

Minggu, 1 September 2024

Nama : Kadek Wisnu Parijata Putra

NIM : 21120122140036

Prodi : Teknik Komputer / 2022

Mata Kuliah : RPLBK / B

(S) Single Responsibility Principle

Single Responsibility Principle (SRP) adalah salah satu dari Lima Prinsip dalam SOLID Principles, SRP ini memiliki prinsip dimana, sebuah kelas atau method seharusnya memiliki satu tanggung jawab atau satu pekerjaan saja, setiap kelas harus memiliki satu alasan untuk berubah, yaitu hanya ada satu tanggung jawab yang harus dipenuhi oleh kelas tersebut.

a. Source Code yang Melanggar Aturan *Single Responsibility Principle* (SRP) :

```
class PemesananPenerbangan:
    def __init__(self, nomor_penerbangan,
nama_penumpang, kelas_tempat_duduk, harga_dasar,
jumlah_tiket):
        self.nomor_penerbangan = nomor_penerbangan
        self.nama_penumpang = nama_penumpang
        self.kelas_tempat_duduk = kelas_tempat_duduk
        self.harga_dasar = harga_dasar
        self.jumlah_tiket = jumlah_tiket
        self.pemesanan = []

    def hitung_total_harga(self):
        pengali_kelas = 1.0
        if self.kelas_tempat_duduk == "Bisnis":
            pengali_kelas = 1.5
        elif self.kelas_tempat_duduk == "Pertama":
            pengali_kelas = 2.0

        total_harga = self.harga_dasar * pengali_kelas *
self.jumlah_tiket
        return total_harga

    def cetak_rincian_pemesanan(self):
        print(f"Nomor {self.nomor_penerbangan} Penerbangan:
{self.nama_penumpang}")
        print(f>Nama Penumpang: {self.nama_penumpang}")
```

```

        print(f"Kelas          Tempat          Duduk:
{self.kelas_tempat_duduk}")
        print(f"Jumlah Tiket: {self.jumlah_tiket}")
        print(f"Total          Harga:
Rp{self.hitung_total_harga() }")

    def simpan_pemesanan(self):
        self.pemesanan.append({
            'nomor_penerbangan':
self.nomor_penerbangan,
            'nama_penumpang': self.nama_penumpang,
            'kelas_tempat_duduk':
self.kelas_tempat_duduk,
            'harga_dasar': self.harga_dasar,
            'jumlah_tiket': self.jumlah_tiket
        })

    def dapatkan_semua_pemesanan(self):
        return self.pemesanan

pemesanan = PemesananPenerbangan("AB123", "Wisnu
Parijata", "Bisnis", 5000000, 2)
pemesanan.hitung_total_harga()
pemesanan.cetak_rincian_pemesanan()
pemesanan.simpan_pemesanan()
print(pemesanan.dapatkan_semua_pemesanan())

```

Hasil Running :

```

# PS C:\Users\Widius\Downloads\Pemesanan Tiket Pesawat Business (Pctode 58P) & C:\Python112\python.exe "C:\Users\Widius\Downloads\Pemesanan Tiket Pesawat Business (Pctode 58P)\Belanggar.py"
Kelas Penerbangan: AB123
Nama Penumpang: Wisnu Parijata
Kelas Tempat Duduk: Bisnis
Jumlah Tiket: 2
Total Harga: Rp5000000.0
[[{'nomor_penerbangan': 'AB123', 'nama_penumpang': 'Wisnu Parijata', 'kelas_tempat_duduk': 'Bisnis', 'harga_dasar': 5000000, 'jumlah_tiket': 2}]]
# PS C:\Users\Widius\Downloads\Pemesanan Tiket Pesawat Business (Pctode 58P)

