Klasifikasi Jenis Perumahan Menggunakan Metode *Decision Tree*

Irfan Purwo Saputro, Eka Mahendra Bagaskara

Teknik Informatika Politeknik Negeri Malang

22417271017, 22417171009

Abstrak **-Rumah adalah sebuah tempat di mana penghuninya akan mendapat perlindungan atau tempat bernaung dari segala kondisi alam yang berada di sekitarnya, seperti hujan, panas terik matahari, dan sebagainya. Rumah juga merupakan sesuatu yang dijadikan tempat beristirahat penghuninya yang telah melakukan berbagai macam aktivitas di luar yang pembuatannya dibuat berdasarkan pondasi bangunan. Maka dari dibuatlah suatu perusahaan yang bergerak di bidang properti di bidang perumahan, Perusahaan tersebut mempunyai beberapa tipe yang berbeda dan harga yang berbeda tergantung dari luas bangunan dan luas tanah. Untuk itu perlu dilakukan klasifikasi untuk menentukan jenis mana yang tepat untuk suatu keluarga dan uang yang bisa dikumpulkan untuk membeli rumah tersebut. Dalam hal ini dilakukan adanya metode dalam pengklasifikasian jenis rumah berdasarkan tipe yang tepat. Metode yang digunakan dalam studi kasus kali menggunakan Decision Tree. Decision tree merupakan salah satu model machine learning yang termasuk ke dalam supervised learning. decision tree berbentuk seperti graf yang memodelkan keputusan. Perhitungan Gini Impurity yaitu menghitung nilai kemurnian dari fitur yang akan dijadikan root. Perhitungan Gini Impurity yang mendekati nilai 0 yang akan dijadikan root atau node selanjutnya. Kesimpulan yang didapat setelah melakukan klasifikasi pada studi kasus perumahan dengan perhitungan Gini Impurity yaitu hasil yang didapat cukup akurat yaitu dengan perhitungan manual dengan kesimpulan bahwa untuk jenis rumah standar yaitu dengan luas bangunan kurang dari 60m2 dan harga kurang dari Rp. 475.000.000, untuk jenis medium yaitu luas bangunan kurang dari 60m2 dan harga lebih dari Rp. 475.000.000, untuk jenis premium yaitu dengan luas bangunan lebih dari 60m2 dan harga lebih dari Rp. 475.000.000.**

BAB I. PENDAHULUAN

Rumah adalah sebuah tempat di mana penghuninya akan mendapat perlindungan atau tempat bernaung dari segala kondisi alam yang berada di sekitarnya, seperti hujan, panas terik matahari, dan sebagainya. Rumah juga merupakan sesuatu yang dijadikan tempat beristirahat penghuninya yang telah melakukan berbagai macam aktivitas di luar yang pembuatannya dibuat berdasarkan pondasi bangunan.

Secara fisik rumah dapat diartikan sebagai suatu bangunan tempat kembali dari berpergian, bekerja, tempat tidur dan beristirahat memulihkan kondisi fisik dan mental yang letih dari melaksanakan tugas sehari-hari bagi penghuninya. Rumah juga menjadi sebuah tempat untuk ditinggali, serta untuk melakukan hal-hal tersebut di atas, dengan tentram, damai, serta menyenangkan bagi penghuninya. Dari pengertian secara psikologis ini lebih menitikberatkan pada situasi dan suasana fisik rumah itu sendiri.

Rumah sudah menjadi kebutuhan primer saat ini, tanpa adanya rumah manusia tidak akan mempunyai tempat yang bisa digunakan untuk menunjang aktivitas sehari-hari. Maka dari manusia perlu membangun rumah untuk tempat mereka melakukan aktivitas. Manusia tidak mungkin membangun rumahnya tanpa bantuan orang lain. Maka dari dibuatlah suatu perusahaan yang bergerak di bidang properti di bidang perumahan, Perusahaan tersebut mempunyai beberapa tipe yang berbeda dan harga yang berbeda tergantung dari luas bangunan dan luas tanah. Untuk itu perlu dilakukan klasifikasi untuk menentukan jenis mana yang tepat untuk suatu keluarga dan uang yang bisa dikumpulkan untuk membeli rumah tersebut.

