In [2]:
```python
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import confusion_matrix, classification_report
```

In [3]:
```python
cancer = load_breast_cancer()
print ("Description Data\n",cancer.keys())
X = cancer['data']
Y = cancer['target']
```

```
Description Data
 dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename'])
```

In [4]:
```python
print ("Ukuran Data\n",cancer['data'].shape,"\n")
print ("Dataset X\n",X[0:5],"\n")
print ("Data Label\n",Y,"\n")
```

```
Ukuran Data
 (569, 30)

Dataset X
 [[1.799e+01 1.038e+01 1.228e+02 1.001e+03 1.184e-01 2.776e-01 3.001e-01
  1.471e-01 2.419e-01 7.871e-02 1.095e+00 9.053e-01 8.589e+00 1.534e+02
  6.399e-03 4.904e-02 5.373e-02 1.587e-02 3.003e-02 6.193e-03 2.538e+01
  1.733e+01 1.846e+02 2.019e+03 1.622e-01 6.656e-01 7.119e-01 2.654e-01
  4.601e-01 1.189e-01]
 [2.057e+01 1.777e+01 1.329e+02 1.326e+03 8.474e-02 7.864e-02 8.690e-02
  7.017e-02 1.812e-01 5.667e-02 5.435e-01 7.339e-01 3.398e+00 7.408e+01
  5.225e-03 1.308e-02 1.860e-02 1.340e-02 1.389e-02 3.532e-03 2.499e+01
  2.341e+01 1.588e+02 1.956e+03 1.238e-01 1.866e-01 2.416e-01 1.860e-01
  2.750e-01 8.902e-02]
 [1.969e+01 2.125e+01 1.300e+02 1.203e+03 1.096e-01 1.599e-01 1.974e-01
  1.279e-01 2.069e-01 5.999e-02 7.456e-01 7.869e-01 4.585e+00 9.403e+01
  6.150e-03 4.006e-02 3.832e-02 2.058e-02 2.250e-02 4.571e-03 2.357e+01
  2.553e+01 1.525e+02 1.709e+03 1.444e-01 4.245e-01 4.504e-01 2.430e-01
  3.613e-01 8.758e-02]
 [1.142e+01 2.038e+01 7.758e+01 3.861e+02 1.425e-01 2.839e-01 2.414e-01
  1.052e-01 2.597e-01 9.744e-02 4.956e-01 1.156e+00 3.445e+00 2.723e+01
  9.110e-03 7.458e-02 5.661e-02 1.867e-02 5.963e-02 9.208e-03 1.491e+01
  2.650e+01 9.887e+01 5.677e+02 2.098e-01 8.663e-01 6.869e-01 2.575e-01
  6.638e-01 1.730e-01]
 [2.029e+01 1.434e+01 1.351e+02 1.297e+03 1.003e-01 1.328e-01 1.980e-01
  1.043e-01 1.809e-01 5.883e-02 7.572e-01 7.813e-01 5.438e+00 9.444e+01
  1.149e-02 2.461e-02 5.688e-02 1.885e-02 1.756e-02 5.115e-03 2.254e+01
  1.667e+01 1.522e+02 1.575e+03 1.374e-01 2.050e-01 4.000e-01 1.625e-01
  2.364e-01 7.678e-02]]

Data Label
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 0
 1 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 1 1 0 1 1 0 1 1
 1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 0 1 1 0 1 1 1 1 0 1
 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 1 0 1 1 0 0 0 1 0
 1 0 1 1 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 1 0 0 1 1
 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 0 0 1 1
 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0
 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1
 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 0 1 1
 0 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1
 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 0 0 1 0 1 0 1 1 1 1 1 0 1 1 0 1 0 1 0 0
 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 1]
```

In [5]:
```python
X_train, X_test, y_train, y_test = train_test_split(X, Y)
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
print("Hasil Preprocessing X-Train\n",X_train,"\n")
print("Hasil Preprocessing X-Test\n",X_test)
```

```
Hasil Preprocessing X-Train
 [[-1.09296042 -1.04653329 -1.05628728 ... -1.10924086 -0.73142765
   0.04875761]
 [ 1.51143892  0.03794906  1.42649404 ...  1.29494582  0.74805376
   0.29865561]
 [ 0.43737217  0.94637907  0.75536078 ...  1.9943291   0.58836371
   0.96937751]
 ...
 [-0.06990475 -0.69442863 -0.13728763 ... -0.88537779 -0.84415004
  -0.69498643]
 [-2.02221237 -1.34699593 -1.97899257 ... -1.72463773  0.04823557
   0.53232646]
 [ 0.1398075  -0.81649158  0.03399546 ... -0.84655673 -0.58113113
  -1.46198937]]

Hasil Preprocessing X-Test
 [[-0.48932923 -0.39631335 -0.46173732 ... -0.24369741 -0.12867596
   1.50487322]
 [-1.67136946  0.35953798 -1.58866476 ... -0.53447771  0.6588152
   3.49540272]
 [ 1.84867888 -0.42682909  1.76823674 ...  1.46563763 -0.33220251
  -0.7263589 ]
 ...
 [ 0.60740912  0.63652698  0.64419147 ... -0.14309497 -0.25861983
  -0.24062643]
 [-0.12374978 -0.66626026 -0.16857973 ... -0.40985758 -0.14902862
  -0.33312115]
 [ 0.21065623  0.94872644  0.35185735 ...  0.88407702  0.15469561
   0.16397033]]
```

In [8]:
```python
model = MLPClassifier(hidden_layer_sizes=(30,30,30))
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
print("Confusion Matriks:\n",confusion_matrix(y_test, y_pred),"\n")
print("Prediction Measure:\n",classification_report(y_test , y_pred))
```

```
Confusion Matriks:
 [[50  3]
 [ 3 87]]

Prediction Measure:
               precision    recall  f1-score   support

           0       0.94      0.94      0.94        53
           1       0.97      0.97      0.97        90

    accuracy                           0.96       143
   macro avg       0.96      0.96      0.96       143
weighted avg       0.96      0.96      0.96       143
```

In [ ]:
```python

```