### TUTORIAL 6 APAP

### Unit Testing dan Load Testing Web App Sederhana

# 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, code tidak error. Namun, saat test dijalankan, menghasilkan error, karena pada database terdapat constraint bahwa pilot\_license\_number tidak boleh bernilai *null*, sedangkan kita tidak membuat object PilotModel dan menghubungkannya ke FlightModel yang kita buat.

#### 3. Jelaskan yang dilakukan code ini:

Mockito.when(flightDb.findByFlightNumber(flight.get().getFlightNumber())).then
Return(flight);

→ Code ini mengkonfigurasi agar mock melakukan return object flight, ketika memanggil method findByFlightNumber().

#### 4. Jelaskan apa vang dilakukan oleh code:

Mockito.when(flightService.getFlightDetailByFlightNumber(flight.get().getFlightNumber())).thenReturn(flight);

→ Code ini mengkonfigurasi agar mock melakukan return object flight, ketika memanggil method getFlightDetailByFlightNumber().

#### 5. Jelaskan apa yang dilakukan oleh code:

```
.andExpect(MockMvcResultMatchers.status().isOk())
```

→ Code ini berfungsi untuk memeriksa bahwa response status/status http yang diterima ketika mengakses link terkait adalah OK (200).

#### 6. Jelaskan apa yang dilakukan oleh code:

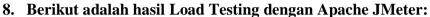
```
.andExpect(MockMvcResultMatchers.jsonPath("$.flightNumber",
Matchers.is(flight.get().getFlightNumber())));
```

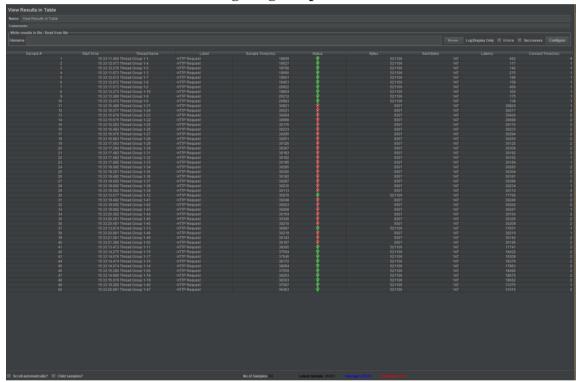
→ Code ini berfungsi untuk memeriksa bahwa ditemukan subset dari response body, dalam hal ini adalah flightNumber ketika kita melakukan view flight, yang cocok dengan yang dibandingkan, dalam hal ini adalah flightNumber dari object flight. JsonPath berfungsi untuk mengambil bagian yang kita perlukan 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*.





Pada hasilnya, status yang hijau menyatakan bahwa hasil tes berhasil, dan yang merah menunjukkan bahwa hasil tes gagal. Pada 50 percobaan, terdapat yang merah karena performa komputer yang digunakan tidak dapat memadai seluruh tes. Pada tes dilakukan penembahan thread pada link terkait, namun terdapat thread yang gagal untuk dipenuhi karena kapasitas loadnya sudah penuh, sehingga tes menjadi gagal.