Nama : Adli Imam Suryadin
NIM : 2209116096
Kelas : Sistem Informasi B 22

Post Test 2 PBO

Penjelasan Kode:

Player.java

```java
package com.Entitas;

public final class Player {

    private final String name;
    private final int strength;
    private int health;

    public Player(String name, int strength, int health){
        this.name = name;
        this.strength = strength;
        this.health = health;
    }

    public final void show_stats(){
        System.err.println("Nama pemain : " + this.name);
        System.out.println("Kekuatan : " + this.strength);
        System.out.println("HP : " + this.health);
    }

    public final int get_health() {
        return this.health;
    }

    public final void set_health(int health) {
        this.health = health;
    }

    public final void hitPillow(Enemy enemy){
        System.out.println(x:"Anda memukul dengan bantal...");
        int healthReduction = enemy.get_health() - 100 + (int)(this.strength * 0.65);
        enemy.set_health(healthReduction);
        if (enemy.get_health() < 0) {
            enemy.set_health(health:0);
        }
        System.out.println("Nyawa musuh : " + enemy.get_health());
    }

    public final void throwPillow(Enemy enemy){
        System.out.println(x:"Anda melempar dengan bantal...");
        int healthReduction = enemy.get_health() - 300 + (int)(this.strength * 0.75);
        enemy.set_health(healthReduction);
        if (enemy.get_health() < 0) {
            enemy.set_health(health:0);
        }
        System.out.println("Nyawa musuh : " + enemy.get_health());
    }
}
```

1. ***class Player {}***

   Pembuatan kelas pemain dengan nama Player.

2. ***private final String name;***
   private final int strength;
   private int health;

   Memberi atribut untuk kelas Player, yaitu name (nama player), strength (kekuatan player) dan health (nyawa player). Keyword private digunakan agar kedua atribut tidak dapat diakses atau diubah secara langsung. Sedangkan keyword final digunakan agar atribut tersebut tidak bisa diubah dan di-override lagi selanjutnya (health tidak menggunakan final karena health akan berkurang saat nanti diserang, sehingga penggunaan keyword final tidak valid untuk atribut health).

3. ***public Player(String name, int strength, int health){}***

   Membuat method constructor agar kita dapat langsung memberi nilai pada atribut *name, strength* dan *health* saat pembuatan (instantiating) objek.

4. ***public final void show_stats(){}***

   Membuat method untuk menampilkan status dari player yang telah dibuat menjadi objek.

5. ***public final int get_health(){}***

   Membuat method untuk mengakses atribut kelas. Method ini jika dipanggil akan mengembalikan nilai health yang bersifat private di dalam kelas.

6. ***public final void set_health(int health) {}***

   Membuat method untuk mengubah nilai health pada objek. Method ini jika dipanggil akan menetapkan this.health (di objek) menjadi health (dari argumen pada method).

7. ***public final void hitPillow(Enemy enemy){}***

   Membuat method untuk melakukan serangan pukulan kepada musuh. Method ini jika dipanggil akan mengurangi health musuh berdasarkan strength dari player (`enemy.get_health() - 100 + (int)(this.strength * 0.65)`). Lalu kita memanggil method set_health() di dalamnya agar health musuh tidak sampai bernilai minus dengan mengubah nilainya ke 0.

8. ***public final void throwPillow(Enemy enemy){}***

   Membuat method untuk melakukan serangan pukulan kepada musuh. Method ini jika dipanggil akan mengurangi health musuh berdasarkan strength dari player (`enemy.get_health() - 300 + (int)(this.strength * 0.75)`). Lalu kita memanggil method set_health() di dalamnya agar health musuh tidak sampai bernilai minus dengan mengubah nilainya ke 0.

