PENUGASAN ANALISA DATA PERFORMA SISWA SAAT MELAKSANAKAN UJIAN BERDASARKAN MEAN, MEDIAN, NILAI MIN DAN MAX SERTA VISUALISASI DATA

I. Profil Dataset

Dataset yang digunakan merupakan dataset yang berisi informasi sehubungan dengan performa ketika melakukan ujian. Adapun untuk menganalisa data yang ada, digunakan Bahasa pemrograman python karena terdapat berbagai macam library untuk menganalisa data yang cukup besar. Berikut tampilan data excel dalam format csv ketika di-import kedalam python:

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
5	female	group B	associate's degree	standard	none	71	83	78
6	female	group B	some college	standard	completed	88	95	92
7	male	group B	some college	free/reduced	none	40	43	39
8	male	group D	high school	free/reduced	completed	64	64	67
9	female	group B	high school	free/reduced	none	38	60	50

II. Cleaning Data

Ketika data tersebut diimport kedalam python otomatis akan dicari informasi yang terkait dengan data tersebut. Untuk menghindari kesalahan pada saat Analisa alangkah lebih baik jika melakukan cleaning data terlebih dahulu menggunakan script python sperti dibawah ini.

```
[104] # Membersihkan setiap baris yang didalamnya terdapat missing data
      df_clean = df.dropna()

      df_clean.describe()
```

Fungsi pada script diatas bertujuan untuk menghilangkan setiap baris yang memiliki kolom kosong atau memiliki tipe data seperti yang seharusnya untuk dihapus atau disesuaikan dengan nilai tertentu. Sehingga dari pembersihan data tersebut, diperoleh informasi sebagai berikut:

	math score	reading score	writing score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

Dari informasi diatas, secara keseluruhan siswa yang ada memiliki performa rata-rata tertinggi pada pelajaran *reading score* dengan nilai 69,17. Sedangkan untuk performa rata-rata terendah didapat pada pelajaran *math score* dengan nilai 66,09.

Keterangan	Math Score	Reading Score	Writing Score
Mean (Rata-rata)	66,09	69,17	68,05
Median (Nilai tengah)	66	70	69
Nilai Maksimum	100	100	100
Nilai Minimum	0	17	10

III. Nilai *Mean* dan *Median* yang ditentukan oleh klasifikasi *gender*

Laki-Laki / Male

Untuk mendapatkan data performa siswa ketika ujian dengan klasifikasi *gender* adalah laki-laki, maka digunakan cara sebagai berikut:

```
[106] df_clean_male = df_clean[df_clean["gender"].isin(["male"])]
      print('Tabel Siswa Laki-Laki')
      df_clean_male.head(10)
```

`df_clean_male` digunakan sebagai objek untuk menampung dari dataset keseluruhan dengan memfilter gender menggunakan `isin("male")` dengan menampilkan data head sebanyak 10 baris. Berikut visualisasi data *male* dalam bentuk tabel:

Tabel Siswa Laki-Laki

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75
7	male	group B	some college	free/reduced	none	40	43	39
8	male	group D	high school	free/reduced	completed	64	64	67
10	male	group C	associate's degree	standard	none	58	54	52
11	male	group D	associate's degree	standard	none	40	52	43
13	male	group A	some college	standard	completed	78	72	70
16	male	group C	high school	standard	none	88	89	86
18	male	group C	master's degree	free/reduced	completed	46	42	46
20	male	group D	high school	standard	none	66	69	63

Dari informasi pada tabel siswa laki-laki, dengan cara yang sama seperti mendapatkan sebaran data berupa *mean, median, max* dan *min* didapatkan hasil sebagai berikut:

