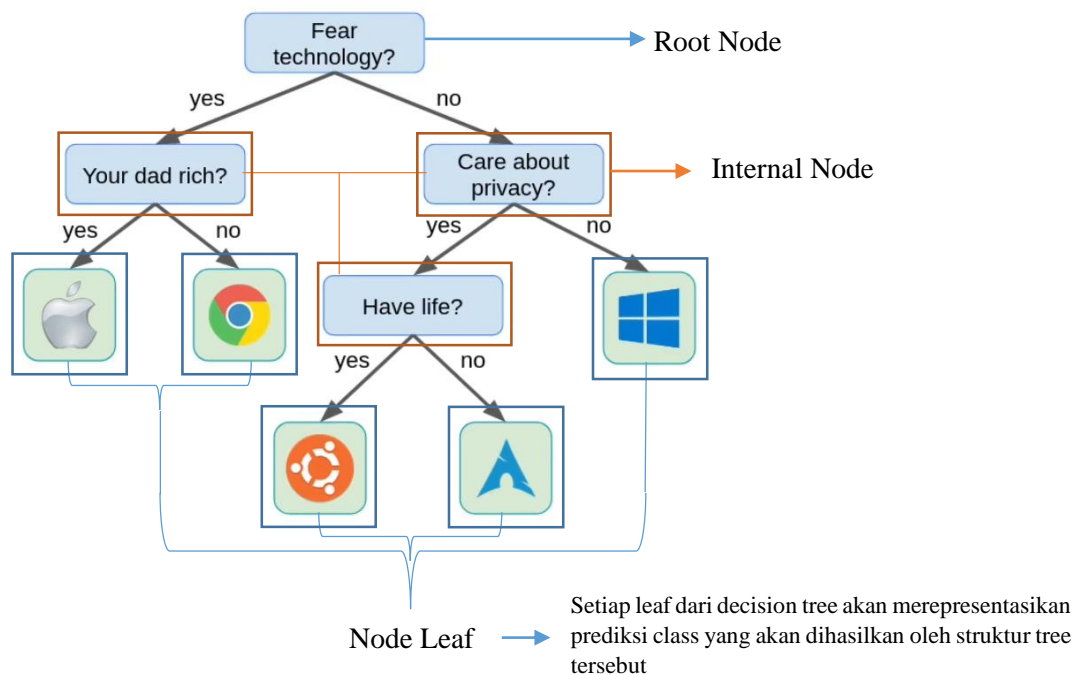


Nama : Suci Maria  
Kelas : TIF-A1  
Npm : 41155050210005

### TUGAS PERTEMUAN 3

1. Lakukan praktik dari [https://youtu.be/5wwXKtLkyqs?si=fn88eveu\\_qbCC6b3](https://youtu.be/5wwXKtLkyqs?si=fn88eveu_qbCC6b3) , buat screenshot dengan nama kalian pada coding, kumpulkan dalam bentuk pdf, dari kegiatan ini:

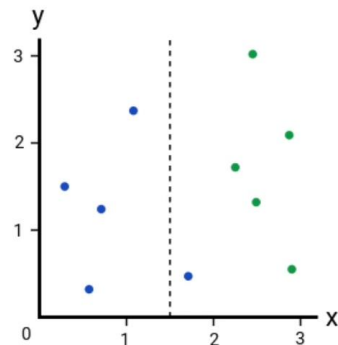
#### 1.1. Pengenalan komponen Decision Tree:root, node, leaf



Root, node, leaf merupakan 3 komponen dasar dalam membentuk suatu decision tree. Dalam machine learning terdapat beberapa Decision Tree Algorithym (Algoritma Decision tree), diantaranya yaitu :

- **CART (Clasification And Regression Tree)**  
CART ini merupakan default alorythm yang diterapkan oleh Scikit-Learn sebagai implementasi decision Tree
- **ID3**
- **C4.5**
- **C5.0**

## 1.2. Pengenalan Gini Impurity (Pengukuran Ke Tidak murnian)



Gini Impurity memiliki jangkauan nilai antara 0 dan 1.

