

In [6]:

```
import pandas as pd
import numpy as np
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
data = pd.read_csv("train.csv")
X = data.iloc[:,0:20] #independent columns
y = data.iloc[:, -1] # target column i.e price range
# apply SelectKBest class to extract
bestfeatures = SelectKBest(score_func=chi2, k=10)
fit = bestfeatures.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Specs','Score'] #naming the dataframe columns
print(featureScores.nlargest(10,'Score'))
```

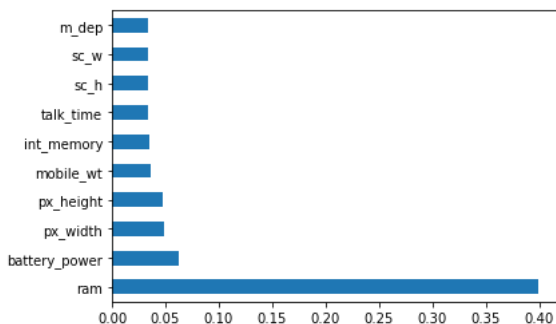
	Specs	Score
13	ram	931267.519053
11	px_height	17363.569536
0	battery_power	14129.866576
12	px_width	9810.586750
8	mobile_wt	95.972863
6	int_memory	89.839124
15	sc_w	16.480319
16	talk_time	13.236400
4	fc	10.135166
14	sc_h	9.614878

Hasil dari perintah `print(featureScores.nlargest(10,'Score'))` adalah mencetak 10 fitur dengan nilai skor terbesar. Dalam hal ini, fitur-fitur tersebut akan dicetak bersama dengan skor-skor mereka.

In [5]:

```
import pandas as pd
import numpy as np
data = pd.read_csv("train.csv")
X = data.iloc[:,0:20] #independent columns
y = data.iloc[:, -1] #target column i.e price range
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(X,y)
print(model.feature_importances_) #use inbuilt class feature_importances of tree based classifiers
#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()
```

```
[0.06304686 0.01992547 0.03259359 0.01931912 0.03157301 0.01694679
 0.03445171 0.03316588 0.03600421 0.03206886 0.03297056 0.04757852
 0.04818372 0.39911194 0.03399705 0.03341296 0.03403916 0.01383788
 0.01733739 0.02043532]
```



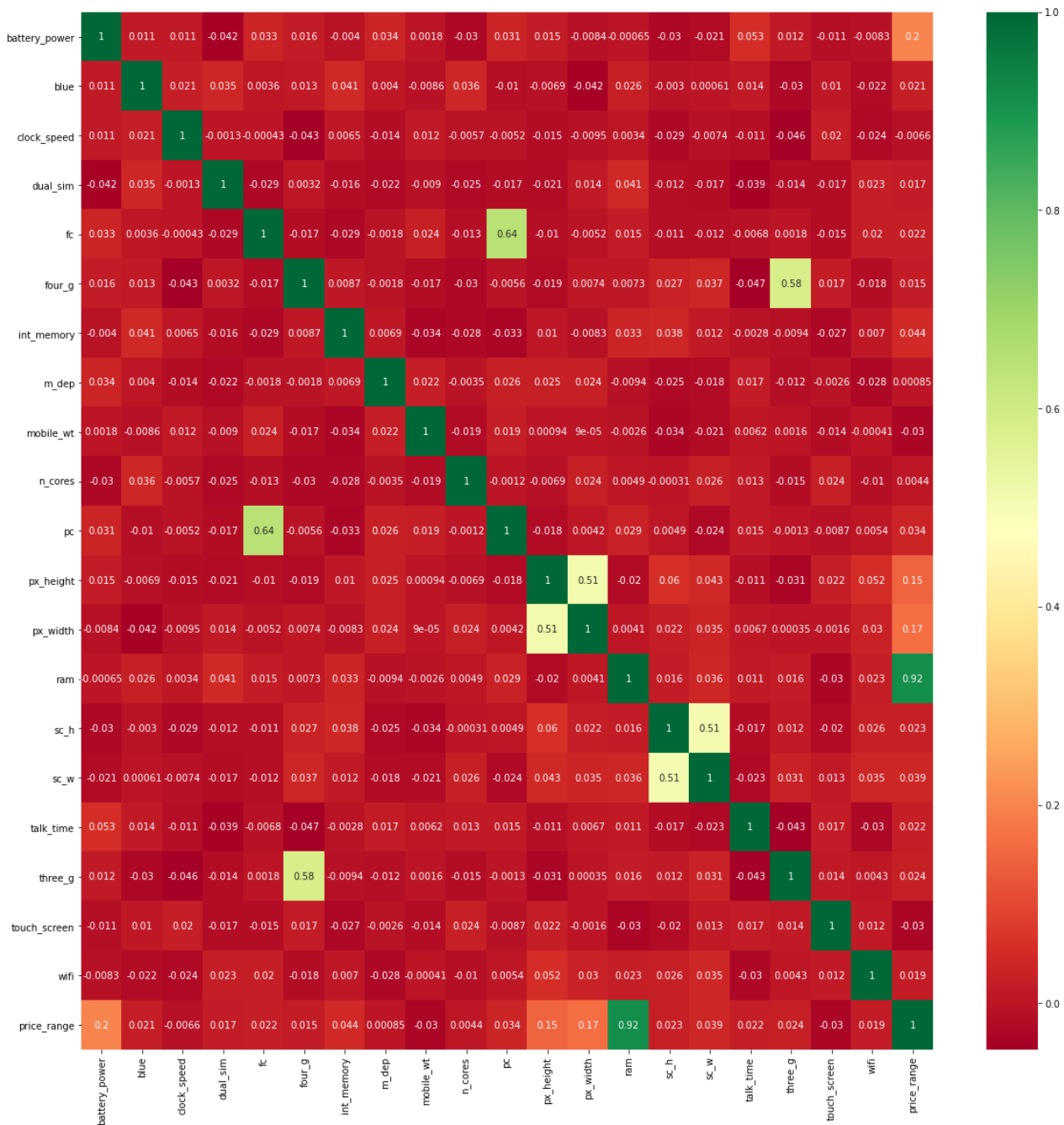
Hasil dari perintah `print(model.feature_importances_)` adalah mencetak nilai penting dari setiap fitur. Sedangkan hasil dari `feat_importances.nlargest(10).plot(kind='barh')` adalah membuat dan menampilkan grafik batang horizontal dari 10 fitur terpenting.

In [4]:

```

import pandas as pd
import numpy as np
import seaborn as sns
data = pd.read_csv("train.csv")
X = data.iloc[:,0:20] #independent columns
y = data.iloc[:,-1] #target column i.e price range
#get correlations of each features in dataset
corrmat = data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")

```



Hasil dari perintah ini adalah sebuah heatmap yang memvisualisasikan korelasi antar fitur. Warna dari setiap sel pada heatmap mencerminkan tingkat korelasi, di mana warna hijau muda menunjukkan korelasi positif, merah tua menunjukkan korelasi negatif, dan warna antara menunjukkan korelasi mendekati nol.

terkuat price range === ram

terlemah price range === mobile_wt & touch_screen