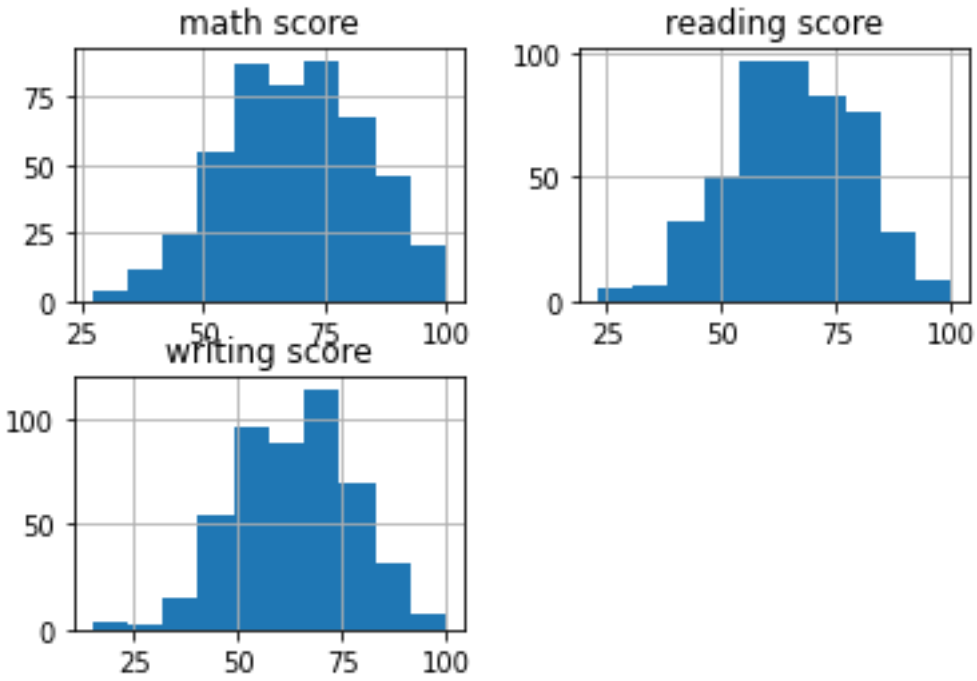
	math score	reading score	writing score
count	482.000000	482.000000	482.000000
mean	68.728216	65.473029	63.311203
std	14.356277	13.931832	14.113832
min	27.000000	23.000000	15.000000
25%	59.000000	56.000000	53.000000
50%	69.000000	66.000000	64.000000
75%	79.000000	75.000000	73.750000
max	100.000000	100.000000	100.000000

Dari informasi yang diperoleh pada tabel siswa laki-laki, secara keseluruhan siswa laki-laki memiliki performa rata-rata tertinggi pada pelajaran *math score* dengan nilai 68,73. Sedangkan untuk performa rata-rata terendah didapat pada pelajaran *writing score* dengan nilai 63,31. Untuk informasi lebih jelasnya akan dijelaskan pada tabel dibawah ini:

Keterangan	Math Score	Reading Score	Writing Score
Mean (Rata-rata)	68,73	65,47	63,31
Median (Nilai tengah)	69	66	64

Nilai Maksimum	100	100	100
Nilai Minimum	27	23	15
Standar Deviasi	14,35	13,9	14,11

Untuk mengetahui persebaran rata-rata pada masing-masing pelajaran untuk siswa laki-laki, maka akan disajikan data dalam bentuk histogram sebagai berikut:



Untuk subjek *math score* memiliki persebaran nilai rata-rata terjadi disekitar nilai tengah yang artinya berada disekitar kuartil 2 atau tepat sedikit dibawah nilai median pada data sebaran dengan standar deviasi 14,35. Sedangkan untuk subjek *reading score* memiliki persebaran nilai rata-rata terjadi disekitar nilai tengah yang artinya berada disekitar kuartil 2 dan kuartil 1 atau tepat sedikit dibawah nilai median pada data sebaran dengan standar deviasi 13,9. Sedangkan untuk subjek *writing score* memiliki persebaran nilai rata-rata terjadi disekitar nilai tengah yang artinya berada disekitar kuartil 2 dan kuartil 1 atau tepat sedikit dibawah nilai median pada data sebaran dengan standar deviasi 14,11.

Perempuan / Female

Untuk mendapatkan data performa siswa ketika ujian dengan klasifikasi *gender* adalah perempuan, maka digunakan cara sebagai berikut:

```
[108] df_clean_female = df_clean[df_clean["gender"].isin(["female"])]
      print('Tabel Siswa Perempuan')
      df_clean_female
```

`df_clean_female` digunakan sebagai objek untuk menampung dari dataset keseluruhan dengan memfilter gender menggunakan `isin("female")` dengan menampilkan data head sebanyak 10 baris. Berikut visualisasi data *male* dalam bentuk tabel:

Tabel Siswa Perempuan

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
5	female	group B	associate's degree	standard	none	71	83	78
6	female	group B	some college	standard	completed	88	95	92
9	female	group B	high school	free/reduced	none	38	60	50
12	female	group B	high school	standard	none	65	81	73
14	female	group A	master's degree	standard	none	50	53	58
15	female	group C	some high school	standard	none	69	75	78
17	female	group B	some high school	free/reduced	none	18	32	28

