**TUGAS RPLBK – Running code**

Irfan Maulana Manaf

1. **LSP Violation**

Kode di bawh menunjukkan pelanggaran terhadap **Liskov Substitution Principle (LSP)**, salah satu prinsip dari SOLID. LSP menyatakan bahwa objek dari subclass harus bisa menggantikan objek dari superclass tanpa mengubah kebenaran program.

class PersegiPanjang:

    def \_\_init\_\_(self, lebar, tinggi):

        self.\_lebar = lebar

        self.\_tinggi = tinggi

    def set\_lebar(self, lebar):

        self.\_lebar = lebar

    def set\_tinggi(self, tinggi):

        self.\_tinggi = tinggi

    def get\_luas(self):

        return self.\_lebar \* self.\_tinggi

class Persegi(PersegiPanjang):

    def set\_lebar(self, lebar):

        self.\_lebar = lebar

        self.\_tinggi = lebar

    def set\_tinggi(self, tinggi):

        self.\_lebar = tinggi

        self.\_tinggi = tinggi

def cetak\_luas(bentuk: PersegiPanjang):

    bentuk.set\_lebar(5)

    bentuk.set\_tinggi(10)

    print(f"Luas: {bentuk.get\_luas()}")

# Contoh penggunaan

persegi = Persegi(4, 4)

cetak\_luas(persegi)  # Ini akan mencetak 100, yang salah

Penjelasan :

1. **Kelas PersegiPanjang:**

Kelas ini memiliki dua atribut, yaitu \_lebar dan \_tinggi, dan metode get\_luas() yang mengembalikan luas persegi panjang. Terdapat juga metode set\_lebar dan set\_tinggi untuk mengatur nilai lebar dan tinggi.

1. **Kelas Persegi:**

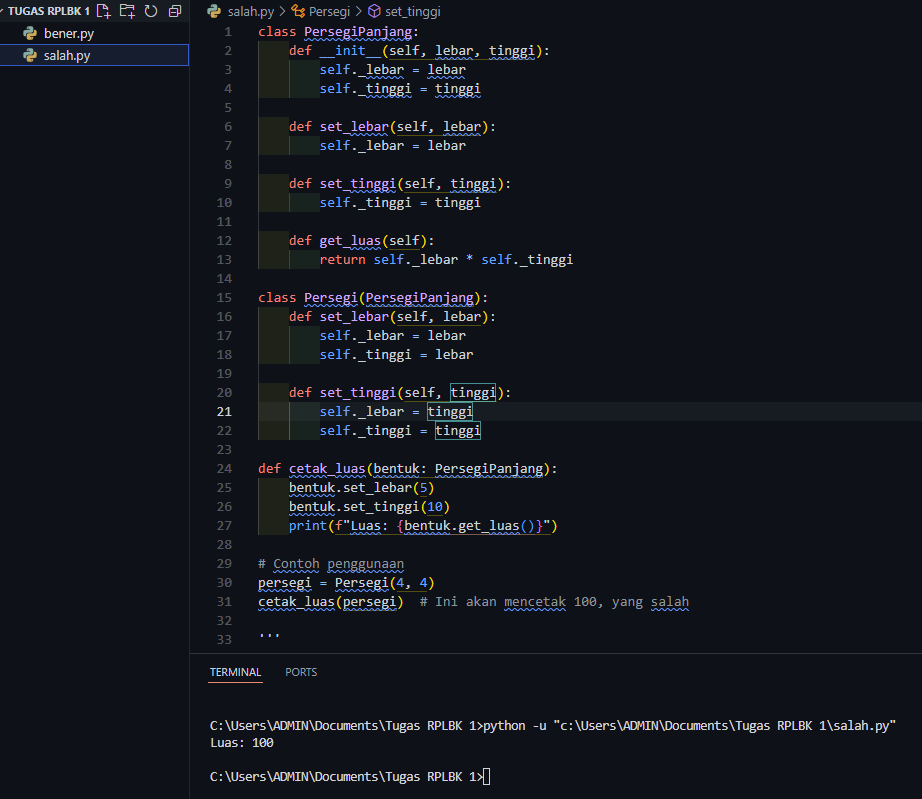
Kelas Persegi adalah subclass dari PersegiPanjang. Namun, karena persegi adalah bentuk khusus dari persegi panjang di mana lebar dan tinggi harus selalu sama, kelas Persegi mengubah perilaku metode set\_lebar dan set\_tinggi. Pada Persegi, ketika salah satu dari lebar atau tinggi diubah, keduanya diperbarui dengan nilai yang sama.

1. **Fungsi cetak\_luas:**

Fungsi ini mengambil objek bertipe PersegiPanjang sebagai parameter, dan kemudian mengubah lebar menjadi 5 dan tinggi menjadi 10. Dalam kasus objek PersegiPanjang, hasilnya akan sesuai dengan yang diharapkan yaitu luas 50 (5 \* 10).

1. **Pelanggaran LSP:**

Ketika Persegi digunakan dalam fungsi cetak\_luas, hasil yang diharapkan (luas 50) tidak diperoleh. Sebaliknya, luas yang dicetak adalah 100 karena kelas Persegi memastikan bahwa lebar dan tinggi selalu sama, yang tidak sesuai dengan perilaku PersegiPanjang. Hal ini melanggar LSP karena kelas Persegi tidak dapat digunakan sebagai pengganti kelas PersegiPanjang tanpa mengubah hasil yang diharapkan.