```

```

PS C:\Users\Wismu\Downloads\Pemesanan Tiket Pesawat Business (Metode SRP)> & c:/Python312/python.exe "c:/Users/Wismu/Downloads/Pemesanan Tiket Pesawat Business (Metode SRP)/Melanggar.py"
Nomor Penerbangan: AB123
Nama Penumpang: Wismu Parijata
Kelas Tempat Duduk: Bisnis
Jumlah Tiket: 2
Total Harga: Rp15000000.0
[{'nomor_penerbangan': 'AB123', 'nama_penumpang': 'Wismu Parijata', 'kelas_tempat_duduk': 'Bisnis', 'harga_dasar': 5000000, 'jumlah_tiket': 2}]
PS C:\Users\Wismu\Downloads\Pemesanan Tiket Pesawat Business (Metode SRP)>

```

Kode diatas termasuk melanggar prinsip *Single Responsibility Principle* (SRP) karena class `PemesananPenerbangan` memiliki lebih dari satu tanggung jawab. SRP diyatakan bahwa setiap kelas harus memiliki satu alasan untuk berubah, yang berarti bahwa kelas hanya boleh memiliki satu tugas atau tanggung jawab. Namun, class `PemesananPenerbangan` menangani beberapa tanggung jawab sekaligus diantaranya:

1. Menghitung total harga tiket berdasarkan kelas tempat duduk.
2. Mencetak rincian pemesanan.
3. Menyimpan pemesanan ke dalam daftar.
4. Mengelola daftar semua pemesanan yang disimpan.

b. Source Code yang Menerapkan Aturan *Single Responsibility Principle* (SRP) :

```

# Kelas yang hanya bertanggung jawab untuk menyimpan data
pemesanan

class DataPemesanan:
    def __init__(self, nomor_penerbangan, nama_penumpang,
        kelas_duduk, harga_dasar, jumlah_tiket):
        self.nomor_penerbangan = nomor_penerbangan
        self.nama_penumpang = nama_penumpang
        self.kelas_duduk = kelas_duduk
        self.harga_dasar = harga_dasar
        self.jumlah_tiket = jumlah_tiket

# Kelas yang hanya bertanggung jawab untuk menghitung total
harga

class PenghitungHarga:
    @staticmethod
    def hitung_total_harga(data_pemesanan):
        pengali_kelas = 1.0
        if data_pemesanan.kelas_duduk == "Bisnis":
            pengali_kelas = 1.5
        elif data_pemesanan.kelas_duduk == "First":
            pengali_kelas = 2.0

        total_harga = data_pemesanan.harga_dasar *
            pengali_kelas * data_pemesanan.jumlah_tiket

```

```

        return total_harga

# Kelas yang hanya bertanggung jawab untuk mencetak detail
pemesanan
class PencetakPemesanan:
    @staticmethod
    def cetak_detail_pemesanan(data_pemesanan,
total_harga):
        print(f"Nomor Penerbangan:
{data_pemesanan.nomor_penerbangan}")
        print(f>Nama Penumpang:
{data_pemesanan.nama_penumpang}")
        print(f"Kelas Duduk:
{data_pemesanan.kelas_duduk}")
        print(f"Jumlah Tiket:
{data_pemesanan.jumlah_tiket}")
        print(f"Total Harga: Rp{total_harga}")

# Kelas yang hanya bertanggung jawab untuk menyimpan dan
mengelola data pemesanan
class RepositoryPemesanan:
    def __init__(self):
        self.pemesanan = []

    def simpan_pemesanan(self, data_pemesanan):
        self.pemesanan.append(data_pemesanan)

    def ambil_semua_pemesanan(self):
        return self.pemesanan

# Contoh Penggunaan
data_pemesanan = DataPemesanan("AB123", "Wisnu Parijata",
"Bisnis", 5000000, 2) # Membuat data pemesanan
penghitung_harga = PenghitungHarga() # Menghitung harga
total_harga =
penghitung_harga.hitung_total_harga(data_pemesanan)
PencetakPemesanan.cetak_detail_pemesanan(data_pemesanan,
total_harga) # Mencetak detail pemesanan

# Menyimpan pemesanan

```

```

repository_pemesanan = RepositoryPemesanan()

repository_pemesanan.simpan_pemesanan(data_pemesanan)

print(repository_pemesanan.ambil_semua_pemesanan()) # Mengambil
semua pemesanan

