LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK
INHERITANCE

Disusun oleh:
Rayhan Surya Destian (105222024)

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN KOMPUTER
PEMROGRAMAN BERORIENTASI OBJEK
UNIVERSITAS PERTAMINA
2023/2024

I. Pendahuluan

Program ini adalah simulasi game petualangan dungeon di mana pemain dapat berinteraksi melalui menu untuk melakukan login, register, menjelajahi dungeon, player bisa bertarung dengan musuh dan juga mendapatkan item yang bisa digunakan.

II. Variabel

Tabel variabel untuk kelas `Game`, `Player`, `Item`, dan `Enemy` dalam program:

No	Nama Variabel	Tipe Data	Fungsi
1	accountManager	AccountManager	Mengelola login dan register player
2	player	Player	Menyimpan data dan status pemain dalam game
3	isRunning	boolean	Mengontrol loop utama game
4	id	String	Identifikasi unik untuk item dan musuh
5	name	String	Nama dari item atau musuh
6	description	String	Deskripsi item
7	type	String	Tipe item
8	stats	String	Statistik yang ditawarkan item
9	value	int	Nilai efek dari item
10	health	int	Darah musuh atau pemain
11	maxHealth	int	Darah maksimum musuh atau pemain
12	damage	int	Damage yang diberikan oleh musuh
13	experience	int	Experience yang diperoleh dari mengalahkan musuh

III. Constructor dan Method

Tabel constructor dan method yang signifikan dalam program:

No	Nama Method	Jenis Method	Fungsi
1	Game()	Constructor	Membuat objek Game dengan inisialisasi variabel `accountManager`, `player`, dan `isRunning`
2	run()	Procedural	Menjalankan loop utama game untuk interaksi menu utama
3	login(Scanner)	Functional	Memproses login pengguna
4	register(Scanner)	Procedural	Memproses registrasi pengguna
5	gameMenu(Scanner)	Procedural	Menampilkan dan mengelola menu permainan setelah login
6	venture(Scanner)	Procedural	Mengelola eksplorasi dungeon oleh pemain
7	encounter(Scanner)	Procedural	Mengelola pertemuan secara acak dengan musuh atau temuan item
8	combat(Scanner, Enemy)	Procedural	Melaksanakan pertarungan antara pemain dan musuh
9	checkInventory()	Procedural	Menampilkan dan mengelola inventaris pemain
10	Item(...)	Constructor	Membuat objek Item dengan atribut seperti nama, deskripsi, dan lainnya
11	use(Player)	Functional	Menerapkan efek item pada pemain
12	Enemy(...)	Constructor	Membuat objek Enemy dengan atribut seperti health, damage, dan pengalaman
13	takeDamage(int)	Functional	Mengurangi health musuh berdasarkan damage yang diterima
14	isAlive()	Functional	Memeriksa apakah musuh masih hidup atau tidak

IV. Dokumentasi dan Pembahasan Code

Program game dungeon ini dirancang menggunakan paradigma pemrograman berorientasi objek (OOP), yang memfasilitasi pembagian dan manajemen tanggung jawab melalim pemisahan fungsi-fungsi dalam kelas-kelas yang terdefinisi dengan jelas.

Inisialisasi dan Manajemen Pengguna: Dalam kelas `Game`, constructor digunakan untuk menginisialisasi objek-objek penting seperti `accountManager` dan `player`. Variabel `isRunning` diatur untuk mengontrol loop utama permainan, sementara method `login()` dan `register()` menangani interaksi pengguna untuk proses autentikasi.

Navigasi dan Encounter Dungeon: Fungsi `venture(Scanner)` memberikan simulasi adventure dungeon oleh pemain, di mana tindakan pemain dapat memicu pertemuan acak yang diatur oleh method `encounter(Scanner)`. Mekanisme ini menunjukkan penggunaan efektif dari polymorphism dan inheritance melalui kelas abstrak `Item` dan `Enemy`, yang memungkinkan perluasan fungsionalitas secara dinamis tanpa perlu mengubah kode yang telah ada.

Mekanisme Pertarungan: Dalam pertarungan, logika interaksi pengguna seperti serangan, penggunaan item, dan pelarian diatur melalui `combat(Scanner, Enemy)`. Metode ini menunjukkan bagaimana manajemen state dinamis dapat diimplementasikan dalam OOP untuk responsif terhadap aksi pengguna. Implementasi ini juga menunjukkan bagaimana aplikasi dapat diuji dan diperluas dengan mudah berkat modularitas yang dihasilkan oleh pendekatan berorientasi objek.

V. Kesimpulan

Program game adventure dungeon ini menggabungkan prinsip-prinsip OOP untuk membangun aplikasi yang modular dan bisa dibilang cukup mudah dikelola. Penggunaan kelas abstrak, inheritance, dan polymorphism memberikan kerangka kerja yang lumayan solid untuk pengembangan lebih lanjut. Walaupun memiliki banyak area yang mungkin dapat Saya perbaiki dan tambah, struktur yang sudah Saya bangun memberikan dasar yang kuat untuk iterasi selanjutnya dan peningkatan fungsionalitas.