Enemy.java

```java
package com.Entitas;
import java.util.concurrent.ThreadLocalRandom;

public final class Enemy{

    private final String name;
    private final int strength;
    private int health;

    public Enemy(){
        this.name = "Musuh";
        this.strength = ThreadLocalRandom.current().nextInt(origin:100, 199 + 1);
        this.health = ThreadLocalRandom.current().nextInt(origin:1000, 2999 + 1);
    }

    public final void show_stats(){
        System.out.println("Nama musuh : " + this.name);
        System.out.println("Kekuatan : " + this.strength);
        System.out.println("HP : " + this.health);
    }

    public final int get_health() {
        return this.health;
    }

    public final void set_health(int health) {
        this.health = health;
    }

    public final void hitPillow(Player player){
        System.out.println(x:"\nMusuh memukul dengan bantal...");
        int healthReduction = player.get_health() - 300 + (int)(this.strength * 0.2);
        player.set_health(healthReduction);
        if (player.get_health() < 0) {
            player.set_health(health:0);
        }
        System.out.println("Nyawa anda : " + player.get_health());
    }

    public final void throwPillow(Player player){
        System.out.println(x:"Musuh melempar dengan bantal...");
        int healthReduction = player.get_health() - 450 + (int)(this.strength * 0.5);
        player.set_health(healthReduction);
        if (player.get_health() < 0) {
            player.set_health(health:0);
        }
        System.out.println("Nyawa anda : " + player.get_health());
    }
```

```
49
50      public final void attacking(Player player) {
51
52          int random = ThreadLocalRandom.current().nextInt(origin:1, 1 + 1);
53
54          if (random == 1) {
55              this.hitPillow(player);
56          }
57          else if (random == 2) {
58              this.throwPillow(player);
59          }
60      }
61
62  }
63
```

Untuk file Enemy. java, atribut - atribut dan method - methodnya memiliki isi dan perilaku yang sama dengan yang ada di file Player.java sebelumnya, kecuali :

9. ***public Enemy(){***
   this.name = "Musuh";
   this.strength = ThreadLocalRandom.current().nextInt(100, 199 + 1);
   this.health = ThreadLocalRandom.current().nextInt(1000, 2999 + 1);
   }

   Karena kita ingin agar saat pertarungan dimulai, musuh akan memiliki atribut otomatis tanpa dimasukkan secara manual. Maka kita mengubah isi konstruktornya disini, dimana *name* musuh akan diinisialisasi dengan “Musuh”. Lalu *strength* dan *health-nya* kita atur agar menjadi nilai random (*strength* dari 100 - 200, *health* dari 1000 - 3000).

10. ***public final void hitPillow(Enemy enemy){}***

    Membuat method untuk melakukan serangan pukulan kepada player. Method ini jika dipanggil akan mengurangi health player berdasarkan strength dari musuh(`player.get_health() - 300 + (int)(this.strength * 0.2)`). Lalu kita memanggil method set_health() di dalamnya agar health player tidak sampai bernilai minus dengan mengubah nilainya ke 0.

11. ***public final void throwPillow(Player player){}***

    Membuat method untuk melakukan serangan pukulan kepada player. Method ini jika dipanggil akan mengurangi health player berdasarkan strength dari musuh(`player.get_health() - 450 + (int)(this.strength * 0.5)`

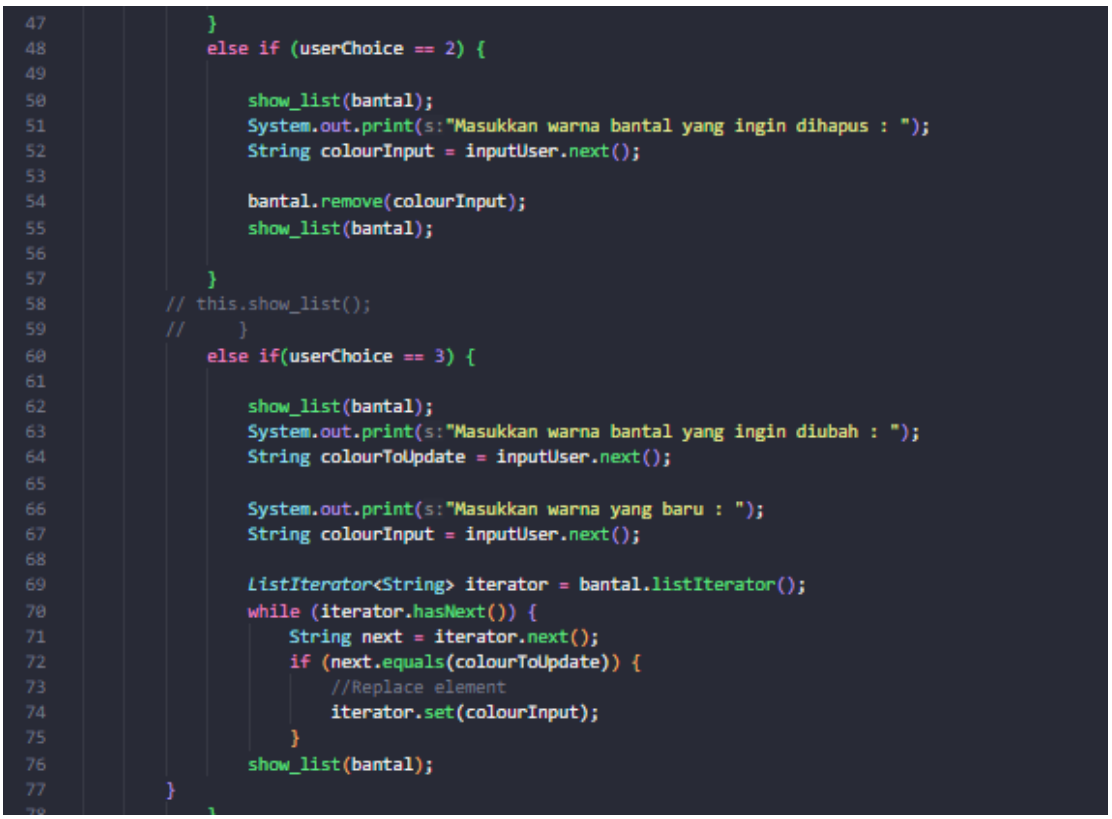
). Lalu kita memanggil method set_health() di dalamnya agar health player tidak sampai bernilai minus dengan mengubah nilainya ke 0.