Gambar 1. 1 Bukti Running Code Salah

1. LSP Violation ( Sudah dibenarkan )

Berikut adalah source code dan bukti screenshot untuk code yang sudah diperbaiki sehingga mematuhi LSP.

class Bentuk:

    def get\_luas(self):

        pass

class PersegiPanjang(Bentuk):

    def \_\_init\_\_(self, lebar, tinggi):

        self.\_lebar = lebar

        self.\_tinggi = tinggi

    def set\_lebar(self, lebar):

        self.\_lebar = lebar

    def set\_tinggi(self, tinggi):

        self.\_tinggi = tinggi

    def get\_luas(self):

        return self.\_lebar \* self.\_tinggi

class Persegi(Bentuk):

    def \_\_init\_\_(self, sisi):

        self.\_sisi = sisi

    def set\_sisi(self, sisi):

        self.\_sisi = sisi

    def get\_luas(self):

        return self.\_sisi \* self.\_sisi

def cetak\_luas(bentuk: Bentuk):

    print(f"Luas: {bentuk.get\_luas()}")

# Contoh penggunaan

persegi\_panjang = PersegiPanjang(5, 10)

persegi = Persegi(4)

cetak\_luas(persegi\_panjang)  # Mencetak 50 dengan benar

cetak\_luas(persegi)          # Mencetak 16 dengan benar

Penjelasan:

1. **Pemisahan Kelas**

Pada kode sebelumnya, kelas Persegi mewarisi dari kelas PersegiPanjang, yang menyebabkan perilaku yang tidak konsisten. Pada kode baru, Persegi dan PersegiPanjang tidak lagi memiliki hubungan pewarisan secara langsung. Keduanya sekarang menjadi subclass dari kelas abstrak Bentuk.

1. **Penggunaan Kelas Abstrak Bentuk**

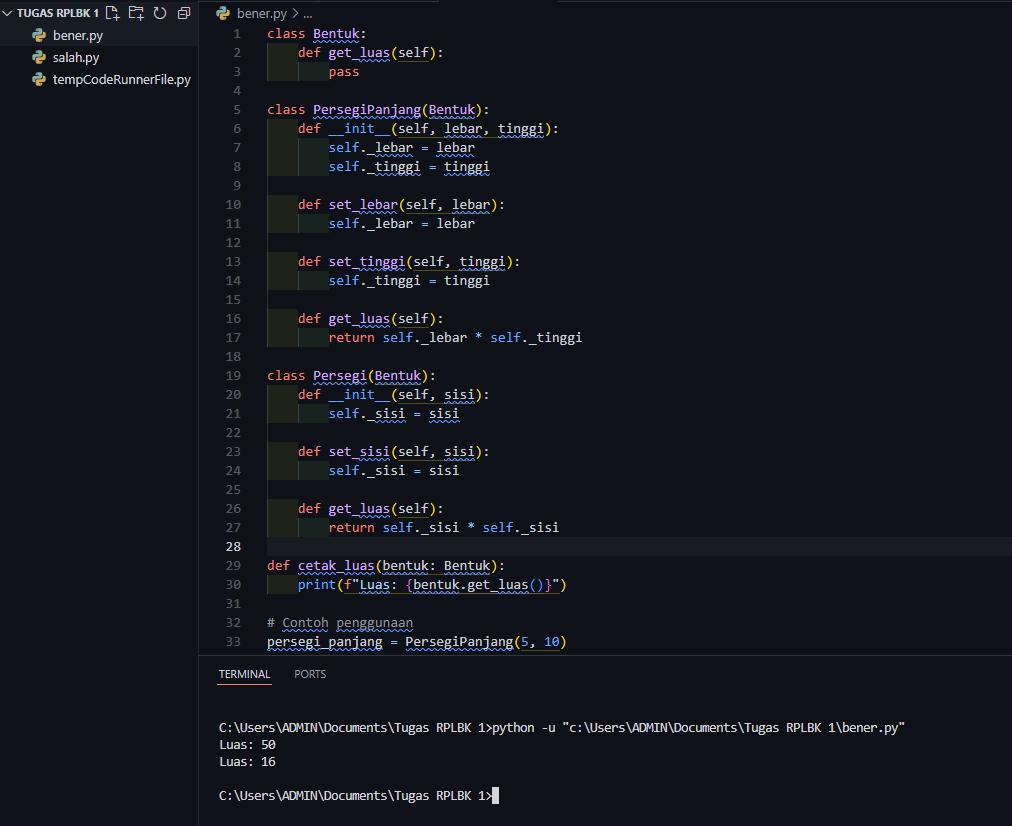
Bentuk digunakan sebagai kelas dasar yang hanya mendefinisikan metode get\_luas(). Maka, kedua kelas (PersegiPanjang dan Persegi) menjadi subclass dari Bentuk dan masing-masing mengimplementasikan metode get\_luas() sesuai dengan karakteristiknya.

1. **Implementasi Metode get\_luas**

Pada kode sebelumnya, Persegi mengoverride metode set\_lebar dan set\_tinggi dari PersegiPanjang, yang menyebabkan tidak konsistensi. Di kode yang baru, Persegi dan PersegiPanjang memiliki implementasi get\_luas() masing-masing tanpa mempengaruhi atau tergantung pada yang lain. Persegi menghitung luas berdasarkan atribut \_sisi, sementara PersegiPanjang menghitung luas berdasarkan atribut \_lebar dan \_tinggi.

1. **Fungsi cetak\_luas**

Sehingga fungsi cetak\_luas sekarang bisa digunakan untuk objek dari kelas Persegi dan PersegiPanjang tanpa perlu khawatirk perubahan perilaku yang tidak diinginkan. Hal ini menunjukkan bahwa kode tersebut mematuhi LSP, karena kedua subclass (Persegi dan PersegiPanjang) bisa menggantikan superclass (Bentuk) tanpa mengubah hasil yang diharapkan.



Gambar 2. 1 Bukti Running Code Benar