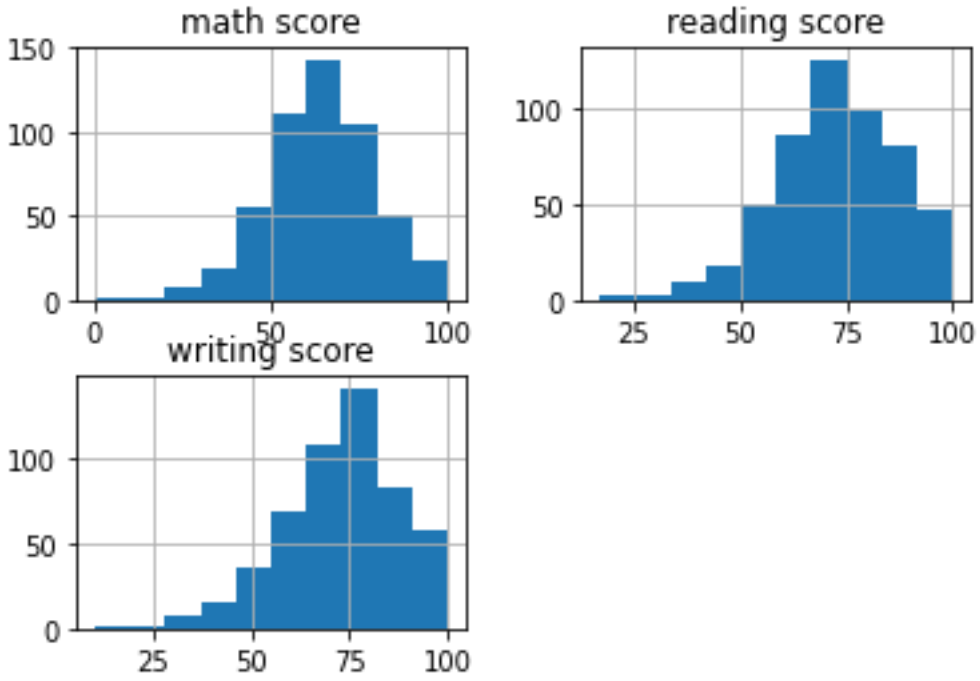
Dari informasi pada tabel siswa perempuan, dengan cara yang sama seperti mendapatkan sebaran data berupa *mean, median, max* dan *min* didapatkan hasil sebagai berikut:

	math score	reading score	writing score
count	518.000000	518.000000	518.000000
mean	63.633205	72.608108	72.467181
std	15.491453	14.378245	14.844842
min	0.000000	17.000000	10.000000
25%	54.000000	63.250000	64.000000
50%	65.000000	73.000000	74.000000
75%	74.000000	83.000000	82.000000
max	100.000000	100.000000	100.000000

Dari informasi yang diperoleh pada tabel siswa perempuan, secara keseluruhan siswa perempuan memiliki performa rata-rata tertinggi pada pelajaran *reading score* dengan nilai 72,6. Sedangkan untuk performa rata-rata terendah didapat pada pelajaran *math score* dengan nilai 63,63. Untuk informasi lebih jelasnya akan dijelaskan pada tabel dibawah ini:

Keterangan	Math Score	Reading Score	Writing Score
Mean (Rata-rata)	63,63	72,6	72,47
Median (Nilai tengah)	65	73	74
Nilai Maksimum	100	100	100
Nilai Minimum	0	17	10
Standar Deviasi	15,5	14,38	14,84

Untuk mengetahui persebaran rata-rata pada masing-masing pelajaran untuk siswa perempuan, maka akan disajikan data dalam bentuk histogram sebagai berikut:

