

**TUGAS : UNIT TESTING**  
**MINGGU PERTEMUAN 10**  
**PEMROGRAMAN BERORIENTASI OBJEK (PR)**



Disusun oleh :  
Muhammad Wildan Gumilang (231511087)

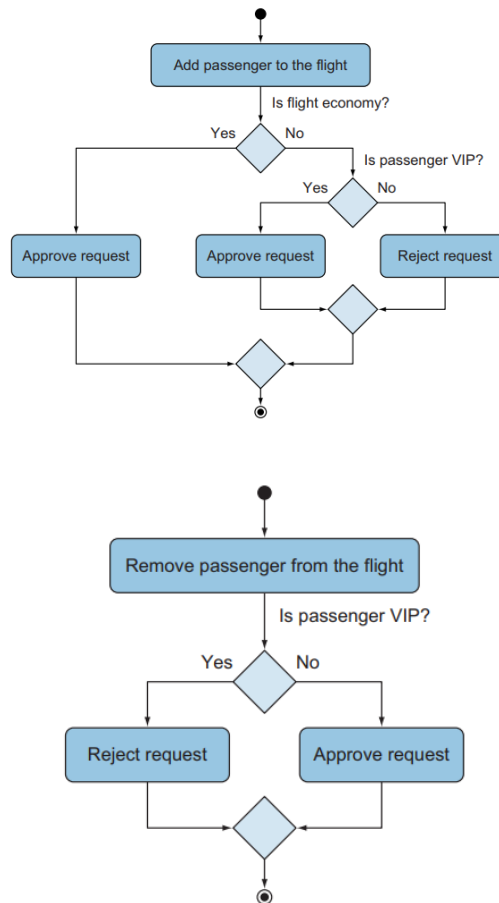
**PROGRAM DIPLOMA III TEKNIK INFORMATIKA**  
**JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA**  
**POLITEKNIK NEGERI BANDUNG**

# DAFTAR ISI

<b>DAFTAR ISI</b> .....	2
<b>Studi Kasus</b> .....	3
<b>Kesimpulan</b> .....	5
<b>Link Repository GitHub</b> .....	5

# Studi Kasus

Pada Praktikum ini, terdapat 2 jenis passenger yaitu Regular dan juga VIP. Selain itu, ada 2 jenis flight, yaitu Economy flight dan juga Bussiness flight.



Berdasarkan diagram di atas, terdapat beberapa aturan pada passenger dalam studi kasus ini. Yaitu sebagai berikut :

## 1. Economy Flight:

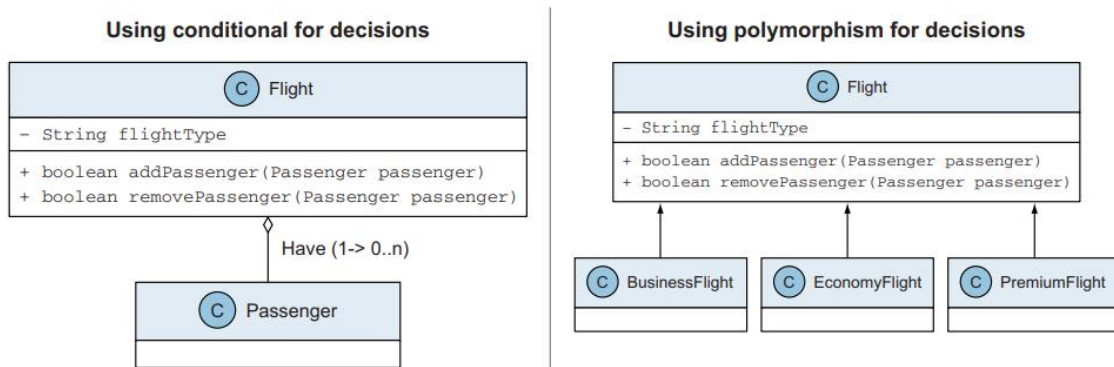
- Penambahan:
  - Regular Passenger: Bisa
  - VIP Passenger: Bisa
- Penghapusan:
  - Regular Passenger: Bisa
  - VIP Passenger: Tidak Bisa

## 2. Business Flight:

- Penambahan:
  - Regular Passenger: Tidak Bisa
  - VIP Passenger: Bisa

- Penghapusan:
  - Regular Passenger: Tidak Bisa
  - VIP Passenger: Tidak Bisa

Dalam praktikum ini, dibuat program untuk melakukan pengecekan menggunakan JUnit dari aturan yang telah dijelaskan di atas. Yaitu untuk mengecek ke-validan passenger jika dilakukan penambahan ataupun penghapusan.



Dalam praktikum ini juga class flight diubah menjadi abstract class yang memiliki concrete class : BussinesFlight, dan EconomyFlight.

Disini untuk melakukan pengecekannya menggunakan fungsi AssertEquals dari JUnit. Misalnya disini ditambahkan objek baru untuk EconomyFlight yaitu :

```
Passenger mike = new Passenger(name:"Mike", vip:false);
```

Objek tersebut akan ditambahkan ke dalam EconomyFlight, dengan status VIP yang false, berdasarkan aturan diatas jika dilakukan pengecekan melalui

```
assertEquals(true, economyFlight.addPassenger(mike));
```

Dengan expected : true, maka hasilnya tidak ada error. Karena passenger EconomyFlight bisa ditambahkan jika penumpang VIP maupun bukan.

Dan jika dilakukan penghapusan, melalui

```
assertEquals(true, economyFlight.removePassenger(mike));
```

Dengan expected : true maka karena Mike bukan penumpang VIP maka hasilnya tidak ada error. Karena passenger EconomyFlight yang bukan VIP dapat dihapus.

The screenshot shows the "Test Runner for Java" window. It displays a green checkmark icon next to the test name `testEconomyFlightRegularPassenger()`, indicating that the test passed successfully.

Sedangkan jika ditambahkan objek baru pada BussinessFlight, yaitu :

```
Passenger james = new Passenger(name:"James", vip:false);
```

Objek tersebut akan ditambahkan ke dalam BussinessFlight, dengan status VIP yang false, berdasarkan aturan diatas jika dilakukan pengecekan melalui

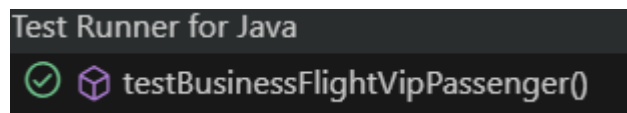
```
assertEquals(false, businessFlight.addPassenger(james));
```

Dengan expected : false, maka hasilnya tidak ada error. Karena passenger BussinessFlight tidak bisa ditambahkan jika penumpang bukan VIP.

Dan jika dilakukan penghapusan, melalui

```
assertEquals(false, businessFlight.removePassenger(james));
```

Dengan expected : false maka karena James bukan penumpang VIP maka hasilnya tidak ada error. Karena seluruh passenger BussinessFlight yang VIP maupun bukan tidak dapat dihapus.



## Kesimpulan

Jadi, dalam praktikum ini digunakan Test-Driven Development (TDD) yang menggunakan JUnit 5. Dengan studi kasus penerbangan, saya mempelajari cara untuk melakukan pengecekan kesalahan dalam suatu program dengan menggunakan JUnit ini. Sehingga dengan digunakan fungsi AssertEqual, akan terlihat jika terdapat kesalahan dengan expected serta actual resultnya pada saat dilakukan testing.

## Link Repository GitHub

<https://github.com/WildanGumilang/PBO-praktek.git>