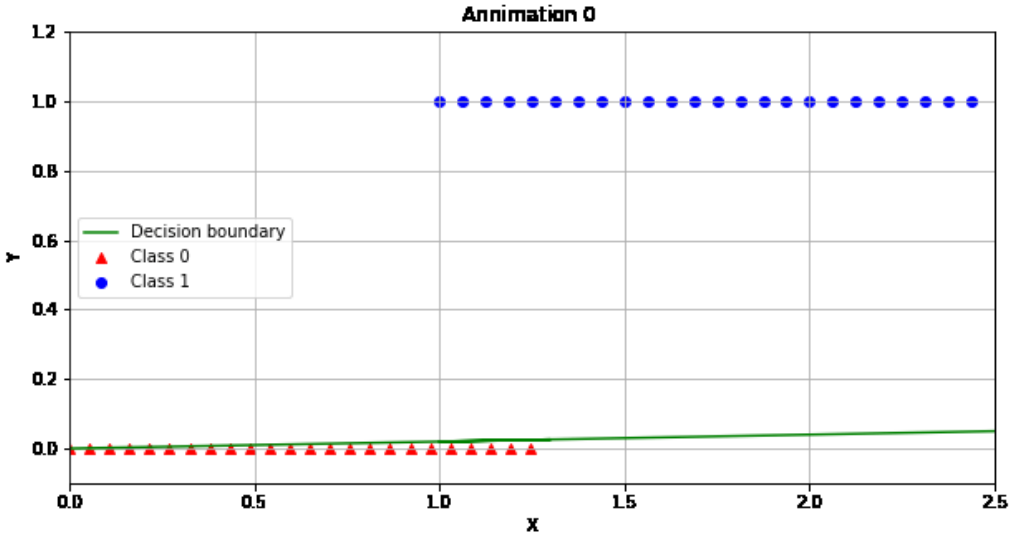
# Logistic Regression

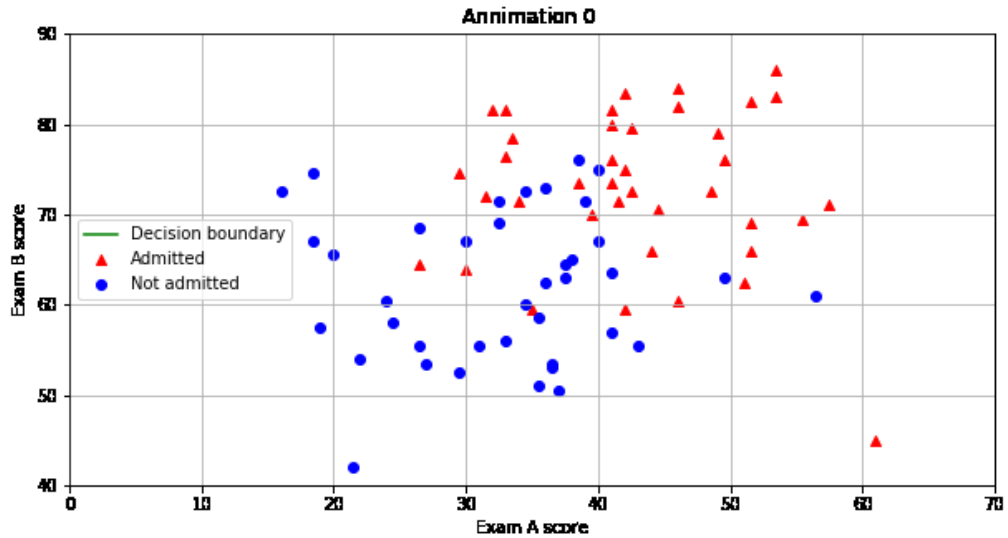
**Logistic linear regression** atau Regresi logistik merupakan analisis regresi yang digunakan ketika variabel dependen bersifat dikotomis (biner). Seperti semua analisis regresi, regresi logistik adalah analisis prediktif. Regresi logistik digunakan untuk menggambarkan data dan untuk menjelaskan hubungan antara satu variabel biner dependen dan satu atau lebih variabel independen nominal, ordinal, interval, atau tingkat rasio.