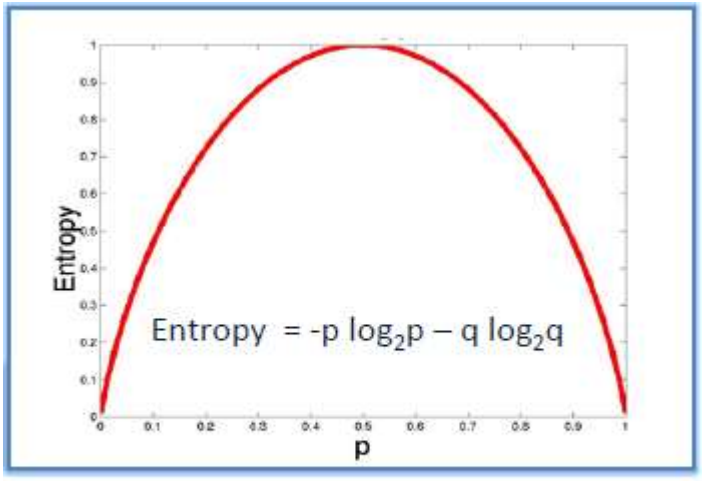


Untuk subjek *math score* memiliki persebaran nilai rata-rata terjadi dibawah *median* yang artinya berada disekitar kuartil 1 atau tepat sedikit dibawah median pada data sebaran dengan standar deviasi 15,5. Sedangkan untuk subjek *reading score* memiliki persebaran nilai rata-rata terjadi dibawah *median* yang artinya berada di kuartil 3 atau tepat sedikit dibawah nilai median pada data sebaran dengan standar deviasi 14,38. Sedangkan untuk subjek *writing score* memiliki persebaran nilai rata-rata terjadi dibawah nilai tengah yang artinya berada di kuartil 3 atau tepat sedikit dibawah nilai median pada data sebaran dengan standar deviasi 14,84

IV. Algoritma *Decision Tree*

Decision Tree atau Pohon Keputusan merupakan salah satu cara Data Mining dalam memprediksi masa depan dengan membangun klasifikasi atau regresi model dalam bentuk struktur pohon. Hal tersebut dilakukan dengan cara memecah terus ke dalam himpunan bagian yang lebih kecil lalu pada saat itu juga sebuah pohon keputusan secara bertahap dikembangkan. Hasil akhir dari proses tersebut adalah pohon dengan node keputusan dan node daun. Sebuah node keputusan (misalnya, Cuaca/ Outlook) memiliki dua atau lebih cabang (misalnya, Panas, Berawan dan Hujan). Node daun (misalnya, Bermain) merupakan klasifikasi atau keputusan. Node keputusan paling atas di pohon yang sesuai dengan prediktor terbaik disebut simpul akar. pohon keputusan dapat menangani data deskriptif maupun numerik.

Algoritma inti untuk membangun pohon keputusan disebut ID3 oleh J. R. Quinlan dengan cara kerja top-down pada pencarian greedy melalui ruang cabang tanpa teknik backtracking. ID3 menggunakan *Entropi* dan *Information Gain* untuk membangun pohon keputusan. Sebuah pohon keputusan dibangun secara top-down dari simpul akar dan melibatkan partisi data ke dalam himpunan bagian yang berisi contoh dengan nilai yang sama (homogen). Algoritma ID3 menggunakan entropi untuk menghitung homogenitas sampel. Jika sampel sudah benar-benar homogen, entropi bernilai nol dan jika pembagian sampel dibagi rata maka entropi bernilai satu.



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

Untuk membangun pohon keputusan, kita perlu menghitung dua jenis entropi menggunakan tabel frekuensi sebagai berikut:

a) Entropi yang menggunakan tabel frekuensi satu atribut

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

b) Entropi yang menggunakan tabel frekuensi dua atribut

$$E(T,X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Prediction and Modelling

Decision Tree hanya bisa mengolah data numerik, apabila data berupa string maka harus dirubah terlebih dahulu ke bentuk numerik. Pada penyelesaian dataset performa siswa ketika ujian pada kali ini akan dicari mana fitur yang memiliki data numerik dengan *script* seperti dibawah ini:

```
[140] num_of_numerical_cols = df_clean._get_numeric_data().columns.shape[0]
      print(num_of_numerical_cols, 'numerical columns')
      name_of_numerical_cols = df_clean._get_numeric_data().columns
      print(name_of_numerical_cols)
```

```
3 numerical columns
Index(['math score', 'reading score', 'writing score'], dtype='object')
```

Berdasarkan *output* yang diperoleh, index yang memiliki tipe data numerik ada pada kolom subjek mata pelajaran yang diikuti oleh masing-masing siswa. Oleh karena itu, pada penyelesaian kali ini akan diambil *Explanatory Variables Categorical* ada pada *gender* sedangkan *Explanatory Variables Quantitative* ada pada subjek masing-masing pelajaran.

