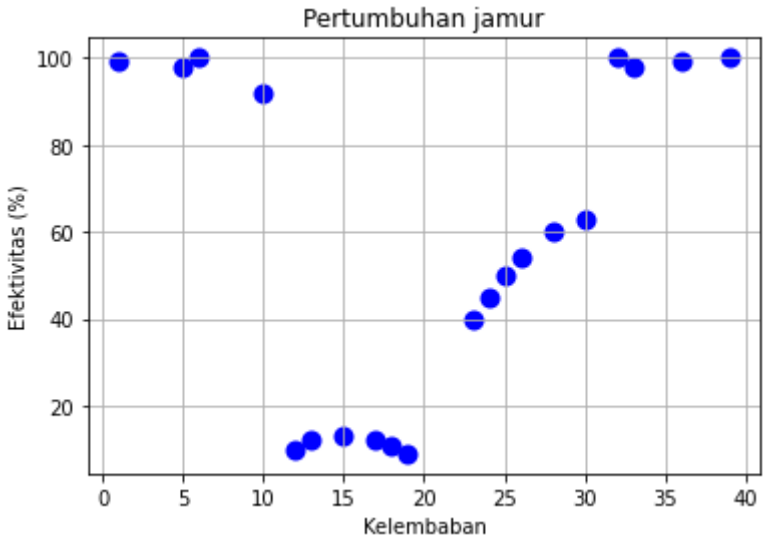
REGRESSION TREE

Menentukan data

In [38]:
```
import numpy as np
# variabel X harus dalam bentuk array 2D
X = np.array([[1, 5, 6, 10, 12, 13, 15, 17, 18, 19, 23, 24, 25, 26, 28, 30, 32,33, 3
# variabel y dalam bentuk array 1D
y = np.array([99, 98, 100, 92, 10, 12, 13, 12, 11, 9, 40, 45, 50, 54, 60, 63, 100, 9
```

Plot data

In [39]:
```
import matplotlib.pyplot as plt #library untuk plot

plt.scatter(X,y,  color='blue', lw=4)
plt.xlabel("Kelembaban")
plt.ylabel("Efektivitas (%)")
plt.title("Pertumbuhan jamur")
#plt.legend(('data',), loc='center left')
plt.grid()
plt.show()
```



Menentukan akar dari pohon

Pada langkah ini akan diberikan contoh menentukan akar dari pohon

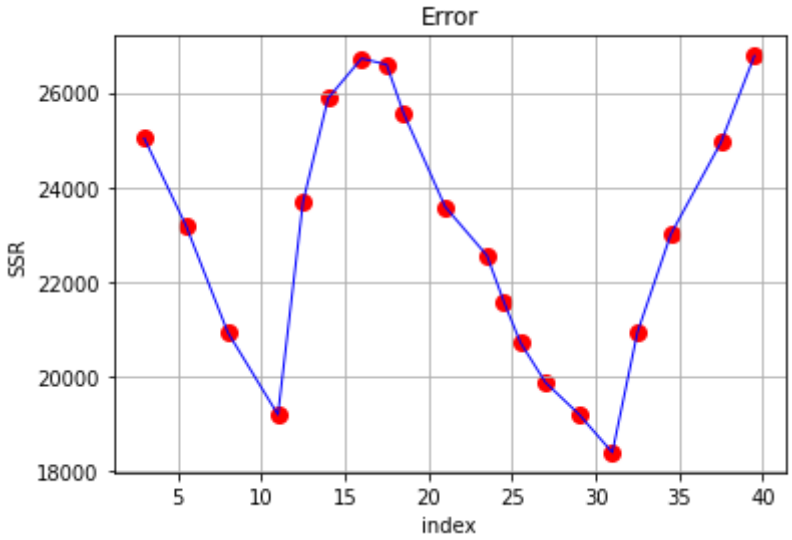
In [40]:
```
m = len(y)
SSR =np.array([])
xssr = np.array([])
for i in range(0,m):

  if (i==m-1):
    ybar = np.average(y)
    SSR = np.append(SSR, np.sum((y -ybar)**2))
    xssr = np.append(xssr,(X[i]+ (X[i]+1))/2)
  else:
    yn1 = y[0:i+1]
    ybar1 = np.average(yn1)
    yn2 = y[i+1:m]
    ybar2 = np.average(yn2)
```

```
    SSR = np.append(SSR, np.sum((yn1-ybar1)**2) + np.sum((yn2 -ybar2)**2))
    xssr = np.append(xssr,(X[i]+X[i+1])/2)
```

