

TUGAS PEMROGRAMAN II



CLI OOP

Oleh:

Muhammad Naufal Khalish

NIM. 2410817110004

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
NOVEMBER 2025**

DAFTAR ISI

DAFTAR ISI.....	2
DAFTAR GAMBAR.....	3
DAFTAR TABEL.....	4
SOAL	5
A. Source Code.....	5
B. Output Program.....	13
C. Pembahasan.....	16
LINK GITHUB.....	18

DAFTAR GAMBAR

Gambar 1 Input Pilihan Nomor 1.....	13
Gambar 2 Input Username	13
Gambar 3 Output Setelah Input Username	13
Gambar 4 Input Pilihan Nomor 2.....	14
Gambar 5 Memilih deskripsi salah satu senjata.....	14
Gambar 6 Output salah satu deskripsi senjata	14
Gambar 7 Input Pilihan Nomor 3.....	15
Gambar 8 Memasukkan Feedback.....	15
Gambar 9 Output Setelah Mengisi Feedback	15

DAFTAR TABEL

Table 1 File Skill.java	5
Table 2 File Staff.java.....	5
Table 3 File NoviceFireStaff.java.....	6
Table 4 File NoviceFireSkill.java	7
Table 5 File AdeptHolyStaff.java	7
Table 6 File AdeptHolySkill.java	7
Table 7 File ExpertGreatStaff.java	8
Table 8 File ExpertGreatSkill.java	8
Table 9 File UI.java	8
Table 10 File weaponMenu.java.....	9
Table 11 File Main.java	11

SOAL

Menggunakan prinsip Abstract class, Interface dan Composition. Buatkan CLI template menggunakan Bahasa Java. Tidak boleh menggunakan Library apapun di luar base Java.

Spec:

1. CLI harus memiliki halaman yang bisa dipilih di menu utama
2. CLI harus memiliki dua fitur memilih menu, scan input user dan menampilkannya.
3. Menggunakan prinsip OOP penyembunyian, main static tidak boleh menjadi GOD class dan harus DUMB
4. Buat report berisikan alasan kalian menggunakan teknik kalian dan insight yang kalian dapatkan

A. Source Code

Table 1 File Skill.java

Skill.java
package Tugas.AlbionOnline; public interface Skill { String getSkillName(); String useSkill(); }

Table 2 File Staff.java

Staff.java
package Tugas.AlbionOnline; public abstract class Staff { protected String name; protected String description; protected int baseDamage; protected Skill skill; public Staff(String name, String description,

```

int baseDamage, Skill skill){
    this.name = name;
    this.description = description;
    this.baseDamage = baseDamage;
    this.skill = skill;
}

public String getName() {
    return name;
}
public String getDescription() {
    return description;
}
public int getBaseDamage() {
    return baseDamage;
}

public String getSkillInfo() {
    return skill.getSkillName() + " → " +
skill.useSkill();
}
}

```

Table 3 File NoviceFireStaff.java

NoviceFireStaff.java

```

package Tugas.AlbionOnline;

public class NovicesFireStaff extends Staff {
    public NovicesFireStaff(){
        super("Novice's Fire Staff",
              "Staff api dasar untuk pemula",
              10,
              new NoviceFireSkill());
    }
}

```

Table 4 File NoviceFireSkill.java

NoviceFireSkill.java
package Tugas.AlbionOnline;
public class NoviceFireSkill implements Skill {
public String getSkillName(){ return "Fire Bolt"; }
public String useSkill(){ return "Menembakkan api kecil ke musuh."; }
}

Table 5 File AdeptHolyStaff.java

AdeptHolyStaff.java
package Tugas.AlbionOnline;
public class AdeptHolyStaff extends Staff {
public AdeptHolyStaff(){
super("Adept's Holy Staff",
"Staff Suci dengan kemampuan penyembuhan",
14,
new AdeptHolySkill());
}
}

Table 6 File AdeptHolySkill.java

AdeptHolySkill.java
package Tugas.AlbionOnline;
public class AdeptHolySkill implements Skill {
public String getSkillName(){ return "Holy Bless"; }
public String useSkill(){ return "Memulihkan HP sekutu."; }
}

Table 7 File ExpertGreatStaff.java

ExpertGreatStaff.java
package Tugas.AlbionOnline;
public class ExpertGreatStaff extends Staff {
public ExpertGreatStaff(){
super("Expert's Great Staff",
"Staff Besar dengan kekuatan
energi",
20,
new ExpertGreatSkill());
}
}

Table 8 File ExpertGreatSkill.java

ExpertGreatSkill.java
package Tugas.AlbionOnline;
public class ExpertGreatSkill implements Skill {
public String getSkillName(){ return "Soul
Beam"; }
public String useSkill(){ return "Menembakkan
sinar energi besar."; }
}

