In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_openml
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LinearRegression
```

In [4]:
```
boston = fetch_openml(data_id=531, parser='auto')
data = pd.DataFrame(boston.data)
data.columns = boston.feature_names
data['PRICE'] = boston.target
```

In [6]:
```
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
x = pd.DataFrame(imputer.fit_transform(data.drop(['PRICE'], axis=1)),
columns=data.drop(['PRICE'], axis=1).columns)
y = data['PRICE']
data = data.dropna()
x = x.loc[data.index]
```

In [8]:
```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2,
random_state=0)
```

In [10]:
```
lm = LinearRegression()
model=lm.fit(xtrain, ytrain)
ytrain_pred = lm.predict(xtrain)
ytest_pred = lm.predict(xtest)
df=pd.DataFrame(ytrain_pred,ytrain)
df=pd.DataFrame(ytest_pred,ytest)
```

In [12]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(ytest, ytest_pred)
print(mse)
mse = mean_squared_error(ytrain_pred,ytrain)
print(mse)
```

```
33.44897999767632
19.326470203585725
```

In [14]:
```
mse = mean_squared_error(ytest, ytest_pred)
print(mse)
```

```
33.44897999767632
```

In [15]:
```
plt.scatter(ytrain ,ytrain_pred,c='blue',marker='o',label='Training data')
plt.scatter(ytest,ytest_pred ,c='lightgreen',marker='s',label='Test data')
plt.xlabel('True values')
plt.ylabel('Predicted')
plt.title("True value vs Predicted value")
plt.legend(loc= 'upper left')
#plt.hlines(y=0,xmin=0,xmax=50)
plt.plot()
plt.show()
```