Ploting SSR

In [41]:

```
plt.scatter(xssr,SSR,  color='red', lw=3)
plt.plot(xssr,SSR,  color='blue', lw=1)
plt.xlabel("index")
plt.ylabel("SSR")
plt.title("Error")
plt.grid()
plt.show()
```



Mencari minimum nilai SSR

In [42]:

```
min_e = np.min(SSR)
min_pos = np.where(SSR == min_e)
print(SSR)
print("Minimum SSR is", min_e, "at x=",xssr[min_pos])
```

```
[25033.78947368 23181.61111111 20920.94117647 19176.75
 23685.73333333 25881.21428571 26723.71428571 26594.25
 25576.18181818 23581.3        22548.90909091 21580.91666667
 20707.23076923 19860.26190476 19188.         18376.75
 20920.94117647 23000.5        24946.94736842 26781.75      ]
Minimum SSR is 18376.75 at x= [31.]
```

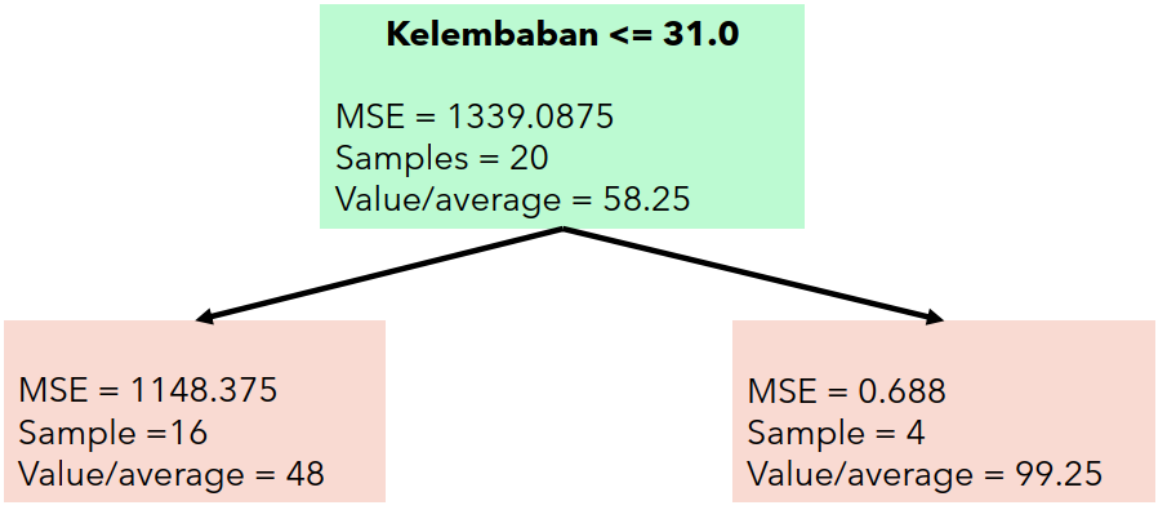
Mencari MSE dan nilai rataan akar

In [43]:

```
m = len(y)
value = np.average(y)
print('Value =', value)
MSE = np.sum((y-value)**2)/m
print('MSE =', MSE)
```

```
Value = 58.25
MSE = 1339.0875
```

Jadi akar dan anakan pohon yang dibentuk akan berbentuk berikut ini:



