

1. Mengapa perlu menginisiasi object PilotModel, sedangkan yang di test hanya FlightModel?
  - ➔ Karena di dalam instance FlightModel, terdapat atribut Pilot yang membutuhkan object dari PilotModel, sehingga perlu adanya object PilotModel untuk mengisi atribut tersebut.
2. Jelaskan apa yang akan terjadi jika object PilotModel dihapus dan tidak dilakukan setPilot pada FlightModel?
  - ➔ Jika dihapus program tidak error, namun ketika dijalank unit test oleh Junit akan muncul error karena pada database terdapat constraint bahwa pilot\_license\_number tidak boleh bernilai null, sedangkan yang terjadi adalah object PilotModel yang kosong dihubungkan dengan FlightModel.
3. Jelaskan apa yang dilakukan oleh code  
`Mockito.when(flightDb.findByFlightNumber(flight.get().getFlightNumber())).thenReturn(flight);`
  - ➔ Tujuannya untuk melakukan test pada method findByFlightNumber pada FlightDb dengan melakukan konfigurasi agar mock melakukan return object flight, ketika flightDb menemukan FlightModel dengan flightNumber sesuai dengan flightNumber dari data mock FlightModel yang telah dibuat, maka Mockito mengembalikan flight tersebut.
4. Jelaskan apa yang dilakukan oleh code  
`Mockito.when(flightService.getFlightDetailByFlightNumber(flight.get().getFlightNumber())).thenReturn(flight);`
  - ➔ Tujuannya untuk melakukan test pada method getFlightDetailByFlightNumber pada Flight Service dengan melakukan konfigurasi agar mock melakukan return object flight, ketika flightService menemukan FlightModel dengan flightNumber sesuai dengan flightNumber dari data mock FlightModel yang telah dibuat, maka Mockito mengembalikan flight tersebut.
5. Jelaskan apa yang dilakukan oleh code  
`.andExpect(MockMvcResultMatchers.status().isOk())`
  - ➔ Code ini berfungsi untuk memeriksa bahwa response status yang diterima ketika mengakses link terkait adalah OK atau berhasil (200).

6. Jelaskan apa yang dilakukan oleh code

```
.andExpect(MockMvcResultMatchers.jsonPath("$.flightNumber",  
Matchers.is(flight.get().getFlightNumber())));
```

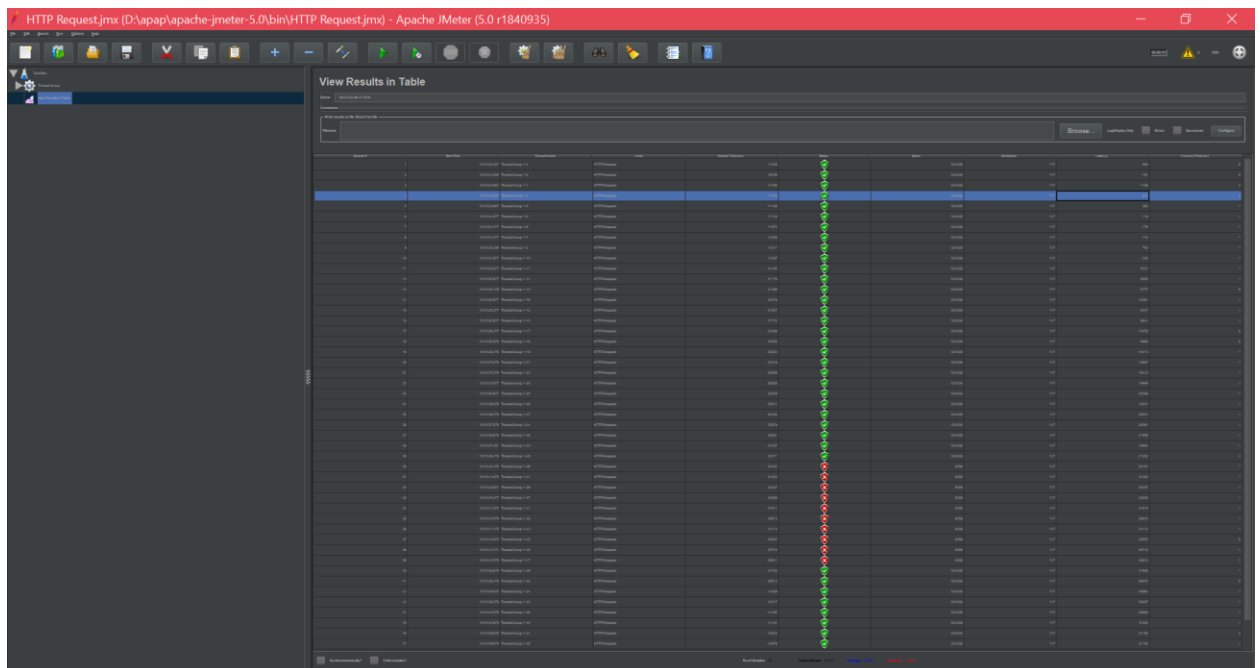
➔ Code ini berfungsi untuk memeriksa bahwa ditemukan subset dari response body, dalam hal ini adalah flightNumber saat melakukan view flight, yang cocok dengan flightNumber dari object flight (pembandingnya). JsonPath berfungsi untuk mengambil bagian yang kita perlukan saja dari dokumen JSON yang dikembalikan

7. Jelaskan anotasi @ResponseBody yang ada pada route "/flight/view"

```
@RequestMapping(value = "/flight/view", method = RequestMethod.GET)  
private @ResponseBody FlightModel view(@RequestParam(value =  
"flightNumber") String flightNumber, Model model) {  
    FlightModel archive =  
    flightService.getFlightDetailByFlightNumber(flightNumber).get();  
    return archive;  
}
```

➔ Anotasi @ResponseBody berfungsi untuk memberitahu controller bahwa object yang direturn secara otomatis ditranslate menjadi JSON dan dikirim kembali menjadi object HttpResponse.

8. Screenshot hasil testing JMeter anda dan letakan pada folder project untuk di push juga ke repository. Jelaskan pada write up, mengapa ada hasil request dengan status merah.



- Gambar di atas menunjukkan bahwa dari 50 percobaan yang dilakukan, hasil test yang muncul adalah 40 test berhasil (ditunjukkan dari icon berwarna hijau) dan 10 test gagal (ditunjukkan dari icon berwarna merah). Percobaan yang gagal ini disebabkan oleh performa komputer yang tidak dapat memadai seluruh test. Pada tes dilakukan penambahan thread pada link terkait, namun terdapat thread yang gagal untuk dipenuhi karena kapasitas loadnya sudah penuh, sehingga tes menjadi gagal