

Nama : Bagas Farrelino Harsono Putro

NIM : 21120122140101

Kelas : Rekayasa Perangkat Lunak Berbasis Komponen C

Group : C

TUGAS RPLBK- Running Code Example

Liskov Substitution Principle (LSP)

Problem

Misalkan kita memiliki sebuah kelas `Pengguna` yang menangani berbagai macam jenis pengguna di sistem aplikasi. Kelas ini memiliki sebuah metode untuk menghitung diskon yang bisa diberikan kepada pengguna berdasarkan statusnya, misalnya pengguna reguler, premium, atau VIP. Namun, ketika kita mencoba menggunakan subclass yang menggantikan superclass, ada potensi masalah karena subclass tidak menghormati kontrak yang ditetapkan oleh superclass.

Implementasi yang melanggar LSP

```
# Bagas Farrelino Harsono Putro 21120122140101
class Pengguna:
    def __init__(self, nama):
        self.nama = nama

    def hitung_diskon(self):
        return 0 # Default pengguna tidak dapat diskon

class PenggunaPremium(Pengguna):
    def hitung_diskon(self):
        return 20 # Diskon 20% untuk pengguna premium

class PenggunaVIP(Pengguna):
    def hitung_diskon(self):
        raise NotImplementedError("Pengguna VIP harus menggunakan metode lain
untuk diskon.") # Tidak menghormati kontrak
```

```

def cetak_diskon(pengguna):
    print(f"Pengguna {pengguna.nama} mendapatkan diskon sebesar {pengguna.hitung_diskon()}%.")

# Pengguna Regular
pengguna1 = Pengguna("Bagas")
cetak_diskon(pengguna1)

# Pengguna Premium
pengguna2 = PenggunaPremium("Farrelino")
cetak_diskon(pengguna2)

# Pengguna VIP, akan menyebabkan error
pengguna3 = PenggunaVIP("Harsono Putro")
cetak_diskon(pengguna3) # Ini akan menyebabkan NotImplementedError

```

Kelas PenggunaVIP tidak memenuhi kontrak dari kelas Pengguna. Kelas induk mengharapkan semua pengguna memiliki diskon yang bisa dihitung, tetapi PenggunaVIP tidak mengimplementasikan metode `hitung_diskon` dengan benar. Akibatnya, ketika kita mencoba memanggil metode tersebut, muncul error `NotImplementedError`.

```

Liskov Benar.py  Liskov Salah.py X
C:\Users\MSI GF63\Documents\UNDIP\Semester 5\RPLBPK> Liskov Salah.py > Pengguna > _init_
1 # Bagas Farrelino Harsono Putro 21120122140101
2 class Pengguna:
3     def __init__(self, nama):
4         self.nama = nama
5
6     def hitung_diskon(self):
7         return 0 # Default pengguna tidak dapat diskon
8
9 class PenggunaPremium(Pengguna):
10     def hitung_diskon(self):
11         return 20 # Diskon 20% untuk pengguna premium
12
13 class PenggunaVIP(Pengguna):
14     def hitung_diskon(self):
15         raise NotImplementedError("Pengguna VIP harus menggunakan metode lain untuk diskon.") # Tidak menghormati kontrak
16
17
18 def cetak_diskon(pengguna):
19     print(f"Pengguna {pengguna.nama} mendapatkan diskon sebesar {pengguna.hitung_diskon()}%.")
20
21
22 # Pengguna Regular
23 pengguna1 = Pengguna("Bagas")
24 cetak_diskon(pengguna1)
25
26 # Pengguna Premium
27 pengguna2 = PenggunaPremium("Farrelino")
28 cetak_diskon(pengguna2)
29
30 # Pengguna VIP, akan menyebabkan error
31 pengguna3 = PenggunaVIP("Harsono Putro")
32 cetak_diskon(pengguna3) # Ini akan menyebabkan NotImplementedError

```

```

File "c:\users\msi gf63\documents\undip\semester 5\rplbpk\liskov salah.py", line 19, in cetak_diskon
    print(f"Pengguna {pengguna.nama} mendapatkan diskon sebesar {pengguna.hitung_diskon()}%.")
File "c:\users\msi gf63\documents\undip\semester 5\rplbpk\liskov salah.py", line 15, in hitung_diskon
    raise NotImplementedError("Pengguna VIP harus menggunakan metode lain untuk diskon.") # Tidak menghormati kontrak
NotImplementedError: Pengguna VIP harus menggunakan metode lain untuk diskon.