BAB II. METODE

Dalam hal ini dilakukan adanya metode dalam pengklasifikasian jenis rumah berdasarkan tipe yang tepat. Metode yang digunakan dalam studi kasus kali menggunakan Decision Tree. Decision tree merupakan salah satu model machine learning yang termasuk ke dalam supervised learning. decision tree berbentuk seperti graf yang memodelkan keputusan. Digunakan untuk merepresentasikan keputusan dan pengambilan keputusan secara visual dan eksplisit. Decision tree disebut juga dengan istilah CART (Classification and Regression Trees) yang diperkenalkan oleh Leo Breiman untuk merujuk pada algoritma Decision Tree yang dapat digunakan untuk masalah klasifikasi atau regresi.

Pada studi kasus kali ini digunakan perhitungan Gini Impurity yaitu menghitung nilai kemurnian dari fitur yang akan dijadikan root. Perhitungan Gini Impurity yang mendekati nilai 0 yang akan dijadikan root atau node selanjutnya.

Dalam studi kasus kali ini diperoleh data sebagai berikut:

No	Tipe	Luas Tanah(m2)	Luas Bangunan(m2)	Harga	Jenis
1	Seruni	70	36	Rp350.000.000	Standar
2	Lily	75	39	Rp400.000.000	Standar
3	Anggrek	87	47	Rp450.000.000	Standar
4	Cendana	90	49	Rp500.000.000	Medium
5	Tulip	105	54	Rp600.000.000	Medium
6	Jasmine	54	54	Rp750.000.000	Premium
7	Lilac	75	36	Rp300.000.000	Standar
8	Morea	87	36	Rp400.000.000	Standar
9	Amarils	105	36	Rp450.000.000	Standar
10	Alamanda	90	36	Rp425.000.000	Standar
11	Morea B	105	36	Rp475.000.000	Medium
12	Alamanda B	93	36	Rp500.000.000	Medium
13	Ruha Ivy	54	54	Rp800.000.000	Medium
14	Dandelion	90	36	Rp575.000.000	Medium
15	Fresia	88	39	Rp500.000.000	Medium
16	Adenium	105	39	Rp625.000.000	Premium

17	Rafflesia	90	47	Rp600.000.000	Medium
18	Ruha	60	49	Rp700.000.000	Medium
19	Gardin	72	60	Rp800.000.000	Premium
20	Boulevard	90	60	Rp820.000.000	Premium
21	Marvella	90	70	Rp850.000.000	Premium
22	Magnolia	105	70	Rp880.000.000	Premium
23	Oliander	130	120	Rp900.000.000	Premium
24	Magnifera	145	120	Rp950.000.000	Premium

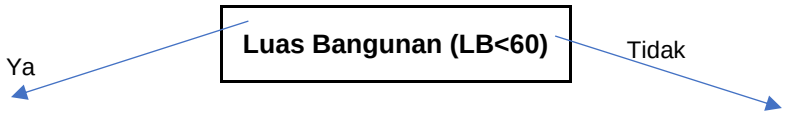
Pada perhitungan Gini Impurity pada data diatas pertama dilakukan adalah membuat Adjacent Average dari setiap fitur yang akan dijadikan parameter. Nilai adjacent average didapat dari nilai rata-rata suatu baris data terhadap baris selanjutnya maka diperoleh nilai seperti dibawah ini.

Tabel Adjacent Average

Data	**Nilai Rata-Rata**	
Row 1-2	72,5	
Row 2-3	81	
Row 3-4	88,5	
Row 4-5	97,5	
Row 5-6	79,5	

Row 6-7	64,5	
Row 7-8	81	
Row 8-9	96	
Row 9-10	97,5	
Row 10-11	97,5	
Row 11-12	99	
Row 12-13	73,5	
Row 13-14	72	
Row 14-15	89	
Row 15-16	96,5	
Row 16-17	97,5	
Row 17-18	75	
Row 18-19	66	
Row 19-20	81	
Row 20-21	90	
Row 21-22	97,5	
Row 22-23	117,5	
Row 23-24	137,5	