12. public final void attacking(Player player) {}

Agar musuh bisa secara random memukul atau melempar bantal, maka kita buatkan fungsi khusus agar saat menyerang, serangannya bisa bervariasi secara otomatis. (int random = ThreadLocalRandom.current().nextInt(1, 1 + 1);)

Main.java

```java
public class Main {
    Run | Debug
    public static void main(String[] args) throws Exception {

        ArrayList<String> bantal = new ArrayList<String>();

        bantal.add(e:"merah");
        bantal.add(e:"kuning");

        Scanner inputUser = new Scanner(System.in);

        show_list(bantal);

        System.out.print(s:"Masukkan nama pemain : ");
        String name = inputUser.next();
        System.out.print(s:"Masukkan kekuatan player : ");
        int strength = inputUser.nextInt();
        System.out.print(s:"Masukkan nyawa player : ");
        int health = inputUser.nextInt();

        Player player1 = new Player(name, strength, health);

        while (true) {

            System.out.println(x:"1. Masukkan warna bantal");
            System.out.println(x:"2. Hapus warna bantal");
            System.out.println(x:"3. Update warna bantal");
            System.out.println(x:"4. Lanjut bertarung");
            System.out.print(s:"Pilih salah satu : ");
            int userChoice = inputUser.nextInt();

            if (userChoice == 1) {

                show_list(bantal);
                System.out.println(x:"Masukkan warna bantal : ");
                String colourInput = inputUser.next();
                bantal.add(colourInput);
                show_list(bantal);

            }
```



```
47              }
48              else if (userChoice == 2) {
49
50                  show_list(bantal);
51                  System.out.print(s:"Masukkan warna bantal yang ingin dihapus : ");
52                  String colourInput = inputUser.next();
53
54                  bantal.remove(colourInput);
55                  show_list(bantal);
56
57              }
58          // this.show_list();
59          //      }
60              else if(userChoice == 3) {
61
62                  show_list(bantal);
63                  System.out.print(s:"Masukkan warna bantal yang ingin diubah : ");
64                  String colourToUpdate = inputUser.next();
65
66                  System.out.print(s:"Masukkan warna yang baru : ");
67                  String colourInput = inputUser.next();
68
69                  ListIterator<String> iterator = bantal.listIterator();
70                  while (iterator.hasNext()) {
71                      String next = iterator.next();
72                      if (next.equals(colourToUpdate)) {
73                          //Replace element
74                          iterator.set(colourInput);
75                      }
76                  show_list(bantal);
77          }
```

```
78              }
79              else if (userChoice == 4) {
80
81                  System.out.println(x:"Generating enemy...");
82                  Thread.sleep(millis:2000);
83
84                  Enemy enemy1 = new Enemy();
85
86                  enemy1.show_stats();
87
88                  while (true) {
89
90                      System.out.println(x:"\n1. Pukul");
91                      System.out.println(x:"2. Lempar");
92                      System.out.print(s:"Pilih serangan : ");
93                      int attack = inputUser.nextInt();
94                      if (attack == 1) {
95                          player1.hitPillow(enemy1);
96                      }
97                      else if (attack == 2) {
98                          player1.throwPillow(enemy1);
99                      }
100
101                     enemy1.attacking(player1);
102
103                     if (player1.get_health() == 0 && enemy1.get_health() == 0 ) {
104                         System.out.println(x:"Seri..");
105                         break;
106                     }
107                     if (player1.get_health() == 0) {
108                         System.out.println(x:"Anda telah kalah");
109                         break;
110                     }
111                     else if (enemy1.get_health() == 0) {
112                         System.out.println(x:"Anda menang!");
113                         break;
114                     }
115                 }
116
117             }
118
119             System.out.print(s:"Lagi ? (y/n) : ");
120             String again = inputUser.next();
121             if (again.equals(anObject:"n")) {
122                 break;
123             }
```

```
5
7       public static final void show_list(ArrayList<String> list){
8
9           for (String item : list) {
0               System.out.println("Warna bantal : " + item);
1           }
2       }
3   }
4
```