- 0 = Mengindikasikan nilai murni yang sempurna
- 1 = Mengindikasikan nilai yang impure atau paling tidak murni

Contoh kasus diatas terdapat 10 data point, dimana data points ini akan terbagi dalam dua class, ada class biru dan ada class hijau. Selanjutnya data dipisahkan kedalam dua bagian sehingga dihasilkan 2 buah ruas yaitu ruas kiri dan ruas kanan. Garis putus-putus diatas berperan sebagai splitter (pemisah). Lalu dilakukannya pengukuran impurity pada kedua ruas, melakukan pengukuran pada kedua ruas dengan formula sebagai berikut :

### Ruas Kiri

$$\begin{aligned} G &= 1 - \sum_i^n P_i^2 \\ &= 1 - P(\text{biru})^2 \\ &= 1 - \left(\frac{4}{4}\right)^2 = 0 \end{aligned}$$

### Penjelasan :

Nilai Gini impurity dapat diperoleh dari 1 diselisihkan dengan total hasil penjumlahan probabilitas kuadrat dari tiap class yang tersedia. Dari kasus diatas pada ruas sebelah kiri hanya menampung data class biru saja yang mana nilai probabilitas yang dihitung hanya nilai probabilitas class biru saja. Maka proses kalkulasi nya adalah

$$1 - P(\text{biru})^2$$

1 diselisihkan dengan dengan nilai probabilitas dari class biru lalu dikuadratkan

Untuk nilai probabilitas dari class biru dapat dicari dengan cara

$$1 - \left(\frac{4}{4}\right)^2 = 0$$

4 diatas adalah jumlah dari class berwarna biru, sedangkan 4 yang di bawah merupakan total seluruh data yang ada di sebelah kiri.

Nilai Gini Impurity dari ruas kiri adalah 0 yang mengindikasikan kemurnian sempurna

### Ruas Kanan

$$\begin{aligned} G &= 1 - \sum_i^n P_i^2 \\ &= 1 - (P(biru)^2 + P(hijau)^2) \\ &= 1 - ((\frac{1}{6})^2 + (\frac{5}{6})^2) = 0.278 \end{aligned}$$

### Penjelasan :

Pada ruas kanan terdapat dua class ada class biru dan class hijau, maka proses kalkulasinya adalah 1 diselisihkan dengan penjumlahan antara nilai probability biru kuadrat dengan probability hijau kuadrat.

$$1 - (P(biru)^2 + P(hijau)^2)$$

1 diselisihkan dengan dengan nilai probability dari class biru kuadrat ditambah dengan nilai probability class hijau kuadrat

Untuk mengetahui nilai probability dari kedua class dapat dicari dengan cara menghitung jumlah dari masing masing class yang ada pada ruas kanan

$$1 - ((\frac{1}{6})^2 + (\frac{5}{6})^2) = 0.278$$

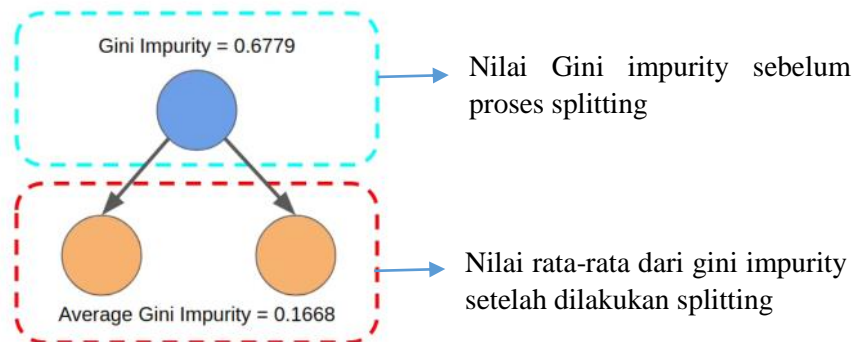
1 diatas adalah jumlah dari class berwarna biru, sedangkan 6 yang di bawah merupakan total seluruh data yang ada di ruas sebelah kanan. Begitupun pada 5 diatas merupakan jumlah class hijau dan 6 total dari seluruh data pada ruas kanan

Nilai Gini Impurity dari ruas kanan adalah 0.278

### Avarage Gini Impurity

$$\begin{aligned} G &= \frac{4}{4+6} \times 0 + \frac{6}{4+6} \times 0.278 \\ &= 0.1668 \end{aligned}$$

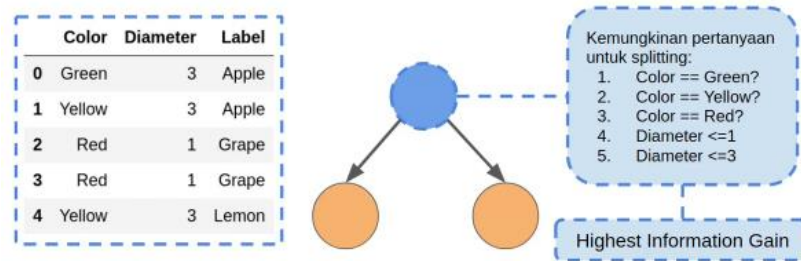
## 1.3. Pengenalan Information Gain



$$\text{Information Gain} = 0.6779 - 0.1668 = 0.51$$

Setelah mengetahui kedua nilai diatas selanjutnya dapat mengkalkulasikan nilai information Gain. Information Gain dapat di peroleh dengan menselisihkan nilai Gini impurity sebelum proses splitting dengan nilai rata-rata gini impurity setelah proses splitting.

