

Nama : Tengku Nabila Zaskiah

NIM : 2309106104

Kelas : Informatika C'23

Posttest 5 Pemrograman Berorientasi Objek

SS Program

1. Menampilkan menu masuk sebagai Admin/Penumpang

```
Masukkan peran Anda (Admin/Penumpang):
```

2. Menampilkan menu admin

```
Masukkan peran Anda (Admin/Penumpang):  
Admin
```

- ```
1. Tambah Penerbangan
2. Hapus Penerbangan
3. Lihat Penerbangan
4. Keluar
```

```
Pilih opsi:
```

3. Menampilkan menu tambah penerbangan pada admin

```
1. Tambah Penerbangan
2. Hapus Penerbangan
3. Lihat Penerbangan
4. Keluar
Pilih opsi: 1
Flight ID: 01
Asal: Samarinda
Tujuan: Bojong Gede
Tanggal: 1 Januari 2025
Maskapai: Garuda Indonesia
Harga: 2.000.000
Penerbangan berhasil ditambahkan!
```

4. Menampilkan menu hapus penerbangan pada admin

```
1. Tambah Penerbangan
2. Hapus Penerbangan
3. Lihat Penerbangan
4. Keluar
Pilih opsi: 2

+-----+-----+-----+-----+-----+-----+
| Flight ID | Asal | Tujuan | Tanggal | Maskapai | Harga |
+-----+-----+-----+-----+-----+-----+
| 01 | Samarinda | Bojong Gede | 1 Januari 2025 | Garuda Indonesia | Rp2000000,00 |
+-----+-----+-----+-----+-----+-----+
Masukkan Flight ID yang ingin dihapus: 01

Penerbangan 01 berhasil dihapus.
```

5. Menampilkan menu lihat penerbangan pada admin

```

1. Tambah Penerbangan
2. Hapus Penerbangan
3. Lihat Penerbangan
4. Keluar
Pilih opsi: 3

+-----+-----+-----+-----+-----+-----+
| Flight ID | Asal | Tujuan | Tanggal | Maskapai | Harga |
+-----+-----+-----+-----+-----+-----+
| 01 | Samarinda | Bohong Gede | 1 Januari 2025 | Garuda Indonesia | Rp2000000,00 |
+-----+-----+-----+-----+-----+-----+

```

6. Keluar pada menu admin akan masuk ke menu utama

```

1. Tambah Penerbangan
2. Hapus Penerbangan
3. Lihat Penerbangan
4. Keluar
Pilih opsi: 4
Kembali ke menu utama...

=====
Airfly Ticket Booking System
=====
1. Tambah Penerbangan (Admin)
2. Lihat Penerbangan
3. Buat Reservasi
4. Lihat Reservasi
5. Verifikasi Pembayaran
6. Keluar
Masukkan pilihan: █

```

7. Langsung ke menu buat reservasi karena menu 1&2 sama seperti di menu admin sebelumnya

```

=====
Airfly Ticket Booking System
=====
1. Tambah Penerbangan (Admin)
2. Lihat Penerbangan
3. Buat Reservasi
4. Lihat Reservasi
5. Verifikasi Pembayaran
6. Keluar
Masukkan pilihan: 3

+-----+-----+-----+-----+-----+-----+
| Flight ID | Asal | Tujuan | Tanggal | Maskapai | Harga |
+-----+-----+-----+-----+-----+-----+
| 01 | Samarinda | Bojong Gede | 1 Januari 2025 | Garuda Indonesia | Rp2000000,00 |
+-----+-----+-----+-----+-----+-----+
Masukkan Reservasi ID: 122
Masukkan Flight ID: 01
Masukkan nama penumpang: Nabila
Masukkan Harga: 2000000
Reservasi berhasil dibuat dengan harga Rp 2000000.0

```

8. Menampilkan menu lihat rservasi

```

=====
Airfly Ticket Booking System
=====
1. Tambah Penerbangan (Admin)
2. Lihat Penerbangan
3. Buat Reservasi
4. Lihat Reservasi
5. Verifikasi Pembayaran
6. Keluar
Masukkan pilihan: 4

+-----+-----+-----+-----+
| Reservasi ID | Flight ID | Nama Penumpang | Status |
+-----+-----+-----+-----+
| 122 | 01 | Nabila | Pending |
+-----+-----+-----+-----+

