebih dahulu

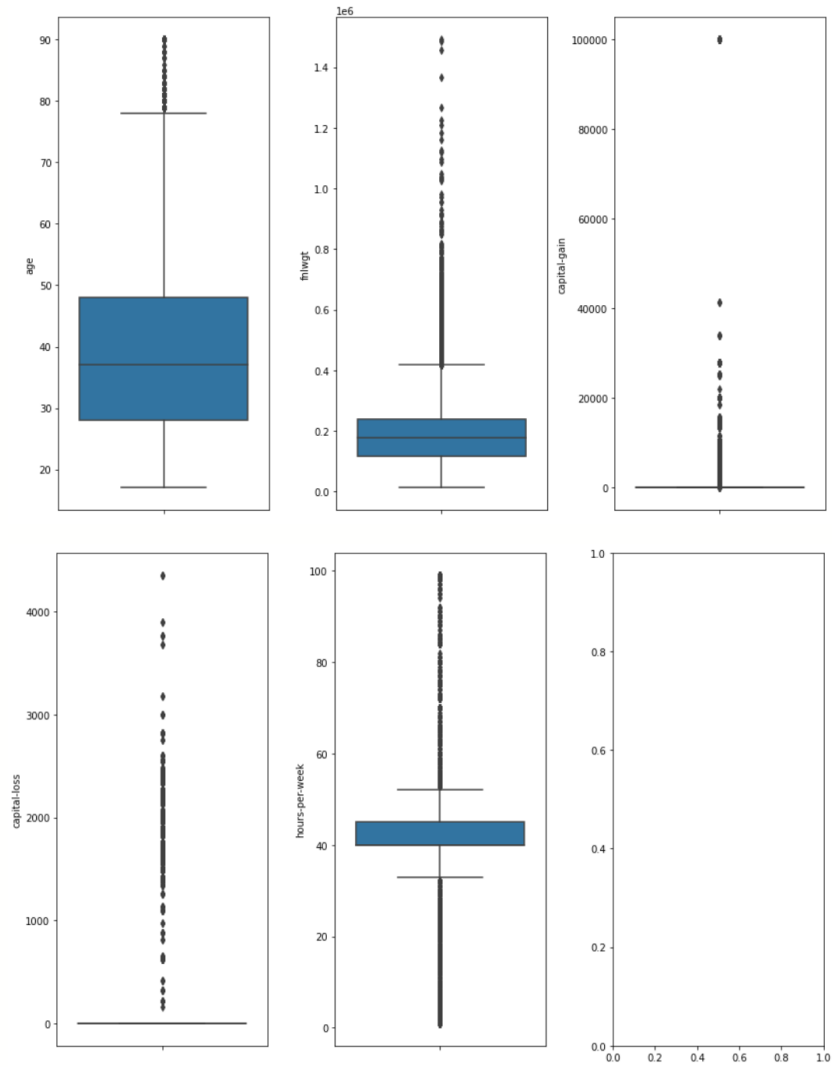
#### 1.1 Handling Outlier



- Skewness normal ( $<1$ ) : missing value diganti dengan mean, outliers dibuang semua
- Skewness abnormal ( $>1$ ) : missing value diganti dengan median, dibuang titik terjauh(gunakan boxplot).

Dapat dilihat bahwa, atribut numerik yang memiliki skewness normal adalah 'hours-per-week' dan 'age'. sedangkan untuk atribut numerik yang memiliki skewness abnormal adalah 'fngwt', 'capital-gain' dan 'capital-loss'.

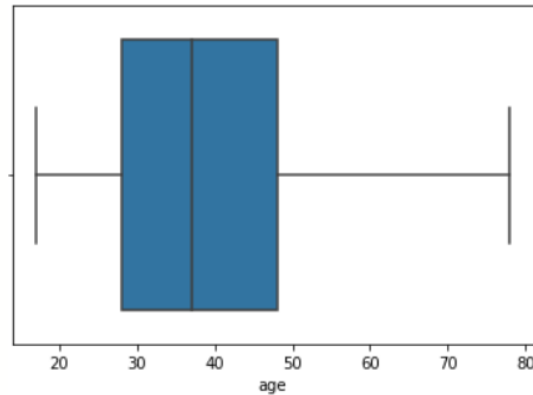
Melakukan boxploting untuk melihat fliers dari atribut numerik:



- banyaknya fliers dari variabel hours-per-week : 13496
- banyaknya fliers dari variabel fnlwgt : 1453
- banyaknya fliers dari variabel capital-gain : 4035
- banyaknya fliers dari variabel capital-loss : 2282

Dapat dilihat dari banyaknya fliers dari atribut 'hours-per-week', walaupun atribut tersebut masuk kedalam kategori skewness normal, membuang semua fliers(outliers) dari attribute tersebut akan mengakibatkan menurunnya jumlah data secara drastis. hal ini mengakibatkan model yang tidak akurat. jadi kami sepakat untuk menghapus semua fliers(outliers) hanya dari atribut 'age' saja.

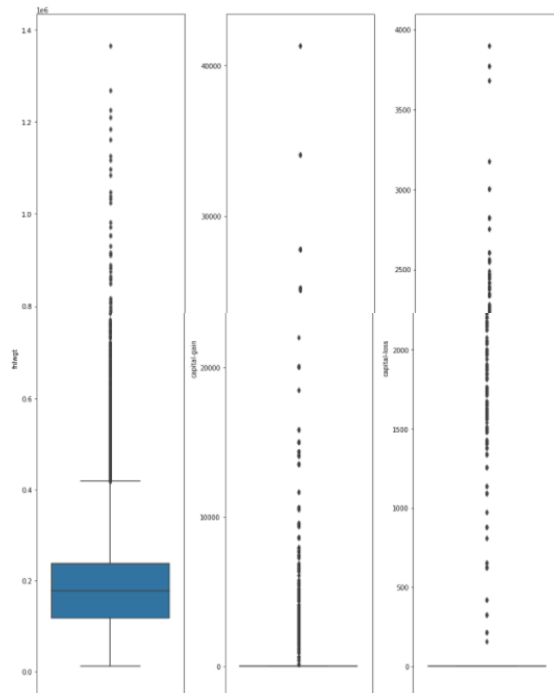
Melakukan removing outliers atribut 'age'.



