

TUGAS RPLBK – Running code
Irfan Maulana Manaf

1. LSP Violation

Kode di bawah 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.

2. Kelas Persegi:

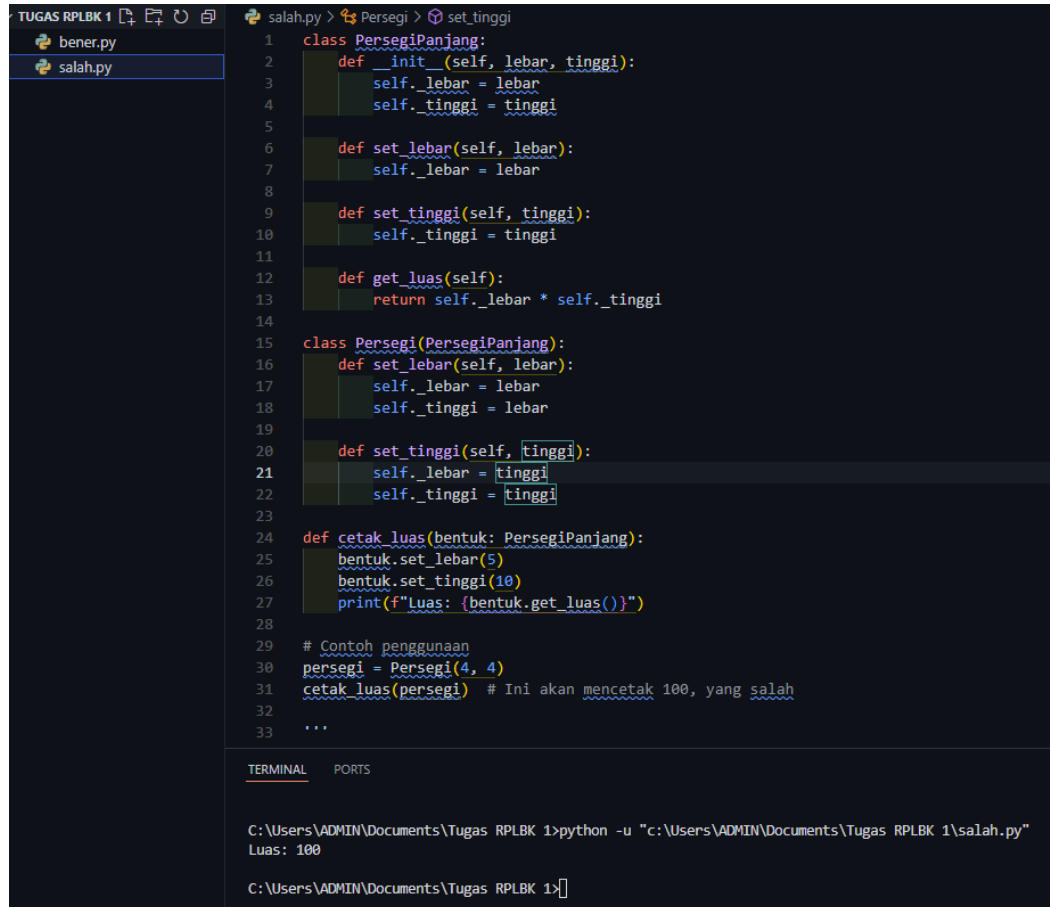
Kelas `Persegi` adalah subclass dari `PersegiPanjang`. Namun, karena persegi adalah bentuk khusus dari persegi panjang di mana lebar dan tinggi harus selalu sama, kelas `Persegi` mengubah perilaku metode `set_lebar` dan `set_tinggi`. Pada `Persegi`, ketika salah satu dari lebar atau tinggi diubah, keduanya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

3. 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 ($5 * 10$).

4. 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.



```
1 class PersegiPanjang:
2     def __init__(self, lebar, tinggi):
3         self.lebar = lebar
4         self.tinggi = tinggi
5
6     def set_lebar(self, lebar):
7         self.lebar = lebar
8
9     def set_tinggi(self, tinggi):
10         self.tinggi = tinggi
11
12     def get_luas(self):
13         return self.lebar * self.tinggi
14
15 class Persegi(PersegiPanjang):
16     def set_lebar(self, lebar):
17         self.lebar = lebar
18         self.tinggi = lebar
19
20     def set_tinggi(self, tinggi):
21         self.lebar = tinggi
22         self.tinggi = tinggi
23
24 def cetak_luas(bentuk: PersegiPanjang):
25     bentuk.set_lebar(5)
26     bentuk.set_tinggi(10)
27     print(f"Luas: {bentuk.get_luas()}")
28
29 # Contoh penggunaan
30 persegi = Persegi(4, 4)
31 cetak_luas(persegi) # Ini akan mencetak 100, yang salah
32
33 ...
```

TERMINAL PORTS

```
C:\Users\ADMIN\Documents\Tugas RPLBK 1>python -u "c:\Users\ADMIN\Documents\Tugas RPLBK 1\salah.py"
Luas: 100
C:\Users\ADMIN\Documents\Tugas RPLBK 1>
```

Gambar 1. 1 Bukti Running Code Salah

2. 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`.