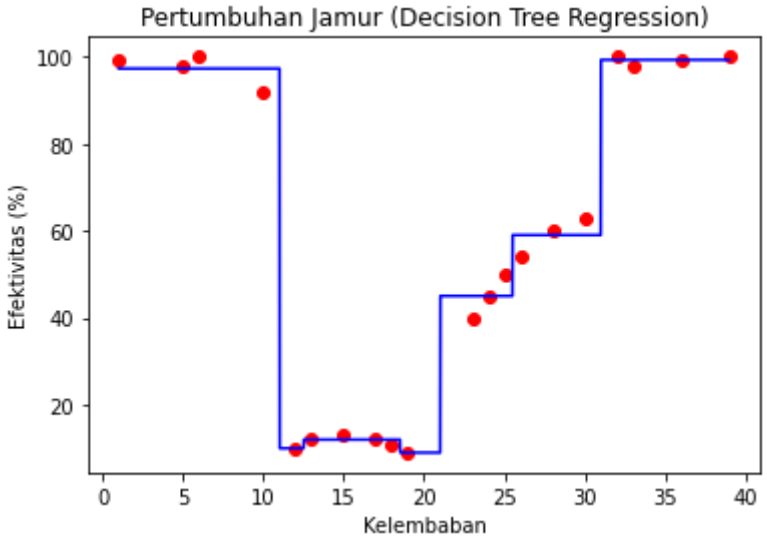
Dengan SKLEARN

In [44]:
```
# import the regressor
from sklearn.tree import DecisionTreeRegressor

# create a regressor object
regressor = DecisionTreeRegressor(random_state = 0, min_samples_split=5)

# fit the regressor with X and Y data
regressor.fit(X, y)
```

Out[44]:
```
DecisionTreeRegressor(min_samples_split=5, random_state=0)
```

Plot Data dan Regresi Tree

In [45]:
```
import matplotlib.pyplot as plt

X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))


plt.scatter(X, y, color = 'red')
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
plt.title('Pertumbuhan Jamur (Decision Tree Regression)')
plt.xlabel('Kelembaban')
plt.ylabel('Efektivitas (%)')
plt.show()
```



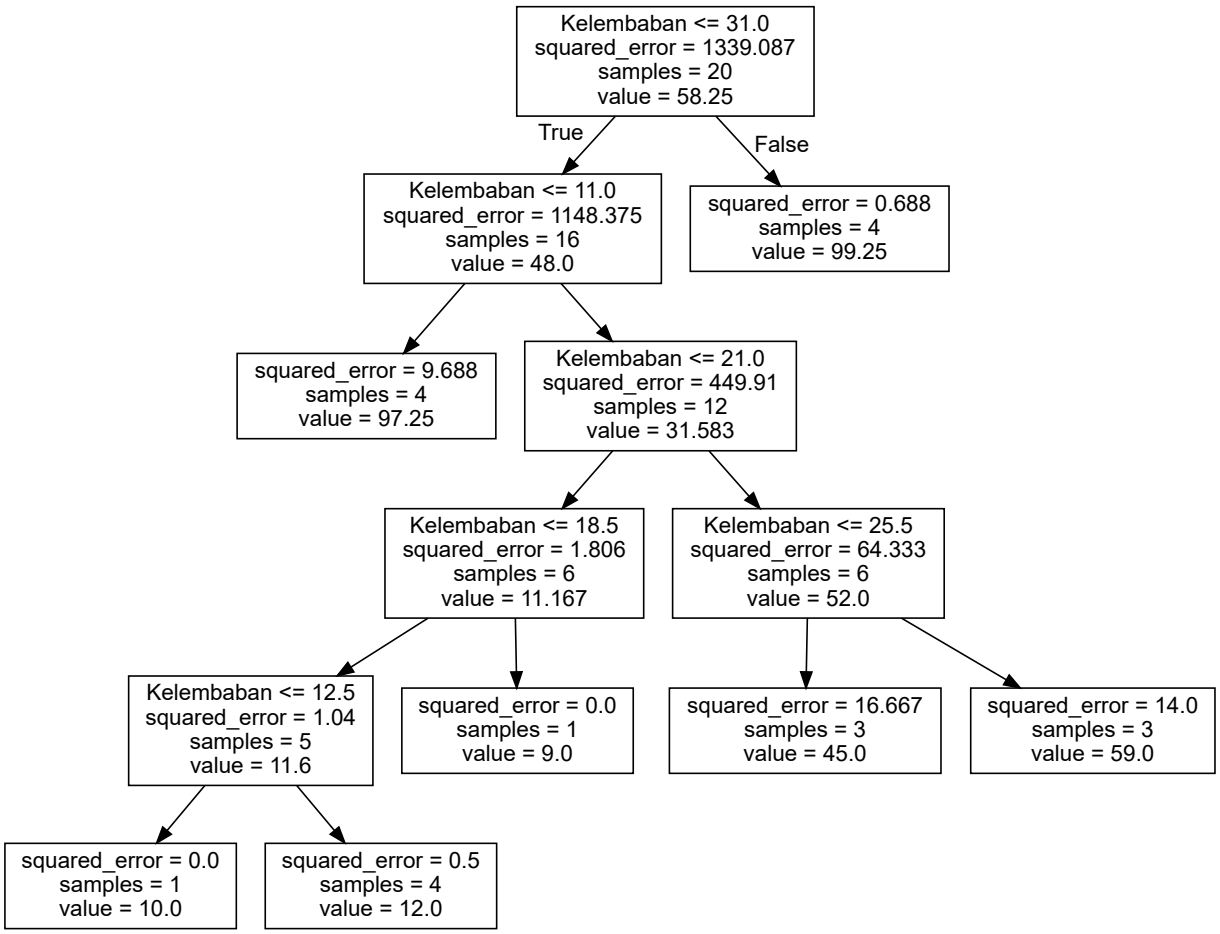
Menyimpan dala dalam format dot untuk visualisasi