kami sepakat bahwa :

- titik terjauh dari fnlwgt ada diatas 1400000
- titik terjauh dari capital-gain ada diatas 80000
- titik terjauh dari capital-loss ada diatas 4000

Melakukan boxploting untuk melihat fliers dari atribut numerik yang sudah di remove pada tahapan sebelumnya:



Sudah dapat dilihat beberapa titik terjauh dari ketiga atribut diatas sudah dibuang.

melihat unique value dari atribut 'education-num' :

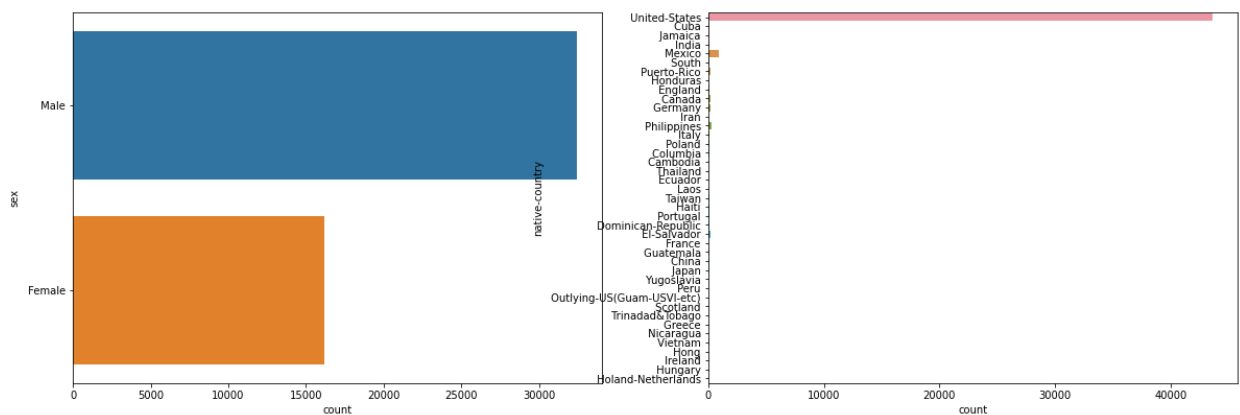
```
[13  9  7 14  5 10 12 11  4 16 15  3  6  2  1  8]
terdapat : 16 unique value dari atribut education-num
```

untuk mengecek outliers atribut 'education-num', dilakukan secara manual, karena atribut ini adalah kategorikal namun tipe datanya numerik. lalu valuenya berdasarkan jumlah unique value dari atribut 'education'.

Dapat dilihat bahwa, dari unique value atribut 'education-num' tidak terdapat value yang diluar range [1-16].

countploting to see outliers from categorical attribute.





kami menggunakan countplot untuk mencari nilai outliers dari atribut kategorikal, dan dapat dilihat semua atribut kategorik tidak memiliki outliers kecuali atribut 'native-country'. unique value atribut tersebut terlalu banyak sehingga sedikit sulit untuk dilihat melalui countplot.

Melihat unique value dari atribut 'native-country' :

```
array(['United-States', 'Cuba', 'Jamaica', 'India', nan, 'Mexico',
      'South', 'Puerto-Rico', 'Honduras', 'England', 'Canada',
      'Germany', 'Iran', 'Philippines', 'Italy', 'Poland',
      'Columbia', 'Cambodia', 'Thailand', 'Ecuador', 'Laos',
      'Taiwan', 'Haiti', 'Portugal', 'Dominican-Republic',
      'El-Salvador', 'France', 'Guatemala', 'China', 'Japan',
      'Yugoslavia', 'Peru', 'Outlying-US(Guam-USVI-etc)', 'Scotland',
      'Trinidad&Tobago', 'Greece', 'Nicaragua', 'Vietnam', 'Hong',
      'Ireland', 'Hungary', 'Holand-Netherlands'], dtype=object)
```

di kamus dataset, tidak terdapat 'South' dalam atribut 'native-country'. Sehingga dapat disimpulkan bahwa value south merupakan outliers dari atribut 'native-country'.

## 1.2 Handling Missing Value

Mengcek missing value dari setiap atribut :

```
age          0
workclass    2779
fnlwgt       0
education    0
education-num 0
marital-status 0
occupation   2789
relationship  0
race         0
sex          0
capital-gain  0
capital-loss  0
hours-per-week 0
native-country 847
income       0
dtype: int64
```



Dapat dilihat bahwa terdapat sebanyak 2779 data yang kosong dari atribut 'workclass', 2789 data yang kosong dari atribut 'occupation', dan 847 data yang kosong dari atribut 'native-country'.

Perlu diperhatikan bahwa ketiga atribut tersebut merupakan atribut kategorikal, sehingga metode handling missing value yang tepat untuk ketiga atribut tersebut adalah dengan mereplaceny dengan nilai modus.

handling missing value (replace dengan modus) :

```
age          0
workclass    0
fnlwgt       0
education    0
education-num 0
marital-status 0
occupation   0
relationship 0
race         0
sex          0
capital-gain 0
capital-loss 0
hours-per-week 0
native-country 0
income       0
dtype: int64
```

### 1.3 Features subset selection

Disini kami mendapati bahwa terdapat atribut yang memiliki makna yang sama. karena pada proses Preprocess ini nantinya data akan di transformasikan kedalam value numerik, maka atribut 'education' yang perlu kami di drop.