## 1.4. Membangun Decision Tree



$$\begin{aligned} G &= 1 - (P(\text{apple})^2 + P(\text{grape})^2 + P(\text{lemon})^2) \\ &= 1 - \left(\left(\frac{2}{5}\right)^2 + \left(\frac{2}{5}\right)^2 + \left(\frac{1}{5}\right)^2\right) \\ &= 0.63 \end{aligned}$$

## 1.5. Persiapan dataset: Iris Dataset

```
[2]: from sklearn.datasets import load_iris
X, y = load_iris(return_X_y=True)
print(f'Dimensi Feature: {X.shape}')
print(f'Class: {set(y)}')
```

Dimensi Feature: (150, 4)  
Class: {np.int64(0), np.int64(1), np.int64(2)}

```
[3]: print("Nama: Suci Maria")
Nama: Suci Maria
```

```
[6]: 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=0)
```

```
[7]: print("Nama: Suci Maria")
Nama: Suci Maria
```

## 1.6. Training model Decision Tree Classifier

```
[8]: from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(max_depth=4)
model.fit(X_train, y_train)
```

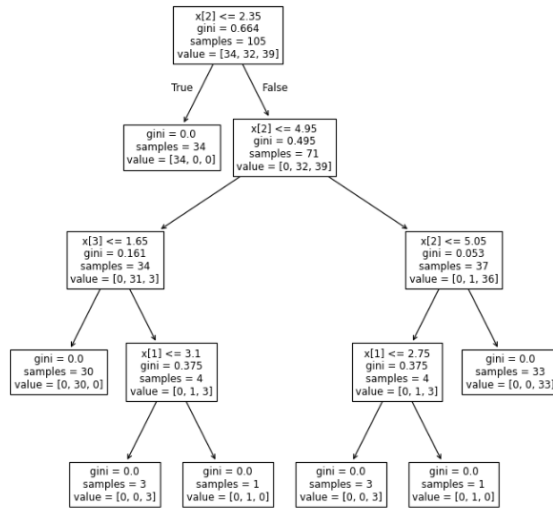
```
[8]: DecisionTreeClassifier
DecisionTreeClassifier(max_depth=4)
```

```
[9]: print("Nama: Suci Maria")
Nama: Suci Maria
```

## 1.7. Visualisasi model Decision Tree

```
[10]: import matplotlib.pyplot as plt
from sklearn import tree

plt.rcParams['figure.dpi'] = 85
plt.subplots(figsize=(10, 10))
tree.plot_tree(model, fontsize=10)
plt.show()
```



```
[11]: print("Nama: Suci Maria")
Nama: Suci Maria
```

