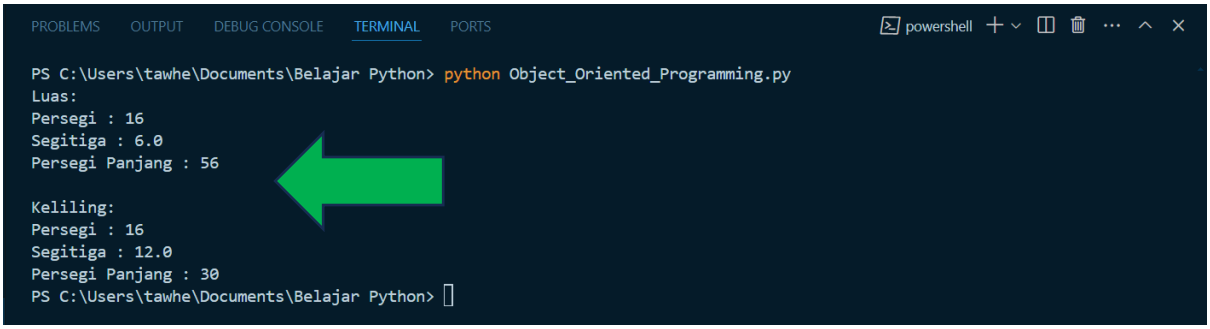
Object Oriented Programming

1. Easy Challenge (Menghitung Luas dan Keliling)

```
class Persegi:
    def __init__(self, sisi):
        self.sisi = sisi

    def luas(self):
        return self.sisi * self.sisi

    def keliling(self):
        return 4 * self.sisi


class Segitiga:
    def __init__(self, alas, tinggi):
        self.alas = alas
        self.tinggi = tinggi

    def luas(self):
        return 0.5 * self.alas * self.tinggi

    def keliling(self):
        sisi_miring = (self.alas**2 + self.tinggi**2) ** 0.5
        return self.alas + self.tinggi + sisi_miring


class PersegiPanjang:
    def __init__(self, panjang, lebar):
        self.panjang = panjang
        self.lebar = lebar

    def luas(self):
        return self.panjang * self.lebar

    def keliling(self):
        return 2 * (self.panjang + self.lebar)


def main():
    print("Luas:")
    persegi = Persegi(4)
    print(f"Persegi : {persegi.luas()}")

    segitiga = Segitiga(3, 4)
    print(f"Segitiga : {segitiga.luas()}")
```

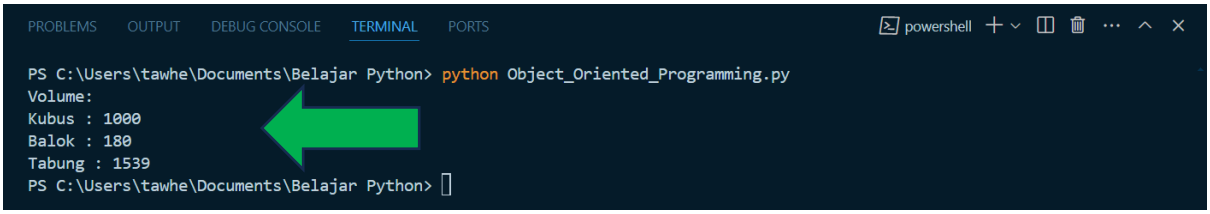
```
    persegi_panjang = PersegiPanjang(7, 8)
    print(f"Persegi Panjang : {persegi_panjang.luas()}")

    print("\nKeliling:")
    print(f"Persegi : {persegi.keliling()}")
    print(f"Segitiga : {segitiga.keliling()}")
    print(f"Persegi Panjang : {persegi_panjang.keliling()}")


if __name__ == "__main__":
    main()
```



2. Easy Challenge (Menghitung Volume)

```
import math


class Kubus:
    def __init__(self, sisi):
        self.sisi = sisi

    def volume(self):
        return self.sisi ** 3


class Balok:
    def __init__(self, panjang, lebar, tinggi):
        self.panjang = panjang
        self.lebar = lebar
        self.tinggi = tinggi

    def volume(self):
        return self.panjang * self.lebar * self.tinggi


class Tabung:
    def __init__(self, jari_jari, tinggi):
        self.jari_jari = jari_jari
```

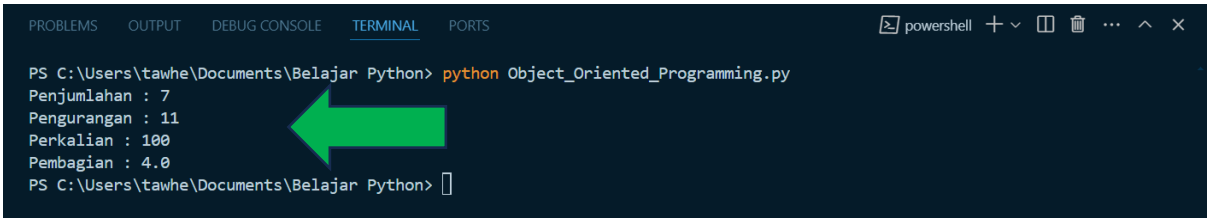
```
        self.tinggi = tinggi

    def volume(self):
        return math.pi * (self.jari_jari ** 2) * self.tinggi


def main():
    print("Volume:")
    kubus = Kubus(10)
    print(f"Kubus : {kubus.volume()}")

    balok = Balok(3, 6, 10)
    print(f"Balok : {balok.volume()}")

    tabung = Tabung(7, 10)
    print(f"Tabung : {round(tabung.volume())}")

if __name__ == "__main__":
    main()
```



3. Medium Challenge (Kalkulator)

```
class Kalkulator:
    def penjumlahan(self, a, b):
        return a + b

    def pengurangan(self, a, b):
        return a - b

    def perkalian(self, a, b):
        return a * b

    def pembagian(self, a, b):
        if b == 0:
            return "Error: Pembagian dengan nol tidak diperbolehkan."
        return a / b


def main():
    kalkulator = Kalkulator()
```

```
    hasil_penjumlahan = kalkulator.penjumlahan(3, 4)
    print(f"Penjumlahan : {hasil_penjumlahan}")

    hasil_pengurangan = kalkulator.pengurangan(15, 4)
    print(f"Pengurangan : {hasil_pengurangan}")

    hasil_perkalian = kalkulator.perkalian(10, 10)
    print(f"Perkalian : {hasil_perkalian}")

    hasil_pembagian = kalkulator.pembagian(12, 3)
    print(f"Pembagian : {hasil_pembagian}")


if __name__ == "__main__":
    main()
```



4. Medium Challenge (Ongkos Kirim)

```
import math

class Barang:
    def __init__(self, panjang, lebar, tinggi, berat):
        self.panjang = panjang
        self.lebar = lebar
        self.tinggi = tinggi
        self.berat = berat

    def hitung_volume(self):
        return self.panjang * self.lebar * self.tinggi

    def hitung_berat_bulat(self):
        return math.ceil(self.berat)

    def hitung_harga_pengiriman(self):
        harga_standar = 5000
        volume = self.hitung_volume()
        berat_bulat = self.hitung_berat_bulat()

        if volume >= 50 and berat_bulat >= 1:
            return harga_standar
        else:
            return 0
```

```
panjang = 5
lebar = 2
tinggi = 4
berat = 1  # dalam kg

barang = Barang(panjang, lebar, tinggi, berat)

harga = barang.hitung_harga_pengiriman()

print(f"Output harga adalah Rp. {5000}")
```