Cek data yang mirip:

```
workclass : [' State-gov' ' Self-emp-not-inc' ' Private' ' Federal-gov' ' Local-gov'
' Self-emp-inc' ' Without-pay' ' Never-worked']

education-num : [13  9  7 14  5 10 12 11  4 16 15  3  6  2  1  8]

marital-status : [' Never-married' ' Married-civ-spouse' ' Divorced'
' Married-spouse-absent' ' Separated' ' Married-AF-spouse' ' Widowed']

occupation : [' Adm-clerical' ' Exec-managerial' ' Handlers-cleaners' ' Prof-specialty'
' Other-service' ' Sales' ' Craft-repair' ' Transport-moving'
' Farming-fishing' ' Machine-op-inspct' ' Tech-support'
' Protective-serv' ' Armed-Forces' ' Priv-house-serv']

relationship : [' Not-in-family' ' Husband' ' Wife' ' Own-child' ' Unmarried'
' Other-relative']

race : [' White' ' Black' ' Asian-Pac-Islander' ' Amer-Indian-Eskimo' ' Other']

sex : [' Male' ' Female']

native-country : [' United-States' ' Cuba' ' Jamaica' ' India' ' Mexico' ' Puerto-Rico'
' Honduras' ' England' ' Canada' ' Germany' ' Iran' ' Philippines'
' Italy' ' Poland' ' Columbia' ' Cambodia' ' Thailand' ' Ecuador' ' Laos'
' Taiwan' ' Haiti' ' Portugal' ' Dominican-Republic' ' El-Salvador'
' France' ' Guatemala' ' China' ' Japan' ' Yugoslavia' ' Peru'
' Outlying-US(Guam-USVI-etc)' ' Scotland' ' Trinidad&Tobago' ' Greece'
' Nicaragua' ' Vietnam' ' Hong' ' Ireland' ' Hungary'
' Holand-Netherlands']

income : [' <=50K' ' >50K' ' <=50K.' ' >50K.']
```

setelah diperhatikan, atribut income memiliki value yang mirip. yaitu value dengan tanda titik dan tanpa tanda titik. hal ini perlu kami atasi dengan menyamakan value yang mirip tersebut.

Value Atribut Kategorial :

	age	workclass	fnlwt	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
0	39.0	6	77516	13	4	0	1	4	1	2174	0	40	37	0
1	50.0	5	83311	13	2	3	0	4	1	0	0	13	37	0
2	38.0	3	215646	9	0	5	1	4	1	0	0	40	37	0
3	53.0	3	234721	7	2	5	0	2	1	0	0	40	37	0
4	28.0	3	338409	13	2	9	5	2	0	0	0	40	4	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48837	39.0	3	215419	13	0	9	1	4	0	0	0	36	37	0
48838	64.0	3	321403	9	6	2	2	2	1	0	0	40	37	0
48839	38.0	3	374983	13	2	9	0	4	1	0	0	50	37	0
48840	44.0	3	83891	13	0	0	3	1	1	5455	0	40	37	0
48841	35.0	4	182148	13	2	3	0	4	1	0	0	60	37	1

48478 rows x 14 columns

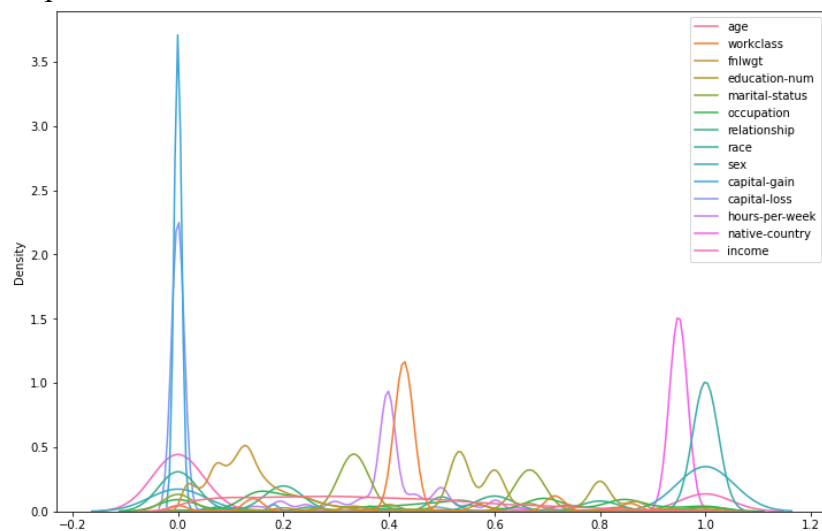
## 1.4 Transformasi Data

Melakukan feature scaling menggunakan method MinMaxScaler:

	age	workclass	fnlwt	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
0	0.360656	0.857143	0.048182	0.800000	0.666667	0.000000	0.2	1.00	1.0	0.052626	0.0	0.397959	0.948718	0.0
1	0.540984	0.714286	0.052463	0.800000	0.333333	0.230769	0.0	1.00	1.0	0.000000	0.0	0.122449	0.948718	0.0
2	0.344262	0.428571	0.150211	0.533333	0.000000	0.384615	0.2	1.00	1.0	0.000000	0.0	0.397959	0.948718	0.0
3	0.590164	0.428571	0.164301	0.400000	0.333333	0.384615	0.0	0.50	1.0	0.000000	0.0	0.397959	0.948718	0.0
4	0.180328	0.428571	0.240889	0.800000	0.333333	0.692308	1.0	0.50	0.0	0.000000	0.0	0.397959	0.102564	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48473	0.360656	0.428571	0.150043	0.800000	0.000000	0.692308	0.2	1.00	0.0	0.000000	0.0	0.357143	0.948718	0.0
48474	0.770492	0.428571	0.228328	0.533333	1.000000	0.153846	0.4	0.50	1.0	0.000000	0.0	0.397959	0.948718	0.0
48475	0.344262	0.428571	0.267904	0.800000	0.333333	0.692308	0.0	1.00	1.0	0.000000	0.0	0.500000	0.948718	0.0
48476	0.442623	0.428571	0.052891	0.800000	0.000000	0.000000	0.6	0.25	1.0	0.132050	0.0	0.397959	0.948718	0.0
48477	0.295082	0.571429	0.125468	0.800000	0.333333	0.230769	0.0	1.00	1.0	0.000000	0.0	0.602041	0.948718	1.0

48478 rows x 14 columns

kdeplot setelah dilakukan MinMaxScaler:

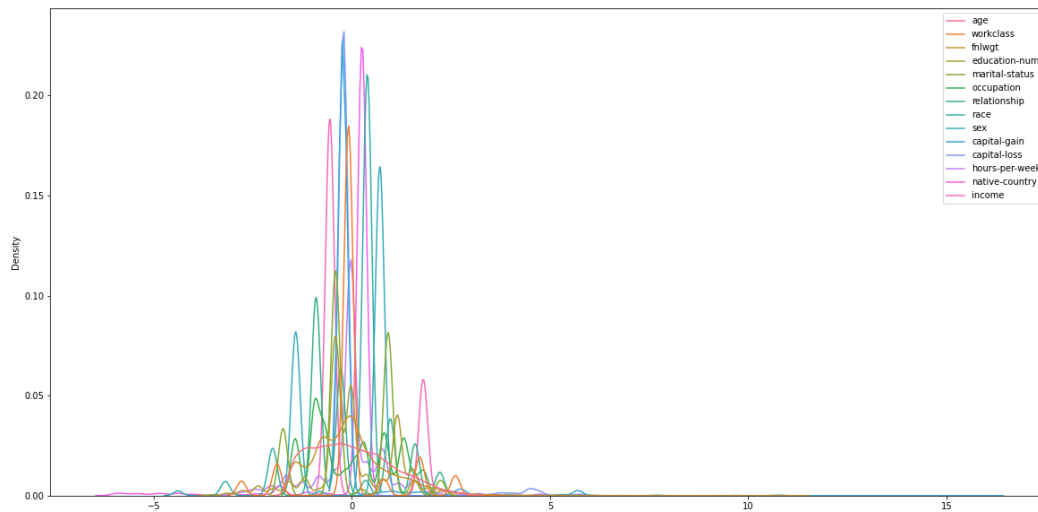


Melakukan feature scaling menggunakan method StandardScaler:

	age	workclass	fnlwgt	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	income
0	0.041250	2.613336	-1.066103	1.146388	0.913682	-1.430308	-0.278726	0.389389	0.705423	0.629215	-0.217654	-0.029704	0.255773	-0.555254
1	0.862612	1.713317	-1.011013	1.146388	-0.411399	-0.683424	-0.903070	0.389389	0.705423	-0.229992	-0.217654	-2.214802	0.255773	-0.555254
2	-0.033419	-0.086719	0.247022	-0.414192	-1.736479	-0.185501	-0.278726	0.389389	0.705423	-0.229992	-0.217654	-0.029704	0.255773	-0.555254
3	1.086620	-0.086719	0.428358	-1.194482	-0.411399	-0.185501	-0.903070	-1.999324	0.705423	-0.229992	-0.217654	-0.029704	0.255773	-0.555254
4	-0.780112	-0.086719	1.414062	1.146388	-0.411399	0.810344	2.218653	-1.999324	-1.417590	-0.229992	-0.217654	-0.029704	-5.427903	-0.555254
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
48473	0.041250	-0.086719	0.244864	1.146388	-1.736479	0.810344	-0.278726	0.389389	-1.417590	-0.229992	-0.217654	-0.353422	0.255773	-0.555254
48474	1.907982	-0.086719	1.252395	-0.414192	2.238762	-0.932385	0.345619	-1.999324	0.705423	-0.229992	-0.217654	-0.029704	0.255773	-0.555254
48475	-0.033419	-0.086719	1.761750	1.146388	-0.411399	0.810344	-0.903070	0.389389	0.705423	-0.229992	-0.217654	0.779592	0.255773	-0.555254
48476	0.414596	-0.086719	-1.005499	1.146388	-1.736479	-1.430308	0.969963	-3.193680	0.705423	1.925930	-0.217654	-0.029704	0.255773	-0.555254
48477	-0.257427	0.813299	-0.071425	1.146388	-0.411399	-0.683424	-0.903070	0.389389	0.705423	-0.229992	-0.217654	1.588888	0.255773	1.800978

48478 rows x 14 columns

kdeplot setelah dilakukan StandardScaler :



### 1.5 Sampling

Sampling ini digunakan agar data yang diuji tidak terlalu banyak, sehingga mempercepat proses machine learning. Kami sepakat untuk memberikan sampling sebanyak 25% dari dataset karena rasio tersebut menurut kami adalah yang paling optimal.

```
#sampling

sampling = df_scaled2.sample(frac=0.25)
sampling.shape

(12120, 14)
```

## 1.6 Data Splitting

Melakukan Data Splitting untuk memilah dataset menjadi training set dan test set.

```
from sklearn.model_selection import train_test_split

x = df_scaled.drop('income',1)
y = df_scaled['income']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
```

## 1.7 Decision Tree Model

Untuk model decision tree kami melakukan tuning parameter max\_depth dengan melakukan try and error. Didapatkan max\_depth = 5 adalah parameter yang optimal untuk kasus ini.

```
from sklearn.tree import DecisionTreeClassifier

model_ID3 = DecisionTreeClassifier(max_depth=5)
model_ID3.fit(x_train, y_train)
y_pred = model_ID3.predict(x_test)
y_pred_train = model_ID3.predict(x_train)
```

## 1.8 Accuracy

Akurasi didefinisikan sebagai tingkat kedekatan antara nilai prediksi dengan nilai aktual. Semakin besar nilai akurasi, maka performansi sistem klasifikasi semakin baik. Persamaan akurasi adalah sebagai berikut.

```
data = {
    'Training Set Accuracy': accuracy_score(y_true = y_train, y_pred = y_pred_train),
    'Test Set Accuracy': accuracy_score(y_true = y_test, y_pred = y_pred)
}

df_models_ID3 = pd.DataFrame(data, index=['Decision Tree'])
df_models_ID3
```

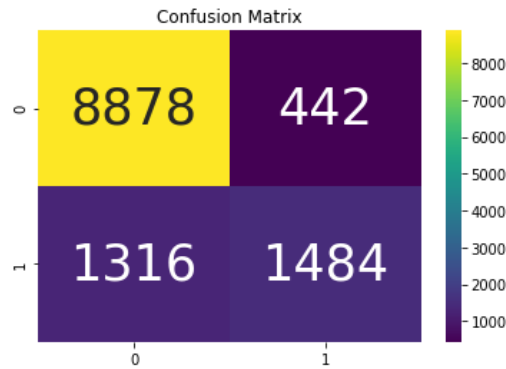
	Training Set Accuracy	Test Set Accuracy
Decision Tree	0.852577	0.85495

Confusion Matrix Hasil Klasifikasi Data Test Menggunakan Decision Tree:

- True Positive (TP) : ketika prediksi 1 dan faktanya 1
- True Negative (TN) : ketika prediksi 0 dan faktanya 0
- False Positive (FP) : ketika prediksi 1 dan faktanya 0
- False Negative (FN) : ketika prediksi 0 dan faktanya 1