# Two-dimensional case

For this exercise, suppose that a high school has a dataset representing 40 students who were admitted to college and 40 students who were not admitted. Each $(x^{(i)}, y^{(i)})$ training example contains a student's score on two standardized exams and a label of whether the student was admitted.

Your task is to build a binary classification model that estimates college admission chances based on a student's scores on two exams.

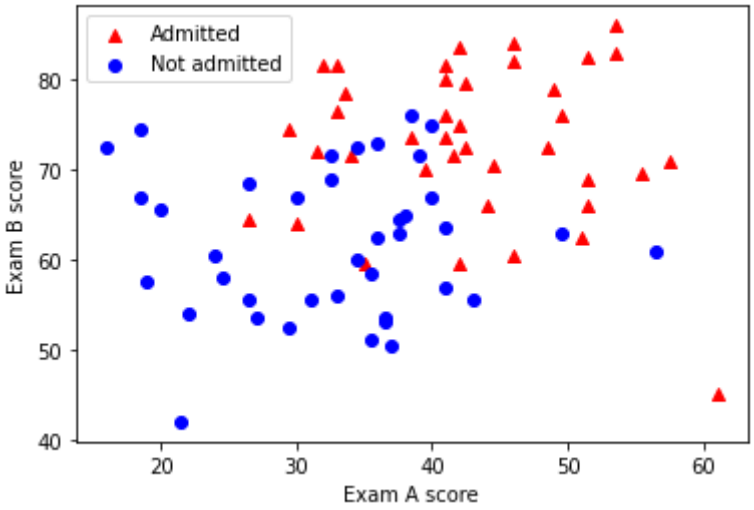


First, download data and save into variable

In [1]:
```
import pandas as pd
import numpy as np

urlx = 'http://bit.ly/32tX8zR'
urly = 'http://bit.ly/2CtJT7m'
dfx = pd.read_table(urlx, sep="\s+",usecols=['ExamA','ExamB'])
dfy = pd.read_table(urly, sep="\s+", usecols=['y'])

dfy = np.ravel(dfy)
```

Plot Data

In [2]:
```
import matplotlib.pyplot as plt


data1=dfx.iloc[0:39]
data2=dfx.iloc[40:79]

plt.xlabel("Exam A score")
plt.ylabel("Exam B score")
plt.scatter(data1.ExamA,data1.ExamB, marker='^', color='red')
plt.scatter(data2.ExamA,data2.ExamB, marker='o', color='blue')
plt.legend(('Admitted','Not admitted'), loc='upper left')
plt.show()
```