In [46]:
```
# import export_graphviz
from sklearn.tree import export_graphviz
export_graphviz(regressor, out_file ='tree.dot',
                feature_names =['Kelembaban'])
```

Plot graph Tree

In [47]:
```
import graphviz
graphviz.Source.from_file('tree.dot')
```

Out[47]:



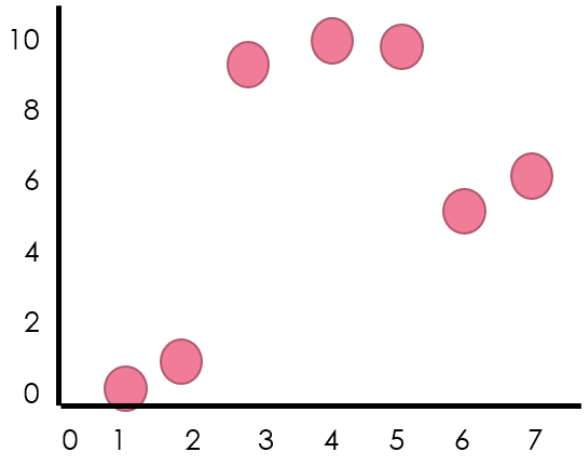
Contoh memprediksi

In [48]:
```
y_pred = regressor.predict([[10]])
print("Prediksi efektivitas: % d\n"% y_pred)
```

```
Prediksi efektivitas:  97
```

LATIHAN

Tentukan Root Node dari data berikut ini menggunakan Python



x	y
1	0
2	1
3	9
4	10
5	9.5
6	5
7	6

Machine Learning Approach

Langkah 1. Menyiapkan data

Pada langkah ini menyiapkan data advertising.