Pada kasus ini

- 0 menyatakan income  $\leq 50k$
- 1 menyatakan income  $> 50k$



- Precision  
Presisi adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem.
- Recall  
Recall adalah tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi.
- F1-Score  
F1 Score adalah perbandingan rata-rata antara precision dan recall. Score ini akan memperhitungkan false positive dan false negative.

```
print('Classification Report : ')\nprint(classification_report(y_test, y_pred))
```

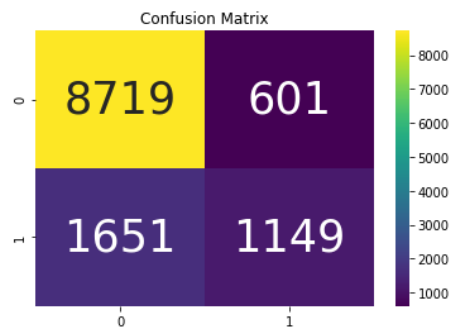
```
Classification Report :\n              precision    recall  f1-score   support\n\n     0.0           0.87       0.95       0.91        9320\n     1.0           0.77       0.53       0.63        2800\n\n   accuracy              0.82       0.74       0.77        12120\n  macro avg              0.82       0.74       0.77        12120\n weighted avg              0.85       0.85       0.84        12120
```

Model naive bayes dibangun dengan parameter default

```
data = {\n    'Training Set Accuracy': accuracy_score(y_true = y_train, y_pred = y_pred_train_NB),\n    'Test Set Accuracy': accuracy_score(y_true = y_test, y_pred = y_pred_NB)\n}\n\ndf_models_NB = pd.DataFrame(data,index=['Naive Bayes'])\ndf_models_NB
```

	Training Set Accuracy	Test Set Accuracy
Naive Bayes	0.811293	0.814191

## Confusion Matrix Hasil Klasifikasi Data Test Menggunakan Naive Bayes:



```
print('Classification Report : ')
print(classification_report(y_test, y_pred_NB))
```

```
Classification Report :
              precision    recall  f1-score   support

    0.0         0.84        0.94        0.89        9320
    1.0         0.66        0.41        0.51        2800

 accuracy          0.81        12120
 macro avg         0.75        0.67        0.70        12120
 weighted avg      0.80        0.81        0.80        12120
```

	Train Set Accuracy	Test Set Accuracy
<b>Decision Tree</b>	0.852577	0.854950
<b>Naive Bayes</b>	0.811293	0.814191

## 2. Dataset Breast Cancer

### A. Formulasi Masalah

Pada dataset yang kedua ini yang berisi tentang berbagai macam atribut yang memengaruhi apakah seseorang penderita kanker payudara melakukan terapi radiasi.

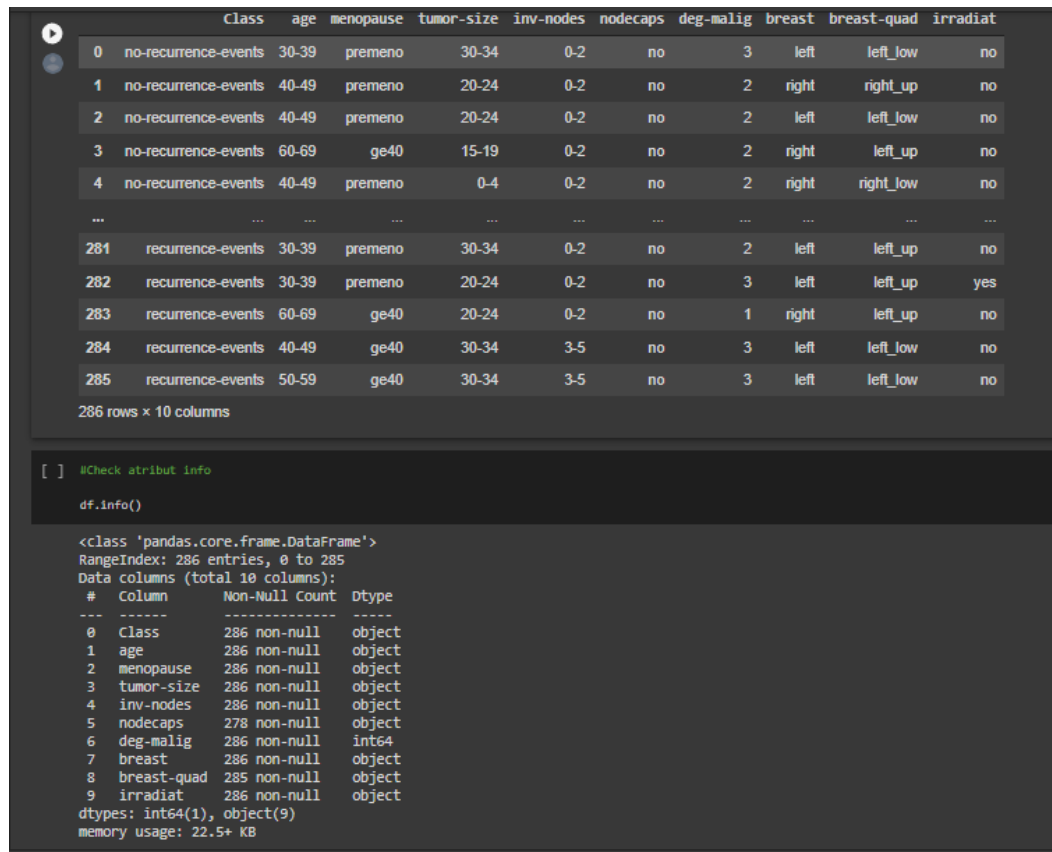
Tugas kami adalah untuk membuat model klasifikasi dan melakukan eksperimen terhadap atribut Irradiat, yaitu attribut untuk memprediksi apakah seseorang penderita kanker payudara membutuhkan terapi radiasi atau tidak. Yang akan kami pertimbangkan dalam eksperimen adalah metode data pre-processing apa dan model klasifikasi apa yang menghasilkan nilai akurasi yang paling baik. Detail dari eksperimen akan kami jelaskan pada bagian berikutnya.