Split data menjadi Training Data dan Test Data

Disini data akan dibagi menjadi dua bagian, yaitu training data dan test data dengan perbandingan 80 % training data dan 20 % test data. Dari data inilah yang akan dijadikan acuan sebagai model classifiernya. Berikut cara untuk membagi data menjadi presentase training dan test yang telah ditetapkan.

```
[142] # Split into training and testing set

      predictors = df_clean[['math score', 'reading score', 'writing score']] # explanatory variable
      targets = df_clean.gender

      X_train, X_test, y_train, y_test = train_test_split(predictors, targets, test_size = .2, random_state = 0)
```

```
[143] print('X_train = ', X_train.shape)
      print('X_test = ', X_test.shape)
      print('y_train = ', y_train.shape)
      print('y_test = ', y_test.shape)
```

```
X_train =  (800, 3)
X_test =  (200, 3)
y_train =  (800,)
y_test =  (200,)
```

Melatih Model dan Tes Data

- Baris pertama dari *confusion array* diatas merupakan prediksi siswa yang memiliki kriteria baik: Melalui 89 pengamatan dengan benar diklasifikasi sebagai siswa yang baik (*true negatives*) dan 10 pengamatan dengan salah diklasifikasi sebagai siswa yang buruk (*false positive*).
- Baris kedua dari *confusion array* diatas merupakan prediksi siswa yang memiliki kriteria buruk: Melalui 20 pengamatan dengan benar diklasifikasi sebagai siswa yang baik (*false negatives*) dan 81 pengamatan dengan benar diklasifikasi sebagai siswa yang buruk (*true positive*).

```
[144] classifier = DecisionTreeClassifier()
      classifier.fit(X_train, y_train)
      predictions = classifier.predict(X_test)
      confussion_array = sklearn.metrics.confusion_matrix(y_test, predictions)

      print(confussion_array)
```

```
[[89 10]
 [20 81]]
```

```
[145] print('TN = ', confussion_array[0,0]) # true negative
      print('FN = ', confussion_array[1,0]) # false negative
      print('TP = ', confussion_array[1,1]) # true positive
      print('FP = ', confussion_array[0,1]) # false positive
```

```
TN =  89
FN =  20
TP =  81
FP =  10
```

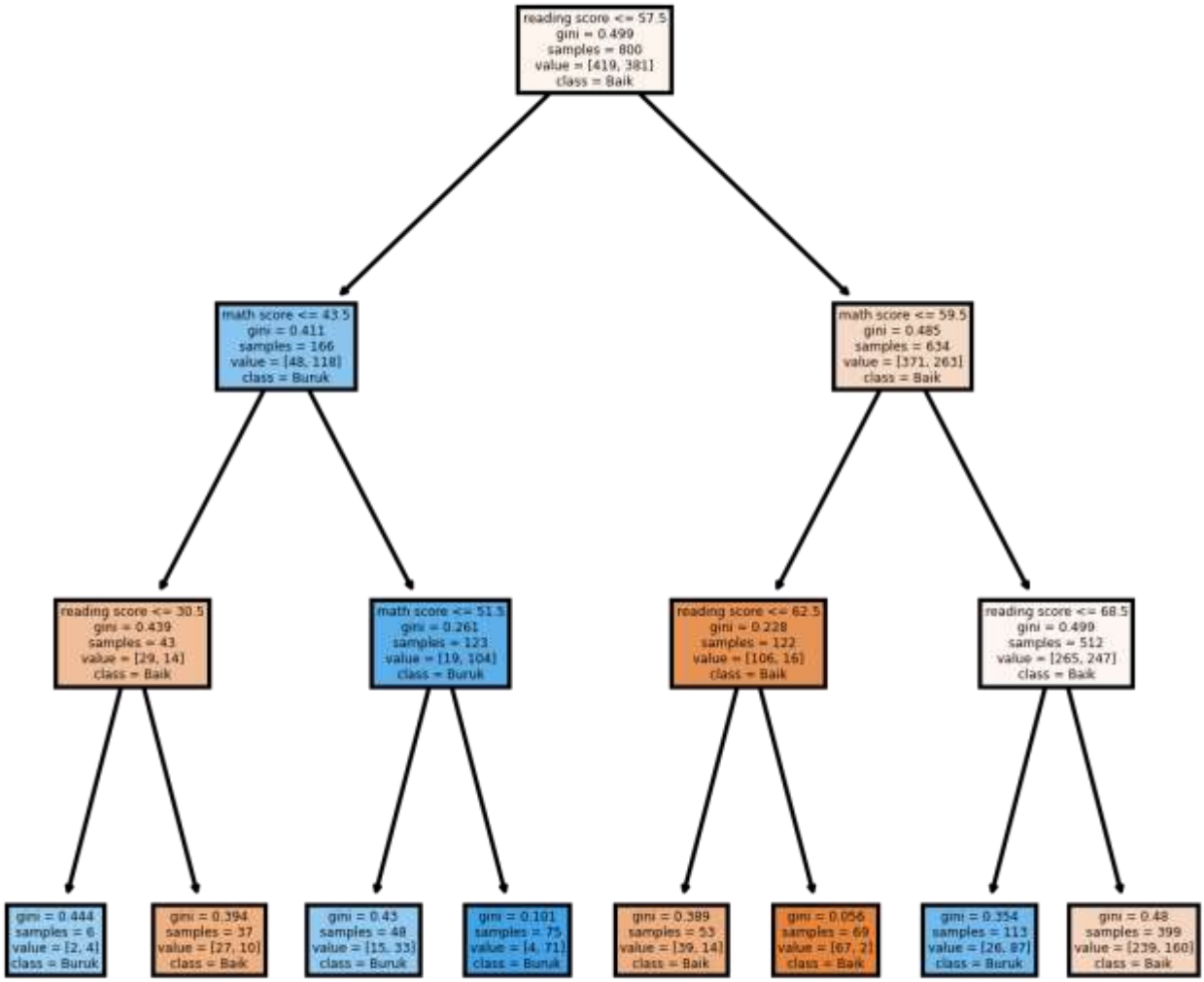
- Selanjutnya akan dihitung keakuratan dari model yang telah dibuat melalui cara sebagai berikut:

```
[146] # hitung accuracy
      print('Akurasi = ', sklearn.metrics.accuracy_score(y_test, predictions))
      print('Error = ', 1 - sklearn.metrics.accuracy_score(y_test, predictions))
```

```
Akurasi =  0.85
Error =  0.15000000000000002
```