In [49]:
```
import pandas as pd

url = 'http://bit.ly/Test-PHN'
data = pd.read_csv(url, index_col=0)

data
```

Out[49]:

	TV	radio	newspaper	sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
...	...	...	...	...
196	38.2	3.7	13.8	7.6
197	94.2	4.9	8.1	9.7
198	177.0	9.3	6.4	12.8
199	283.6	42.0	66.2	25.5
200	232.1	8.6	8.7	13.4

200 rows × 4 columns

Langkah 2. Membagi data menjadi 80\% training dan 20\% testing

In [50]:
```
import numpy as np
msk = np.random.rand(len(data)) < 0.8
train = data[msk]
test = data[~msk]
test.head()
```

Out[50]:

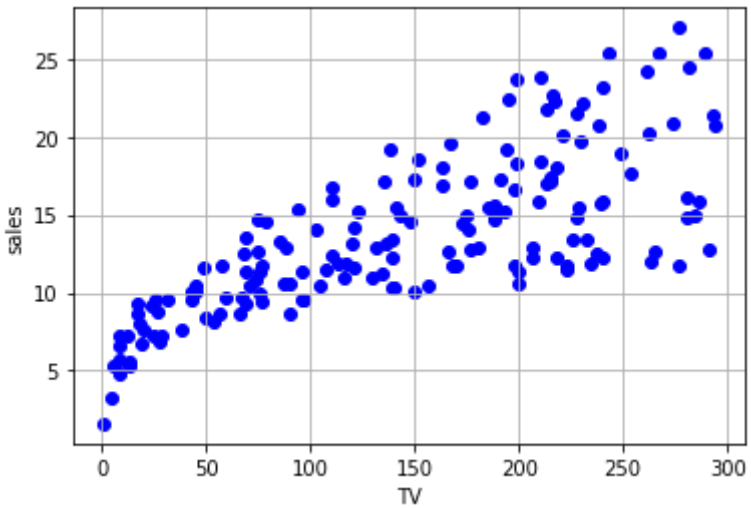
	TV	radio	newspaper	sales
14	97.5	7.6	7.2	9.7
15	204.1	32.9	46.0	19.0
25	62.3	12.6	18.3	9.7
34	265.6	20.0	0.3	17.4
41	202.5	22.3	31.6	16.6

Langkah 3. Menyiapkan data x (TV) dan y(sales)

In [51]:
```
m = len(train.TV) #number of rows data

X = np.asanyarray(train[['TV']])
y = np.asanyarray(train[['sales']])
```

plot sebaran data

In [52]:
```
import matplotlib.pyplot as plt #library untuk plot

plt.scatter(X,y,  color='blue')
plt.xlabel("TV")
plt.ylabel("sales")
plt.grid()
plt.show()
```



Langkah 4. Menentukan model yang akan digunakan.

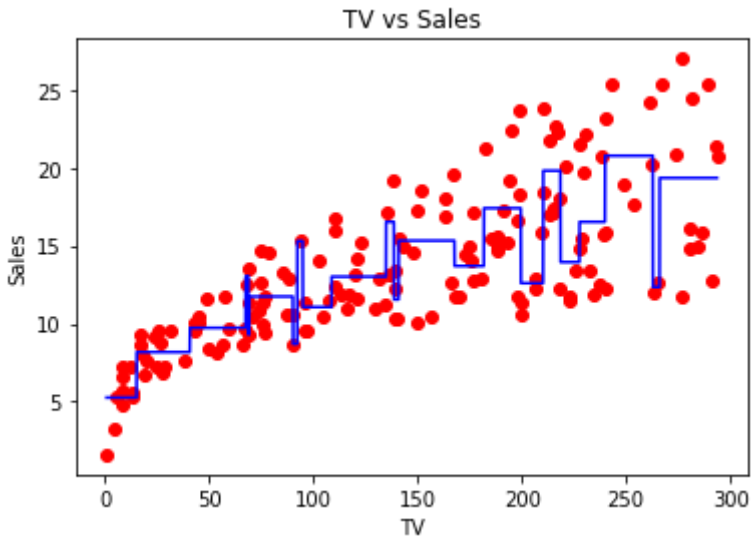
Menggunakan regression Tree

In [53]:
```
# import the regressor
```

```
from sklearn.tree import DecisionTreeRegressor

# create a regressor object
regressor = DecisionTreeRegressor(random_state = 0, min_samples_split=15)

# fit the regressor with X and Y data
regressor.fit(X, y)
```