Masalah yang dapat diselesaikan menggunakan hasil dari dataset yang kedua ini adalah masalah tentang seorang penderita kanker payudara, apakah mereka melakukan terapi radiasi atau tidak. Hasil prediksi dapat dihasilkan dari model klasifikasi yang sudah dibuat.

## B. Data Pre-Processing

Data cleaning adalah salah satu tahapan dalam data preparation. Dilakukan analisis dengan cara menghapus atau memodifikasi data salah, tidak relevan, duplikat, dan tidak terformat. Proses data cleaning merupakan proses yang penting dilakukan karena akan mempengaruhi hasil modelling machine learning.

Langkah pertama yang kami lakukan adalah melakukan review data yang akan di proses.



	Class	age	menopause	tumor-size	inv-nodes	nodecaps	deg-malig	breast	breast-quad	irradiat
0	no-recurrence-events	30-39	premeno	30-34	0-2	no	3	left	left_low	no
1	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	right	right_up	no
2	no-recurrence-events	40-49	premeno	20-24	0-2	no	2	left	left_low	no
3	no-recurrence-events	60-69	ge40	15-19	0-2	no	2	right	left_up	no
4	no-recurrence-events	40-49	premeno	0-4	0-2	no	2	right	right_low	no
...	...	...	...	...	...	...	...	...	...	...
281	recurrence-events	30-39	premeno	30-34	0-2	no	2	left	left_up	no
282	recurrence-events	30-39	premeno	20-24	0-2	no	3	left	left_up	yes
283	recurrence-events	60-69	ge40	20-24	0-2	no	1	right	left_up	no
284	recurrence-events	40-49	ge40	30-34	3-5	no	3	left	left_low	no
285	recurrence-events	50-59	ge40	30-34	3-5	no	3	left	left_low	no

286 rows x 10 columns

```
[ ] #Check atribut info
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 286 entries, 0 to 285
Data columns (total 10 columns):
#   column      Non-Null Count  Dtype
---  -
0   Class        286 non-null     object
1   age          286 non-null     object
2   menopause    286 non-null     object
3   tumor-size   286 non-null     object
4   inv-nodes    286 non-null     object
5   nodecaps     278 non-null     object
6   deg-malig    286 non-null     int64
7   breast       286 non-null     object
8   breast-quad  285 non-null     object
9   irradiat     286 non-null     object
dtypes: int64(1), object(9)
memory usage: 22.5+ KB
```



## 1. Handling Outlier

Setelah mengetahui isi dataset maka kami melakukan handling outlier pada dataset yang kedua ini. Kami melakukan handling outlier menggunakan countplot.



setelah dilakukan countploting untuk seluruh atribut, maka kami dapat melihat values apa saja yang terdapat pada atribut tersebut. setelah dicek seluruh atribut bebas dari outliers.

## 2. Handling Missing Value

Selanjutnya yang kami lakukan adalah mengecek apakah terdapat null value pada dataset. Untuk menjaga distribusi dataset, kami tidak melakukan drop langsung pada null value, melainkan atribut yang terdapat null valuenya akan kami ganti dengan nilai modus dari atribut tersebut.

```
[ ] #Cek missing value dari setiap atribut
df.isna().sum()

Class      0
age         0
menopause  0
tumor-size 0
inv-nodes  0
nodelcaps  8
deg-malig  0
breast      0
breast-quad 1
irradiat    0
dtype: int64

[ ] #handling missing value (replace dengan modus)

df['nodelcaps'].fillna(df['nodelcaps'].mode()[0],inplace=True)
df['breast-quad'].fillna(df['breast-quad'].mode()[0],inplace=True)

[ ] df.isna().sum()

Class      0
age         0
menopause  0
tumor-size 0
inv-nodes  0
nodelcaps  0
deg-malig  0
breast      0
breast-quad 0
irradiat    0
dtype: int64
```

### 3. Features subset selection

Setelah dataset bersih dari record yang berisi null value, kami akan melakukan konversi atribut yang bernilai kategorikal, menjadi atribut yang bernilai numerikal yang bertujuan untuk mempermudah proses perbandingan

```
#encoding value atribut kategorikal

categorical = df.dtypes==object
categorical_cols = df.columns[categorical].tolist()
df[categorical_cols] = df[categorical_cols].apply(lambda col: preprocessing.LabelEncoder().fit_transform(col))
df
```

	Class	age	menopause	tumor-size	inv-nodes	nodecaps	deg-malig	breast	breast-quad	irradiat
0	0	1	2	5	0	0	3	0	1	0
1	0	2	2	3	0	0	2	1	4	0
2	0	2	2	3	0	0	2	0	1	0
3	0	4	0	2	0	0	2	1	2	0
4	0	2	2	0	0	0	2	1	3	0
...	...	...	...	...	...	...	...	...	...	...
281	1	1	2	5	0	0	2	0	2	0
282	1	1	2	3	0	0	3	0	2	1
283	1	4	0	3	0	0	1	1	2	0
284	1	2	0	5	4	0	3	0	1	0
285	1	3	0	5	4	0	3	0	1	0

286 rows x 10 columns

```
[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 286 entries, 0 to 285
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Class        286 non-null    int32
1   age          286 non-null    int32
2   menopause    286 non-null    int32
3   tumor-size   286 non-null    int32
4   inv-nodes    286 non-null    int32
5   nodecaps     286 non-null    int32
6   deg-malig    286 non-null    int64
7   breast       286 non-null    int32
8   breast-quad  286 non-null    int32
9   irradiat     286 non-null    int32
dtypes: int32(9), int64(1)
memory usage: 12.4 KB
```

Tahapan terakhir yang kami lakukan pada data pre-processing adalah membuat dataset menjadi dua jenis, yaitu dataset yang dilakukan minmaxscaler dan yang dilakukan standarscaler. Dua jenis dataset ini yang nantinya akan menjadi bahan eksperimen kami.

