**Class Abstract dan decorator**

Pada kali ini kita akan mengubungkan abstract method dengan decorator yang sudah kita pelajari sebelumnya. Misalnya static method, class method, property, getter dan setter. Lihat kode berikut :

from abc import ABC, abstractmethod

class Button(ABC) :

    @abstractmethod

    def click(self) :

        pass

class PushButton(Button) :

    def click(self) :

        print('push button click')

tombol1 = PushButton()

tombol1.click()

pada kode diatas saya ingin membuat link, linknya adalah kita bisa mencampur abstract method dan kita juga bisa tidak mencampur abstract method. Berikut perubahan kode yang terjadi :

from abc import ABC, abstractmethod

class Button(ABC):

    def \_\_init\_\_(self, set\_link):

*self*.link = set\_link

    @abstractmethod

    def click(self):

        pass

class PushButton(Button):

    def click(self):

        print(f"Go to: {*self*.link}")

tombol1 = PushButton("www.muhamadamin.id")

yang menjadi masalah adalah jika kita tidak mengetahui abstract class nya itu seperti apa, maka kita tidak akan tahu dari mana **self.link** itu berasal dari mana, sehingga membuat kita menjadi tidak mengerti kodenya. Berikut cara memaksa implementasi dari **self.link** itu ada di subclass bukan di super class :

kita bisa menggunakan getter setter. Berikut kode yang bisa anda masukkan kedalam class Button :

    @property

    @abstractmethod

    def link(self):

        pass

ingat urutan penulisannya. Sekarang kita bisa mengimplemetasikan link di subclassnya. Berikut kode yang saya masukkan kedalam class **PushButton** :

    @Button.link.setter

    def link(self, input):

*self*.\_\_link = input

    @link.getter

    def link(self):

        return *self*.\_\_link

jadi kita bisa mengetahui kalau si **self.link** itu adalah kode diatas.

Sampai sini, kita sudah belajar OOP dasar pada python. Setelah belajar pemrograman dan OOP python, biasanya akan lanjut belajar cara menggunakannya. Jika ingin tahu cara menggunakannya kita harus belajar Object Oriented Design.