LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023

Prepared By:

ALI MABRUR MUBAROK

210511112 / R3

Nama : Ali Mabrur Mubarok

Nim : 210511112

Kelas : R3 / TI21C

Tugas-7 PBO2 2023

Bmimeta.py

```
#Nama : ALI MABRUR MUBAROK
#Nim : 210511112
#Kelas : R3 / TI21C

class BodyMassIndexMeta(type):
    def __init__(cls, name, bases, attrs):
        super().__init__(name, bases, attrs)
        cls.tb_standar = ""

    def to_pria(cls, tb):
        return (tb - 100) - ((tb - 100) * (10/100))

    def to_wanita(cls, tb):
        return (tb - 100) - ((tb - 100) * (15/100))

class Bmi(metaclass=BodyMassIndexMeta):
    def __init__(self, tb, bb):
        self.tb = tb
        self.bb = bb

    def ke_unit(self, unit):
        if unit == "Pria":
            self.tb = self.__class__.to_pria(self.tb)
            self.__class__.tb_standar = "(Kg) Pria"
```

```
        elif unit == "Wanita":
            self.tb = self.__class__.to_wanita(self.tb)
            self.__class__.tb_standar = "(Kg) Wanita"
        elif unit == "Bmi":
            pass # do nothing
        else:
            raise ValueError(f"Unit {unit} tidak dikenal.")

    def mutu(self):

        bmikalkulator = (self.bb / (self.tb/100*2))
        if bmikalkulator < 18.5:
            return bmikalkulator, "KURUS"
        elif bmikalkulator >= 18.5:
            return bmikalkulator, "NORMAL"
        elif bmikalkulator >= 24.9:
            return bmikalkulator, "GEMUK"
        else:
            return bmikalkulator, "OBESITAS LALALA"

    def __repr__(self):
        return f"{self.tb:.2f} {self.__class__.tb_standar}"

# Membuat objek tb dengan nilai 100 Bmi
c = Bmi(160, 65)
b = c.mutu()
# Mengubah objek tb menjadi Fahrenheit
c.ke_unit("Pria")
print("Berat Ideal Anda :",c)
```

print("Hasil BMI Anda Adalah:",b)