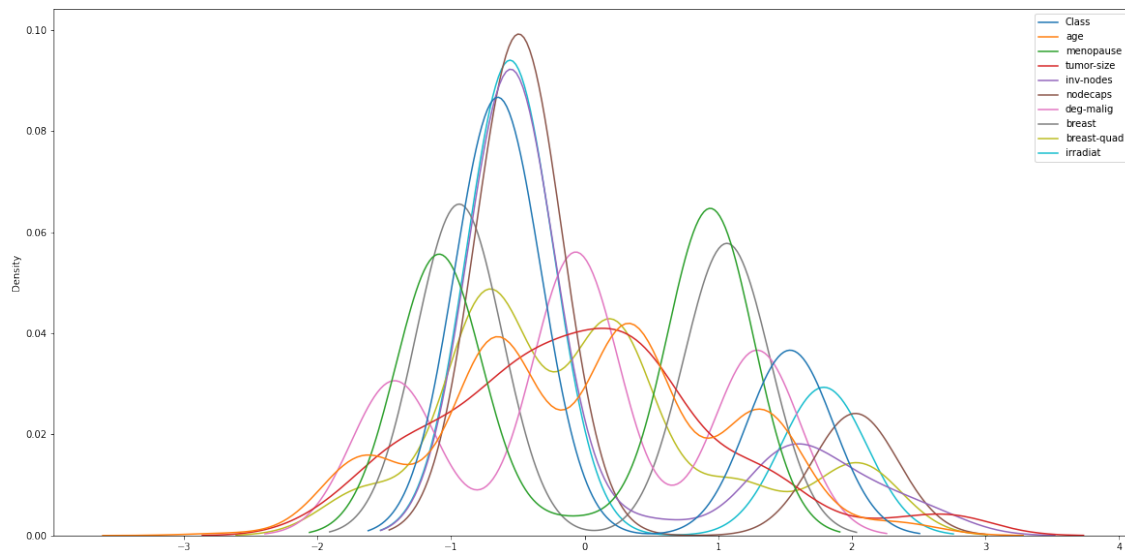
## Dataset yang dilakukan standarscaler

```
#feature scalling menggunakan method StandardScaler
from sklearn.preprocessing import StandardScaler

scaler2 = StandardScaler()
df_scaled2 = pd.DataFrame(scaler2.fit_transform(df.astype(float)), columns=df.columns)
df_scaled2
```

	Class	age	menopause	tumor-size	inv-nodes	nodecaps	deg-malig	breast	breast-quad	irradiat
0	-0.850297	-1.647779	0.940728	0.436366	-0.555623	-0.493435	1.290564	-0.938924	-0.718171	-0.558504
1	-0.850297	-0.657727	0.940728	-0.494983	-0.555623	-0.493435	-0.066426	1.065049	2.045017	-0.558504
2	-0.850297	-0.657727	0.940728	-0.494983	-0.555623	-0.493435	-0.066426	-0.938924	-0.718171	-0.558504
3	-0.850297	1.322377	-1.089825	-0.960657	-0.555623	-0.493435	-0.066426	1.065049	0.202891	-0.558504
4	-0.850297	-0.657727	0.940728	-1.892006	-0.555623	-0.493435	-0.066426	1.065049	1.123954	-0.558504
...	...	...	...	...	...	...	...	...	...	...
281	1.537760	-1.647779	0.940728	0.436366	-0.555623	-0.493435	-0.066426	-0.938924	0.202891	-0.558504
282	1.537760	-1.647779	0.940728	-0.494983	-0.555623	-0.493435	1.290564	-0.938924	0.202891	1.790498
283	1.537760	1.322377	-1.089825	-0.494983	-0.555623	-0.493435	-1.423416	1.065049	0.202891	-0.558504
284	1.537760	-0.657727	-1.089825	0.436366	1.514841	-0.493435	1.290564	-0.938924	-0.718171	-0.558504
285	1.537760	0.332325	-1.089825	0.436366	1.514841	-0.493435	1.290564	-0.938924	-0.718171	-0.558504
286 rows x 10 columns										

kdeplot setelah dilakukan StandardScaler



## Dataset yang dilakukan minmaxscaler

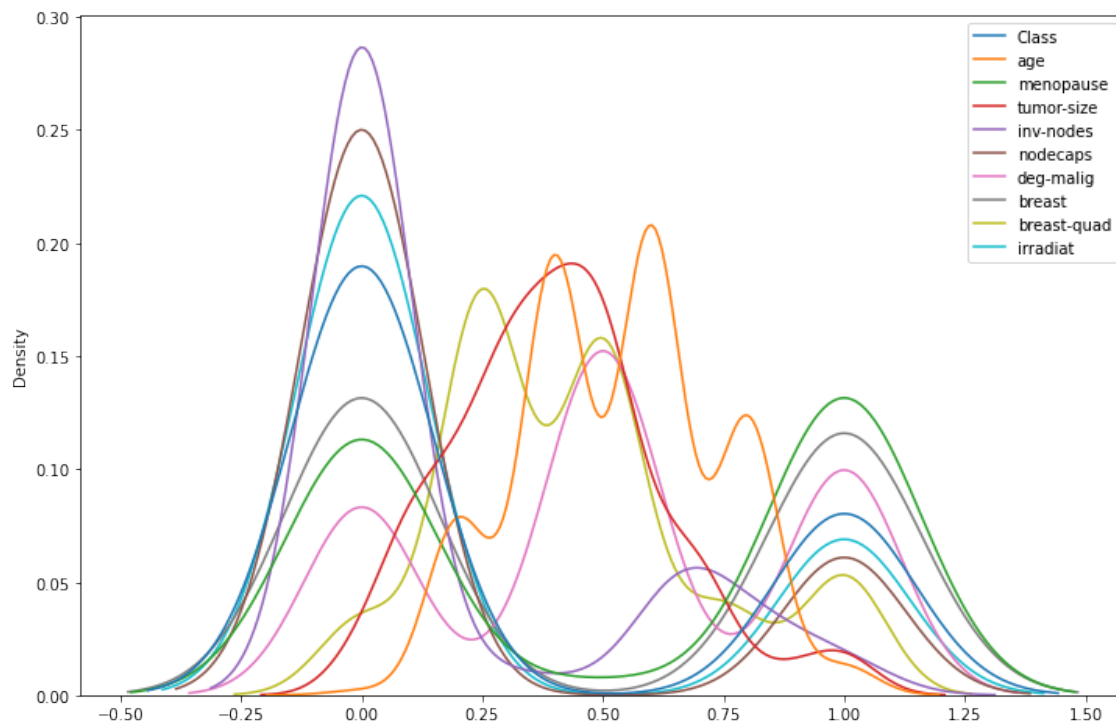
```
[ ] #feature scalling menggunakan method MinMaxScaler
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df.astype(float)), columns=df.columns)
df_scaled
```