Out[53]:
```
DecisionTreeRegressor(min_samples_split=15, random_state=0)
```

In [54]:
```
import matplotlib.pyplot as plt

X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))


plt.scatter(X, y, color = 'red')
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
plt.title('TV vs Sales')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.show()
```



Langkah 5. Mengevaluasi model

Menentukan tabel baru yang berisi data latih/testing

In [55]:
```
X = np.asanyarray(test[['TV']])
y = np.asanyarray(test[['sales']])
y = y.reshape(-1)

print(y)


y_pred = np.array([])
for i in range(0,len(y)):
  y_pred = np.append(y_pred, regressor.predict([X[i]]))

print(y_pred)
```

```
[ 9.7 19.   9.7 17.4 16.6  8.5 10.7  5.5 18.  18.9 12.  19.4 22.2 23.8
 19.8 21.8 15.9  6.6  7.  11.6  8.8 24.7 19.6 10.8 20.7 13.2 10.9 19.
 14.5 20.2 26.2 22.6 10.8  5.9  9.7 25.5]
[11.06666667 12.6         9.73333333 12.35       12.6         8.17692308
```

```
11.06666667  5.26       13.01538462 16.53       11.74285714 19.81111111
20.77142857 19.34615385 20.77142857 20.77142857 13.01538462  8.17692308
 8.17692308 13.95       11.74285714 13.95       13.95        8.17692308
17.40769231 20.77142857  8.17692308 13.7        15.32727273 20.77142857
19.34615385 12.6         8.17692308  8.17692308 15.3        19.34615385]
```

Find MSE training data

In [56]:

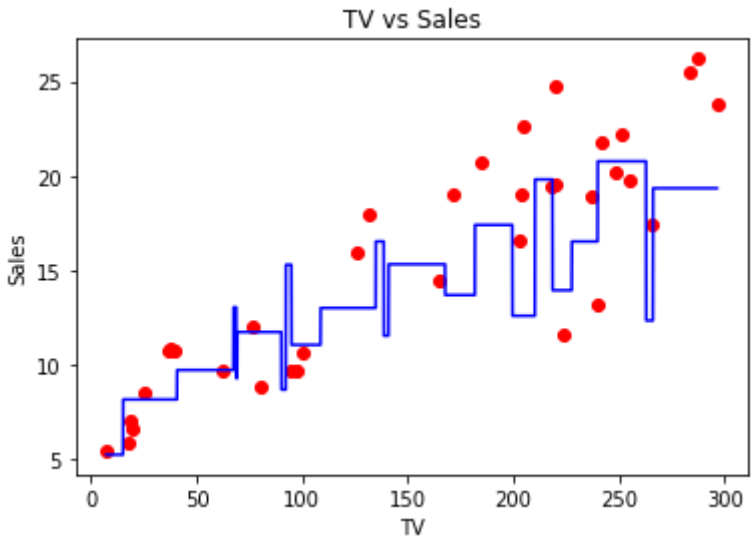
```
MSE = np.sum((y-y_pred)**2)/len(y)
print(MSE)
```

```
18.136070359244933
```

Plot test data and regression

In [57]:

```
import matplotlib.pyplot as plt

X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))


plt.scatter(X, y, color = 'red')
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
plt.title('TV vs Sales')
plt.xlabel('TV')
plt.ylabel('Sales')
plt.show()
```



Homework

1. Gunakan data Advertising (X=**radio** dan y=sales) untuk memodelkan Regresi pohon dalam pendekatan Machine Learning, evaluasi hitung MSE dan plot regresinya.
2. Gunakan data Advertising X=**TV dan radio**, y = sales untuk memodelkan Regresi pohon dalam Machine Learning, serta hitung MSEnya.
3. Gunakan data Advertising X=**TV,radio dan newspaper**, y =sales untuk memodelkan Regresi pohon dalam Machine Learning, serta hitung MSEnya.