Kemudian setelah mengetahui nilai tersebut, maka dataset akan dikelompokkan berdasrkan nilai kurang dari nilai adjacent average seperti dibawah ini:



Jenis		
Standar	**Medium**	**Premium**
7	9	2

Jenis		
Standar	**Medium**	**Premium**
0	0	4

Pada data kali ini yaitu didapat nilai average yaitu 72,5 kemudian mengelompokkan nilai berdasarkan jenis perumahan yaitu standar, premium, dan medium yang kurang dari 72,5 dan lebih dari 72,5. Setelah ditentukan nilai Gini dari setiap jenis perumahan dengan rumus berikut ini:

GI "Standar" = 1 - (probabilitas "Standar") - (probabilitas "Medium") - (probabilitas "Premium)

$= 1 - (1/(1+2+2))^2 - (7/(6+7+6))^2 - (6/(6+7+6))^2$

$= 0{,}72454294$

GI "Medium" = 1 - (probabilitas "Medium") - (probabilitas "Standar") - (probabilitas "Premium)

$= 1 - (2/(1+2+2))^2 - (6/(6+7+6))^2 - (6/(6+7+6))^2$

$= 0{,}64055402$

GI "Premium" = 1 - (probabilitas "Premium") - (probabilitas "Standar") - (probabilitas "Medium)

$= 1 - (2/(1+2+2))^2 - (7/(6+7+6))^2 - (6/(6+7+6))^2$

$= 0{,}60454294$

GI "<72,5" $= ((7/24)*0{,}72454294) + ((9/24)*0{,}64055402) + ((8/24)*0{,}60454294$

$=$ **0,65304709**

Kemudian akan ditentukan nilai terkecil yang akan dijadikan nilai gini pada fitur tersebut.

Tabel Adjacent Average		
Data	**Nilai Rata-Rata**	**Gini Impurity**
Row 1-2	72,5	**0,653047091**
Row 2-3	81	**0,660375915**

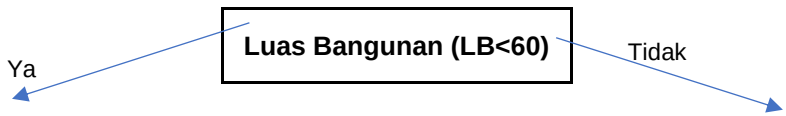
Row 3-4	88,5	**0,62829932**
Row 4-5	97,5	**0,598468799**
Row 5-6	79,5	**0,660375915**
Row 6-7	64,5	**0,574074074**
Row 7-8	81	**0,660375915**
Row 8-9	96	**0,598468799**
Row 9-10	97,5	**0,598468799**
Row 10-11	97,5	**0,598468799**
Row 11-12	99	**0,598468799**
Row 12-13	73,5	**0,653047091**
Row 13-14	72	**0,615982802**
Row 14-15	89	**0,62829932**
Row 15-16	96,5	**0,598468799**
Row 16-17	97,5	**0,598468799**
Row 17-18	75	**0,591280277**
Row 18-19	66	**0,574074074**
Row 19-20	81	**0,660375915**
Row 20-21	90	**0,451875**
Row 21-22	97,5	**0,598468799**

Row 22-23	117,5	**0,216253444**
Row 23-24	137,5	**0,218021424**
	GI "Luas Tanah"	**0,216253444**

Pada fitur Luas Tanah kali ini diperoleh nilai terkecil pada nilai gini setiap nilai average yaitu 0,216253444. Selanjtunya dilakukan pada fitur fitur yang akan dijadikan parameter dan terdapat nilai gini seperti di bawah ini:

Variabel		**Gini Impurity**
1.	Luas Tanah	0,216253444
2.	Luas Bangunan	0,026748971
3.	Harga	0,201292705
"Luas Bangunan"		**0,026748971**

Pada hasil tabel diatas didapatkan nilai gini dari setiap fitur. Nilai gini dari Luas Bangunan yang merupakan nilai terkecil akan dijadikan root node pada pembuatan decision tree.