13. ***ArrayList<String> bantal = new ArrayList<String>();***

    Membuat arraylist bertipe *String* dengan nama “bantal”.

14. ***bantal.add("merah");***
    bantal.add("kuning");

    Agar arraylist memiliki beberapa isi, kita tambahkan warna “merah” dan “kuning” menggunakan .add().

15. ***Scanner inputUser = new Scanner(System.in);***

    Membuat objek input agar pengguna bisa melakukan input nantinya.

16. ***show_list(bantal);***

    Menampilkan list warna dalam arraylist bantal yang telah kita buat dan tambahkan tadi.

17. ***System.out.print("Masukkan nama pemain : ");***
    String name = inputUser.next();
    System.out.print("Masukkan kekuatan player : ");
    int strength = inputUser.nextInt();
    System.out.print("Masukkan nyawa player : ");
    int health = inputUser.nextInt();
    Player player1 = new Player(name, strength, health);

    Meminta input *name, strength,* dan *health* dari pengguna sebagai atribut untuk objek player, lalu menginisialisasikan objek *player1* dengan *name, strength,* dan *health* yang telah dimasukkan tadi.

18. ***while (true) {***

    System.out.println("1. Masukkan warna bantal");
    System.out.println("2. Hapus warna bantal");
    System.out.println("3. Update warna bantal");
    System.out.println("4. Lanjut bertarung");
    System.out.print("Pilih salah satu : ");
    int userChoice = inputUser.nextInt();

    Memasuki perulangan, dan meminta input dari pengguna.

19. ***if (userChoice == 1) {***

```
show_list(bantal);
System.out.println("Masukkan warna bantal : ");
String colourInput = inputUser.next();
bantal.add(colourInput);
show_list(bantal);

}
```

Jika sebelumnya pengguna memasukkan nomor 1, maka pengguna akan diminta memasukkan warna bantal yang nanti akan ditambahkan ke dalam arraylist bantal.

20. ***else if (userChoice == 2) {***

```
show_list(bantal);
System.out.print("Masukkan warna bantal yang ingin dihapus : ");
String colourInput = inputUser.next();

bantal.remove(colourInput);
show_list(bantal);

}
```

Jika pengguna memasukkan angka 2, pengguna akan diminta memasukkan warna bantal yang ingin dihapus dari arraylist bantal. Lalu program akan menghapus bantal tersebut dari arraylistnya.

21. ***else if(userChoice == 3) {***

```
show_list(bantal);
System.out.print("Masukkan warna bantal yang ingin
diubah : ");
String colourToUpdate = inputUser.next();

System.out.print("Masukkan warna yang baru : ");
String colourInput = inputUser.next();

ListIterator<String> iterator = bantal.listIterator();
while (iterator.hasNext()) {
   String next = iterator.next();
   if (next.equals(colourToUpdate)) {
      //Replace element
      iterator.set(colourInput);
   }
show_list(bantal);
```

Jika pengguna memasukkan angka 3, maka pengguna akan diminta memasukkan warna bantal yang ingin diubah. Lalu pengguna akan diminta untuk mengubah ke warna apa. Selanjutnya program akan mengiterasi ke arraylist, menemukan warna sesuai yang pengguna mau, lalu mengubah warna tersebut dengan inputan dari pengguna.