REGRESSION TREE

1. Gunakan data Advertising (X=radio dan y=sales) untuk memodelkan Regresi pohon

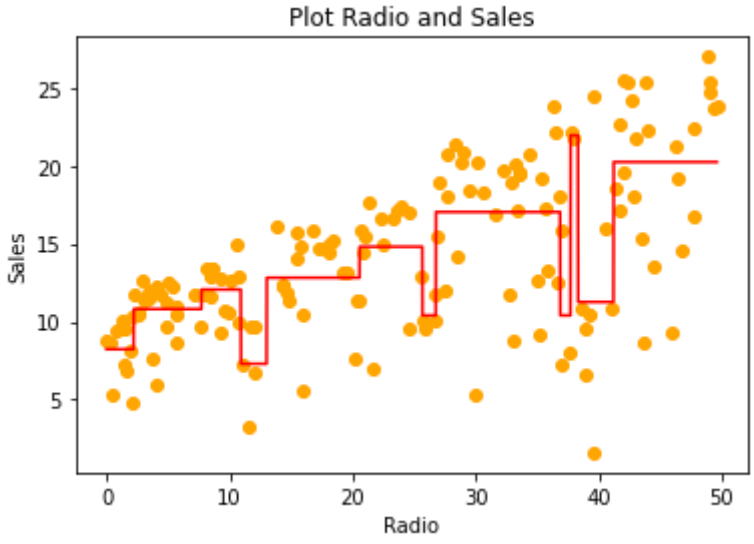
dalam pendekatan Machine Learning, evaluasi hitung MSE dan plot regresinya

In [58]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeRegressor
from sklearn import tree
```

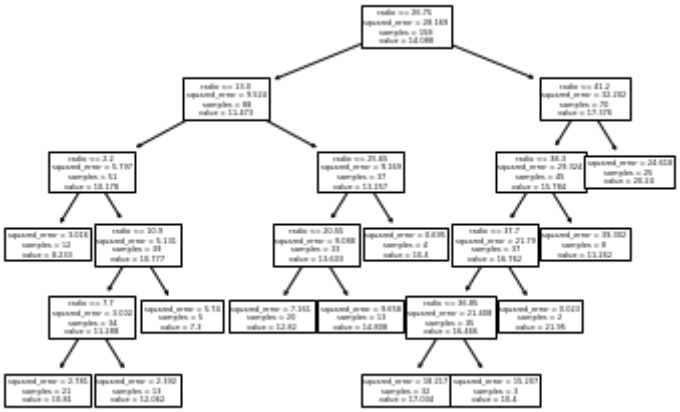
In [59]:
```
url = 'http://bit.ly/Test-PHN'
data = pd.read_csv(url, index_col=0)
```

In [60]:
```
msk = np.random.rand(len(data)) < 0.8
train = data[msk]
test = data[~msk]
m = len(train.radio) #number of rows data
X = np.asanyarray(train[['radio']])
y = np.asanyarray(train[['sales']])
regressor = DecisionTreeRegressor(random_state = 0, min_samples_split=33)
```

In [61]:
```
# fit the regressor with X and Y data
regressor.fit(X, y)
X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'orange')
plt.plot(X_grid, regressor.predict(X_grid), color = 'red')
plt.title('Plot Radio and Sales')
plt.xlabel('Radio')
plt.ylabel('Sales')
plt.show()
```



In [62]:
```
tree.plot_tree(regressor,feature_names=["radio"])
X = np.asanyarray(test[['radio']])
y = np.asanyarray(test[['sales']])
y = y.reshape(-1)
y_pred = np.array([])
for i in range(0,len(y)):
    y_pred = np.append(y_pred,regressor.predict([X[i]]) )
```



In [63]:
```
MSE = np.sum((y-y_pred)**2)/len(y)
print("MSE : ",MSE)
```