# Logistic Regression

In [3]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(dfx, dfy, test_size=0.25, random
```

In [4]:
```
x_test
```

Out[4]:

| | ExamA | ExamB |
|---|---|---|
| **50** | 26.5 | 68.5 |
| **27** | 49.0 | 79.0 |
| **30** | 42.0 | 83.5 |

| | ExamA | ExamB |
|---|---|---|
| **77** | 18.5 | 74.5 |
| **45** | 37.5 | 64.5 |
| **71** | 36.0 | 73.0 |
| **66** | 56.5 | 61.0 |
| **68** | 49.5 | 63.0 |
| **48** | 21.5 | 42.0 |
| **69** | 34.5 | 72.5 |
| **40** | 36.5 | 53.0 |
| **63** | 24.5 | 58.0 |
| **2** | 53.5 | 86.0 |
| **72** | 27.0 | 53.5 |
| **52** | 18.5 | 67.0 |
| **34** | 41.5 | 71.5 |
| **3** | 46.0 | 84.0 |
| **22** | 31.5 | 72.0 |
| **7** | 42.0 | 75.0 |
| **26** | 61.0 | 45.0 |

In [5]:
```
y_test
```

Out[5]:
```
array([0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 1.,
       1., 1., 1.])
```

In [6]:
```
from sklearn.linear_model import LogisticRegression
logisticRegr = LogisticRegression()
logisticRegr.fit(x_train, y_train)
```

Out[6]:
```
LogisticRegression()
```

In [7]:
```
Y_predic = logisticRegr.predict(x_test)
Y_predic
```

Out[7]:
```
array([0., 1., 1., 0., 0., 1., 1., 1., 0., 1., 0., 0., 1., 0., 0., 1., 1.,
       0., 1., 1.])
```

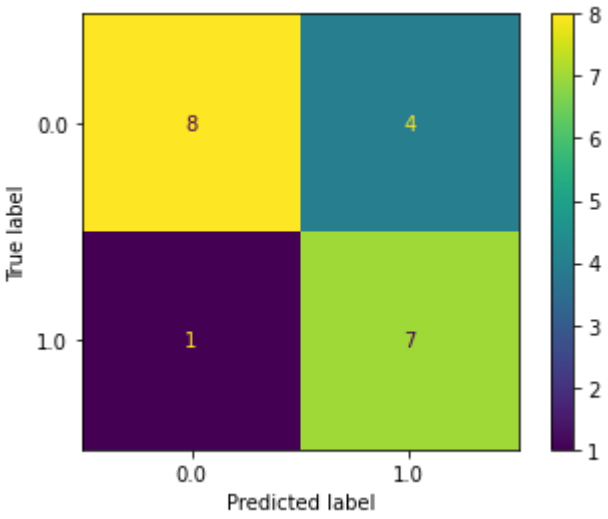
Confusion Matrix

In [8]:
```
from sklearn.metrics import classification_report, confusion_matrix, plot_confusion_
print(confusion_matrix(y_test, Y_predic))
plot_confusion_matrix(logisticRegr, x_test, y_test)
```

```
[[8 4]
 [1 7]]
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.
py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
onfusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the cla
```

```
ss methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
```

Out[8]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x16c35d90670>
```




In [9]:
```
print(classification_report(y_test, Y_predic))
```

```
              precision    recall  f1-score   support

         0.0       0.89      0.67      0.76        12
         1.0       0.64      0.88      0.74         8

    accuracy                           0.75        20
   macro avg       0.76      0.77      0.75        20
weighted avg       0.79      0.75      0.75        20
```

# Homework

A. Diberikan data pada tautan berikut ini https://www.kaggle.com/uciml/pima-indians-diabetes-database

1. Buatlah model Logistic Regression dengan menggunakan fitur pregnant & glucose
2. Buatlah model Logistic Regression dengan menggunakan fitur bp & insulin
3. Buatlah model Logistic Regression dengan menggunakan fitur skin & BMI

Tambahkan spliting data training dan testing, serta evaluasi menggunakan data testing

In [10]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import plot_confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
import numpy as np
```

In [11]:
```
df = pd. read_csv('diabetes.csv')
print(df)
print(df.isna().sum())
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
```

```
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

**SOAL 1**

In [12]:

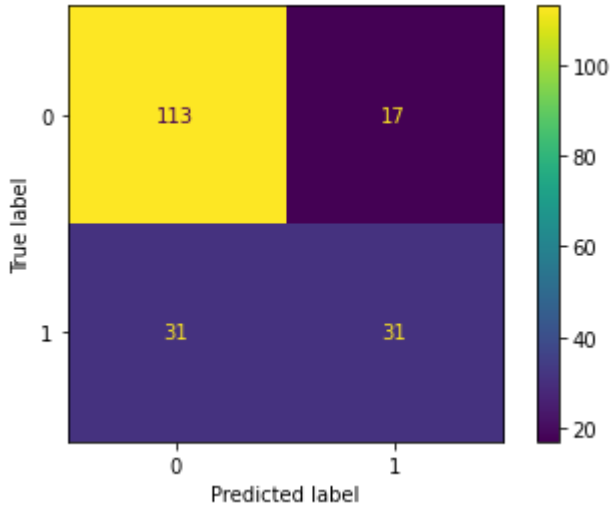
```
x=df[["Pregnancies","Glucose"]]
y=df["Outcome"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [13]:

```
logisticRegr=LogisticRegression()
logisticRegr.fit(x_train,y_train)
Y_predic=logisticRegr.predict(x_test)
plot_confusion_matrix(logisticRegr,x_test,y_test)
print(classification_report(y_test,Y_predic))
```

```
              precision    recall  f1-score   support

           0       0.78      0.87      0.82       130
           1       0.65      0.50      0.56        62

    accuracy                           0.75       192
   macro avg       0.72      0.68      0.69       192
weighted avg       0.74      0.75      0.74       192
```

```
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.
py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
onfusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the cla
ss methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
```



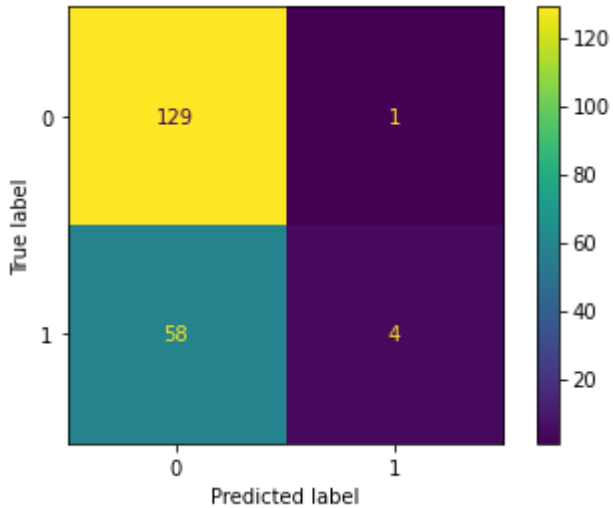

**SOAL 2**

In [14]:
```
x=df[["BloodPressure","Insulin"]]
y=df["Outcome"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [15]:
```
logisticRegr=LogisticRegression()
logisticRegr.fit(x_train,y_train)
Y_predic=logisticRegr.predict(x_test)
plot_confusion_matrix(logisticRegr,x_test,y_test)
print(classification_report(y_test,Y_predic))
```

```
              precision    recall  f1-score   support

           0       0.69      0.99      0.81       130
           1       0.80      0.06      0.12        62

    accuracy                           0.69       192
   macro avg       0.74      0.53      0.47       192
weighted avg       0.73      0.69      0.59       192
```

```
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_confusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```



**SOAL 3**

In [16]:
```
x=df[["SkinThickness","BMI"]]
y=df["Outcome"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [17]:
```
logisticRegr=LogisticRegression()
logisticRegr.fit(x_train,y_train)
Y_predic=logisticRegr.predict(x_test)
plot_confusion_matrix(logisticRegr,x_test,y_test)
print(classification_report(y_test,Y_predic))
```

```
C:\Users\USER\anaconda3\envs\tensorflow\lib\site-packages\sklearn\utils\deprecation.
py:87: FutureWarning: Function plot_confusion_matrix is deprecated; Function `plot_c
onfusion_matrix` is deprecated in 1.0 and will be removed in 1.2. Use one of the cla
ss methods: ConfusionMatrixDisplay.from_predictions or ConfusionMatrixDisplay.from_e
stimator.
  warnings.warn(msg, category=FutureWarning)
              precision    recall  f1-score   support

           0       0.71      0.89      0.79       130
           1       0.52      0.24      0.33        62

    accuracy                           0.68       192
   macro avg       0.61      0.57      0.56       192
weighted avg       0.65      0.68      0.64       192
```