Jenis		
Standar	**Medium**	**Premium**
7	9	2

Jenis		
Standar	**Medium**	**Premium**
0	0	4

Pada root node kali ini merupakan nilai gini dari Luas Bangunan yang nilainya didapat dari nilai gini Luas Bangunan kurang dari 60. Pada nilai tersebut didapatkan pada Luas Bangunan kurang dari 60 dengan jenis Standar berjumlah 7, jenis Medium berjumlah 9, dan jenis Premium yang berjumlah 2 yang belum merupakan nilai mutlak atau nilai murni. Kemudian untuk nilai yang lebih dari 60 yaitu

jenis Standar yaitu 0, jenis Medium yaitu 0 dan jenis Premium yaitu 0. Maka bisa disimpulkan untuk Luas Bangunan yang lebih dari 60 merupakan nilai mutlak dan bisa dijadikan sebagai leaf. Langkah selanjutnya yaitu menentukan nilai dari fitur yang akan dijadikan node selanjutnya yaitu Luas tanah dengan ketentuan Luas Bangunan kurang dari 60 dan Harga dengan Luas Bangunan kurang dari 60.Langkah yang dilakukan sama yaitu menentukan adjacent average dan kemudian menentukan nilai gini, maka nilai yang didapat yaitu:

Gini impurity Luas Tanah dengan Luas Bangunan <60

Tabel Adjacent Average		
Data	**Nilai Rata-Rata**	**Gini Impurity**
Row 1-2	72,5	**0,653047091**
Row 2-3	81	**0,660375915**
Row 3-4	88,5	**0,62829932**
Row 4-5	97,5	**0,598468799**
Row 5-6	79,5	**0,660375915**
Row 6-7	64,5	**0,574074074**
Row 7-8	81	**0,660375915**
Row 8-9	96	**0,598468799**
Row 9-10	97,5	**0,598468799**
Row 10-11	97,5	**0,598468799**
Row 11-12	99	**0,598468799**
Row 12-13	73,5	**0,653047091**
Row 13-14	72	**0,615982802**

Row 14-15	89	**0,62829932**
Row 15-16	96,5	**0,598468799**
Row 16-17	97,5	**0,598468799**
Row 17-18	75	**0,591280277**
	GI "Luas Tanah"	**0,574074074**

Gini impurity Harga dengan Luas Bangunan <60

Tabel Adjacent Average		
Data	**Nilai Rata-Rata**	**Gini Impurity**
Row 1-2	Rp375.000.000	**0,47899449**
Row 2-3	Rp425.000.000	**0,458795014**
Row 3-4	Rp475.000.000	**0,354166667**
Row 4-5	Rp550.000.000	**0,487379171**
Row 5-6	Rp675.000.000	**0,419176955**
Row 6-7	Rp525.000.000	**0,487379171**
Row 7-8	Rp350.000.000	**0,491477273**
Row 8-9	Rp425.000.000	**0,458795014**
Row 9-10	Rp437.500.000	**0,442059095**
Row 10-11	Rp450.000.000	**0,42733564**

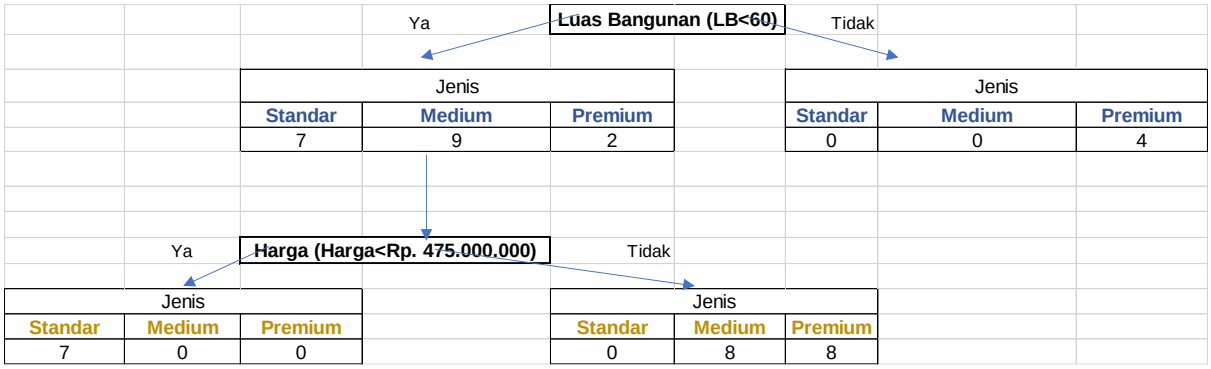
Row 11-12	Rp487.500.000	**0,447916667**
Row 12-13	Rp650.000.000	**0,419176955**
Row 13-14	Rp687.500.000	**0,419176955**
Row 14-15	Rp537.500.000	**0,487379171**
Row 15-16	Rp562.500.000	**0,487379171**
Row 16-17	Rp612.500.000	**0,381666667**
Row 17-18	Rp650.000.000	**0,419176955**
	GI "Harga"	**0,354166667**

