

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL VIII
JAVA INHERITANCE (PEWARISAN) II



Disusun Oleh:
Muhammad Ali Akbar (105223021)

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN KOMPUTER
UNIVERSITAS PERTAMINA

2025

I. Pendahuluan

Laporan ini menjelaskan implementasi program sederhana berbasis Java menggunakan konsep dasar Pemrograman Berorientasi Objek (OOP). Program yang dibuat merupakan game petualangan teks, di mana pemain dapat melakukan login, registrasi, dan menjelajah dunia secara linear (maju/mundur). Konsep OOP yang digunakan meliputi enkapsulasi, pembuatan objek (instansiasi), dan pemanfaatan class.

II. Variabel

No	Kelas	Nama Variabel	Tipe Data	Sifat
1	User	username	String	Private
2	User	password	String	Private
3	Main	users	HashMap	Local Variable
4	Main	scanner	Scanner	Local Variable
5	Main	posisi	int	Local Variable
6	Main	random	Random	Local Variable

III. Constructor dan Method

No	Nama Method	Jenis	Fungsi
Class User			
1	<code>User(...)</code>	Constructor	Menginisialisasi objek User dengan username dan password.
2	<code>checkPassword(...)</code>	Method	Memvalidasi kecocokan password yang dimasukkan saat login.
Class Main			
3	<code>main(...)</code>	Method	Titik masuk program. Menangani menu utama, registrasi, login, dan gameplay.
4	<code>tampilkanEvent(...)</code>	Method (static)	Menampilkan event acak setelah pemain bergerak.

IV. Dokumentasi dan Pembahasan Code

Program ini ditulis dalam satu file Java dengan dua class utama: User dan GamePetualangan (yang berisi method main). Berikut pembahasan kode secara lebih rinci berdasarkan struktur program:

```
class User {  
    private String username;  
    private String password;  
  
    public User(String username, String password) {  
        this.username = username;  
        this.password = password;  
    }  
  
    public boolean checkPassword(String input) {  
        return this.password.equals(input);  
    }  
}
```

Penjelasan:

- Atribut username dan password diset private untuk menerapkan *enkapsulasi*. Hal ini mencegah akses langsung dari luar kelas.
- Constructor User(String, String) digunakan untuk membuat objek User dengan data username dan password yang dimasukkan saat registrasi.
- Method checkPassword(String) digunakan saat login. Password yang dimasukkan dibandingkan dengan yang tersimpan. Return true jika cocok.

```
GamePetualangan.java > ...  
import java.util.*;  
  
public class GamePetualangan {  
    Run | Debug | Run main | Debug main  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        Map<String, User> users = new HashMap<>();  
        Random random = new Random();  
    }  
}
```

Penjelasan:

- Scanner digunakan untuk membaca input dari pengguna.
- Map<String, User> menyimpan data akun user dengan key = username.
- Random digunakan untuk memilih event secara acak.

```
while (true) {
    System.out.println(x:"\n=== Game Petualangan ===");
    System.out.println(x:"1. Register");
    System.out.println(x:"2. Login");
    System.out.println(x:"3. Keluar");
    System.out.print(s:"Pilih: ");
    int menu = scanner.nextInt(); scanner.nextLine();
}
```

Penjelasan:

- Program menampilkan menu pilihan utama.
- Loop while(true) memastikan program berjalan terus hingga pengguna memilih keluar.

```
if (menu == 1) {
    System.out.print(s:"Username: ");
    String username = scanner.nextLine();
    System.out.print(s:"Password: ");
    String password = scanner.nextLine();
    if (users.containsKey(username)) {
        System.out.println(x:"Username sudah digunakan!");
    } else {
        users.put(username, new User(username, password));
        System.out.println(x:"Registrasi berhasil!");
    }
}
```

Penjelasan:

- Program meminta username dan password dari pengguna.
- Cek apakah username sudah terdaftar.
- Jika belum, buat objek User baru dan simpan di Map.

```
} else if (menu == 2) {
    System.out.print(s:"Username: ");
    String username = scanner.nextLine();
    System.out.print(s:"Password: ");
    String password = scanner.nextLine();

    User user = users.get(username);
    if (user != null && user.checkPassword(password)) {
        System.out.println(x:"Login berhasil!");
        int posisi = 0;
        while (true) {

```

Penjelasan:

- Setelah input username dan password, program memeriksa apakah user tersebut terdaftar dan passwordnya cocok.
- Jika login berhasil, variabel posisi diinisialisasi ke 0.
- Lalu pemain masuk ke mode petualangan.

```

while (true) {
    System.out.println("\nPosisi saat ini: " + posisi);
    System.out.println(x:"1. Maju");
    System.out.println(x:"2. Mundur");
    System.out.println(x:"3. Keluar");
    System.out.print(s:"Pilih: ");
    int aksi = scanner.nextInt();
}

```

Penjelasan:

- Pemain diberikan 3 opsi: maju, mundur, atau keluar.
- Posisi disesuaikan berdasarkan aksi pemain.

```

static void tampilkanEvent(Random random) {
    int event = random.nextInt(bound:3);
    if (event == 0) {
        System.out.println(x:">> Kamu menemukan item!");
    } else if (event == 1) {
        System.out.println(x:">> Kamu bertemu musuh!");
    } else {
        System.out.println(x:">> Tidak terjadi apa-apa.");
    }
}
}

```

Penjelasan:

- Method ini menerima objek `Random` dan menghasilkan angka 0–2.
- Masing-masing angka mewakili jenis kejadian:
 - 0: Menemukan item
 - 1: Bertemu musuh
 - 2: Tidak terjadi apa-apa
- Event hanya ditampilkan melalui `System.out.println()` karena belum ada sistem HP atau inventory.

- Pemain menghadapi event acak yang dipilih melalui `Random`.
- Ada 4 kemungkinan event yang bisa terjadi: menemukan item, bertemu musuh, terkena jebakan, atau tidak terjadi apa-apa.

- Event akan mempengaruhi kondisi pemain sesuai dengan logika di masing-masing class turunan.

V. Kesimpulan

Praktikum Modul VIII ini berhasil mengimplementasikan konsep dasar Pemrograman Berorientasi Objek (OOP) menggunakan bahasa Java melalui pembuatan program game petualangan berbasis teks. Meskipun program ini bersifat sederhana, ia telah mampu merepresentasikan prinsip-prinsip utama OOP seperti enkapsulasi, instansiasi objek, serta pemodelan kelas.

Konsep enkapsulasi diterapkan dengan baik dalam class User, di mana atribut username dan password dibuat private untuk menjaga keamanan data. Interaksi terhadap data tersebut dilakukan melalui method checkPassword, yang merupakan contoh penggunaan akses terbatas terhadap data internal objek. Hal ini menekankan pentingnya pembatasan akses langsung terhadap data, dan penggunaan method publik sebagai jembatan interaksi.

Struktur program juga telah memperlihatkan prinsip pemisahan tanggung jawab. Class User hanya bertugas menangani akun pengguna, sedangkan seluruh logika permainan dikendalikan dalam class utama. Dengan ini, struktur program menjadi modular, memudahkan pengembangan lebih lanjut di masa depan.

Mekanisme registrasi dan login memberikan pemahaman bagaimana sebuah aplikasi menyimpan dan memverifikasi data pengguna. Selain itu, fitur gameplay yang memungkinkan pemain untuk bergerak maju dan mundur, serta menghadapi event acak (seperti menemukan item, bertemu musuh, atau tidak terjadi apa-apa), menunjukkan pemanfaatan logika kondisional dan struktur perulangan secara efektif.