## 1.8. Evaluasi model Decision Tree

```
[12]: from sklearn.metrics import classification_report

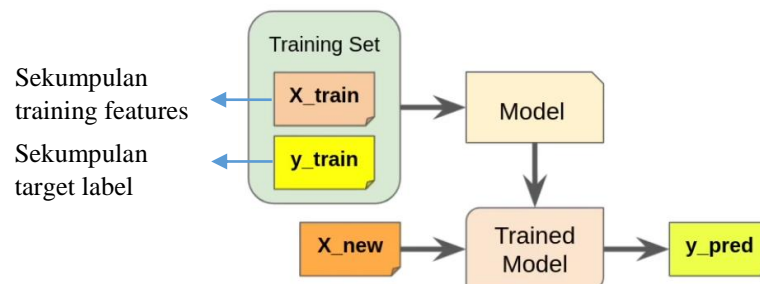
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	16
1	1.00	0.94	0.97	18
2	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45
weighted avg	0.98	0.98	0.98	45

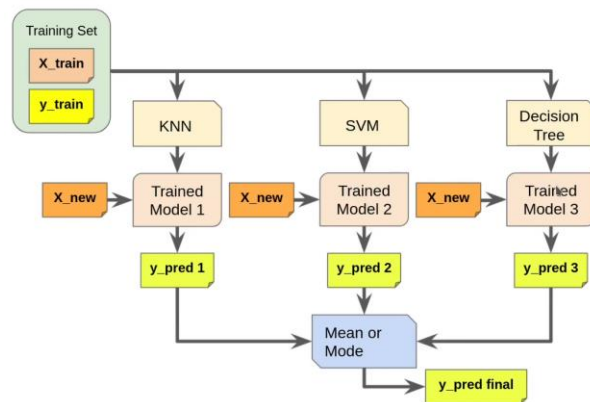
```
[13]: print("Nama: Suci Maria")
Nama: Suci Maria
```

2. Lakukan praktik dari [https://youtu.be/yKovaQ6tyV8?si=HnHG6kcoCsDwvo\\_0](https://youtu.be/yKovaQ6tyV8?si=HnHG6kcoCsDwvo_0) ,  
buat screenshot dengan nama kalian pada coding, kumpulkan dalam bentuk pdf,  
dari kegiatan ini:

### 2.1. Proses training model Machine Learning secara umum

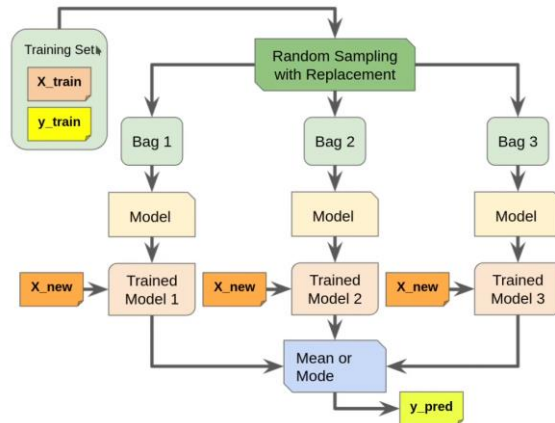


## 2.2. Pengenalan Ensemble Learning



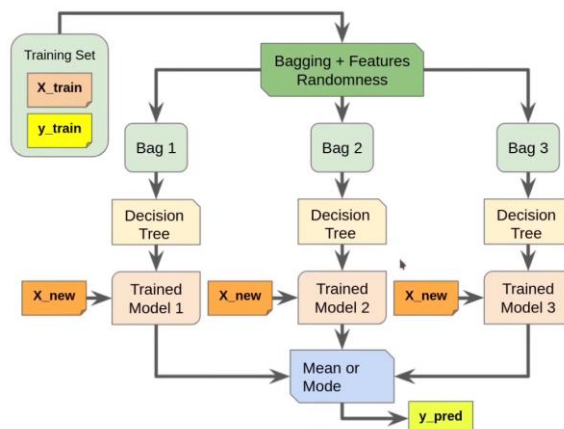
## 2.3. Pengenalan Bootstrap Aggregating | Bagging

Memadukan model Machine Learning yang sejenis.



Dari 3 buah trained model pada gambar di atas akan menghasilkan prediksi yang berbeda-beda dari tiap trained modelnya. Maka dari itu dari ketiga prediksi tersebut perlu disatukan agar menghasilkan satu prediksi final.

## 2.4. Pengenalan Random Forest | Hutan Acak



## 2.5. Persiapan dataset | Iris Flower Dataset

```
[15]: from sklearn.datasets import load_iris
      X, y = load_iris(return_X_y=True)
      print(f'Dimensi Feature: {X.shape}')
      print(f'Class: {set(y)}')
      Dimensi Feature: (150, 4)
      Class: {np.int64(0), np.int64(1), np.int64(2)}

[16]: print("Nama: Suci Maria")
      Nama: Suci Maria

[17]: 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=0)

[18]: print("Nama: Suci Maria")
      Nama: Suci Maria
```

## 2.6. Implementasi Random Forest Classifier dengan Scikit Learn

```
[19]: from sklearn.ensemble import RandomForestClassifier
      model = RandomForestClassifier(n_estimators=100,
                                    random_state=0)
      model.fit(X_train, y_train)

[19]: + RandomForestClassifier
      RandomForestClassifier(random_state=0)

[20]: print("Nama: Suci Maria")
      Nama: Suci Maria
```

## 2.7. Evaluasi model dengan Classification Report

```
[21]: from sklearn.metrics import classification_report
      y_pred = model.predict(X_test)
      print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

      0       1.00        1.00        1.00        16
      1       1.00        0.94        0.97        18
      2       0.92        1.00        0.96        11

   accuracy          0.97          0.98          0.98         45
  macro avg          0.97          0.98          0.98         45
 weighted avg          0.98          0.98          0.98         45

[22]: print("Nama: Suci Maria")
      Nama: Suci Maria
```