Kemudian akan didapat nilai gini yang akan dijadikan node selanjutnya dengan nilai yang didapat yaitu:

Menentukan node selanjutnya

Variabel		**Gini Impurity**
1.	Luas Tanah	0,574074074
2.	Harga	0,354166667
"Harga"		**0,354166667**

Dari data diatas bis akita dapatkan yang akan menjadi node selanjutnya yaitu Harga dengan nilai gini 0,354166667. Maka bisa dapatkan decision tree yaitu:



Dari gambar diatas bisa digambarkan decision tree dengan node selanjutnya yaitu harga yang nilai gininya diperoleh dari harga kurang dari Rp. 475.000.000 dan diperoleh jenis rumah Standar 7 dan jenis Medium 0 dan jenis Premium 0. Kemudian harga lebih dari Rp. 475.000.000 diperoleh jenis rumah Standar 0, jenis Medium 8, dan jenis Premium 8. Nilai tersebut merupakan nilai mutlak atau murni yang dapat dijadikan leaf untuk decision tree tersebut dan tidak perlu di split.

Pengujian Decision Tree Pada Program Python

Import Library

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
```

Preprocessing

Import file dari google drive

```
from google.colab import drive
drive.mount('/content/drive')
```

Baca Dataset rumah csv dari file google drive yang menapilkan tipe perumahan, luas tanah, luas bangunan, harga, dan jenis perumahan

```
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/dataset_rumah.csv')
data.head()
```

	Tipe	Luas Tanah	Luas Bangunan	Harga	Jenis
0	Seruni	70	36	350000000	Standar
1	Lily	75	39	400000000	Standar
2	Anggrek	87	47	450000000	Standar
3	Cendana	90	49	500000000	Medium
4	Tulip	105	54	600000000	Medium

Lihat data type dari file csv

```
data.dtypes
```

```
Tipe              object
Luas Tanah         int64
Luas Bangunan      int64
Harga              int64
Jenis             object
dtype: object
```

Melakukan Format data dengan 4 variabel yaitu X1 luas tanah, X2 Luas Bangunan, X3 Harga, Y Jenis

```
from numpy.core.arrayprint import format_float_positional
y = data['Jenis'].map({'Standar':0, 'Medium':1, 'Premium':2})
X1 = data['Luas Tanah']
X2 = data['Luas Bangunan']
X3 = data['Harga']

v1=[]
v2=[]
v3=[]
for x1 in X1:
  v1.append(x1)
for x2 in X2:
  v2.append(x2)
for x3 in X3:
  v3.append(x3)
format_float_positional

X = pd.DataFrame()

X['Luas Tanah'] = v1
X['Luas Bangunan'] = v2
X['Harga'] = v3
print(X)
```

Hasil

	Luas Tanah	Luas Bangunan	Harga
0	70	36	350000000
1	75	39	400000000
2	87	47	450000000
3	90	49	500000000
4	105	54	600000000
5	54	54	750000000
6	75	36	300000000
7	87	36	400000000
8	105	36	450000000
9	90	36	425000000
10	105	36	475000000
11	93	36	500000000
12	54	54	800000000
13	90	36	575000000
14	88	39	500000000
15	105	39	625000000
16	90	47	600000000
17	60	49	700000000
18	72	60	800000000
19	90	60	820000000
20	90	70	850000000
21	105	70	880000000
22	130	120	900000000
23	145	120	950000000

Melakukan Training dan testing akurasi dengan decision tree classifier