22. ***else if (userChoice == 4) {***

```
System.out.println("Generating enemy...");
Thread.sleep(2000);

Enemy enemy1 = new Enemy();

enemy1.show_stats();

while (true) {

   System.out.println("\n1. Pukul");
   System.out.println("2. Lempar");
```

```
                System.out.print("Pilih serangan : ");
                int attack = inputUser.nextInt();
                if (attack == 1) {
                    player1.hitPillow(enemy1);
                }
                else if (attack == 2) {
                    player1.throwPillow(enemy1);
                }

                enemy1.attacking(player1);

                if (player1.get_health() == 0 && enemy1.get_health()
== 0 ) {
                    System.out.println("Seri..");
                    break;
                }
                if (player1.get_health() == 0) {
                    System.out.println("Anda telah kalah");
                    break;
                }
                else if (enemy1.get_health() == 0) {
                    System.out.println("Anda menang!");
                    break;
                }
            }

        }
```

Jika pengguna memasukkan angka 4, maka pengguna bersiap bertarung sebagai *player1.* Program akan membuat musuh secara random dan menampilkan status musuh tersebut. Lalu akan memasuki perulangan, dimana pengguna akan diminta memilih serangan bantal, yaitu memukul atau dilempar. Jika pilih memukul, maka akan menjalankan method *hitPillow()* dengan *enemy1* sebagai argumennya. Jika pilih dilempar, maka akan menjalankan method *throwPillow()* dengan *enemy1* sebagai argumennya. Karena memiliki argumen *enemy1,* maka musuh sudah pasti akan menerima serangan dan *health*-nya akan berkurang. Setelah itu, musuh akan otomatis

lngsung menyerang pengguna secara otomatis. Tidak lupa kita memberi pengkondisian, dimana jika *health* musuh mencapai 0, maka program akan mengeluarkan output “Anda menang!”. Lalu sebaliknya, jika *health* player mencapai 0, maka program akan mengeluarkan output “Anda kalah”. Dan jika *health* dari kedua sisi adalah 0 secara bersamaan, maka pertandingan berujung “Seri”.

23. ```
    System.out.print("Lagi ? (y/n) : ");
            String again = inputUser.next();
            if (again.equals("n")) {
               break;
            }
    ```

Blok kode ini adalah lanjutan dari perulangan di awal. Jika ingin lanjut lagi, maka program akan mengulang dari proses CRUD bantal. Jika tidak, program akan berhenti.

24. ```
    public static final void show_list(ArrayList<String> list){

        for (String item : list) {
                        System.out.println("Warna bantal : " + item);
                }
                }
    ```

Lalu di akhir program adalah pendeklarasian dari fungsi untuk menampilkan isi dari arraylist (show_list()). Fungsi ini akan menerima argumen arraylist bertipe string, lalu mengiterasi dan menampilkan item - item yang ada di dalam arraylist tersebut.

Output :

```
Masukkan nama pemain : adli
Masukkan kekuatan player : 200
Masukkan nyawa player : 2000
1. Masukkan warna bantal
2. Hapus warna bantal
3. Update warna bantal
4. Lanjut bertarung
Pilih salah satu : 4
Generating enemy...
Nama musuh : Musuh
Kekuatan : 138
HP : 1708

1. Pukul
2. Lempar
Pilih serangan : 2
Anda melempar dengan bantal...
Nyawa musuh : 1258

Musuh memukul dengan bantal...
Nyawa anda : 1672

1. Pukul
2. Lempar
Pilih serangan : 2
Anda melempar dengan bantal...
Nyawa musuh : 808

Musuh memukul dengan bantal...
Nyawa anda : 1344

1. Pukul
2. Lempar
Pilih serangan : 2
Anda melempar dengan bantal...
Nyawa musuh : 358

Musuh memukul dengan bantal...
Nyawa anda : 1016
```

```
1. Pukul
2. Lempar
Pilih serangan : 2
Anda melempar dengan bantal...
Nyawa musuh : 0

Musuh memukul dengan bantal...
Nyawa anda : 688
Anda menang!
```