Table 9 File UI.java

UI.java
package Tugas.AlbionOnline;
public class UI {
public void showPage(String content){
System.out.println("=====
=====");
System.out.println(" ALBION ONLINE
- STAFF INFO");
System.out.println("=====

```

=====\\n");
    System.out.println(content);

System.out.println("\\n=====
=====");
    System.out.println("
> __<
                ");

System.out.println("=====
=====\\n");

}
}

```

Table 10 File weaponMenu.java

weaponMenu.java
<pre> package Tugas.AlbionOnline; import java.util.Scanner; public class weaponMenu { private UI ui; private Scanner input; private Staff staff1 = new NovicesFireStaff(); private Staff staff2 = new AdeptHolyStaff(); private Staff staff3 = new ExpertGreatStaff(); public weaponMenu(UI ui, Scanner input){ this.ui = ui; this.input = input; } public void start(){ while(true){ ui.showPage("1. Masukkan Username\\n" + "2. Lihat Deskripsi Senjata\\n" + "3. Pesan Feedback\\n" + "4. Keluar"); } } }</pre>

```

    );
    System.out.print(">> ");
    String choice = input.nextLine();

    switch(choice) {
        case "1": inputUsername(); break;
        case "2":
selectWeaponDescription(); break;
        case "3": feedbackMessage();
break;
        case "4": return;
        default: break;
    }
}

private void inputUsername() {
    ui.showPage("Masukkan username:");
    System.out.print(">> ");
    String username = input.nextLine();
    ui.showPage("Halo " + username + "! (ENTER untuk kembali)");
    input.nextLine();
}

private void selectWeaponDescription(){
    ui.showPage(
        "Pilih Senjata:\n\n" +
        "1. " + staff1.getName() +
"\n" +
        "2. " + staff2.getName() +
"\n" +
        "3. " + staff3.getName() +
"\n"
    );
    System.out.print(">> ");
    String choice = input.nextLine();

    switch(choice) {
        case "1": showStaffDetail(staff1);
break;
        case "2": showStaffDetail(staff2);
break;
    }
}

```

```

        case "3": showStaffDetail(staff3);
break;
default:
    ui.showPage("Input tidak valid.");
    input.nextLine();
}
}

private void showStaffDetail(Staff s){
    ui.showPage(
        "Nama : " + s.getName() +
        "\nDeskripsi : " +
        s.getDescription() +
        "\nBase Damage : " +
        s.getBaseDamage() +
        "\nSkill : " +
        s.getSkillInfo() +
        "\n\n(ENTER untuk
kembali)"
    );
    input.nextLine();
}

private void feedbackMessage(){
    ui.showPage("Tulis feedback:");
    System.out.print(">> ");
    String text = input.nextLine();
    ui.showPage("Feedback diterima:\n\n" +
text + "\n\n(ENTER untuk kembali)");
    input.nextLine();
}
}

```

Table 11 File Main.java

Main
<pre> package Tugas.AlbionOnline; import java.util.Scanner; public class Main { public static void main(String[] args) { UI ui = new UI(); Scanner input = new Scanner(System.in); } } </pre>

```
        weaponMenu menu = new weaponMenu(ui,  
input);  
        menu.start();  
    }  
}
```

B. Output Program

```
=====
ALBION ONLINE - STAFF INFO
=====

1. Masukkan Username
2. Lihat Deskripsi Senjata
3. Pesan Feedback
4. Keluar

>_<

>> 1
```

Gambar 1 Input Pilihan Nomor 1

```
=====
ALBION ONLINE - STAFF INFO
=====

Masukkan username:

>_<

>> Khalish|
```

Gambar 2 Input Username

```
=====
ALBION ONLINE - STAFF INFO
=====

Halo Khalish! (ENTER untuk kembali)

>_<
```

Gambar 3 Output Setelah Input Username

```
=====
ALBION ONLINE - STAFF INFO
=====

1. Masukkan Username
2. Lihat Deskripsi Senjata
3. Pesan Feedback
4. Keluar

=====
>_<
=====

>> 2
```

Gambar 4 Input Pilihan Nomor 2

```
=====
ALBION ONLINE - STAFF INFO
=====

Pilih Senjata:

1. Novice's Fire Staff
2. Adept's Holy Staff
3. Expert's Great Staff

=====
>_<
=====

>> 1
```

Gambar 5 Memilih deskripsi salah satu senjata

```
=====
ALBION ONLINE - STAFF INFO
=====

Nama : Novice's Fire Staff
Deskripsi : Staff api dasar untuk pemula
Base Damage : 10
Skill : Fire Bolt → Menembakkan api kecil ke musuh.

(ENTER untuk kembali)

=====
>_<
=====
```

Gambar 6 Output salah satu deskripsi senjata

```
=====
      ALBION ONLINE - STAFF INFO
=====

1. Masukkan Username
2. Lihat Deskripsi Senjata
3. Pesan Feedback
4. Keluar

>_<

>> 3
```

Gambar 7 Input Pilihan Nomor 3

```
=====
      ALBION ONLINE - STAFF INFO
=====

Tulis feedback:

>_<

>> Semangat Yahh
```