```

Gambar 1 Bukti Running Code Salah=

Solusi yang Memenuhi LSP

Untuk memperbaiki pelanggaran terhadap LSP, kita harus memastikan bahwa setiap subclass dari `Pengguna` dapat digunakan secara konsisten dengan cara yang sama tanpa mengubah perilaku yang diharapkan.

```
class Pengguna:
    def __init__(self, nama):
        self.nama = nama

    def hitung_diskon(self):
        return 0 # Default pengguna tidak dapat diskon

class PenggunaPremium(Pengguna):
    def hitung_diskon(self):
        return 20 # Diskon 20% untuk pengguna premium

class PenggunaVIP(Pengguna):
    def hitung_diskon(self):
        return 30 # Diskon 30% untuk pengguna VIP

def cetak_diskon(pengguna):
    print(f"Pengguna {pengguna.nama} mendapatkan diskon sebesar {pengguna.hitung_diskon()}%.")

# Pengguna Reguler
pengguna1 = Pengguna("Bagas")
cetak_diskon(pengguna1)

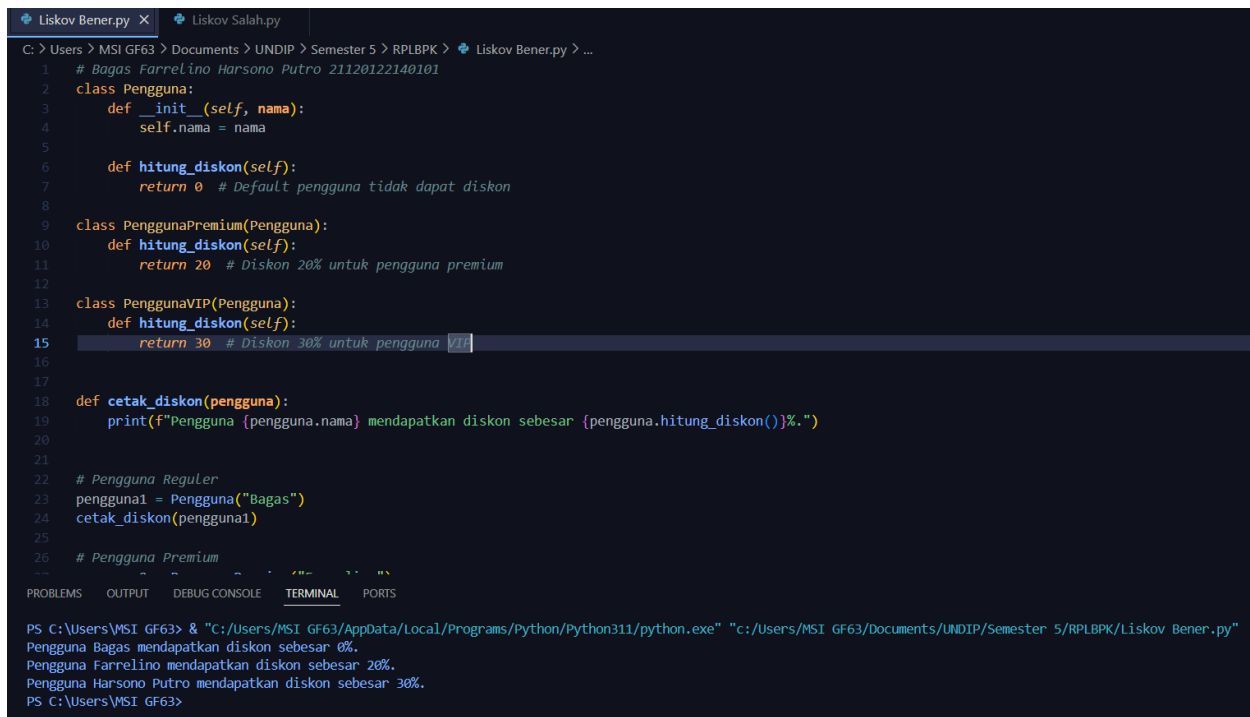
# Pengguna Premium
pengguna2 = PenggunaPremium("Farrelino")
cetak_diskon(pengguna2)

# Pengguna VIP, sekarang tidak menyebabkan error
pengguna3 = PenggunaVIP("Harsono Putro")
cetak_diskon(pengguna3)
```

Penjelasan

- **Kelas** `Pengguna` adalah kelas induk yang memiliki metode `hitung_diskon()` default yang mengembalikan 0 (tanpa diskon).
- **Kelas** `PenggunaPremium` mewarisi kelas `Pengguna` dan menimpa metode `hitung_diskon()` untuk memberikan diskon sebesar 20%.
- **Kelas** `PenggunaVIP` juga menimpa metode `hitung_diskon()` dan memberikan diskon sebesar 30%.

Fungsi `cetak_diskon()` dipakai untuk mencetak diskon yang didapatkan oleh setiap pengguna. Fungsi ini akan bekerja dengan semua jenis pengguna (baik `Pengguna`, `PenggunaPremium`, maupun `PenggunaVIP`), dan tidak ada error yang terjadi karena setiap subclass menghormati kontrak yang ditetapkan oleh superclass.



```
C: > Users > MSI GF63 > Documents > UNIP > Semester 5 > RPLBPK > Liskov Bener.py > ...
1 # Bagas Farrelino Harsono Putro 21120122140101
2 class Pengguna:
3     def __init__(self, nama):
4         self.nama = nama
5
6     def hitung_diskon(self):
7         return 0 # Default pengguna tidak dapat diskon
8
9 class PenggunaPremium(Pengguna):
10     def hitung_diskon(self):
11         return 20 # Diskon 20% untuk pengguna premium
12
13 class PenggunaVIP(Pengguna):
14     def hitung_diskon(self):
15         return 30 # Diskon 30% untuk pengguna VIP
16
17
18 def cetak_diskon(pengguna):
19     print(f"Pengguna {pengguna.nama} mendapatkan diskon sebesar {pengguna.hitung_diskon()}%")
20
21
22 # Pengguna Regular
23 pengguna1 = Pengguna("Bagas")
24 cetak_diskon(pengguna1)
25
26 # Pengguna Premium
27 pengguna2 = PenggunaPremium("Farrelino")
28 cetak_diskon(pengguna2)
29
30 # Pengguna VIP
31 pengguna3 = PenggunaVIP("Harsono")
32 cetak_diskon(pengguna3)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MSI GF63> & "C:/Users/MSI GF63/AppData/Local/Programs/Python/Python311/python.exe" "c:/Users/MSI GF63/Documents/UNIP/Semester 5/RPLBPK/Liskov Bener.py"

Pengguna Bagas mendapatkan diskon sebesar 0%.

Pengguna Farrelino mendapatkan diskon sebesar 20%.

Pengguna Harsono Putro mendapatkan diskon sebesar 30%.

PS C:\Users\MSI GF63>

Gambar 2 Bukti Running Code Benar