Visualisasi Decision Tree

Decision tree yang dibuat dibawah ini adalah dengan mengambil data numerik antara nilai *math score* dan *reading score* saja. Untuk hasil pada *decision tree* akan dihitung nilai gini atau nilai entropi harus berbanding lurus dengan jumlah sampel yang diambil. Apabila nilai sampel banyak tetapi menghasilkan nilai entropi yang kecil, maka dapat dikatakan keputusan yang diambil adalah buruk sehingga bukan merupakan prediksi siswa yang dimaksud.


reading score <= 57.5
gini = 0.499
samples = 800
value = [419, 381]
class = Baik
math score <= 43.5
gini = 0.411
samples = 166
value = [48, 118]
class = Buruk
math score <= 59.5
gini = 0.485
samples = 634
value = [371, 263]
class = Baik
reading score <= 30.5
gini = 0.439
samples = 43
value = [29, 14]
class = Baik
math score <= 51.5
gini = 0.261
samples = 123
value = [19, 104]
class = Buruk
reading score <= 62.5
gini = 0.228
samples = 122
value = [106, 16]
class = Baik
reading score <= 68.5
gini = 0.499
samples = 512
value = [265, 247]
class = Baik
gini = 0.444
samples = 6
value = [2, 4]
class = Buruk
gini = 0.394
samples = 37
value = [27, 10]
class = Baik
gini = 0.43
samples = 48
value = [15, 33]
class = Buruk
gini = 0.101
samples = 75
value = [4, 71]
class = Buruk
gini = 0.389
samples = 53
value = [39, 14]
class = Baik
gini = 0.056
samples = 69
value = [67, 2]
class = Baik
gini = 0.354
samples = 113
value = [26, 87]
class = Buruk
gini = 0.48
samples = 399
value = [239, 160]
class = Baik

IV. Kesimpulan

Untuk melakukan klasifikasi pada suatu dataset, perlu diketahui dulu maksud dan tujuan untuk mengelola suatu data mulai dari input hingga output yang diharapkan. Maksud dari tujuan ini adalah untuk memilih algoritma yang lebih baik untuk melakukan suatu prediksi dengan benar sehingga didapatkan kesimpulan yang diharapkan.

Nama *: Dimas Ari Nugroho*
Tanggal Tugas *: 01 Oktober 2020 selesai pada 03 Oktober 2020*
Posisi yang dilamar *: Data Science*
Penugasan *: Mean, Median, Max, and Min*

https://github.com/dimasarinugroho/Project/blob/master/PenugasanBRI.ipynb