Gambar 8 Memasukkan Feedback

```
=====
      ALBION ONLINE - STAFF INFO
=====

Feedback diterima:
Semangat Yahh
(ENTER untuk kembali)

>_<
```

Gambar 9 Output Setelah Mengisi Feedback

C. Pembahasan

Alur Program:

Program dimulai dari file Main.java. di dalam file ini hanya terdapat tugas sederhana yaitu menginisialisasi objek yang akan digunakan di program objek antarmuka (UI), objek untuk membaca input (Scanner), dan objek pengelola menu (weaponMenu). Setelah semua objek nya tersedia, Main hanya memanggil menu.start(). Karena Main tidak berisi logika lain selain memulai aplikasi. Dikategorikan sebagai DUMB class, sesuai aturan bahwa Main tidak boleh menjadi GOD class.

Selanjutnya alur berlanjut ke WeaponMenu.java, yang berfungsi sebagai controller. Kelas ini mengatur alur interaksi pengguna, seperti menampilkan menu utama, menerima input pilihan menu dan mengarahkan proses ke fungsi yang tepat. Weapon tidak membuat tampilan namun memanggil UI untuk menampilkan halaman. Hal ini weaponMenu bergantung pada UI untuk output tampilan.

Pada weaponMenu memiliki objek Staff. Yaitu NovicesFireStaff.java, AdeptHolyStaff.java, ExpertGreatStaff.java ketiga objek ini bukan diwariskan dari weaponMenu, tetapi Composition, artinya weaponMenu hanya menggunakan objek-objek itu ditampilkan informasinya.

Setiap staff adalah turunan dari abstract class Staff, yang mendefinisikan struktur umum senjata yaitu nama, deskripsi , base damage, dan informasi skill. Namun kelas staff tidak tahu cara menampilkan skill, kelas staff hanya mendeklarasikan method getSkillInfo() yang harus diimplementasikan di kelas turunannya atau konkret. Skill itu sendiri ditetapkan melalui interface Skill, sehingga setiap skill dapat memiliki implementasinya masing-masing. Inilah bentuk Interface Polymorphism.

Alasan Menggunakan Prinsip ini :

Dalam program ini , penggunaan prinsip Abstract Class, Interface, Composition dan Single Responsibility Principle diterapkan karena struktur kodennya yang rapi, dan fleksibel saat dilakukan pengembangan. Kelas Staff dibuat sebagai abstract class karena seluruh jenis senjata memiliki karakteristik dasar yang sama seperti nama, deskripsi, dan nilai damage, namun masing-masing memiliki keunikan yang berbeda. Dengan abstraksi ini, seluruh kelas turunan cukup mengisi detail perbedaannya tanpa menulis ulang struktur yang sama. Sementara itu, konsep *interface* digunakan pada bagian Skill untuk menetapkan aturan umum perilaku skill tanpa membatasi bagaimana bentuk

implementasinya. Hal ini memungkinkan setiap senjata memiliki sistem skill yang berbeda-beda, tetapi tetap berada dalam pola yang seragam. Di sisi lain, *composition* diterapkan karena objek-objek dalam program saling memiliki keterkaitan langsung secara fungsi, misalnya WeaponMenu memiliki objek UI untuk menampilkan tampilan dan objek Staff untuk ditampilkan informasinya, serta setiap Staff memiliki sebuah Skill sebagai kemampuan khususnya.

Selain itu, program ini juga menerapkan *Single Responsibility Principle* (SRP) dengan memberikan satu tanggung jawab yang jelas pada setiap kelas. Kelas UI hanya bertugas mengatur tampilan layar dan tidak mengelola logika menu. Kelas WeaponMenu berfungsi khusus untuk menangani alur interaksi pengguna. Kelas Staff hanya mendefinisikan atribut dan karakteristik dasar senjata, sementara detail skill didefinisikan pada kelas-kelas skill yang berbeda.

Insight :

Setelah mengerjakan kode yang telah saya buat penggunaan abstract class untuk staff menunjukkan bahwa ada kesamaan konsep antar semua senjata, tetapi cara perlakuan atau detail isinya yang berbeda. Dengan ini, tidak memaksakan semua staff untuk memiliki bentuk yang sama persis, namun tetap memiliki hubungan melalui kerangka dasar.

Sementara itu, Interface pada Skill bagaimana menetapkan perilaku tanpa tanpa menbatasi cara implementasinya. Kalau di analogikan “Semua staff harus memiliki skill , tapi skill yang bekerja itu terserah senjatanya”

Composition pada hubungan antar kelas , misalnya weaponMenu menggunakan UI dan objek Staff menekankan bahwa objek-objek seharusnya bekerja sama , bukan berdiri sendiri

Kemudian dengan menerapkan Single Responsibilty Principle (SRP), setiap kelas hanya menangani satu hal, sehingga ketika terjadi perubahan spesifik, hanya kelas tersebut yang perlu disesuaikan tanpa merambat ke kode lain.

LINK GITHUB

[Link Github](#)