

1. Mengapa perlu menginisiasi objek PilotModel, sedangkan yang di test hanya FlightModel?
 - Pada objek FlightModel memiliki atribut dari objek PilotModel sebagai tanda penerbangan setiap Flight yang di Pilot-i.
2. Jelaskan apa yang akan terjadi jika objek PilotModel dihapus dan tidak dilakukan setPilot pada FlightModel?
 - Testing akan fail ketika menjalankan baris test `assertThat(found.get(),Matchers.equalTo(flightModel));`

3. Jelaskan apa yang dilakukan oleh *code*

```
Mockito.when(flightDb.findByFlightNumber(flight.get().getFlightNumber())).thenReturn(flight);
```

- Get data flight di db dengan parameter flightNumber, ketika data objek flight tersebut ditemukan, maka akan mengembalikan data tersebut.
4. Jelaskan apa yang dilakukan oleh *code*

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

- Get data flight di db melalui service dari Flight dan menggunakan parameter flightNumber. Data objek flight yang ditemukan akan dikembalikan.
5. Jelaskan apa yang ditest oleh *code*

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

- Digunakan untuk mengecek response terhadap request page yang di test. Jika response HttpStatusnya = 200, maka isOk() menandakan bahwa response terhadap request nya sesuai dengan data yang diekspektasikan.

6. Jelaskan apa yang ditest oleh *code*

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

- Mengetest response flightNumber, apakah sama dengan flightNumber yang ingin di jadikan parameter untuk test. Response selanjutnya diakses dengan jsonPath untuk mengambil flightNumber pada response.

7. Jelaskan anotasi @ResponseBody

```
@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;
}
```

- Memberi tahu *controller* bahwa objek yang dikembalikan secara otomatis dijadikan ke dalam bentuk JSON dan diteruskan kembali ke objek HttpResponse.
8. Load Testing

33	19:37:13.394 Thread Group 1-33 HTTP Request	32087		516929	147	22076	1
34	19:37:13.794 Thread Group 1-35 HTTP Request	32471		516929	147	23796	1
35	19:37:16.195 Thread Group 1-47 HTTP Request	30490		9358	147	30499	1
36	19:37:16.397 Thread Group 1-48 HTTP Request	30477		9358	147	30476	1
37	19:37:13.594 Thread Group 1-34 HTTP Request	33661		516929	147	24033	1
38	19:37:15.596 Thread Group 1-44 HTTP Request	31662		9358	147	31662	1
39	19:37:14.794 Thread Group 1-40 HTTP Request	33743		516929	147	24004	1
40	19:37:13.995 Thread Group 1-36 HTTP Request	34617		516929	147	24795	1
41	19:37:14.194 Thread Group 1-37 HTTP Request	34927		516929	147	24575	1
42	19:37:14.394 Thread Group 1-38 HTTP Request	34958		516929	147	25484	1
43	19:37:14.595 Thread Group 1-39 HTTP Request	35394		516929	147	26834	1
44	19:37:16.797 Thread Group 1-50 HTTP Request	33376		9358	147	33375	1
45	19:37:15.996 Thread Group 1-46 HTTP Request	34181		9358	147	34181	1
46	19:37:15.396 Thread Group 1-43 HTTP Request	36946		516929	147	30357	1
47	19:37:15.196 Thread Group 1-42 HTTP Request	37475		516929	147	30509	1
48	19:37:15.795 Thread Group 1-45 HTTP Request	37107		516929	147	30709	2
49	19:37:14.998 Thread Group 1-41 HTTP Request	38017		516929	147	30877	1
50	19:37:16.597 Thread Group 1-49 HTTP Request	36426		516929	147	31114	1

☒ Scroll automatically?

☐ Child samples?

No of Samples 50

Latest Sample 36426

Average 34197

Deviation 9792

Hasil request ber-status merah menandakan warning akan meningkatnya konsumsi CPU yang mengakibatkan masalah kinerja. Penggunaan *assertion* pada setiap load test mengakibatkan meningkatnya penggunaan *resource* komputer saya sehingga menimbulkan *Out Of Memory errors*.