```
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, shuffle=True)
dt = DecisionTreeClassifier()


dt.fit(X_train, y_train)


y_pred_dt = dt.predict(X_test)


acc_dt = accuracy_score(y_test, y_pred_dt)
print("Test set accuracy: {:.2f}".format(acc_dt))
print(f"Test set accuracy: {acc_dt}")
```

Hasil

Akurasi yang didapatkan dari testing data tersebut mendapatkan akurasi yang cukup baik dan hampir mendekati angka 0

```
Test set accuracy: 0.62
Test set accuracy: 0.625
```

Gini dan entropy

Training Gini dan Entropy, melakukan training dengan kedalaman maksimum 8

```
from sklearn.tree import DecisionTreeClassifier


dt_entropy = DecisionTreeClassifier(max_depth=8, criterion='entropy', random_state=10)


dt_entropy.fit(X_train, y_train)
```

```
dt_gini = DecisionTreeClassifier(max_depth=8, criterion='gini', random_state=10)
dt_gini.fit(X_train, y_train)
```

Testing akurasi Gini dan Entropy

```
from sklearn.metrics import accuracy_score
y_pred = dt_entropy.predict(X_test)
y_pred_gini = dt_gini.predict(X_test)

accuracy_entropy = accuracy_score(y_test, y_pred)
accuracy_gini = accuracy_score(y_test, y_pred_gini)

# Print accuracy_entropy
print("Accuracy achieved by using entropy: ", accuracy_entropy)

# Print accuracy_gini
print("Accuracy achieved by using gini: ", accuracy_gini)
```

Hasil

Akurasi yang diberikan oleh gini dan entropy adalah sama 0.625

```
Accuracy achieved by using entropy:  0.625
Accuracy achieved by using gini:  0.625
```

Training regression tree

```
from sklearn.tree import DecisionTreeRegressor

# Instantiate dt
dt = DecisionTreeRegressor(max_depth=8, min_samples_leaf=0.3, random_state=10)

# Fit dt ke dalam the training set
dt.fit(X_train, y_train)
```

Evaluasi Regression Tree

```
from sklearn.metrics import mean_squared_error

y_pred = dt.predict(X_test)
```

```
mse_dt = mean_squared_error(y_test, y_pred)

# Compute rmse_dt
rmse_dt = mse_dt ** (1/2)

# Print rmse_dt
print("Test set RMSE dt: {:.2f}".format(rmse_dt))
```

Hasil

```
Test set RMSE dt: 0.66
```

Training Linear Regression Tree

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()

lr.fit(X_train, y_train)
```

Testing Akurasi Regression Tree dan Linear Regression Tree

```
y_pred_lr = lr.predict(X_test)

mse_lr = mean_squared_error(y_test, y_pred_lr)

rmse_lr = mse_lr ** 0.5

print("Linear Regression test set RMSE: {:.2f}".format(rmse_lr))

print("Regression Tree test set RMSE: {:.2f}".format(rmse_dt))
```

Hasil

Hasil perbandingan dari regression tree test dan linear regreggsion test

```
Linear Regression test set RMSE: 0.29
Regression Tree test set RMSE: 0.66
```

BAB II. HASIL DAN KESIMPULAN

Kesimpulan yang didapat setelah melakukan klasifikasi pada studi kasus perumahan dengan perhitungan Gini Impurity yaitu hasil yang didapat cukup akurat yaitu dengan perhitungan manual dengan kesimpulan bahwa untuk jenis rumah standar yaitu dengan luas bangunan kurang dari 60m2 dan harga kurang dari Rp. 475.000.000, untuk jenis medium yaitu luas bangunan kurang dari 60m2 dan harga lebih dari Rp. 475.000.000, untuk jenis premium yaitu dengan luas bangunan lebih dari 60m2 dan harga lebih dari Rp. 475.000.000.