```
MSE :  19.282841879132906
```

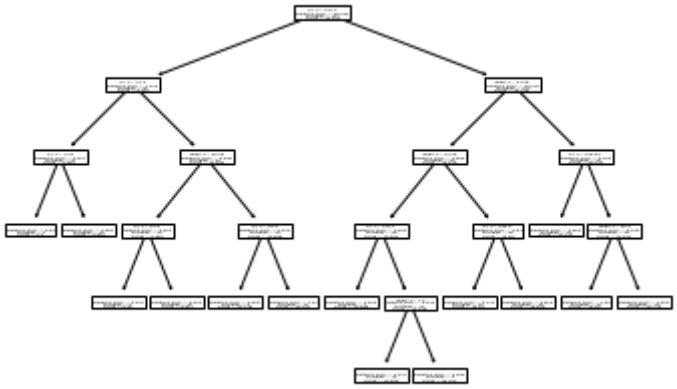
2. Gunakan data Advertising X=TV dan radio, y = sales untuk memodelkan Regresi pohon dalam Machine Learning, serta hitung MSEnya

In [64]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeRegressor
from sklearn import tree
```

In [65]:
```
url = 'http://bit.ly/Test-PHN'
data = pd.read_csv(url, index_col=0)
```

In [66]:
```
msk = np.random.rand(len(data)) < 0.8
train = data[msk]
test = data[~msk]
m = len(train.radio) #number of rows data
X= np.asanyarray(train[['radio',"TV"]])
y = np.asanyarray(train[['sales']])
regressor = DecisionTreeRegressor(random_state = 0, min_samples_split=20)
```

In [67]:
```
# fit the regressor with X and Y data
regressor.fit(X, y)
tree.plot_tree(regressor,feature_names=["radio","TV"])
X = np.asanyarray(test[['radio',"TV"]])
y = np.asanyarray(test[['sales']])
y = y.reshape(-1)
y_pred = np.array([])
for i in range(0,len(y)):
    y_pred = np.append(y_pred,regressor.predict([X[i]]))
```



In [68]:
```
MSE = np.sum((y-y_pred)**2)/len(y)
print("MSE : ",MSE)
```

```
MSE :  1.5460102681699355
```

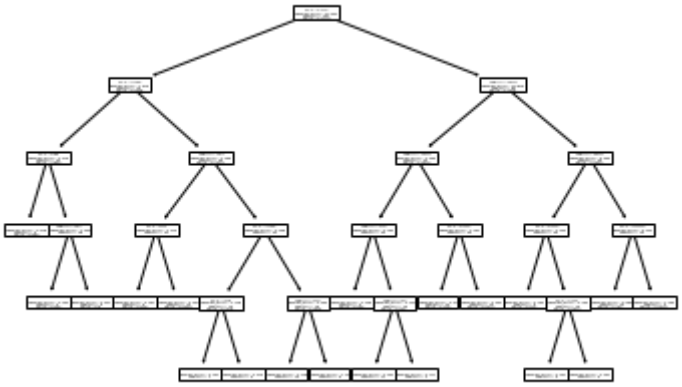
3. Gunakan data Advertising X=TV,radio dan newspaper, y =sales untuk memodelkan Regresi pohon dalam Machine Learning, serta hitung MSEnya

In [69]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeRegressor
from sklearn import tree
```

In [70]:
```
url = 'http://bit.ly/Test-PHN'
data = pd.read_csv(url, index_col=0)
```

In [71]:
```
msk = np.random.rand(len(data)) < 0.8
train = data[msk]
test = data[~msk]
m = len(train.radio) #number of rows data
X= np.asanyarray(train[['radio',"TV","newspaper"]])
y = np.asanyarray(train[['sales']])
regressor = DecisionTreeRegressor(random_state = 0, min_samples_split=15)
```

In [72]:
```
# fit the regressor with X and Y data
regressor.fit(X, y)
tree.plot_tree(regressor,feature_names=["radio","TV","newspaper"])
X = np.asanyarray(test[['radio',"TV","newspaper"]])
y = np.asanyarray(test[['sales']])
y = y.reshape(-1)
y_pred = np.array([])
for i in range(0,len(y)):
    y_pred = np.append(y_pred,regressor.predict([X[i]]))
```



In [73]:

```
MSE = np.sum((y-y_pred)**2)/len(y)
print("MSE : ",MSE)
```

```
MSE :  2.4608699941318957
```