```

9. Menampilkan menu verifikasi pembayaran, lalu mengecek Kembali status pembayarannya apakah sudah berubah atau belum

```

=====
Airfly Ticket Booking System
=====
1. Tambah Penerbangan (Admin)
2. Lihat Penerbangan
3. Buat Reservasi
4. Lihat Reservasi
5. Verifikasi Pembayaran
6. Keluar
Masukkan pilihan: 5
Masukkan Reservasi ID untuk verifikasi: 122

Pembayaran berhasil diverifikasi!

=====
Airfly Ticket Booking System
=====
1. Tambah Penerbangan (Admin)
2. Lihat Penerbangan
3. Buat Reservasi
4. Lihat Reservasi
5. Verifikasi Pembayaran
6. Keluar
Masukkan pilihan: 4

+-----+-----+-----+-----+
| Reservasi ID | Flight ID | Nama Penumpang | Status |
+-----+-----+-----+-----+
| 122 | 01 | Nabila | Lunas |
+-----+-----+-----+-----+

```

10. Menampilkan menu keluar, jika memilih “y” maka akan keluar habis dari program, tetapi jika memilih “n” maka tidak jadi keluar dari program

```

=====
Airfly Ticket Booking System
=====
1. Tambah Penerbangan (Admin)
2. Lihat Penerbangan
3. Buat Reservasi
4. Lihat Reservasi
5. Verifikasi Pembayaran
6. Keluar
Masukkan pilihan: 6
Apakah Anda yakin ingin keluar? (y/n):

```

## Source Code :

```
1 import java.util.ArrayList;
2 import java.util.List;
3 import java.util.ListIterator;
4 import java.util.Scanner;
5
6 // Abstract class sebagai parent
7 abstract class Pengguna {
8 protected String userID;
9 protected String nama;
10
11 public Pengguna(String userID, String nama) {
12 this.userID = userID;
13 this.nama = nama;
14 }
15
16 public String getUserID() {
17 return userID;
18 }
19
20 public String getNama() {
21 return nama;
22 }
23
24 // Abstract method wajib dioverride di subclass
25 public abstract void menu(SistemPemesanan sistem, Scanner scanner);
26 }
27
28 // Subclass Admin
29 class Admin extends Pengguna {
30 private final String role = "Admin"; // final variable
31
32 public Admin(String userID, String nama) {
33 super(userID, nama);
34 }
35
36 @Override
37 public void menu(SistemPemesanan sistem, Scanner scanner) {
38 int choice;
39 do {
40 System.out.println("\n1. Tambah Penerbangan");
41 System.out.println("2. Hapus Penerbangan");
42 System.out.println("3. Lihat Penerbangan");
43 System.out.println("4. Keluar");
44 System.out.print("Pilih opsi: ");
45
46 try {
47 choice = scanner.nextInt();
48 scanner.nextLine();
49
50 switch (choice) {
51 case 1:
52 System.out.print("Flight ID: ");
53 String fid = scanner.nextLine();
54 System.out.print("Asal: ");
55 String asal = scanner.nextLine();
56 System.out.print("Tujuan: ");
57 String tujuan = scanner.nextLine();
58 System.out.print("Tanggal: ");
59 String tanggal = scanner.nextLine();
60 System.out.print("Maskapai: ");
61 String maskapai = scanner.nextLine();
62 System.out.print("Harga: ");
63 double harga = scanner.nextDouble();
64 scanner.nextLine();
65 sistem.tambahPenerbangan(fid, asal, tujuan, tanggal, maskapai, harga);
66 break;
67 case 2:
68 sistem.lihatPenerbangan();
69 System.out.print("Masukkan Flight ID yang ingin dihapus: ");
70 String hapusID = scanner.nextLine();
71 sistem.hapusPenerbangan(hapusID);
72 break;
73 case 3:
74 sistem.lihatPenerbangan();
75 break;
76 case 4:
77 System.out.println("Kembali ke menu utama...");
78 return;
79 default:
80 System.out.println("Pilihan tidak valid.");
81 }
82 } catch (Exception e) {
83 System.out.println("Input tidak valid! Masukkan angka.");
84 scanner.nextLine(); // clear buffer
85 }
86 } while (true);
87 }
88
89 // Final method
90 public final void tampilDataAdmin() {
91 System.out.println("Admin: " + nama + " | ID: " + userID + " | Role: " + role);
92 }
93 }
94
```

```

93 }
94
95 // Subclass Penumpang
96 class Penumpang extends Pengguna {
97 public Penumpang(String userID, String nama) {
98 super(userID, nama);
99 }
100
101 @Override
102 public void menu(SistemPemesanan sistem, Scanner scanner) {
103 int choice;
104 do {
105 System.out.println("\n1. Lihat Penerbangan");
106 System.out.println("2. Buat Reservasi");
107 System.out.println("3. Lihat Reservasi");
108 System.out.println("4. Bayar Reservasi");
109 System.out.println("5. Keluar");
110 System.out.print("Pilih opsi: ");
111
112 try {
113 choice = scanner.nextInt();
114 scanner.nextLine();
115
116 switch (choice) {
117 case 1:
118 sistem.lihatPenerbangan();
119 break;
120 case 2:
121 sistem.lihatPenerbangan();
122 System.out.print("Masukkan Flight ID yang ingin dipesan: ");
123 String flightID = scanner.nextLine();
124 System.out.print("Masukkan Reservasi ID: ");
125 String reservasiID = scanner.nextLine();
126 sistem.buatReservasi(reservasiID, flightID, nama, 0.0);
127 break;
128 case 3:
129 sistem.lihatReservasi();
130 break;
131 case 4:
132 sistem.lihatReservasi();
133 System.out.print("Masukkan Reservasi ID yang ingin dibayar: ");
134 String bayarID = scanner.nextLine();
135 sistem.verifikasiPembayaran(bayarID);
136 break;
137 case 5:
138 System.out.println("Kembali ke menu utama...");
139 return; // Keluar dari menu
140 default:
141 System.out.println("Pilihan tidak valid.");
142 }
143 } catch (Exception e) {
144 System.out.println("Input tidak valid! Masukkan angka.");
145 scanner.nextLine();
146 }
147 } while (true);
148 }
149 }
150
151 class Flight {
152 String id;
153 Flight(String id) {
154 this.id = id;
155 }
156 }
157
158 // Class Pembayaran dengan Encapsulation
159 class Pembayaran {
160 private String reservasiID;
161 private double jumlah;
162 private boolean status;
163
164 public Pembayaran(String reservasiID, double jumlah) {
165 this.reservasiID = reservasiID;
166 this.jumlah = jumlah;
167 this.status = false;
168 }
169
170 // Getter untuk jumlah
171 public double getJumlah() {
172 return jumlah;
173 }
174
175 // Metode untuk menampilkan jumlah pembayaran
176 public void tampilkanJumlah() {
177 System.out.println("Jumlah pembayaran: Rp " + jumlah);
178 }
179
180 // Getter untuk status pembayaran
181 public boolean isLunas() {
182 return status;
183 }
184
185 // Metode untuk konfirmasi pembayaran
186 public void konfirmasiPembayaran() {
187 this.status = true;
188 }
189 }