```

Hasil *Running* :

```

# Kelas yang bertanggung jawab untuk menyimpan data pemesanan
class DataPemesanan:
    def __init__(self, nomor_penerbangan, nama_penumpang, kelas_duduk, harga_dasar, jumlah_tiket):
        self.nomor_penerbangan = nomor_penerbangan
        self.nama_penumpang = nama_penumpang
        self.kelas_duduk = kelas_duduk
        self.harga_dasar = harga_dasar
        self.jumlah_tiket = jumlah_tiket

# Kelas yang bertanggung jawab untuk menghitung total harga
class PenghitungHarga:
    def hitung_total_harga(data_pemesanan):
        pengali_kelas = 1.0
        if data_pemesanan.kelas_duduk == "Bisnis":
            pengali_kelas = 1.5
        elif data_pemesanan.kelas_duduk == "First":
            pengali_kelas = 2.0
        total_harga = data_pemesanan.harga_dasar * pengali_kelas * data_pemesanan.jumlah_tiket
        return total_harga

# Kelas yang bertanggung jawab untuk mencetak detail pemesanan
class PencetakPemesanan:
    def cetak_detail_pemesanan(data_pemesanan, total_harga):
        print(f"Nomor Penerbangan: {data_pemesanan.nomor_penerbangan}")
        print(f>Nama Penumpang: {data_pemesanan.nama_penumpang}")
        print(f>Kelas Duduk: {data_pemesanan.kelas_duduk}")
        print(f"Jumlah Tiket: {data_pemesanan.jumlah_tiket}")

# Kelas yang bertanggung jawab untuk mengelola daftar pemesanan
class RepositoryPemesanan:
    def __init__(self):
        self.pemesanan_list = []

    def simpan_pemesanan(self, data_pemesanan):
        self.pemesanan_list.append(data_pemesanan)

    def ambil_semua_pemesanan(self):
        return self.pemesanan_list

# Fungsi utama
def main():
    repo = RepositoryPemesanan()
    data = DataPemesanan("AB123", "Wisnu Parijata", "Bisnis", 15000000.0, 2)
    repo.simpan_pemesanan(data)
    hitung = PenghitungHarga
    total = hitung.hitung_total_harga(data)
    cetak = PencetakPemesanan
    cetak.cetak_detail_pemesanan(data, total)

if __name__ == "__main__":
    main()

```

```

PS C:\Users\Wisnu\Downloads\Pemesanan Tiket Pesawat Business (Metode SRP)> & C:/Python312/python.exe
Nomor Penerbangan: AB123
Nama Penumpang: Wisnu Parijata
Kelas Duduk: Bisnis
Jumlah Tiket: 2
Total Harga: Rp15000000.0
[<_main_.DataPemesanan object at 0x000001BBC51E07A0>]
PS C:\Users\Wisnu\Downloads\Pemesanan Tiket Pesawat Business (Metode SRP)>

```

Kode diatas menerapkan prinsip *Single Responsibility Principle* (SRP). Karena setiap kelas dalam kode tersebut memiliki satu tanggung jawab yang jelas dan spesifik. Bahwa setiap kelas harus memiliki satu alasan untuk berubah, yang berarti kelas tersebut hanya melakukan satu tugas atau fungsi tertentu, diantaranya sebagai kode berikut :

1. 'DataPemesanan' hanya bertanggung jawab untuk menyimpan informasi atau data pemesanan.
2. 'PenghitungHarga' bertanggung jawab untuk menghitung total harga berdasarkan data pemesanan yang diberikan.
3. 'PencetakPemesanan' fokus pada tugas mencetak detail pemesanan ke *output*.
4. 'RepositoryPemesanan' bertugas untuk menyimpan dan mengelola daftar semua pemesanan yang dibuat.

Dengan memisahkan tanggung jawab ke dalam kelas yang spesifik, kode menjadi lebih mudah dipelihara, dikelola dan lebih fleksibel dalam hal perubahan.