2. Penggunaan Kelas Abstrak `Bentuk`

`Bentuk` digunakan sebagai kelas dasar yang hanya mendefinisikan metode `get_luas()`. Maka, kedua kelas (`PersegiPanjang` dan `Persegi`) menjadi subclass

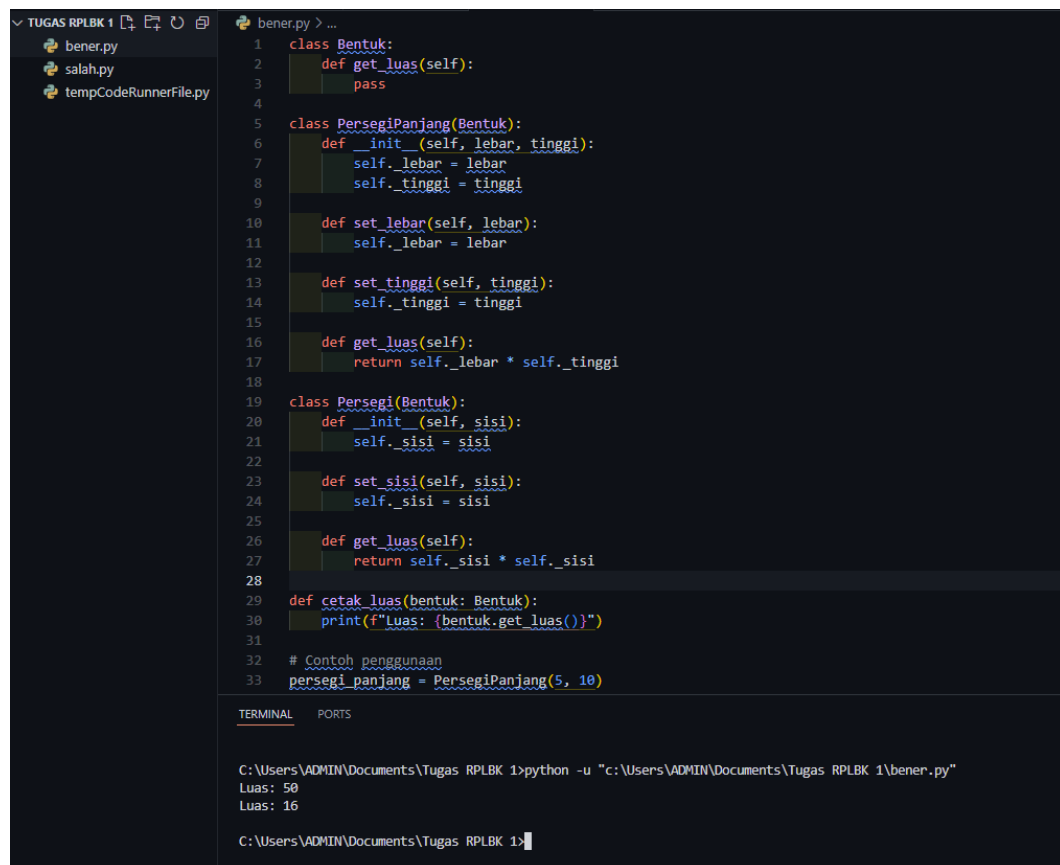
dari Bentuk dan masing-masing mengimplementasikan metode `get_luas()` sesuai dengan karakteristiknya.

3. 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`.

4. Fungsi `cetak_luas`

Sehingga fungsi `cetak_luas` sekarang bisa digunakan untuk objek dari kelas Persegi dan PersegiPanjang tanpa perlu khawatir 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.



```
1 class Bentuk:
2     def get_luas(self):
3         pass
4
5 class PersegiPanjang(Bentuk):
6     def __init__(self, lebar, tinggi):
7         self._lebar = lebar
8         self._tinggi = tinggi
9
10    def set_lebar(self, lebar):
11        self._lebar = lebar
12
13    def set_tinggi(self, tinggi):
14        self._tinggi = tinggi
15
16    def get_luas(self):
17        return self._lebar * self._tinggi
18
19 class Persegi(Bentuk):
20     def __init__(self, sisi):
21         self._sisi = sisi
22
23     def set_sisi(self, sisi):
24         self._sisi = sisi
25
26     def get_luas(self):
27         return self._sisi * self._sisi
28
29 def cetak_luas(bentuk: Bentuk):
30     print(f"Luas: {bentuk.get_luas()}")
31
32 # Contoh penggunaan
33 persegi_panjang = PersegiPanjang(5, 10)
```

TERMINAL PORTS

```
C:\Users\ADMIN\Documents\Tugas RPLBK 1>python -u "c:\Users\ADMIN\Documents\Tugas RPLBK 1\benar.py"
Luas: 50
Luas: 16
C:\Users\ADMIN\Documents\Tugas RPLBK 1>
```

Gambar 2. 1 Bukti Running Code Benar