```

```

188 }
189
190 public String getReservasiID() {
191 return reservasiID;
192 }
193 }
194
195 // Class Penerbangan dengan Encapsulation
196 class Penerbangan {
197 private String flightID, asal, tujuan, tanggal, maskapai;
198 private double harga;
199
200 public Penerbangan(String flightID, String asal, String tujuan, String tanggal, String maskapai, double harga) {
201 this.flightID = flightID;
202 this.asal = asal;
203 this.tujuan = tujuan;
204 this.tanggal = tanggal;
205 this.maskapai = maskapai;
206 this.harga = harga;
207 }
208
209 // Getter untuk Flight ID
210 public String getFlightID() {
211 return flightID;
212 }
213
214 // Getter untuk Harga
215 public double getHarga() {
216 return harga;
217 }
218
219 @Override
220 public String toString() {
221 return String.format("| %-10s | %-10s | %-10s | %-12s | %-10s | Rp%-10.2f |", flightID, asal, tujuan, tanggal, maskapai, harga);
222 }
223 }
224
225 // Class Reservasi dengan Encapsulation
226 class Reservasi {
227 private String reservasiID, flightID, namaPenumpang;
228 private boolean statusPembayaran;
229
230 public Reservasi(String reservasiID, String flightID, String namaPenumpang) {
231 this.reservasiID = reservasiID;
232 this.flightID = flightID;
233 this.namaPenumpang = namaPenumpang;
234 this.statusPembayaran = false;
235 }
236
237 // Getter untuk reservasiID
238 public String getReservasiID() {
239 return reservasiID;
240 }
241
242 // Getter untuk status pembayaran
243 public boolean isLunas() {
244 return statusPembayaran;
245 }
246
247 // Setter untuk mengubah status pembayaran
248 public void setStatusPembayaran(boolean status) {
249 this.statusPembayaran = status;
250 }
251
252 @Override
253 public String toString() {
254 return String.format("| %-12s | %-10s | %-15s | %-10s |", reservasiID, flightID, namaPenumpang, (statusPembayaran ? "Lunas" : "Pending"));
255 }
256 }
257
258 // Sistem Pemesanan dengan perubahan Encapsulation
259 class SistemPemesanan {
260 private ArrayList<Penerbangan> penerbanganList = new ArrayList<>();
261 private ArrayList<Reservasi> reservasiList = new ArrayList<>();
262 private ArrayList<Pembayaran> pembayaranList = new ArrayList<>();
263
264 public void tambahPenerbangan(String flightID, String asal, String tujuan, String tanggal, String maskapai, double harga) {
265 for (Penerbangan p : penerbanganList) {
266 if (p.getFlightID().equals(flightID)) {
267 System.out.println("Flight ID sudah ada!");
268 return;
269 }
270 }
271 penerbanganList.add(new Penerbangan(flightID, asal, tujuan, tanggal, maskapai, harga));
272 System.out.println("Penerbangan berhasil ditambahkan!");
273 }
274
275 // Menghapus penerbangan
276 public void hapusPenerbangan(String flightID) {
277 if (penerbanganList.isEmpty()) {
278 System.out.println("\nTidak ada penerbangan yang tersedia.");
279 return;
280 }
281
282 ListIterator<Penerbangan> iterator = penerbanganList.listIterator();
283 while (iterator.hasNext()) {
284 Penerbangan p = iterator.next();
285 if (p.getFlightID().equals(flightID)) {
286 iterator.remove();
287 System.out.println("\nPenerbangan " + flightID + " berhasil dihapus.");
288 return;