	Class	age	menopause	tumor-size	inv-nodes	nodecaps	deg-malig	breast	breast-quad	irradiat
0	0.0	0.2	1.0	0.5	0.000000	0.0	1.0	0.0	0.25	0.0
1	0.0	0.4	1.0	0.3	0.000000	0.0	0.5	1.0	1.00	0.0
2	0.0	0.4	1.0	0.3	0.000000	0.0	0.5	0.0	0.25	0.0
3	0.0	0.8	0.0	0.2	0.000000	0.0	0.5	1.0	0.50	0.0
4	0.0	0.4	1.0	0.0	0.000000	0.0	0.5	1.0	0.75	0.0
...	...	...	...	...	...	...	...	...	...	...
281	1.0	0.2	1.0	0.5	0.000000	0.0	0.5	0.0	0.50	0.0
282	1.0	0.2	1.0	0.3	0.000000	0.0	1.0	0.0	0.50	1.0
283	1.0	0.8	0.0	0.3	0.000000	0.0	0.0	1.0	0.50	0.0
284	1.0	0.4	0.0	0.5	0.666667	0.0	1.0	0.0	0.25	0.0
285	1.0	0.6	0.0	0.5	0.666667	0.0	1.0	0.0	0.25	0.0

286 rows x 10 columns

## kdeplot setelah dilakukan MinMaxScaler



### C. Proses Classification

Pada tahap ini, kami menggunakan dua jenis model klasifikasi, yaitu algoritma Naive Bayes dan algoritma ID3.

Karena diperbolehkan untuk menggunakan library, maka kami membuat model klasifikasinya dengan memanfaatkan library yang sudah ada.

#### Algoritma Naive Bayes

```
from sklearn.naive_bayes import GaussianNB

model_NB = GaussianNB()
model_NB.fit(x_train, y_train)
y_pred_NB = model_NB.predict(x_test)
y_pred_train_NB = model_NB.predict(x_train)
```

```
[ ] from sklearn.tree import DecisionTreeClassifier

model_ID3 = DecisionTreeClassifier(max_depth=4)
model_ID3.fit(x_train, y_train)
y_pred = model_ID3.predict(x_test)
y_pred_train = model_ID3.predict(x_train)
```

Sebelum masuk ke rincian eksperimen, perlu kami beritahu bahwa pada eksperimen kali ini nilai 0.0 menunjukkan nilai numerikal dari nilai kategorikal “Tidak Terapi Radiasi”. Sedangkan nilai 1.0 menunjukkan nilai numerikal dari nilai kategorikal “Melakukan Terapi Radiasi”.

1. Prediksi atribut Irradiat
  - Dataset yang sudah dilakukan minmaxscaler
    - Algoritma Naive Bayes

#### ▼ Confusion Matrix Hasil Klasifikasi Data Test Menggunakan Naive Bayes

```
[ ] matrix = confusion_matrix(y_test, y_pred_NB)
sns.heatmap(data=matrix, cmap="viridis", annot=True, fmt=".0f", annot_kws={"size":35},).set_title('Confusion Matrix')
plt.show()
```



#### ▼ Classification Report Data Test Menggunakan Naive Bayes

```
[ ] print('Classification Report : ')
print(classification_report(y_test, y_pred_NB))
```

```
Classification Report :
              precision    recall  f1-score   support

     0.0         0.86      0.93      0.89         67
     1.0         0.64      0.47      0.55         19

 accuracy          0.83
 macro avg         0.75      0.70      0.72
 weighted avg      0.81      0.83      0.82
```

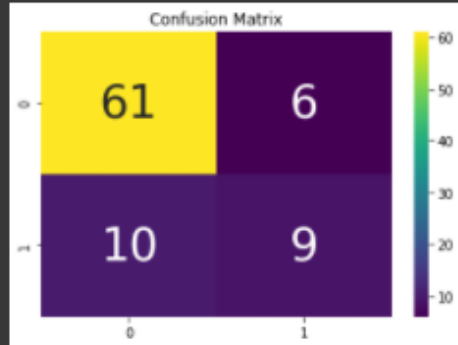
83%. Jika mengacu pada nilai precision, prediksi tidak terapi radiasi memiliki tingkat akurasi sebesar 86%, sedangkan untuk prediksi melakukan terapi radiasi memiliki tingkat akurasi sebesar 64%.

#### ➤ Algoritma ID3

Untuk model decision tree kami melakukan tuning parameter max\_depth dengan melakukan try and error. Didapatkan max\_depth = 4 adalah parameter yang optimal untuk kasus ini.

#### ▼ Confusion Matrix Hasil Klasifikasi Data Test Menggunakan Decision Tree

```
[ ] matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(data=matrix, cmap="viridis", annot=True, fmt=".0f", annot_kws={"size":35}).set_title('Confusion Matrix')
plt.show()
```



#### ▼ Classification Report Data Test Menggunakan Decision Tree

```
[ ] print('Classification Report : ')
print(classification_report(y_test, y_pred))
```

```
Classification Report :
              precision    recall  f1-score   support

     0.0         0.86      0.91      0.88         67
     1.0         0.60      0.47      0.53         19

 accuracy          0.81
 macro avg         0.73
 weighted avg      0.80
```

81%. Jika mengacu pada nilai precision, prediksi tidak terapi radiasi memiliki tingkat akurasi sebesar 86%, sedangkan untuk prediksi melakukan terapi radiasi memiliki tingkat akurasi sebesar 60%.

#### Perbandingan Akurasi Model Decision Tree dengan Naive Bayes

	Train Set Accuracy	Test Set Accuracy
Decision Tree	0.84	0.813953
Naive Bayes	0.75	0.825581



## **E. Kesimpulan**

Ada beberapa kesimpulan yang kami dapatkan setelah melakukan eksperimen dan analisis pada bagian sebelumnya, yaitu:

1. Perbedaan metode pada data pre-processing dan perbedaan metode klasifikasi dapat menghasilkan output tingkat akurasi yang berbeda juga.