```

```

288 return;
289 }
290 }
291 System.out.println("\nPenerbangan tidak ditemukan!");
292 }
293
294 public void lihatPenerbangan() {
295 if (penerbanganList.isEmpty()) {
296 System.out.println("\nTidak ada penerbangan yang tersedia.");
297 } else {
298 System.out.println("\n+-----+-----+-----+-----+-----+");
299 System.out.println("| Flight ID | Asal | Tujuan | Tanggal | Maskapai | Harga |");
300 System.out.println("+-----+-----+-----+-----+-----+");
301 for (Penerbangan p : penerbanganList) {
302 System.out.println(p);
303 }
304 System.out.println("+-----+-----+-----+-----+-----+");
305 }
306 }
307
308 // Versi utama: dengan harga eksplisit (walau tidak digunakan karena pakai harga dari penerbangan)
309 public void buatReservasi(String reservasiID, String flightID, String namaPenumpang, double hargaReservasi) {
310 if (isReservasiIDExist(reservasiID)) {
311 System.out.println("Reservasi ID sudah digunakan!");
312 return;
313 }
314
315 Penerbangan flight = cariPenerbangan(flightID);
316 if (flight != null) {
317 reservasiList.add(new Reservasi(reservasiID, flightID, namaPenumpang));
318 pembayaranList.add(new Pembayaran(reservasiID, flight.getHarga()));
319 System.out.println("Reservasi berhasil dibuat dengan harga Rp " + flight.getHarga());
320 } else {
321 System.out.println("Penerbangan tidak ditemukan!");
322 }
323 }
324
325 // Overloading 1: tanpa harga, akan otomatis ambil dari harga penerbangan
326 public void buatReservasi(String reservasiID, String flightID, String namaPenumpang) {
327 buatReservasi(reservasiID, flightID, namaPenumpang, 0);
328 }
329
330 // Overloading 2: tanpa flightID (misal untuk pre-booking atau draft reservasi)
331 public void buatReservasi(String reservasiID, String namaPenumpang) {
332 if (isReservasiIDExist(reservasiID)) {
333 System.out.println("Reservasi ID sudah digunakan!");
334 return;
335 }
336 reservasiList.add(new Reservasi(reservasiID, "Belum-Ditentukan", namaPenumpang));
337 pembayaranList.add(new Pembayaran(reservasiID, 0));
338 System.out.println("Reservasi sementara berhasil dibuat untuk " + namaPenumpang);
339 }
340
341 // Fungsi bantu
342 private boolean isReservasiIDExist(String reservasiID) {
343 for (Reservasi r : reservasiList) {
344 if (r.getReservasiID().equals(reservasiID)) {
345 return true;
346 }
347 }
348 return false;
349 }
350
351 private Penerbangan cariPenerbangan(String flightID) {
352 for (Penerbangan p : penerbanganList) {
353 if (p.getFlightID().equals(flightID)) {
354 return p;
355 }
356 }
357 return null;
358 }
359
360 public void verifikasiPembayaran(String reservasiID) {
361 for (Pembayaran p : pembayaranList) {
362 if (p.getReservasiID().equals(reservasiID)) {
363 p.konfirmasiPembayaran();
364 for (Reservasi r : reservasiList) {
365 if (r.getReservasiID().equals(reservasiID)) {
366 r.setStatusPembayaran(true);
367 System.out.println("\nPembayaran berhasil diverifikasi!");
368 return;
369 }
370 }
371 }
372 }
373 System.out.println("\nReservasi tidak ditemukan atau pembayaran belum dilakukan.");
374 }
375
376 public void lihatReservasi() {
377 if (reservasiList.isEmpty()) {
378 System.out.println("\nTidak ada reservasi.");
379 } else {
380 System.out.println("\n+-----+-----+-----+-----+");
381 System.out.println("| Reservasi ID | Flight ID | Nama Penumpang | Status |");
382 System.out.println("+-----+-----+-----+-----+");
383 for (Reservasi r : reservasiList) {
384 System.out.println(r);
385 }
386 System.out.println("+-----+-----+-----+-----+");
387 }
388 }

```

```

387 }
388 }
389 }
390
391
392 // Main Program
393 public class AirflyTicketPSystem {
394 public static void main(String[] args) {
395 Scanner scanner = new Scanner(System.in);
396 SistemPemesanan sistem = new SistemPemesanan();
397 boolean running = true; // Tambahkan variabel running agar tidak error
398
399 do {
400 System.out.println("Masukkan peran Anda (Admin/Penumpang): ");
401 String role = scanner.nextLine();
402
403 if (role.equalsIgnoreCase("Admin")) {
404 Admin admin = new Admin("A001", "Admin1");
405 admin.menu(sistem, scanner);
406 } else {
407 Penumpang penumpang = new Penumpang("P001", "Nabila");
408 penumpang.menu(sistem, scanner);
409 }
410
411 while (running) {
412 System.out.println("\n=====");
413 System.out.println(" AirFly Ticket Booking System ");
414 System.out.println("=====");
415 System.out.println("1. Tambah Penerbangan (Admin)");
416 System.out.println("2. Lihat Penerbangan");
417 System.out.println("3. Buat Reservasi");
418 System.out.println("4. Lihat Reservasi");
419 System.out.println("5. Verifikasi Pembayaran");
420 System.out.println("6. Keluar");
421 System.out.print("Masukkan pilihan: ");
422
423 int choice = scanner.nextInt();
424 scanner.nextLine();
425
426 switch (choice) {
427 case 1:
428 System.out.print("Masukkan Flight ID: ");
429 String fid = scanner.nextLine();
430 System.out.print("Masukkan Asal: ");
431 String asal = scanner.nextLine();
432 System.out.print("Masukkan Tujuan: ");
433 String tujuan = scanner.nextLine();
434 System.out.print("Masukkan Tanggal: ");
435 String tanggal = scanner.nextLine();
436 System.out.print("Masukkan Maskapai: ");
437 String maskapai = scanner.nextLine();
438 System.out.print("Masukkan Harga: ");
439 double harga = scanner.nextDouble();
440 scanner.nextLine();
441 sistem.tambahPenerbangan(fid, asal, tujuan, tanggal, maskapai, harga);
442 break;
443 case 2:
444 sistem.lihatPenerbangan();
445 break;
446 case 3:
447 sistem.lihatPenerbangan();
448 System.out.print("Masukkan Reservasi ID: ");
449 String reservasiID = scanner.nextLine();
450 System.out.print("Masukkan Flight ID: ");
451 String flightID = scanner.nextLine();
452 System.out.print("Masukkan nama penumpang: ");
453 String namaPenumpang = scanner.nextLine();
454 System.out.print("Masukkan Harga: ");
455 double hargaReservasi = scanner.nextDouble();
456 scanner.nextLine();
457 sistem.buatReservasi(reservasiID, flightID, namaPenumpang, hargaReservasi);
458 break;
459 case 4:
460 sistem.lihatReservasi();
461 break;
462 case 5:
463 System.out.print("Masukkan Reservasi ID untuk verifikasi: ");
464 String verifikasiID = scanner.nextLine();
465 sistem.verifikasiPembayaran(verifikasiID);
466 break;
467 case 6:
468 System.out.print("Apakah Anda yakin ingin keluar? (y/n): ");
469 String konfirmasi = scanner.nextLine();
470 if (konfirmasi.equalsIgnoreCase("y")) {
471 System.out.println("Terimakasih Telah Menggunakan Program Ini....");
472 running = false;
473 }
474 break;
475 default:
476 System.out.println("Pilihan tidak valid, coba lagi.");
477 }
478 }
479 } while (running);
480
481 scanner.close();
482 }
483 }
484
485

```